République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et Recherche Scientifique

Université Saad Dahlab de Blida Faculté des sciences

Département d'Informatique

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique

Filière : Informatique

Spécialité : Informatique

Option : Ingénieure de logiciel

Système de détection d'intrusion avec une approche D'apprentissage automatique

Réalisépar : - BAH DIDI EL MOKHTAR SALEM

Encadreur : GHEBGHOUB.Y

Devant le jury composé de :

Mlle Boustia N…….Professeur U. BLIDA 1

Mme. Mezzi M……..MCB U. BLIDA 1

Remerciements

Tout d’abord, nous remercieront Allah de nous avoir aidé et donné la force et la volonté de réaliser ce travail

Ensuite, nous tenons à exprimer nos plus vifs remerciements et gratitude à notre Promotrice**Mme Yasmine GHEBGHOUB** sur son encadrement continu, pour les remarques constructives Qu’il nous a fournies ainsi que pour ses précieux conseils durant toute la période de notre travail. On la remercie également pour la confiance qu’il nous a accordée et pour la grande liberté d’idées et de travail qu’il nous a donnée. Nous n’oublierons pas aussi de les remercier pour ses qualités humaines, son hospitalité et son soutien qui ont permis de bien mener bien cet ouvrage.

Nous tenons à remercier les membres du jury d'avoir bien voulu participer à l'évaluation de ce travail.

Résumé

Avec le développement de la technologie les réseaux informatiques sont devenus de plus en plus vastes et ouverts. Cette évolution a donné naissance à de nouvelles techniques permettant l'accessibilité aux réseaux et aux systèmes d'information dans le but de faciliter les transactions. Par conséquent, ces techniques ont également donné naissance à de nouvelles formes de menaces

Le système de détection d'intrusion est un processus important dans la sécurité des réseaux. Aujourd'hui les techniques d'intelligence artificielle sont plus en plus utilisées pour renforcer et augmenter les taux de détection des attaques.

Dans ce travail, nous proposons un système intelligent qui se base sur l'apprentissage Multi niveau. Notre proposition a été testée en utilisant la base de connaissance KDD 99.

Mots clés : IDS, apprentissage, KDD,attaque

Abstract

With the development growing of network technology and the information exchange, the computer networks became increasingly wide and opened. This evolution gave birth to new new techniques allowing the accessibility of the networks and information systems with an aim of facilitating the transactions. Consequently, these techniques gave also birth to new forms of threatsNowadays new intelligent techniques have been used to improve the intrusion detection process. This work proposes a hybrid intelligent intrusion detection system to improve the detection rate for known and unknown attacks.

The proposed model consists of multi-level. Each level is implemented with the technique which gave best experimental results. We have used KDDCUP'99 .it's the mostly widely used data set evaluation of these systems.

Key words: IDS , machine learning, attack,KDD

ملخص

يمكن تعريف نظام كشف التسلل بأنه نظام آلي يتمثل دوره في اكتشاف الاختراقات في نظام الكمبيوتر، وهناك نوعان، الأول يعتمد على التوقيع والآخر يعتمد على الحالات الشاذة، الهدف من هذا العمل الماجستير هو إيجاد حلول للحد من الإنذارات الخاطئة عن الحالات الشاذة القائمة علىن.كتباستخدام طريقة الفرقة التقليدية وطريقة جديدة مقترحة.

مزيج من ن.كت والتعلم الآلي يعطين.كت أفضل منن.كتالمستندة إلى التوقيع، مجموعة البيانات المستخدمةهي*KDDcup-99* حقق الأسلوب الجديد المقترح نتائج جيدة مقارنةً بالمصنف القياسي

الكلمات المفتاحية: نظام كشف التسلل، تقنية الكشف، ، التعلم الآلي، أمن الشبكات، ،KDD99.

Table des matières

Liste des figures

Liste des tableaux

Introduction générale

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

Les attaques informatiques ne concernent pas uniquement les entreprises et les sociétés mais touchent de plus en plus les gouvernements et les structures sensibles des pays et même les individus. A tel point que les services de sécurité ont clairement émis

des menaces, en affirmant qu'ils envisageraient toutes les options possibles en cas de cyber-attaques.

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

Dans ce travail de master, nous nous somme focalisés l'amélioration des systèmes de la détection d'intrusion dans les réseaux en se basant sur certains algorithmes d'apprentissage automatique.

1) Contexte :

La sécurité des réseaux est devenue plus en plus importante surtout que les données sont aujourd'hui stockées et manipulées en ligne. D'autre part, La prévention des attaques est très difficile en utilisant les solutions de sécurité passives, parfeu ou autres mécanismes. Les systèmes de détection (IDS) représentent une technologie efficace aide à la protection contre les différentes attaques.

Le principal but de notre travail est de concevoir et développer un système de prévention d'attaques qui se base sur des travaux existants dans ce domaine. Ce

modèle vise à améliorer le taux de détection des intrusions en utilisant les techniques de l'apprentissage automatique.

2) Problématique et objectifs :

L'avènement des réseaux offre les services immenses à ceux qui l'utilisent. Ces services font l'objet de plusieurs attaques et les mécanismes de sécurité deviennent une nécessité pour les protéger. Les systèmes de détection d'intrusion réseaux (NIDS) sont l'un des mécanismes le plus utilisé aujourd'hui pour détecter les intrusions. Le but des systèmes de détection d'intrusion réseaux est de protéger les réseaux des attaques qui ne peuvent pas être identifiés par des firewalls. L'un des problèmes majeurs est le taux des faux positifs c à d la mauvaise classification de certaines actions

Notre objectif dans ce travail est d'améliorer les résultats des classifications des NIDS en se basant sur plusieurs algorithmes d'apprentissage automatique.

3) Structure du document :

Ce mémoire se décompose en quatre chapitres, incluant cette introduction

Nous allons présenter dans le chapitre 1 une étude générale des attaques et intrusions rencontrées, pour avoir une meilleure idée du problème. Nous étudions en détail la sécurité des réseaux informatiques, y compris les attaques, les vulnérabilités, l'intrusion, puis, les systèmes de détection d'intrusion comme contre-mesure pour y faire face. La dernière section de ce chapitre sera consacrée à l'exposition de quelques systèmes de détection d'intrusions existants.

Dans le chapitre 2, nous allons présenter une vue globale de l'apprentissage automatique suivie par une brève présentation de quelques méthodes innées de l'apprentissage telle que Réseaux de neurone, la régression.

Nous présenterons dans le chapitre 3 l'apport de ces méthodes d'intelligence décrites dans le chapitre 2 à la sécurité des réseaux et en particulier à la détection d'intrusion, dans lequel nous présentons notre travail qui se base sur l'apprentissage multi niveau.

Au final, nous terminerons ce travail par d'expérimentation et résultats pour présenter la partie application.

CHAPITRE 1 : Introduction sur les systèmes de détection d'intrusions

1.1 Introduction

Dans le chapitre courant, nous allons expliquer plusieurs termes et définitions relatifs à notre thème que nous avons jugé nécessaires à connaitre pour une bonne compréhension du sujet.

1.2. Définitions

- **Intrusion**

Une intrusion peut être considérée comme l'ensemble d'actions qui ont pour but de compromettre l'intégrité, la confidentialité ou la disponibilité d'une ressource. Ces actions de franchissement d'un accès non-autorisé ou de manipulation interdite d'une ressource, peuvent être menées par un individu externe n'ayant aucun privilège sur les ressources d'un système, ou par un individu interne qui outrepasse ses privilèges.

- Détection d'intrusion

La détection d'intrusion est un ensemble de techniques et de méthodes employées dans l'analyse des informations collectées par les mécanismes d'audit de sécurité pour détecter toute activité suspecte au niveau du réseau et ses hôtes.

- Système de détection d'intrusion

C'est une combinaison de logiciel et de matériel qui essaie de réaliser la détection d'intrusions. Un système de détection d'intrusions peut se définir comme un système automatisé dont le rôle est la détection des intrusions dans un système informatique tout en examinant les audits de sécurité fournis par le système d'exploitation ou bien les outils de contrôle du réseau. Son but principal est la détection des utilisations non autorisées, les mauvaises utilisations et les abus dans un système informatique par les utilisateurs internes et externes [1].

Dans [2], un système de détection d'intrusions est défini comme étant comparé à une alarme de cambriolage. Par exemple, le système de serrure dans une voiture protège la voiture contre le vol. Mais si quelqu'un casse le système de serrure et essaie de voler la voiture, c'est l'alarme anti cambriolage qui détecte que la serrure a été cassée et alerte le propriétaire en donnant une alarme. Le système de détection d'intrusions d'une manière similaire complète la sécurité du pare-feu. Le pare-feu protège un système contre des attaques malveillantes et le système de détection d'intrusions détecterait si quelqu'un tente de passer le pare-feu et d'accéder au côté sûr du système, et alerte le gestionnaire au cas où il y aurait infraction dans la sécurité [2].

1.3 Types des systèmes de détection d'intrusion

Géneralement,il existe deux types majeurs d'IDS :

1.3.1 IDSs à base de signature

Est une technique de détection qui n'est pas axée sur la recherche d'intrusions. Elle se concentre sur l'analyse d'un comportement en le comparant à un modèle considéré comme normal

Généralement, les IDS réseaux se basent sur un ensemble de signatures qui représentent chacune le profil d'une attaque. Une approche à base de signature consiste à rechercher dans un flux réseau les empreintes d'attaques connues, à l'instar des antivirus.

Une signature est définie comme une séquence d'événements et de conditions relatant une tentative d'intrusion. La reconnaissance est alors basée sur le concept de" Pattern Matching". Si une attaque est détectée, une alarme peut être remontée si l'IDS est en mode actif, sinon, l'IDS se contente d'archiver cette attaque [3].

1.3.2 IDSs à base d'anomalie

Les IDSs à base d'anomalie dont le déploiement nécessite une phase d'apprentissage pendant laquelle l'outil va apprendre le comportement "normal" des flux applicatifs présents sur son réseau. Ainsi, chaque flux et son comportement Standard doivent être déclarés. L'IDS se chargera d'émettre une alarme si un flux anormal est détecté, mais ne pourra pas spécifier la criticité de l'éventuelle attaque. Les IDS comportementaux sont apparus bien plus tard que les IDS à signature et ne bénéficient pas encore de leur maturité. Ainsi, l'utilisation de tels IDS peut s'avérer délicate dans le sens où les alarmes remontées pourraient contenir une quantité importante de fausses alertes [3].

1.4 Familles de système de détection d'intrusion

Les IDSs peuvent se classer selon trois catégories majeures selon qu'ils s'attachent à surveiller :

1.4.1 Les NIDSs (Network Based Intrusion Detection Systems)

Un NIDS se découpe en trois grandes parties : la capture, les signatures et les alertes. Il écoute donc tout le trafic réseau, puis analyse les flux en transit sur le réseau et génère des alertes si des paquets semblent dangereux (voir figure 1).

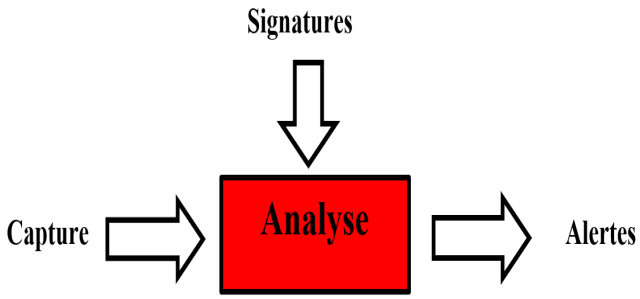

Figure 1: les composants d'un NIDS

Le rôle essentiel d'un NIDS est l'analyse et l'interprétation des paquets circulant sur ce réseau. L'implantation d'un NIDS sur un réseau se fait de la façon suivante : des capteurs sont placés aux endroits stratégiques du réseau et génèrent des alertes s'ils détectent une attaque. Ces alertes sont envoyées à une console sécurisée, pour les analyser et les traiter éventuellement [4].

Les capteurs du réseau sont placés en mode furtif (ou stealth mode), de façon à être invisibles aux autres machines. Pour cela, leur carte réseau est configurée en mode « promiscuous », c'est à dire le mode dans lequel la carte réseau lit l'ensemble du trafic, de plus aucune adresse IP n'est configurée [4].

1.4.2 Les HIDSs (Host Based Intrusion Detection Systems)

Le HIDS réside sur un hôte particulier, il analyse exclusivement l'information concernant cet hôte. Le HIDS se comporte comme un démon ou un service standard sur un hôte serveur/système. De plus, l'impact sur la machine concernée est sensible immédiatement, par exemple dans le cas d'une attaque réussie par un utilisateur. Ces systèmes de détection d'intrusions utilisent deux types de sources pour fournir une information sur l'activité de la machine : les logs et les traces d'audit du système d'exploitation : les traces d'audit sont plus précises et détaillées et fournissent une meilleure information alors que les logs qui ne fournissent que l'information essentielle sont plus petits (voir figure 2) [4].

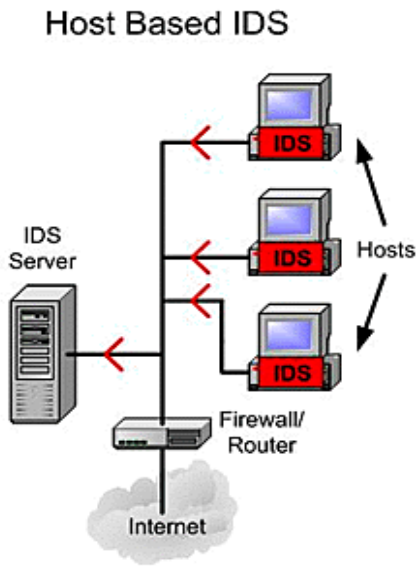

Figure 2: IDS basé sur l'hôte [3]

1.4.3 Les IDSs hybrides

Les IDS hybrides rassemblent les caractéristiques de plusieurs IDS différents. En pratique, on ne retrouve que la combinaison de NIDS et HIDS. Ils permettent, en un seul outil, de surveiller les réseaux et les terminaux. Les sondes sont placées en des points stratégiques, et agissent comme NIDS et/ou HIDS suivant leurs emplacements. Toutes ces sondes remontent alors les alertes à une machine qui va centraliser le tout (Figure 3), et agréger/lier les informations d'origines multiples [5].

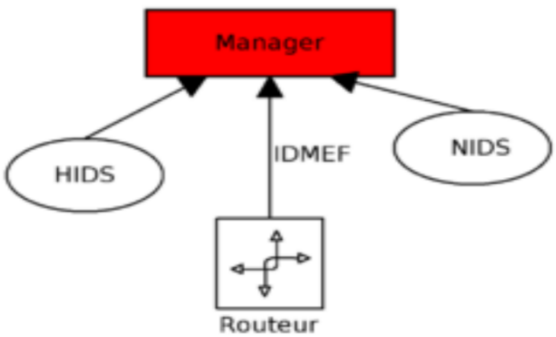

Figure 3: IDS hybride

Les IDSs hybrides sont donc basés sur une architecture distribuée, où chaque composant unifie son format d'envoi d'alerte (Exemple typique : IDMEF : Intrusion Detection Message Exchange Format) permettant à des composants divers de communiquer et d'extraire des alertes plus pertinentes.

Modi et al. proposent dans [60] un tableau récapitulatif, mettant en avant les avantages,

désavantages de chacun de ces IDS. Ils discutent également dans ce tableau de la responsabilitédu déploiement et de mise à jour de ces systèmes.

Type d'IDS	Avantages	Désavantages	Placement	Déploiement & responsabilité
HIDS	— Détecte les intrusions en surveillant les fichiers, appels système ou événements réseau de l'hôte — Pas besoin d'équipement en plus	— Besoin de l'installer sur chaque machine — Détection d'attaques locales uniquement	Machine virtuelle ou physique	Utilisateur & administrateur
NIDS	— Détecte les intrusions en surveillant le trafic réseau — Besoin d'être placé sur le réseau (physiquement) — Peut surveiller plusieurs systèmes en même temps	— Difficile de détecter des intrusions provenant de contenu chiffré — Ne peut pas détecter les attaques ne transitant pas par le NIDS	Réseau physique ou virtuel	Administrateur
Hypervisor-IDS	— Détecte les intrusions entre les VM en analysant le trafic réseau	— Récent et difficile de s'interfacer avec les hyperviseurs — Composant critique	Hyperviseur	Administrateur
DIDS	— Caractéristiques des HIDS/-NIDS — Détecte les intrusions en associant plusieurs systèmes de détection d'intrusions (HIDS/NIDS)	— Coût de déploiement et de configuration — Surcoût en communication — Coopération entre les systèmes complexe	Partout	Utilisateur & Administrateur

Tableau1 : récapitulatif des différents placements des IDS

1.5Attaques réseaux

Dans cette section, nous nous intéressons aux attaques que l'on souhaite détecter dans notre travail.

Dans [35], Ghorbani et al. Définissent les attaques réseaux comme l'activité malicieuse visant l'interruption, la dégradation, la perturbation de services accessibles aux travers d'un réseau. L'objectif de ces attaques est de porter atteinte à l'intégrité, la disponibilité ou la confidentialité de ces services. Les attaques sont diverses et variées, et peuvent cibler une machine ou un système complet.

Quant à Hansman et al., ils définissent dans [38] les attaques réseaux comme des attaques visant un réseau ou des utilisateurs en manipulant les protocoles réseaux de la

couche physique à la couche application. Dans ce mémoire, nous considérons qu'une attaque réseau est une attaque dont le médium de communication est le réseau. Nous présentons dans la suite de cette section des attaques réseaux.

- **Les dépassements de tampon [19, 52]** (buffer overflows en anglais) sont des opérations consistent à prendre le contrôle, à interrompre ou à dégrader un processus. Ces attaques s'appuient sur le concept de l'espace mémoire et de son placement, et écrivent en dehors de son espace mémoire, de manière à effacer des informations importantes pour le processus en question. Bien que ces attaques ciblent un système et non le réseau, leur moyen de propagation privilégié reste le réseau, avec par exemple les attaques XSS (Cross-Site Scripting), qui exécutent des routines lors d'une navigation habituelle sur le web.

- **Les virus, les vers et les chevaux de Troie [91, 100]** sont des programmes dont l'objectif est d'infecter une machine hôte. Les actions alors réalisées par ces programmes sont variées : il peut s'agir d'envoi de courriels indésirables, d'écoute passive ou de recherche d'informations confidentielles, ou encore de manipulation de l'hôte pour réaliser des actions malicieuses.
  Les vecteurs de propagation de ces attaques sont également très divers, mais la plupart de ces menaces se répliquent en utilisant le réseau. Qu'il s'agisse d'une pièce jointe dans un courriel, ou bien de recherche active de victimes potentielles, la phase de réplication peut être détectée en cherchant des motifs particuliers dans les communications portées par le réseau.

- **Les botnets [29]** sont des réseaux de machines composés de dizaines jusqu'à des milliers de machines infectées, utilisées à l'insu de leur propriétaire pour perpétrer des attaques réseaux. L'hôte infecté, aussi appelé zombi ou bot, peut être infecté grâce aux attaques décrites dans les paragraphes précédents. Le serveur de contrôle peut alors dicter les actions que les bots doivent réaliser, à distance et au travers d'un canal de communication spécifique et sécurisé. Une

fois formé, un botnet peut être monétisé : vous louez à la durée ou au service un certain nombre de machines infectées.

- **Le Déni de service [102, 65] (Denial of Service en anglais)** est une attaque dont l'objectif principal est de dégrader les performances d'un système. Ces attaques peuvent cibler différents composants du système, tels que la mémoire, le processeur ou encore la carte réseau. La popularité de ces attaques est grandissante ces dernières années, à tel point que l'on parle maintenant de Distributed Denial of Service (DDoS) où l'attaque est menée non plus par un seul système, mais par un grand nombre d'hôtes attaquants. Le vecteur de propagation par excellence de cette attaque est le réseau.

1.6 Evolution des IDSs

Les systèmes de détection d'intrusions ont beaucoup évolué depuis le modèle proposé par [6]. Plusieurs techniques ont été introduites afin d'améliorer leurs performances et rendre leur détection plus précise. Dans ce qui suit nous allons présenter un ensemble de systèmes des plus connus.

1.6.1 IDES

Comme cité précédemment, le premier modèle d'IDS "IDES" (Intrusion Detection Expert System) a été réalisé par Dorothy et Peter Denning [6]. Ce modèle fait appel à des techniques statistiques afin de caractériser un comportement anormal, et se base sur un ensemble de règles pour détecter les violations. Dorothy a émis, plus tard [6], qu'il était possible de détecter des intrusions dans un système informatique indépendamment de ce dernier, des applications installées et de sa vulnérabilité, et cela à travers un modèle reformulant les comportements des utilisateurs par les systèmes de détection d'intrusion.

1.6.2 Haystack

"Haystack" [7] représente une variante de IDES. Ce prototype [7] a été développé pour la détection des intrusions dans un environnement multi-utilisateurs de l'Air Force Computer System (il s'agissait de la plateforme standard de l'Air force à

l'époque) [57]. Pour détecter les intrusions, le système utilise deux méthodes de détection : la détection

1.6.3 MIDAS

MIDAS [8] a été développé par le centre national de la sécurité (NCSC), en collaboration avec le laboratoire informatique international SRI, pour fournir une détection d'intrusions pour les réseaux mainframe. Les auteurs se sont inspirés des travaux antérieurs de Denning et al. [56]. Ce modèle est construit autour de l'idée de détection d'intrusions heuristique et comporte une base de règles appelée P-Best écrite en Lisp.

Après MIDAS les recherches se sont dirigées vers l'utilisation des systèmes experts dans l'écriture des règles [9], ce qui permet une mise à jour dynamique de ces règles.

1.6.4 Discovery

L'IDS Discovery regroupe une approche hybride basée sur les statistiques et les systèmes experts [10]. Cet IDS analyse les fichiers journaux d'une application. Discovery nécessite des méthodes statistiques pour la détection de profiles et se base sur un système expert pour la détection d'intrusions.

1.6.5 Wisdom&Sense

Wisdom&Sense ou WRS [11] est un système de détection d'anomalies, qui a été développé entre 1984 et 1989. Il est intéressant de noter que WRS à sa création n'a pas été destiné à être appliqué à la sécurité informatique, mais plutôt à un problème lié au contrôle de matières nucléaires [12]. Ce système est unique dans son approche de détection d'anomalies : il étudie l'historique des données d'audit et produit un arbre de règles décrivant le comportement normal, ces règles sont ensuite introduites dans un système expert qui évalue les données de vérification récentes et déclenche une alerte lorsque les règles indiquent un comportement anormal.

1.6.6 NSM

NSM [13], [14] a été le premier système à utiliser directement le trafic réseau entant comme source de données d'audit. Cet IDS écoute passivement tout le trafic qui passe par un réseau local de diffusion et en déduit le comportement intrusif. L'idée de

cette approche découle de l'observation que plusieurs autres systèmes de détection d'intrusions essayaient d'atténuer les problèmes des différentes plateformes en suivant leurs pistes d'audit.

1.6.7 Hyperview

Hyperview [15] est un système de détection d'intrusions fondé sur deux composantes principales. La première consiste en un système expert qui surveille les pistes de vérification de signes d'intrusions connues de la communauté de sécurité. La seconde est une composante de réseau de neurones qui apprend le comportement d'un utilisateur de manière adaptative et envoie une alarme quand la piste d'audit s'écarte du comportement appris auparavent.

1.6.8 DIDS

DIDS [16] est un système de détection d'intrusions distribué incorporant Haystack et NSM, vues précédemment. DIDS est constitué de trois composantes principales. Sur chaque hôte, un moniteur effectue la détection locale d'intrusion, résume les résultats et les communique au directeur DIDS. En outre, chaque segment broadcast du LAN possède ses propres moniteurs qui surveillent le trafic dans ce LAN et rapportent au directeur du DIDS leurs observations. Finalement, le directeur DIDS, qui est un système constitué du responsable de communication et d'un système expert, analyse les observations des moniteurs et en communique les résultats à l'agent de sécurité du système (SSO).

1.6.9 NIDES

NIDES [22] [23] est le successeur du projet IDES. NIDES suit les mêmes principes généraux que les versions ultérieures de l'IDES : il a une base forte de détection d'anomalies, complétée par un composant de système expert à base de signatures. Ce dernier est mis en œuvre en utilisant un système expert P-BEST. Le système NIDES est fortement modulaire, avec des interfaces bien définies entre les composants, construit sur une architecture client-serveur. Il est centralisé dans la mesure où l'analyse s'exécute sur un hôte spécifique et recueille des données de différents hôtes à travers un réseau informatique.

1.6.10 GrIDS

GrIDS [24] permet de construire des graphiques représentant l'activité du réseau pour faciliter la détection d'intrusions, particulièrement dans les grands réseaux. Les graphiques codifient les hôtes sur un réseau comme des nœuds, et les connexions entre les hôtes comme des raccords entre ces nœuds. Le choix du trafic fait pour représenter l'activité sous forme de bords est effectué sur la base d'ensembles de règles écrites par l'utilisateur. Les événements du réseau sont représentés à travers un mode graphique qui permet à l'observateur de déterminer la présence d'une activité suspecte. Il serait contraignant de reporter toute l'activité réseau dans un même graphique, par conséquent, le système permet pour plusieurs ensembles de règles de définir un graphe pour chaque ensemble. Dans ce cas, toutes les données recueillies sont prises en considération pour définir l'inclusion dans tous les ensembles de règles, et donc deux ensembles de règles différents pourraient rendre ainsi les mêmes données d'audit que deux graphiques différents.

1.7Conclusion

Dans ce chapitre, nous avons donné un aperçu général sur les Ids en décrivant les types des Ids existants. En outre nous avons montré certains types d'attaques qui peuvent menacer les systèmes informatiques à partir des réseaux. Dans le chapitre suivant, nous allons parler sur l'apprentissage automatique.

CHAPITRE 2
LES ALGORITHMES D'APPRENTISSAGE AUTOMATIQUE APPLIQUES DANS LA DETECTION D'INTRUSION

2.1 Introduction

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

Dans ce deuxième chapitre, nous allons présenter les principaux algorithmes d'apprentissage automatique et les différentes Techniques de détection basées sur l'apprentissage automatique.

2.2 Définition d'apprentissage automatique

L'apprentissage automatique ou statistique, aussi appelé « *machine learning* » est un domaine à la jonction des statistiques et de l'intelligence artificielle qui a pour but la résolution automatique de problèmes complexes à partir d'exemples. La démarche de conception d'un modèle par apprentissage nécessite de postuler une fonction, dont les variables sont susceptibles d'avoir une influence sur la grandeur à modéliser. Cette fonction dépend des paramètres ajustables.

L'apprentissage statistique consiste en l'ajustement de ces paramètres de telle manière que le modèle ainsi obtenu présente les qualités requises d'apprentissage et de généralisation [41].

2.3 Différents types d'apprentissage

2.1. 1 Apprentissage supervisé

Dans l'apprentissage supervisé, les données fournies sont des paires : une entrée et une étiquette. On parle alors d'entrées étiquetées. Le but de l'apprentissage est d'inférer la valeur de l'étiquette étant donnée la valeur de l'entrée. On peut distinguer deux grands types d'apprentissage supervisé : la classification et la régression [42].

A. Classification

Lorsqu'on fait de la classification, l'entrée est l'instance d'une classe et l'étiquette est la classe correspondante. La classification consiste donc à apprendre une fonction f class de X= Rd dans Y= N qui associe à un vecteur sa classe. Si le nombre de classe est égal à 2, on parle alors de la classification binaire [41].

B. Régression

Dans le cas de la régression, l'entrée n'est pas associée à une classe mais à une ou plusieurs quantités continues. Ainsi, l'entrée pourrait être les caractéristiques d'une personne (son âge, son sexe, son niveau d'études) et l'étiquette son revenu. La

régression consiste donc à apprendre une fonction de *X =Rd dans Y=Rk* qui associe à un vecteur sa valeur associée [41].

- **Régression des moindres carrés ordinaires :**
- Les moindres carrés sont une méthode permettant d'effectuer une régression linéaire. Vous pouvez considérer la régression linéaire comme une tâche consistant à ajuster une ligne droite à travers un ensemble de points. Il existe plusieurs stratégies possibles pour ce faire, et la stratégie des « moindres carrés ordinaires » est la suivante : vous pouvez tracer une ligne, puis pour chacun des points de données, mesurer la distance verticale entre le point et la ligne et les additionner. ; la ligne aménagée serait celle où cette somme de distances est la plus petite possible [43].

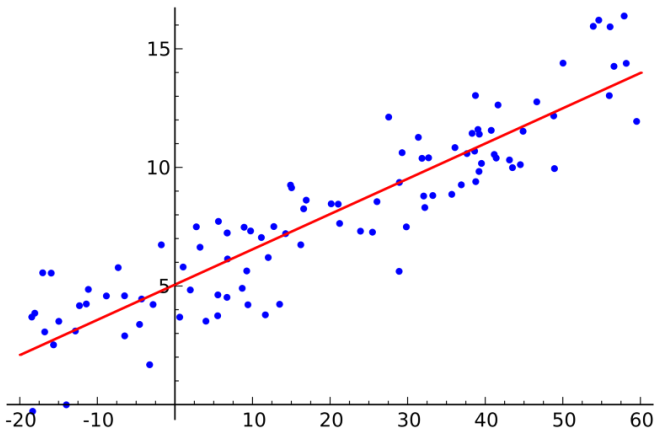

Figure 5 : Régression des moindres carrés ordinaires [43].

Linéaire fait référence au type de modèle que vous utilisez pour ajuster les données, tandis que les moindres carrés correspondent au type de mesure d'erreur que vous minimisez [43].

- **Régression logistique :**La régression logistique est un moyen statistique puissant de modéliser un résultat binomial avec une ou plusieurs variables explicatives. Il mesure la relation entre la variable dépendante catégorique et une ou plusieurs variables indépendantes en estimant les probabilités à l'aide d'une fonction logistique, qui est la distribution logistique cumulative.

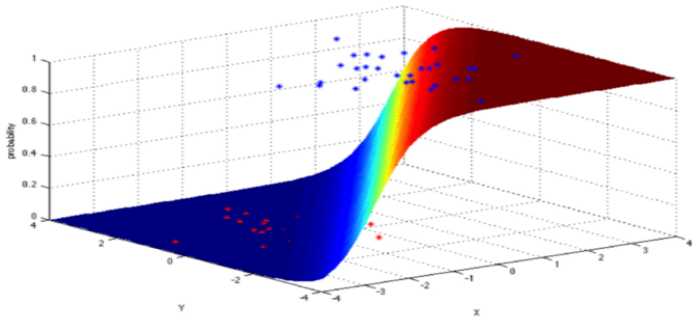

Figure 6: Régression logistique [43].

En général, les régressions peuvent être utilisées dans des applications du monde réel telles que :

- Pointage de crédit.
- Mesurer les taux de réussite des campagnes marketing.
- Prédire les revenus d'un produit donné.
- Est-ce qu'il va y avoir un tremblement de terre un jour particulier [43].

C. **Machines à vecteurs de support (Support Vector Machine) :**

SVM est un algorithme de classification binaire. Étant donné un ensemble de points de 2 types dans N lieu dimensionnel, SVM génère un hyperplan dimensionnel (N - 1) pour séparer ces points en 2 groupes. Supposons que certains points de 2 types soient séparables linéairement. SVM trouvera une ligne droite qui sépare ces points en 2 types et située aussi loin que possible de tous ces points. En termes d'échelle, certains des problèmes les plus importants qui ont été résolus à l'aide de SVM (avec des implémentations correctement modifiées) sont la publicité écran, la reconnaissance de sites de jonction humaine, la détection de genre basée sur l'image, la classification d'images à grande échelle [43].

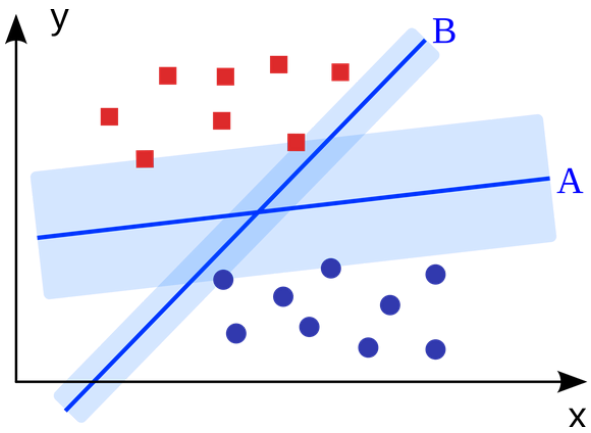

Figure 7: Support Vector Machine [43].

D. **Arbres de décision :** Un arbre de décision est un outil d'aide à la décision qui utilise un graphique ou un modèle arborant des décisions et leurs conséquences possibles, y compris les conséquences d'un événement fortuit, le coût des ressources et l'utilité [43].

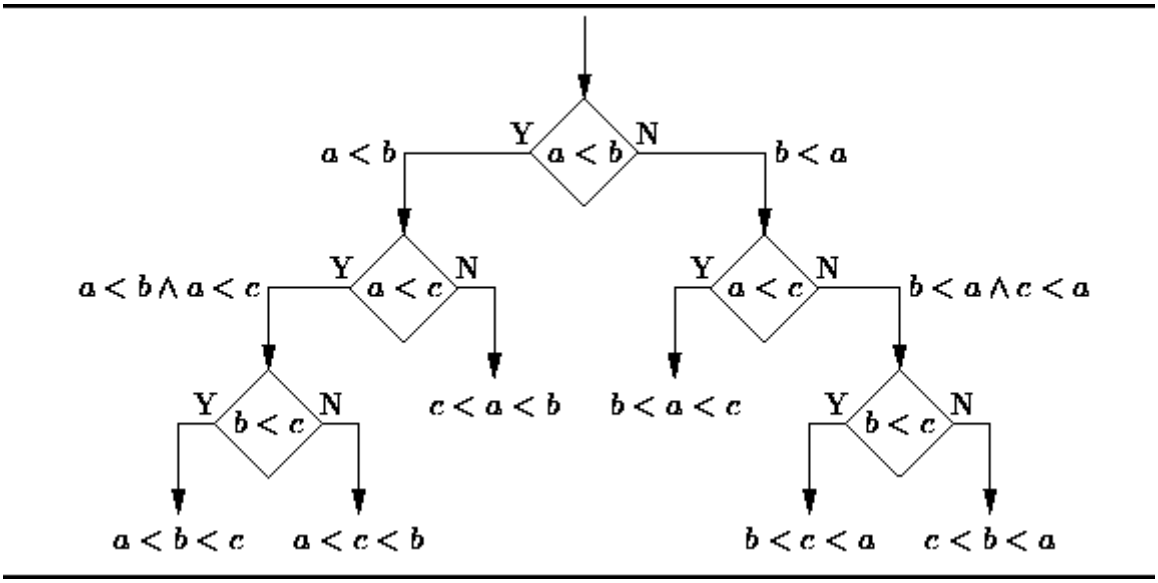

Figure 8: Graphe de l'Arbre de décision [43].

Du point de vue de la décision de gestion, un arbre de décision représente le nombre minimum de questions oui / non à poser pour évaluer la probabilité de prendre une décision correcte, la plupart du temps. En tant que méthode, elle vous permet d'aborder le problème de manière structurée et systématique pour arriver à une conclusion logique [43].

E. K-plus proches voisins (kppv) :

La méthode des plus proches voisins (noté parfois k-PPV ou k-NN pour k-Nearest-Neighbor) consiste à déterminer pour chaque nouvel individu que l'on veut classer, la liste des plus proches voisins parmi les individus déjà classés. L'individu est affecté à la classe qui contient le plus d'individus parmi ces plus proches voisins. Cette méthode nécessite de choisir une distance, la plus classique est la distance euclidienne, et le nombre de voisins à prendre en compte.

Cette méthode supervisée et non-paramétrique est souvent performante. De plus, son apprentissage est assez simple, car il est de type apprentissage par coeur. Cependant, le temps de prédiction est très long, car il nécessite le calcul de la distance avec tous les exemples, mais il existe des heuristiques pour réduire le nombre d'exemples à prendre en compte [33]

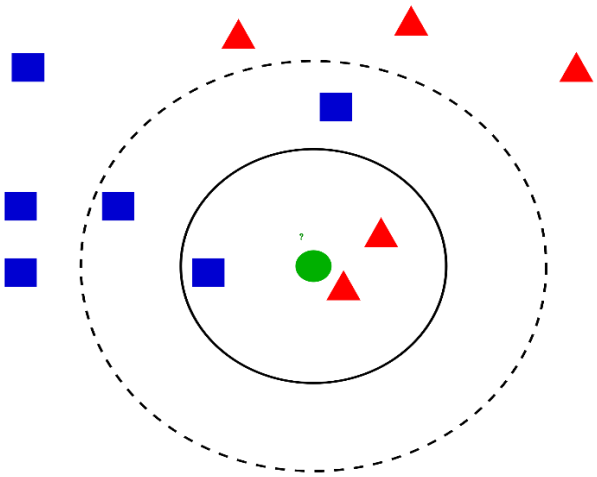

Figure 9: Exemple de classification kNN (k=3 et k=5)

F. Naïve Bayésien

Naïve Bayésien [54] est un algorithme populaire en Apprentissage automatique. Cet algorithme est utilisé pour la classification, il est particulièrement utile pour les problématiques de classification de texte.

Ce dernier est très rapide pour la classification en effet les calculs de probabilités ne sont pas très coûteux et la classification est possible avec un petit jeu de données.

Le terme P(A|B) se lit : la probabilité que l'événement ***A*** se réalise sachant que l'événement ***B*** s'est déjà réalisé.

On appelle ***le terme A*** : l'évidence. Le terme ***B*** s'appelle Outcome.

G. Artificial Neural Network (ANN)

Un réseau de neurones artificiels (RNA) [58] est un modèle informatique basé sur la structure et les fonctions des réseaux de neurones biologiques. Les informations qui circulent à travers le réseau affectent la structure de l'ANN, car un réseau neuronal change (ou apprend, dans un sens) en fonction de ces entrées et sorties.

Les ANN sont considérés comme des outils de modélisation de données statistiques non linéaires dans lesquels les relations complexes entre les entrées et les sorties sont modélisées ou des modèles sont trouvés.

2.4 Apprentissage non supervisé

Dans l'apprentissage non supervisé, les données sont uniquement constituées d'entrées. Dans ce cas, les tâches à réaliser diffèrent de l'apprentissage supervisé. Bien que de manière plus implicite, ces tâches sont également effectuées par les humains [29].

2.4.1 Regroupement ou « clustering »

Le regroupement est l'équivalent non supervisé de la classification. Comme son nom l'indique, son but est de regrouper les données en classes en utilisant leurs similarités. La difficulté du regroupement réside dans l'absence de mesure générale de similarité. Celle-là doit donc être définie en fonction du problème à traiter. L'un des algorithmes de regroupement les plus couramment utilisés est l'algorithme des k-moyennes. Le regroupement consiste donc à apprendre une fonction fregrp de Rd dans N qui associe à un vecteur son groupe. Contrairement à la classification, le nombre de groupes n'est pas connu a priori [29].

2.4.2 Estimation de densité

Le but de l'estimation de densité est d'inférer la répartition des données dans l'espace des entrées (ou, plus formellement, leur distribution). L'estimation de densité consiste donc à apprendre une fonction fest-dens telle que : ∫x fest-dens=1 qui associe à un vecteur sa probabilité [29].

2.4.3 Algorithmes de clustering :

Le clustering consiste à grouper un ensemble d'objets de sorte que les objets du même groupe (cluster) se ressemblent davantage que ceux d'autres groupes.

Chaque algorithme de clustering est différent, et en voici quelques-uns :

- Centroid-basedalgorithms.
- Algorithmes basés sur la connectivité.
- Algorithmes basés sur la densité.
- Probabiliste.
- Réduction de la dimensionnalité.
- Réseaux de neurones / Deep Learning.

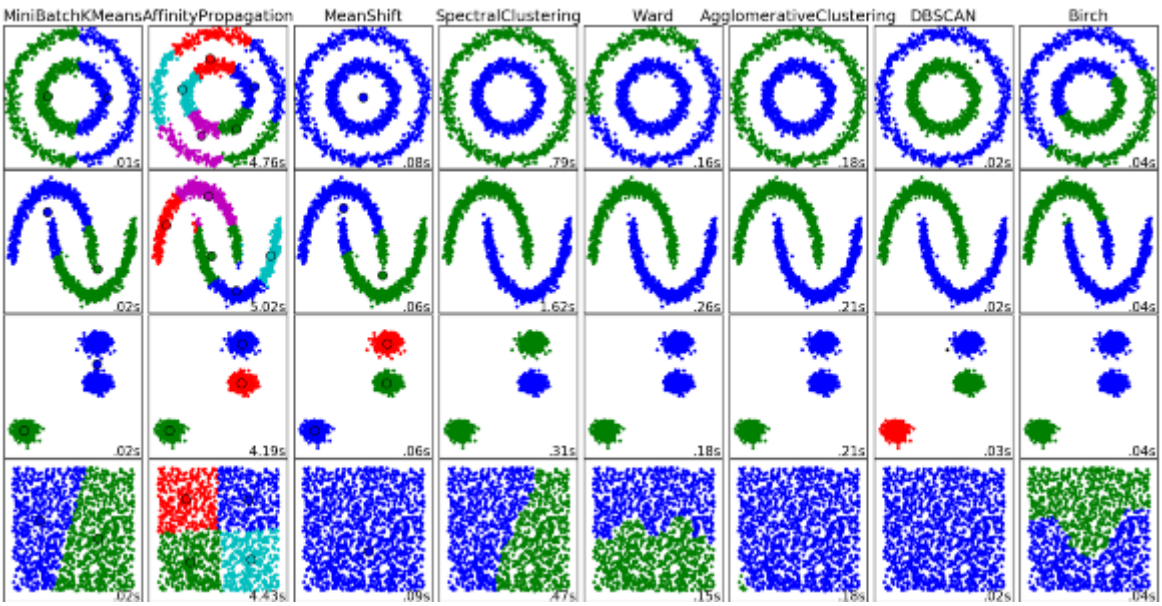

Figure 10: Clustering Algorithmes [43].

2.5 Autres travaux

Dans cette partie nous allons citer des auteurs et leurs travaux qui ont mélangée entre plusieurs algorithmes afin de parvenir à une meilleure détection d'intrusion.

Parmi eux [59] proposent un NIDS basé sur une méthode de sélection de caractéristiques appelée RGC (Récursive Feature Addition) et technique de bigram. Le système a été conçu, mis en œuvre et testé .

Le modèle a été testé sur l'ensemble de données ISCX 2012, qui est l'un des ensembles de données les plus connus et les plus récents pour la détection d'intrusion.

De plus, une technique de *bigram* est proposée pour coder les fonctions de chaîne de charge utile en une représentation utile qui peut être utilisée dans la sélection de caractéristiques.

En outre, ils ont proposé une nouvelle métrique d'évaluation appelée (combinée) qui combine la précision, le taux de détection et le taux de fausses alarmes d'une manière qui aide à comparer différents systèmes et à sélectionner les meilleurs d'entre eux. Le système basé sur la sélection des fonctionnalités a montré une amélioration notable de la performance en utilisant différentes métriques.

Kummar [60] utilise les réseaux de pétri pour représenter les scénarios d'attaques. Les transitions sont étiquetées par des appels systèmes alors que les jetons évoluent chaque fois qu'un événement permet de tirer une transition. Un jeton possède une couleur qui est une évaluation des variables.

Les avantages des réseaux de pétri est que même les signatures complexe peuvent être facilement écrites néanmoins il est couteux en terme calcul et alourdit le processus.

La spécification des exécutions normales des processus constitue une autre technique de la détection d'intrusions comportementale.

Fink et Levittet [61] s'intéressent aux programmes avec privilège "super utilisateur" et développent un langage de spécification qui se base sur la logique des prédicats et les expressions régulières. Cependant la réalisation de ces spécifications est une tâche assez difficile. Les auteurs proposent une méthode utilisant une logique inductive pour synthétiser directement les spécifications à partir des traces valides [C. Ko et al 2000].

Dans ce contexte, Nuansri [N. Nuansri et al 1999] construit un diagramme de transition d'états pour représenter les changements de privilège. Il définit ainsi un ensemble de règles dont la violation révèle la présence d'une attaque. **[62**]

2.6 Conclusion

Nous avons présenté dans ce chapitres un état de l'art présentant différentes techniques qui dérive généralement de l'IA qui est utilisé en vue d'une détection d'intrusions. Nous avons abordé principalement les algorithmes d’apprentissage automatique dont nous les utiliserons dans notre solution décrite dans le prochain chapitre.

CHAPITRE 3

CONCEPTION ET MISE EN OEUVRE

3.1 Introduction :

Pour cela nous avons montré dans ce chapitre les différentes phases du développement de notre application, en commençant par une description de la base de données KDD et les étapes de prétraitement que nous avons fait sur cette dernière, ensuite nous présentons notre modèle de classification suivi par la discussion des résultats obtenus.

3.2 Architecture du système :

Notre système est un NIDS dont il analyse les paquets entrants et il utilise les techniques de data mining pour les classifier les paquets comme attaque ou normal ainsi il permet d'identifier le type d'attaque.Notre système est un système :

- Multi niveau : il a la capacité de classifier les paquets de données suivant plusieurs niveaux. Le premier niveau classifie les paquets en deux types attaque ou normal. Le deuxième niveau peut identifier quatre classes d'attaques. Le troisième identifie le type de chaque attaque.
- Hybrid : car il se repose dans ce mécanisme sur plusieurs algorithmes d'apprentissage automatique.

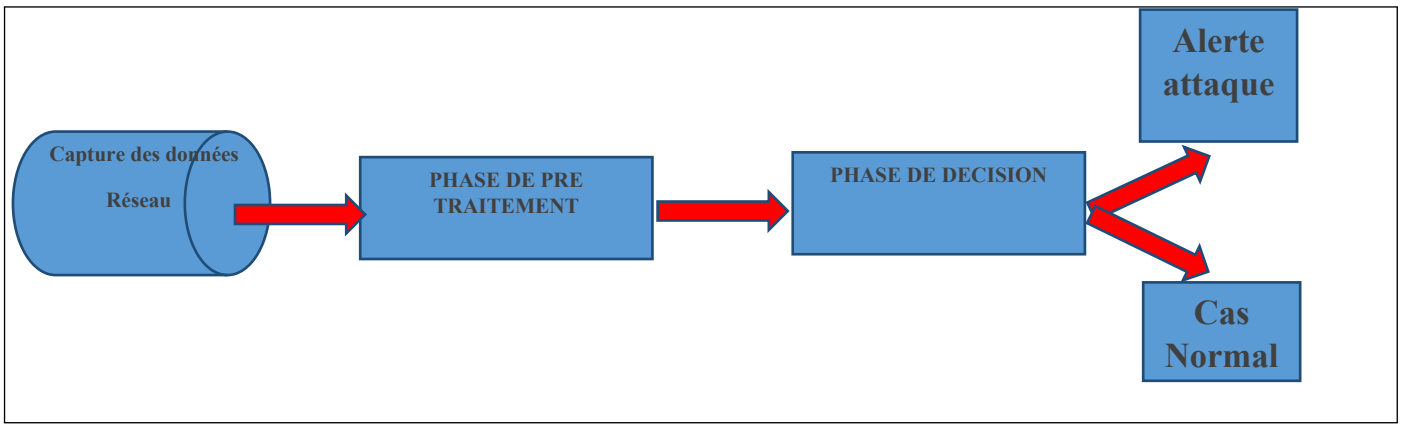

Figure 11: Architecture de la détection des attaques

La figure montre l'architecture du système qui se compose de trois phases importantes pour la détection des attaques :

1. **La capture des données**

D’abord le système doit récupérer les données et stoker afin d’être traitées dans la deuxième phase

2. La phase de prétraitement

Les Données de KDD sont de trois types: numérique, Nominale et binaire. Avant de passer au travail expérimental, l'ensemble de données KDD est d'abord passé par une opération de prétraitement des données et la conversion du type des attributs en suivant les étapes décrites dans ce qui suit :

- ✓ **La numérisation comme** nous l’avons dit précédemment la base de données KDD contient des d données nominales. Sachant que les modèles de réseaux de neurones n’acceptent que des attributs numériques. Nous avons converti les ces attributs vers des données numériques.
- ✓ **Normalisation**:

  Les valeurs obtenues après l’opération de la numérisation sont très variées et constituent un grand intervalle, Certains attributs prennent de grandes valeurs (src_bytes, dst_bytes, etc.), alors que d’autres ne prennent que des petites valeurs (serror_rate, same_srvrate, etc.), et cela peut nuire à la rentabilité du modèle de détection d’intrusions. Afin d'éviter ce problème et garantir l’efficacité du modèle généré,les valeurs de la base données doivent être ajustées ou normalisées ; dans notre cas les données de la base KDD sont normalisés dans l’intervalle de [0, 1] en se basant sur une fonction de transfert. Nous avons utilisé la fonction MinMax décrite par la formule suivante : **[63]**

$$nouvval=\left(\frac{\text{ancval - Minanc}}{\text{Maxanc - Minanc}}\right)\times (Max_{nouv}-Min_{nouv})+Min_{nouv}$$

Où :

- val_{anc}: est la valeur à normaliser.
- val_{nouv} : est la valeur après la normalisation.
- Min_{anc}: est la limite inférieure de l’intervalle à qui val_{anc} appartient.

3. La phase de décision

Le module de décision comporte deux phases de fonctionnement. La phase d'apprentissage et de détection.

- ✓ La phase d'apprentissage Dans la phase d'apprentissage, le classificateur utilise les profils d'utilisateur réseau capturés prétraités comme modèles d'apprentissage d'entrée. Cette phase se poursuit jusqu'à ce qu'un taux de classement correct satisfaisant soit obtenu.
- ✓ La phase de détection Une fois le classificateur appris, sa capacité de généralisation identifie correctement les différents types d'utilisateurs.

  Ce processus de détection peut être considéré comme une classification des modèles d'entrée en normal ou en attaque.

  Alors la responsabilité fondamentale du module de décision est de transmettre une alerte à l'administrateur du système pour l'informer d'une attaque à venir. Cela donne à l'administrateur système la possibilité de surveiller la progression du module de détection**.**

3.3 Mesures de performance

Pour évaluer notre système, nous avons utilisé deux principaux indices de performance. Nous calculons le taux de détection et le taux de fausses alarmes selon [64] les hypothèses suivantes :

Faux positif (FP): le nombre total d'enregistrements normaux qui sont classés comme anormaux

Faux négatif (FN): le nombre total d'enregistrements anormaux qui sont classés comme normaux

Total Normal (TN): le nombre total d'enregistrements normaux

Total Attack (TA): le nombre total d'enregistrements d'attaque

Taux de détection = [(TA-FN) / TA] * 100

Taux de fausses alarmes = [FP / TN] * 100

Taux de classification correct = Nombre d'enregistrements correctement classés / Nombre total d'enregistrements dans l'ensemble de données utilisé

3.4Mise en œuvre :

Dans l'implémentation, plusieurs algorithmes de reconnaissance de formes et d'apprentissage automatique ont été testés sur l'ensemble de données. Ces algorithmes sélectionnés à partir du domaine des réseaux de neurones et des arbres de décision. :**Réseaux de neurones , Perceptron multicouche (MLP), Arbre de décision .**

3.5Jeu de données utilisé :

NSL-KDD est un ensemble de données suggéré pour résoudre certains des problèmes inhérents à l'ensemble de données KDD'99 . Bien que cette nouvelle version de l'ensemble de données KDD souffre encore de certains des problèmes discutés par McHugh et peut ne pas être un parfait représentant des réseaux réels existants, en raison de l'absence d'ensembles de données publiques pour les IDS basés sur le réseau, nous croyons qu'il peut encore être appliqué comme un ensemble de données de référence efficace pour aider les chercheurs à comparer les différentes méthodes de détection des intrusions.De plus, le nombre de dossiers dans le train et les ensembles d'essais NSL-KDD est raisonnable. Cet avantage rend abordable d'exécuter les expériences sur l'ensemble complet sans avoir besoin de sélectionner au hasard une petite partie. Par conséquent, les résultats de l'évaluation des différents travaux de recherche seront cohérents et comparables. [65]

L'ensemble de données NSL-KDD présente les avantages suivants par rapport à l'ensemble de données KDD d'origine :

- ✓ Il n'inclut pas les enregistrements redondants dans l'ensemble du train, de sorte que les classificateurs ne seront pas biaisés vers des enregistrements plus fréquents.
- ✓ Il n'y a pas de doublons dans les ensembles de tests proposés ; par conséquent, les performances des apprenants ne sont pas biaisées par les méthodes qui ont de meilleurs taux de détection sur les dossiers fréquents.
- ✓ Le nombre d'enregistrements sélectionnés de chaque groupe de niveau de difficulté est inversement proportionnel au pourcentage d'enregistrements dans l'ensemble de données KDD d'origine. Par conséquent, les taux de classification des méthodes distinctes d'apprentissage automatique varient dans un plus large éventail, ce qui rend plus efficace d'avoir une évaluation

En se basant toujours sur le site officiel, On vous montre les statistiques suivantes :

- ✓ **Statistiques des enregistrements redondants dans l'ensemble de trains KDD :**

Enregistrements originaux | Enregistrements distincts | Taux de réduction

Attaques : 3 925 650 | 262 178 | 93.32%

Normal: 972 781 | 812 814 | 16.44%

Total : 4 898 431 | 1 074 992 | 78.05%

- ✓ **Statistiques des enregistrements redondants dans l'ensemble de tests KDD**

Enregistrements originaux | Enregistrements distincts | Taux de réduction

Attaques : 250 436 | 29 378 | 88.26%

Normal: 60,591 | 47,911 | 20.92%

Total : 311 027 | 77 289 | 75.15%

En outre, les attaques trouvées sont classées selon quatre catégories principales comme la montre la figure 19 :

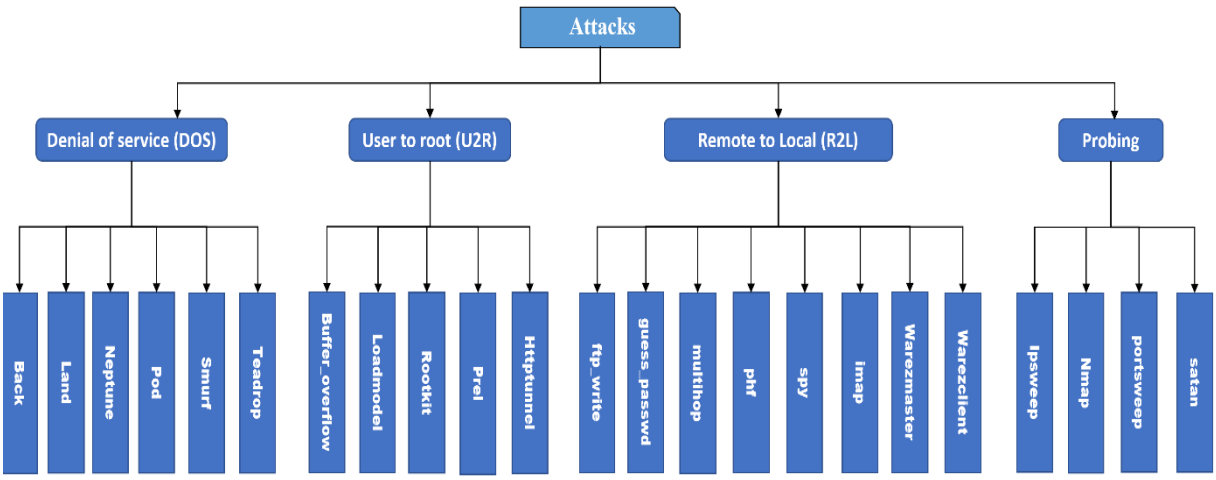

Figure 19: les catégories et les types d'attaques

- **La catégorie DOS (Denial Of Service)** : provoque un déni de service via des requêtes d'écho ICMP, manipulées à une adresse de diffusion d'un réseau.
- **U2R (User to Rootattacks*) :***l'attaquant essaie d'avoir les droits d'accès au système par le biais d'un poste.
- **R2L (Remote to Local access)** : Ce type d'attaque essaie d'exploiter la vulnérabilité du système afin de contrôler la machine distante.
- **Probe(sondage et surveillance)** *:* Ces actions ne sont pas vraiment des attaques puisqu'elles ne sont pas "destructrices" elles n'empêchent pas une entité de

fonctionner correctement, mais permettent d'acquérir des informations parfois cruciales pour mener une attaque de plus grande envergure plus tard.

Attribut	Désignation	Description
A1	La durée	Longueur (nombre de secondes) de la connexion
A2	Type de protocol	Type du protocole, par exemple tcp, udp, etc.
A3	Service	Service réseau sur la destination, par exemple, http, telnet, etc.
A4	Src_bytes	Nombre d'octets de données de la source à la destination
A5	Dst_bytes	Nombre d'octets de données de la destination à la source
A6	Drapeau	État normal ou d'erreur de la connexion
A7	La terre	1 si la connexion est de / vers le même hôte / port ; 0 sinon
A8	Incorrect_Fragment	Nombre de "mauvais" fragments
A9	Urgent	Nombre de colisurgents
A10	Chaud	Nombre d'indicateurs "chauds"
A11	num_failed_logins	Nombre de tentatives de connexion infructueuses
A12	log_in	1 si connecté avec succès ; 0 sinon
A13	num_compromised	Nombre de conditions "compromises"
A14	root_shell	1 si la racine est obtenue ; 0 sinon

Ces données sont décrites au moyen de différentes variables explicatives. Pour une meilleure compréhension, celles-ci ont été classifiées en cinq catégories.

Les variables d'une même machine décrivent seulement les connexions faites durant les deux dernières secondes et ayant le même destinataire que la connexion courante. Les variables de même service décrivent seulement les connexions faites durant les deux dernières secondes et ayant le même service que la connexion courante. On trouve aussi des variables de connections TCP individuelles.

Enfin, il existe des variables qui indiquent un comportement anormal dans les données, ainsi le nombre de tentatives d'ouverture échouées. Il s'agit des variables de contenu.

3.6 Attributs utilisés

Chaque 'connexion' est caractérisée par 41 attributs tels que sa durée, le type du protocole, etc. Ces attributs ont été fixés suite à un travail de fouille de données dans NSL KDD est considérée comme étant une 'connexion' normale ou bien une attaque.

Les attributs caractérisant chaque 'connexion', sont détaillés dans le tableau 3. Certains attributs sont de type discret (admettant un nombre fini de valeurs), d'autres sont de type continu.

Tableau 2: Les attributs du jeu de données NSLKDD [55].

A15	su_attentée	1 si la commande `` su root '' a été tentée ; 0 sinon
A16	num_root	Nombred'accès `` root ''
A17	num_file_creations	Nombre d'opérations de création de fichier
A18	num_shells	Nombred'invites de Shell
A19	num_access_files	Nombre d'opérations sur les fichiers de contrôle d'accès
A20	num_outbound_cmds	Nombre de commandes sortantes dans une session ftp
A21	is_hot_login	1 si le login appartient à la liste `` chaude ''; 0 sinon
A22	is_guest_login	1 si le login est un "invité" ; 0 sinon
A23	Compter	Nombre de connexions au même hôte que la connexion actuelle au cours des deux dernières secondes
A24	serror_rate	% de connexions comportant des erreurs `` SYN ''
A25	reerror_rate	% de connexions comportant des erreurs `` REJ ''
A26	same_srv_rate	% de connexions au même service
A27	diff_srv_rate	% de connexions à différents services
A28	srv_count	Nombre de connexions au même service que la connexion actuelle au cours des deux dernières secondes
A29	srv_serror_rate	% de connexions comportant des erreurs `` SYN ''
A30	srv_rerror_rate	% de connexions à comportant des erreurs "REJ"
A31	srv_diff_host_rate	% de connexions à différents hôtes
A32	dst_host_count	Nombre de connexions pour le même hôte
A33	dst_host_svr_count	Nombre de connexions pour le même hôte utilisant le même service
A34	dst_host_same_srv_rate	% de connexions pour le même hôte utilisant le même service
A35	dst_host_diff_srv_rate	% de connexions pour le même hôte utilisant le différent service
A36	dst_host_same_src_port_rate	% de connexions pour le même hôte ayant port src
A37	dst_host_srv_diff_host_rate	% de connexions pour le même hôte et le même service utilisant différents hôtes
A38	dst_host_serror_rate	% de connexions pour le même hôte ayant l'erreur "SYN"
A39	dst_host_srv_serror_rate	% de connexions pour le même hôte et le même service ayant l'erreur "SYN"
A40	dst_host_rerror_rate	% de connexions pour le même hôte ayant l'erreur "REJ"
A41	dst_host_srv_rerror_rate	% de connexion pour le même hôte et le même service ayant l'erreur "REJ"

3.7Description du code utilisé :

Dans cette section, nous allons expliquer le code utilisé en détaillons les différentes bibliothèques, fonction, ….

3.7.1 Apprentissage de l'algorithme :

La figure suivante montre la méthode d'importer les algorithmes de classification utilisés par notre proposition :

```
import numpy as np
import sklearn
from sklearn.decomposition import PCA
from sklearn.feature_selection import SelectKBest, chi2
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from sklearn import tree, linear_model
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.model_selection import KFold
from sklearn.neighbors import NearestNeighbors
from sklearn import svm
from sklearn.ensemble import BaggingClassifier, AdaBoostClassifier, RandomForestClassifier, VotingClassifier, \
    GradientBoostingClassifier
```

3.7.2 Chargement du Data Set :

D'abord, Avant de commencer nos test, il faut charger nos données dans le fichier test .phyhon propose la fonction read-table() qui va nous envoyer un Data frame contient les 42 attributs :

```
bankdata0 = pd.read_table("C:\\Users\\PC\\Desktop\\....csv")
```

3.7.3Afiichage des calculs

```
print("-----------------------------------------")
print(">>>>> Score Total >>>>>")
print()
print("Dos Accuracy: ",toc1*100 ,"% ,Total: (",oc1,"/",i1,"), Wrong = ",i1-oc1)
print("U2R Accuracy: ",toc2*100,"% ,Total: (",oc2,"/",i2,"), Wrong = ",i2-oc2)
print("Probe Accuracy: ",toc3*100,"% ,Total: (",oc3,"/",i3,"), Wrong = ",i3-oc3)
print("R2L Accuracy: ",toc4*100,"% ,Total: (",oc4,"/",i4,"), Wrong = ",i4-oc4)
print("Normal Accuracy: ",toc5*100,"% ,Total: (",oc5,"/",i5,"), Wrong = ",i5-oc5)
print()
print("Total Accuracy: ",accuracy_score(y_test24, Big_Predict_24)*100,"%")
print('Number of wrong prediction: ',len(Wrong_Predict_24))
```

3.8Conclusion

Nous avons présenté dans ce chapitre une solution pour améliorer ladétection des intrusions en se basant sur les algorithmes de machines learning :k-NN, DecisionTree, Random Forest, Bagging, Boosting, Stacking et VotingClassifier..Nous avons utilisé le jeu de données NSL KDD car il s'agit du meilleur jeu de données par rapport à KDD99.

Conclusion Générale

Parmi les différents outils de sécurité informatique, on trouve le system de détection d'intrusion. Cet outil est devenu très indispensable pour tout réseau informatique, il nous permet de connaitre toutes activités anormales qui peuvent présenter un danger pour notre réseau.

Les systèmes de détection d'intrusions ont deux types. Le premier type est basé sur signature, ce type a montré beaucoup de limites avec la rapidité et l'augmentation du trafic réseau mais il ne peut pas détecter les nouvelles formes des attaques. Le deuxième type basé sur les anomalies a été proposé afin de traiter les problèmes du premier type, ou les techniques d'apprentissage automatique ont été utilisés. Malgré la puissance et l'efficacité des techniques de l'apprentissage automatique, les systèmes de détection d'intrusion de ce dernier souffrent de certaines limites comme la nécessité de

faire une mise à jour régulière, la nécessité de préparer les données d'entrainement, la difficulté de détecter les nouvelles formes d'attaques etc.

Dans notre étude nous avons proposé une approche qui se repose sur l'apprentissage automatique, cette propose est basée sur plusieurs niveaux de détection d'intrusions en utilisant plusieurs algorithmes d'apprentissage dont le but d'améliorer la performance et la précision des IDs.

Le travail consiste à sélectionner où bien classer la catégorie d'attaque avant de classer son type a l'aide du Modèle de sélection, qui a été implémenté par le k-NN, DecisionTree, Random Forest, Bagging, Boosting, Stacking et Voting Classifier. Concernant la deuxième étape, elle contient quatre modèles nommés Classificateur de Type, chaque catégorie (DOS, Probe, U2R et R2L) a son Classificateur de Types spécifique qui est offert par sa spécialisation avec une précision élevée, ces classificateurs utilisent Random Forest.

Références

[1] J. Kim. Integrating Artificial Immune Algorithms for Intrusion Detection.

[2] InfoSec Reading Room. Intrusion detection systems: definition, need and challenges. SANS Institut, 2001.

[3] Slimane TAJNI. Mise en place d'une sonde SNORT monitoring network 2006-01-03.

[4] Ghenima BOURKACHE. Un IDS réparti basé sur une société d'agents intelligents, mémoire de magister informatique, université de Boumerdes. Algérie 2007.

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

[6] Dorothy Denning. An intrusion detectionmodels. IEEE, transaction on software engineering, 13(2) :222–232, 1987.

[7] Stephen E. Smaha. Haystack: An intrusion detection system.21th National Computer Security Conference,. :37–44, 1988.

[8] Mary E. Hanna Michael M. Sebring, Eric Shellhouse and R. Alan Whitehurst. Experts systems in intrusion detection: A case study. Proceeding of the 11th National Computer Security Conference, Baltimore, Maryland, . :74–81, 1988.

[9] TRW Defense System Groupe. Intrusion detection expert system feasibility study. Technical report, Final report 46761, 1986.

[10] Yiming Yang. An evaluation of statistical approaches to text categorization. Journal of Information Retrieval, 1 :67–88, 1999.

[11] H S Vaccaro and G E Liepins. Detection of anomalous computer session activity. Proceeding of the 1989 IEEE Symposium on Security and Privacy. Oakland, California, . :280–289, 1-3 May 1989.

[12] Stefan Axelsson. Intrusion detection systems: A survey and taxonomy. Departement of Computer Engineering Chalmers University of Technology, Goteborg, Sweden. :15–23, 14 March 2000.

[13] Karl Levitt Biswanath Mukherjee Jeff Wood Todd Heberlein, Gihan Dias and David Wolber. A network security monitor. Proceeding of the 1990 IEEE Symposium on Research in Security and Privacy. Soc Press, Los Alamitos, CA: USA,1990.

[14] L Todd HeberleinBiswanath Mukherjee and Karl Levitt. Network intrusion detection. IEEE Network, 8(3) :26-41, May/June 1994.

[15] Monique Becker Hervé Debar and Didier Siboni. A neural network component for an intrusion detection system. Proceeding of the 1992 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, USA,. :240–25L0, May 1992.

[16] Daniel M Teal Steven R Snapp, Stephen E. Smaha and Tim Garance. The dids (distributed intrusion detection system) prototype. Proceeding of the Summer USENIX Conference, San Antonio, Texas. :227–233, 8-12 June 1992.

[17] Sandeep Kumar and Eugene H. Spafford. A software architecture to support misuse intrusion detection. Technical report, The COAST Project, Department of Computer Sciences, Purdu University, West Lafayette, IN, 47907-1398, USA, 17March 1995.

[18] Sandeep kumar. classification and detection of computer Intrusions. PhD thesis, Purdu University, West Lafayette, Indiana, USA, August 1995.

[19] Sandeep kumar and Eugene H. Spafford. A pattern matching model of misuse intrusion detection. Proceeding of the 17th National Computer Security Conference, Baltimore MD, USA, .:11-21-1994.

[20] Sandeep kumar and Eugene H. Spafford. An application of pattern matching in intrusion detection. technical report csd-tr-94-013, the coast project. Technical report, dept of computer Sciences, Purdue University, West Lafatette, IN, USA, June 1994.

[21] Todd Ellis Ivan Krsul Mark Ceosbie, Bryn Dole and Eugene Spafford. Idiot user guide. Technical report The COAST Project, Dept of Computer Science, Purdu University, West Lafayette, IN, USA, 4 September 1996.

[22] T. Frivold D. Anderson and A. Valdes. Next-generation intrusion-detection expert system (nides). Technical report, Computer Science Laboratory, SRI International, Menlo Park, CA 94025-3493, USA, May 1995.

[23] Harold Javitz Ann Tamaru Debra Anderson, Teresa F. Lunt and Alfonso Valdes. Detecting unusual programgram behavior using the statistical component of the next-generation intrusion detection system (nides). Technical report, Computer Science Laboratory, SRI International, Menlo Park, CA, USA, May 1995.

[24] R. Crawford M. Dilger J. Frank J. Hoagland K Levitt C. Wee R. Yip S. StaniFord Chen, S. Cheung and D. Zerkle. Grids-a graph-based intrusion detection system for large networks. Proceeding of the 19th National Information Systems Security Conference, 1996.

[25] Philip A Porras and Peter G Neumann. Emerald: Event monitoring enabling responses to anomalous live disturbances. Proceeding of the 20th National Information Systems Security Conference, Baltimore, Maryland, USA, . :353–365,7-10 October 1997.

[26] Philip A Porras and Alfonso Valdes. Live traffic analisys of tcp/ip gateways. Proceeding of the 1998 ISOC Symposium on Network and Distributed System Security, San Diego: California, 11-13 March 1998.

[27] Jajodia S.-Popyack L. Barbara D., Julia Couto J. and Wu N. Adam: detecting intrusions by data mining. Proceedings of the 2001 IEEE workshop on information assurance and security, NY, USA :310 – 18, 2001.

[28] Mining audit data to build intrusion detection models. wenke lee and salvatore j. stolfo and kui w. mok. 1998.

[29] KenazaTayeb.Détection d'intrusion coopérative basée sur la fusion de données. Master'sthesis, Institut National de formation en Informatique / ALGER,2006.

[30] Tadeusz Pietrazek. Alert classification to reduce false positives in intrusion detection. PhD thesis, Albert-Ludwigs-University of Freibury, Germany,2006.

[31] Nist. icatmetabase. web page at http://icat.nist.gov/20002004.

[32] Thomas H. Ptacek and Timothy N Newsham. Insertion, evasion and denial of service: eluding network intrusion detection. Technical report, Secure Networks Inc, 1998.

[33] Umesh Shankar and Vern Paxson. Active mapping: Resisting nids evasion without altering traffic. In Proceedings of the 2003 IEEE Symposium on Security and Privacy, Oakland, CA: 4462, 2001.

[34] Robin Sommer and Vern Paxson. Enhancing byte-level network intrusion detection signatures with context. In Proceeding of the 10th ACM Conference on Computer and Communication Security, Washington, DC :262-271,2003.

[35] Seth Webster Richard Lippmann and Dauglas Stetson. The effect of identifying vulnerabilities and patching software on the utility of network intrusion detection. In Recent Advances in Intrusion Detection (RAID2002), Spring-Verlag, 2516 LNCS: 307-326, 2002.

[36] C. Kruegel F. Valeur, G. Vigna and R. Kemmmerer. Acomprehensiveapproch to Dependable and Secure Computing, 1(3) :146-169,2004.

[37] Julisch K. Using Root Cause Analysis to handle Intrusion Detection Alarms. PhD thesis, University of Dortmund, Germany, 2003.

[38] Wenke Lee and Salvator J. Stolfo. Data mining approaches for intrusion detection. Proceeding of the 7th USENIX Security Symposium, san Antonio, 1998.

[39] Schwartzbard A. Ghosh A. A study in using neural networks for anomaly and misuse detection. In the eighth USENIX security symposium, Washington, USA.14151,1999.

[40] Janoski G.H. Mukkamala S. Intrusion detection: support vector machines and neural networks In the IEEE international joint conference on neural networks, Honolulu USA.,2002.

[41] M^elle MAHDJANE Karima. Détection d'anomalies sur des données biologiques par SVM. Université Mouloud Mammeri de Tizi Ouzou, 14 Octobre 2012.

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

[43] James Le, Machine Learning Engineer, The 10 Machine Algorithms. Learning Engineers Need to Know.Technical report,https://www.kdnuggets.com/2016/08/10-algorithms-machine-learning-engineers.html, August,2019.

[44] Carla Sauvanaud. Monitoring et Détection d'anomalie par Apprentissage dans des infrastructure virtualisées. l'Institut National des Sciences Appliquées de Toulouse, 2016.

[45] Heutte, L. Combinasion de Classificateur : pourquoi et comment les combiner ?.université de Rouen .2005

[46] breiman, L. Bagging Predictors. Machine Learning Journal.Vol.24, No.2. pp.123-140.1996.

[47] Freund, Y.et Schapire, R.E. Experiments with a new boosting algorithm. The 13th International Conference on Machine Learning.pp.148-156.1996.

[48] Wolpert, D.H. Stacked generalization. Neural Networks, Pergamon Press. Vol. 5. No 2. pp.241-259.1992.

[49] Pellerin, E. Méta-apprentissage des algorithmes génétique. Thèse d'exigence partielle de la maitrise en mathématique et informatique appliquées, Université du Québec. Décembre 2005.

[50] Larkey, L.S. et Croft W.B. Combining classifiers in text categorization. The 19th annual international ACM SIGIR conference on Research and development in information retrieval, Zurich, Suisse. PP. 289-297.1996.

[51] Merz, C.J. Using correspondence analysis to combine classifiers. Machine Learning. Vol. 36, No 1-2. pp. 33-58.1999.

[52] Opitz, D. et Maclin, R. Popular ensemble méthode: an empirical study. Journal of AI Research. Vol. 11. pp. 169-198. 1999.

[53] Atilla.Özgür, Hamit. Erdem. A Review of KDD99 Dataset Usage in Intrusion Detection and Machine Learning between 2010 and 2015. Baskent University. 14 Apr 2016.

[54] YounesBenzaki, website Mr.Mint, **Naive Bayes Classifier.**https://mrmint.fr/naive-bayes-classifier, 26 Juillet2017.

[55] University of California Irvine, KDD 99 Cup 1999 Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, October 28 2019

[56] H. Yao, X. Sun, Z. Zhou, L. Tang, and L. Shi, Joint optimization of sub channel selection and spectrum sensing time for multiband cognitive radio networks, in International Symposium on Communications & Information Technologies, 2010.

[57] Steven R. Snapp, Stephen E. Smaha. Haystack Laboratories, Daniel M. Tim Grance. The DIDS (Distributed Intrusion Detection System) Prototype, United States, June 8 1992

[58] Carlos Gershenson, Artificial Neural Networks for Beginners, Universidad Nacional Autónoma de México, September 2003.

[59] Hamed, T., Ernst, J. B., & Kremer, S. C. (2018). A Survey and

Taxonomy of Classi_ers of Intrusion Detection Systems. In Computer and Network Security

Essentials (pp. 21-39). Springer, Cham.

[60] Kumar, S., & Spa_ord, E. H. (1994). A pattern matching model

for misuse intrusion detection.

[61] C. Kahn, D. Bolinger, and D. Schackenberg. Communication in the

common intrusion detection framework v 0.7. http ://www.isi.edu/brian/cidf/drafts/communication.txt,

Jun. 1998.

[62] Nuansri, N., Singh, S., & Dillon, T. S. (1999). A process statetransition analysis and its application to intrusion detection. In Computer Security Applications Conference, 1999.(ACSAC'99) Proceedings. 15th Annual (pp. 378-387). IEEE.

[63] M BOUROUH ; Détection d'intrusions à base des réseaux de neurones et algorithmes génétiques ;2017.

[64] .S.T. Sarasamma, Q.A. Zhu, and J. Huff, "Hierarchal Kohonenen Net for Anomaly Detection in Network Security," IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 35(2), 2005, pp. 302-312.

[65] https://www.unb.ca/cic/datasets/nsl.html consulter le 12/11/2020