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

Centre Universitaire Belhadj BOUCHAIB d'Aïn-Témouchent

Institut des Sciences
Département des Mathématiques et de l'Informatique

Mémoire

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

Option :

Réseaux et Ingénierie des Données (RID)

Présenté par :
M^r Abdallah MOHAMMED CHAABANI
M^r Mohamed Amine HAMAIDA

Système de Détection D'intrusions Informatiques et la selection des caractéristiques

Encadrant :
Dr Fatima Zohra BELGRANA
Maitre de Conférences "B" au C.U.B.B.A.T.

Soutenu le : 29 Juin 2020

Devant le jury composé de :

Président :	*Dr* Mohamed Lamine BENOMAR (MCB)	C.U.B.B.A.T.
Examinateur :	*Dr* Mohamed Hakim BENDIABDALLAH (MCB)	C.U.B.B.A.T.

Année universitaire 2019/2020

Remerciements

Tout d'abord, nous remercions le bon dieu tout-puissant qui nous a guidé dans nos pas, qui nous a donné le courage, la volonté, et la patience pour surmonter les épreuves que nous avons rencontrées tout au long de nos vies.

Nous tenant à exprimer nos sincères gratitudes et reconnaissances à notre encadrant Mme Fatima Zohra BELGRANA maître de conférences au CUBBAT pour avoir accepté de diriger ce travail, pour son aide, sa disponibilité et ses conseils.

Nous remercions les examinateurs d'avoir accepté d'examiner ce travail.

Enfin nous remercions toute personne ayant contribué de près ou loin à la réalisation de ce travail.

HAMAIDA & MOHAMMED CHAABANI

Dédicaces

À mes chers parents Pour leurs soutiens morals et financiers durant mes études, à toute ma famille proche soit-elle ou lointaine, et à tous mes fidèles amis, à tous ceux que j'aime, et à tous ceux qui me sont chers.

Mohamed Amine HAMAIDA

Je dédie ce travail à mes très chers parents, je vous dois ce que je suis aujourd'hui grâce à votre amour, à votre patience et vos innombrables sacrifices. Que ce modeste travail soit pour vous une petite compensation et reconnaissance envers ce que vous avez fait d'incroyable pour moi. Que Dieu le tout-puissant vous préserve et vous procure santé et longue vie.

Abdallah MOHAMMED CHAABANI

Sommaire

Table des figures

Liste des tableaux

Liste des acronymes

ACL : Access Control List
ACP : Analyse en Composantes Principales
ACPN : Analyse en Composante Principale à Noyau
ADL : Analyse Discriminante Linéaire
AP : Apprentissage Profond
ARG : Algorithme de Recherche Gravitationnelle
BDDA : base de données d'apprentissage
BDDT : base de données test
CNN : Condensed Nearest Neighbor
FN : Faux négatifs
FP : Faux positifs
HIDS : Host-Based IDS
IA : Intelligence Artificiel
IAD : Intelligence Artificiel Distribué
IBK : Instance based K
IDS : Intrusion Detection System
JSP : Java Server Pages
K-PPV : K Plus Proches Voisins
MAE : Machines d'Apprentissage Extrême
MVS : Machines à Vecteurs de Support
NIDS : Network Intrusion Detection System
PMC : Perceptron Multicouche
PPVC : Plus Proches Voisins Condensés
RL : Régression Logistique
RNA : Réseaux de Neurones Artificiels
RNP : Réseau de Neurones Probabiliste
RNRG : Réseaux de Neurones de Régression Généralisée
RPD : Résolution de problème distribuée
RPV : Réseaux Privés Virtuels
SDI : Système de Détection d'Intrusion
SDIH : Système de Détection d'Intrusion basé Hôte
SDIR : Système de Détection d'Intrusion Réseau
SMA : Systèmes Multi-Agents
SSI : Sécurité des Systèmes d'Informations
TAM : Taux des attaque manquées
TFA : Taux des fausses alertes
TVP : Taux des vrais positifs
VN : Vrai négatifs
VP : Vrai positifs

Introduction générale

L'internet est parmi les inventions les plus importantes du 20e siècle qui ont affecté nos vies et qui ne cessent d'évoluer. Ce phénomène nous a pris par surprise, il a changé toute perspective que nous avions sur nos vies quotidiennes et sur nos habitudes. Il a bouleversé nos modes de communication et interactions les uns avec les autres. Il a changé aussi la façon dont nous exerçons nos fonctions ou même apprécions nos passe-temps.

L'internet a rendu tout plus simple et plus accessible, mais tout cela à un prix que beaucoup de gens ont dû payer durement, ces gens sont ceux qui ont souffert de la cybercriminalité. Ce problème a poussé beaucoup personnes à remettre en question leurs décisions dans ce nouveau monde motivant ainsi les chercheurs à essayer de trouver des moyens pour sécuriser les réseaux et les systèmes. Tout cela a conduit par la suite à la création d'une nouvelle discipline appelée la sécurité des systèmes d'information qui vise à protéger les internautes contre les cyberattaques et les personnes malveillantes derrière elles.

La sécurité est cependant réalisée grâce à la mise en œuvre de multiples mécanismes, tels que les pare-feux qui visent à filtrer le trafic afin de garantir un réseau pur de tout signe d'acte malveillant, les proxys et les réseaux privés virtuels où le but de ces outils est de créer un tunnel sécurisé entre l'utilisateur et la destination souhaitée.

Hélas ces outils ont leurs limites et sont incapables d'arrêter toutes les attaques à cause de leur constante évolution, sans oublier les outils développés par les pirates informatiques afin de contourner les systèmes de sécurité mis en place. Ces limitations ont incité les développeurs à trouver une nouvelle approche pour lutter contre ces personnes malintentionnées ; ce fut alors la naissance d'un nouveau système dit Système de Détection d'Intrusion (SDI). Cet outil est le plus efficace pour protéger un réseau ou un système informatique contre les attaques avant même qu'elles ne commencent. Cela est dû grâce à sa capacité à reconnaître les intrusions ou les tentatives d'intrusion via la surveillance des comportements anormaux des utilisateurs ou par la reconnaissance d'attaque à partir du flux des données du réseau. Mais il faudrait toutefois noter l'importance des autres mécanismes de sécurité afin de renforcer cette dernière.

Le processus de détection d'intrusion est basé sur la collecte d'informations en observant un certain nombre d'événements afin que le système puisse analyser et traiter les données collectées. Ces deux opérations ont un obstacle que toute approche proposée dans ce domaine doit prendre en considération et tenter de le surpasser. Cette limitation est le temps. Toute opération effectuée afin de détecter les intrusions doit être en parallèle avec le trafic réseaux

or le SDI doit être le plus rapide possible. L'objectif principal de notre travail est de réaliser un Système de Détection d'Intrusion Réseau (SDIR) fiable avec de bonnes performances de détection et un temps de traitement et d'analyse réduit. Notre SDIR devrait être capable d'être déployé en temps réel.

Nos approches proposées sont basées principalement sur des algorithmes dérivant de l'Intelligence Artificiel IA. Dans un premier temps nous appliquons l'algorithme des k Plus Proches Voisins (K-PPV). Ce dernier est gourmand en temps de calcul or nous optons pour un autre algorithme connu sous le nom des Plus Proches Voisins Condensés (PPVC), afin de réduire la base d'apprentissage et optimiser en temps et en performances.

Notre problème est physiquement distribué, or l'utilisation des SMAs est évidente dans ce travail où l'aspect temporelle joue un rôle crucial, mais afin de donner une vie à ce système et atteindre notre objectif principal nous avons fait appel à une nouvelle notion. Il s'agit de la sélection des caractéristiques qui est la clé de ce travail, l'élimination des attributs non essentielle pour la classification a pu améliorer le temps d'analyse et de précision.

Notre mémoire est organisé en quatre chapitres, ou dans le premier chapitre nous présentons quelques notions de base sur la sécurité informatique. Ce qui nous permet par la suite d'introduire dans le deuxième chapitre un de ses axes fondamentale qui sont les systèmes de détection d'intrusion, a savoir leurs fonctionnement ainsi que leur classification selon différents critères tels que la source d'information, les méthodes de détection et le comportement âpres la détection.

Le troisième chapitre est dédié à un état de l'art sur les différentes techniques utilisées en vue d'une réalisation d'un SDI dérivant généralement de l'intelligence artificielle, telles que les réseaux de neurones, quelque méthodes classification sans oublier l'Intelligence Artificiel Distribué IAD et plus particulièrement les systèmes multi-agents. Nos approches, application et résultats obtenus font sujets du quatrième chapitre. Dans ce dernier nous détaillons les différentes étapes nécessaires à la réalisation de notre SDIR à partir du prétraitement, méthodes et données utilisées jusqu'aux simulations. Nous présentons également quelques comparaisons afin de prouver l'efficacité des approches proposées.

Chapitre 1

Sécurité informatique

1.1 Introduction

Quand l'utilisation des mainframes (des ordinateurs puissants en matière de traitement, qui sert d'unité centrale) était essentielle pour accéder aux réseaux, il était difficile de trouver des méthodes pour attaquer ses systèmes à cause de leur accès limité mais quand le réseautage a commencé dans les années 80, les réseaux sont devenus plus accessibles à n'importe quel individu, cette mise en réseau à changer l'architecture des réseaux, ou au lieu d'avoir un serveur central, plusieurs appareils distribués capables de traiter des informations sont maintenant utilisés. Cette distribution a rendu les réseaux plus vulnérables à des attaques qui visent à perturber la confiance entre l'utilisateur et son système, ce qui a poussé les développeurs à trouver des solutions pour sécuriser les différents systèmes et assurer leur fiabilité. Afin de garantir cette protection, plusieurs mécanismes sont utilisés.

Nous avons consacré ce chapitre pour discuter les différentes méthodes et techniques mises en œuvre pour arrêter les attaques et protéger ainsi le système.

1.2 Définitions

La sécurité informatique est le processus qui consiste à protéger les systèmes, les réseaux et les programmes contre des actes malveillants, ces actes visent généralement à accéder, modifier ou détruire des informations sensibles [Cisco, 2020a].

La sécurité Informatique est constituée de deux axes principaux :

a) **La cryptographie :** est la science des codes secrets et l'art de créer des cryptogrammes, de telle façon que ces derniers sont compris uniquement par leur destinataire légitime [Zimmermann, 1998] La cryptographie vise généralement cinq objectifs principaux :

- **La confidentialité :** consiste à rendre l'information lisible que par les personnes autorisées à accéder à la ressource.
- **L'intégrité :** c'est-à-dire que les données n'ont pas été altérées durant la communication (de manière fortuite ou intentionnelle).
- **L'authentification :** consiste à garantir pour chacun des correspondants que son partenaire est bien celui qu'il croit être (assurer l'identité d'un utilisateur).
- **La disponibilité :** garantir l'accès à un service ou à des ressources, c'est-à-dire permettre de maintenir le bon fonctionnement du système.
- **La non-répudiation :** c'est garantir qu'aucun des correspondants ne pourra nier la transaction

b) **Le BDA (Before During and After) :** ces trois termes décrivent les étapes que le système peut exécuter lors d'une attaque :

- **Before (avant) :** c'est les mécanismes à mettre en place pour essayer de rendre les accès non autorisés impossibles, cela est réalisé par la mise en place des Pare-feu, SDI et SPI.
- **During (pendant) :** cette étape consiste à arrêter l'attaque ou de réduire les dégâts dans la mesure du possible.
- **After (après) :** c'est l'ensemble des étapes à suivre pour restaurer le système et traiter les dégâts laissés par l'attaque telle que les virus et/ou les backdoor cela est réalisé à l'aide de l'antivirus. Il s'agit aussi de retracer l'attaque jusqu'au responsable et le poursuivre en justice.

1.3 Les attaques informatiques

Une attaque est une action qui exploite une vulnérabilité d'un système informatique (système d'exploitation, logiciel ou bien même de l'utilisateur) à des fins non connues par l'exploitation du système et qui sont généralement préjudiciables [Pillou et Bay,2016].
Pour qu'une attaque soit accomplie un objectif doit être établi, parmi ces motivations nous citons :

- obtenir un accès au système ;
- voler des informations ;
- troubler le bon fonctionnement d'un service ;
- utiliser le système de l'utilisateur comme un « rebond » pour une attaque ;
- utiliser les ressources du système de l'utilisateur.

1.3.1 Technique d'attaque

Toute attaque informatique suit un ensemble d'étapes pour atteindre son objective [Burgermeister et Krier, 2006].

- la reconnaissance : cette étape vise à récolter le maximum d'informations possibles sur la cible ;
- le scan : l'attaquant doit trouver des failles en rassemblant les informations obtenues lors de la reconnaissance ;
- l'accès et le maintien : à ce niveau l'attaquant exploit la faille trouver pour accéder au système cible et essayer de maintenir son emprise du système le plus longtemps possible si c'est nécessaire ;
- la couverture : l'étape finale consiste à couvrir les traces pour éviter d'éveiller les soupçons d'une compromission tout en maintenant l'accès à la cible.

1.3.2 Catégorie des attaques

Il existe quatre catégories d'attaques [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 ;
- **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 ;
- **Attaque par modification :** il s'agit d'une attaque portée à l'intégrité, tel 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 ;
- **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.

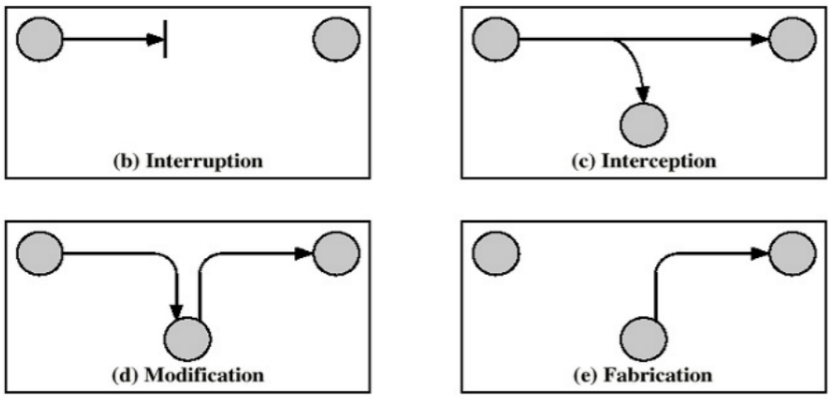


Figure 1.1 – Catégories des attaques [Ritambhara, 2020]

1.3.3 Types d'attaques

Plusieurs techniques d'attaque sont utilisées par les pirates informatiques, ces attaques peuvent être regroupées en deux familles différentes [Securiteinfo, 2006] :

a) **Les attaques directes :** cette technique est utilisée par la plupart des attaquants débutants, car c'est la plus simple.
Le hacker attaque directement sa victime à partir de son ordinateur, en effet les programmes d'attaque (hack) qu'ils utilisent ne sont que faiblement paramétrable, et un grand nombre de ces logiciels envoient directement les paquets à la victime.

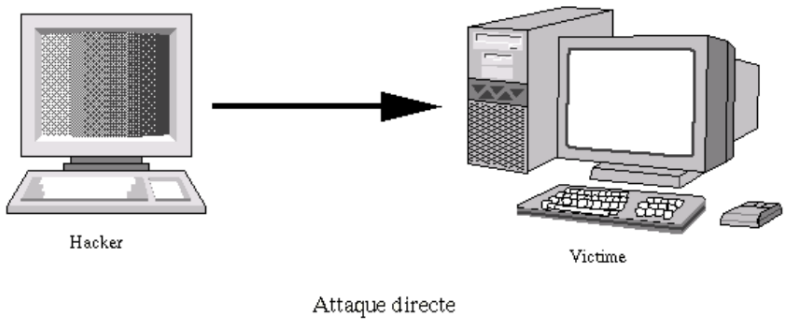


Figure 1.2 – Attaque directe [Arnaud, 2002]

b) **Les attaques indirectes :**

- **Par rebond :** le principe est simple, les paquets d'attaques sont envoyés à l'ordinateur intermédiaire, qui répercute l'attaque vers la victime. D'où le terme par rebond. Cette méthode permet de masquer l'identité du hacker et utiliser les ressources de l'ordinateur intermédiaire.

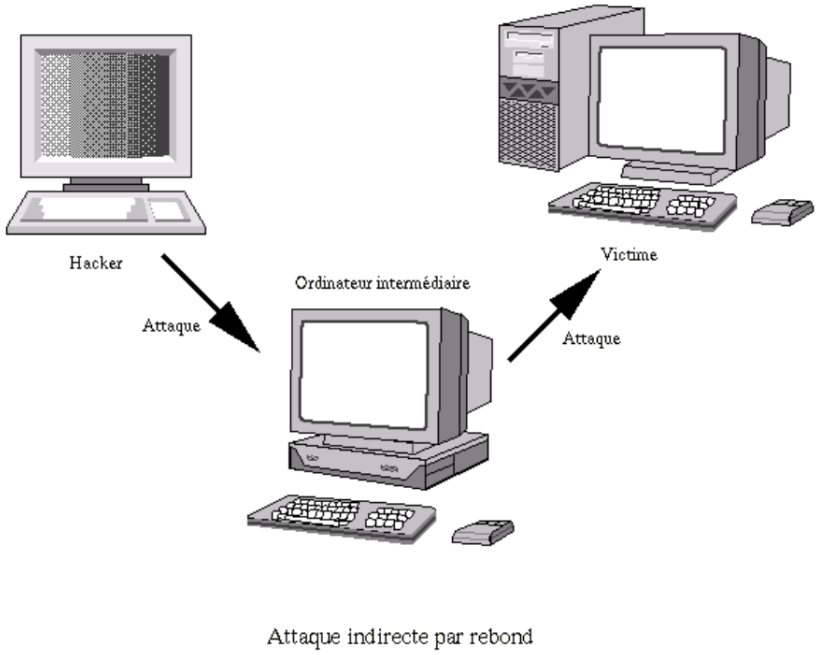


Figure 1.3 – Attaque indirecte par rebond [Arnaud, 2002]

- **Par réponse :** la similarité de cette attaque avec l'attaque par rebond offre les mêmes avantages. 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 la victime.

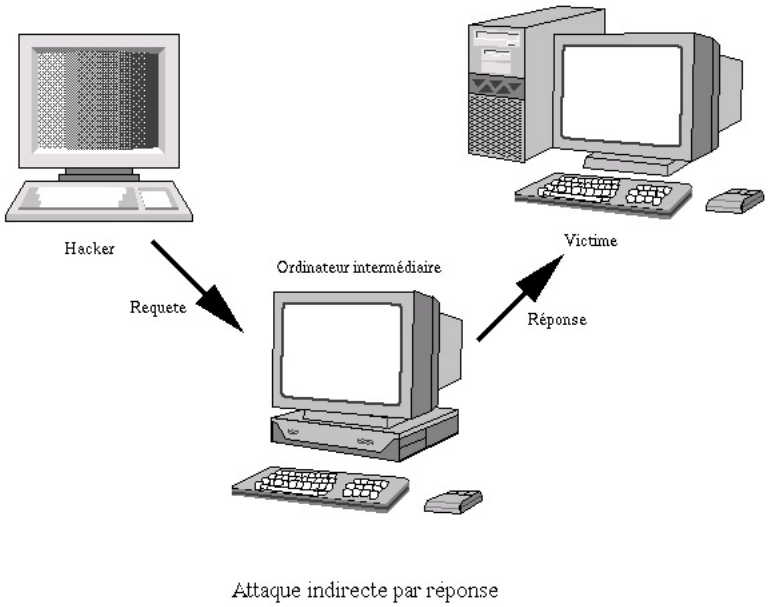


Figure 1.4 – Attaque indirecte par réponse [Arnaud, 2002]

1.3.4 Quelques types d'attaques

- **Les injections SQL[Cisco, 2020b] :** 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 vérifiés. 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.
- **IP Spoofing [Valgasu, 2017] :** consiste à se faire passer pour un autre système en falsifiant son adresse IP. L'attaquant doit déterminer les systèmes ou les adresses IP autorisés à se connecter au système cible en récoltant le maximum de détails sur ce dernier. Le pirate ensuite attaque la cible en utilisant l'adresse IP falsifié.
- **Man In The Middle (MITM) [Cisco, 2020b] :** 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.

1.4 Moyens de protection

Le but de la sécurité informatique est de protéger le système de l'utilisateur de toute intrusion, cela est réalisé en assurant la sécurité des éléments qui composent le réseau ou il est situé. Il existe plusieurs mécanismes et dispositifs de sécurité, nous citrons :

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

1.4.1 Pare-feu [Boucherba et Ziane, 2015] [Ogungbemi, 2019]

Connu sous le nom « Firewall » le pare-feu est un système physique ou logique qui permet de filtrer les paquets de données échangées avec le réseau, il s'agit ainsi d'une passerelle filtrante comportant au minimum une interface pour le réseau à protéger (réseau interne) et une interface pour le réseau externe.

Un pare-feu permet de créer une barrière de sécurité pour empêcher les accès non autorisés et d'arrêter les activités nuisibles avant qu'elles ne se propagent de l'autre côté et endommagent le réseau privé.

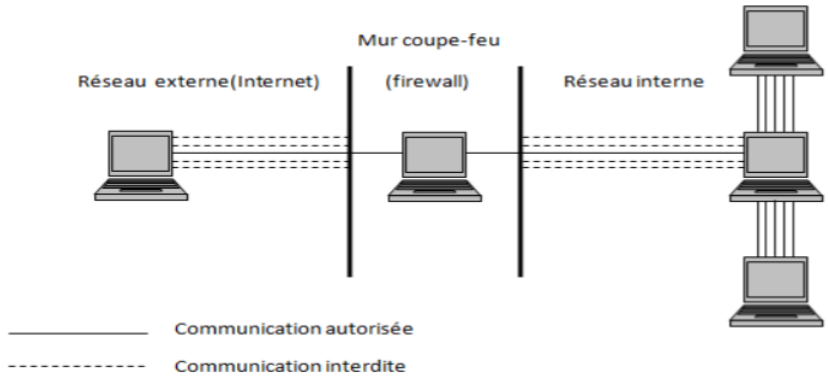


Figure 1.5 – Architecture d'un pare-feu [Boucherba et Ziane, 2015]

A) **Principe de fonctionnement :**

Le but du pare-feu est d'autoriser (allow) ou d'interdire (deny) le passage des paquets ou de rejeter la demande de connexion sans avertir l'émetteur (drop) en se basant sur un ensemble de règles appelées ACL (Access Control List).
La mise en œuvre d'une méthode de filtrage dépend de la politique de sécurité adoptée, ces deux concepts permettent de construire l'ensemble des règles utilisées par le

pare-feu.

- **Politiques de sécurité :** on distingue deux types de politiques de sécurité, la première consiste à autoriser uniquement les communications ayant étés explicitement autorisées. Quant à la deuxième, consiste à empêcher les échanges qui ont été explicitement interdits.

- **Méthodes de filtrage :** Il existe trois méthodes de filtrage :
  - Filtrage simple de paquets (stateless packet filtering) : il analyse les en-têtes de chaque paquet de données (datagramme) entrant ;
  - Filtrage dynamique de paquets (stateful packet filtering) : il est basé sur l'inspection des couches 3 et 4 du modèle OSI, permettant d'effectuer un suivi des transactions entre le client et le serveur ;
  - Filtrage applicatif (application filtering) : consiste à filtrer les communications par application, il opère sur les 7 ème couche du modèle OSI. Ce filtrage a une connaissance des applications présentes sur le réseau, et notamment de la manière dont les données sont échangées (ports, etc.).Un Firewall effectuant un filtrage applicatif est appelé Proxy.

B) **Limitations :**

- Un pare-feu ne peut pas empêcher les utilisateurs ou les attaquants équipés de modems de se connecter au réseau interne ou de le quitter, contournant ainsi complètement le pare-feu et sa protection ;
- L'introduction de supports de stockage provenant de l'extérieur ou bien des ordinateurs portables sur des machines internes au réseau peut nuire à la politique de sécurité globale ;
- Les pare-feu sont inefficaces contre les risques de sécurité non techniques tels que l'ingénierie sociale ;
- Un Firewall enregistre les tentatives d'intrusions dans un journal d'où la nécessité d'administrer le réseau afin de surveiller son journal d'activité pour contrôler l'accès aux applications et d'empêcher le détournement d'usage ;
- Les pare-feu ne peuvent pas empêcher les utilisateurs internes d'accéder à des sites Web avec un code malveillant, ce qui rend la formation des utilisateurs essentielle.

1.4.2 Serveurs mandataires (Proxy)

Un serveur Proxy est une machine qui joue le rôle d'intermédiaire entre les machines de deux réseaux séparés, Internet et le réseau prive. La plupart du temps, le serveur Proxy est utilisé pour le web, il s'agit alors d'un Proxy HTTP.

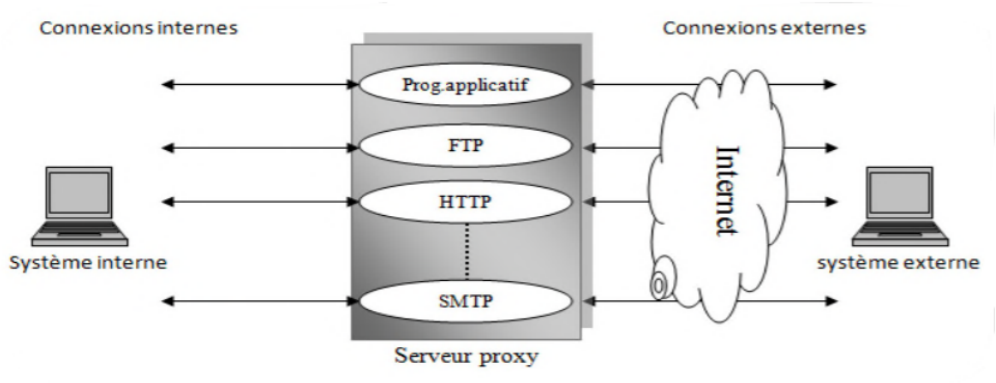


Figure 1.6 – Architecture d'un proxy [Boucherba et Ziane, 2015]

A) **Principe de fonctionnement :**
Lorsqu'un utilisateur de serveur Proxy se connecte à l'aide d'une application cliente configurée pour utiliser un serveur Proxy envoie une requête, l'application se connecte en premier lieu au serveur proxy et lui donne sa requête. Le serveur Proxy va alors se connecter au serveur que l'application cliente cherche à joindre et lui transmet la requête (le serveur Proxy contacte le serveur externe sollicité sur Internet avec sa propre adresse ou une adresse issue d'un pool d'adresses IP libres).
Le serveur va ensuite donner sa réponse au Proxy, qui va à son tour la transmettre à l'application cliente [Pillou et Bay, 2013].
Le Proxy fait en sorte que toute l'infrastructure du réseau local ne dévoile en aucun cas les adresses des machines internes (masquage d'adresse) [Ghernaouti-Hélie, 2000].

B) **Limitations [Souvik, 2020] :**

- Un serveur proxy offre une protection de base de la confidentialité, il ne protège que les adresses IP ;
- De plus, cette protection ne peut être appliquée que lorsque l'accès aux sites Web est via le site proxy ;
- Si un nouvel onglet dans le navigateur est ouvert ou une application est lancée dans le système d'exploitation, la même protection n'est pas garantie ;
- La plupart des serveurs proxy utilisent uniquement un certificat SSL (Secure Sockets Layer) pour crypter les données ce qui est insuffisant contre le stripage SSL (une attaque pour décrypter) ;
- Des serveurs limités peuvent être disponibles pour le proxy qui ne permet pas de contourner correctement le contenu protégé par région.

1.4.3 Réseaux Privés Virtuels (RPV)

Le RPV ou VPN en Anglais pour « Virtual Private Network » est un système qui permet d'étendre un réseau privé sur un réseau public afin de créer un lien direct entre des appareille distants. Un RPV isole les échanges des deux partie 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].

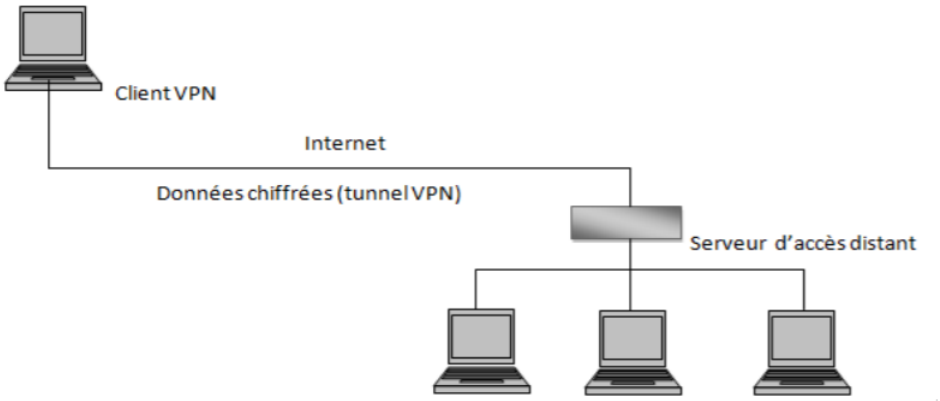


Figure 1.7 – Architecture d'un RPV [Boucherba et Ziane, 2015]

A) **Principe de fonctionnement [Suini, 2015] :**

Basé sur le protocole de « tunneling » qui permet de faire circuler les données de manière cryptée, le principe du RPV consiste à créer un chemin virtuel entre le destinataire et l'émetteur appelé le tunnel chiffré, cette expression est utilisée pour symboliser le fait que les données sont cryptées entre les deux extrémités du RPV et donc incompréhensibles pour toute personne située au milieu de cette connexion. Par la suite, la source chiffre les données et les envoie en empruntant le chemin virtuel. Il existe différents protocoles de tunneling parmi nous citons :

- 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.

B) **Limitations :**

Malgré les nombreux avantages que les réseaux privés virtuels peuvent offrir pour garantir la confidentialité en ligne, il existe divers inconvénients nous citons [Simmons, 2019] :

- Un service RPV peut surveiller les activités de leur utilisateur et utiliser les données collectées au sein de ses activités pour leurs propres avantages ;
- Étant donné que la connexion au réseau privé fonctionne en connectant le réseau de l'utilisateur à un serveur privé avant que ce dernier 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 RPV sont des connexions point à point et ne tendent pas à soutenir les domaines de diffusion ou les lier ; par conséquent, les communications, les logiciels et la mise en réseau, qui sont basés sur la couche 2 du modelé 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 [Hacker Combat, 2019].

1.4.4 Antivirus

C'est un logiciel qui analyse les fichiers ou la mémoire d'un système à la recherche de certaines anomalies, il permet d'identifier et de supprimer les logiciels malveillants. D'après les recherches de « Frederick B. Cohen's » [Cohen, 1987] il n'y a pas d'algorithme capable de détecter parfaitement tous les virus possibles pour cela différentes approches sont utilisées pour identifier les virus tels que la méthode du dictionnaire qui compare le contenu du fichier avec les définitions des virus stockées dans la base de données de l'antivirus. Ces principes peuvent être utilisés comme différentes couches de défense pour améliorer le taux de détection.

A) **Principe de fonctionnement [Advisio, 2019] :**

Le logiciel antivirus vérifie les programmes dans le système et les compare aux types de logiciels malveillants connus. Il analyse le système afin de détecter les comportements

susceptibles pouvant signaler la présence d'un nouveau malware inconnu. En général, un logiciel antivirus utilise les trois processus de détection suivants :

- **Détection spécifique :** cela fonctionne en recherchant les logiciels malveillants connus par un ensemble spécifique de caractéristiques ;
- **Détection générique :** ce processus recherche les logiciels malveillants qui sont des variantes de «familles» connues ou des logiciels malveillants liés par une base de code commune ;
- **Détection heuristique :** ce processus recherche les virus précédemment inconnus en recherchant les comportements suspects connus ou les structures de fichiers.

B) **Limitations [CERT, 2010] :**

Les antivirus sont des logiciels à part entière et à ce titre ne sont pas exempts de bogues. Ainsi, la complexité de détection des virus peut amener deux principaux types d'erreurs :

- La sur-détection de virus, autrement dit le « faux positif »
- La non-détection de virus, autrement dit le « faux négatif »

De plus un antivirus doit traiter différents types de fichier tels que les exécutables, les archives compressés et les fichiers média ... etc. Chacun de ces formats a sa propre complexité ce qui augmente la difficulté des antivirus de traiter tous ces formats d'une manière appropriée. La mutation rapide des virus rend les antivirus qui utilise la recherche de signatures afin d'identifier un code malveillant incapable d'assurer une protection contre une attaque ciblée faisant intervenir un code développé sur mesure et non encore présent dans la base de signature de virus.

1.4.5 Système de détection d'intrusion SDI

Un système de détection d'intrusion (ou IDS : Intrusion Detection System) est un ensemble de composants logiciels et/ou matériel destiné à repérer toute tentative d'effraction qui soit volontaire ou non [Burgermeister et Krier, 2006]. La détection des intrusions est une tâche très complexe, cette complexité est due à la variété des environnements et les différents paramètres entrant dans le processus de détection [Zaidi, 2001].

Les SDIs sont sujet de notre travail, ils englobent de multiples architectures et approches raison pour là quelle ils feront l'objet du chapitre suivant.

1.5 Comparaison entre les outils de sécurité

Bien que les outils de sécurité soient similaires, il existe des différences qui permettent de créer un lien complémentaire entre ces derniers.

1.5.1 Différences entre un proxy et un RPV [Petters, 2020]

Sans considérer leur principe de fonctionnement, d'autres différences doivent être parcourues telles que la différence de protection offerte par un RPV par rapport à un proxy. Malgré que ces deux mécanismes protègent l'adresse IP de l'utilisateur, un RPV peut offrir un cryptage total pour tout le trafic, ce manque de fonctionnalité de protection pour un proxy engendre l'existence d'autres inconvénients tels que le vol de données ou de l'identité de l'utilisateur, cela est dû à la mémoire cache de proxy qui peut mémoriser les mots de passe et autres informations sensibles, cette mémoire donne la possibilité aux fournisseurs de surveiller les activités en ligne de l'utilisateur et garder une trace sur ces derniers.

1.5.2 Différences entre un antivirus et un pare-feu[Meade, 2019] [GeeksforGeeks, 2020]

Ces deux mécanismes de protection pourraient être considérés comme similaires, mais leur approche sur les différentes menaces externes et internes crée une ligne qui les sépare. Tout d'abord du point de vue de la protection un pare-feu surveille le trafic réseau et empêche les données malveillantes d'entrer dans le réseau tel que les virus. Cependant, un virus peut également pénétrer dans le système avec d'autres moyens tels qu'un téléchargement ou une clé USB. Une fois que le virus a contourné la protection par pare-feu, le rôle d'un antivirus entre en jeu, il permet d'exécuter un « scan » et de détecter le « malware » pour l'empêcher de se propager en supprimant ou en isolant le fichier. Cela permet de déduire qu'un antivirus fonctionne au niveau des fichiers tandis qu'un pare-feu protège le système au niveau des protocoles réseau bloquant tous les paquets vulnérables sur le port.

Quant au point de vue de conception de ces deux mécanismes, un antivirus est un logiciel qui protège d'autres logiciels et un pare-feu est capable de préserver à la fois les logiciels et le matériel sur le réseau.

1.6 Mise en œuvre d'une politique de sécurité [Baudoin et Karle, 2004]

Une politique de sécurité informatique définit les objectifs et les mécanismes à mettre en œuvre dans les différents services pour maintenir un certain niveau de sécurité. Elle reflète la vision stratégique de la direction de l'entreprise en matière de Sécurité des Systèmes d'Informations (SSI).

L'objectif principal d'une politique de sécurité est de garantir les droits d'accès aux données et ressources, c'est-à-dire sécuriser l'accès aux données de façon logicielle (authentification, contrôle d'intégrité) et de façon physique (serveurs placés dans des salles blindées avec badge d'accès...). Un autre aspect très important est de définir les actions à suivre et les personnes à contacter en cas de détection d'intrusion.

Enfin, il est essentiel pour l'administrateur de sécurité de s'informer continuellement, des

nouvelles attaques existantes, des outils disponibles, etc. de façon à pouvoir maintenir à jour son système de sécurité et à combler les brèches de sécurité qui pourraient exister.

1.7 Conclusion

Nous avons présenté dans ce chapitre un aperçu général sur la sécurité informatique, et cela en parcourant les diffèrent types d'attaques pouvant nuire les systèmes, nous avant également présenter les outils à mettre en place pour éliminer tout type de menace sur le réseau ainsi que la stratégie à prendre pour mettre en place une politique de sécurité.

Tout cela n'arrête pas les pirates informatiques qui ne cessent de chercher et de trouver de nouvelles failles et des méthodes pour contourner les moyens de défense des systèmes informatiques ce qui oblige tout utilisateur et administrateur de sécurité à être à jour en s'informant sur toute nouveauté concernant la sécurité. Nous nous intéressant tout au long de notre travail particulièrement aux systèmes de détection d'intrusion qui feront l'objet du chapitre prochain.

Chapitre 2

Système de détection d'intrusions

2.1 Introduction

Les Systèmes de Détection d'Intrusions (IDSs) sont des systèmes logiciels ou matériels qui automatisent le processus de surveillance des événements se produisant dans un système informatique ou un réseau en effectuant une certaine analyse pour détecter les signes de problèmes de sécurité. Comme les attaques de réseau ont augmenté en nombre et en gravité au cours des dernières années, les systèmes de détection d'intrusions sont devenus nécessaires pour assurer la sécurité de la plupart des organisations. En effet, les SDIs sont considérés comme une des lignes de défense efficaces contre les attaques réseau dirigées contre les systèmes informatiques.

Ce chapitre est consacré aux systèmes de détection d'intrusions où nous présentons quelques définitions, nous verrons également les principaux types de SDI en considérant différents critères pour les classifier.

2.2 Définition

Les systèmes de détection d'intrusion sont des systèmes matériels et logiciels qui surveillent les événements survenus sur les ordinateurs et les réseaux informatiques afin d'analyser les problèmes de sécurité. Le nombre et la gravité de ces attaques ne cessent d'augmenter, par conséquent, les SDI sont devenus une partie intégrante de l'infrastructure de sécurité des organisations.

Les intrusions dans les réseaux informatiques sont appelées « attaques » et ces attaques menacent la sécurité des réseaux en violant les mécanismes de confidentialité, d'intégrité et d'accessibilité. Les attaques peuvent provenir d'utilisateurs qui se connectent à l'ordinateur en utilisant internet et en essayant d'obtenir des droits de super utilisateur ou d'administrateur et d'autres utilisateurs qui abusent des droits dont ils disposent. Les SDI automatisent la surveillance et l'analysent des attaques [Aydin et al., 2009].

2.3 Principaux types d'SDI

Il existe plusieurs types de système de détection d'intrusion caractérisés par différentes approches de surveillance et d'analyse. Chaque approche a ses propres avantages et inconvénients. Il existe plusieurs taxonomies qui dépendent de plusieurs critères (voir figure 2.1) nous allons voir dans ce qui suit quelques-unes.

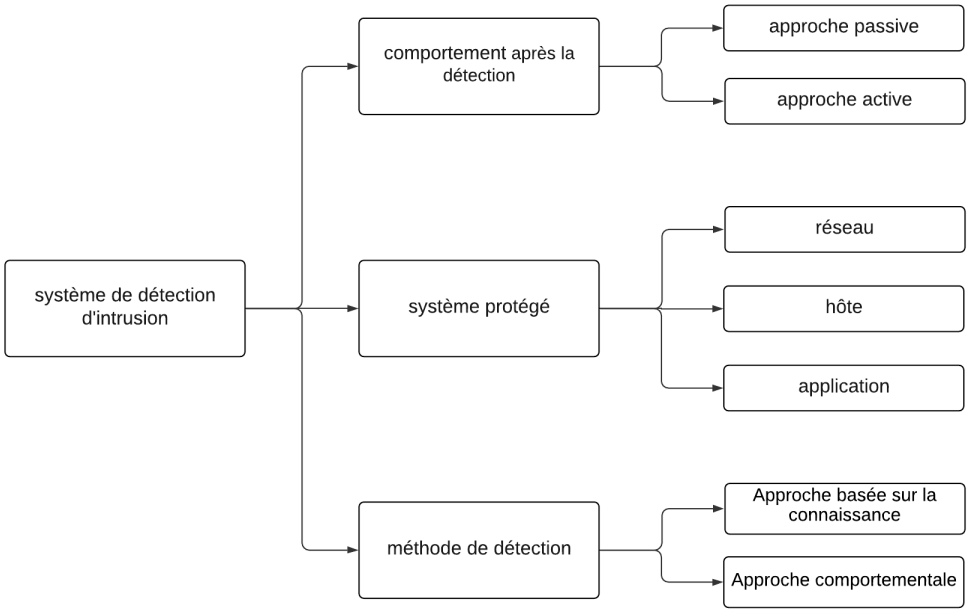


Figure 2.1 – Principaux types d'SDI [Azad et Jha, 2013]

2.3.1 Classification selon le système protégé

On distingue trois types d'SDI majeurs en se basant sur la nature du système à protéger :

A) **La détection d'intrusion basée sur le réseau :**

Un SDIR (Système de Détection d'Intrusion basé Réseau) ou un NIDS (Network Intrusion Detection System) est configuré à un point planifié du réseau pour examiner le trafic de tous les appareils du réseau. Il effectue une observation du trafic passant sur l'ensemble du trafic du sous-réseau et le fait correspondre à la collection d'attaques connues. Une fois l'attaque et identifiée ou un comportement anormal est détecté, une alerte est envoyée à l'administrateur.

On peut distinguer trois localisations les plus courantes où on peut installer un SDIR [Müller et al, 2003] :

- À l'intérieur du pare-feu : si les SDIRs se trouvent à l'intérieur du pare-feu, il est plus facile de savoir si le pare-feu a été mal configuré et on peut ainsi savoir si une attaque est venue par ce pare-feu. L'expérience a montré que les SDIRs génèrent moins de faux positifs s'ils agissent moins discrètement tout en réduisant le trafic (diminution des fausses alertes).
- À l'extérieur du pare-feu : les SDIRs sont souvent placés à l'extérieur du pare-feu, la raison en est que le SDIR peut recevoir et analyser l'ensemble du trafic de l'internet. Il n'est pas certain que toutes les attaques soient filtrées et détectées

dans cette position, par exemple les attaque TCP. Dans ce cas, on devra essayer de détecter l'attaque en utilisant les signatures. Cet emplacement est le préféré des experts, car il offre l'avantage d'écrire dans les logs et d'analyser les attaques, ainsi l'administrateur voit ce qu'il doit modifier dans la configuration du pare-feu.

- À l'intérieur et à l'extérieur du pare-feu : cette variante réunit les deux cas mentionnés ci-dessus, mais elle est très dangereuse en cas de mal configuration du SDIR et/ou du pare-feu.

B) **La détection d'intrusion basée sur l'hôte [Caswell et Beale, 2004] :**

Le SDIH (Système de Détection d'Intrusion basé Hôte) ou HIDS (Host-Based IDS) diffèrent du SDIR de deux manières. Premièrement un SDIH installé protège seulement le système où il réside et pas tout le réseau. La carte réseau d'un système avec un SDIH installé fonctionne normalement en mode non promiscuité.

Un SDIH a l'avantage de pouvoir adapter l'ensemble des règles pour qu'il soit très spécifique au système hôte particulier, par exemple il n'est pas nécessaire de configurer plusieurs règles conçues pour détecter les exploits du système de fichiers réseau sur une hôte qui ne l'utilise pas. Être en mesure d'affiner le jeu de règles améliorera les performances et diminuera les faux positifs (ou les vrais positifs).

Un autre avantage très important du SDIH réside cependant dans sa capacité à détecter des modifications spécifiques des fichiers et du système d'exploitation de son hôte. Il peut surveiller la taille des fichiers et les sommes de contrôle pour s'assurer que les fichiers systèmes cruciaux ne sont pas modifiés de manière malveillante sans que quelqu'un ne s'en aperçoive. Il peut intercepter les appels systèmes non fiables qui peuvent être une tentative d'exploiter une vulnérabilité locale.

C) **La détection d'intrusion basée sur une application :**

Les SDIs basés sur les applications sont un sous-groupe des SDIs basés hôtes qui contrôlent l'interaction entre un utilisateur et le programme en ajoutant des fichiers log afin de fournir de plus amples informations sur les activités. Puisqu'il s'agit d'opérer entre l'utilisateur et le programme, il est facile de filtrer tout comportement anormal [Müller et al, 2003].

La capacité d'interagir directement avec l'application, avec des connaissances importantes sur un domaine ou une application spécifiques incluses dans le moteur d'analyse permet aux SDIs basés sur l'application de détecter un comportement suspect dû au fait que les utilisateurs autorisés dépassent leurs droits spécifiés. En effet, ces problèmes sont plus susceptibles d'apparaître dans l'interaction entre l'utilisateur, les données et l'application[Bace et Mell, 2001].

2.3.2 Classification selon la méthode de détection

Deux approches existent selon les méthodes de détection :

A) **Approche basée sur la connaissance :**

La forme la plus répandue de détection d'intrusion est la correspondance de signatures appelée SDI basé sur les signatures. Ces systèmes surveillent le réseau ou le serveur et font correspondre les paquets des attributs du trafic avec un ensemble de listes d'attaques ou de signatures prédéterminées. Si une conversation réseau particulière correspond à une signature configurée sur le SDI, le système alerte les administrateurs ou prend d'autres mesures préconfigurées [Sweeney et al., 2003].

Malgré leur efficacité, les SDIs basés sur la signature montrent plusieurs inconvénients, la base de données dédiée doit être volumineuse pour couvrir la majorité des attaques. Il est difficile aussi au SDI de suivre le trafic réseau dû à sa vitesse qui ne cesse d'augmenter. De même manière, puisque les SDIs basés sur la signature se réfèrent à une base de données de signature, les nouvelles attaques ne peuvent être détectées sauf si une mise à jour de la base est effectuée.

Le reniflement de paquets est considéré comme la principale méthode basée sur les signatures, elle consiste que tout SDI qui examine le trafic réseau effectue un reniflement de paquets. Selon Caswell et Beale [Caswell et Beale, 2004] les SDIRs fonctionnent en définissant une interface en mode promiscuité où il s'agit de capturer chaque paquet qui traverse le sous-réseau local. Ils ne verront pas les paquets qui traversent une pile TCP / IP interne à une machine, mais ils verront potentiellement tous sur le câble local. Cependant, de nombreux SDIHs qui effectuent une analyse du trafic réseau utilisent également des techniques similaires sans utiliser le mode promiscuité, pour collecter le trafic spécifique à l'hôte sur lequel ils résident. Le reniflement de paquets est un moyen classique pour détecter les intrusions, mais existe des techniques tout aussi classiques qui peuvent être utilisées contre les SDIs reniflant les paquets tels que les attaques par fragmentation, qui répartissent la charge utile de l'attaque entre plusieurs paquets [Caswell et Beale, 2004].

B) **Approche comportementale :**

Les SDIs basés sur les anomalies n'utilisent pas de signatures statiques pour détecter les événements de sécurité potentiels. Ces SDIs utilisent plutôt des lignes de base du trafic réseau pour déterminer un état « normal » pour le réseau et comparer le trafic actuel avec cette ligne de base. S'il est question d'anomalies, le SDI alerte les administrateurs de sécurité [Sweeney et al., 2003].

Contrairement à l'approche basée sur les signatures, les SDIs basés anomalie ont le potentiel de détecter les nouvelles attaques dès qu'elles se produisent grâce aux méthodes suivantes [Caswell et Beale, 2004] :

- **Analyse des journaux :** une des excellentes sources de données de sécurité provient des fichiers journaux systèmes, de nombreux SDIs peuvent extraire des données de ces derniers et alerter l'administrateur en cas d'un comportement anormal. En fait, certaines implémentations d'SDI d'origine utilisaient la surveillance des

journaux comme méthode de collecte de données et récupération des empreintes d'attaques.

- **Surveillance des appels système :** les SDIHs sont capables de s'installer en tant que résidents dans le noyau du système d'exploitation et de regarder (ou dans certains cas d'intercepter) les appels système potentiellement malveillants. Un appel système est une demande qu'un programme fait au noyau du système d'exploitation, si le SDIH pense que l'appel système peut être malveillant tel que la demande d'un changement de son ID utilisateur à celui de l'utilisateur Root, il peut créer une alerte ou carrément interdire l'appel système tel le cas de certains SDIHs comme celui de Linux SDIL (Système de Détection d'Intrusion Linux).
- **Surveillance du système de fichiers :** une autre tactique très courante des SDIHs est de garder un œil sur les tailles et les attributs des fichiers cruciaux dans un système de fichiers, si le noyau du système d'exploitation change soudainement de taille et qu'aucun des administrateurs système n'ai informé, c'est probablement quelque chose à vérifier, si des répertoires sont devenus accessibles en écriture ou si des fichiers sensibles ont été modifiés les administrateurs sont alors alertés d'une éventuelle activité malveillante. Tripwire est peut-être l'exemple le plus connu d'un outil pour surveiller les fichiers en cas de changements, mais il y en a beaucoup d'autres qui font la même chose, y compris l'outil open source AIDE Advanced Intrusion Environnement de détection.

2.3.3 Classification selon le comportement après la détection

Un système de détection d'intrusion propose un ensemble d'actions lors de la détection d'une anomalie, celles-ci sont souvent regroupées en deux approches, active et passive.

A) **L'approche active :**
Les réponses des SDIs actives sont des actions automatisées prises lorsque certains types d'intrusions sont détectés. Il existe trois catégories de réponses actives [Bace et Mell, 2001].

- **Recueillir des informations supplémentaires :** il s'agit de la réponse active la plus inoffensive, mais parfois la plus productive. Elle consiste à collecter des informations supplémentaires sur une attaque présumée. Cela peut impliquer l'augmentation du niveau de sensibilité des sources d'information (par exemple, l'augmentation du nombre d'événements consignés par une piste d'audit du système d'exploitation, ou l'augmentation de la sensibilité d'un moniteur réseau pour capturer tous les paquets, pas seulement ceux ciblant un port ou un système cible particulier). La collecte d'informations supplémentaires est utile pour plusieurs raisons, les informations supplémentaires collectées peuvent aider à résoudre la détection de l'attaque (aider le système à diagnostiquer si une attaque a eu lieu ou non). Cette option permet également de collecter des informations qui peuvent être utilisées pour appuyer l'enquête et l'arrestation de l'attaquant, et pour sou-

tenir les recours judiciaires pénaux et civils.

- **Changer l'environnement :** une autre réponse active consiste à stopper une attaque en cours, puis à bloquer tout accès ultérieur par l'attaquant. En règle générale, les SDIs n'ont pas la possibilité de bloquer l'accès d'une personne spécifique, mais plutôt de bloquer les adresses IP (Internet Protocol) dont l'attaquant semble provenir. Hélas il est très difficile de bloquer un attaquant déterminé et bien informé.
- **Agir contre l'intrus :** la forme la plus agressive de cette réponse consiste à lancer des contres attaques ou à tenter d'obtenir activement des informations sur l'hôte ou le site de l'attaquant. Aussi tentante soit-elle, cette réponse est malavisée, en raison des ambiguïtés juridiques concernant la responsabilité civile, cette option peut représenter un risque plus élevé que l'attaque qu'elle est censée bloquer. La première raison d'approcher cette option avec beaucoup de prudence est qu'elle peut être illégale. Enfin, la contre-attaque peut aggraver l'attaque, provoquant un attaquant qui à l'origine avait uniquement l'intention de parcourir un site pour prendre des mesures plus agressives.

B) **L'approche passive :**

Les réponses des SDIs passives fournissent des informations aux utilisateurs du système, en s'appuyant sur les humains pour prendre des mesures ultérieures sur la base de ces informations. De nombreux SDIs commerciaux reposent uniquement sur des réponses passives [Bace et Mell, 2001].

Les alarmes et les notifications sont générées par les SDIs pour informer les utilisateurs lorsque des attaques sont détectées. La plupart des SDIs commerciaux offrent aux utilisateurs une grande latitude pour déterminer comment et quand les alarmes sont générées et à qui elles sont affichées. La forme d'alarme la plus courante est une alerte à l'écran ou une fenêtre contextuelle. Ceci est affiché sur la console du SDI ou sur d'autres systèmes comme spécifié par l'utilisateur lors de la configuration. Les informations fournies dans le message d'alarme varient considérablement, allant d'une notification qu'une intrusion a eu lieu à des messages extrêmement détaillés décrivant les adresses IP de la source et de la cible de l'attaque, l'outil d'attaque spécifique utilisé pour y accéder et le résultat de l'attaque. Un autre ensemble d'options utiles aux grandes entreprises ou aux organisations réparties est celui qui consiste à notifier à distance les alarmes ou les alertes. Ceux-ci permettent aux organisations de configurer le SDI afin qu'il envoie des alertes aux téléphones cellulaires et aux téléavertisseurs portés par les équipes de réponse aux incidents ou le personnel de sécurité du système [Bace et Mell, 2001].

2.4 Conclusion

Nous avons introduit dans ce chapitre les systèmes de détection d'intrusion ainsi que leurs différents types selon plusieurs classifications existantes. Ces classifications sont faites suivant

certains critères où nous avons présenté la classification basée sur la source d'information qui peut être réseaux, hôtes ou applications, la classification basée sur la méthode de détection incluant l'approche basée sur la connaissance et celle basée sur le comportement puis nous avons clôturé ce chapitre avec les SDIs classés selon leur comportement après la détection d'une intrusion et comporte deux principales approches, l'approche active et l'approche passive. Dans le chapitre suivant nous présenterons un état de l'art sur les différentes méthodes et techniques utilisées pour réaliser des SDIs.

Chapitre 3

État de l'art sur les SDIs

3.1 Introduction

Afin de réaliser un SDI, il faudrait faire recoure aux techniques de classification, il s'agit d'un processus représentant la catégorisation algorithmique des objets à travers des connaissances extraites d'un ensemble de données lors d'une phase d'apprentissage. Les données (d'apprentissage ou de test) lors de la phase de prédiction sont regroupées en classe homogènes dans le sens de propriétés communes. Le but de la classification est donc de construire un modèle pouvant être représenté sous plusieurs formes selon la méthode d'apprentissage utilisée.

L'apprentissage supervisé et non supervisé sont deux manières distinctes pouvant être appliquées pour la phase d'apprentissage. L'apprentissage supervisé consiste à utiliser un ensemble de données étiquetées afin de corriger les erreurs de prédiction durant l'apprentissage (correction par rapport aux négatifs) [Torre, 1999]. Les arbres de décision, les réseaux de neurones artificiels, la régression linéaire, les K-plus proches voisins et les machines à vecteurs de support sont des algorithmes d'apprentissage supervisé.

L'apprentissage non supervisé (par exemple l'algorithme K-moyennes) quant à lui n'utilise pas de données étiquetées et produit directement une classification à partir des données.

Dans ce chapitre, nous mettons l'accent sur quelques méthodes de classification et en plus particulier celles qui sont utilisées lors de la réalisation d'un SDI telles que les arbres de décision, les réseaux de neurones, les algorithmes des K-moyennes et des K-plus proches voisins ainsi que les machines à vecteur de support. Nous verrons également le paradigme multi-agents connu pour sa flexibilité où il est largement utilisé dans plusieurs domaines et en particulier lorsqu'il s'agit d'une distribution physique ou fonctionnelle. Ces différentes approches et méthodes seront accompagnées d'une description de leur algorithme ainsi que leur apport dans le domaine des SDIs.

3.2 Les arbres de décision

Les arbres de décision sont une approche de modélisation prédictive. Elle se compose de nœuds qui forment un arbre enraciné, ce qui signifie qu'il s'agit d'un arbre dirigé avec un nœud appelé «racine» qui n'a pas d'arêtes entrantes. Tous les autres nœuds ont exactement une arête entrante. Un nœud avec des bords sortants est appelé nœud interne ou nœud de test. Tous les autres nœuds sont appelés feuilles, nœuds terminaux ou nœuds de décision. Les branches sont étiquetées avec les différentes valeurs que peut prendre l'attribut choisi pour le nœud [Rokach et Maimon, 2005].

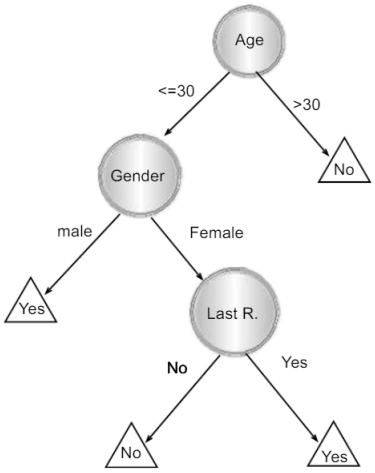


Figure 3.1 – Exemple d'arbre de décision [Rokach et Maimon, 2005]

Afin de parcourir un arbre de décision et trouver une solution, il faut partir de la racine. Chaque nœud est une décision atomique. Chaque réponse possible est prise en compte et permet de se diriger vers un des fils du nœud. De bronche en bronche, on descend dans l'arbre jusqu'à tomber sur une feuille qui représente la réponse qu'apporte l'arbre au cas que l'on vient de tester. Il s'agit de parcourir les étapes suivantes [Devéze et Fouquin, 2005] :

- commencer à partir de la racine de l'arbre ;
- descendre dans l'arbre en passant par les nœuds de test ;
- la feuille atteinte à la fin permet de classer l'instance testée.

Les arbres de décision moins complexes peuvent être considérés comme plus compréhensibles. Selon Breiman et son équipe [Breiman et al., 1984] la complexité de l'arbre a un effet crucial sur sa précision. La complexité de l'arbre est explicitement contrôlée par les critères d'arrêt utilisés et la méthode d'élagage employée. Habituellement, la complexité de l'arbre est mesurée par l'une des mesures suivantes : le nombre total de nœuds, le nombre total de feuilles, la profondeur de l'arbre et le nombre d'attributs utilisés.

3.2.1 Avantage et Inconvénients [Gareth et al., 2015]

Les arbres de décision présentent plusieurs avantages où ils sont simples à comprendre et à interpréter, ils peuvent également être affichés graphiquement ce qui ajoute un point à leur simplicité. Ils sont capables de gérer des données numériques et catégoriques contrairement à d'autres techniques et ils sont généralement spécialisés dans l'analyse d'ensemble de données qui n'ont qu'un seul type de variable, tandis que la plupart des autres techniques nécessitent souvent une normalisation des données les arbres de décision nécessitent peu de préparation des données.

En revanche, les arbres de décision présentent certains inconvénients, ils n'ont généralement

pas le même niveau de précision prédictive que certaines approches. Un petit changement dans les données d'apprentissage peut entraîner un grand changement dans l'arbre et par conséquent les prédictions finales. De plus, l'apprentissage par arbre de décision peut créer des arbres trop complexes qui généralisent mal l'ensemble d'apprentissage, afin de contourner ces problèmes des procédures d'élagage sont utilisée.

3.2.2 Construction de l'arbre de décision

La construction d'arbre de décision est réalisée depuis un ensemble d'apprentissage constitué de données étiquetées en suivant 3 étapes, la première étape vérifie si un nœud est terminal (c'est-à-dire décider s'il doit être étiqueté comme une feuille). Pour pouvoir réaliser ce processus des conditions doivent être respectées, par exemple tous les exemples sont dans la même classe. Cette étape permet d'arrêter l'expansion de la branche de l'arbre. La deuxième consiste à sélectionner un test pour l'associer à un nœud en utilisent des critères de segmentation. La dernière étape est de créer les branches du nœud, où chacune de ce dernier prend une des différentes valeurs que l'attribut du nœud peut prendre. Pour chacune des branches créées, il faut recommencer le processus en prenant les exemples correspondants à la branche [Girard, 2007].

3.2.2.1 Critères de construction

Les algorithmes de construction d'arbres de décision fonctionnent généralement de haut en bas, en choisissant une variable à chaque étape qui divise le mieux l'ensemble des éléments. Il existe différents algorithmes de construction utilisant différentes métriques qui cherchent l'homogénéité de la variable cible au sein des sous-ensembles [Rokach et Maimon , 2005].

Parmi ces métriques nous citons :

- Gain informationnel : utilisé par les algorithmes de génération d'arbres ID3, C4.5 et C5.0. Le gain d'informations est basé sur le concept d'entropie de Shannon en théorie de l'information ;
- Ratio de gain : utilisé par l'algorithme C4.5, c'est une notion complémentaire au gain informationnel, il permet d'éviter de tomber dans le sur-apprentissage ;
- Critère Gini : utilisé par l'algorithme CART (Classification And Regression Trees : arbre de classification et de régression), c'est une mesure de la fréquence à laquelle un élément choisi au hasard dans l'ensemble serait incorrectement étiqueté s'il était étiqueté de manière aléatoire en fonction de la distribution des étiquettes dans le sous-ensemble ;
- Khi deux : utilisé par l'algorithme CHAID (CHi-squared Automatic Interaction Detector), c'est une loi utilisée généralement pour la création des arbres de régression.

3.2.2.2 Définir la taille de l'arbre [Rakotomalala, 2005]

Lors de la création d'arbre de décision, le but initial est d'avoir un arbre le plus compréhensible que possible, mais le phénomène de sur-apprentissage rend cette tâche difficile, car ce dernier va pousser l'algorithme de construction à produire un arbre trop complexe qui va affecter les performances de l'arbre final, à cause de cet obstacle la construction de l'arbre optimal nécessite un des deux mécanismes « pré-élagage et post-élagage » :

- Pré-élagage : consiste à proposer des critères d'arrêt lors de la phase d'expansion.
- Post-élagage : consiste à construire l'arbre en deux temps : une première phase d'expansion, puis on réduit l'arbre.

3.2.3 Type d'arbre de décision

Il existe deux principaux types d'arbre de décision :

A) **Les arbres de classification :** ce type d'arbre s'applique aux données où le résultat est une étiquette de classification telle que les types d'attaques après le traitement des paquets par les systèmes de détection d'intrusion où ce dernier doit construire une règle de décision qui prédit le résultat à l'aide des donnes d'apprentissage, de plus il est souhaitable de construire une règle de décision qui soit exacte non seulement pour les données disponibles, mais également pour les données externes (c'est-à-dire que la règle de décision devrait avoir de bonnes performances de prédiction). En même temps, il est utile d'avoir une règle de décision compréhensible.

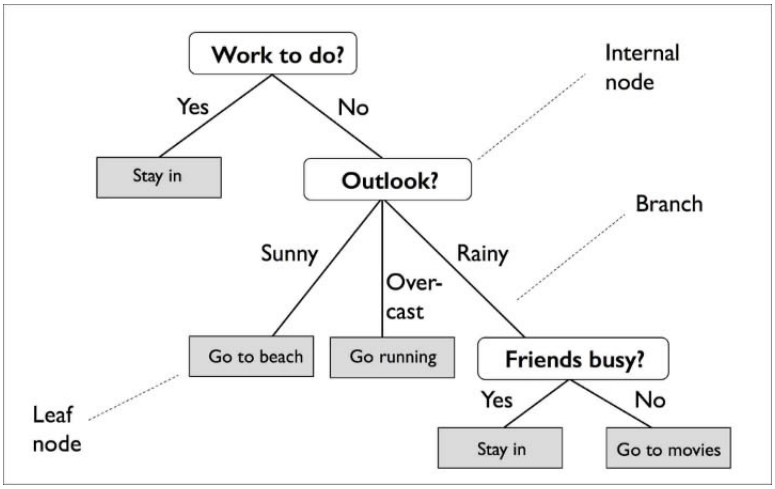


Figure 3.2 – Exemple d'arbre de classification [DEV Community, 2019]

B) **Les arbres de régression :** ces arbres de décision sont utilisés pour analyser les données lorsque le résultat est une quantité réelle (par exemple, le prix d'une maison ou la durée de séjour d'un patient dans un hôpital), dans ce cas, la prédiction est une valeur numérique. Les arbres de régression peuvent être construits en utilisant un partitionnement

récursif similaire aux arbres de classification. Les valeurs des nœuds terminaux dans un arbre de régression sont définies comme la valeur moyenne des résultats.

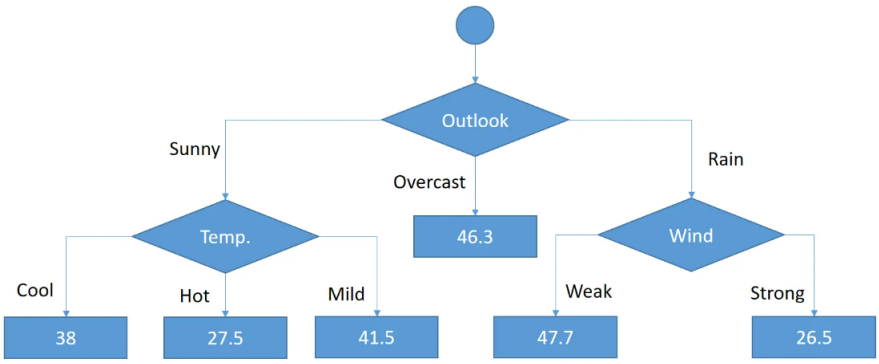


Figure 3.3 – Forme finale d'un arbre de régression [Serengil, 2018]

Il existe un autre type d'arbre qui peut être considéré comme arbre de décision, il s'agit des **arbres de survie** qui sont utilisés dans l'analyse de survie dans le domaine médical, où les données relatives au temps (Time-To-event) sont souvent concernées. Pour ces données, l'analyse se concentre sur la compréhension de la façon dont l'événement relatif au temps varie en matière de différentes variables qui pourraient être collectées pour un patient [Ishwaran et Rao, 2009].

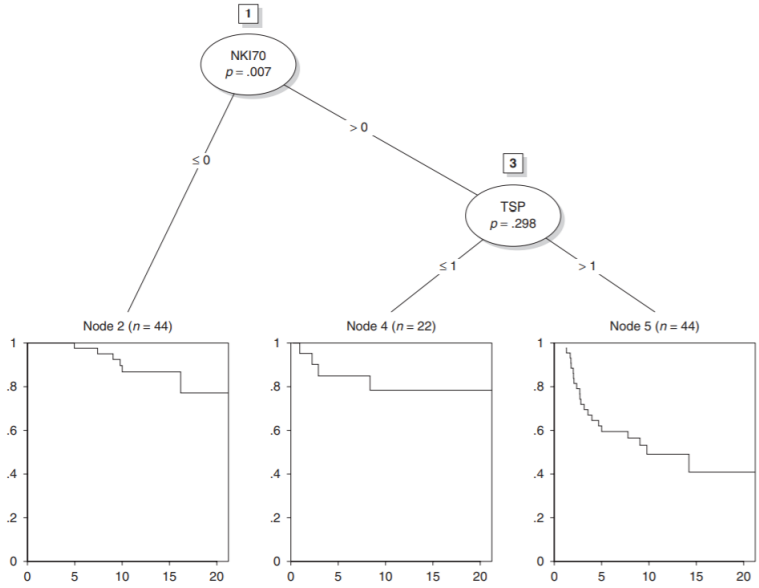


Figure 3.4 – Arbre de survie binaire pour des patientes atteintes d'un cancer du sein [Ishwaran et Rao, 2009]

3.2.4 Arbre de décision dans les SDIs

Plusieurs auteurs ont fait appel aux arbres de décision dont l'objectif de réaliser un système de détection d'intrusion tel qu'Elekar et Waghmare [Elekar etWaghmare, 2015] qui ont implémenté différents classifieurs tels que l'arbre de décision C4.5, forêt aléatoire « Random Forest », l'arbre Hoeffding pour la détection des intrusions et 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.

Aggarwal et Sharma [Aggarwal et Sharma, 2015] évaluent dix algorithmes de classification tels que les Forêts d'Arbres Décisionnels FAD « Random Forest », C4.5, la classification naïve bayésienne (Naïve Bayes) et les tables de décision (Decision Table). Ils simulent ensuite ces algorithmes de classification dans WEKA avec la base de données KDD'99. Ces classifieurs sont analysés en fonction de paramètres tels que la fidélité, la précision. L'algorithme FAD montre les meilleurs résultats dans l'ensemble tandis que l'algorithme C4.5 avait un taux de détection élevé et un faible taux de fausses alarmes.

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. Avec l'utilisation de l'algorithme SCA (Simple CART Algorithm) les résultats obtenus donnent la plus grande précision environ 66,77%, alors que la classification avec l'arbre de décision C4.5 le résultat est de 65,65 % seulement. 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.

3.3 Les réseaux de neurones

Les Réseaux de Neurones Artificiels (RNAs) sont un assemblage interconnecté d'éléments, d'unités ou de nœuds de traitement simples qui sont développés en modélisant le cerveau humain, auquel ils sont similaires en deux manières.Premièrement, la manière dont l'information acquise est véhiculée dans le réseau, deuxièmement, la capacité de traitement du réseau qui réside dans les poids de connexion inter-unités, obtenus par un processus d'adaptation ou d'apprentissage à partir d'un ensemble de modèles d'apprentissage. Les RNAs peuvent être décrits comme une simple copie des réseaux de neurones biologiques [Gurney, 1997][Staub et al.,2015].

Les RNAs ont plusieurs avantages tels que leur capacité d'apprentissage adaptatif, c'est-à-dire la capacité d'apprendre à effectuer des tâches en fonction des données fournies lors de l'expérience initiale. Un RNA peut créer sa propre organisation ou représentation des informations qu'il reçoit 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 entraîne la dégradation correspondante des performances. Cependant, certaines capacités du réseau peuvent être conservées même en cas de dommages importants du RN [Eluyodeet Akomolafe, 2013].

3.3.1 Architecture des réseaux de neurones

Les différents types de connexions entre les neurones qui composent le réseau décrivent l'architecture du modèle, selon la figure ci-dessus, nous pouvons distinguer deux types d'architecture de réseaux de neurones : les RNs non bouclés et les RN bouclés [Bendaoud, 2014].

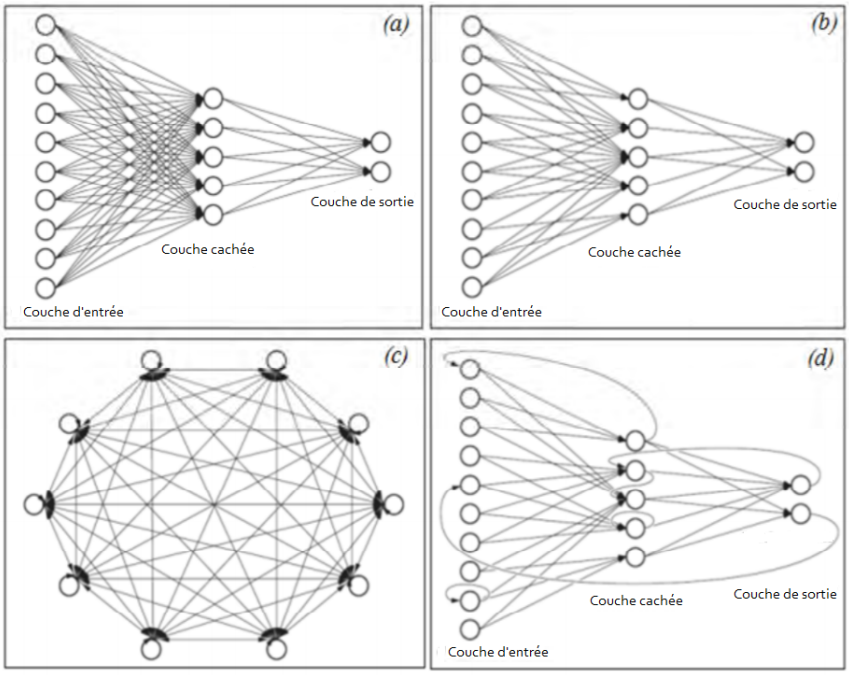


Figure 3.5 – Type de RNA (a) Réseau multicouche à connexions totales, (b) Réseau multicouche à connexions locales, (c) Réseau multicouche à connexions complexes, (d) réseau multicouche à connexions récurrentes (dynamique) [Bendaoud, 2014]

3.3.1.1 Les réseaux de neurones non bouclés

Le Perceptron MultiCouche PMC est l'exemple le plus simple dans les RNs non bouclés. Il est composé de plusieurs couches de neurones où la sortie d'un neurone d'une couche devient l'entrée d'un neurone de la couche suivante. Ces couches peuvent être décrites comme suit [Derras, 2011] :

1. Couche d'entrée : c'est la couche dans laquelle les groupes de données d'entrée sont introduits dans le réseau. Le nombre de neurones dans une couche d'entrée est égal au nombre de données d'entrée ;

2. Couche cachée : la couche cachée est la fonction de base du réseau. Dans cette couche, les données reçues de la couche d'entrée sont traitées correctement puis transmises à la couche de sortie ;
3. Couche de sortie : cette couche donne le résultat obtenu après compilation par le réseau des données entrée dans la première couche. Sa taille est directement déterminée par le nombre de variables dont on a besoin en sortie.

Les RNs monocouches et les RNs à connexions locales sont deux autres sous-ensembles des RNs non bouclés.

- **Les RNs monocouches :** cette architecture est la plus simple appelée le perceptron elle se caractérise par un nombre d'entrées N et de sortie P. Le perceptron est composé de p neurones qui sont représentés généralement alignés verticalement où chacun peut avoir une fonction d'activation différente théoriquement, mais généralement, ce n'est pas le cas en pratique[Djeffal, 2020].

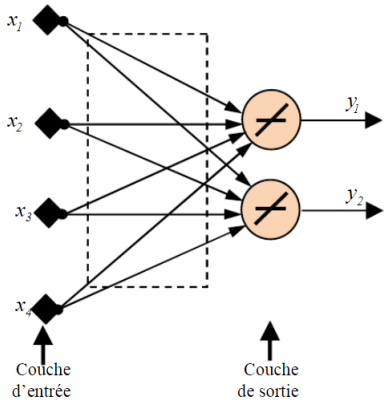


Figure 3.6 – Architecture d'un RN monocouche[Djeriri, 2017]

- **Les RNs à connexions locales :** il s'agit d'une structure multicouche, mais chaque neurone entretient des relations avec un nombre réduit et localisé de neurones de la couche avale. Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique [Touzet, 1992].

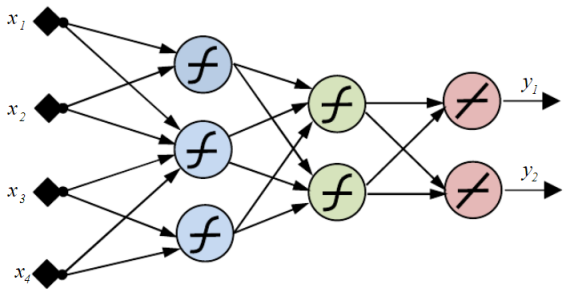


Figure 3.7 – Architecture d'un RN à connexions locales [Djeriri, 2017]

3.3.1.2 Les réseaux de neurones bouclés [Djeriri, 2017]

Les réseaux bouclés sont dotés de l'architecture la plus générale pour un RN, contrairement à l'architecture précédente, le graphe de connexion de cette architecture est cyclique. Les réseaux de neurones bouclés peuvent avoir une topologie de connexions quelconque, comprenant notamment des boucles qui ramènent aux entrées la valeur d'une ou plusieurs sorties. Pour qu'un tel système soit causal, il faut évidemment qu'à toute boucle soit associé un retard, un réseau de neurones bouclé est donc un système dynamique, régi par des équations différentielles. Comme l'immense majorité des applications sont réalisées par des programmes d'ordinateurs, on se place dans le cadre des systèmes à temps discret, où les équations différentielles sont remplacées par des équations aux différences. Il s'agit donc de réseaux de neurones avec retour en arrière (feedback network or recurrent network).

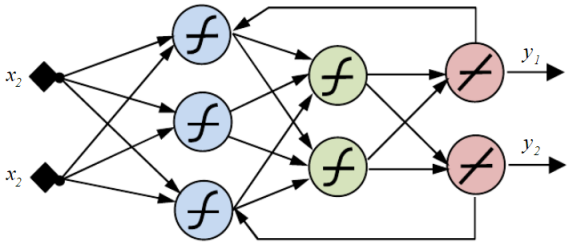


Figure 3.8 – Schéma du réseau de neurones bouclé[Djeriri, 2017]

3.3.2 Apprentissage [Beckenkamp, 2002]

L'apprentissage est le processus de recherche de poids dont les valeurs détermineront si le neurone est capable d'accomplir une certaine tâche ou non. En pratique, le processus d'apprentissage est la modification des valeurs de poids afin de trouver les bonnes valeurs de sortie. Souvent les valeurs des poids sont déterminées par un algorithme d'apprentissage qui est en lui-même une méthode automatique qui essaie d'adapter les poids appropriés à la solution système. Les algorithmes d'apprentissage sont devisés en deux catégories :

- **L'apprentissage supervisé :** dans cette catégorie, l'apprentissage est ajusté en envoyant des informations aux réseaux de ce qu'il doit apprendre pour pouvoir l'évaluer par la suite. Ce type d'apprentissage dispose d'informations permettant de décider la durée l'apprentissage,la fréquence de présentation de chaque donnée pour l'apprentissage. L'apprentissage supervisé peut être aussi subdivisé en deux sous-catégories, l'apprentissage structurel et temporel.
- **L'apprentissage non supervisé :** il utilise uniquement des informations locales pendant l'apprentissage. Il organise les exemples présentés afin de découvrir leurs propriétés collectives.

3.3.3 Les réseaux de neurones dans les SDIs

L'implémentation des réseaux de neurones dans le domaine de détection d'intrusion était introduite par « James Canady » dans l'année 1998 [Canady, 1998] depuis plusieurs travaux ont suivit. Certains concepteurs des systèmes de détection d'intrusion exploitent les réseaux de neurones artificiels comme étant une technique de reconnaissance de formes ou cette dernière peut être mise en œuvre en utilisant un réseau de neurones à propagation avant (Feedforward neural network). Pendant l'apprentissage, les paramètres du RN sont optimisés pour associer les sorties aux modèles d'entrée correspondants. Lors de l'étape du test si un enregistrement donné en entrée n'a pas de sortie associée alors le RN donne la sortie qui correspond à un modèle d'entrée déjà acquis qui est le moins différent du modèle donné. Deux types de RN sont les plus utilisés dans le domaine de la détection d'intrusion, les réseaux de neurones à propagation avant multicouche et les carte auto adaptatives de Kohonen [Reddy, 2013].

Plusieurs d'autres auteurs se sont intéressés à l'usage des RNs tels que Gautam et son équipe [Gautam et al., 2016] qui proposent les deux modèles de réseaux de neurones destinés à la détection d'intrusion sur hôte, à savoir les Réseaux de Neurones de Régression Généralisée (RNRG) ainsi que le Perceptron Multicouche (PMC) qui ont étés évalués en utilisant le journal log généré sur une machine personnelle où les analyses de simulation ont montré que le RNRG donne de bons résultats en termes de taux de précision et de valeur de rappel, mais la valeur de précision est inférieure à celle du PMC. Selon les auteurs les deux modèles conviennent à la détection d'intrusion basée hôte.

Shenfield et son groupe quant à eux [Shenfield et al., 2018] proposent un système de détection d'intrusion basé sur les signatures en utilisant le perceptron multicouche comme classificateur pour l'identification des modèles de shellcode malicieux dans le trafic réseau. Ce classificateur a pu atteindre une sensibilité parfaite sur l'ensemble de données de test (identifiant toutes les instances de shellcode) et présente une excellente précision (minimisant le nombre de faux positifs identifiés).

Raman et al [Raman et al., 2017] proposent une nouvelle approche destinée à la détection d'intrusion basée sur la propriété Helly (PH :Hellyproperty) d'un hypergraphe et un Réseau de Neurones Probabiliste (RNP) basé sur l'arithmétique modulaire où la première notion

était exploitée pour sélectionner un sous-ensemble de caractéristiques optimales qui est par la suite utilisé pour l'apprentissage du RNP. Les performances de leur approche étaient évaluées en utilisant la base de données KDD99.

3.4 La méthode des K-moyennes

L'algorithme des K-moyennes (dit en anglais K-means) est une méthode de classification non supervisée ou le but des algorithmes de cette catégorie est de regrouper les individus vus similaires et constituer des classes avec un ensemble de données non étiquetés. Le terme «k-means» a été introduit pour la première fois par James MacQueenen en 1967, bien que l'idée revienne à Hugo Steinhaus [Steinhaus, 1956]. K-moyennes est dédié aux tâches de partitionnement qui consiste à diviser le jeu de données en K groupes homogènes appelés clusters où K est spécifié par l'utilisateur. Cette algorithme est considéré le plus populaire parmi les méthodes de Clustering où il doit ça à sa simplicité et sa capacité de traiter de larges ensembles de données [Kogan, 2007].

3.4.1 Algorithme

La méthode K-moyenne vise à partitionner n objets en k clusters dans lesquels chaque objet appartient au cluster ayant la moyenne la plus proche qui permet de produire exactement k groupes différents de la plus grande distinction possible. La figure 3.9 montre le processus du déroulement de la méthode k-moyenne étape par étape quant à l'algorithme est décrit comme suit [Mohd et al., 2012] :

Algorithme 1 : Algorithme K-moyenne

Entrées : E={t1, t2, . . . , tn} //ensemble d'élément
k // nombre de partitions souhaité ;
Sorties : K // Ensemble des partitions ;
début
  Attribuer des valeurs initiales pour les moyennes m1, m2,. . . , mk ;
  répéter
    Attribuer chaque élément ti au cluster «K» qui a la moyenne la plus proche;
    Calculez les nouvelles moyennes pour chaque partition;
  jusqu'à *les mêmes points sont attribués à chaque partition lors de tours consécutifs*;
fin

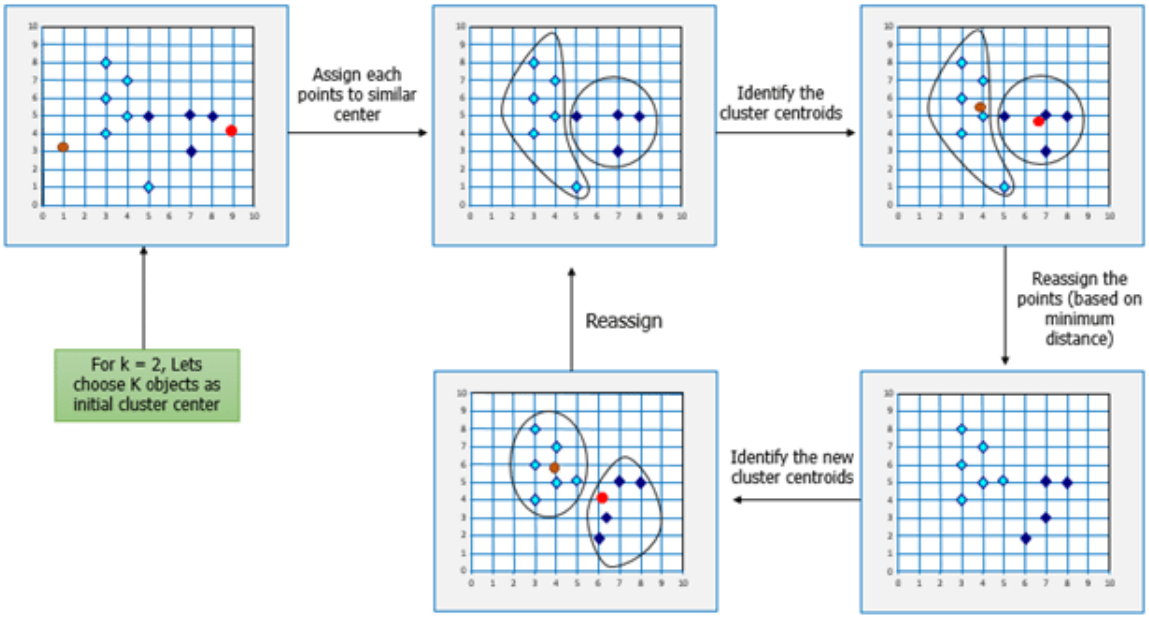


Figure 3.9 – Processus de déroulement de la méthode k-moyenne [Edureka, 2020]

3.4.2 Variations de l'algorithme

1. **K-means++ [David et Vassilvitskii, 2006] :** Un désavantage de l'algorithme K-means est qu'il est sensible à l'initialisation des centroïdes. Ainsi, si un centroïde est initialisé ou il est assez loin, il pourrait simplement se retrouver sans aucun point associé et en même temps, plusieurs clusters pourraient se retrouver liés à un seul centroïde. De même, plus d'un centroïde peut être initialisé dans le même cluster, ce qui entraîne un mauvais partitionnement. Cet inconvénient est surmonté grâce à l'amélioration introduite dans K-means++. Cet algorithme assure une initialisation plus intelligente des centroïdes et améliore la qualité du Clustering. Hormis l'initialisation, le reste de l'algorithme est le même que l'algorithme K-means standard.

2. **L'algorithme des k-médoïdes [Kaur et al., 2014] :** dit en anglais « Partitioning Around Medoids (PAM)», il s'agit d'une méthode qui consiste à rechercher des clusters, mais au lieu de prendre la valeur moyenne des objets d'un cluster comme point de référence, un médoïde peut être utilisé, qui est l'objet situé le plus au centre d'un cluster. Chaque objet dans l'ensemble des données est regroupé avec le médoïdes vers lequel il est le plus similaire. K-médoïde est plus robuste par rapport à K-moyennes cela revient à plusieurs avantages tels que dans K-Medoids, nous trouvons k objets représentatifs pour minimiser la somme des dissemblances des objets de données, alors que K-moyennes utilise la somme des distances euclidiennes au carré pour les objets de données. Et cette mesure de distance réduit le bruit et les valeurs aberrantes. En plus de son insensibilité aux objets isolés cette méthode s'adapte à n'importe quel type de donnée.

3. **L'algorithme de Forgy / Lloyd [Morissette et Chartier, 2013] :** L'algorithme de Lloyd publié en 1982 et l'algorithme de Forgy publié en 1965 sont tous deux des modèles de centroïdes par lots où le centroïde est le centre géométrique d'un objet convexe et peut être considéré comme une généralisation de la moyenne. Les algorithmes par lots sont des algorithmes où une étape de transformation est appliquée à tous les cas à la fois. Ils sont bien adaptés pour analyser de grands ensembles de données, car les algorithmes itératives tel que k-moyennes nécessitent de stocker l'appartenance au cluster de chaque cas ou de faire deux calculs du cluster le plus proche au fur et à mesure du traitement de chaque cas, ce qui est coûteux en calcul sur les grands ensembles de données. La différence entre l'algorithme Lloyd et l'algorithme Forgy est que l'algorithme Lloyd considère la distribution de données discrète tandis que l'algorithme Forgy considère la distribution continue.

4. **La méthode C-Moyennes Floues CMF (Fuzzy C-mean : FCM) [Malhotra et al., 2014] :** basée sur le principe du partitionnement flou, CMF est une forme de clustering dans laquelle chaque point de données peut appartenir à plus d'un cluster. Cet algorithme attribue une valeur d'appartenance aux éléments de données pour les clusters dans une plage de zéro à un avec la stipulation que la somme de leurs valeurs soit égale à un. Un autre point est introduit est celui du paramètre de fuzzification m qui est dans la plage [1, n]. Ce paramètre détermine le degré de flou dans ces clusters.

3.4.3 Avantages et inconvénients de la méthode k-moyennes [Kateb et Guerram, 2011]

La méthode k-moyennes apporte de nombreux avantages, faire partie des méthodes de classification non supervisée peut être considérée comme un avantage où elle nécessite aucune information sur les données afin de résoudre une tache. De plus, cet algorithme est applicable à tout type de données (mêmes textuelles), en choisissant une bonne notion de distance. Cette méthode n'est pas aussi parfaite, car elle fait face à plusieurs obstacles tels que la difficulté de trouver une bonne fonction de distance et de choisir K ou il est nécessaire d'avoir la bonne valeur afin d'éviter les mauvais résultats. Un autre inconvénient est la difficulté d'expliquer certains clusters où il n'est pas évident d'attribuer une signification aux groupes constitués.

3.4.4 K-moyennes dans les SDIs

La méthode de groupement(Clustering) K-moyennes a été adoptée dans plusieurs travaux qui concernent la détection d'intrusion parmi nous citons l'approche de Biswas et al.[Biswas et al.,2015] où les auteurs ont proposé un modèle de détection d'intrusions en utilisant la technique d'Analyse en Composantes Principales (ACP) pour sélectionner les caractéristiques pertinentes. K-moyennes était utilisée afin de grouper les attaques d'un type spécifique et les réseaux de neurones artificiels ont était introduit sur chacun des groupes formés. Une

autre approche hybride de détection d'intrusion similaire était proposée par Varuna et Natesan[Varuna et Natesan, 2015] où ils ont utilisé l'algorithme K-moyennes pour trouver la distance entre chaque objet dans l'ensemble de données et le nombre de centroïdes, à l'aide de ces informations rassemblées de nouvelles fonctionnalités ont été formées où elles ont ensuite été entrées dans le classificateur pour l'apprentissage et la détection.

Al-Yaseen et son équipe [Al-Yaseen et al., 2017] ont proposé un modèle hybride de détection d'intrusions à plusieurs niveaux en utilisant les Machines à Vecteurs de Support MVS(en anglais support vector machine : SVM) et les Machines d'Apprentissage Extrême MAE (en anglais : Extreme Learning Machine ELM) qui s'est avéré efficace pour détecter les attaques inconnues et connues. Dans cette approche l'algorithme k-moyennes était utilisé afin de générer un ensemble de données d'apprentissage réduit et de haute qualité. L'objectif principal était de séparer les données d'apprentissage originales en cinq catégories où les nouvelles données d'apprentissage sont celles qui comprennent le moins d'instances qui sont utilisées par la suite pour créer les modèles MVS et MAE tout en réduisant le temps d'apprentissage.

3.5 La méthode des K-plus proches voisins

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 [Ralaivola, 2007] [Morin, 2014].

3.5.1 Algorithme

L'idée générale de cette méthode est simple où à partir d'une base de données étiquetées, on peut estimer la classe d'une nouvelle donnée en regardant quelle est la classe majoritaire des k objets lui ressemblant le plus dans la base d'apprentissage (d'où le nom de l'algorithme). Le seul paramètre à fixer est k, le nombre de voisins à considérer. La figure suivante illustre le principe de fonctionnement de ce classifieur où la nouvelle donnée représentée par le cercle vert va être classée comme un triangle rouge qui est la classe majoritaire la plus proche [Data Analytics Post, 2018].

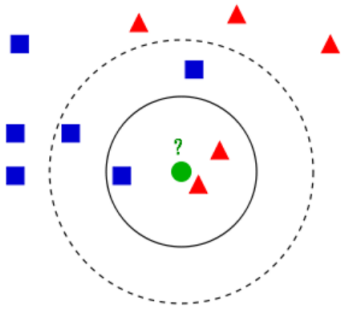


Figure 3.10 – Principe de fonctionnement de l'algorithme pour k=3 [Data Analytics Post, 2018]

Deux aspects importants de l'algorithme K-PPV peuvent être notés :

- à chaque nouvelle classification il est nécessaire de parcourir l'ensemble de la base d'apprentissage, ce qui en fait un algorithme qui n'est pas nécessairement très efficace ;
- problème du choix de la fonction de distance utilisée pour mesurer la proximité des objets. Il n'existe pas de distance/similarité universellement optimale et une bonne connaissance du problème traité guide généralement le choix de cette distance/similarité [Ralaivola, 2007].

Un pseudocode de la méthode peut être décrit comme suit [Tay et al., 2014] :

Algorithme 2 : Algorithme K-PPV

Classer(X,Y,x)
//X : données d'apprentissage ;Y : les labels des classes pour X ;x : nouvelle donnée
début
  pour *chaque exemple dans la base d'apprentissage* **faire**
    Calculer la distance $d(X_i,x)$;
  fin
  Calculer l'ensemble I contenant des indices pour les k plus petites distances ;
  Sorties : l'étiquette de la classe majoritaire dans I
fin

3.5.1.1 Variations du K-PPV

A) **1-PPV :**
Le classifieur 1-PPV est une simplification de la méthode classique du plus proche voisin où la valeur de k est prédéfinie comme 1 et non pas donnée par l'utilisateur. L'idée reste toujours la même, afin de classer une nouvelle observation X il suffit de trouver X' son plus proche voisin parmi les échantillons d'apprentissage et attribuer à X l'étiquette de X'. Ça simplicité rend son implémentation assez facile. De plus, cette

méthode fonctionne très bien dans les faibles dimensions pour les surfaces de décision complexes. Malheureusement, il reste toujours des lacunes, la classification par 1-PPV est lente et le classifieur souffre de beaucoup de la malédiction de la dimensionnalité (curse of dimensionality) où il s'agit d'un ensemble de phénomènes qui apparaissent lorsque les données sont traitéesdans des espaces de grandes dimensions alors qu'elles ne sont pas présentes dans des espaces de dimension moindre [Castelli, 2003].

B) **Le plus proche voisin pondéré :**
Le K-PPV pondéré est une version modifiée de l'algorithme classique, cette modification a était apporté suite à de nombreux problèmes qui affectent les performances de l'algorithme KPPV parmi le problème du choix du paramètre k. Si k est trop petit, l'algorithme serait plus sensible aux valeurs aberrantes et si k est trop grand, alors le voisinage peut inclure trop de points d'autres classes. Un autre problème est l'affectation des étiquettes aux classes, la méthode la plus simple consiste à prendre le vote majoritaire, mais cela peut être un problème si les voisins les plus proches varient considérablement dans leurs distances et les voisins les plus proches indiquent de manière plus fiable la classe de l'objet.

Dans k-PPV pondéré, les k points les plus proches reçoivent un poids à l'aide d'une fonction appelée fonction de noyau. Il est question de donner plus de poids aux points les plus proches et moins de poids aux points les plus éloignés. Toute fonction peut être utilisée comme fonction de noyau pour le classifieur K-ppv pondéré dont la valeur diminue à mesure que la distance augmente [Gou et al., 2012] [Hechenbichler et Schliep, 2004].

Le calcul des poids w_i est défini comme suit :

$$w'_i = \begin{cases} \frac{d(x',x_k^{NN})-d(x',x_i^{NN})}{d(x',x_k^{NN})-d(x',x_1^{NN})} & \text{, if } d(x',x_k^{NN}) \neq d(x',x_1^{NN}) \\ 1 & \text{, if } d(x',x_k^{NN}) = d(x',x_1^{NN}) \end{cases} \tag{3.1}$$

La Prédiction de la classe est réalisée via formule suivante :

$$y' = \arg\max_{y} \sum_{(x_i^{NN},y_i^{NN})\in T'} w'_i \times \delta(y = y_i^{NN}) \tag{3.2}$$

C) **Discriminant Adaptative Plus Proche Voisin DA-PPV [Hastie et Tibshirani, 1996] :**
L'un des problèmes de K-PPV est sa souffrance dans les bases de données à grande dimension. Le rayon relatif de la sphère la plus proche voisinecroît par $r^{1/p}$ où p est la dimension et r le rayon, pour $p = 1$ entraînant un biais grave au point cible x. La figure suivante illustre la situation d'un exemple simple :

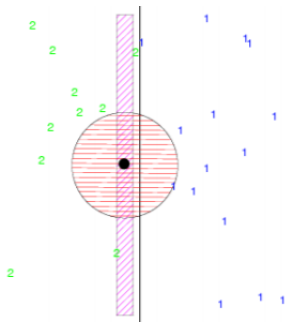


Figure 3.11 – Motivation du DA-PPV

Selon la figure 3.11 la bande verticale indique la région PPV en utilisant uniquement la coordonnée X pour trouver le voisin le plus proche du point cible (point solide). La sphère montre la région PPV en utilisant les deux coordonnées, et ça montre dans ce cas qu'elle s'est étendue à la région de classe 1 (et a trouvé la mauvaise classe dans ce cas). Ce phénomène se produit pour n'importe quelle K donné, les techniques du plus proche voisin sont basées sur l'hypothèse que localement les probabilités postérieures de classe sont constantes. Bien que cela soit clairement vrai dans la bande verticale en utilisant uniquement les coordonnées X, en utilisant X et Y ce n'est plus vrai. Ce problème est résolu par l'algorithme qui utilise une analyse discriminante linéaire locale pour estimer une métrique efficace pour le calcul des voisinages. De plus, cette méthode utilise un petit paramètre ϵ de réglage pour réduire ou étirer les quartiers (les voisinages s'étirent dans des directions pour lesquelles les probabilités de classe ne changent pas beaucoup).

3.5.2 K-PPV et leur implémentation dans les SDIs

Parmi les auteurs ayant utilisé K-PVV pour assurer une détection d'intrusions Nikhitha et Jabbar [Nikhitha et Jabbar, 2019] qui ont appliqué l'algorithme sur l'ensemble de données ISCX en classant les attaques par type comme Dos, prob, U2R et R2L. Leur méthode est validée par 5 et 10 validations croisées pour la classification. Plusieurs comparaisons ont été réalisées par les auteurs, le modèle proposé à augmenter les valeurs de la précision, du rappel et le F-mesure.

Liao et Vemuri proposent une approche similaire à celle de Nikhitha et Jabbar[Liao et Vemuri, 2002] où les auteurs introduisent la considération des fréquences des appels système pour déterminer si le comportement d'un programme est normal ou intrusif cependant, des bases de données distinctes des appels système à courte durée doivent être construites pour différents programmes ce qui peut prendre beaucoup de temps. Les auteurs ont proposé un nouvel algorithme pour modéliser les comportements des programmes afin de détecter les intrusions via K-PPV où l'ensemble des appels est représenté par un document word, leur test qui a été mené sur la base de données 1998 DARPA BSm a montré que leur intégration des appels systèmes dans ce classifieur a donné l'avantage à ne pas créer des profils sépa-

rés d'appel systèmes pour différents programmes ainsi une large réduction de calcul pour la classification.

3.6 Les machines à vecteurs de support

Les Machines à Vecteurs de Support (MVS) ou les Séparateurs à Vaste Marge (SVM) sont des modèles d'apprentissage supervisé avec des algorithmes d'apprentissage associés qui analysent les données utilisées pour résoudre des problèmes de discrimination et de régression.

Étant donné un ensemble d'apprentissage, chacun marqué comme appartenant à l'une des deux catégories existantes, un algorithme d'apprentissage MVS construit un modèle qui attribue de nouveaux exemples à une catégorie ou à l'autre, ce qui en fait un classificateur linéaire binaire non probabiliste.

Les MVSs peuvent aussi effectuer une classification non-linéaire à l'aide de ce qu'on appelle l'astuce du noyau(en anglais kernel trick), mappant implicitement leurs entrées dans des espaces d'entités de grandes dimensions [Cortes et Vapnik, 1995].

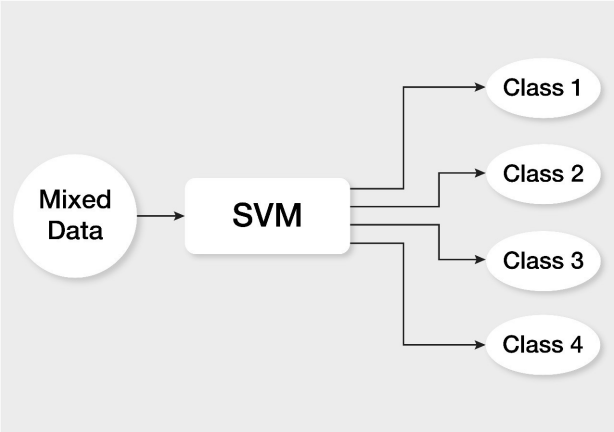


Figure 3.12 – Classifieur MVS [Towards Data Science, 2019]

3.6.1 Notions de base

Une MVS fonctionne en mappant les données sous forme de points dans l'espace, cartographiée de manière à ce que les exemples des catégories distinctes soient divisés par un séparateur, ces données sont ensuite transformées de telle sorte que le séparateur pourrait être dessiné comme un hyperplan.

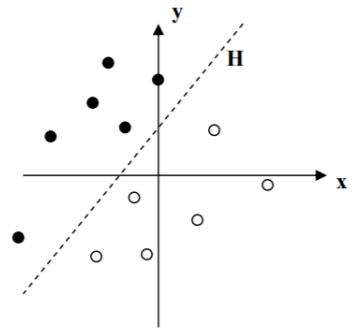


Figure 3.13 – Exemple d'un hyperplan séparateur [Hasan et Boris, 2006]

Les points les plus proches, qui sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support.

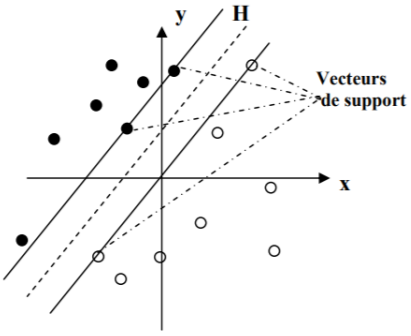


Figure 3.14 – Exemple de vecteurs de support [Hasan et Boris, 2006]

Ces deux mécanismes vont permettre aux nouveaux exemples à être mappés dans ce même espace et trouver leur catégorie en fonction du côté de l'écart sur lequel ils se trouvent.

Ce principe de fonctionnement montre que plusieurs hyperplans valides peuvent être trouvés, mais le but reste de trouver l'hyperplan valide optimal, celui qui passe « au milieu » des points des deux classes d'exemples. Intuitivement, cela revient à chercher l'hyperplan le «plus sûr ». En effet, supposons qu'un exemple n'ait pas été décrit parfaitement, une petite variation ne modifiera pas sa classification si sa distance à l'hyperplan est grande. Formellement, cela revient à chercher un hyperplan dont la distance minimale aux exemples d'apprentissage est maximale. On appelle cette distance « marge » entre l'hyperplan et les exemples. L'hyperplan séparateur optimal est celui qui maximise la marge. Comme on cherche à maximiser cette marge, on parlera de séparateurs à vaste marge [Cornuéjols, 2002].

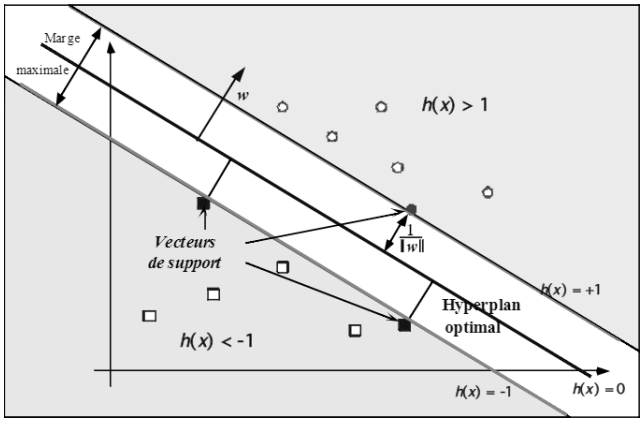


Figure 3.15 – Exemple de marge maximale (hyperplan valide)[Cornuéjols, 2002]

3.6.2 Marge maximale

Le but d'avoir une marge plus large est d'avoir plus de sécurité lorsqu'un nouvel exemple est en train d'être classé, de plus, si le classifieur qui se comporte le mieux vis-à-vis des données d'apprentissage est trouvés, il est clair qu'il sera aussi celui qui permettra au mieux de classer les nouveaux exemples. Dans le schéma qui suit, la partie droite montre qu'avec un hyperplan optimal, un nouvel exemple reste bien classé alors qu'il tombe dans la marge(Figure 3.16 (b)), contrairement à la partie gauche (Figure 3.16 (a)) qu'avec une plus petite marge, l'exemple se voit mal classé [Hasan et Boris, 2006].

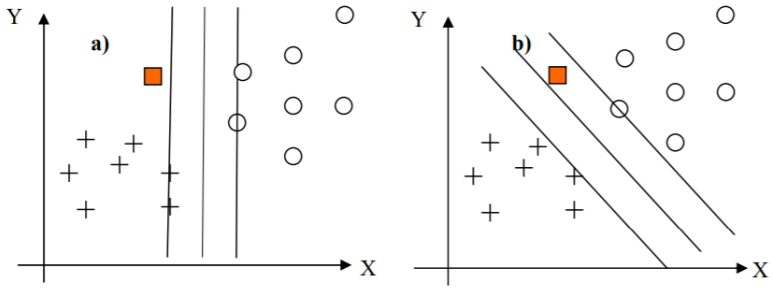


Figure 3.16 – MVS :(a) Hyperplan avec faible marge, (b) Meilleur hyperplan séparateur

En général, la classification d'un nouvel exemple inconnu est donnée par sa position par rapport à l'hyperplan optimal. Dans le schéma suivant, le nouvel élément sera classé dans la catégorie des « + ».

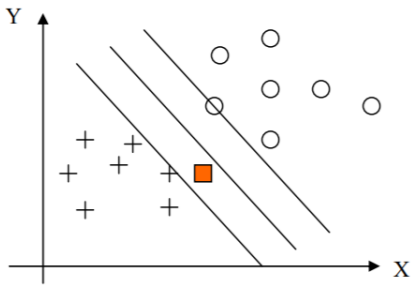


Figure 3.17 – Exemple de classification d'un nouvel élément

3.6.3 Les différents types de MVS

Il existe deux types de machine à vecteur de support.

3.6.3.1 MVS linéaire

C'est le cas où les données d'apprentissage sont linéairement séparables, c'est-à-dire qu'il existe un hyperplan qui sépare les données.

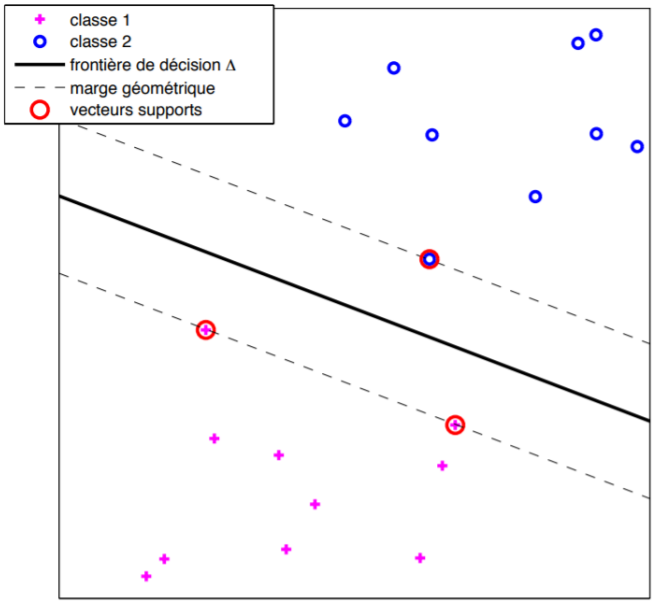


Figure 3.18 – Illustration de la notion de vecteur support dans le cas d'un problème linéairement séparable [WikiStat, 2016]

Dans ce cas, le but est de trouver l'hyperplan de marge maximale :

$$F(x) = w^T x + b \tag{3.3}$$

Ou w est (pas nécessairement normalisé) le vecteur normal à l'hyperplan et b est le déplacement par rapport à l'origine [WikiStat, 2016].

Cela ne résout pas le problème de l'hyperplan optimal d'où viennent les deux notions, la marge dure « hard margin » et marge molle « soft margin » [Mariette et Rahul, 2015].

A) **Marge dure :**

Il s'agit dans cette méthode de sélectionner deux hyperplans parallèles qui séparent les deux classes de données, afin que la distance entre elles soit aussi grande que possible. La région délimitée par ces deux hyperplans est appelée la "marge", et l'hyperplan à marge maximale est l'hyperplan qui se trouve à mi-chemin entre eux (voir Figure 3.19). Ces hyperplans peuvent être décrits par les équations suivantes :

$$w^T x_i + b = 1 \tag{3.4}$$

$$w^T x_i + b = -1 \tag{3.5}$$

Sachant que tout ce qui se trouve sur ou au-dessus de la limite présentée par l'équation (3.4) appartient à une classe, avec l'étiquette 1 et tout ce qui se trouve sur ou en dessous de la limite présentée par l'équation (3.5) appartient à l'autre classe, avec l'étiquette -1.

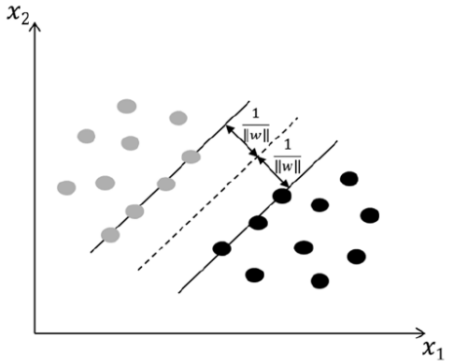


Figure 3.19 – Hyperplan de séparation à marge maximale dure[Mariette et Rahul, 2015]

B) **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 la figure suivante.

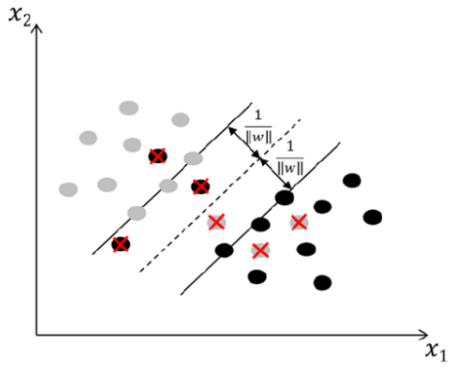


Figure 3.20 – Quelques erreurs de classification, dans le cadre du MVS à marge molle[Mariette et Rahul, 2015]

3.6.3.2 MVS non-linéaire [Hasan et Boris, 2006]

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 appelé « espace de re-description ».

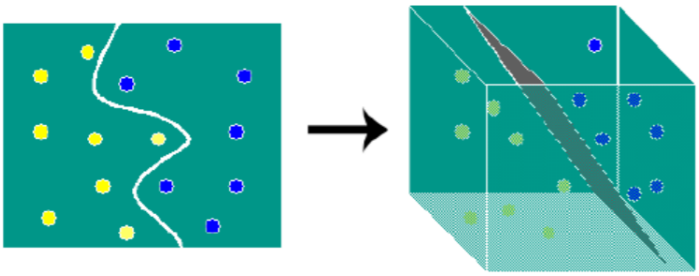


Figure 3.21 – Exemple de changement de l'espace de données.

Cette transformation non-linéaire est réalisée via une fonction noyau dite en anglais " kernel trick ". En pratique, quelques familles de fonctions noyaux paramétrables sont connues et il revient à l'utilisateur de la MVS d'effectuer des tests pour déterminer celle qui convient le mieux pour son application. On peut citer les exemples de noyaux suivants :

- Polynomiale (homogène) :

$$k(\overrightarrow{x_i}, \overrightarrow{x_j}) = (\overrightarrow{x_i} * \overrightarrow{x_j})^d \tag{3.6}$$

- Polynomiale (inhomogène) :

$$k(\overrightarrow{x_i}, \overrightarrow{x_j}) = (\overrightarrow{x_i} * \overrightarrow{x_j} + 1)^d \tag{3.7}$$

- Gaussien :

$$k(\overrightarrow{x_i}, \overrightarrow{x_j}) = exp(-\gamma \parallel \overrightarrow{x_i} - \overrightarrow{x_j} \parallel^2) \text{ pour } \gamma > 0 \tag{3.8}$$

3.6.4 Limitation des MVSs dans les SDIs [Jha et Ragha, 2013]

MVS est essentiellement une méthode d'apprentissage automatique supervisé conçue pour la classification binaire cela crée des limitations quand ils sont utilisés dans le domaine des SDIs. À cause de cette limitation structurelle, une MVS ne peut gérer que la classification de classe binaire alors que la détection d'intrusion nécessite une classification multi-classes. Bien qu'il y ait quelques améliorations, le nombre de dimensions affecte toujours les performances du classifieur basé sur MVS. Étant donné que les MVSs sont des méthodes d'apprentissage automatique supervisé, il faut des informations étiquetées pour un apprentissage efficace d'où la nécessité d'avoir des connaissances préexistantes pour une classification qui ne peuvent pas être disponibles tout le temps.

MVS traite chaque caractéristique des données de la même manière ce qui n'est pas idéal dans un ensemble de données de détection d'intrusion, car plusieurs attributs peuvent être redondants ou moins importants.

La construction d'une MVS en vue d'une détection d'intrusion prend du temps et nécessite un stockage important de l'ensemble de données ce qui est coûteux en calcul pour un réseau. De plus, une MVS nécessite le traitement de caractéristiques brutes pour la classification, ce qui augmente la complexité de l'architecture et diminue la précision de détection des intrusions.

3.7 Système Multi-Agents

L'intelligence artificielle distribuée (IAD) est classées en trois catégories en fonction des méthodes fondamentales utilisées pour résoudre les tâches, à savoir : l'IA parallèle, la résolution de problèmes distribuée (RPD) et les Systèmes Multi-Agents SMAs, ces derniers sont les plus récurrents.

Les SMAs sont composés d'entités autonomes appelées agents. Ces agents utilisent leurs interactions avec les agents voisins ou avec l'environnement pour apprendre de nouvelles connaissances afin de les utiliser pour décider et effectuer une action sur l'environnement pour résoudre la tâche qui leur a été attribuée. C'est cette flexibilité qui rend les SMAs aptes à résoudre des problèmes dans une variété de disciplines, telles que l'informatique. Pour développer les SMAs, plusieurs défis complexes doivent être surpassés tels que la coordination entre les agents, l'apprentissage et la sécurité sans oublier bien sûr l'étape de modélisation.

3.7.1 Agent

Le terme agent n'a pas de définition précise dans le domaine de l'intelligence artificielle cela est due à la diversité des caractéristiques des agents. Les agents peuvent être pré-

sents sous de nombreuses formes physiques (des robots) ou virtuelles (programme informatique). De plus, le domaine d'application de l'agent est très varié. Les chercheurs ont utilisé des termes comme softbots (agents logiciels), knowbots (agents de connaissances), taskbots (agents basés sur les tâches) basés sur le domaine d'application où les agents étaient employés [Nwana, 1996]. Afin de contourner ces problèmes, trois définitions peuvent être listées, car elles semblent être plutôt générales et largement acceptées par différentes communautés de recherche.

Selon [Russel et Norvig, 1995]« Un agent est une entité autonome flexible capable de percevoir l'environnement à travers les capteurs qui lui sont connectés et agit sur cet environnement via des effecteurs (en robotique et en génie mécanique, un effecteur est l'outil mis en mouvement par les actionneurs) ». Tandis que Ferber [Ferber, 1995] donne une définition très détaillée en prenant en compte toutes les composantes d'un agent où il dit « On appelle agent une entité physique ou virtuelle qui est capable de percevoir et d'agir dans son environnement et se représenter partiellement dans ce dernier, communiquer avec d'autres agents et qui est mue par un ensemble de tendances (buts, recherche de satisfaction). Cette entité possède ses propres ressources et compétences et peut se reproduire, un agent présente un comportement autonome qui est la conséquence de ses perceptions, représentations et de ses communications».

Une autre définition qui est introduite par Maess [Maess, 1995] qui dit « les agents autonomes sont des systèmes informatiques qui habitent un environnement dynamique complexe, détectent et agissent de manière autonome dans cet environnement, et réalisent ainsi un ensemble d'objectifs ou de tâches pour lesquels ils sont daignés ».

3.7.1.1 Type d'agent

Les experts ont réparti les agents en deux grandes catégories : les agents réactifs et les agents cognitifs où la différence entre ces deux catégories d'agents concerne la représentation de son environnement. Un troisième type d'agent dit hybride est une association des caractéristiques des deux autres types d'agents.

- **Les agents réactifs :** selon Brooks [Brooks, 1991] ce type d'agent est celui dont la représentation de l'environnement n'est que sub-symbolique, ce qui signifie qu'elle provient uniquement des perceptions de l'agent du monde « visible» à l'instant courant. Dans cette catégorie l'agent manque de raisonnement propre, ses actions sont limitées et il ne peut que réagir à des stimuli simples provenant de son environnement, il n'apas de mémoire et ne peut donc ni prendre le passé en compte, ni prévoir au-delà du court terme.

- **Les agents cognitifs :** contrairement aux agents réactifs, les agents cognitifs sont capables de raisonner sur leur représentation symbolique. De plus, ce type d'agents dispose d'une base de connaissances comprenant diverses informations nécessaires à l'accomplissement des tâches auxquelles ils sont associés.

Table 3.1 – Comparaison entre l'agent cognitif et l'agent réactif [Major et Reichgelt, 1990]

Agent cognitif	**Agent réactif**
Représentation explicite de l'environnement	Pas de représentation explicite de l'environnement
Peut tenir compte de son passé	Pas de mémoire locale
Agents complexes	Fonctionnement stimulus/action
Nombre d'agents réduit	Nombre d'agents élevé

- **Les agents hybrides :** ce type d'agent peut être vu comme un système à plusieurs niveaux qui combine les agents réactifs et cognitifs. Dans cette architecture, la couche de haut niveau ressemble à des agents purement cognitifs, s'occupe du raisonnement et de la prise de décision du système. La couche de bas niveau ressemble à des agents réactifs qui exécutent généralement des tâches élémentaires sous les ordres de la couche supérieure ou par leur propre initiative. Quant aux couches intermédiaires, elles peuvent regrouper les deux types d'agent (réactifs et cognitifs), le nombre de couches intermédiaires dépend du modèle du système à concevoir [Méziane, 2007].

3.7.1.2 Communication entre agents [Benmammar, 2009]

Dans le domaine des agents, la communication est vue comme une forme d'interaction qui permet d'échanger des informations et des connaissances, la communication peut être divisée en deux types :

1. Communication indirecte : par partage d'informations via l'environnement ;
2. Communication directe : par envoi de messages.

Afin que les agents puissent communiquer un ensemble de protocoles doit être respecté :

- Les protocoles de coopération : la coopération entre les agents consiste à décomposer les tâches en sous-tâches puis à les répartir entre les différents agents ;
- Les protocoles de coordination : aident les agents à gérer leurs engagements et maintenir la cohérence de l'ensemble ;
- Les protocoles de négociation : la négociation intervient lorsque des agents interagissent pour prendre des décisions communes, alors qu'ils poursuivent des buts différents.

3.7.2 SMA

Un SMA est un groupe organisé d'agents autonomes connectés qui interagissent selon certaines relations afin d'agir dans un environnement pour atteindre un objectif commun. Cela est réalisé par leur coopération ou concurrence et partage de connaissances entre eux. La figure suivante donne la structure générale d'un système multi-agents :

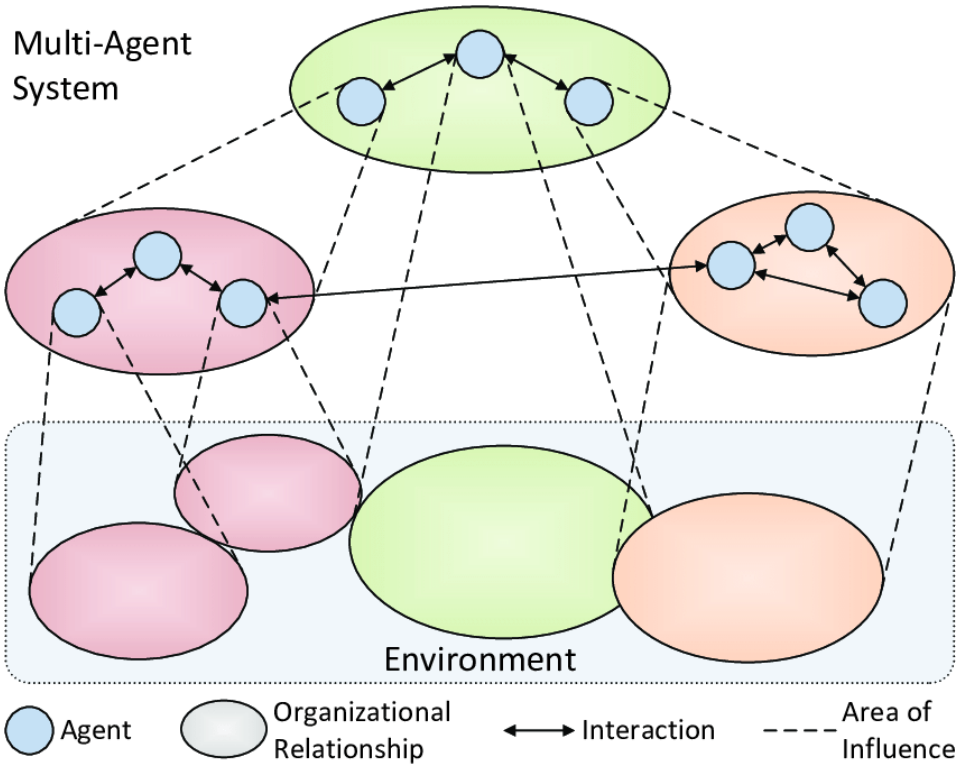


Figure 3.22 – Structure générale d'un système multi-agents [Lepuschitz, 2018]

Les systèmes multi-agents ont été largement adoptés dans de nombreux domaines d'application en raison de leurs avantages tels que l'augmentation de la vitesse et de l'efficacité des opérations grâce au calcul parallèle et au fonctionnement asynchrone.

3.7.2.1 Défis des systèmes multi-agents

Bien que les systèmes multi-agents apportent plus d'avantages que les systèmes mono-agents, il reste toujours les quelques obstacles à surpasser.

a) **Le contrôle de la coordination :** dans les systèmes multi-agents, l'environnement peut être modifié par n'importe quelle action réalisée par un agent dans le système à l'insu de ses voisins. Cela nécessite que chaque agent doive prédire l'action des autres agents afin de pouvoir trouver l'action optimale qui bénéficiera l'objectif initial. Ce type d'apprentissage simultané peut entraîner un comportement non-stable et peut éventuellement provoquer des problèmes qui peuvent être plus compliqués si l'environnement est dynamique. De plus, les agents dans le système doivent différencier entre les effets causés par les actions d'autres agents et l'altération dans l'environnement lui-même. Le contrôle de la coordination fait référence à la gestion des agents pour atteindre en collaboration leurs objectifs [Ma et al., 2015].

b) **L'apprentissage :** dans les SMAs, chaque agent décide de manière autonome de l'action appropriée pour atteindre son objectif, ces entités peuvent découvrir et prévoir les changements dans l'environnement et s'adapter aux situations imprévues en exploitent les algorithmes d'apprentissage automatique [Goldberg et Holland, 1988]. Cela leur permet de former des systèmes d'apprentissage multi-agents (Multi Agent Learning systems), de plus il existe plusieurs défis qui peuvent augmenter la complexité de l'adoption de systèmes d'apprentissage pour les SMA tels que l'environnement SMA qui peut être dynamique. Ainsi, les agents doivent fréquemment détecter les informations mises à jour à utiliser par la machine d'apprentissage, qui à son tour consomme une quantité importante de ressources d'agent. L'Apprentissage par Renforcement (AR) et la programmation génétique (PG) sont deux principales méthodes d'apprentissage automatique utilisées dans les systèmes d'apprentissage multi-agents.

c) **Sécurité [Wooldridge, 2009] :** la sécurité peut être vue très difficile à atteindre dans les SMAs en raison de la décentralisation, de la sociabilité et de la mobilité.

- **Décentralisation :** la vérification de l'identité des agents et la création de relations de confiance entre eux est un véritable défi. Cela est dû à l'absence d'une autorité centrale de confiance.
- **Sociabilité :** un agent est vulnérable contre les entités malveillantes qui peuvent partager des données falsifiées pour influencer ça décision, car il utilise les informations acquises auprès des agents voisins ou de l'environnement pour le processus décisionnel.
- **Mobilité :** un agent mobile peut être affecté par des agents malveillants. Si tel est le cas, il diffuse de fausses informations à tous les agents qu'il rencontre lors de ses déplacements.

Afin de remédier à ces problèmes, les SMAs doivent appliquer les cinq principes de la cryptographie (voir chapitre 1) l'ensemble des agents qu'ils contiennent.

3.7.2.2 Classification des systèmes multi-agents

La classification des SMAs peut être effectuée sur la base de plusieurs attributs différents, une classification générale englobant la plupart de ces attributs est présentée dans la figure ci-dessous [Balaji et Srinivasan, 2010].

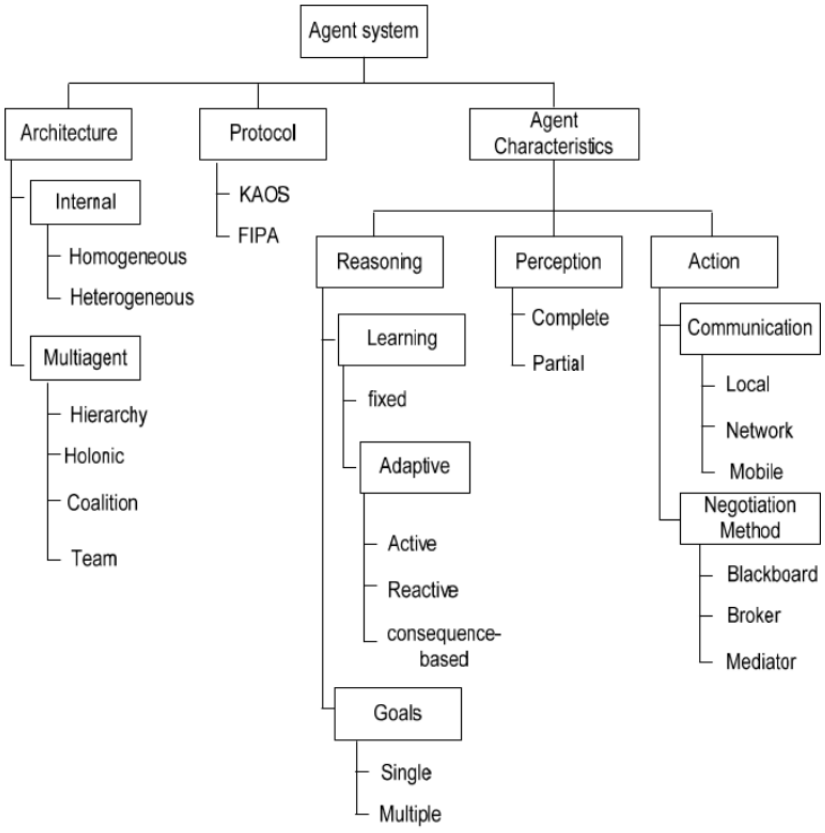


Figure 3.23 – Classification des Systèmes multi-agents [Balaji et Srinivasan, 2010]

3.7.3 Environnement de modélisation et simulation

Plusieurs plateformes de modélisation et d'évaluation des systèmes multi-agents existent, elles varient en fonction du domaine d'application et de leurs objectifs. Les plateformes SMA permettent aux développeurs d'améliorer le processus de conception et de réalisation de leurs applications en termes de temps grâce aux fonctions prédéfinies dans ces plateformes. Parmi ces plates-formes, nous citons :

- **JADE :** (Java Agent Development framework) est l'un des simulateurs open-source les plus utilisés dans les SMAs. La popularité de JADE découle de ses caractéristiques, cette plateforme est basée sur Java ce qui lui permet de bénéficier des bibliothèques externes, elle répond à la norme FIPA, son simulateur prend en charge la simulation de systèmes distribués. JADE cache la complexité des SMAs au concepteur grâce à l'interface graphique de conception et il peut être lié à Matlab ou ce dernier peut être utilisé pour étudier la performance du SMA[Eddy et al., 2015] [Bellifemine et al., 2007].
- **GAMA [GAMA, 2018] :** est une plate-forme de modélisation et de simulation pour la

construction de systèmes multi agents. GAMA présente un certain nombre d'avantages, tel que la possibilité d'être utilisé pour modéliser/simuler les SMA dans n'importe quelle application, de plus elle prend en charge GAML, un langage basé sur des agents de haut niveau et intuitifs, qui peut être facilement utilisé pour simuler les SMA, GAMA prend en charge les SMA à grande échelle qui se composent de millions d'agents.

- **MADKIT [Gutknecht et Ferber, 2001] :** est une plate-forme développée par le Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) de l'Université Montpellier II. MADKIT est libre pour l'utilisation dans l'éducation, elle est écrite en Java et elle est fondée sur le modèle organisationnel ALAADIN (cf. section 2.5). Elle utilise un moteur d'exécution où chaque agent est construit en partant d'un micro-noyau. Chaque agent a un rôle et peut appartenir à un groupe. Elle est dotée environnement de développement graphique qui permet facilement la construction des applications.

3.7.4 SMA et SDIs

Dans les systèmes de détection d'intrusion où le but est de surveiller les événements sur un système informatique ou un réseau et les analyser pour détecter des signes d'intrusion, cette tâche peut être rendue très difficile selon le trafic traité, cela a poussé plusieurs développeurs des SDIs à implémenter les SMAs pour rendre cette tache moins difficile.

Byrski et Carvalho [Byrski et Carvalho, 2008] utilisent des agents autonomes pour remplir les fonctions d'un seul détecteur de sécurité et communiquer avec des agents voisins pour partager des informations. La tâche principale de ces détecteurs est de surveiller les paquets acheminés et de construire un modèle du comportement normal / anormal du système. Une approche similaire était adoptée par Mokhtari et Moulkhaloua [Mokhtari et Moulkhaloua, 2018] où ils utilisent un SMA dans leur SDI pour réaliser la simulation du réseau et réduire le temps de la détection qui est assurée par les RNs et la méthode K-PPV.

Quant à Okamoto et Ishida [Okamoto et Ishida, 2007] proposent un Framework dans lequel chaque ordinateur génère divers agents qui sont partagés avec d'autres ordinateurs sur le LAN. Ce partage contribue à la diversité des agents. Des agents spécifiques à l'utilisateur sont générés pour chaque utilisateur sur chaque ordinateur ; chaque agent possède un profil unique présenté par un paramètre de la méthode de détection.

Liu et son équipe [Liu et al., 2007] utilisent les Systèmes Immunitaires Artificiels (SIA) afin d'améliorer la communication entre agents. Les auteurs proposent trois types d'agents LMA (Local Monitor Agent) et CMA (Central Monitor Agent) le premier analyse l'état du réseau local, tandis que le deuxième surveille l'ensemble du réseau alors que le troisième est identifié comme agent immunitaire qui est le noyau de ce modèle où il est distribué dans chaque nœud hôte afin d'identifier les intrusions.

3.8 Autres travaux

D'autres auteurs menant des recherches dans le domaine des SDIs ont fait recours à d'autres techniques et notamment à des hybridations, parmi ces auteurs Lv et son équipe [Lv et al., 2020] qui proposent un nouveau système de détection d'intrusions qui s'appuie sur des signatures d'attaque spécifiques pour distinguer les activités normales de celle qui sont anormal en utilisant les machine d'apprentissage extrême, il s'agit d'un type de réseau de neurones qui a qu'une seule couche de nœuds cachés, où les poids des entrées de connexion de nœuds cachés sont répartis au hasard et jamais mis à jour. La précision de prédiction de ces machines peut être relativement faible lorsque le modèle est appliqué à plusieurs ensembles de données de test inconnues. Afin de pallier ce problème, une fonction de noyau hybride était introduite basée sur une fonction de noyau polynomial et une fonction à base radiale pour améliorer la généralisation et la capacité d'apprentissage du modelé proposé. De plus, une combinaison de l'Algorithme de Recherche Gravitationnelle ARG (Gravitational Search Algorithm : GSA) et de l'algorithme d'Evolution Différentielle ED (Differential Evolution DE) sont utilisés pour optimiser les paramètres du modèle proposé, ce qui améliore ses capacités d'optimisation globale et locale lors de la prédiction des attaques. Ensuite l'algorithme d'Analyse en Composante Principale à Noyau ACPN (Kernel Principal Component Analysis : KPCA) est introduit pour réduire la dimensionnalité et l'extraction des caractéristiques des données de détection d'intrusion cela a permis à Lv et son équipe d'obtenir le modèle "KPCA-DEGSA-HKELM". Ce modèle a était testé sur différentes bases de données et comparé à plusieurs travaux où il a montré sa supériorité grâce aux résultats numériques qui valident à la fois la haute précision et le gain de temps.

Quant à Ye ainsi que Song et son équipe s'intéressent à l'utilisation des chaînes de Markov dans la détection d'anomalies [Ye, 2000] [Song et al., 2009]. Ils utilisent 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.

Subba et al. [Subba et al., 2015] proposent deux nouveaux modèles de détection d'intrusion basés sur des méthodes statistiques à savoir l'Analyse Discriminante Linéaire (ADL) qui est une méthode couramment utilisée comme technique de réduction de dimensionnalité pendant l'étape de prétraitement pour les applications d'apprentissage automatique et peut être même utilisée pour la classification. ADL permet de réduire les calculs Grâce à sa simplicité d'implémentation et son efficacité. Quant à la deuxième méthode qui est utilisée par ces auteurs est la Régression Logistique (RL) qui est utilisée pour les problèmes de classification binaire. Ces modèles proposés ont été testés avec la base de données NSL-KDD et comparés avec différentes méthodes où ils ont montré que leurs performances sont à égalité avec la

MVS et l'algorithme C4.5. De plus contrairement au modèle des SDIs basés sur une méthode complexe comme la MVS, les modèles de SDI basés sur ADL et RL sont calculés de manière plus efficace, ce qui les rend plus adaptés au déploiement dans la surveillance de réseau en temps réel et dans l'analyse de détection d'intrusion.

Nous pouvons également citer le travail de Javaid et son équipe [Javaid et al., 2016] qui proposent une approche basée sur l'Apprentissage Profond AP (Deep Learning : DL) , il s'agit d'un sous-ensemble de l'apprentissage automatique du le domaine de l'intelligence artificielle. L'AP est une fonction de l'IA qui simule le fonctionnement du cerveau humain dans la façon dont il traite les données et crée des modèles à utiliser dans le processus de prise de décision. Le modèle proposé dispose de réseaux qui peuvent apprendre de manière supervisée et non supervisée à partir de données étiquetées et non étiquetées. Afin de développer un système de détection d'intrusion efficace et flexible, les auteurs implémentent des autos-encodeurs épars, un type de réseau de neurones dédié à l'apprentissage profond. Finalement pour évaluer la précision de la détection des anomalies de cette approche les auteurs utilisent la base de données NSL-KDD.

3.9 Conclusion

La détection d'intrusion fait généralement recours à l'IA et plus particulièrement aux méthodes de classification. Afin de pouvoir trouver une approche permettant de réaliser une bonne détection d'intrusion réseau un état de l'art doit avoir lieu, sur cela nous avons présenté dans ce chapitre un ensemble méthodes et de techniques différentes utilisées pour réaliser des SDIs. Certaines d'entre elles dérivent du domaine de fouille de données telles que les arbres de décision, les séparateurs à vaste marge et les deux algorithmes K-moyenne et K-PPV. Nous avons également présenté les réseaux de neurones et les systèmes multi-agents ainsi que leur apport dans le domaine de la détection d'intrusion. Et pour finir nous avons clôturé le chapitre avec quelques travaux de la littérature faisant à leur tour appel à d'autres méthodes dérivant de l'IA ainsi que des hybridations telles que les chaînes de Markov et l'apprentissage profond.

Chapitre 4

Approches proposées, résultats et discussion

4.1 Introduction

Le rythme croissant du développement des réseaux Internet a entraîné l'apparition de menaces plus complexes et plus difficiles à identifier par les techniques de la sécurité informatique. Il s'agit alors de réduire les fausses alertes et augmenter le taux de détection. Ces différentes techniques opèrent en temps réel d'où le besoin de considérer le facteur temps en plus des différentes mesures d'évaluation des performances de la détection. Le but de notre travail alors et d'atteindre ces deux objectifs.

Nous allons présenter dans ce chapitre nos deux approches proposées en vue d'une détection d'intrusion réseau. La première approche est basée sur l'algorithme des K Plus Proches Voisins (K-PPV) tandis que la deuxième est une dérivée de la première nommée les Plus Proches Voisins Condensés (PPVC).

Afin d'atteindre les deux principaux objectifs cités en haut, nous avons procédé à la réduction de la base de données. Cette réduction consiste dans un premier temps à diminuer le nombre de paquets de la base de données d'apprentissage grâce à la méthode PPVC. Dans un deuxième temps en sein du paquet lui-même en réduisant le nombre d'attributs et en gardant ceux les plus pertinents grâce à des méthodes de réduction de dimensions telles que la méthode des taux de variance et l'élimination en arrière.

Afin d'assurer les simulations de nos approches proposées, nous avons fait recours au Système Multi- Agents (SMA). Grace à la notion de distribution d'expertise et physique ainsi qu'au parallélisme, ce paradigme a permit notamment à contribuer dans la réduction du temps.

4.2 Principe général de l'approche proposée

Afin de réaliser une détection d'intrusion réseau nous avons réalisé deux approches basées toutes les deux sur un modèle d'apprentissage supervisé, nous avons utilisé dans la première approche la méthode des K plus proche voisin (K-PPV) quant à la deuxième méthode nous avons opté pour les plus proches voisins condensés (PPVC). Nous avons utilisé chacune de ces méthodes afin de déterminer si un paquet reçut représente une attaque ou non.

Dans un premier temps nous avons remarqué que la méthode K-PPV est très gourmande en temps de calcul, cela est dû principalement à la nature de son algorithme sans négliger le nombre important des vecteurs inclut dans l'ensemble des données d'apprentissage. Suite à cela nous avons constaté que la meilleure façon de réduire le temps de calcul passe par la réduction de la dimensionnalité de l'ensemble des données d'apprentissage en se basant sur deux axes. Le premier axe consiste à la réduction du nombre paquet de la base d'apprentissage en appliquant dans un deuxième temps alors l'algorithme PPVC afin d'avoir les vecteurs les plus pertinents ou il est question d'éliminer les vecteurs affectant la prédiction négativement, ce qui permet également d'améliorer les taux de bonne détection. Le deuxième axe représente la sélection des caractéristiques dans le paquet lui-même où nous avons notamment fait appel à une comparaison entre diverses méthodes pour pouvoir procéder à la sélection. Ainsi nous avons pu réduire le temps d'exécution et obtenir une augmentation dans

la majorité des mesures d'évaluation.

Les SDIs sont des applications qui fonctionnent en temps réel, et avec le nombre de trafic existant aujourd'hui sur le net, la notion du temps est très importante. Or nous avons introduit le paradigme multi-agents et pour simuler notre SDIR et pour encore réduire le temps de détection grâce au parallélisme et à la distribution physique et d'expertise offerte par les SMAs.

Afin de tester nos approches proposées nous avons utilisé la base de données NSL-KDD [Unb, 2019] ou nous avons effectué plusieurs prétraitements pour pouvoir l'exploité tel que le codage et la normalisation. La figure suivant un schéma représentatif des différentes étapes de nos approches proposées.

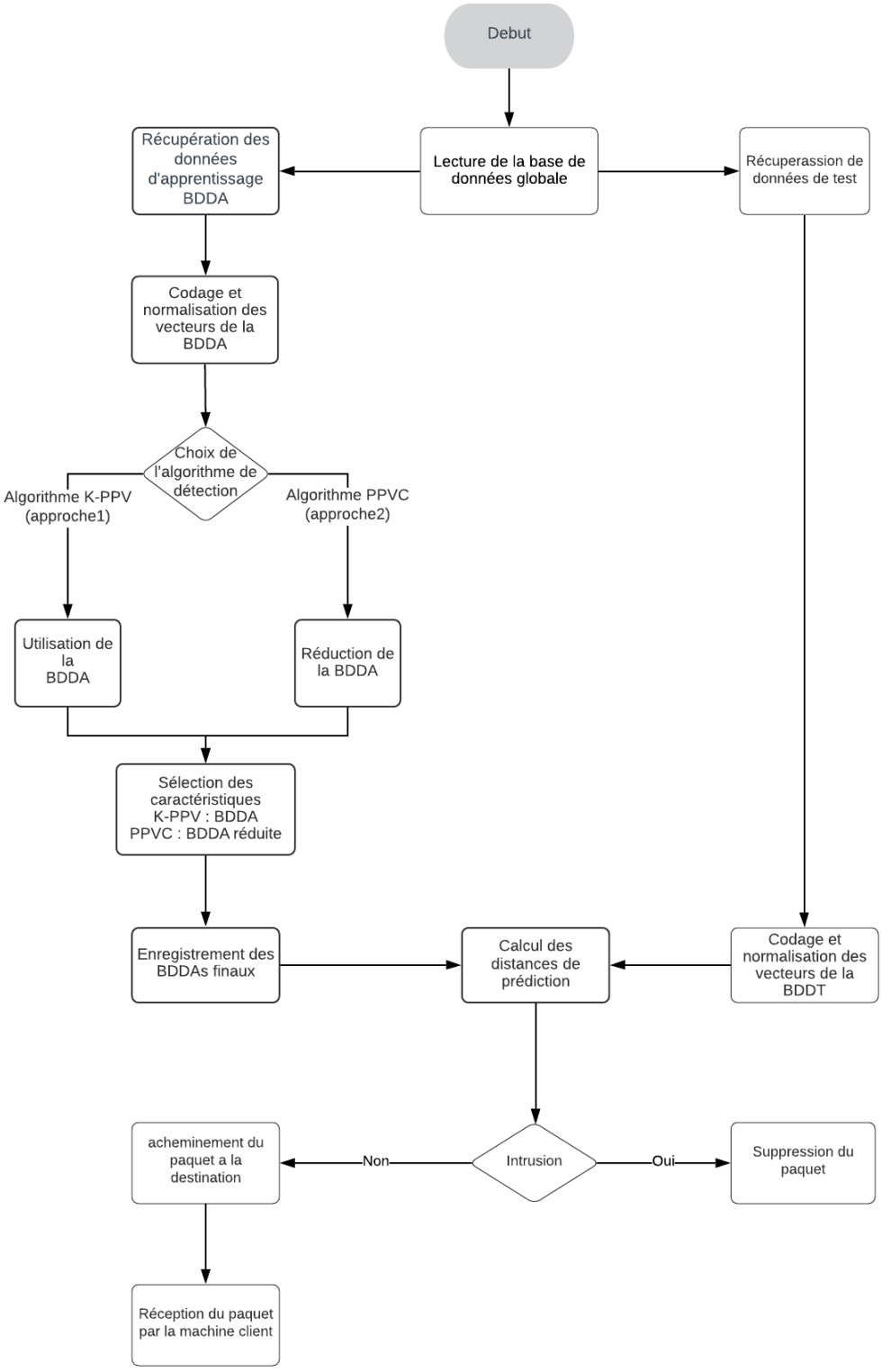


Figure 4.1 – Organigramme de nos approches proposées

4.3 Méthodes adoptées

4.3.1 K-Plus proches voisins

La méthode K-PPV consiste à attribuer un vecteur échantillon d'entrée y, dont la classification est inconnue, à la classe de son plus proche voisin [Cover et Hart, 1967].Cette idée peut être étendue aux voisins les plus proches, le vecteur y étant attribué à la classe qui est représentée par une majorité parmi les voisins les plus proches. Bien sûr, lorsque plusieurs voisins sont pris en compte, il existe la possibilité qu'il y ait une égalité entre les classes avec un nombre maximum de voisins dans le groupe de voisins les plus proches. Une façon simple de gérer ce problème est de restreindre les valeurs possibles de K. Par exemple, étant donné un problème à deux classes, si nous limitons K à des valeurs impaires, aucune égalité ne sera possible. Bien sûr, lorsque plus de deux classes sont possibles, cette technique n'est pas utile. Un moyen de gérer l'occurrence d'une égalité est le suivant. Le vecteur échantillon est affecté à la classe, des classes liées, pour laquelle la somme des distances de l'échantillon à chaque voisin de la classe est minimale. Bien sûr, cela pourrait encore conduire à une égalité, dans ce cas l'affectation est faite à la dernière classe rencontrée parmi celles concernées par l'égalité il s'agit alors d'une affectation arbitraire. De toute évidence, il y aura des cas où la classification d'un vecteur deviendra une affectation arbitraire, quelles que soient les procédures supplémentaires incluses dans l'algorithme [Keller et al., 1985].

Il existe plusieurs métriques de distance nous avons utilisé la distance euclidienne définie comme suit [O'neill, 2006] :

$$D = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2} \tag{4.1}$$

Le choix de la bonne valeur de k est un processus appelé réglage des paramètres, il est important pour une meilleure précision, il dépend généralement de l'ensemble des données d'apprentissage. Après une série de tests, nous avons opté pour la valeur K=1 dans notre travail afin d'obtenir les meilleurs résultats possibles.

4.3.2 Les plus proches voisins condensés

En raison des exigences de stockage qu'elle impose, la méthode K-PPV n'est pas un candidat idéal pour de nombreuses applications. En effet la méthode PPVC (en anglais condensed nearest neighbor) est suggérée comme une règle qui conserve l'approche de base K-PPV sans imposer des exigences de stockage aussi strictes [Hart, 1968].

PPVC assure une réduction de dimensionnalité qui aide à se débarrasser de la redondance et diminuer le nombre de paramètres libres. C'est une alternative aux modèles paramétriques abstraits, elle est dotée d'une procédure d'édition qui élimine sélectivement la partie redondante de l'ensemble d'apprentissage. Hart [Hart, 1968] a proposé de réduire au minimum le nombre de motifs enregistrés en stockant un sous-ensemble de l'ensemble d'apprentissage en

raison de similarité entre les différents vecteurs contenus dans cet ensemble [Alpaydin, 1997].

Algorithme PPVC :
Prenons T un ensemble d'apprentissage et C ensemble vide, PPVC opère comme suit [Angiulli, 2005] :

- Scanner tous les éléments de T, à la recherche d'un élément x dont le plus proche voisin dans C a une étiquette différente de celle de x ;
- Placer x dans C ;
- Répétez le scan jusqu'à qu'aucun élément x n'est ajouté à C ;
- Déterminer le plus proche voisin de x dans C se fait par la méthode KPPV classique avec k=1.

4.4 Base de données et prétraitement

Tout au long de notre travail, nous avons utilisé la base de données NSL-KDD [Unb, 2019] générée en 2009 à partir de KDD99 [KDD, 2007]. L'ensemble de données NSL-KDD a été utilisé pour résoudre certains des problèmes implicites de l'ensemble de données KDD 99. Cet ensemble de données contient une variété d'attributs, qui peuvent être utiles pour mesurer les attaques. Le jeu de données NSL-KDD a 22 544 instances (KDD test) et 125 973 instances pour le jeu de données de formation (KDD apprentissage) [Choudhury et al., 2019].

NSL-KDD contient 41 attributs plus un attribut classe, ces attributs sont présentés dans la table suivante accompagnés de leurs descriptions.

Table 4.1 – Attributs de la base NSL_KDD

Attribut	**Description**
`Duration`	La durée de la connexion
`protocol_type`	Protocole utilisé dans la connexion
`Service`	Service réseau de destination utilisé
`Flag`	NStatut de la connexion - Normal ou Erreur
`src_bytes`	Nombre d'octets de données transférés de la source à la destination en une seule connexion
`dst_bytes`	Nombre d'octets de données transférés de la destination à la source dans une connexion unique
`Land`	Si les adresses IP source et de destination et les numéros de port sont égaux, cette variable prend la valeur 1 sinon 0

wrong_fragment	Nombre total de fragments incorrects dans cette connexion
Urgent	Nombre de paquets urgents dans cette connexion. Les paquets urgents sont des paquets avec le bit urgent activé
Hot	Nombre d'indicateurs « hot » dans le contenu tels que : entrer dans un répertoire système, créer des programmes et exécuter des programmes
num_failed_logins	Nombre de tentatives de connexion échouées
logged_in	État de connexion : 1 si la connexion a réussi ; 0 sinon
num_compromised	Nombre de conditions compromises
root_shell	1 si la coque racinaire (root shell) est obtenue ; 0 sinon
su_attempted	1 si la commande «su root » a été tentée ou utilisée ; 0 sinon
num_root	Nombre d'accès "root" ou nombre d'opérations effectuées en tant que root dans la connexion
num_file_creations	Nombre d'opérations de création de fichiers dans la connexion
num_shells	Nombre d'invites du shell
num_access_files	Nombre d'opérations sur les fichiers de contrôle d'accès
num_outbound_cmds	Nombre de commandes sortantes dans une session ftp
is_host_login	1 si la connexion appartient à la liste "hot", c'est-à-dire root ou admin ; sinon 0
is_guest_login	1 si la connexion est une connexion « invité » ; 0 sinon
Count	Nombre de connexions au même hôte de destination que la connexion actuelle au cours des deux dernières secondes
srv_count	Nombre de connexions au même service (numéro de port) que la connexion actuelle au cours des deux dernières secondes

serror_rate	Pourcentage de connexions qui ont activé l'indicateur « flag » (4) s0, s1, s2 ou s3, parmi les connexions regroupées dans count (23)
srv_serror_rate	Pourcentage de connexions qui ont activé l'indicateur « flag » (4) s0, s1, s2 ou s3, parmi les connexions regroupées dans srv_count (24)
rerror_rate	Pourcentage de connexions qui ont activé l'indicateur « flag » (4) REJ, parmi les connexions regroupées dans count (23)
srv_rerror_rate	Pourcentage de connexions qui ont activé l'indicateur « flag » (4) REJ, parmi les connexions regroupées dans srv_count (24)
same_srv_rate	Le pourcentage de connexions qui étaient au même service, parmi les connexions regroupées en count (23)
diff_srv_rate	Le pourcentage de connexions qui étaient à différents services, parmi les connexions regroupées en count (23)
srv_diff_host_rate	Le pourcentage de connexions qui étaient vers différentes machines de destination parmi l'agrégat des connexions dans srv_count (24)
dst_host_count	Nombre de connexions ayant la même adresse IP d'hôte de destination
dst_host_srv_count	Nombre de connexions ayant le même numéro de port
dst_host_same_srv_rate	Le pourcentage de connexions qui étaient à différents services, parmi les connexions regroupées dans dst_host_count (32)
dst_host_diff_srv_rate	Le pourcentage de connexions qui étaient à différents services, parmi les connexions regroupées dans dst_host_count (32)
dst_host_same_src_port_rate	Pourcentage de connexions qui se trouvaient sur le même port source, parmi les connexions regroupées dans dst_host_srv_count (33)

dst_host_srv_diff_host_rate	Pourcentage de connexions qui étaient vers différentes machines de destination, parmi les connexions regroupées dans dst_host_srv_count (33)
dst_host_serror_rate	Pourcentage de connexions qui ont activé l'indicateur (4) s0, s1, s2 ou s3, parmi les connexions regroupées dans dst_host_count (32)
dst_host_srv_serror_rate	Pourcentage de connexions qui ont activé l'indicateur (4) s0, s1, s2 ou s3, parmi les connexions regroupées dans dst_host_srv_count (33)
dst_host_rerror_rate	Pourcentage de connexions qui ont activé l'indicateur (4) REJ, parmi les connexions agrégées dans dst_host_count (32)
dst_host_srv_rerror_rate	Pourcentage de connexions qui ont activé l'indicateur (4) REJ, parmi les connexions agrégées dans dst_host_srv_count (33)
Classe	Classification de l'entrée de trafic

L'attribut classe peut prendre 5 occurrences qui se divisent en deux catégories, la première concerne les paquets normaux déterminés par le label « normal » et ne contenant aucune attaque, et la deuxième catégorie est « attaque » spécifiant le type d'attaque à savoir [Salvatore et al., 2000] :

- DOS : les attaques par déni de service.
- R2L : accès non autorisé à partir d'une machine distante.
- U2R : accès non autorisé aux privilèges de super utilisateur local (root).
- Probing : surveillance et autres sondages.

4.4.1 Prétraitement de données

A) **Insertion SQL**

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)) même avec utilisation des agents (voire la table 4.2). Donc pour cela nous avons opté d'utiliser la méthode « copy » de PostgreSQL qui permet de copier un fichier.csv contenant l'ensemble des vecteurs en utilisant une seule requête.

Table 4.2 – Comparaison des approches d'insertion

Méthode	Insert (1 thread)	Insert (4 threads)	Copy (1 thread)
Temps	84s	24s	4s

B) **Codages des attributs textuels**

L'ensemble de données NSL-KDD brute contient parmi ses attributs trois variables textuels. D'où le besoin de la numérisation de ces derniers. Les champs 2,3 et 4 sont des champs textuels représentant respectivement les attributs ("protocol_type", "service" et "flag").

Tenant en compte que la sortie de l'algorithme KPPV est numérique, l'attribut classe doit aussi être codé. Les tables suivantes représentent les différents codes utilisés :

- **L'attribut protocol_type**

Table 4.3 – Codage de l'attribut « protocol_type »

Protocol	**Codage**
TCP	1
UDP	2
ICMP	3

- **L'attribut flag**

Table 4.4 – Codage de l'attribut « flag »

Flag	Codage
OTH	1
REJ	2
RSTO	3
RSTOS0	4
RSTR	5
S0	6
S1	7
S2	8
S3	9
SF	10
SH	11

- **L'attribut service**

Table 4.5 – Codage de l'attribut « service »

Service	Codage	Service	Codage
Aol	1	auth	2
Bgp	3	courier	4
csnet	5	ctf	6
daytime	7	discard	8
domain	9	domain_u	10
echo	11	eco_i	12
ecr_i	13	efs	14
exec	15	finger	16
ftp	17	ftp_data	18
gopher	19	harvest	20
hostnames	21	http	22
http_2784	23	http_443	24
http_8001	25	imap4	26
IRC	27	iso_tsap	28
klogin	29	kshell	30
Idap	31	link	32
login	33	mtp	34

name	35	netbios_dgm	36
netbios_ns	37	netbios_ssn	38
netstat	39	nnsp	40
nntp	41	ntp_u	42
other	43	pm_dump	44
pop_2	45	pop_3	46
printer	47	private	48
red_i	49	remote_job	50
Rje	51	shell	52
smtp	53	sql_net	54
Ssh	55	sunrpc	56
supdup	57	systat	58
telnet	59	tftp_u	60
tim_i	61	time	62
urh_i	63	urp_i	64
uucp	65	uucp_path	66
vmnet	67	Whois	68
X11	69	Z39_50	70

C) **Normalisation**

Quand il s'agit de calculs incluent plusieurs champs avec différents intervalles, il est nécessaire de normaliser tous les attributs à un intervalle commun. Le plus populaire est l'intervalle [0, 1] en utilisant la fonction de normalisation suivante :

$$X_{nouv} = \frac{X_{anc} - X_{min}}{X_{max} - X_{min}} \tag{4.2}$$

Où :

- X_{nouv} : la nouvelle valeur obtenue.
- X_{anc} : la valeur a normalisée.
- X_{max} : la valeur maximale du champ.
- X_{min} : la valeur minimale du champ.

Pendant la normalisation, nous avons aussi bénéficié de la rapidité de la méthode « copy ». La comparaison des temps de normalisation avec la méthode classique « insert » est représentée dans la table suivante.

Table 4.6 – comparaison des approches de normalisation

Méthode	Insert (1 thread)	Insert (4 threads)	Copy (1 thread)
Temps	58.375s	27.06s	14.25s

4.5 SMA proposé

Afin de simuler nos approches de détection d'intrusion réseau nous avons fait recours aux systèmes multi-agents. Dotés de différentes caractéristiques telles que la distribution, l'expertise, et le parallélisme ce paradigme nous a permis et une bonne modélisation ainsi qu'une réduction en temps de calcul surtout que le facteur temps est très important vu qu'il s'agit d'applications opérant en temps réel. Sachant que dans un SMA chaque agent est un thread, où chacun de ces agents a ses responsabilités commençant par la réception des paquets, leurs traitements et l'envoi au client comme c'est représenté dans la figure 4.2.

Les agents sont des systèmes informatiques qui exécutent des actions, ils sont capables d'être indépendants et autonomes afin de satisfaire leurs objectifs de conception. Alors que les objets encapsulent et ont une autonomie sur leur état, les agents ont une autonomie sur leur état et leur comportement. Ainsi, par exemple, il n'existe aucune notion d'invocation de méthode dans le monde orienté agent. Comme les agents ont le contrôle de leur propre comportement, ils doivent coopérer et négocier avec les autres pour atteindre leurs objectifs [Ciancarini et Wooldridge, 2000]. Notre SMA (voir figure 4.2) est composé de 5 types d'agents décrit comme suit :

1. **Agent contrôleur**

   Cet agent est le premier à être crée dès l'exécution du programme. Il est chargé d'abord de l'étape de prétraitement de la BDDA, puis il crée les agents analyseurs et il se met en mode récepteur. En étant en mode récepteur, cet agent ouvre un serveur (socket) et attend l'envoi des paquets par l'attaquant. Dé la réception du paquet et le prétraite et l'émet aux analyseurs.
2. **Agent analyseur**

   Le nombre d'agents analyseurs dépend du nombre de cœur du processeur utilisé(M) ; c'est l'agent contrôleur qui le détermine. Un agent analyseur est toujours en attente d'un paquet, dès la réception de ce dernier il exécute un algorithme de détection (K-PPV pour l'approche 1 et PPVC pour l'approche 2) afin de prédire la nature de ce paquet et de décider soit de l'envoyer s'il est sain ou non s'il ne l'est pas.
3. **Agent expéditeur**

   C'est un agent lancé par un agent analyseur après avoir déterminé la nature du paquet, il est chargé d'établir la connexion avec le client et lui envoie le paquet.
4. **Agent client**

   Le client est toujours en attente d'un ensemble de paquets qui lui sont envoyés par

l'agent expéditeur, chaque agent client représente une machine du réseau (N machines).

5. **Agent attaquant**

   Cet agent est chargé d'envoyer des vecteurs extraits de l'ensemble de données de tests de la base NSL-KDD.

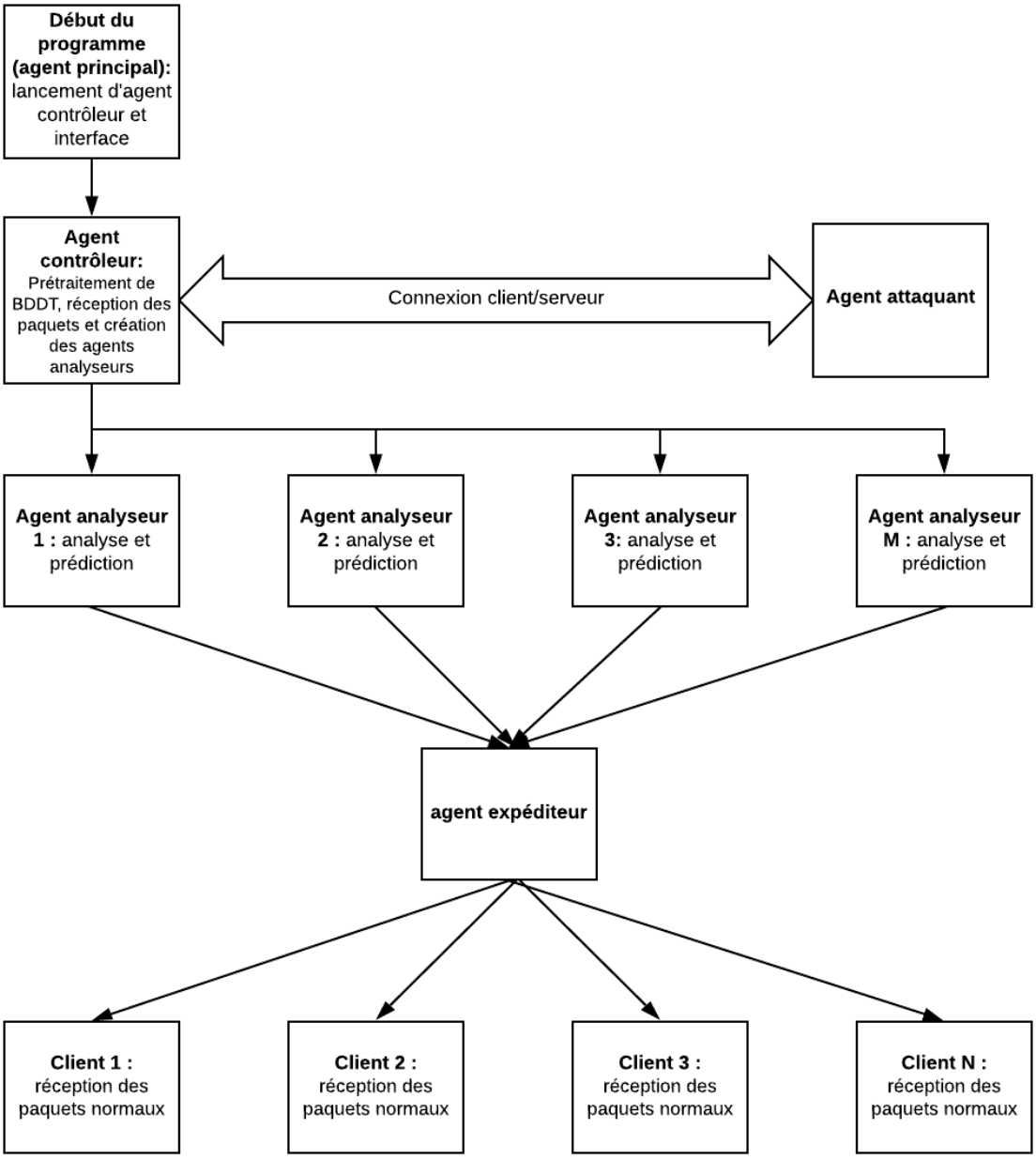


Figure 4.2 – Architecture de notre SMA

4.6 Réduction de dimensionnalité : Sélection de caractéristiques

Suite à la consistance et à la multiplication des données, les techniques de prédictions commencent à prendre un plus grand temps pendant l'analyse des données et l'apprentissage. Tandis que les méthodes non supervisées prennent aussi plus de temps pour la prédiction. Les notions de sélection de caractéristiques et de réduction de dimensionnalité donc ont été introduites. Elles consistent à trouver une représentation de données initiales dans un espace réduit. En général, on trouve deux méthodes de réduction de dimensionnalité [Chouaib, 2011].

- Réduction basée sur la sélection des caractéristiques qui consiste à identifier les caractéristiques les plus pertinentes à partir de l'ensemble général des variables de la base de données.
- Réduction basée sur la transformation des données nommée aussi extraction des caractéristiques, qui consiste à construire un ensemble de vecteurs de données réduit à partir de l'ensemble initial.

Dans notre approche proposée nous avons utilisé une réduction basée sur la sélection des caractéristiques dans l'objectif de réduire le temps pris par notre SDI lors de la prise de décision qui doit être faite en temps réel. Il existe plusieurs méthodes de sélection des caractéristiques, certaines peuvent être appliquées sur la base d'apprentissage directement, et ne dépendent pas du modèle à construire. D'autres utilisent les modèles d'apprentissage automatique pour éliminer des variables qui sont dispensables. Ces deux types d'approches peuvent aussi être utilisés, les unes avec les autres pour plus de fiabilité et un meilleur résultat.

4.6.1 Méthodes non basées sur le modèle

Il existe plusieurs méthodes qui s'intègrent sous cette catégorie à savoir :

- **La corrélation par paire :** cette méthode consiste à trouver les variables qui ont une forte corrélation entre elle, ce qui conduit à une redondance. Si deux variables sont fortement corrélées, l'élimination d'une variable ne causera pas assez de pertes d'information.
- **Analyse de cluster :** cette technique a le même principe que sa précédente avec une corrélation entre un groupe de variable au lieu de deux seulement.
- **Pourcentage des valeurs manquantes :** certains vecteurs ont des champs manquants qui peuvent nuire la prédiction. Le pourcentage est calculé en divisant le nombre de vecteurs contenant des valeurs manquantes sur le nombre total des vecteurs. L'élimination se fait pour les variables avec le plus grand pourcentage jusqu'à arriver à un seuil défini par le programmeur.
- **Taux de variance :** cette méthode consiste à calculer le taux de variance de chaque variable et éliminer celles avec les taux minimaux selon un seuil déterminé à partir des résultats.

Lors de la sélection des caractéristiques les plus pertinentes de la base NLS-KDD nous avons opté pour la méthode des taux de variance où la variance est calculée en utilisant la formule suivante [Mukhopadhyay et al., 2008] :

$$Var(x) = \sigma = 1/n \sum_{i=1}^{n} (x_i - \mu)^2 \tag{4.3}$$

La table 4.7 présentée ci-dessous correspond aux attributs de la base NSL-KDD ainsi que leurs variances. Plus la valeur de la variance est réduite plus l'attribut est candidat de l'élimination. Dans notre cas il s'agit des attributs dont la variance est proche du 0. La courbe présentée par la figure 3 nous a permis de déterminer le seuil a la valeur de 0.01 : var(seuil)=0.01.

Les attributs supprimés selon leurs taux de variance sont : « hot », « su_attempted », « land », « num_file_creations », « num_shells », « num_access_files », « num_failed_logins », « Urgent », « src_bytes », « num_root », « num_compromised », « dst_bytes », « is_host_login », « num_outbound_cmds ».

Table 4.7 – Taux de variance des valeurs des attributs de la base NSL-KDD

Variable (attribut du vecteur NSL-KDD)	**Variance**
logged_in	0,239128934
dst_host_same_srv_rate	0,201553931
srv_serror_rate	0,199827528
serror_rate	0,199321042
dst_host_srv_serror_rate	0,198619391
dst_host_serror_rate	0,197831281
same_srv_rate	0,193266727
dst_host_srv_count	0,188465967
dst_host_count	0,15135401
srv_rerror_rate	0,104746697
rerror_rate	0,102678108
dst_host_srv_rerror_rate	0,102053492
dst_host_same_src_port_rate	0,095478469
dst_host_rerror_rate	0,093976729
protocol_type	0,079863859
Flag	0,07232626
srv_diff_host_rate	0,067511352
Service	0,059988616
Count	0,050214717
dst_host_diff_srv_rate	0,035691163

diff_srv_rate	0,032513027
srv_count	0,020204899
dst_host_srv_diff_host_rate	0,01267051
is_guest_login	0,009333868
wrong_fragment	0,007141883
Duration	0,00368446
root_shell	0,001339758
Hot	7,80E-04
su_attempted	5,10E-04
Land	1,98E-04
num_file_creations	1,27E-04
num_shells	1,23E-04
num_access_files	1,22E-04
num_failed_logins	8,19E-05
Urgent	2,29E-05
src_bytes	1,81E-05
num_root	1,07E-05
num_compromised	1,02E-05
dst_bytes	9,42E-06
is_host_login	7,94E-06
num_outbound_cmds	0

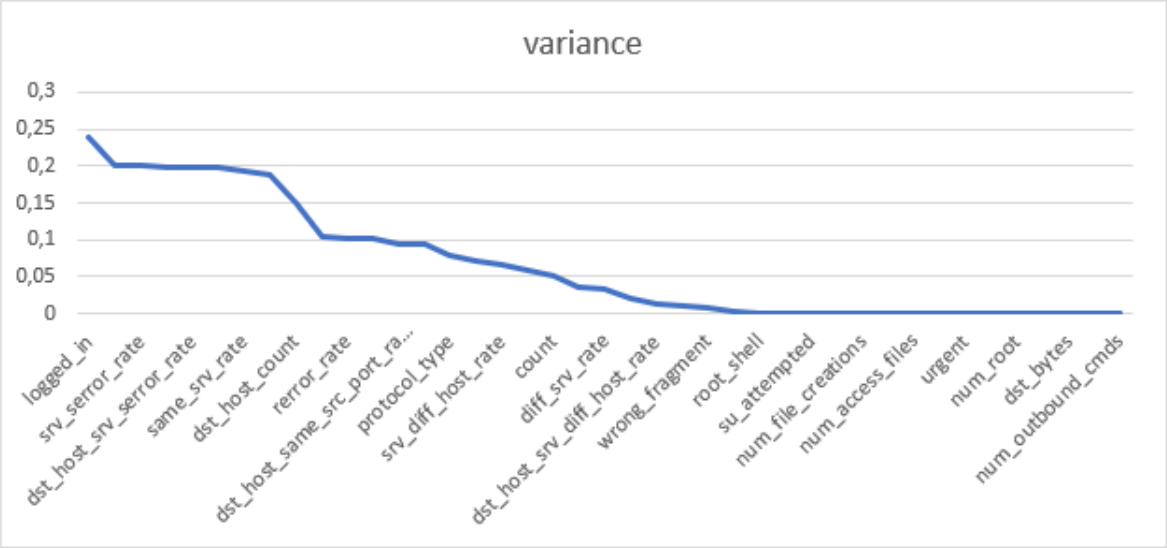


Figure 4.3 – Courbe de variance des attributs de la base NLS-KDD

4.6.2 Méthodes basées sur le modèle

Les méthodes basées sur le modèle de prédiction (ou la technique d'apprentissage) dépendent seulement de l'algorithme de classification choisi. Parmi ces méthodes nous citons :

- **Pré sélection :** consiste à identifier la meilleure variable en utilisant le modèle, puis ajouter la deuxième meilleure variable et ainsi de suite jusqu'à arriver à une condition d'arrêt qui dépend généralement de temps et de précision.
- **Elimination en arrière :** partage le même principe avec la présélection mais cette fois-ci en commençant avec toutes les variables du modèle et supprime la moins utile (par exemple celle qui conduit a une augmentation de précision lors de sa suppression). Le processus est répété jusqu'à l'arrivée à une condition d'arrêt.
- **Sélection pas à pas :** similaire à la présélection mais une variable peut être éliminée si elle affecte le modèle ou si elle n'est pas utile pour la réduction.

Durant notre travail nous avons opté pour l'approche de l'élimination en arrière grâce à sa rapidité, son adaptabilité à un usage prédictif. L'élimination en arrière n'exige aucune présélection de la meilleure variable, tout ce qui est demandé est d'éliminer à chaque tour la variable la moins utile [Sutter et Kalivas, 1993], les résultats sont présentés dans la section (4.7.3 c.)

4.7 Etude comparative

Nous avons pensé à utiliser d'autres approches afin de pouvoir comparer les résultats obtenus et enrichir notre étude. Pour cela, nous avons fait recours à l'application Weka, il s'agit d'un logiciel libre d'apprentissage automatique écrit en JAVA et développée à l'université Waikato en Nouvelle-Zélande [Eibe et al., 2016]. Ces différentes approches sont utilisées lors de la phase d'extraction des attributs les plus pertinents au sein des paquets de la BDDA afin de soutenir le choix et la décision prise à ce propos et de démontrer l'efficacité de l'étape de réduction dimensionnelle.

4.7.1 Approches utilisées lors de la comparaison

Les algorithmes choisis sont les arbres de décision (pour plus de détails voire chapitre 3) et plus précisément l'algorithme j48 ainsi que l'algorithme IBK (une version de K-PPV de Weka).

a) **Algorithme J48**

La classification est le processus de construction d'un modèle de classes à partir d'un ensemble d'enregistrements contenant des étiquettes de classe. L'algorithme d'arbre de décision consiste à découvrir le comportement du vecteur d'attributs pour un certain nombre d'instances. Toujours sur la base d'apprentissage, les classes des instances nouvellement générées sont déterminées [Kaur et al., 2014]. Cet algorithme génère les règles de prédiction de la variable cible. Grâce à cet algorithme la distribution critique

des données est facilement compréhensible [Nadali et al., 2011].

J48 est une extension d'ID3 (Dichotomiseur itératif 3). Les fonctionnalités supplémentaires de J48 prennent en compte les valeurs manquantes, l'élagage des arbres de décision, les plages de valeurs d'attribut continues, la dérivation des règles, etc.

Dans l'outil d'exploration de données WEKA, J48 est une implémentation Java open source de l'algorithme C4.5. L'outil WEKA fournit un certain nombre d'options associées à l'élagage des arbres. En cas de surajustement potentiel, l'élagage peut être utilisé comme un outil de précision. Dans d'autres algorithmes, la classification est effectuée de manière récursive jusqu'à ce que chaque feuille soit pure, c'est-à-dire que la classification des données doit être aussi parfaite que possible. Cet algorithme génère les règles à partir desquelles une identité particulière de ces données est générée. L'objectif est de généraliser progressivement un arbre de décision jusqu'à ce qu'il atteigne un équilibre de flexibilité et de précision [Kaur et al., 2014].

Algorithme 3 : Algorithme J48

J48(D)
Entrées : Ensembl de données D
début
  Arbre = {} ;
  si D est vide || autres condition d'arrêt **alors**
    FIN;
  fin
  pour chaque attributs $a \in D$ **faire**
    Calculer les critères d'impureté si nous nous séparons sur *a;*
    $a_{meilleure}$ = meilleur attribut selon les criteres calculés ci-dessus*;*
    Arbre = créer un nœud de décision qui teste $a_{meilleure}$à la racine *;*
    D_V =sous ensemble de données induites de D basé sur $a_{meilleur}$*;*
  fin
  pour chaque D_V **faire**
    $Arbre_V$ = J48(D_V)*;*
    Attacher $Arbre_V$ a la branche correspondante de Arbre *;*
  fin
  Retourner Arbre*;*
fin

b) **Algorithme IBK**

IBK (instance based K) est la version implémentée de K-PPV sous l'outil Weka, dont l'algorithme est défini dans la section (4.2.1).

Durant nos tests, nous allons utiliser la distance euclidienne, le nombre de voisins à choisir K = 1.

4.7.2 Mesures d'évaluation

La matrice de confusion est généralement utilisée pour évaluer ou visualiser le comportement des modèles dans des contextes de classification supervisée [Goutte et al., 2005]. Il s'agit d'une matrice carrée dans laquelle les lignes représentent la classe réelle des instances et les colonnes leur classe prédite. Si nous traitons une tâche de classification binaire, la matrice de confusion est une matrice 2×2 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 4.8 [Caelen, 2017]. La table illustre la matrice de confusion qui contient toutes les informations brutes sur les prévisions effectuées par un modèle de classification sur un ensemble de données

Table 4.8 – Matrice de confusion

Classe prédite	**Classe actuelle**	
	Normal	**Attaque**
Normal	VN	FP
Attaque	FN	VP

a) **Taux de réussite** [Bourouh et Kanoun, 2018]

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.

$$\text{Précision} = \frac{VN + VP}{VN + FP + FN + VP} * 100 \tag{4.4}$$

b) **Taux des Fausses Alertes (TFA)**

Les fausses alertes (FP) représentent la classification des paquets normaux (qui ne contient pas d'attaque) comme des attaques. Leur taux est calculé comme suit : [Pietraszek, 2004]

$$FP = \frac{FP}{FP + VN} * 100 \tag{4.5}$$

c) **Taux des Attaques Manquées (TAM)**

Les attaques manquées sont les paquets d'attaque classifiés comme étant des vecteurs normaux, ils sont calculés comme suit : [Gharaee et Hosseinvand, 2016]

$$FN = \frac{FN}{FN + VP} * 100 \tag{4.6}$$

d) **Taux des vrais positifs (TVP)**

Les attaques correctement classées sont connues sous le nom de Vrais Positifs (VP). La formule de calcul du TVP est : [Pietraszek, 2004]

$$VP = \frac{VP}{VP + FN} * 100 \tag{4.7}$$

4.7.3 Résultats et discussions

Dans cette partie, nous allons présenter les différents résultats obtenus suite à des séries de tests en utilisant l'ensemble complet des données d'apprentissage (100%) ainsi que la totalité de l'ensemble de tests (100%) de la base de données NSL-KDD. Les résultats sont divisés selon le nombre d'attributs présents dans les deux ensembles en commençant par la totalité des attributs (42) jusqu'à l'attribution de l'élimination en arrière et en passant par l'élimination des attributs par leurs taux de variance.

a) **Résultats sans suppression d'attributs**

La table 4.9 présentée ci-dessous présente les résultats de tests des différentes mesures d'évaluation sur nos deux approches implémentées K-PPV et PPVC ainsi que les approches de l'application weka J48 et IBK.

Selon les résultats obtenus, PPVC est l'algorithme le plus performant en mesure de taux de réussite où cette dernière a atteint 86.03%, suivit des algorithmes K-PPV et IBK avec des taux de réussite presque semblables qui sont respectivement 84.96%, et 84.93% et en dernière position vient l'algorithme J 48 avec un taux de précision de 84.64%.

Quant au TFA l'algorithme PPVC a obtenu la valeur la plus réduite égale à 2.25% suivit de l'algorithme K-PPV avec un taux de 2.27%, puis les algorithmes iIBK et j 48 qui ont enregistré respectivement des taux de 2.28% et 3.88%. Ceci montre encore une fois que l'algorithme PVVC est plus performant en réduisant le nombre des fausses alertes.

Comme le temps est très important dans ce contexte, une comparaison du temps d'exécution entre ces approches est réalisée. Encore une fois l'algorithme PPVC a montré son efficacité avec le temps d'exécution le plus réduit de 3.4 secondes, suivi de l'algorithme J48 avec 56 secondes, puis des algorithmes KPPV et IBK avec respectivement 267 et 651 secondes.

Table 4.9 – Performances des approches K-PPV, PPVC, J48 et IBK avec 42 attributs

Algorithme	**Taux de réussite**	**TFA**	**TAM**	**TVP**	**Temps**
K-PPV	84.93%	2.27%	28.75%	71.25%	267s
PPVC	86.03%	2.25%	26.5%	73.5%	3.4s
J48	84.64%	3.88%	23.95%	76.05%	56s
IBK	84.96%	2.28%	28.66%	71.34%	651s

b) **Suppression des attributs selon leurs taux de variances**

Dans la table 4.10 présentée ci-dessous, nous pouvons remarquer que la suppression des attributs ayant un taux de variance minimal (voir section 4.6.1) a pu conduire à une augmentation de précision pour les quatre approches et une légère réduction en temps. Cette procédure nous a permis de gagner en temps de calcul en réduisant ce dernier par rapport aux tests précédents.

Les résultats obtenus illustrés dans la table 4.10 montrent encore une fois que l'algorithme PPVC est toujours l'algorithme le plus performant en taux de réussite, où cette dernière a atteint 87,27%, suivi respectivement des algorithmes K-PPV, IBK et J48 avec des taux de réussite de 86.50, 85.78 et 84.64.

Une réduction en temps de calcul est enregistrée sur l'ensemble des algorithmes due principalement au nombre d'attributs réduit , où l'algorithme PPVC reste le plus rapide avec un temps de calcul de 3 secondes, suivi de l'algorithme J48 de 35 secondes, l'algorithme IBK avec 207 secondes , et en dernière position l'algorithme K-PPV gourmand en temps avec 230 secondes .

Table 4.10 – Performances des approches K-PPV, PPVC, J48 et IBK avec 28 attributs

Algorithme	**Taux de réussite**	**TFA**	**TAM**	**TVP**	**Temps**
K-PPV	86.50%	3.4%	24.28%	75.72%	230s
PPVC	87.27%	3.47%	22.63%	77.63%	3s
J48	84.64%	2.36%	29.25%	70.75%	35s
IBK	85.78%	2.48%	26.77%	73.23%	207s

c) **Suppression des attributs via l'élimination en arrière**

L'élimination en arrière nous a permis de gagner beaucoup de temps et d'augmenter le taux de réussite à des niveaux satisfaisants pour la majorité des approches, mais particulièrement pour le PPVC où l'augmentation enregistrée est de 8.27%, quant au temps de calcul une diminution de 1400 ms a été remarquée pour un total de 18794 vecteurs de tests. Quant à la mesure d'attaques manquées TAM à son tour a enregistré une diminution, où elle est passée de 22.63% à 4.64%.

Table 4.11 – performances après attribution de l'élimination en arrière

Algorithme	Attribut restant	Taux de réussite	TFA	TAM	TVP	Temps
K-PPV	2,3,4,8,12,14,22, 23,29,30,38,41,42	87.45%	3.07%	22.66%	77.34%	165s
PPVC	2,3,4,8,12,14,22, 23,26,28,29,30,33, 34,36,38,39,41,42	95.54%	4.64%	4.27%	95.73%	2.5s
J48	2,3,8,12,22,23,26, 33,34,41,42	87.95%	2.75%	20.87%	79.13%	12s
IBK	2,3,4,8,12,14,22, 23,29,30,38,41,42	88.03%	3.25%	21.27%	78.73%	124s

Les optimisations réalisées grâce à la réduction dimensionnelle en performances de détection et en temps de calcul sont illustrées dans la figure 4.4 à la figure 4.6 Ou on remarque clairement une réduction en temps de calcul et une augmentation en taux de réussite. Une comparaison des mesures d'évaluation TFA, TAM et TVP des approches KPPV, PPVC, J48 et IBK après suppression des attributs est réalisée afin de démontrer l'efficacité de l'algorithme notre approche proposée PPVC avec sélection des attributs les plus pertinents.

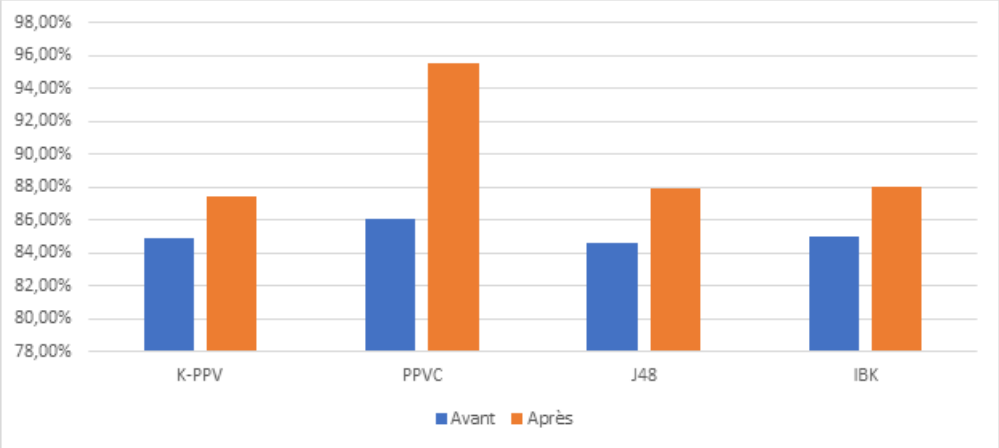


Figure 4.4 – Comparaison des taux de réussite des approches K-PPV, PPVC, J48 et IBK avant et après sélection des caractéristiques

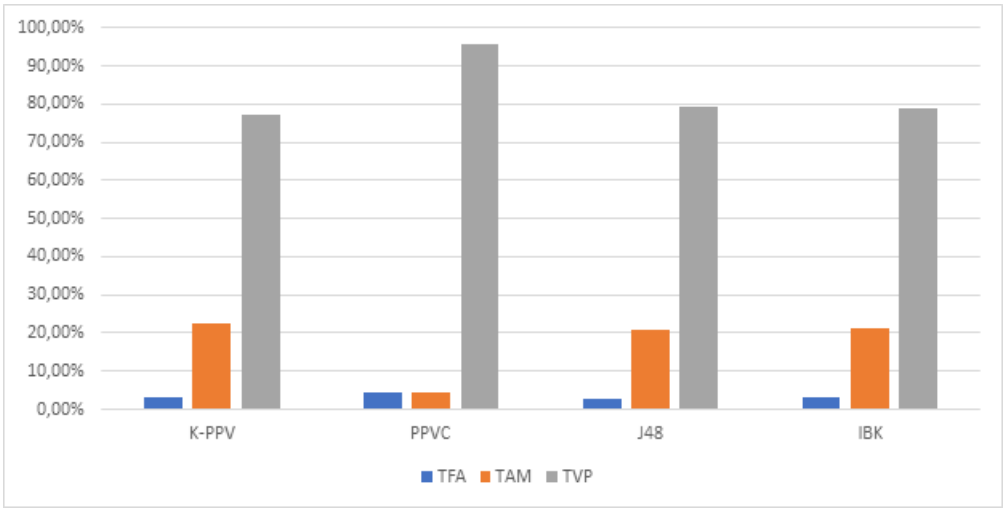


Figure 4.5 – Comparaison des mesures d'évaluation TFA, TAM et TVP des approches K-PPV, CPPV, J48 et IBK âpres suppression des attributs

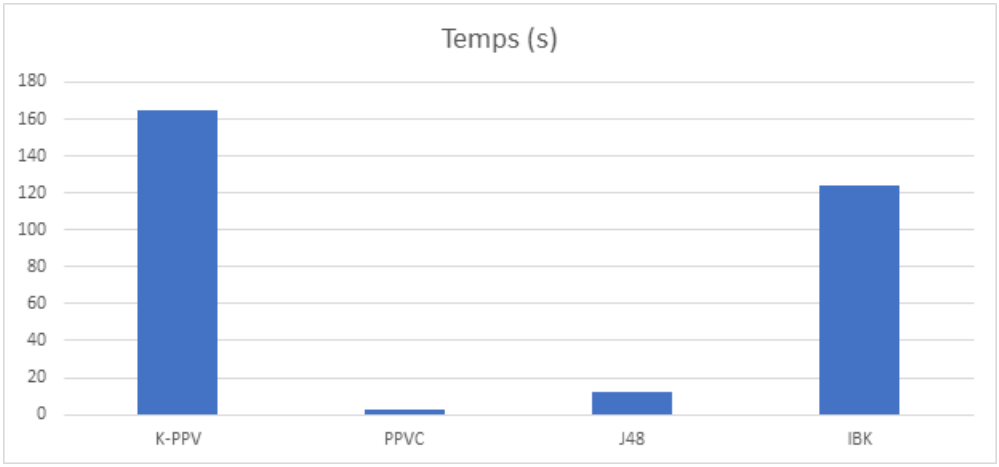


Figure 4.6 – Comparaison en temps de calcul entre les approches KPPV, PPVC, J48 et IBK en secondes

d) **Améliorations apportées par les SMAs**

L'introduction des SMAs dans une application peut mener à diminution de temps en raison des calculs simultanés et parallèles. C'est ce que nous avons aperçu sur nos deux approches (PPVC et K-PPV) après avoir tester la mesure de temps d'exécution avec et sans SMA.

Les trois tables suivantes représentent l'apport apporté en mesure de temps sans et avec utilisation des SMAs.

Table 4.12 – Amélioration en temps de calcul sur l'ensemble total des attributs (42)

Approche	Sans SMA (1 thread)	Avec SMA (4 threads)
PPVC	3.4s	2.4s
K-PPV	267s	259s

Table 4.13 – Amélioration en temps de calcul après la première sélection (taux de variance) avec 28 attributs

Approche	Sans SMA (1 thread)	Avec SMA (4 threads)
PPVC	3s	2s
K-PPV	230s	221s

Table 4.14 – Amélioration en temps de calcul après l'élimination en arrière

Approche	Sans SMA (1 thread)	Avec SMA (4 threads)
PPVC	2.5s	1.7s
K-PPV	165s	164s

4.8 Simulation de notre SDI

4.8.1 Outils utilisés

1. **Caractéristiques techniques du matériel utilisé**

Processeur : Intel Core i5-4460.
Mémoire : 8 Go.
Système d'exploitation : Windows 10 64bits.

2. **Environnement De Développement ED**

La nature des SDIs réseau est d'être déployée comme un serveur dans un réseau, c'est pour cette raison que nous avons utilisé le langage de programmation JAVA. Ces particularités en matière de programmation orientée objet et de connectivité grâce aux sockets ainsi que ces avantages en conception de système multi-agent, programmation parallèle et l'intégration d'un serveur web avec possibilité d'accès depuis un navigateur web, nous ont permis de bien concevoir notre application.

Afin de pouvoir exploiter JAVA, nous avons opté à l'utilisation de NetBeans 11.2 comme ED, ce dernier offre une meilleure expérience avec ses bibliothèques intégrées de gestion de base de données, de serveur web et d'autres. NetBeans offre aussi la possibilité de construire une application portable qui peut s'exécuter sous n'importe quel ordinateur possédant JAVA

[NetBeans, 2019].

3. **PostgreSQL**

PostgreSQL est un système de base de données puissante, objet-relationnel open source. Il a plus de 25 ans de phase de développement actif et une architecture éprouvée lui offrant ainsi une solide réputation de fiabilité, d'intégrité des données et d'exactitude [PostgreSQL, 1996].

4. **Apache Tomcat**

Apache Tomcat est un conteneur web libre de servlets et JSP que nous avons utilisé pour concevoir notre Interface homme machine.

4.8.2 Interface de notre SDI

A) **Machine serveur**

Notre SDIR résidera sur cette machine, c'un serveur Apache Tomcat qui permet de créer une page web pour notre application. Dans ce qui suit nous allons présenter son IHM avec quelques détails.

- **Fenêtre principale**

  Après avoir lancer le serveur, une fenêtre de configuration simple s'ouvrira, elle permet a l'utilisateur de décider l'approche qu'il veut utiliser, ensuite démarrer le serveur et d'ouvrir le tableau de bord avec son navigateur par défaut.

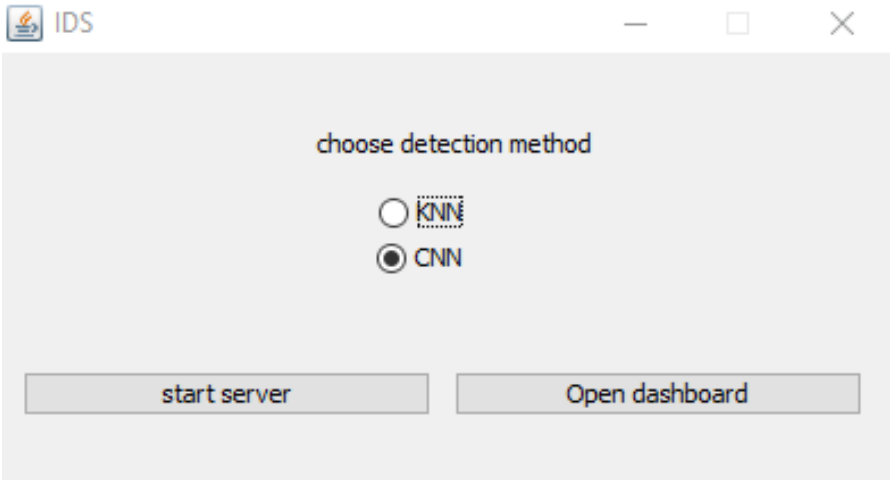


Figure 4.7 – Fenêtre principale de notre SDI

- **Tableau de bord (Dashboard)**

  C'est une page web contenant les informations du serveur, des graphiques pour une

représentation en temps réel, et des tables pour la visualisation du trafic et de la prédiction (résultats de détection).

(a) **Première fenêtre :**
Cette fenêtre est un aperçu général de ce qui se passe en temps réel au niveau du serveur, les graphiques représentent le nombre de paquets reçus et classés ; la fenêtre illustre notamment les performances de détection.

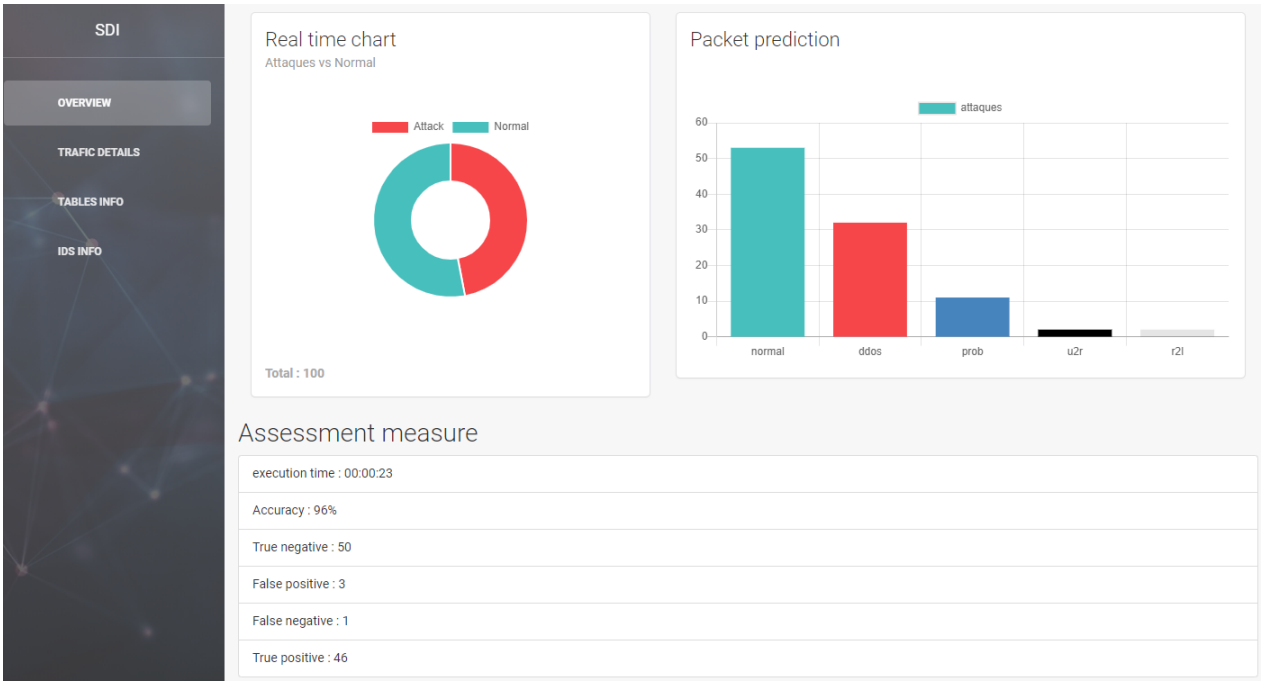


Figure 4.8 – Fenêtre d'aperçu

(b) **Deuxième fenêtre :** la première partie de la fenêtre est une table illustrant les paquets reçus en temps réel (voir figure 4.9), tandis que la deuxième représente les détails de chaque paquet où l'utilisateur peut sélectionner un élément du tableau et voir ces détails dans la partie 2 (voir figure 4.10).

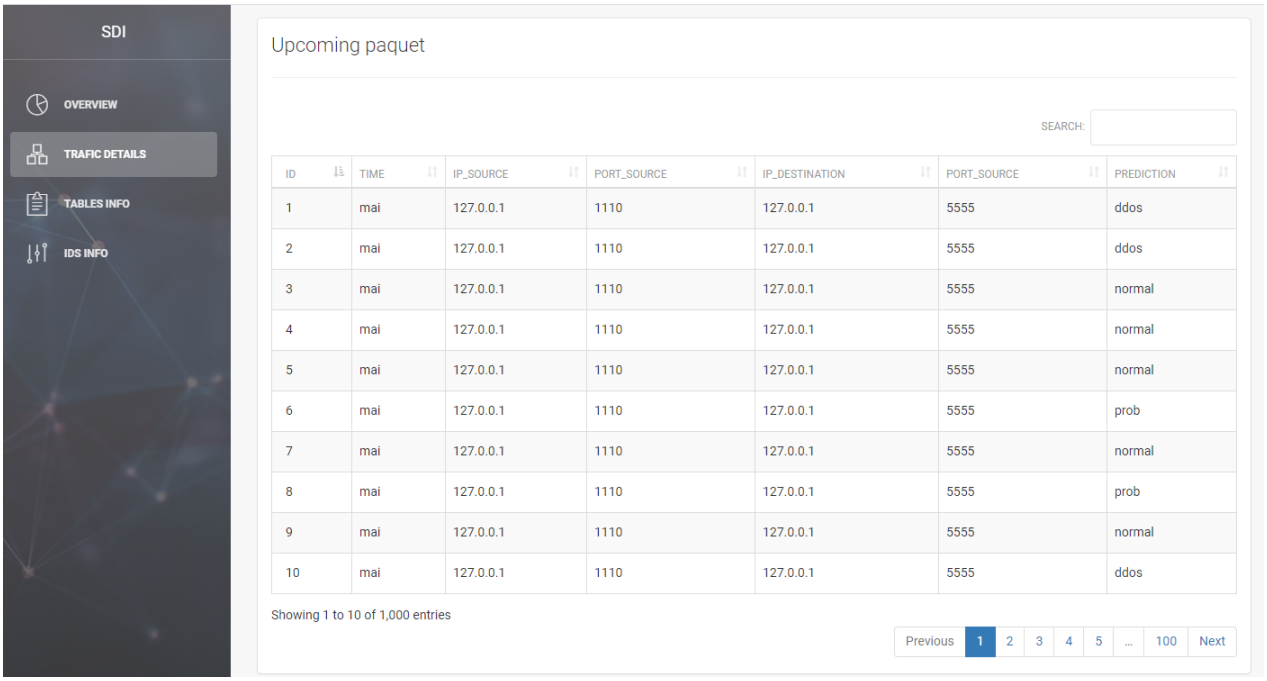


Figure 4.9 – Fenêtre pour les détails du trafic – partie 1

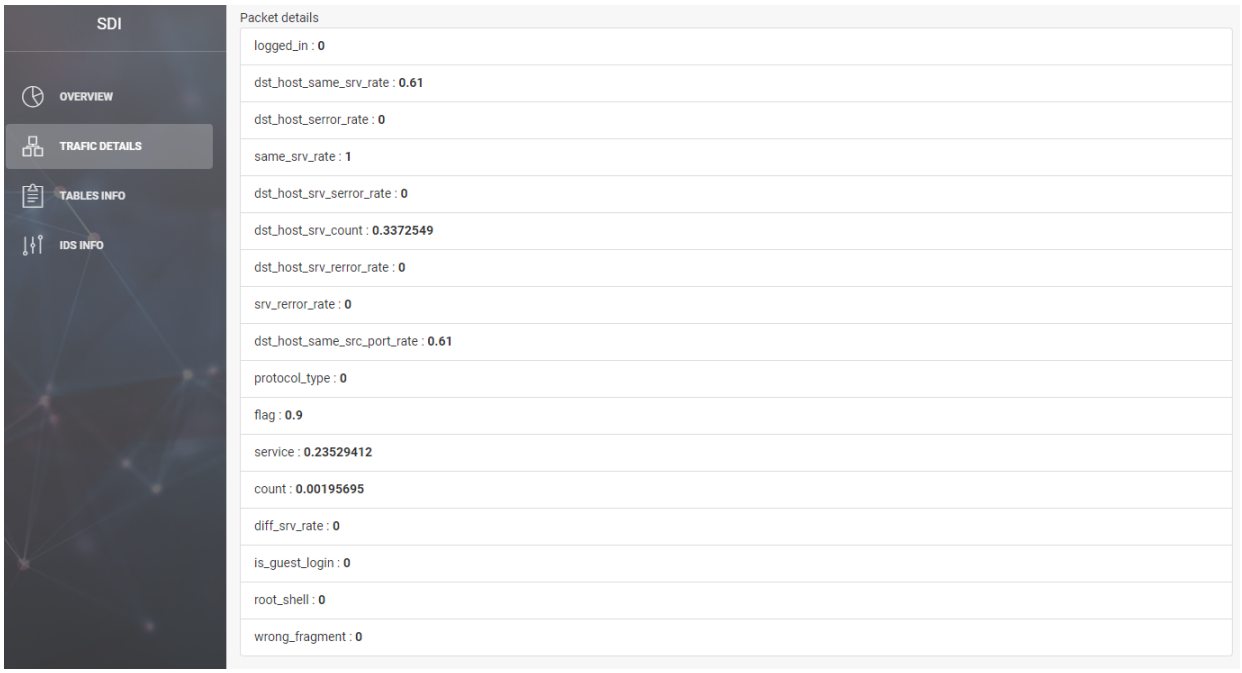
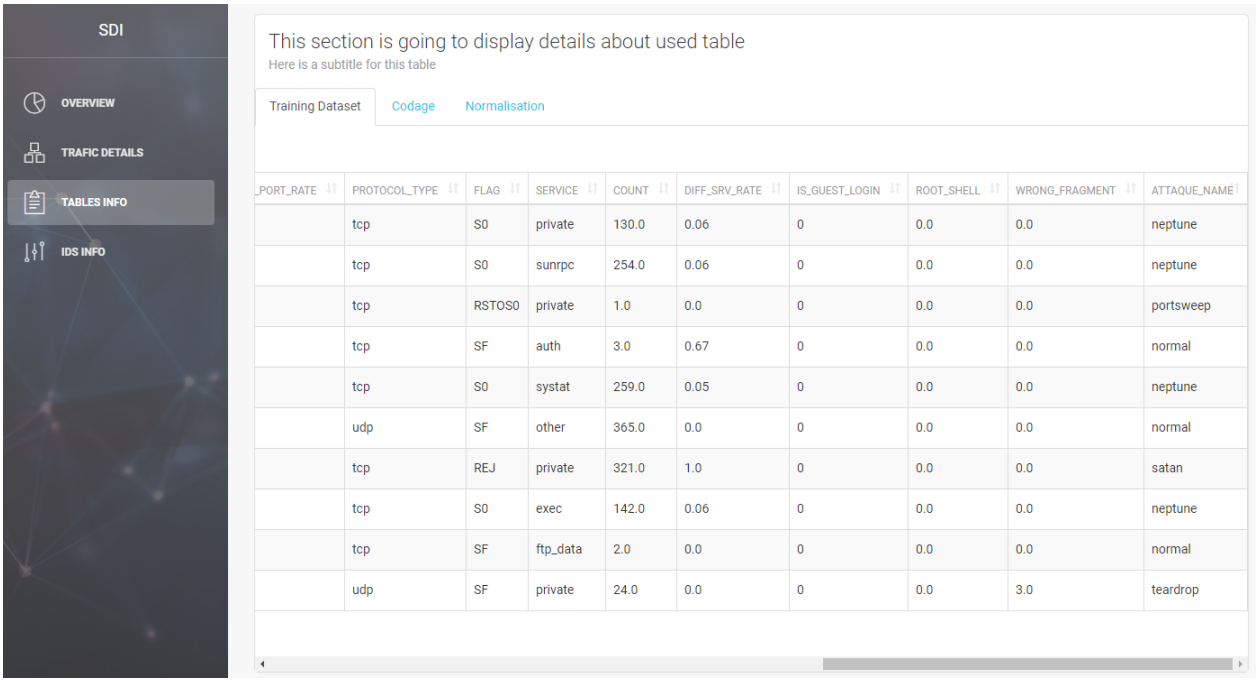


Figure 4.10 – Fenêtre pour les détails du trafic - partie 2

(c) **Troisième fenêtre**

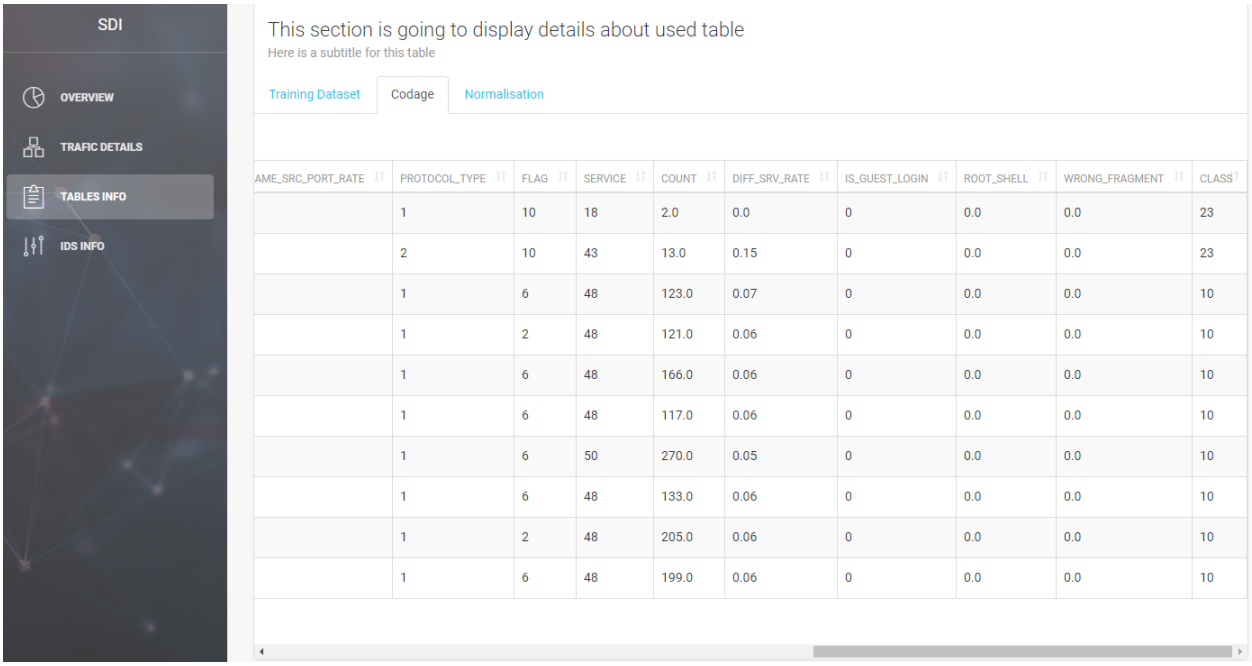
1/ Fenêtre responsable de charger la base de données d'apprentissage brute utilisée (sans prétraitement).



_PORT_RATE	PROTOCOL_TYPE	FLAG	SERVICE	COUNT	DIFF_SRV_RATE	IS_GUEST_LOGIN	ROOT_SHELL	WRONG_FRAGMENT	ATTAQUE_NAME
	tcp	S0	private	130.0	0.06	0	0.0	0.0	neptune
	tcp	S0	sunrpc	254.0	0.06	0	0.0	0.0	neptune
	tcp	RSTOS0	private	1.0	0.0	0	0.0	0.0	portsweep
	tcp	SF	auth	3.0	0.67	0	0.0	0.0	normal
	tcp	S0	systat	259.0	0.05	0	0.0	0.0	neptune
	udp	SF	other	365.0	0.0	0	0.0	0.0	normal
	tcp	REJ	private	321.0	1.0	0	0.0	0.0	satan
	tcp	S0	exec	142.0	0.06	0	0.0	0.0	neptune
	tcp	SF	ftp_data	2.0	0.0	0	0.0	0.0	normal
	udp	SF	private	24.0	0.0	0	0.0	3.0	teardrop

Figure 4.11 – Fenêtre de représentation des données d'apprentissage utilisées

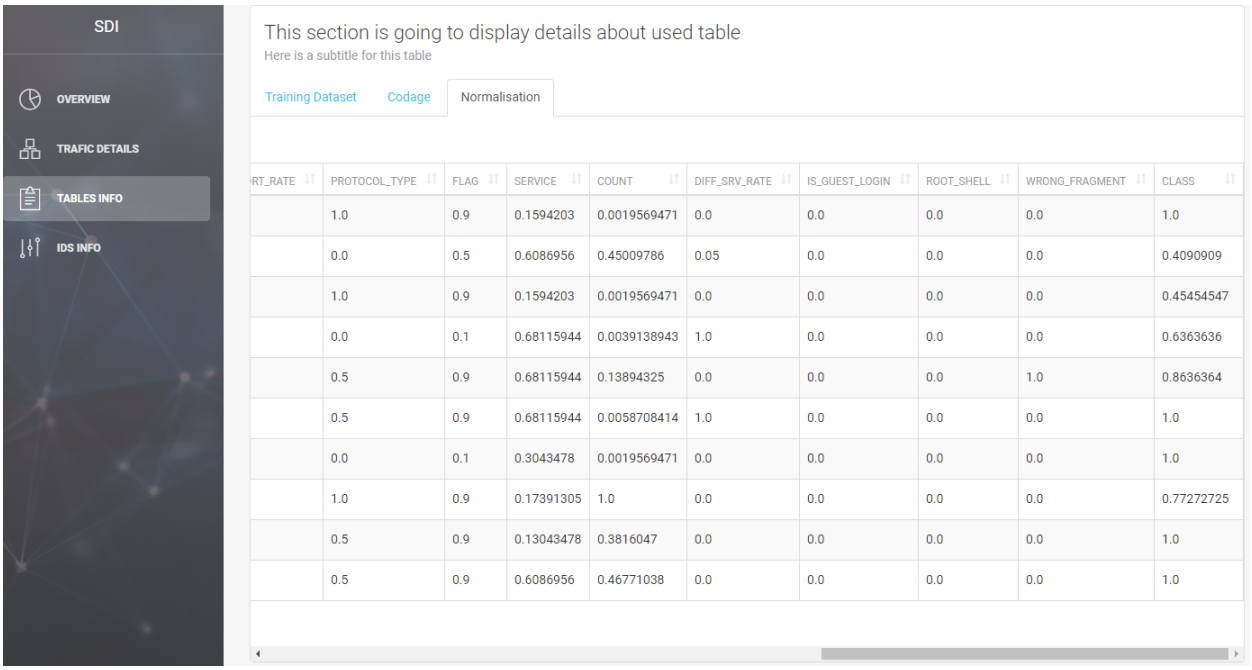
2/ Dans cette fenêtre on affiche la base de données d'apprentissage après le codage des données textuels.



AME_SRC_PORT_RATE	PROTOCOL_TYPE	FLAG	SERVICE	COUNT	DIFF_SRV_RATE	IS_GUEST_LOGIN	ROOT_SHELL	WRONG_FRAGMENT	CLASS
	1	10	18	2.0	0.0	0	0.0	0.0	23
	2	10	43	13.0	0.15	0	0.0	0.0	23
	1	6	48	123.0	0.07	0	0.0	0.0	10
	1	2	48	121.0	0.06	0	0.0	0.0	10
	1	6	48	166.0	0.06	0	0.0	0.0	10
	1	6	48	117.0	0.06	0	0.0	0.0	10
	1	6	50	270.0	0.05	0	0.0	0.0	10
	1	6	48	133.0	0.06	0	0.0	0.0	10
	1	2	48	205.0	0.06	0	0.0	0.0	10
	1	6	48	199.0	0.06	0	0.0	0.0	10

Figure 4.12 – Fenêtre de représentation des données d'apprentissage codées

3/ La figure 4.13 est une table représentative de la base de données d'apprentissage après la normalisation.



RT_RATE	PROTOCOL_TYPE	FLAG	SERVICE	COUNT	DIFF_SRV_RATE	IS_GUEST_LOGIN	ROOT_SHELL	WRONG_FRAGMENT	CLASS
	1.0	0.9	0.1594203	0.0019569471	0.0	0.0	0.0	0.0	1.0
	0.0	0.5	0.6086956	0.45009786	0.05	0.0	0.0	0.0	0.4090909
	1.0	0.9	0.1594203	0.0019569471	0.0	0.0	0.0	0.0	0.45454547
	0.0	0.1	0.68115944	0.0039138943	1.0	0.0	0.0	0.0	0.6363636
	0.5	0.9	0.68115944	0.13894325	0.0	0.0	0.0	1.0	0.8636364
	0.5	0.9	0.68115944	0.0058708414	1.0	0.0	0.0	0.0	1.0
	0.0	0.1	0.3043478	0.0019569471	0.0	0.0	0.0	0.0	1.0
	1.0	0.9	0.17391305	1.0	0.0	0.0	0.0	0.0	0.77272725
	0.5	0.9	0.13043478	0.3816047	0.0	0.0	0.0	0.0	1.0
	0.5	0.9	0.6086956	0.46771038	0.0	0.0	0.0	0.0	1.0

Figure 4.13 – Fenêtre de représentation des données d'apprentissage après la normalisation

B) **Machine Client**

Le terminal client est une machine indépendante qui se situe dans le même réseau local que le serveur, cette fenêtre nous permet de spécifier le numéro de port sur le quel la machine recevra les paquets.

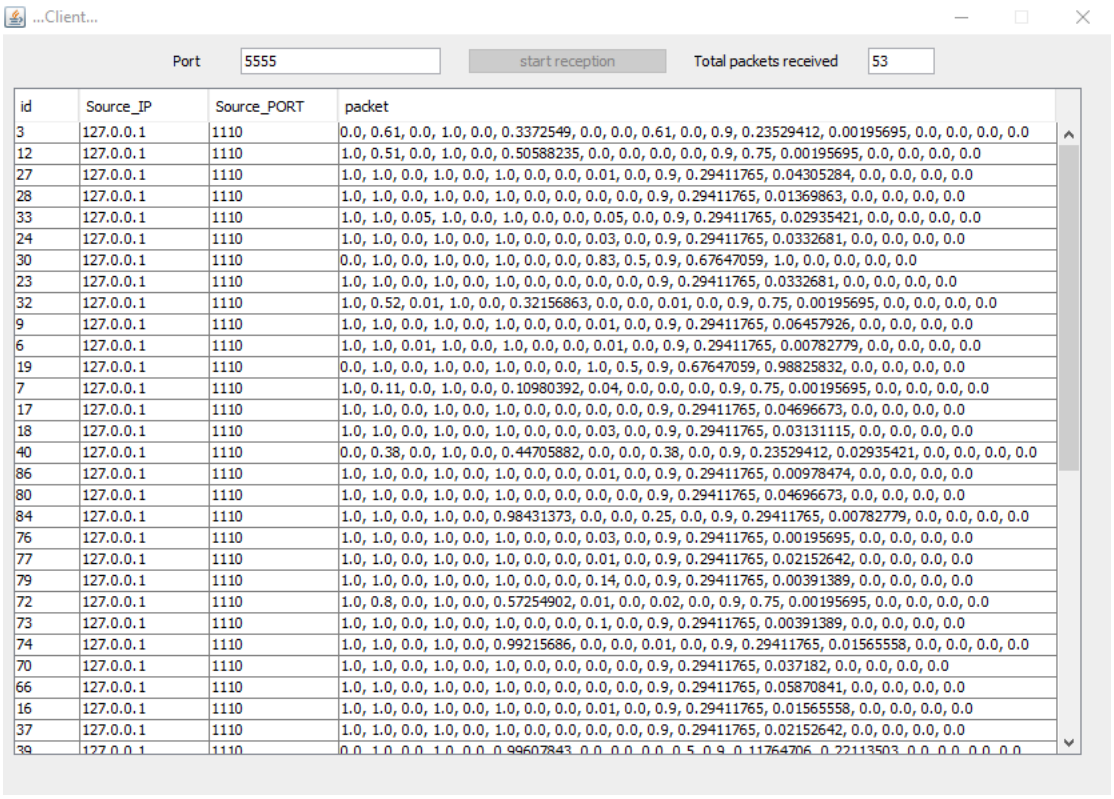
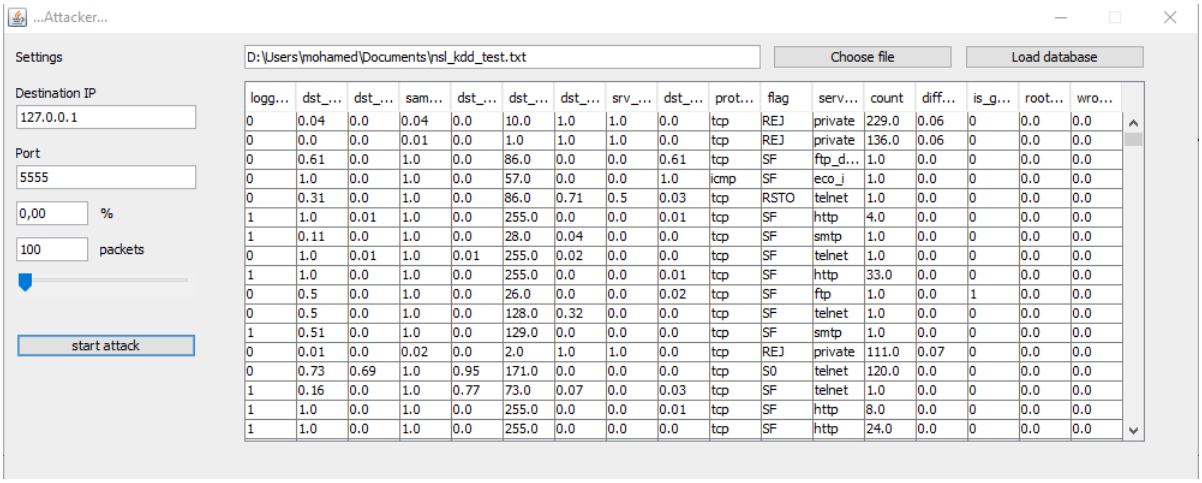


Figure 4.14 – Fenêtre concernant de la machine client

C) **Machine de l'attaquant**

La fenêtre de l'attaquant est composée de trois champs essentiels pour l'envoi des paquets, il suffit de faire rentrer l'adresse IP du client, son numéro de port, et le nombre de paquets à envoyer, de charger la base puis de cliquer sur le bouton commencer l'attaque. Notons toutefois que l'attaquant a accès à la BDDT sans prétraitement.



logg...	dst_...	dst_...	sam...	dst_...	dst_...	dst_...	srv_...	dst_...	prot...	flag	serv...	count	diff...	is_g...	root...	wro...
0	0.04	0.0	0.04	0.0	10.0	1.0	1.0	0.0	tcp	REJ	private	229.0	0.06	0	0.0	0.0
0	0.0	0.0	0.01	0.0	1.0	1.0	1.0	0.0	tcp	REJ	private	136.0	0.06	0	0.0	0.0
0	0.61	0.0	1.0	0.0	86.0	0.0	0.0	0.61	tcp	SF	ftp_d...	1.0	0.0	0	0.0	0.0
0	1.0	0.0	1.0	0.0	57.0	0.0	0.0	1.0	icmp	SF	eco_i	1.0	0.0	0	0.0	0.0
0	0.31	0.0	1.0	0.0	86.0	0.71	0.5	0.03	tcp	RSTO	telnet	1.0	0.0	0	0.0	0.0
1	1.0	0.01	1.0	0.0	255.0	0.0	0.0	0.01	tcp	SF	http	4.0	0.0	0	0.0	0.0
1	0.11	0.0	1.0	0.0	28.0	0.04	0.0	0.0	tcp	SF	smtp	1.0	0.0	0	0.0	0.0
0	1.0	0.01	1.0	0.01	255.0	0.02	0.0	0.0	tcp	SF	telnet	1.0	0.0	0	0.0	0.0
1	1.0	0.0	1.0	0.0	255.0	0.0	0.0	0.01	tcp	SF	http	33.0	0.0	0	0.0	0.0
0	0.5	0.0	1.0	0.0	26.0	0.0	0.0	0.02	tcp	SF	ftp	1.0	0.0	1	0.0	0.0
0	0.5	0.0	1.0	0.0	128.0	0.32	0.0	0.0	tcp	SF	telnet	1.0	0.0	0	0.0	0.0
1	0.51	0.0	1.0	0.0	129.0	0.0	0.0	0.0	tcp	SF	smtp	1.0	0.0	0	0.0	0.0
0	0.01	0.0	0.02	0.0	2.0	1.0	1.0	0.0	tcp	REJ	private	111.0	0.07	0	0.0	0.0
0	0.73	0.69	1.0	0.95	171.0	0.0	0.0	0.0	tcp	S0	telnet	120.0	0.0	0	0.0	0.0
1	0.16	0.0	1.0	0.77	73.0	0.07	0.0	0.03	tcp	SF	telnet	1.0	0.0	0	0.0	0.0
1	1.0	0.0	1.0	0.0	255.0	0.0	0.0	0.01	tcp	SF	http	8.0	0.0	0	0.0	0.0
1	1.0	0.0	1.0	0.0	255.0	0.0	0.0	0.0	tcp	SF	http	24.0	0.0	0	0.0	0.0

Figure 4.15 – Fenêtre de l'attaquant

4.9 Conclusion

Dans ce dernier chapitre, nous avons présenté les démarches prises afin de réaliser notre SDIR. Nous avons commencé par une présentation du principe général des approches proposées illustrée par un organigramme, nous avons par la suite détaillé ces approches en tenant en compte les défis à surmonter.

Nos approches ont été testées en utilisant la base de données NSL-KDD [Unb, 2019] qui a été réduite par le biais des méthodes de réduction de dimensionnalité que nous avons utilisées et cela a plusieurs niveaux. Le premier niveau concerne l'approche PPVC où nous avons réduit le nombre de vecteurs (paquets) de la base d'apprentissage. Le deuxième niveau consiste à la sélection des caractéristiques en appliquant une méthode non basée sur le modèle qui est le taux de variance ainsi qu'une autre méthode basée sur le modèle qui est l'élimination en arrière.

Cette réduction de dimensionnalité a permit d'améliorer le taux de détection qui est démontré via des mesures de performances tout en réduisant le temps de calcul. Ce dernier est un facteur très important vus qu'il s'agit d'application en temps réel.

L'utilisation des systèmes multi-agents a permit une modélisation et une simulation de notre SDIR. Grace à la notion de parallélisme et de distribution d'expertises, ce paradigme à contribuer a son tour à la réduction du temps de calcul. Ceci a permis d'atteindre un de nos objectifs principaux de notre étude. Les résultats obtenus sont satisfaisants et montre l'efficacité de nos approches proposées.

Conclusion générale

Avec l'augmentation du nombre d'attaques qu'est en train de vivre le monde, l'introduction de la mesure du temps pour les SDIs est devenue primordiale surtout avec l'importance et la vitesse du flux de données. Les SDIs doivent être capables alors d'effectuer une détection en temps réel et avec une très grande efficacité. C'est pour cela que plusieurs méthodes ont été mises en place. Par conséquent, nous avons commencé tout d'abord dans un premier chapitre par la présentation de la sécurité et des attaques informatiques, ces derniers qui peuvent nuire à une application, un système ou un réseau.

Les deux chapitres suivants décrivent les systèmes de détection d'intrusion et donnent une vision générale à propos de ces systèmes, en mettant l'accent sur les différentes approches et techniques utilisées afin de réaliser un SDI. L'état de l'art présenté dans le chapitre trois nous a permit cependant de nous situer quant aux approches adoptées.

Pour atteindre le plein potentiel de notre SDI, nous avons dû faire appel au paradigme multi-agent pour le but d'exploiter leurs atouts tels que la distribution et le parallélisme. Les SMAs ont apporté à leur tour une contribution dans la réduction du temps de calcul, un facteur important dans ce contexte. Ils nous ont notamment permis de réaliser nos simulations en mode réseau. Tout au long de notre travail nous nous sommes situés dans une approche de détection d'intrusion par signatures. Nos deux approches se basent sur l'algorithme K-PPV et l'algorithme PPVC avec une sélection des caractéristiques en utilisant le taux de variance et l'élimination en arrière.

Pour tester et évaluer nos approches proposées, nous avons choisi de travailler avec l'ensemble de données NSL-KDD, une base de données étiqueté et spécialement conçue pour l'évaluation des SDIs.

En conclusion, les résultats obtenus sont satisfaisants, la majorité des objectifs du thème ont était réalisées. Nos deux approches de détection peuvent prédire la nature du paquet, soit attaque ou normale avec un bon taux de réussite. Notre SDI est notamment capable de prédire le type précis de l'attaque. Afin de prouver l'efficacité de nos approches des comparaisons avec les méthodes IBK et J48 sont réalisées. Notant toutefois que l'approche SMA-PPVC a donné les meilleurs résultats en performances et en temps de calcul par rapport aux autres méthodes.

Cependant, il reste quelques perspectives et des perfectionnements à réaliser tels que l'intégration d'un agent renifleur capable de recevoir les paquets circulant dans le réseau même

s'ils ne sont pas directement envoyés au serveur. Le flux réseau est en réalité composé de paquets de type (tcp, udp, icmp), une transformation de la BDD NSL-KDD sous ce format permettra d'avoir un SDIR opérant dans un système réel plutôt que virtuel. Il est notamment possible d'explorer d'autres techniques dérivant de l'IA appliquant un apprentissage automatique tel que l'apprentissage profond (Deep Learning). Un SDI autoadaptatif appliquant un apprentissage par renforcement pourrait notamment régler le problème de mise à jour de la BDD.

Bibliographie

Bibliographie

[Aydin et al,2009] Aydin, M. Ali, Zaim, A. Halim, et Ceylan, K. Gökhan. "A hybrid intrusion detection system design for computer network security." Computers & Electrical Engineering, 2009, vol. 35, no 3, p. 517-526.

[Azad et Jha., 2013] Azad, C., et Jha, V. K. "Data mining in intrusion detection a comparative study of methods, types and data sets." International Journal of Information Technology and Computer Science (IJITCS), 2013, vol. 5, no 8, p. 75-90.

[Alazab et al., 2012] Alazab, A., Hobbs, M., Abawajy, J.,et Alazab, M."Using Feature Selection for Intrusion Detection System." In : Communications and Information Technologies (ISCIT), International Symposium, 2012. p. 296-301.

[Al-Yaseen et al., 2017] Al-Yaseen, W. L.,Othman, Z. A.,Nazri, M. Z. A. "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system."2017, vol. 67, p. 296-303.

[Aggarwal et Sharma, 2015] Aggarwal, P., et Sharma, S.K."An Empirical Comparison of Classifiers to Analyze Intrusion Detection." In : Fifth International Conference an Advanced Computing and Communication Technologies, 2015. p. 446-450.

[Alpaydin, 1997] Alpaydin, E. Voting over multiple condensed nearest neighbors. In : Lazy learning. Springer, Dordrecht, 1997. p. 115-132.

[Angiulli, 2005] Angiulli, F. Fast condensed nearest neighbor rule. In : Proceedings of the 22nd international conference on Machine learning, 2005. p. 25-32.

[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, Aug, 2013, vol. 76, no 1, p. 5-11.

[Biswas et al., 2015] Biswas, N. A., Tammi, W. M., Shah,F. M.,Chakraborty,S. "FP-ANK : An Improvised Intrusion Detection System with Hybridization of Neural Network and K-Means Clustering over Feature Selection by PCA." 18th International Conference on Computer and Information Technology (ICCIT), 2015. p. 317-322.

[Bellifemine et al., 2007] Bellifemine,F. L.,Caire, G., et Greenwood, D."Developing multi-agentsystems with JADE." John Wiley & Sons, 2007.

[Bace et Mell,2001] Bace, R., et Mell, P. "NIST special publication on intrusion detection systems." BOOZ-ALLEN AND HAMILTON INC MCLEAN VA, 2001.

[Bendaoud, 2014] Bendaoud, Y. "Prédiction Des Résistances Mécaniques Des Bétons à Base Des Ciments Composés En Utilisant Les Réseaux Neurones Artificiels." 2014.

[Beckenkamp, 2002] Beckenkamp, F. G. "A component architecture for artificial neural network systems."Diss, 2002.

[Balajiet Srinivasan, 2010] Balaji, P. G., et Srinivasan, D. "An introduction to multi-agent systems." In :Innovations in multi-agent systems and applications-1. Springer, Berlin, Heidelberg, 2010. p. 1-27.

[Benmammar, 2009] Benmammar, B. "Systèmes Multi-Agents." Engineering school, 2009.

[ByrskietCarvalho, 2008] Byrski, A.,et Carvalho, M. "Agent-based immunological intrusion detection system for mobile ad-hoc networks." In : International Conference on Computational Science. Springer, Berlin, Heidelberg, 2008. p. 584-593.

[Breiman et al., 1984] Breiman L., Friedman J., Olshen R., et Stone C. "Classification and Regression Trees." Wadsworth Int. Group, 1984.

[Boucherba et Ziane, 2015] Boucherba, K., et Ziane, S. "Mise en place d'un pare-feu d'entreprise open source PfSense." 2015. Thèse de doctorat. Université A/Mira de Bejaia.

[Baudoin et Karle, 2004] Baudoin, N., et Karle, M."NT Réseaux –IDS et IPS." support de cours, Enseignant Etienne Duris, 2004.

[Burgermeister et Krier, 2006] urgermeister, D.,et Krier, J. "Les systèmes de détection d'intrusions." 2006.

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

[Caswell et Beale,2004] Caswell, B., et Beale, J. "Snort 2.1 intrusion detection." Elsevier, 2004.

[Canady, 1998] Cannady, J. "Artificial neural networks for misuse detection." In : National information systems security conference. Vol. 26, 1998. p. 443-456

[Castelli, 2003] Castelli, V. "Nearest NeighborClassifiers ."Columbia Engineering University, 2003

[Cortes etVapnik, 1995] Cortes, C., et Vapnik, V. "Support-vector networks." Machine learning 20.3, 1995, , vol. 20, no 3, p. 273-297.

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

[Cohen, 1987] Cohen, F. "Computer viruses : theory and experiments." Computers & security 6.1, 1987, vol. 6, no 1, p. 22-35.

[Coveret et Hart, 1967] Cover, T. M., & Hart, P. E. Nearest neighbour pattern recognition. IEEE Trans. On information theory, 13(1), 1967, vol. 13, no 1, p. 21-27.

[Chouaib, 2011] Chouaib, H. Sélection de caractéristiques : méthodes et applications. Paris Descartes University : Paris, France, 2011.

[Choudhury et al., 2019] Choudhury, S., Mishra, R., Mishra, R. G., et Kumar, A. (Eds.). Intelligent Communication, Control and Devices : Proceedings of ICICCD 2018 (Vol. 989). Springer Nature, 2019.

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

[Ciancarini et Wooldridge, 2000] Ciancarini, P., et Wooldridge, M. Agent-oriented software engineering (workshop session). In Proceedings of the 22nd international conference on Software engineering, 2000. p. 816-817.

[David etVassilvitskii, 2006] David, A.,et Vassilvitskii, S."k-means++ : The advantages of careful seeding."Stanford, 2006.

[Djeriri, 2017] Djeriri, Y."Les Réseaux de Neurones Articiels." mémoire fin d'études, Sidi Bel Abbès, Algerie, 2017.

[Devéze et Fouquin, 2005] Devéze, B., et Fouquin, M. "DATAMINING C4. 5–DBSCAN." PROMOTION, SCIA Ecole pour l'informatique et techniques avancees, 2005.

[Derras, 2011] Derras, B. "Contribution des données accélérométriques de KiKNet à la prédiction du mouvement sismique par l'approche neuronale avec la prise en compte des effets de site."Diss, 2011. Thèse de doctorat.

[ElekaretWaghmare, 2015] Elekar,K.S., etWaghmare,M.M. "Comparison of Tree base Data Mining Algorithms for Network Intrusion Detection." International Journal of Engineering, Education and Technology, 2015, vol. 2, no 10, p. 3253-3257.

[Eddy et al., 2015] Eddy, Y. F., Gooi, H. B.,et Chen, S. X. "Multi-agent system fordistributed management of microgrids." IEEE Transactions on powersystems, 2015.

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

[Eibe et al., 2016] Eibe F., M. A. Hall, and I. H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining : Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

[Ferber, 1995] Ferber, J."Les systèmes multi-agents : vers une intelligence collective." Inter-Editions, Paris, 1995.

[Gareth et al., 2015] Gareth, J., Witten, D., Hastie, T., Tibshirani, R. "An introduction to statistical learning." New York : springer, 2015.

[Gautam et al., 2016] Gautam, S. K., et Om, H. "Computational neural network regression model for host based intrusion detection system." Perspectives in Science 8, 2016, vol. 8, p. 93-95.

[Gurney, 1997] Gurney, K. "An introduction to neural networks." CRC press, 1997.

[Gou et al., 2012] Gou, J., Du, L., Zhang, Y., etXiong, T. "A new distance-weighted k-nearest neighbor classifier." J. Inf. Comput. Sci, 2012, vol. 9, no 6, p. 1429-1436.

[Ghernaouti-Hélie, 2000] Ghernaouti-Hélie, S. "Sécurité Internet : stratégies et technologies." Dunod, 2000.

[Goldberg et Holland, 1988] Goldberg, D. E.,et Holland, J. H. "Genetic algorithms and machinelearning." 1988.

[Girard, 2007] Girard, A. "Exploration d'un algorithme génétique et d'un arbre de décision à des fins de catégorisation. "2007. Thèse de doctorat. Université du Québec à Trois-Rivières.

[Gutknecht et Ferber, 2001] Gutknecht, O., et Ferber, J. "The madkit agent platform architecture." In :Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems, 2001. p 48-55.

[Goutte et al., 2005] Goutte, C., etGaussier, E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In : European Conference on Information Retrieval. Springer, Berlin, Heidelberg, 2005. p. 345-359.

[Gharaee et Hosseinvand, 2016] Gharaee, H., et Hosseinvand, H. A new feature selection IDS based on genetic algorithm and SVM. In : 2016 8th International Symposium on Telecommunications (IST). IEEE, 2016. p. 139-144

[Hasan et Boris, 2006] Hasan, M., & Boris, F."SVM machine à vecteurs de support ou séparateur à vaste marge." BD Web, ISTY3, Versailles St Quentin, France, Janvier 2006.

[Hastie et Tibshirani, 1996] Hastie, T., et Tibshirani, R. "Discriminant adaptive nearest-neighbor classification and regression." In : Advances in Neural Information Processing-Systems, 1996. p. 409-415

[Hechenbichler et Schliep, 2004] Hechenbichler, K., et Schliep, K. "Weighted k-nearest-neighbor techniques and ordinal classification." 2004.

[Hart ; 1968] Hart, P. The condensed nearest neighbor rule (Corresp.). IEEE transactions on information theory, 1968, vol. 14, no 3, p. 515-516.

[Ishwaran et Rao, 2009] Ishwaran, H., et Rao, J. S."DecisionTree : Introduction" 2009.

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

[Javaid et al., 2016] Javaid, A., Niyaz, Q., Sun, W., etAlam, M. "A deep learning approach for network intrusion detection system." In : Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), 2016. p. 21-26.

[Kateb et Guerram, 2011] Kateb, N., et Guerram, T. "Une Approche multi agents pour les datdmining." 2011.

[Kaur et al., 2014] Kaur, N. K., Kaur, U., et Singh, D. "K-Medoid clustering algorithm-a review." International Journal of Computer Application and Technology (IJCAT) 1.1, 2014.

[Kogan, 2007] Kogan, J."Introduction to Clustering Large and High-Dimensional Data." Cambridge University Press, Cambridge, 2007.

[Liu, et al., 2007] Liu, S., Li, T., Wang, D., Hu, X., & Xu, C. "Multi-Agent Network Intrusion Active Defense Model Based on Immune Theory." Wuhan University Journal of Natural Sciences, 2007, vol. 12, no 1, p. 167-171.

[Lv et al., 2020] Lv, L., Wang, W., Zhang, Z., & Liu, X. "A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine." Knowledge-Based Systems, 2020.

[Lepuschitz, 2018] Lepuschitz, W. "Self-reconfigurable manufacturing control based on ontology-driven automation agents." Technische Universität Wien, 2018. Thèse de doctorat.

[Liao etVemuri, 2002] Liao, Y., etVemuri, V. R. "Use of k-nearest neighbor classifier for intrusion detection." Computers & security, 2002, vol. 21, no 5, p. 439-448

[Müller et al,2003] Müller, K., alias'Socma, K. M., et Tarbouriech, G. "IDS-Systèmes de Détection d'Intrusion, Partie I." LinuxFocus article number 292, 2003.

[Ma et al., 2015] Ma, L., Min, H., Wang, S., Liu, Y., et Liao, S. "An overview of research chin distributed attitude coordination control," IEEE/CAA Journal of Auto-maticaSinica, 2015, vol. 2, no 2, p. 121-133.

[Maess, 1995] Maess, P. "Artificial life meets entertainment : Life like autonomous agents" Communications of the ACM, 1995, vol. 38, no 11, p. 108-114.

[Major et Reichgelt, 1990] Major, N., et Reichgelt, H. "ALTO : An automated laddering tool." IOS Press, 1990, p. 222-236.

[Malhotra et al., 2014] Malhotra, V. K., Kaur, H., etAlam, M. A. "An Analysis of Fuzzy Clustering Methods." International Journal of Computer Applications, 2014, vol. 94, no 19, p. 9-12.

[MorissetteetChartier, 2013] Morissette, L., etChartier, S."The k-means clustering technique : General considerations and implementation in Mathematica." Tutorials in Quantitative Methods for Psychology, 2013, vol. 9, no 1, p. 15-24.

[Mokhtari et Moulkhaloua, 2018] Mokhtari, M, et Moulkhaloua, A. "Système DE Détection D'intrusions Informatiques par Système Multi Agents." mémoire fin d'études, Centre universitaire BELHADJ Bouchaib Ain Temouchent, 2018.

[Morin, 2014] Morin, V. "Étude comparative d'algorithmes de data mining dans le contexte du jeu vidéo." 2014. Thèse de doctorat. Université du Québec à Chicoutimi.

[Mohd et al., 2012] Mohd, W. M. W., Beg, A. H., Herawan, T., & Rabbi, K. F."MaxD K-Means : A Clustering Algorithm for Auto-generation of Centroids and Distance of Data Points in Clusters." 2012.

[Mariette et Rahul, 2005] Mariette, A., et Rahul, K. "Support Vector Machines for Classification." In : Efficient Learning Machines. Apress, Berkeley, CA, 2015. p. 39-66.

[Méziane, 2007] Méziane, I."Stratégie de rendez-vous dans les systèmes multi-agents." mémoire fin d'études, Université AboubekrBelkaid - Ingénieur en informatique, 2007.

[Mukhopadhyay et al., 2008] Mukhopadhyay, A., Roy, A., Das, S., Das, S., et Abraham, A. Population-variance and explorative power of harmony search : an analysis. In : 2008 Third International Conference on Digital Information Management. IEEE, 2008. p. 775-781.

[NikhithaetJabbar, 2019] Nikhitha, M.,etJabbar, M.A. "K NearestNeighbor Based Model for Intrusion Detection System." International Journal of Recent Technology and Engineering (IJRTE), Juillet 2019, vol. 8, no. 2.

[Nwana, 1996] Nwana, H. "Software agents : An overview," Knowledge and Engineering Review, 1996, vol. 11, no 3, p. 205-244.

[Nadali et al., 2011] Nadali, A., Kakhky, E. N., et Nosratabadi, H. E. Evaluating the success level of data mining projects based on CRISP-DM methodology by a Fuzzy expert system. In : 2011 3rd International Conference on Electronics Computer Technology. IEEE, 2011. p. 161-165.

[Okamoto et Ishida, 2007] Okamoto, T.,et Ishida,Y."Framework of an immunity-based anomaly detection system for user behavior." In : Proceedings of the International Conference on Knowledge-Based Intelligent Information and Engineering Systems, KES, 2007.

[Ogungbemi, 2019] Ogungbemi, H. S."Etude pour la sécurisation d'un réseau par la mise en place d'un pare-feu open source : cas de C.A.F.E. informatique & télécommunications." 2019.

[O'neill, 2006] O'neill, B. Elementary differential geometry. Elsevier, 2006.

[Pillou et Bay, 2016] Pillou, J. F., et Bay, J. P. "Tout sur la sécurité informatique-4e édition." Dunod, 2016.

[Poinsot, 2019] Poinsot, L. "Introduction à la sécurité informatique." support de cours, Université Paris 13.3,2019.

[Pillou et Bay, 2013] Pillou, J. F., et Bay, J. P."Sécurité informatique 3ème édition." Dunod, Paris 2013.

[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.

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

[Russel et Norvig, 1995] Russel, S., et Norvig, P. "Artificial intelligence – A modern approach."Prentice Hall, 1995.

[Reddy, 2013] Reddy, E. K. "Neural networks for intrusion detection and its applications." In : Proceedings of the World Congress on Engineering. Vol. 2. No. 5, 2013. p. 3-5.

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

[Rakotomalala, 2005] Rakotomalala, R. "Arbres de décision." Revue Modulad, 2005, vol. 33, p. 163-187.

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

[Sweeney et al,2003] Sweeney, M., Baumrucker, C. T., Burton, J. D., et Dubrawsky, I. "Cisco security professional's guide to secure intrusion detection systems." SyngressPublishing, 2003.

[Steinhaus, 1956] Steinhaus, H. "Sur la division des corpmateriels en parties." 1956, vol. 1, no 804, p. 801.

[Subba et al., 2015] Subba, B., Biswas, S., etKarmakar, S. "Intrusion detection systems using linear discriminant analysis and logistic regression." 2015 Annual IEEE India Conference (INDICON). IEEE, 2015.

[Shenfield et al., 2018] Shenfield, A., Day, D., et Ayesh, A. "Intelligent intrusion detection systems using artificial neural networks." ICT Express, 42, 2018, vol. 4, no 2, p. 95-99.

[Staub et al., 2015] Staub, S., Karaman, E., Kaya, S., Karapınar, H., etGüven, E. "Artificial neural network and agility." Procedia-Social and Behavioral Sciences, 2015, vol. 195, p. 1477-1485.

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

[Sutter et Kalivas., 1993] Sutter, J. M., et Kalivas, J. H. "Comparison of forward selection, backward elimination, and generalized simulated annealing for variable selection." Microchemical journal, 1933, vol. 47, no 1-2, p. 60-66.

[Salvatore et al., 2000] Stolfo, J., Fan, W., Lee, W., Prodromidis, A., et Chan, P. K. "Cost-based modeling and evaluation for data mining with application to fraud and intrusion detection." Results from the JAM Project by Salvatore, 2000. p. 1-15.

[Tay et al., 2014] Tay, B., Hyun, J. K., et Oh, S."A machine learning approach for specification of spinal cord injuries using fractional anisotropy values obtained from diffusion tensor images." Computational and mathematicalmethods in medicine, 2014.

[Torre, 1999] Torre, F. "GloBo : un algorithme stochastique pour l'apprentissage supervisé et non-supervisé. " In : M. Sebag, editeur, Actes de la Première Conférence d'Apprentissage, 1999.

[Touzet, 1992] Touzet, C. "les réseaux de neurones artificiels, introduction au connexionnisme."1992.

[VarunaetNatesan, 2015] Varuna,S., et Natesan, P."An Integration of K-Means Clustering and Naïve Bayes Classifier for Intrusion Detection." In : 2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN), 2015. p. 1-5.

[Wooldridge, 2009] Wooldridge, M."An introduction to multiagent systems."John Wiley & Sons, 2009.

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

[Zimmermann, 1998] Zimmermann, P. "An introduction to cryptography." Network Associates Inc NAI, 1998.

[Zaidi, 2011] Zaidi, A."Recherche et détection des patterns d'attaques dans les réseaux IP à haut débits."Janvier 2011. Thèse de doctorat. Université d'Évry-Val-d'Essonne.

Webographie

[Advisio, 2019] Advisio. "Antivirus : A Quoi Servent-Ils ?" advisio-com.fr, 2019, consulté le 17 Mars 2020.

[Arnaud, 2002] Arnaud, J. "Les trois types d'attaques réseaux." Securiteinfo, securiteinfo.com 2002, consulté le 7 Février 2020.

[Cisco, 2020a] Cisco. "What Is Cybersecurity ?" Cisco.com, consulté le 7 Février 2020.

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

[CERT, 2010] CERT."Limites Et Défis Des Antivirus." Computer Emergency Response Team - Industrie Services Et Tertiaire, cert-ist.com, 2010, consulté le 5 Juin 2020.

[Data Analytics Post, 2018] Data Analytics Post "k-Nearest Neighbours." dataanalytics-post.com,consulté le13 Mai 2020.

[DEV Community, 2019] Lorraine, L."Classification and Regression Analysis with Decision Trees." DEV Community, dev.to, 15 Mai 2019,consulté le 19 Mai 2020.

[Djeffal, 2020] Djeffal, A."Réseaux De Neurones." Abdelhamid-Djeffal.net, abdelhamid-djeffal.net,consulté le 19 Mai 2020.

[Edureka, 2020] Edureka. "Understanding K-Means Clustering with Examples."edureka.co,consulté le 18 Mai 2020.

[GAMA, 2018] GAMA. "GAMA-Platform." gama-platform.github.io, 2018,consulté le 12 Mai 2020.

[GeeksforGeeks, 2020] GeeksforGeeks."Difference between Firewall and Antivirus."geeksforgeeks.org, 2020, consulté le 30 Avril 2020.

[Hacker Combat, 2019] Hacker Combat. "Virtual Private Network (VPN) : What VPN Is And How It Works"hackercombat.com, 2019,consulté le4 Juin 2020.

[KDD, 2007] UCI Knowledge Discovery in Databases Archive "KDD Cup 1999 Data." kdd.ics.uci.edu, consulté Juin 2020.

[Meade, 2019] Meade, T."What Is the Difference between Firewall and Antivirus ?" Colocation America, Colocation American Staff, colocationamerica.com, 2019, consulté le 30 Avril 2020.

[NetBeans, 2019] NetBeans. "NetBeans." netbeans.org, consulté Juin 2020.

[Petters, 2020] Petters, J. "What's The Difference Between a Proxy and a VPN ?"Inside Out Security, www.varonis.com, 2020, consulté le28 Avril 2020.

[PostgreSQL, 1996] PostgreSQL. "PostgreSQL." postgresql.org, consulté Juin 2020.

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

[Simmons, 2019] Simmons, J.H. "Disadvantages of VPN That You Should Know Before Using It."VPNCrew, vpncrew.com, 2019,consulté le 4 Juin 2020.

[Suini, 2015] Suini, J."Le principe du VPN." Supinfo.com,2015, consulté le 10 Février 2020.

[Souvik, 2020] Souvik. "Proxy Server : The Advantages Disadvantages of a Proxy Server."RS Web Solutions, rswebsols.com,2020 consulté le 9 Février 2020.

[Securiteinfo, 2006] Securiteinfo"Le grand livre de la sécurité informatique." securiteinfo.com, Editions du 6 novembre 2006, consulté le 7 Février 2020.

[Serengil, 2018] Serengil, S. "A Step By Step Regression Tree Example." Sefik Ilkin Serengil, sefiks.com, 2018,consulté le 30 Avril 2020.

[Towards Data Science,2019] Pupale, R. "Support Vector Machines(SVM) - An Overview." Medium, Towards Data Science, towardsdatascience.com,11 Feb. 2019,consulté le 10 Avril 2020.

[Unb, 2019] Canadian Institute for Cybersecurity."NSL-KDD dataset." unb.ca, consulté Juin 2020.

[Valgasu, 2017] Valgasu. "IP Spoofing." Sécurité Informatique : Audits Et Conseils En Cybersécurité, securiteinfo.com, 20 Mai 2017, consulté le 8 Février 2020.

[WikiStat, 2016] WikiStat "Machines à vecteurs supports." wikistat.fr, 2016,consulté le15 Avril 2020.

Bibliographie

[Aydin et al,2009] Aydin, M. Ali, Zaim, A. Halim, et Ceylan, K. Gökhan. "A hybrid intrusion detection system design for computer network security." Computers & Electrical Engineering, vol. 35, no 3, p. 517-526, 2009.

[Azad et Jha., 2013] Azad, C., et Jha, V. K. "Data mining in intrusion detection a comparative study of methods, types and data sets." International Journal of Information Technology and Computer Science (IJITCS), vol. 5, no 8, p. 75-90, 2013.

[Alazab et al., 2012] Alazab, A., Hobbs, M., Abawajy, J.,et Alazab, M."Using Feature Selection for Intrusion Detection System." In : Communications and Information Technologies (ISCIT), International Symposium. p. 296-301, 2012.

[Al-Yaseen et al., 2017] Al-Yaseen, W. L.,Othman, Z. A.,Nazri, M. Z. A. "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system.", vol. 67, p. 296-303, 2017.

[Aggarwal et Sharma, 2015] Aggarwal, P., et Sharma, S.K."An Empirical Comparison of Classifiers to Analyze Intrusion Detection." In : Fifth International Conference an Advanced Computing and Communication Technologies. p. 446-450, 2015.

[Alpaydin, 1997] Alpaydin, E. Voting over multiple condensed nearest neighbors. In : Lazy learning. Springer, Dordrecht. p. 115-132, 1997.

[Angiulli, 2005] Angiulli, F. Fast condensed nearest neighbor rule. In : Proceedings of the 22nd international conference on Machine learning. p. 25-32, 2005.

[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, Aug, vol. 76, no 1, p. 5-11, 2013.

[Biswas et al., 2015] Biswas, N. A., Tammi, W. M., Shah,F. M.,Chakraborty,S. "FP-ANK : An Improvised Intrusion Detection System with Hybridization of Neural Network and K-Means Clustering over Feature Selection by PCA." 18th International Conference on Computer and Information Technology (ICCIT). p. 317-322, 2015.

[Bellifemine et al., 2007] Bellifemine,F. L.,Caire, G., et Greenwood, D."Developing multi-agentsystems with JADE." John Wiley & Sons, 2007.

[Bace et Mell,2001] Bace, R., et Mell, P. "NIST special publication on intrusion detection systems." BOOZ-ALLEN AND HAMILTON INC MCLEAN VA, 2001.

[Bendaoud, 2014] Bendaoud, Y. "Prédiction Des Résistances Mécaniques Des Bétons à Base Des Ciments Composés En Utilisant Les Réseaux Neurones Artificiels." 2014.

[Beckenkamp, 2002] Beckenkamp, F. G. "A component architecture for artificial neural network systems."Diss, 2002.

[Balajiet Srinivasan, 2010] Balaji, P. G., et Srinivasan, D. "An introduction to multi-agent systems." In :Innovations in multi-agent systems and applications-1. Springer, Berlin, Heidelberg. p. 1-27, 2010.

[Benmammar, 2009] Benmammar, B. "Systèmes Multi-Agents." Engineering school, 2009.

[ByrskietCarvalho, 2008] Byrski, A.,et Carvalho, M. "Agent-based immunological intrusion detection system for mobile ad-hoc networks." In : International Conference on Computational Science. Springer, Berlin, Heidelberg. p. 584-593, 2008.

[Breiman et al., 1984] Breiman L., Friedman J., Olshen R., et Stone C. "Classification and Regression Trees." Wadsworth Int. Group, 1984.

[Boucherba et Ziane, 2015] Boucherba, K., et Ziane, S. "Mise en place d'un pare-feu d'entreprise open source PfSense." Thèse de doctorat. Université A/Mira de Bejaia, 2015.

[Baudoin et Karle, 2004] Baudoin, N., et Karle, M."NT Réseaux –IDS et IPS." support de cours, Enseignant Etienne Duris, 2004.

[Burgermeister et Krier, 2006] urgermeister, D.,et Krier, J. "Les systèmes de détection d'intrusions." 2006.

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

[Caswell et Beale,2004] Caswell, B., et Beale, J. "Snort 2.1 intrusion detection." Elsevier, 2004.

[Canady, 1998] Cannady, J. "Artificial neural networks for misuse detection." In : National information systems security conference. Vol. 26. p. 443-456, 1998.

[Cortes etVapnik, 1995] Cortes, C., et Vapnik, V. "Support-vector networks." Machine learning 20.3, vol. 20, no 3, p. 273-297, 1995.

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

[Cohen, 1987] Cohen, F. "Computer viruses : theory and experiments." Computers & security 6.1, vol. 6, no 1, p. 22-35, 1987.

[Coveret et Hart, 1967] Cover, T. M., & Hart, P. E. Nearest neighbour pattern recognition. IEEE Trans. On information theory, 13(1), vol. 13, no 1, p. 21-27, 1967.

[Chouaib, 2011] Chouaib, H. Sélection de caractéristiques : méthodes et applications. Paris Descartes University : Thèse de doctorat, Paris, France, 2011.

[Choudhury et al., 2019] Choudhury, S., Mishra, R., Mishra, R. G., et Kumar, A. (Eds.). Intelligent Communication, Control and Devices : Proceedings of ICICCD 2018 (Vol. 989). Springer Nature, 2019.

[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.

[Ciancarini et Wooldridge, 2000] Ciancarini, P., et Wooldridge, M. Agent-oriented software engineering (workshop session). In Proceedings of the 22nd international conference on Software engineering. p. 816-817, 2000.

[David etVassilvitskii, 2006] David, A.,et Vassilvitskii, S."k-means++ : The advantages of careful seeding."Stanford, 2006.

[Djeriri, 2017] Djeriri, Y."Les Réseaux de Neurones Articiels." mémoire fin d'études, Sidi Bel Abbès, Algerie, 2017.

[Devéze et Fouquin, 2005] Devéze, B., et Fouquin, M. "DATAMINING C4. 5–DBSCAN." PROMOTION, SCIA Ecole pour l'informatique et techniques avancees, 2005.

[Derras, 2011] Derras, B. "Contribution des données accélérométriques de KiKNet à la prédiction du mouvement sismique par l'approche neuronale avec la prise en compte des effets de site."Diss. Thèse de doctorat, 2011.

[ElekaretWaghmare, 2015] Elekar,K.S., etWaghmare,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. 3253-3257, 2015.

[Eddy et al., 2015] Eddy, Y. F., Gooi, H. B.,et Chen, S. X. "Multi-agent system fordistributed management of microgrids." IEEE Transactions on powersystems, 2015.

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

[Eibe et al., 2016] Eibe F., M. A. Hall, and I. H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining : Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

[Ferber, 1995] Ferber, J."Les systèmes multi-agents : vers une intelligence collective." Inter-Editions, Paris, 1995.

[Gareth et al., 2015] Gareth, J., Witten, D., Hastie, T., Tibshirani, R. "An introduction to statistical learning." New York : springer, 2015.

[Gautam et al., 2016] Gautam, S. K., et Om, H. "Computational neural network regression model for host based intrusion detection system." Perspectives in Science 8, vol. 8, p. 93-95, 2016.

[Gurney, 1997] Gurney, K. "An introduction to neural networks." CRC press, 1997.

[Gou et al., 2012] Gou, J., Du, L., Zhang, Y., etXiong, T. "A new distance-weighted k-nearest neighbor classifier." J. Inf. Comput. Sci, vol. 9, no 6, p. 1429-1436, 2012.

[Ghernaouti-Hélie, 2000] Ghernaouti-Hélie, S. "Sécurité Internet : stratégies et technologies." Dunod, 2000.

[Goldberg et Holland, 1988] Goldberg, D. E.,et Holland, J. H. "Genetic algorithms and machinelearning." 1988.

[Girard, 2007] Girard, A. "Exploration d'un algorithme génétique et d'un arbre de décision à des fins de catégorisation. " Thèse de doctorat. Université du Québec à Trois-Rivières, 2007.

[Gutknecht et Ferber, 2001] Gutknecht, O., et Ferber, J. "The madkit agent platform architecture." In :Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems. p 48-55, 2001.

[Goutte et al., 2005] Goutte, C., et Gaussier, E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In : European Conference on Information Retrieval. Springer, Berlin, Heidelberg. p. 345-359, 2005.

[Gharaee et Hosseinvand, 2016] Gharaee, H., et Hosseinvand, H. A new feature selection IDS based on genetic algorithm and SVM. In : 2016 8th International Symposium on Telecommunications (IST). IEEE p. 139-144, 2016.

[Hasan et Boris, 2006] Hasan, M., & Boris, F."SVM machine à vecteurs de support ou séparateur à vaste marge." BD Web, ISTY3, Versailles St Quentin, France, Janvier 2006.

[Hastie et Tibshirani, 1996] Hastie, T., et Tibshirani, R. "Discriminant adaptive nearest-neighbor classification and regression." In : Advances in Neural Information Processing-Systems. p. 409-415, 1996.

[Hechenbichler et Schliep, 2004] Hechenbichler, K., et Schliep, K. "Weighted k-nearest-neighbor techniques and ordinal classification." 2004.

[Hart ; 1968] Hart, P. The condensed nearest neighbor rule (Corresp.). IEEE transactions on information theory, vol. 14, no 3, p. 515-516, 1968.

[Ishwaran et Rao, 2009] Ishwaran, H., et Rao, J. S."DecisionTree : Introduction" 2009.

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

[Javaid et al., 2016] Javaid, A., Niyaz, Q., Sun, W., etAlam, M. "A deep learning approach for network intrusion detection system." In : Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS). p. 21-26, 2016.

[Kateb et Guerram, 2011] Kateb, N., et Guerram, T. "Une Approche multi agents pour les datdmining." 2011.

[Kaur et al., 2014] Kaur, N. K., Kaur, U., et Singh, D. "K-Medoid clustering algorithm-a review." International Journal of Computer Application and Technology (IJCAT) 1.1, 2014.

[Kogan, 2007] Kogan, J."Introduction to Clustering Large and High-Dimensional Data." Cambridge University Press, Cambridge, 2007.

[Liu, et al., 2007] Liu, S., Li, T., Wang, D., Hu, X., & Xu, C. "Multi-Agent Network Intrusion Active Defense Model Based on Immune Theory." Wuhan University Journal of Natural Sciences, vol. 12, no 1, p. 167-171, 2007.

[Lv et al., 2020] Lv, L., Wang, W., Zhang, Z., & Liu, X. "A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine." Knowledge-Based Systems, 2020.

[Lepuschitz, 2018] Lepuschitz, W. "Self-reconfigurable manufacturing control based on ontology-driven automation agents." Technische Universität Wien. Thèse de doctorat, 2018.

[Liao etVemuri, 2002] Liao, Y., et Vemuri, V. R. "Use of k-nearest neighbor classifier for intrusion detection." Computers & security, vol. 21, no 5, p. 439-448, 2002.

[Müller et al,2003] Müller, K., alias'Socma, K. M., et Tarbouriech, G. "IDS-Systèmes de Détection d'Intrusion, Partie I." LinuxFocus article number 292, 2003.

[Ma et al., 2015] Ma, L., Min, H., Wang, S., Liu, Y., et Liao, S. "An overview of researchin distributed attitude coordination control," IEEE/CAA Journal of Auto-maticaSinica, vol. 2, no 2, p. 121-133, 2015.

[Maess, 1995] Maess, P. "Artificial life meets entertainment : Life like autonomous agents" Communications of the ACM, , vol. 38, no 11, p. 108-114, 1995.

[Major et Reichgelt, 1990] Major, N., et Reichgelt, H. "ALTO : An automated laddering tool." IOS Press, p. 222-236, 1990.

[Malhotra et al., 2014] Malhotra, V. K., Kaur, H., etAlam, M. A. "An Analysis of Fuzzy Clustering Methods." International Journal of Computer Applications, vol. 94, no 19, p. 9-12, 2014.

[MorissetteetChartier, 2013] Morissette, L., et Chartier, S."The k-means clustering technique : General considerations and implementation in Mathematica." Tutorials in Quantitative Methods for Psychology, vol. 9, no 1, p. 15-24, 2013.

[Mokhtari et Moulkhaloua, 2018] Mokhtari, M, et Moulkhaloua, A. "Système DE Détection D'intrusions Informatiques par Système Multi Agents." mémoire fin d'études, Centre universitaire BELHADJ Bouchaib Ain Temouchent, 2018.

[Morin, 2014] Morin, V. "Étude comparative d'algorithmes de data mining dans le contexte du jeu vidéo." Thèse de doctorat. Université du Québec à Chicoutimi, 2014.

[Mohd et al., 2012] Mohd, W. M. W., Beg, A. H., Herawan, T., & Rabbi, K. F."MaxD K-Means : A Clustering Algorithm for Auto-generation of Centroids and Distance of Data Points in Clusters." 2012.

[Mariette et Rahul, 2005] Mariette, A., et Rahul, K. "Support Vector Machines for Classification." In : Efficient Learning Machines. Apress, Berkeley, CA. p. 39-66, 2015.

[Méziane, 2007] Méziane, I."Stratégie de rendez-vous dans les systèmes multi-agents." mémoire fin d'études, Université AboubekrBelkaid - Ingénieur en informatique, 2007.

[Mukhopadhyay et al., 2008] Mukhopadhyay, A., Roy, A., Das, S., Das, S., et Abraham, A. Population-variance and explorative power of harmony search : an analysis. In : 2008 Third International Conference on Digital Information Management. IEEE. p. 775-781, 2008.

[NikhithaetJabbar, 2019] Nikhitha, M.,etJabbar, M.A. "K NearestNeighbor Based Model for Intrusion Detection System." International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 2, 2019.

[Nwana, 1996] Nwana, H. "Software agents : An overview," Knowledge and Engineering Review, vol. 11, no 3, p. 205-244, 1996.

[Nadali et al., 2011] Nadali, A., Kakhky, E. N., et Nosratabadi, H. E. Evaluating the success level of data mining projects based on CRISP-DM methodology by a Fuzzy expert system. In : 2011 3rd International Conference on Electronics Computer Technology. IEEE. p. 161-165, 2011.

[Okamoto et Ishida, 2007] Okamoto, T.,et Ishida,Y."Framework of an immunity-based anomaly detection system for user behavior." In : Proceedings of the International Conference on Knowledge-Based Intelligent Information and Engineering Systems, KES, 2007.

[Ogungbemi, 2019] Ogungbemi, H. S."Etude pour la sécurisation d'un réseau par la mise en place d'un pare-feu open source : cas de C.A.F.E. informatique & télécommunications." 2019.

[O'neill, 2006] O'neill, B. Elementary differential geometry. Elsevier, 2006.

[Pillou et Bay, 2016] Pillou, J. F., et Bay, J. P. "Tout sur la sécurité informatique-4e édition." Dunod, 2016.

[Poinsot, 2019] Poinsot, L. "Introduction à la sécurité informatique." support de cours, Université Paris 13.3,2019.

[Pillou et Bay, 2013] Pillou, J. F., et Bay, J. P."Sécurité informatique 3ème édition." Dunod, Paris 2013.

[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. p. 102-124, 2004.

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

[Russel et Norvig, 1995] Russel, S., et Norvig, P. "Artificial intelligence – A modern approach."Prentice Hall, 1995.

[Reddy, 2013] Reddy, E. K. "Neural networks for intrusion detection and its applications." In : Proceedings of the World Congress on Engineering. Vol. 2. No. 5. p. 3-5, 2013.

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

[Rakotomalala, 2005] Rakotomalala, R. "Arbres de décision." Revue Modulad, vol. 33, p. 163-187, 2005.

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

[Sweeney et al,2003] Sweeney, M., Baumrucker, C. T., Burton, J. D., et Dubrawsky, I. "Cisco security professional's guide to secure intrusion detection systems." SyngressPublishing, 2003.

[Steinhaus, 1956] Steinhaus, H. "Sur la division des corpmateriels en parties." vol. 1, no 804, p. 801, 1956.

[Subba et al., 2015] Subba, B., Biswas, S., etKarmakar, S. "Intrusion detection systems using linear discriminant analysis and logistic regression." 2015 Annual IEEE India Conference (INDICON). IEEE, 2015.

[Shenfield et al., 2018] Shenfield, A., Day, D., et Ayesh, A. "Intelligent intrusion detection systems using artificial neural networks." ICT Express, 42, vol. 4, no 2, p. 95-99, 2018.

[Staub et al., 2015] Staub, S., Karaman, E., Kaya, S., Karapınar, H., etGüven, E. "Artificial neural network and agility." Procedia-Social and Behavioral Sciences, vol. 195, p. 1477-1485, 2015.

[Song et al., 2009] Song, Y., Keromytis, A. D., et Stolfo, S. J. "Spectrogram : A Mixture-of-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.

[Sutter et Kalivas., 1993] Sutter, J. M., et Kalivas, J. H. "Comparison of forward selection, backward elimination, and generalized simulated annealing for variable selection." Microchemical journal, vol. 47, no 1-2, p. 60-66, 1933.

[Salvatore et al., 2000] Stolfo, J., Fan, W., Lee, W., Prodromidis, A., et Chan, P. K. "Cost-based modeling and evaluation for data mining with application to fraud and intrusion detection." Results from the JAM Project by Salvatore. p. 1-15, 2000.

[Tay et al., 2014] Tay, B., Hyun, J. K., et Oh, S."A machine learning approach for specification of spinal cord injuries using fractional anisotropy values obtained from diffusion tensor images." Computational and mathematicalmethods in medicine, 2014.

[Torre, 1999] Torre, F. "GloBo : un algorithme stochastique pour l'apprentissage supervisé et non-supervisé. " In : M. Sebag, editeur, Actes de la Première Conférence d'Apprentissage, 1999.

[Touzet, 1992] Touzet, C. "les réseaux de neurones artificiels, introduction au connexionnisme."1992.

[VarunaetNatesan, 2015] Varuna,S., et Natesan, P."An Integration of K-Means Clustering and Naïve Bayes Classifier for Intrusion Detection." In : 2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN). p. 1-5, 2015.

[Wooldridge, 2009] Wooldridge, M."An introduction to multiagent systems."John Wiley & Sons, 2009.

[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.

[Zimmermann, 1998] Zimmermann, P. "An introduction to cryptography." Network Associates Inc NAI, 1998.

[Zaidi, 2011] Zaidi, A."Recherche et détection des patterns d'attaques dans les réseaux IP à haut débits." Thèse de doctorat. Université d'Évry-Val-d'Essonne, 2011.

Webographie

[Advisio, 2019] Advisio. "Antivirus : A Quoi Servent-Ils ?" advisio-com.fr, 2019, consulté le 17 Mars 2020.

[Arnaud, 2002] Arnaud, J. "Les trois types d'attaques réseaux." Securiteinfo, securiteinfo.com 2002, consulté le 7 Février 2020.

[Castelli, 2003] Castelli, V. "Nearest Neighbor Classifiers ."Columbia Engineering University, ee.columbia.edu, 2003, consulté le 5 Avril 2020.

[Cisco, 2020a] Cisco. "What Is Cybersecurity ?" Cisco.com, consulté le 7 Février 2020.

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

[CERT, 2010] CERT."Limites Et Défis Des Antivirus." Computer Emergency Response Team - Industrie Services Et Tertiaire, cert-ist.com, 2010, consulté le 5 Juin 2020.

[Data Analytics Post, 2018] Data Analytics Post "k-Nearest Neighbours." dataanalytics-post.com,consulté le 13 Mai 2020.

[DEV Community, 2019] Lorraine, L."Classification and Regression Analysis with Decision Trees." DEV Community, dev.to, 15 Mai 2019,consulté le 19 Mai 2020.

[Djeffal, 2020] Djeffal, A."Réseaux De Neurones." Abdelhamid-Djeffal.net, abdelhamid-djeffal.net,consulté le 19 Mai 2020.

[Edureka, 2020] Edureka. "Understanding K-Means Clustering with Examples."edureka.co,consulté le 18 Mai 2020.

[GAMA, 2018] GAMA. "GAMA-Platform." gama-platform.github.io, 2018,consulté le 12 Mai 2020.

[GeeksforGeeks, 2020] GeeksforGeeks."Difference between Firewall and Antivirus."geeksforgeeks.org, 2020, consulté le 30 Avril 2020.

[Hacker Combat, 2019] Hacker Combat. "Virtual Private Network (VPN) : What VPN Is And How It Works"hackercombat.com, 2019,consulté le4 Juin 2020.

[KDD, 2007] UCI Knowledge Discovery in Databases Archive "KDD Cup 1999 Data." kdd.ics.uci.edu, consulté Juin 2020.

[Meade, 2019] Meade, T."What Is the Difference between Firewall and Antivirus ?" Colocation America, Colocation American Staff, colocationamerica.com, 2019, consulté le 30 Avril 2020.

[NetBeans, 2019] NetBeans. "NetBeans." netbeans.org, consulté Juin 2020.

[Petters, 2020] Petters, J. "What's The Difference Between a Proxy and a VPN ?"Inside Out Security, www.varonis.com, 2020, consulté le28 Avril 2020.

[PostgreSQL, 1996] PostgreSQL. "PostgreSQL." postgresql.org, consulté Juin 2020.

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

[Simmons, 2019] Simmons, J.H. "Disadvantages of VPN That You Should Know Before Using It."VPNCrew, vpncrew.com, 2019,consulté le 4 Juin 2020.

[Suini, 2015] Suini, J."Le principe du VPN." Supinfo.com,2015, consulté le 10 Février 2020.

[Souvik, 2020] Souvik. "Proxy Server : The Advantages Disadvantages of a Proxy Server."RS Web Solutions, rswebsols.com,2020 consulté le 9 Février 2020.

[Securiteinfo, 2006] Securiteinfo"Le grand livre de la sécurité informatique." securiteinfo.com, Editions du 6 novembre 2006, consulté le 7 Février 2020.

[Serengil, 2018] Serengil, S. "A Step By Step Regression Tree Example." Sefik Ilkin Serengil, sefiks.com, 2018,consulté le 30 Avril 2020.

[Towards Data Science,2019] Pupale, R. "Support Vector Machines(SVM) - An Overview." Medium, Towards Data Science, towardsdatascience.com,11 Feb. 2019,consulté le 10 Avril 2020.

[Unb, 2019] Canadian Institute for Cybersecurity."NSL-KDD dataset." unb.ca, consulté Juin 2020.

[Valgasu, 2017] Valgasu. "IP Spoofing." Sécurité Informatique : Audits Et Conseils En Cybersécurité, securiteinfo.com, 20 Mai 2017, consulté le 8 Février 2020.

[WikiStat, 2016] WikiStat "Machines à vecteurs supports." wikistat.fr, 2016,consulté le15 Avril 2020.

Résumé

Résumé

L'expansion rapide des réseaux informatiques au cours de la dernière décennie a entraîné une augmentation considérable du nombre d'attaques et d'intrusions. La sécurité est devenue un problème crucial pour les systèmes informatiques. Le système de détection d'intrusions est l'un des moyens importants pour atteindre une haute sécurité dans les réseaux informatiques.

Plusieurs approches d'apprentissage automatiques ont été proposées pour développer des systèmes de détection d'intrusions qui visent à augmenter la précision et diminuer le taux de faux positif sans la considérer le facteur temps. Cela rend leurs systèmes incapables d'opérer dans les réseaux en temps réel.

L'objectif de notre travail est de réaliser un système de détection d'intrusions basée réseau tout en maximisant les taux de détection et en réduisant le temps de calcul sans perte d'informations. Afin d'atteindre notre objectif nous avons dans un premier temps appliqué l'algorithme des k Plus Proches Voisins (K-PPV). Cet algorithme est gourmant en temps de calcul par conséquent nous avons appliqué dans un deuxième temps l'algorithme des Plus Proches Voisins Condensés (PPVC) qui consiste principalement à réduire la base d'apprentissage ce qui diminue considérablement le temps de calcul. Et pour contribuer encore à cette diminution en temps, nous avons réduit le nombre d'attributs dans les paquets de la Base de données en procédant à une sélection des caractéristiques les plus pertinentes.

Afin d'assurer une simulation de nos SDIRs, nous avons fait recours au Système Multi-agents (SMA). Ce dernier permet une distribution physique et d'expertise contribuant à son tour à la réduction du temps d'exécution.

Les résultats expérimentaux sur l'ensemble des données NSL-KDD, une version améliorée de l'ensemble de données KDD 99, montrent que nos approches proposées SMA-K-PPV et SMA-PPVC sont efficaces en taux de détection. En termes de temps de calcul, l'approche SMA-PPVC est plus rapide, ce qui la rend plus adaptés pour le déploiement dans la surveillance des réseaux en temps réel et l'analyse de détection d'intrusion.

Mots-clés : Système de Détection d'Intrusions basé sur le Réseau (SDIR), sélection des caractéristiques, Algorithme des K-Plus proches Voisins (K-PPV), Plus Proches Voisins Condensés (PPVC), Système Multi-Agents (SMA), Réduction de dimensionnalité, Taux de variance, Elimination en arrière.

Abstract

The fast expanding of computer networks over the past decade has resulted in a dramatic increase in the number of attacks and intrusions. Security has become a crucial problem for computer systems. One of the important ways to achieve high security in computer networks is by intrusion detection systems.

Several machine learning approaches have been proposed to develop intrusion detection systems that aim to increase accuracy and decrease the rate of false positives. These approaches, unfortunately don't take the factor of time into account. This makes their systems unable to operate in real-time networks.

The objective of our work is to achieve a network-based intrusion detection system while maximizing detection rates and reducing computing time without loss of information. In order to achieve this objective, we first apply the algorithm of the k Nearest Neighbours (KNN). This algorithm is time-consuming ; therefore, we apply in a second time the algorithm of the Condensed Nearest Neighbours (CNN). It consists mainly to reduce the learning data set which considerably reduces the computation time. To further contribute to reach our goal we reduced the number of attributes in the data sets packages by selecting the most relevant characteristics.

In order to simulate our SDIRs, we used a Multi-agent system (MAS). This paradigm allows physical distribution and expertise, which in turn contributes to reducing execution time.

Experimental test has been done on e NSL-KDD dataset, an improved version of the KDD 99 data set. The obtained results indicate that our proposed approaches MAS-KNN and MAS-CNN give a good detection rates. In terms of computation time, the MAS-CNN approach is faster than MAS-KNN. It is than more suitable for deployment in real-time network monitoring and intrusion detection analysis.

Keywords : Network Intrusion Detection System (NIDS), Features selection, K Nearest Neighbours (KNN), Condensed Nearest Neighbours (CNN), Multi-Agents System (MAS), Dimensionality reduction, Backward elimination, Variance rate.

ملخص

أدى التوسع السريع لشبكات الكمبيوتر على مدى العقد الماضي إلى زيادة كبيرة في عدد الهجمات وعمليات التطفل. أصبح الأمن مشكلة حاسمة لأنظمة الكمبيوتر. تعد أنظمة كشف التسلل من الطرق المهمة لتحقيق أمان عالي في شبكات الكمبيوتر.

تم اقتراح العديد من مناهج التعلم الآلي لتطوير أنظمة كشف التسلل التي تهدف إلى زيادة الدقة وتقليل معدل الإيجابيات الخاطئة. هذه الأساليب، للأسف، لا تأخذ عامل الوقت في الاعتبار. وهذا يجعل أنظمتهم غير قادرة على العمل في شبكات الوقت الحقيقي.

الهدف من عملنا هو تحقيق نظام كشف التسلل مستوى على الشبكة مع زيادة معدلات الكشف وتقليلٍ وقت الحوسبة دون فقد المعلومات. من أجل تحقيق هذا الهدف، نطبق أولاً خوارزمية ك جار اقرب (ك ج أ). هذه الخوارزمية تستغرق وقتا طويلا. لذلك، نطبق في المرة الثانية خوارزمية أقرب الجيران المكثف (أ ج م) والتي تهدف أساسًا من تقليل مجموعة بيانات التعلم مما يقلل بشكل كبير من وقت الحساب. لزيادة المساهمة في الوصول إلى هدفنا، قمنا بتخفيض عدد الحقولفي حزم قاعدة البيانات من خلال تحديد الخصائص الأكثر دلالة. من أجل محاكاة نظام كشف التسلل الخاص بنا، استخدمنا نظام متعدد الوكلاء (ن م و). يسمح هذا النموذج بتوزيع مادي وخبرة تساهم بدورها في تقليل وقت التنفيذ.

تم إجراء الاختبار التجريبي على مجموعة بيانات NSL-KDD وهي نسخة محسنة من مجموعة بيانات KDD 99 . تشير النتائج التي تم الحصول عليها إلى أن نهجينا المقترحين (ن م و - ك ج أ) و (ن م و - أ ج م) يعطيان معدلات كشف جيدة . من حيث الوقت الحسابي، فإن نهج (ن م و - أ ج م) أسرع من (ن م و - ك ج أ) . إنه أكثر ملائمة لمراقبة الشبكة في الوقت الحقيقي وتحليل كشف التسلل.

كلمات مفتاحية : نظام كشف التسلل إلى الشبكة (ن ك ت ش) ، تخفيض عدد السمات، خوارزمية ك جار اقرب (ك ج أ) ، خوارزمية أقرب الجيران المكثف (أ ج م)، نظام متعدد الوكلاء (ن م و)، تخفيض الأبعاد، الإزالة من الخلف، معدل التباين