République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université 20 Aout 1955-SKIKDA

Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études en vue de l'obtention du diplôme de Master en Informatique
Option : Réseaux et Systèmes Distribués (RSD)

Thème

Méthodes d'apprentissage en détection d'intrusion

Réalisé par :
Loucif Asma
Berrak Insaf

Encadré par :
Dr NAFIR ABDENACER
Dr MAZOUZI SMAINE

Année Universitaire
2021-2022

Remerciements

Avant tout de chose je tiens à remercier le grand "DIEU" de nous avoir données le courage et la volonté pour réaliser ce travail.

*Nos remerciements particuliers à MM.**Nafir Abdenacer et Mazouzi Smaine** d'avoir accepté d'encadrer ce projet. Merci pour vos précieux aides, encouragements et éclaircissements.*

Nous tenons à remercier les membres de jury pour nous avoir honorées d'accepter d'évaluer notre travail.

Nous tenons à remercier tous les professeurs qui nous

ont enseigné durant cette année

Merci à toutes les personnes qui ont contribué de près ou de loin à l'aboutissement de ce travail.

Merci

Dédicace

Je dédié ce mémoire à :

À mes parents qui ont partagés avec moi tous les moments d'émotion lors de la réalisation de ce travail.

Ma mère, qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il l'expression de mes sentiments et de mon éternelle gratitude.

Mon père, qui fier à moi et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit.

Merci pour les valeurs nobles, l'éducation et le soutient permanent venu de toi.

À mon frère « Khaled» ET mes sœurs « Mouna et Amina» pour leurs encouragements durant tout mon parcours.

À mes neveux « Louay,Nadjemeddine, Amir , Islam»

À toute ma famille Loucif et Bouras.

À toutes mes amies.

Àtous ceux qui m'ont aidé de loin ou de prés.

Loucif Asma

Dédicace

Que nulle dédicace ne puisse exprimer ce que je dois, pour votre bienveillance, votre affection et votre soutien …en ce jour solennel, je voudrais renouveler mon indéfectible attachement à mes très chers parents,

À celle qui a éclairé mon chemin et m'a aidé de tout mon temps, à ma mère, la bien-aimée démon cœur, qui m'a toujours poussé et motivé dans mes études. Puisse Dieu, le tout puissant, la repose et la mette au paradis

À mon père pour sa patience et encouragement.

À ma très chère sœur Iméne et à ses filles Tsabihe et Malak.

À mon petit frère Baha eddine.

À ma chère amie et binôme tout au long de mon parcours universitaire : Asma.

Àtoutes mes amies avec qui j'ai partagé les plus beaux et inoubliables moments de ma vie :Iméne, Aya, Sara et Chaima .

À Toute personne qui m'a aidé un jour à réussir jusqu'ici.

Berrak Insaf

Résumé

Malgré la multiplicité des outils de protections qui se font offrir sur le marché de sécurité, les systèmes informatiques, et plus particulièrement ces systèmes connectés sont soumis à d'intenses attaques, qui tentent dans leur majorité à prendre le contrôle des systèmes par diverses techniques ingénieuses d'intrusion.

Ce travail illustre l'utilisation de la méthode boosting avec quelques techniques d'apprentissage automatique à savoir le naiveBayes, et l'arbre de décision où nous avons comparé leurs précisions et leurs temps d'exécution afin d'obtenir le meilleur modèle pour les systèmes de détection d'intrusion.

Mots Clés : sécurité des réseaux informatiques, système de détection d'intrusion, apprentissage automatique, boosting.

Summary

Despite the multiplicity of protection tools offered on the security market, computer systems, and more particularly connected systems, are subject to intense attacks, which mostly attempt to take control of systems through various ingenious techniques. of intrusion.

This work illustrates the use of the boosting method with some machine learning techniques namely the naïve Bayes, and the decision tree and we compared their accuracies and their execution times in order to obtain the best model for the systems. intrusion detection.

Keywords: computer network security, intrusion detection system, machine learning, boosting.

ملخص

على الرغم من تعدد أدوات الحماية المتوفرة في سوق الأمن ،فإن أنظمة الكمبيوتر و الأنظمة المتصلة بشكل أكثر تحديدًا تتعرض لهجمات مكثفة ،والتي تحاول في الغالب السيطرة على الأنظمة من خلال تقنيات عبقرية مختلفة.

يوضح هذا العمل استخدام طريقة التعزيز مع بعض تقنيات التعلم الآلي وهي Bayes naive وشجرة القرار وقمنا بمقارنة دقتها وأوقات تنفيذها للحصول على أفضل نموذج لكشف التسلل

الكلمات الرئيسية: أمن شبكات الكمبيوتر، نظام كشف التسلل، التعلم الآلي، التعزيز

Sommaire

Chapitre2 : Machine learning

Chapitre 4 : Test et implémentation

La liste des Figures

La liste des Tableaux

Introduction générale

Introduction générale

Les réseaux et les systèmes informatiques sont devenus des outils indispensables au fonctionnement des entreprises. Ils sont aujourd'hui déployés dans tous les secteurs professionnels.

L'informatique gérée par ces systèmes fait l'objet de convoitises. Elle peut être exposée à des attaques qui exploitent des éléments vulnérables du système d'information. La détection des actions malveillantes est rapidement devenue une nécessité. Les mesures de prévention se sont révélées insuffisantes et ont amené la création de systèmes de détection d'intrusions (IDS : Intrusion Détection systèmes).

Une intrusion est définie comme étant toute tentative pouvant nuire à l'intégralité, la confidentialité ou la disponibilité dans le réseau ainsi que toute tentative visant à contourner les dispositifs de sécurité mis en place sur le réseau ou une machine. Ces tentatives d'intrusions peuvent être bénignes comme extrêmement dangereuses et préjudiciable pour l'entreprise.

Le domaine de la détection d'intrusion est encore jeune mais en plein développement.

Nous dénombrons à l'heure actuelle environ une centaine de systèmes de détections d'intrusions (ou IDS pour Intrusion Détection System), que ce soit des produits commerciaux ou du domaine public.Systèmes de surveillance du réseau sont devenus pratiquement indispensables dû à l'incessant accroissement en nombre et en dangerosité des attaques réseaux depuis quelques années.

Les techniques de détection d'intrusion peuvent être classées en 3 catégories : 1) les techniques basées signature, les techniques basées comportement, et 3) les techniques basées sur les algorithmes du machine-Learning,

Dans notre travail on s'intéresse par les différentes techniques d'apprentissage automatique À titre d'exemple : Arbres de décision, Bayes naïf ……

Le but de ce mémoire est l'étude de la performance de l'utilisation des algorithmes de construction de classifieurs d'ensemble pour la détection d'intrusion donc Nous avons choisis l'algorithme de construction d'un classifieur d'ensemble dit AdaBoost comparés aux classifieurs pris individuellement nous avons considéré deux classifieurs de bases, à savoir,

leJ48, le Naïve Bayes, L'objectif est de montrer quel est le classifieurs le mieux adapté à l'AdaBoost, et quel est le meilleur classifieurs entre ces deux classifieurs. En considérant un dataset KDD Cup99 et nous avons codé notre système en utilisant le langage phyton
Ce mémoire est organisé comme suit ;

Le premier chapitre est consacré à la présentation des différents aspects de la sécurité informatique et les systèmes de détection d'intrusion.
Le deuxième chapitre présente L'intelligence Artificielle et les méthodes d'apprentissage automatique utilisées pour la détection d'intrusion.
Le troisième chapitre consiste à présenter la recherche réalisée dans le cadre de ce projet, nous avons présentées les méthodes d'apprentissage supervisé et les différents algorithmes d'ensemble.
Le quatrième chapitre consiste à présenter les outils logiciels utilisés, les éléments d'implémentation, et quelques résultats expérimentaux.

Chapitre 1

Sécurité Informatique

Introduction :

Les architectures des nouvelles technologies de l'information et de télécommunication sont en train de se développer vers des systèmes basés sur des infrastructures facilement accessibles, ouverts à une large gamme d'utilisateurs et avec des services de plus en plus nombreux. Ces propriétés offrent beaucoup d'avantages aux utilisateurs tels que la facilite d'accès, la souplesse d'utilisation, la mobilité. Mais pour les concepteurs et les administrateurs systèmes, ces avantages peuvent engendrer beaucoup d'inconvénients, tels que les problèmes de mises à jour, la gestion des utilisateurs et les problèmes de la sécurité et la protection de l'information.

Les problèmes liés à la sécurité sont un facteur déterminant dans l'évaluation de la fiabilité des systèmes. Si ces problèmes sont bien gérés alors la fiabilité du système augmente Considérablement et la plus-value du système sera considérable. Sinon, la fiabilité est remise en cause et le système ne sera pas exploitable et donc sans valeur rajoutée. Donc la protection du système est devenue une tache très importante pour les administrateurs. Ils peuvent utiliser plusieurs outils et mécanismes de protection tels que :

Les mécanismes de chiffrement pour assurer la confidentialité et l'intégrité des données.

· Les firewalls pour contrôler les accès et filtrer le trafic réseau.

· Les scanners de vulnérabilité pour détecter les failles de sécurité dans le système.

· Les antivirus pour protéger le système contre les programmes malveillants.

· Les systèmes de détection d'intrusion pour détecter toute utilisation illégale du système.

Ces outils jouent des rôles complémentaires, ils peuvent être utilises séparément comme ils peuvent cohabiter pour mener à bien le processus de protection du système [20].

Dans ce chapitre, on a deux grandes parties .Dans la première partie, nous présentons tout d'abord la sécurité informatique c'est quoi la sécurité et quel le sont ses objectifs. Ensuite, nous parlerons sur la classification des attaques informatique et ses différents types enfin, nous allons présenter quelques outils de sécurité utilisées.

Notre deuxième partie concernent la notion de système de détection d'intrusion ainsi que son architecture. Nous présentons également la classification des IDS, Dans ce cadre plusieurs critère sont pris en compte L'emplacement d'IDS, Les méthodes de détection, les types de réponse et la Fréquence d'utilisation. Nous parlons ensuite sur les critères de choix d'IDS.

Enfin, nous parlons sur les différentes méthodes pour tester les systèmes de détection d'intrusions.

1/ La Sécurité des systèmes informatiques

1.1. Définition :

La sécurité informatique est l'ensemble des moyens techniques, organisationnels, Juridiques et humains nécessaires et mis en place pour conserver, rétablir, et garantir la sécurité des systèmes informatiques. Elle est intrinsèquement liée à la sécurité de l'information et des systèmes d'information [1].

1.2. Terminologie:

Parmi les termes les plus utilisés dans le domaine de la sécurité informatique, on
Retrouve notamment vulnérabilité, attaque, intrusion, menace, risque et incident [2].

- **Vulnérabilité**: faute créée durant le développement du système, pouvant être exploitée Afin de créer une intrusion.
- **Attaque**: faute d'interaction malveillante, à travers laquelle un attaquant cherche à Délibérément violer une ou plusieurs propriétés de sécurité. Il s'agit d'une tentative d'intrusion.
- **Intrusion**: faute malveillante externe résultant d'une attaque qui a réussi à exploiter une vulnérabilité.
- **Menace**: possibilités et probabilités d'attaque contre la sécurité. Une menace est définie Par le processus d'attaque, par la cible et par le résultat (conséquences de la réussite d'une attaque).
- **Risque** : résultat de la combinaison de menaces et de vulnérabilités.
- **Incident**: évènement qui ne fait pas partie des opérations standards d'un service et qui provoque ou peut provoquer une interruption de service ou altérer sa qualité.

1.3. Les objectifs de la sécurité d'un réseau informatique :

La sécurité des réseaux informatique tente d'atteindre plusieurs objectifs parmi eux nous citons:

- **Confidentialité :** seules les personnes autorisées peuvent avoir accès aux informations qui leur sont destinées (notions de droits ou permissions). Tout accès indésirable doit être empêché.[1]
- **Intégrité :** les données doivent être celles que l'on attend, et ne doivent pas être altérées de façon fortuite, illicite ou malveillante. En clair, les éléments considérés doivent être exacts et complets. Cet objectif utilise généralement des méthodes de calcul de checksum ou de hachage.[1]
- **Disponibilité :** l'accès aux ressources du système d'information doit être permanent et sans faille durant les plages d'utilisation prévues. Les services et ressources sont accessibles rapidement et régulièrement.[1]
- **La non répudiation de l'origine :**

La technique de non répudiation à l'origine vise à éliminer le risque qu'un émetteur puisse nier avoir envoyé un message alors que réellement tel a été le cas. A titre d'exemple, lors d'une transaction commerciale électronique, le service de la non répudiation de l'origine oblige le client à ne pas démentir le fait qu'il a adressé une requête d'achat au fournisseur **[3].**

- **La non répudiation de la délivrance :**

Le destinataire ne pourra jamais nier l'arrivée d'un message qu'il a réellement reçu. A titre d'exemple, lors d'une transaction commerciale électronique, le service du non répudiation de la délivrance oblige le fournisseur à ne pas démentir le fait que le client lui a adressé une requête d'achat **[3].**

- **Le secret du flux :**

L'intérêt est d'empêcher tout utilisateur non autorisé d'avoir la possibilité d'analyser les flux des données à travers le réseau. Tout accès illégal, même en lecture, à un flux de données permet à l'utilisateur de déduire des informations utiles et qui peuvent, ultérieurement, servir ses intentions malveillantes. La taille des messages échangés, leurs sources et leurs destinations, ainsi que la fréquence des communications entre les utilisateurs sont des exemples de données à préserver pour prévenir le secret des flux dans le réseau et le rendre plus sûr **[3].**

1.4. Classification des attaques informatiques :

Une attaque peut être classées selon son objectif, son point d'initiation ou la façon d'adresser la victime désirée.[4]

1.4.1. Selon l'objectif d'attaque :

On trouve deux types d'attaques principaux : passives et actives.

• **Les attaques passives :** ce type d'attaque ne provoque pas d'altération aux ressources du système ciblé ce qui le rend généralement indétectable (récupération du contenu d'un message ou bien l'observation du trafic).

• **Les attaques actives :** consistent à effectuer des modifications ou bien une des tractions des ressources d'un système d'une manière non autorisée. Ce type d'attaque est plus dangereux que le premier et peut causer des dégâts (usurpation de l'identité, modification ,replay, déni de service...etc.).

1.4.2. Selon le point d'initiation :

On distingue deux types d'attaques pour ce critère de classification : attaques de l'intérieur et attaques de l'extérieur.

• **Les attaques de l'intérieur :** provenant des utilisateurs légitimes d'un système lorsqu'ils se comportent de façon non autorisée.

• **Les attaques de l'extérieur :** venant de l'extérieur, souvent via Internet, en utilisant des techniques comme l'usurpation d'identité.

1.4.3. Selon la façon d'adresser la victime :

Il existe deux façons pour adresser la victime soit d'une manière directe ou bien indirecte.

• **Les attaques directes :** dans ce type d'attaque, l'intrus adresse ses paquets directement à la victime sans passer par un intermédiaire.

• **Les attaques indirectes :** dans ce type d'attaque, l'adversaire envoie ses paquets vers une entité intermédiaire qui à son tour les retransmet vers la victime.

1.5. Les types d'attaques :

Il existe essentiellement 04 types d'attaques :

1.5.1. Les attaques réseaux :

Les attaques réseaux profitent des vulnérabilités du réseau. Voici quelques exemples d'attaques réseaux :

- **Usurpation d'adresse IP :**

L'usurpation d'adresse IP (IP spoofing) est une technique qui consiste à envoyer des paquets IP en utilisant une adresse IP source qui n'a pas été attribuée à l'ordinateur qui les émet. Le but peut être de masquer sa propre identité lors d'une attaque d'un serveur, ou d'usurper en quelque sorte l'identité d'un autre équipement du réseau pour bénéficier des services auxquels il a accès [05].

- **DNS Spoofing :**

Elle consiste à fournir de fausses réponses aux requêtes DNS, c'est-à-dire indiquer une fausse adresse IP pour un nom de domaine, afin de rediriger, à leur insu, des internautes vers des sites pirates. Grâce à cette fausse redirection, l'utilisateur peut envoyer ses identifiants en toute confiance. Il existe deux techniques pour effectuer cette attaque :[6]

- **Empoisonnement du cache DNS :**

L'empoisonnement du cache DNS ou pollution de cache DNS (DNS cache poisoning) est une technique permettant de leurrer les serveurs DNS afin de leur faire croire qu'ils reçoivent une réponse valide à une requête qu'ils effectuent, alors qu'elle est frauduleuse. Une fois que le serveur DNS a été empoisonné, l'information est mise dans un cache, rendant ainsi vulnérable tous les utilisateurs de ce serveur. Ce type d'attaque permet, par exemple, d'envoyer un utilisateur vers un faux site dont le contenu peut servir à de l'hameçonnage (dans le cas du DNS) ou comme vecteur de virus et autres applications malveillantes.

- **DNS ID Spoofing :**

Lors d'une requête pour obtenir l'adresse IP à partir d'un nom, un numéro d'identification est placé dans la trame afin que le client et le serveur puissent identifier la requête. L'attaque consiste ici à récupérer ce numéro d'identification (en sniffant le réseau) lors de la communication entre un client et un serveur DNS, puis, envoyer des réponses falsifiées au client avant la réponse du serveur DNS.

❖ **ARP Spoofing :**

Cette attaque consiste à rediriger le trafic d'une machine vers une autre. Grâce à cette redirection, une personne mal intentionnée peut se faire passer pour une autre. De plus, le pirate peut rerouter les paquets qu'il reçoit vers le véritable destinataire, ainsi l'utilisateur usurpé ne se rendra compte de rien. La finalité est la même que l'IP spoofing, mais ARP Spoofing (ARP Redirect) travaille au niveau de la couche liaison de données.[6]

❖ **TCP Session Hijacking :**

Cette attaque consiste à rediriger un flux TCP afin de pouvoir outrepasser une protection par mot de passe.Ainsi le contrôle d'authentification s'effectuant uniquement à l'ouverture de la session, un pirate réussissant cette attaque parviendra à prendre possession de la connexion pendant toute la durée de la session. Dans un premier temps, le pirate doit écouter le réseau, puis lorsqu'il estime que l'authentification a pu se produire(délai de n secondes par exemple), il désynchronisera la session entre l'utilisateur et le serveur. Pour ce faire, il construit un paquet avec, comme adresse IP source, celle de la machine de l'utilisateur et le numéro d'acquittement TCP attendu par le serveur. En plus de désynchroniser la connexion TCP, ce paquet permettra au pirate d'injecter une commande via la session préalablement établie [07].

1.5.2. Les attaques applicatives :

Les attaques applicatives se basent sur des failles dans le programme utilisé, ou encore sur des erreurs de configuration. Toutes fois, il est possible de classifier ces attaques selon leur provenance :

❖ **Les problèmes de configurations :**

En général les administrateurs réseau se contente d'utiliser les configurations par défaut. Celles-ci sont souvent non sécurisées afin de faciliter l'exploitation du logiciel. De plus, des erreurs peuvent apparaitre lors de la configuration d'un logiciel. Une mauvaise configuration d'un serveur peut entrainer l'accès à des fichiers important ou mettre en jeu l'intégrité du système d'exploitation[6].

❖ **Les scripts :**

Les scripts s'exécutent sur un serveur et renvoient un résultat au client. Cependant lorsqu'ils sont dynamiques ils utilisent des entrées saisies par un utilisateur. Des failles peuvent apparaitre si les entrées ne sont pas correctement contrôlées. L'exemple classique est l'exploitation de fichier à distance, tel quel' affichage du fichier mot de passe du système en remontant l'arborescence depuis le répertoire web[6].

❖ **Les injections SQL :**

Tout comme les attaques de scripts, 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 (exemple : des mots de passe) ou encore de détruire des données [08].

❖ **Man in the middle:**

Cette attaque permet de détourner le trafic entre deux stations. Imaginons un client communiquant avec un serveur. Un pirate peut détourner le trafic du client en faisant passer les requêtes du client vers le serveur par sa machine, puis transmettre les requêtes de sa machine vers le serveur. Et inversement pour les réponses du serveur vers le client. Totalement transparente pour le client, la machine du pirate joue le rôle de proxy. Il accédera ainsi à toutes les communications et pourra en obtenir les informations sans quel' utilisateur s'en rende compte.[6]

1.5.3. Le Déni de service :

Le déni de service est une attaque visant à rendre indisponible un service. Ceci peut s'effectuer de plusieurs manières, par le biais d'une surcharge réseau rendant ainsi la machine totalement injoignable, ou bien de manière applicative en crashant l'application à distance. Grâce à quelques instructions malicieuses et suite à une erreur de programmation, une personne mal intentionnée peut rendre indisponible un service (serveur web, serveur de messagerie) voire un système complet. Voici quelques attaques réseaux permettant de rendre indisponible un service :

❖ **SYN Flooding :**

Le principe est de laisser un grand nombre de connexions TCP en attente. Le pirate envoie de nombreuses demandes de connexion (SYN), reçoit les SYN-ACK mais ne répond jamais avec ACK. Les connexions encours occupent des ressources mémoire ce qui va entraîner une saturation et l'effondrement du système.[6]

❖ **UDP Flooding:**

Le trafic UDP est prioritaire sur TCP. Le but est donc d'envoyer un grand nombre de paquets UDP, ce qui va occuper toute la bande passante et ainsi rendre indisponible toutes les connexions TCP.[6]

❖ **Packet Fragment :**

Cette attaque utilise une mauvaise gestion de la défragmentation au niveau ICMP.

Exemple :ping of death.

La quantité des données est supérieure à la taille maximum d'un paquet IP.[6]

- **Smurfling :**

Le pirate fait des requêtes ICMP ECHO à des adresses de broadcast en spoofant l'adresse source (en indiquant l'adresse de la machine cible). Cette machine cible va recevoir un nombre énorme de réponses,car toutes les machines vont lui répondre, et ainsi utiliser toute sa bande passante [07].

1.5.4.Les attaques des données :

Les données transportées par le protocole applicatif (Contenu) peuvent constituer une menace pour l'intégrité du système qui les reçoit. Les principales attaques de ce type, nous trouvons : virus, ver, Applet Java, trojans…désignés par les codes malicieux ou Malwares.

- **Les virus :**

Un virus informatique est un programme doté des propriétés, infection, multiplication et possession d'une fonction nocive. La fonction d'infection permet au virus de s'introduire dans des programmes et données utilisant un langage de script. Lors de l'accès à ces derniers, le code du virus s'exécutera de façon d'abord silencieuse (phase de multiplication pendant laquelle il infectera d'autres fichiers) puis visible (activation de la fonction nocive). Cette dernière pourra être déclenchée par des facteurs très variables selon le virus
(au bout de n réplications, à une date fixe, lors de l'exécution de certaines tâches précises…).
Elle peut se limiter à l'affichage d'un message agaçant ou conduire à des perturbations graves de l'ordinateur(ralentissement du fonctionnement, effacement ou corruption de fichiers, formatage du disque dur...) [05].

- **Le cheval de Troie :**

Initialement un cheval de Troie désignait un programme se présentant comme un programme normal destiné à remplir une tâche donnée, voire ayant parfois un nom connu (en quelque sorte déguisé sous une fausse apparence) mais qui, une fois installé exerçait une action nocive totalement différente de sa fonction officielle. Actuellement le terme désigne à peu près tout programme qui s'installe de façon frauduleuse (souvent par le biais d'un mail ou d'une page web piégés) pour remplir une tâche hostile à l'insu de l'utilisateur. Les fonctions nocives

peuvent être l'espionnage de l'ordinateur, l'envoi massif de spams, l'ouverture d'un accès pour un pirate [05].

❖ **Un ver :**

Un ver est un logiciel malveillant qui se reproduit sur plusieurs ordinateurs en utilisant un réseau informatique. Contrairement à un virus informatique, un ver n'a pas besoin d'un programme hôte pour se reproduire, il exploite les différentes ressources de l'ordinateur qui l'héberge pour assurer sa reproduction.

L'objectif du ver est d'espionner l'ordinateur où il se trouve, offrir une porte dérobée à des pirates informatiques, détruire les données de l'ordinateur infecté et envoyer de multiples requêtes vers un serveur Internet dans le but de le saturer (déni de service). Il a pour effet le ralentissement de la machine infectée.[6]

❖ **Les portes dérobées (backdoor) :**

Une porte dérobée peut être introduite soit par le développeur du logiciel ou un pirate informatique. La personne connaissant la porte dérobée peut l'utiliser pour surveiller les activités du logiciel, voire en prendre le contrôle par contournement de l'authentification. Parmi les motivations amenant les développeurs de logiciel à créer des portes dérobées, il ya:[6]

- L'intérêt pratique d'un accès facile et toujours ouvert au logiciel pour pouvoir mener efficace mentles actions de maintenance.
- La possibilité de désactiver secrètement le logiciel en cas de désaccord avec son client (non-paiement de licence).Parmi les motivations amenant les pirates informatiques à installer une porte dérobée :
- La possibilité de surveiller ce que fait l'utilisateur légitime et de copier ou détruire des données ayant une valeur (mots de passe, clé privée pour déchiffrer des messages privés, coordonnées bancaires, secrets commerciaux).
- La possibilité de prendre le contrôle d'un ordinateur et de pouvoir l'utiliser pour mener des actions mal faisantes (envoi de courriels notamment pour l'hameçonnage, de virus informatiques, déni de service).

- Le contrôle d'un vaste réseau d'ordinateurs, qui peut être utilisé pour du chantage au déni de service distribué (DDoS), ou revendu à des criminels.

1.6. Buts des attaques :

Il existe plusieurs objectifs pour les attaques informatiques :[9]

• **Fabrication :** vise l'authenticité des Informations.

• **Interruption :** vise la disponibilité des informations.

• **Interception :** vise la confidentialité des informations.

• **Modification :** vise l'intégrité des informations.

Les objectifs sont illustrés dans les figures suivantes :

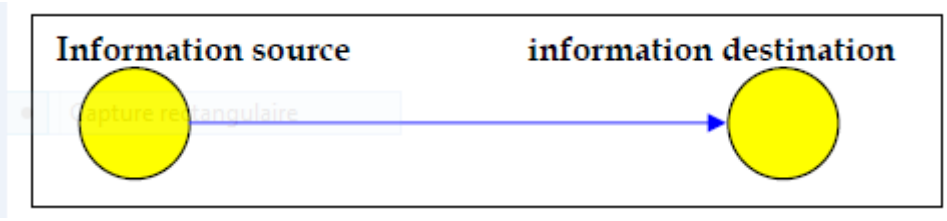


Figure 1.1 : Flux normal de transmission de l'information.

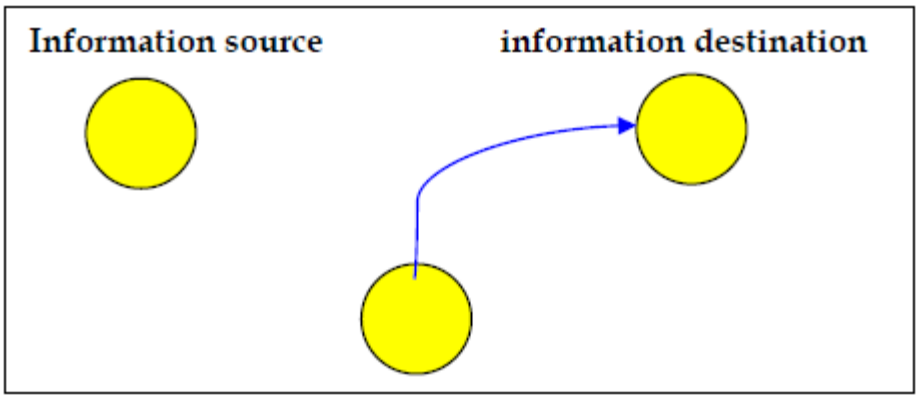


Figure 1.2 : Fabrication.

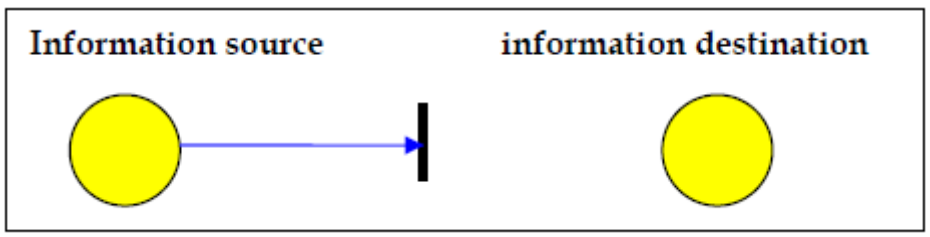


Figure 1.3 : Interruption.

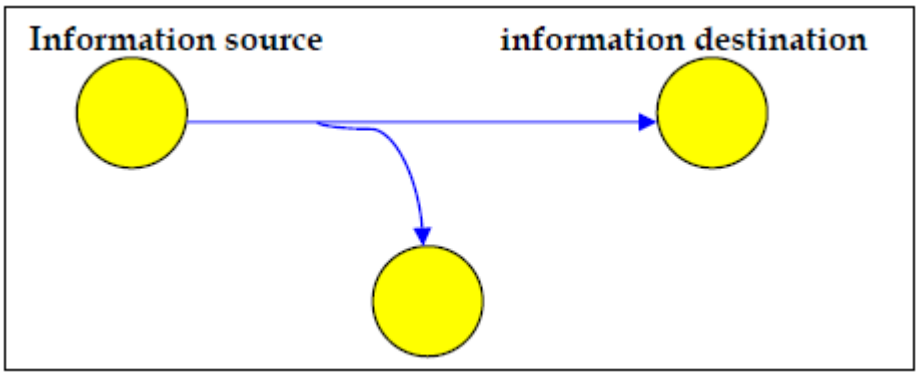


Figure 1.4 : Interception.

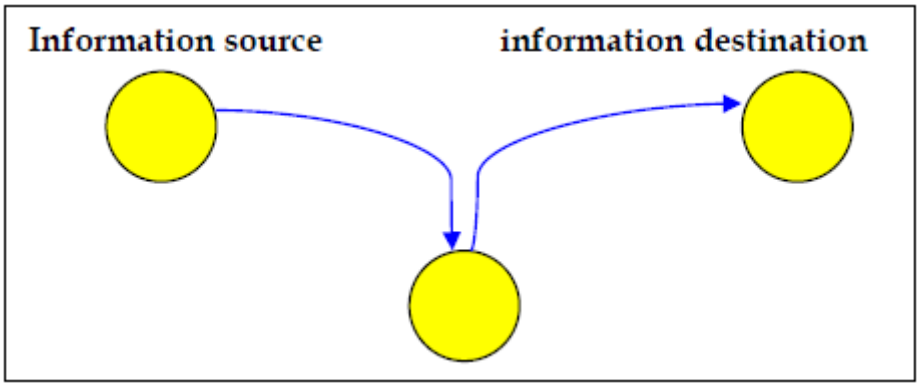


Figure 1.5 : Modification.

1.7. Outils de sécurité :

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

1.7.1.Cryptographie :

Les données qui peuvent être lues et comprises sans mesures spéciales sont appelées texte clair. Le procédé qui consiste à dissimuler du texte clair de façon à cacher sa substance est appelée cryptographie ou chiffrement. Le chiffrement des données fut inventé pour assurer la confidentialité des données. Il est assuré par un système de clé (algorithme) appliqué sur le message. Ce dernier est décryptable par une clé unique correspondant au cryptage ,Il existe à l'heure actuelle deux grands principes de cryptage : le cryptage symétrique basé sur l'utilisation d'une clé privée et le cryptage asymétrique qui repose sur un codage à deux clés, une privée et l'autre publique [10].

1.7.1.1. Le cryptage symétrique :

Le cryptage à clé privée ou symétrique est basé sur une clé (ou algorithme) partagée entre les deux parties communicantes. Cette même clé sert à crypter et décrypter les messages.

Figure 1.6 : Cryptage symétrique.

1.7.1.2.Le cryptage asymétrique :

Pour pallier la complexité induite par la gestion de la distribution des clés par cryptographie symétrique. Un autre type de cryptage qualifié d'asymétrique a été conçu et utilisé largement dans le monde de l'internet.

Ce système de cryptage utilise deux clés différentes pour chaque utilisateur, une privée et n'est connue que de l'utilisateur, l'autre publique et donc accessible par tout le monde.

- Une première clé, visible, appelé clé publique est utilisée pour chiffrer un texte en clair.
- Une deuxième clé, secrète, appelée clé privée est connue seulement par le destinataire, qui est utilisé pour déchiffrer un texte.

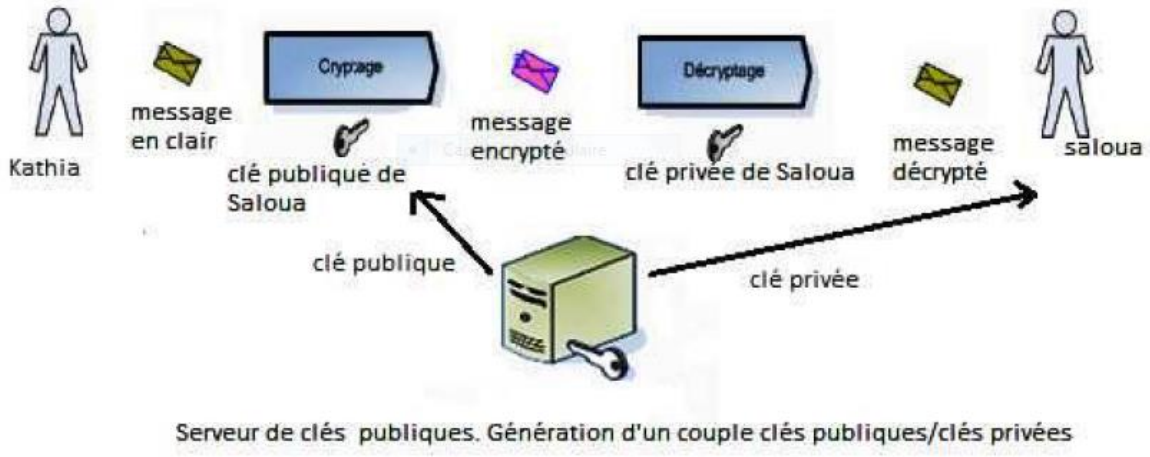


Figure 1.7 : Le cryptage asymétrique.

1.7.2. Les Anti-virus :

Un antivirus est un logiciel informatique destiné à identifier et à effacer des logiciels malveillants(malwares), également appelés virus, Chevaux de Troie ou vers selon les formes [08].

L'antivirus analyse les fichiers entrants (fichiers téléchargés ou courriers électroniques), la mémoire vive de l'ordinateur et les périphériques de stockage comme les disques durs, internes ou externes, les clés USB et les cartes à mémoire Flash. La détection d'un logiciel malveillant peut reposer sur trois méthodes :

- Reconnaissance d'un code déjà connu (appelé signature) et mémorisé dans une base de données.
- Analyse du comportement d'un logiciel.
- Reconnaissance d'un code typique d'un virus.

1.7.3. Les VPN :

VPN (Virtual Private Network) ou RPV (Réseau privé virtuel) est une technique permettant à un ou plusieurs postes distants de communiquer de manière sure, tout en empruntant les infrastructures publiques.

Ce type de liaison est apparu suite à un besoin croissant des entreprises de relier les différents sites, et ce de façon simple et économique.

Un réseau VPN repose sur le protocole de tunneling. Ce protocole permet de faire circuler les informations de l'entreprise de façon cryptée d'un bout à l'autre du tunnel. Ainsi, les utilisateurs ont l'impression de se connecter directement sur le réseau de leur entreprise.
Le principe de tunneling consiste à construire un chemin virtuel après avoir identifié l'émetteur et le destinataire. Par la suite, la source chiffre les données et les achemine en empruntant ce chemin virtuel. Afin d'assurer un accès aisé et peu coûteux aux intranets ou aux extranets de l'entreprise, les VPN d'accès simulent un réseau privé, alors qu'ils utilisent en réalité une infrastructure d'accès partagé, comme Internet[05].

1.7.4. Le NAT :

Dans les entreprises de grandes tailles, différents réseaux interconnectés peuvent utiliser les mêmes adresses IP.Pour que la communication soit possible entre nœuds des deux coté, il est nécessaire de modifier les références de l'émetteur de paquets afin qu'il n'y ait pas de conflits et que la transmission soit fiable.
Des équipements de translation d'adresse NAT (Network Address Translation) sont chargés d'adopter cette fonctionnalité. Ils permettent le changement d'une adresse IP par une autre.
Trois types d'adresse sont possibles :

- La translation de port PAT (Port Address Translation), joue sur une allocation dynamique des ports TCP ou UDP, en conservant l'adresse IP d'origine.
- La conversion dynamique d'adresses (NAT dynamique) change à la volée d'adresse IP par rapport à une externe disponible dans une liste.
- La conversion statique d'adresse (NAT statique), effectue également un changement d'adresse IP,mais une table est maintenue, permettant à une adresse IP interne de toujours être remplacée par lamême adresse IP externe [11].

1.7.5. Les ACL :

Les listes de contrôle d'accès (Access Control List) ont pour objectif de disposer d'une fonction de filtrage prenant en compte l'historique des connexions en cours, afin de ne pas accepter du trafic qui n'aurait pas été demandé à partir d'une zone précise du réseau [12].
Les ACL semblent avoir toujours existé sur les routeurs et rares sont les configurations où elles n'apparaissent pas. Elles servent principalement au filtrage des paquets sur les interfaces physiques.

Cependant leur mode de définition est employé pour catégoriser les réseaux en vue, entre autres, de les injecter dans un protocole de routage ou de les soumettre à une règle de qualité de service.

Il existe deux types d'ACL :

- **Les ACL standard :** permettent d'autoriser ou de refuser le trafic en provenance d'adresse IP source et la destination du paquet, tandis que les ports n'ont aucune incidence.
- **Les ACL étendues :** filtrent les paquets IP en fonction de plusieurs attributs, dont le type de protocole, l'adresse IP source, l'adresse IP destination, les ports TCP ou UDP source et destination et les informations facultatives sur le type de protocole pour une meilleure précision du contrôle. De la configuration des ACL, chaque liste est identifiée par un numéro unique attribué. Ce numéro permet d'identifie le type d'ACL créé et doit être compris dans les plages suivantes :

- Les ACL standard : 1-99 ,1300-1999.
- Les ACL étendues : 100-199, 2000-2699.

2. Système de détection d'intrusion

2.1. Définition :

Un système de détection d'intrusions (IDS, de l'anglais intrusion détection system) inclure tous les systèmes logiciels et matériels permettant d'automatiser les processus de surveillance et d'analyse des évènements au sein d'un système informatique afin de détecter toute activité pouvant conduire à une défaillance de sécurité[1].

Le but des IDS est de détecter des violations de confidentialité et d'intégrité, et la Disponibilité réduite des ressources. Un IDS surveille le réseau et améliore l'activité de L'utilisateur pour détecter l'intrusion [13].

2.2. Architecture d'un IDS [4] :

Plusieurs architectures ont été proposées pour décrire les différents éléments constituant un système de détection d'intrusions. L'architecture la plus simple est composée de trois modules:

a) Le capteur

Chargé de collecter, filtrer et formater les informations brutes envoyées par la source de données concernant l'évolution de l'état du système. Le résultat de traitement est un message formaté appelé événement.

b) L'analyseur

Permet d'analyser les événements générés par le capteur en détectant toute activité malveillante qui peut se produire à partir d'un sous ensemble de ces événements, et donc envoyer une alerte qui sera stockée dans les journaux du système ou bien utilisée pour lutter contre les attaques selon le type d'IDS.

c) Le manager

Permet de collecter et notifier les alertes envoyées par l'analyseur. Éventuellement, le manager est chargé de la réaction à adopter qui peut être :

• Isolement de l'attaque pour réduire les dégâts.

• Suppression d'attaque.

• Restauration du système dans un état sain.

• Identification du problème qui a engendré cette attaque

2.3. Classification des IDS :

Nous allons présenter ici la technologie de détection d'intrusion d'une manière taxonomique. Il y a plusieurs types d'IDS disponibles aujourd'hui, caractérisé par différente approche de surveillance et de l'analyse. Chaque approche a ses avantages et des inconvénients distincts. Toutes les approches peuvent être décrites en termes d'un modèle d'IDS (voir figure 1.8)

- L'emplacement d'IDS
- Les méthodes de détection
- Les types de réponse
- Fréquence d'utilisation [14].

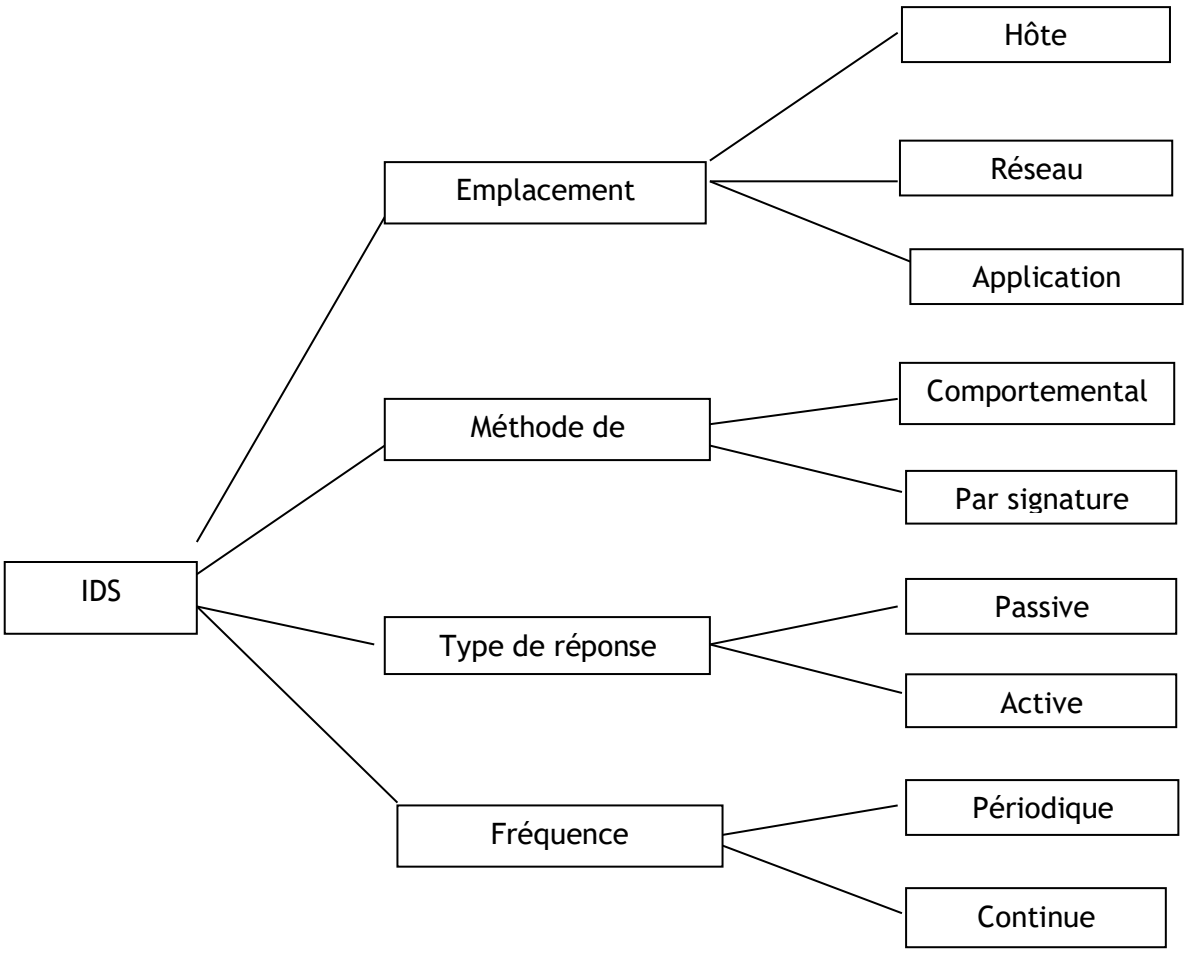


Figure 1.8: Les différentes sortes des IDS

2.3.1. L'emplacement d'IDS :

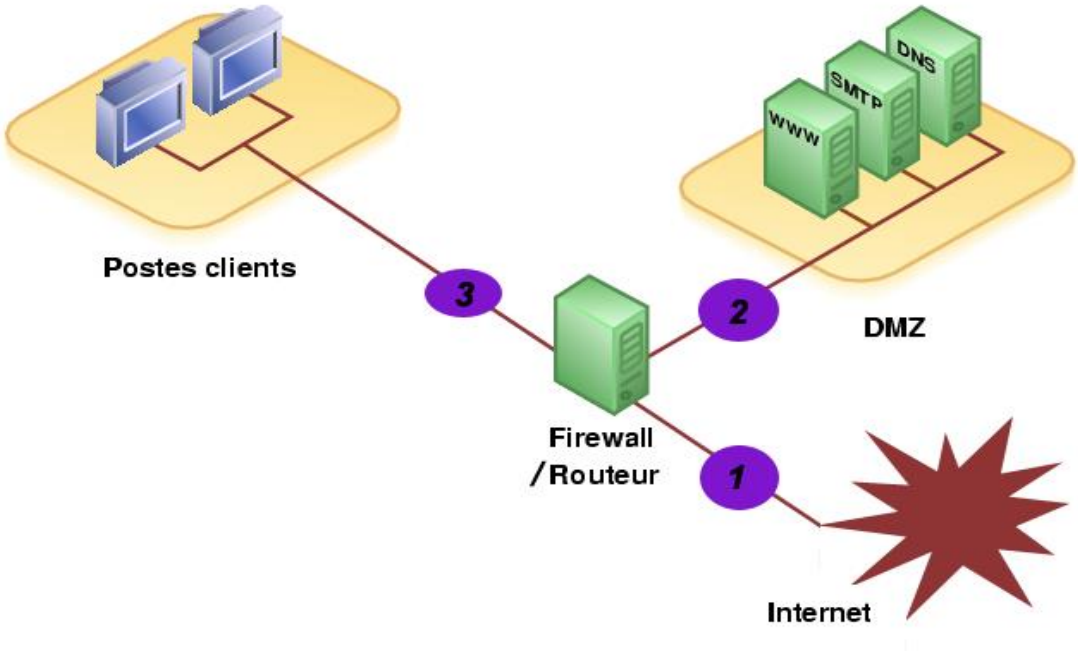


Figure 1.9 : L'emplacement d'IDS [18]

a. Système de détection d'intrusion de type hôte(HIDS) :

Les systèmes de détection d'intrusion basés sur l'hôte ou HIDS (Host IDS) analysent exclusivement l'information concernant cet hôte. Comme ils n'ont pas à contrôler le trafic du réseau mais "seulement" les activités d'un hôte, ils se montrent habituellement plus précis sur les types d'attaques subies.

De plus, l'impact sur la machine concernée est sensible immédiatement, par exemple dans le cas d'une attaque réussie par un utilisateur. Ces IDS 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.

Chacun a ses avantages : 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.

Ces derniers peuvent être mieux contrôlés et analysés en raison de leur taille, mais certaines attaques peuvent passer inaperçues, alors qu'elles sont détectables par une analyse des traces d'audit.

Ce type d'IDS possède un certain nombre d'avantages : il est possible de constater immédiatement l'impact d'une attaque et donc de mieux réagir. Grâce à la quantité des informations étudiées, il est possible d'observer les activités se déroulant sur l'hôte avec précision et d'optimiser le système en fonction des activités observées.

De plus, les HIDS sont extrêmement complémentaires des NIDS. En effet, ils permettent de détecter plus facilement les attaques de type "Cheval de Troie", alors que ce type d'attaque est difficilement détectable par un NIDS. Les HIDS permettent également de détecter des attaques impossibles à détecter avec un NIDS, car elles font partie de trafic crypté.

Néanmoins, ce type d'IDS possède également ses faiblesses, qui proviennent de ses qualités du fait de la grande quantité de données générées, ce type d'IDS est très sensible aux attaques de type DoS, qui peuvent faire exploser la taille des fichiers de logs.

Un autre inconvénient tient justement à la taille des fichiers de rapport d'alertes à examiner, qui est très contraignante pour le responsable sécurité. La taille des fichiers peut en effet atteindre plusieurs Mégaoctets.

Du fait de cette quantité de données à traiter, ils sont assez gourmands en CPU et peuvent parfois altérer les performances de la machine hôte. Enfin, ils ont moins de facilité à détecter les attaques de type hôte que les IDS réseaux.

Les HIDS sont en général placés sur des machines sensibles, susceptibles de subir des attaques et possédant des données sensibles pour l'entreprise. Les serveurs, web et applicatifs, peuvent notamment être protégés par un HIDS.

Pour finir, voici quelques HIDS connus: Trip Wire, WATCH, DragonSquire, Tiger,Security Manager... [15].

b. Système de détection d'intrusion basée sur une application :

Les IDS basés sur les applications (AIDS) sont un sous-groupe des IDS hôtes. Ils contrôlent l'interaction entre un utilisateur et un programme en ajoutant des fichiers de log afin de fournir de plus amples informations sur les activités d'une application particulière. Puisque vous opérez entre un utilisateur et un programme, il est facile de filtrer tout comportement notable. Un ABIDS se situe au niveau de la communication entre un utilisateur et l'application surveillée.

L'avantage de cet IDS est qu'il lui est possible de détecter et d'empêcher des commandes particulières dont l'utilisateur pourrait se servir avec le programme et de surveiller chaque transaction entre l'utilisateur et l'application. De plus, les données sont décodées dans un contexte connu, leur analyse est donc plus fine et précise.

Par contre, du fait que cet IDS n'agit pas au niveau du noyau, la sécurité assurée est plus faible, notamment en ce qui concerne les attaques de type "Cheval de Troie".

De plus, les fichiers de log générés par ce type d'IDS sont des cibles faciles pour les attaquants et ne sont pas aussi sûrs, par exemple, que les traces d'audit du système. Ce type d'IDS est utile pour surveiller l'activité d'une application très sensible, mais son utilisation s'effectue en général en association avec un HIDS. Il faudra dans ce cas contrôler le taux d'utilisation CPU des IDS afin de ne pas compromettre les performances de la machine [14].

c. Système de détection d'intrusion réseau (NIDS) :

Les systèmes de détection d'intrusion dans le réseau (NIDS) sont placés à un ou plusieurs points stratégiques du réseau pour surveiller le trafic vers et depuis tous les appareils du réseau. Il effectue une analyse du trafic passant sur l'ensemble du sous- réseau et fait correspondre le trafic transmis sur les sous-réseaux à la bibliothèque des attaques connues. Une fois qu'une attaque est identifiée ou qu'un comportement anormal est détecté, l'alerte peut être envoyée à l'administrateur.

Un exemple de NIDS serait de l'installer sur le sous-réseau où se trouvent lepare-feu afin de voir si quelqu'un essaie de s'introduire dans le pare-feu. Idéalement, il faudrait analyser tout le trafic entrant et sortant, mais cela pourrait créer un goulot d'étranglement qui réduirait la vitesse globale du réseau. OPNET et Net Sim sont des outils couramment utilisés pour simuler des systèmes de détection d'intrusion réseau. Les systèmes NID sont également capables de comparer les signatures de paquets similaires pour lier et supprimer les paquets détectés nuisibles qui ont une signature correspondant aux enregistrements dans le NIDS.

Lorsque nous classons la conception du NIDS en fonction de la propriété d'interactivité du système, il existe deux types : les NIDS en ligne et hors ligne, souvent appelés mode en ligne et mode tap, respectivement. Le NIDS en ligne traite le réseau en temps réel. Il analyse les paquets Ethernet et applique quelques règles, pour décider s'il s'agit d'une attaque ou non. Le NIDS hors ligne traite les données stockées et les transmet à certains processus pour décider s'il s'agit d'une attaque ou non [19].

2.3.2. Méthode de détection [16]:

Une autre différenciation se fait sur la manière de détecter une attaque. Elle peut se baser sur ses signatures (aussi appelée approche par scénario) ou en se basant sur des profils normaux d'utilisation (aussi appelée approche comportementale). Dans le premier cas, on regarde la suite d'actions effectuées par une personne et on la compare à une attaque connue. Dans le deuxième cas, on regarde le comportement d'une personne et on le compare à son comportement normal. En d'autres mots, l'un sait ce qui est mal et l'autre ce qui est bien.

a. Approche par scénario :

Ce type d'IDS contient une base de données des signatures d'attaques et essaye de faire correspondre une donnée, obtenue par les sources d'information du système, avec celles connues. Ces signatures sont les caractéristiques d'une attaque, c'est-à-dire l'empreinte d'une attaque connue, et peuvent varier d'un système à l'autre. Pour la correspondance, on utilise les algorithmes de pattern Matching tel qu'E2xB, Boyer-Moore et Knuth-Morris-Pratt.

Un avantage de cette approche est la normalisation possible de la description des signatures et ainsi la possibilité de diffuser et de configurer rapidement et facilement les IDS. De plus, ceci permet d'avoir un faible taux de fausses alarmes. Néanmoins, la problématique est le besoin d'une mise à jour régulière de la base de données des signatures pour détecter de nouvelles attaques. Ils sont donc faillibles aux attaques 0day qui sont des exploitations d'une faiblesse le jour de leur publication. De plus, une même attaque peut être faite de plusieurs manières légèrement différentes. Ainsi, le pattern Matching ne semble pas très efficace.

A sa place, on peut utiliser le machine Learning pour apprendre leurs signatures de manière générique et ainsi les repérer même si celles-ci évoluent légèrement au cours du temps.

b. Détection comportementale :

Le but de cette technique est la prédiction de comportement. Pour cela, cette technique utilise une base de données des comportements normaux des utilisateurs, d'un groupe d'utilisateurs, des services ou d'un système entier pour constituer un profil

Des attaques inconnues peuvent donc être détectées contrairement à l'approche par scénario qui est biaisé par l'empreinte des signatures connues. Quand un comportement s'éloigne trop du comportement normal, une alarme se déclenche. Néanmoins, cet éloignement ne signifie pas forcément un comportement hostile, ce qui semble générer un taux élevé de fausses alarmes. La création du profil peut se faire grâce à plusieurs métriques : les commandes utilisées, la charge CPU, les heures de connexions, Un des inconvénients de cette méthode est l'obligation d'un temps d'apprentissage pour réaliser le profil des utilisateurs.

D'autres techniques existent pour implémenter un comportement. Par exemple, on peut utiliser les techniques de machine Learning à qui on va apprendre les différents profils. De plus, on peut réaliser un IDS totalement distribué se basant sur le système immunitaire des êtres vivants, ou sur les algorithmes génétiques se basant sur la sélection naturelle d'une population. Le système immunitaire représente l'IDS par un ensemble de systèmes distribués, diversifiés, autonomes et auto réparant pouvant détecter un intrus représenté par un corps étranger inconnu du système immunitaire. La vérification qu'une certaine propriété est toujours respectée est parfois utilisée.

Une dernière technique est la découverte d'un comportement anormal quand le protocole réalisé ne suit pas le protocole habituel.

2.3.3.Les types des réponses [15] :

Il existe deux types de réponses, suivant les IDS utilisés. La réponse passive est disponible pour tous les IDS alors que la réponse active est-elle plus ou moins implémentée.

a. Réponse passive :

La réponse passive d'un IDS consiste à enregistrer les intrusions détectées dans un fichier de log qui sera analysé par le responsable de sécurité.

Certains IDS permettent de logger l'ensemble d'une connexion identifiée comme malveillante.

Ceci permet de remédier aux failles de sécurité pour empêcher les attaques enregistrées de se reproduire, mais elle n'empêche pas directement une attaque de se produire.

b. Réponse active :

La réponse active, au contraire a pour but de stopper une attaque au moment de sa détection. Pour cela on dispose de deux techniques : la reconfiguration du firewall et l'interruption d'une connexion TCP.

La reconfiguration du firewall permet de bloquer le trafic malveillant au niveau du firewall, en fermant le port utilisé ou en interdisant l'adresse de l'attaquant. Cette fonctionnalité dépend du modèle de firewall utilisé, tous les modèles ne permettant pas la reconfiguration par un IDS. De plus, cette reconfiguration ne peut se faire qu'en fonction des capacités du firewall.

L'IDS peut également interrompre une session établie entre un attaquant et sa machine cible, de façon à empêcher le transfert de données ou la modification du système attaqué. Pour cela l'IDS envoie un paquet TCP reset aux deux extrémités de la connexion (cible et attaquant). Un paquet TCP reset a le flag RST de positionner, ce qui indique une déconnexion de la part de l'autre extrémité de la connexion. Chaque extrémité en étant destinataire, la cible et l'attaquant pensent que l'autre extrémité s'est déconnectée et l'attaque est interrompue.

Dans le cas d'une réponse active, il faut être sûr que le trafic détecté comme malveillant l'est réellement, sous peine de déconnecter des utilisateurs normaux.

En général, les IDS ne réagissent pas activement à toutes les alertes. Ils ne répondent aux alertes que quand celles-ci sont positivement certifiées comme étant des attaques. L'analyse des fichiers d'alertes générés est donc une obligation pour analyser l'ensemble des attaques détectées.

2.3.4.Fréquence d'utilisation [16]:

La différenciation peut se faire sur la caractéristique online ou offline de l'IDS.
Le premier détecte une attaque au moment où elle se produit alors que le deuxième s'exécute périodiquement et ne voit que son résultat.
La deuxième méthode est préférable pour avoir une dépense plus faible du point de vue du temps de calcul que la première, dû à l'aspect temps-réel de l'online. Néanmoins, la deuxième

méthode, contrairement à la première, souffre du fait qu'une attaque ne peut pas être détectée le plus tôt possible pour l'éviter. Plus une attaque est détectée tardivement, plus les dommages sont importants.

Dans la plupart des cas, avant d'attaquer un réseau, l'attaquant doit scanner l'environnement pour récolter des informations. Par conséquent, en voyant cela, on peut détecter une attaque avant même qu'elle ne se produise et ainsi y répondre le plus tôt possible

2.4. Critères de Choix D'un IDS :

Les systèmes de détection d'intrusion sont devenus indispensables lors de la mise en place d'une infrastructure de sécurité opérationnelle. Ils s'intègrent donc toujours dans un contexte et dans une architecture imposante des contraintes très diverses.

- **Fiabilité** : Les alertes générées doivent être justifiées et aucune intrusion ne doit Pouvoir lui échapper.
- **Réactivité** : Un IDS doit être capable de détecter les nouveaux types d'attaques le plus Rapidement possible ; pour cela il doit rester constamment à jour. Des capacités de mise à jour Automatique sont indispensables.
- **Facilité de mise en œuvre et adaptabilité** : Un IDS doit être facile à mettre en œuvre, surtout s'adapter au contexte dans lequel il doit opérer. Il est inutile d'avoir un IDS émettant des alertes en moins de 10 secondes si les ressources Nécessaires à une réaction ne sont pas disponibles pour agir dans les mêmes contraintes de Temps.
- **Performance** : la mise en place d'un IDS ne doit en aucun cas affecter les Performances des systèmes surveillés.

De plus, il faut toujours avoir la certitude que l'IDS a la capacité de traiter toute l'information à sa disposition (par exemple un IDS réseau doit être capable de traiter l'ensemble du flux pouvant se présenter à un instant donné sans jamais supprimer de paquets) car dans le cas contraire il devient trivial de masquer les attaques en augmentant la quantité d'information [18].

2.5. Les Différents méthodes pour tester les IDS

Avent la mise en place d'un IDS, il est nécessaire de tester ses limites. Pour cela, il existe plusieurs méthodes:

2.5.1. L'Attaque :

On va utiliser les outils exploités par les attaquants pour détecter une faille dans le système ou dans l'IDS, telles que les techniques d'évasion ou d'insertion.

2.5.2. L'Alarme :

On va regarder le taux des alarmes telles que les faux positifs.

2.5.3. La Qualité des informations :

On va regarder la qualité des informations fournies par l'IDS lors d'une alarme.

2.5.4. Le Réalisme :

Il est nécessaire de tester l'IDS dans un milieu réel et non pas uniquement avec un générateur d'informations tel que « Network Security Auditor ».

2.5.5. La Flexibilité et le mise à jour :

Il est souvent intéressant de pouvoir modifier les configurations d'un IDS telles que la base de signature, c'est pourquoi il est aussi nécessaire d'avoir une bonne réactivité du constructeur en cas de nouvelle attaque non encore détectée par l'IDS.

2.5.6. La Qualité des signatures :

Dans le cas des IDS se basant sur les signatures, il est nécessaire de pouvoir évaluer la qualité des signatures.

2.5.7. La Rapidité du système :

Il est nécessaire que l'IDS soit capable de gérer un grand nombre de données en un temps raisonnable et de détecter l'attaque en un minimum de temps pour réduire les dommages causés.

2.5.8. L'Intégration :

Puisque les IDS ne suffisent pas pour garantir l'ensemble de la sécurité, ils doivent être facilement installés et intégrés à son infrastructure.

2.5.9. L'Interaction :

Le nombre d'interactions entre un IDS et l'administrateur système doit être minime.

2.5.10. Le Dataset :

On va comparer les performances de l'IDS avec d'autres IDS grâce à des datasets [16].

Conclusion :

Dans ce chapitre, nous avons abordé différentes notions concernant la sécurité informatique, où nous avons présenté les différents types d'attaques, leur classification, et les techniques utilisées pour protéger le système contre ces attaques. Parmi ces mécanismes, nous avons détaillé les systèmes de détection d'intrusions.

Donc, nous présentons leur définition, architecture, classification et nous terminons par les méthodes utilisées pour les tester.

Chapitre 2
Machine learning

Introduction :

L'apprentissage automatique est un sous domaine de l'intelligence artificielle qui se divise principalement en deux façons d'apprendre: l'apprentissage supervisé et l'apprentissage non supervisé, c'est dans l'approche dite supervisée que s'inscrit la façon dont nous abordons aujourd'hui le problème de la détection d'intrusion par apprentissage automatique.

La plupart des algorithmes d'apprentissage supervisés tentent de trouver un modèle qui explique le lien entre les données d'entrée el les classes de sortie ce modèle de classification est dit classifieur.

Nous avons présenté dans ce chapitre les différentes techniques d'apprentissages automatique utilisées pour la Détection d'intrusion et le Cycle de vie d'implémentation de ces techniques

1. L'intelligence Artificielle :

1.1. Définitions de l'Intelligence Artificielle

L'Intelligence Artificielle (IA, ou AI en anglais pour *Artificial Intelligence*) est une branche de l'informatique dont le but est de réaliser des systèmes intégrants un grand nombre de connaissances et de traitements dits systèmes intelligents .Par conséquent, l'Intelligence Artificielle conduit à cet ensemble d'interprétation où :

1. c'est l'étude des techniques de résolution des problèmes exponentiellement difficiles dans un temps polynomial en exploitant la connaissance du domaine de la problématique en question.
2. on doit avoir l'impression que la machine se comporte comme un partenaire *(cas d'un bras manipulateur)*, et même parfois, de façon complètement autonome *(cas d'un robot)*.

3. la programmation des ordinateurs pour faire des tâches qui sont mieux présentées ou faites par des humains. (Minsky, 1968).

4. l'utilisation des machines pour faire des choses qui seraient considérées intelligentes si elles étaient faites par des humains (Boden 1977). C'est le

comportement par une machine qui serait considéré intelligent, comme s'il était fait par un humain.

5. L'automatisation d'activités qui nous associons à la pensée humaine, comme la prise de décision, la résolution de problème ou l'apprentissage. (BELLMAN 1978).

6. L'étude des facultés mentales à travers l'utilisation de modèles informatiques. (CHARMIAK & MCDERMOTT 1985).

7. L'étude de comment programmer les ordinateurs pour qu'ils réalisent des tâches pour lesquelles les êtres humains sont actuellement meilleurs. (RICH & KNIGHT 1991). Il s'agit, de simuler le raisonnement humain pour faire résoudre des problèmes complexes par un ordinateur.

8. L'IA est la partie de l'informatique consacrée à l'automatisation de comportements intelligents. (LUGGER & STUBBLEELD 1993)**[20].**

1.2. Comment fonctionne l'intelligence artificielle ?

Les machines dotées d'une intelligence artificielle mémorisent des comportements. Ce travail de mémorisation leur permet par la suite de résoudre des problèmes, et d'agir correctement face à telle ou telle situation. Cet apprentissage se réalise à l'aide de bases de données et d'algorithmes. Ce travail complexe aide la machine à mesurer l'importance d'un problème, à passer au crible les solutions possibles et les situations passées similaires afin de bien agir.

C'est en réalité un système de statistiques sophistiqué et très performant qui conduisent la machine à prendre une décision ou à avoir le comportement attendu. Pour mesurer son degré d'intelligence, une machine est soumise au test de Turing. Ce test porte le nom de l'inventeur de l'IA, Alan Turing. Ce mathématicien britannique fut l'un des premiers à se demander, en 1950, si une machine était capable de penser. Le test de Turing consiste à converser avec la machine et à lui demander de créer quelque chose avec des critères précis qu'elle se doit de respecter **[21].**

1.3. Approches de l'intelligence artificielle :

L'IA forte est la vision de l'IA de demain, elle a pour but de créer des machines autonomes dotées de conscience. Tandis que L'intelligence artificielle faible utilise l'IA pour des tâches précises. Elle peut réaliser des calculs, traiter un grand volume de données, résoudre des problèmes spécifiques et apprendre de manière automatisée. Mais elle n'a pas de conscience propre et agit en fonction de la manière dont elle a été programmée par l'être hu Aujourd'hui, l'IA intégrée dans les applications et les machines appartient à l'IA faible. L'IA forte constitue en revanche le défi et certainement le projet le plus ambitieux de l'IA**[22].**

1.4. Les sous-domaines de l'intelligence artificielle :

L'IA est un vaste champ d'études comprenant de nombreuses théories, méthodes et technologies, ainsi que les sous-domaines suivants :

1.4.1. Deep Learning : ou apprentissage profond, est un sous-ensemble du Machine Learning, ou apprentissage automatique, basé sur des réseaux neuronaux artificiels. Le processus d'apprentissage est qualifié de *profond* parce que la structure des réseaux neuronaux artificiels se compose de plusieurs couches d'entrée, de sortie et masquées. Chaque couche contient des unités qui transforment les données d'entrée en informations que la couche suivante peut utiliser une tâche prédictive spécifique. Grâce à cette structure, une machine est capable d'apprendre au travers de son propre traitement de données.

1.4.2. Machine Learning :est un sous-ensemble de l'intelligence artificielle utilisant des techniques (telles que le Deep Learning), qui permettent aux machines de tirer des enseignements de leur expérience pour améliorer la manière dont elles exécutent leurs tâches**[23].**

1.4.3. L'informatique cognitive :est un sous-domaine de l'IA qui vise une interaction naturelle, quasiment humaine, avec les machines. Le but ultime est de combiner IA et informatique cognitive pour simuler les processus humains en interprétant les images et la parole, et de répondre par un discours cohérent.

1.4.4. Le traitement du langage naturel : désigne la capacité des ordinateurs à analyser, comprendre et générer le langage humain, notamment sous sa forme orale. Dans sa phase évoluée, ce traitement consiste en une interaction en langage naturel, qui permet aux humains de communiquer avec des ordinateurs en parlant normalement, dans le langage de tous les jours, afin d'exécuter des tâches

1.5. Enjeux de l'intelligence artificielle :

L'intelligence artificielle va transformer tous les secteurs, encore faut-il avoir conscience de ses limites.

La première étant que l'IA apprend des données. Il n'existe aucun autre moyen d'intégrer des connaissances. Donc, toute inexactitude dans les données se reflétera dans les résultats. Par ailleurs, les éventuelles couches supplémentaires de prédiction ou d'analyse doivent être ajoutées séparément.

Aujourd'hui, les systèmes d'IA sont formés pour exécuter une tâche clairement définie. Un système qui joue au poker ne peut pas jouer au solitaire ni aux échecs. Un système élaboré pour détecter la fraude ne peut pas conduire un véhicule ni donner de conseils juridiques. En fait, un système d'IA conçu pour détecter la fraude aux soins de santé n'est pas en mesure de détecter la fraude fiscale ni les escroqueries en matière d'assurance.

En d'autres termes, ces systèmes sont extrêmement spécialisés. Ils se focalisent sur une tâche unique, à la différence du comportement humain.

De la même manière, les systèmes d'auto-apprentissage ne sont pas autonomes. Les technologies d'IA imaginaires vues au cinéma et à la tv relèvent encore de la science-fiction. Cependant, les ordinateurs capables d'analyser des données complexes pour apprendre et perfectionner des tâches spécifiques sont désormais monnaie courante. **[24].**

2. L'apprentissage automatique :

2.1. Définition :

Le machine Learning ou « apprentissage automatique » en français est un concept qui fait de plus en plus parler de lui dans le monde de l'informatique, et qui se rapporte au domaine de l'intelligence artificielle. Encore appelé « apprentissage statistique », ce terme renvoie à un processus de développement, d'analyse et d'implémentation conduisant à la mise en place de procédés systématiques. Pour faire simple, il s'agit d'une sorte de programme permettant à un ordinateur ou à une machine un apprentissage automatisé, de façon à pouvoir réaliser un certain nombre d'opérations très complexes.

L'objectif visé est de rendre la machine ou l'ordinateur capable d'apporter des solutions à des problèmes compliqués, par le traitement d'une quantité astronomique d'informations. Cela offre ainsi une possibilité d'analyser et de mettre en évidence les corrélations qui existent entre deux ou plusieurs situations données, et de prédire leurs différentes implications **[25].**

2.2. Cycle de vie d'une implémentation d'un machine Learning :

2.2.1. Préparation initiale de données :

La bonne préparation de données est l'étape préalable clef dans un processus KDD (Knowledge Discovery in Databases). La préparation des données prend environ 60 à 80 % du temps consacré au processus d'extraction de données. Dans notre cas, cette première phase consiste à trouver un échantillon d'attaque types qui doit être représentatif de la sorte de données à classer.

2.2.1.1. Nettoyage : Cette phase consiste à :

- Éliminer les doublons et les erreurs de saisie : Les données redondantes peuvent se révéler gênants parce qu'ils vont donner plus d'importance aux valeurs

répétées. Une erreur de saisie pourra à l'inverse occulter une répétition.

- Vérifier l'intégrité de domaine : Un contrôle sur les domaines des valeurs permet de retrouver des valeurs aberrantes.
- Traiter les informations manquantes : C'est le terme utilisé pour désigner le cas où des champs ne contiennent aucune donnée. Parfois, il est intéressant de conserver ces enregistrements car l'absence d'information peut être une information. D'autre part, les valeurs contenues dans les autres champs risquent aussi d'être utiles.

2.2.1.2. Codage et normalisation

Ce sont les valeurs des champs des enregistrements de données. Ces données possèdent un type qu'il faut préciser. En effet, la plupart des méthodes sont sensibles aux données manipulées. Par exemple, certaines méthodes sont mises en défaut par les données continues alors que d'autres peuvent être sensibles à la présence de données discrètes :

- **Les données discrètes** : les données binaires ou logiques : 0 ou 1 ; oui ou non vrai ou faux, sont des données être bon client. Les données énumératives sont des données discrètes pour lesquelles il n'existe pas d'ordre défini à priori, par exemple : la couleur.
- **Les données énumératives ordonnées :** concernent les réponses à une enquête d'opinion (1: très satisfait ; 2 : satisfait ; ...), les données issues de la discrétisation de données continues (1 : solde moyen < 2000 ; 2 : 2000 £ solde moyen < 5000 ;...)

- **Les données continues :** ce sont les données entières ou réelles : l'âge, le revenu moyen, ... mais aussi les données pouvant prendre un grand nombre de valeurs ordonnées.

Pour les applications en fouille de données (datamining) ,il est fréquent de les transformer en données continues ou en données énumératives ordonnées. Nous transformons une date de naissance en âge entier ou en une variable énumérative ordonnée correspondant à des tranches d'âge.

- **Les données textuelles :** un texte peut, pour certaines applications, être résumé comme un N-Uplet constitué du nombre d'occurrences dans le texte de mots clés d'un dictionnaire prédéfini. Comme elles peuvent être aussi transformés en données

entières.

L'étape de préparation de données peut inclure d'autre phases tels que la sélection de données qui consiste à obtenir des données en accord avec les objectifs que nous nous imposons et l'enrichissement qui se traduire par l'ajout de nouveaux champs en conservant souvent le même nombre d'enregistrements (objet)**[26].**

2.2.2. La modélisation et la mise en production

Une fois les données prétraitées, il va s'agir de trouver le bon algorithme. De nombreux choix d'algorithmes d'apprentissage et de leurs hyper paramètres s'offrent aux Data Scientists. La nature du problème à résoudre permet en partie de guider ce choix. Ce choix doit donc être fait en fonction des résultats désirés ainsi que de la complétude des données

Les facteurs qui peuvent entrer en jeu pour choisir le bon algorithme peuvent être nombreux ,notamment le nombre de caractéristiques (*features),*la quantité de données qu'on a…etc.

Une autre distinction qui nous aidera dans le choix d'un algorithme de machine Learning est le type de sortie que l'on attend de notre programme : est-ce une valeur **continue** (un nombre)ou bien une valeur **discrète** (une catégorie) ? Le premier cas est appelé une **régression**, les second une **classification[27].**

- **Classification**:

Les méthodes de classification s'appliquent lorsque l'ensemble des valeurs résultats est discret. Ceci revient à attribuer une classe (aussi appelée étiquette ou label) pour chaque valeur d'entrée. Les techniques de classification peuvent être basées sur des hypothèses probabilistes (exemple, naïf bayésien), des notions de proximité (exemple, *k* plus proches voisins) ou des recherches dans des espaces d'hypothèses (exemple, arbres de décisions). Le choix de la technique convenable est important ; il faut pouvoir choisir la méthode la plus adapté qui sera capable de séparer au mieux les données d'apprentissage.

- **Régression :**

Les méthodes de régression s'appliquent lorsque le résultat que l'on cherche à estimer est une valeur continue. En ML, la régression est un outil important de l'apprentissage

supervisé pour la modélisation et l'analyse des données. Elle est notamment utilisée en statistique et en économie.

Régression linéaire :

On appelle modèle de régression tout modèle capable à établir une relation linéaire entre une variable, dite expliqué ou dépendante, et une ou plusieurs variables, dite explicatives ou variables indépendantes. Le but principal c'est d'ajuster une meilleure droite représentée par une équation linéaire $Y=(X)+\varepsilon$ afin de prédire $\hat{Y}=f(X)$ pour une valeur de X quelconque.

− Y représente les variables dépendantes.

− X représente les variables indépendantes.

− ε représente le terme d'erreur ou perturbation.

La régression linéaire est principalement divisée en deux catégories : la régression linéaire simple et la régression linéaire multiple. La régression linéaire simple est caractérisée par une variable indépendante.

Par contre, la régression linéaire multiple est caractérisée par au moins de deux variables indépendantes.

Régression logistique :

La régression logistique a été développée par le statisticien David Cox en 1958. Le modèle logistique est un modèle statistique qui utilise une fonction logistique, il est également utilisé dans les problèmes de classification. La régression logistique ne nécessite pas de relation linéaire entre les variables dépendantes et indépendantes, mais elle nécessite des échantillons de grande taille afin d'avoir plus de précision lors de l'estimation de la vraisemblance **[28].**

2.3. Phase d'apprentissage:

En machine Learning, l'algorithme se construit une "représentation interne" afin de pouvoir effectuer la tâche qui lui est demandée(prédiction ,identification ,etc.).

Pour cela, Il Va d'abord falloir lui entrer un jeu de données d'exemples afin qu'il puisse s'entraîner et s'améliorer, d'où le mot apprentissage**.** On sépare donc les données entre un jeu d'entraînement**,** sur lequel on apprend le modèle (Ce jeu de données s'appelle le training set.) Et un jeu de test**,** sur lequel on l'évalue**.** On peut appeler une entrée dans le

jeu de données une instance ou une observation**[27].**

2.3.1. Méthodes d'apprentissage :

La machine Learning implique deux principaux systèmes d'apprentissage qui définissent ses différents modes de fonctionnement. Il s'agit de

2.3.1.1. Apprentissage supervisé :

L'apprentissage supervisé consiste à utiliser un ensemble de données pour prédire des événements futurs statistiquement probables, c'est-à-dire qu'il frome un modèle de prédiction à partir des évènements déjà prédits auparavant.

2.3.1.2. Apprentissage non supervisé :

Dans l'apprentissage non supervisé il n'y a pas de valeurs de sortie, il s'agit de trouver des structures cachées à partir d'un ensemble de données qui doivent être regroupé d'où le terme «clustering ». Le but de ce type d'apprentissage est de séparer les données en groupes ou en catégories.

Clustering :

Le clustering est une technique d'apprentissage automatique non supervisé, utilisé pour le regroupement des données non étiquetées dans de nombreux domaines. Si on dispose d'un nombre fini de points de données et on cherche à les classer dans des groupes de sorte que chaque groupe contient des points de données ayant des propriétés et/ou caractéristiques similaires. Le problème principal qui se pose dans ces algorithmes c'est le choix des propriétés à prendre en compte au cours du regroupement. L'un des algorithmes de clustering les plus utilisés est le « *K*-Means ».

2.3.2. Algorithmes de classification :

2.3.2.1 Naïf Bayésien :

La classification naïve bayésienne repose sur l'hypothèse que toutes les caractéristiques sont conditionnellement indépendantes les unes des autres. Cette méthode est basée sur le théorème de Bayes qui calcule la probabilité d'un événement à l'aide de la connaissance au préalable des conditions connexes. Ce théorème a été découvert par un statistique en anglais, Thomas Bayes, au 18ème siècle mai il n'a jamais publié son travail. Après son décès, ses notes ont été éditées et publiées par le mathématicien Richard Price.

Le théorème est donné par la formule suivante :

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- A et B sont des événements.
- (A) est la probabilité d'observer l'événement A
- $P(B)$ est la probabilité d'observer l'événement B.
- $P(A \mid B)$ est la probabilité conditionnelle d'observer A, sachant qu'un autre événement B de probabilité non nulle s'est réalisé.

Dans un problème de classification, notre tâche est de trouver l'étiquette la plus probable A, étant donné les caractéristiques B, le théorème de Bayes devient :

$$P(y|x_1, \dots, x_n) = \frac{P(x_1,\dots,x_n|y)P(y)}{P(x_1,\dots,x_n)}$$

Où n représente le nombre de caractéristiques, y est l'évènement qu'on cherche à classer. Par conséquent, en tenant compte de l'hypothèse d'indépendance, Bayes prédit la classe qui constitue la probabilité la plus élevée **[25]** :

$$\hat{y} = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

2.3.2.2 .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.

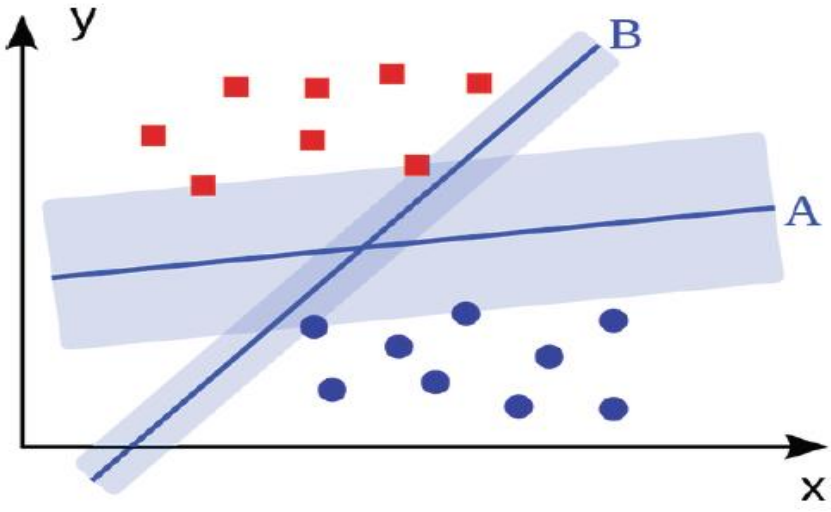


Figure 2.1 : Support Vector Machine

2.3.2.3.k-plus proches voisins :

Parmi les algorithmes d'apprentissage automatique les plus basiques, le *k*-plus proche voisin, souvent abrégé en *k*-NN où *k* est un entier positif. En Data Science, cet algorithme est largement utilisé pour les problèmes de classification des données. Mais avant de se lancer dans cette méthode, il faut savoir que les calculs peuvent s'avérer très couteuses en temps de calcul, ainsi, les données doivent être prétraitées. Cette méthode peut être également utilisée dans les problèmes de régression.

Prenons par exemple le problème de classification suivant : Dans le diagramme ci-dessous, il y a des objets ronds verts et des objets carrés bleus. Ceux-ci appartiennent à deux classes différentes : la classe des ronds et la classe des carrés. Lorsqu'un nouvel objet est inséré dans l'espace - dans ce cas, un cercle rouge – nous voulons que l'algorithme d'apprentissage automatique classe le cercle dans une certaine classe.

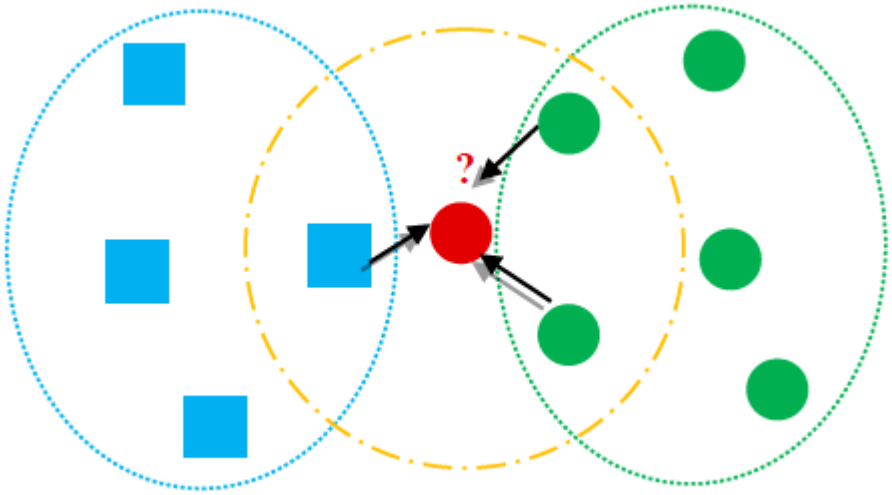


Figure 2.2 : exemple de classification par la méthode des k-NN

Si on choisit $k = 3$, l'algorithme cherche les trois plus proches voisins du cercle rouge pour pouvoir le classer soit dans la classe des cercles, soit dans la classe des carrés. Dans ce cas, les trois plus proches voisins du cercle rouge sont un carré et deux cercles. Par conséquent, l'algorithme classera la sphère dans la classe des cercles.

Dans la méthode de *k*-NN, le résultat est l'appartenance à une classe. L'algorithme stocke tous les cas disponibles et classe tout nouvel objet en vérifiant ses *k*-plus proches voisins. Ensuite, l'objet est attribué à la classe avec laquelle il a le plus en commun.

2.3.2.2. Arbres de décision :

Les arbres de décision (AD) sont un outil de classification très utilisé. Son principe repose sur la construction d'un arbre de taille limitée. Considérons l'exemple simple représenté ci-dessous:

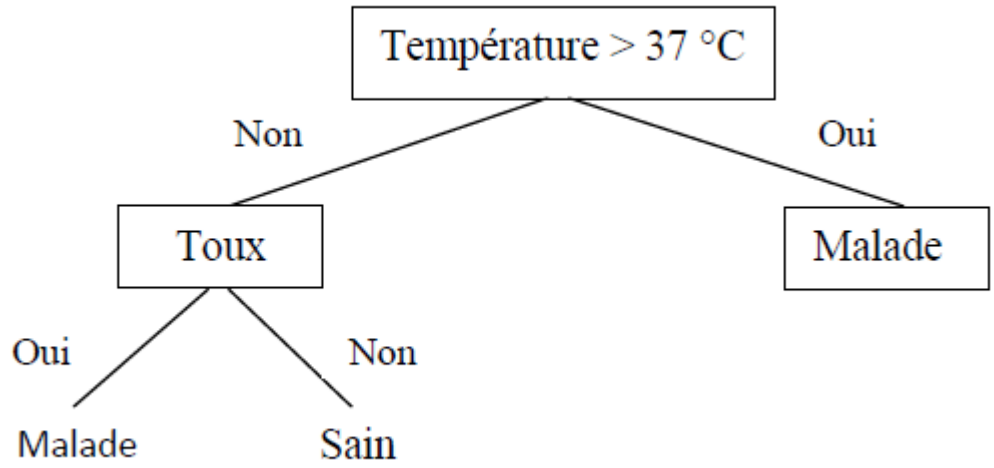


Figure 2.3 : schéma d'un arbre de décision

Dans cet exemple, le sommet « Température > 37 °C » représente la racine. Les feuilles correspondent aux classes ou décisions (appelées aussi nœuds terminales), ici « malade » ou « sain ». De plus, on appelle « Température > 37 °C » et « Toux » les attributs ou les variables (appelés aussi nœuds intermédiaires).

Les AD jouent un rôle très important dans ML. Ils sont capables de gérer les variables continues et discrètes et fournissent par la suite une indication claire pour la prédiction ou la classification sans effectuer beaucoup de calcul. Les AD sont si simples à comprendre et à interpréter, qu'ils permettent une meilleure approximation quel que soit la complexité des données. L'arbre le plus simple est souvent le meilleur, à condition que tous les autres arbres possibles produisent les mêmes résultats. Leur construction se réalise en divisant l'arbre du sommet vers les feuilles (du haut vers le bas) en choisissant à chaque étape une variable de séparation sur un nœud d'où les critères de segmentation. Pour cela, les algorithmes les plus utilisés sont « la classification et arbre de régression » et « le Dichotomiser itératif3».

2.3.2.3. Réseaux de neurones artificiels :

Le terme réseau de neurones est une référence à la neurobiologie. Originalement, ce concept est inspiré du fonctionnement des neurones du cerveau humain, apprend essentiellement de l'expérience.

Le fonctionnement exact du cerveau humain est encore un mystère, mais certains aspects sont connus. En particulier l'élément fondamental du cerveau est un type spécifique de cellule incapable de se régénérer. Ces cellules nous fournissent la capacité de nous rappeler, de penser et de réagir en basant sur des expériences antérieures. Le corps humain comporte 100 milliards de ces cellules appelées neurones. Chacun de ces neurones se connecte à 200 milles autres neurones. Les réseaux de neurones artificiels tentent de reproduire que les éléments les plus fondamentaux de cet organisme complexe et puissant.

Un réseau de neurones est une organisation hiérarchique de neurones connectés entre eux. Ces derniers transmettent un message ou un signal à d'autres neurones en fonction des paramètres d'entrés reçus et forment un réseau complexe. Ci-dessous une représentation simpliste d'un réseau de neurones de base **[28] :**

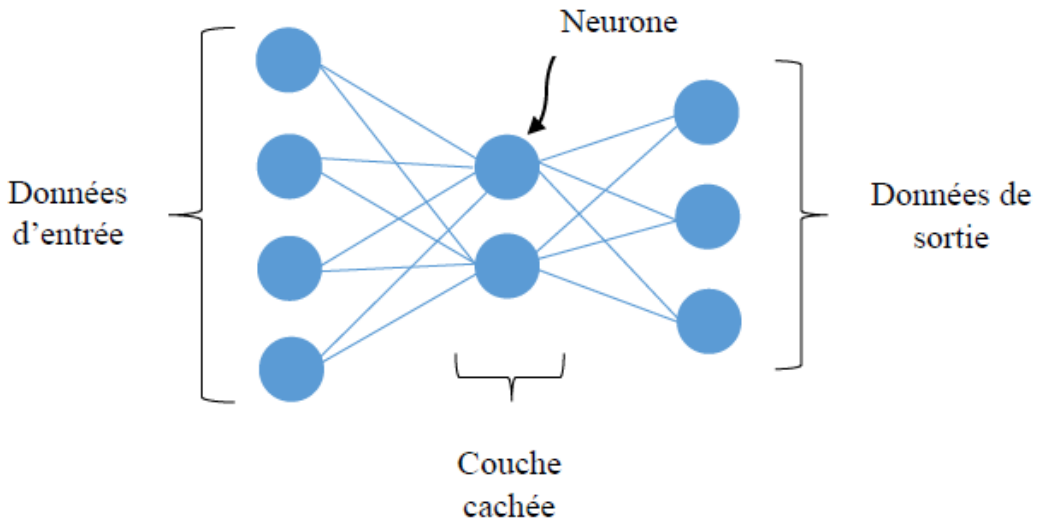


Figure 2.4 : Exemple simple d'un réseau de neurones artificiel.

2.4. Phase de validation:

Durant cette phase, on va tester et valider le modèle et ses paramètres selon des critères sebasantsursesrésultats.Ilpermetd'obtenirlemeilleurmodèlegénéralisantlesdonnéesobtenues lors de la phase d'apprentissage. Pour cela, on a un ensemble d'exemples pour l'apprentissage et un autre pour les tests. Voici quelques méthodes pour les tests :

- ✓ **Hold-out :** On coupe aléatoirement l'ensemble des informations en deux groupes : groupe d'apprentissage et groupe de tests.
- ✓ **Leave one out:** Cette méthode sort de l'ensemble des informations une donnée en particulier la laisse de côté ,puis construit le modèle avec celles restantes et on évalue la structure avec l'exemple laissé de côté. On répète le processus pour chacune des données de l'ensemble de données. Ainsi, on peut avoir une moyenne globale de la précision du modèle.
- ✓ **Cross-validation:** Cette méthode réalise un partitionnement des données de manière aléatoire en n groupes. On utilise une partition comme un ensemble de test et le reste pour former celui d'entraînement. Comme précédemment, on applique un algorithme à l'ensemble d'entraînement et on évalue le modèle résultant sur celui de tests. On répète ce processus pour chaque partition et on regarde l'erreur moyenne. (Cette méthode sera celle que nous allons utiliser par la suite)**[27].**

2.5. Performance du modèle

Après avoir déroulé son algorithme sur ses données d'entraînement *(Training set)* et faire des prédictions avec le jeu de test *(Test Set)*, il est temps d'évaluer la performance de notre algorithme.

On mesure la performance du modèle sur la base de test. Il existe certaine façon standard de le faire en fonction des types de problèmes:

Classification:

- Matrice de confusion
- Courbe ROC
- Précision/rappel

Régression:

- Erreur de prédiction
- Graphe XY valeur à prédire/valeur prédite

Clustering:

- Variance intraclasse,interclasse
- Nombre d'arc coupés

Conclusion

Une bonne connaissance des données est nécessaire pour concevoir des modèles efficaces. Le machine Learning est l'étape finale du processus d'analyse des données. Avant de considérer faire des prédictions, il faut déjà savoir collecter les données, les prétraiter les stocker, les visualiser pour mieux les connaître et enfin seulement il est intéressant d'envisager le machine Learning. Le prochain chapitre va expliquer l'utilisation du machine Learning dans le domaine des détections d'intrusions : notre méthodologie de travail.

Chapitre 3

Méthodes d'apprentissage en détection d'intrusion

Introduction :

Dans la littérature, diverses approches sont proposées pour arriver à une classification plus performante parmi elles les méthodes d'ensemble. Dans ce chapitre nous avons présenté quelques algorithmes d'apprentissage supervisés, à savoir le Naïve Bayes et Arbre de décision. Nous exposons après les différentes techniques de combinaison des classifieurs et nous concentrons sur l'AdaBoost en l'étudiants avec deux classifieurs.

1. Etat de l'art :

AdaBoost Ensemble avec les algorithmes naïfs de Bayes et J48pour discriminer les fausses et authentiques informations des réseaux sociaux :

[Mehmet Bozuyla,21] a utilisé les algorithmes conventionnels naïve bayes(NB) et C4.5 pour détecter les fakes news en turc. Afin d'améliorer les résultats obtenus, Il est proposé une approche d'ensemble AdaBoost. Les résultats expérimentaux montrent que l'apprentissage d'ensemble AdaBoost peut améliorer les performances des algorithmes en termes de précision de classification.

Les données des fausses nouvelles sont obtenues à partir de la célèbre source de fausses nouvelles Zaytung (Github, 2021) et les vraies nouvelles sont collectées à partir de Hurriyet. Les données se composent de 2163 fausses nouvelles et de 2296 vraies nouvelles.

Le dispositif expérimental se compose principalement de deux étapes : 1) la sélection d'algorithmes d'apprentissage automatique largement utilisés dans la littérature et 2) la génération de contrepartie d'ensemble AdaBoost d'algorithme de l'étape (1).

Pour la première étape, il a sélectionné Naïve Bayes (NB) et C4.5, il a ensuite obtenu des précisions basées sur une validation croisée 10 fois pour la détection des fausses nouvelles.

Dans la deuxième étape, il a généré des ensembles AdaBoost à partir d'algorithme sélectionné et réévalué les performances des ensembles sur la base d'une validation croisée 10 fois.

Toute comparaison de classification est réalisée à l'aide de mesures d'évaluation et de validation telles que l'exactitude (Acc) et le Kappa (Kp).

Il présente les résultats expérimentaux de l'approche proposée en termes d'Acc et de Kp. Les résultats expérimentaux montrent que l'ensemble AdaBoost de l'algorithme NB et AdaBoost de l'algorithme C4.5sont améliorés Par rapport aux naïves bayes et C4.5.

2. Classifieurs à tester

2.1. Les arbres de décision.

L'arbre de décision est un algorithme de classification dans lequel l'espace d'instance est exprimé sous la forme d'une partition récursive. Il composé de nœuds formant un nœud racine, ce qui signifie qu'il s'agit d'un arbre orienté sans bords vers l'intérieur. Tous les autres les nœuds d'un arbre ont précisément 1 bord intérieur. Chaque nœud interne est divisé en 2 ou plusieurs sous-espaces en fonction de certaines fonctions discrètes des valeurs des attributs d'entrée (Chen et al., 2003). Dans le plus simple et cas le plus fréquent, chaque test considère un seul attribut ; de sorte que l'espace de l'instance soit partitionné en correspondent à la valeur de chaque attribut (Maimon et Rokach 2010).

Un arbre est obtenu à partir d'algorithmes tels que la Classification And RegressionTree (CART), Iterative Dichotomiser 3 (ID3) C4.5 (Quinlan 1993) est le successeur de ID3 (Iterative Dichotomiser 3) (Quinlan 1979). C4.5 profondément surpasse ID3. Il peut recevoir à la fois des caractéristiques nominales et numériques pour les ensembles de données d'entrée **[29].**

2.1.1. Algorithmes d'induction d'arbres de décisions

a. Généralités

Ces algorithmes construisent automatiquement les arbres de décisions à partir d'une base de données. Ce sont des algorithmes « *top – down* » car la construction se fait de la racine jusqu'aux feuilles. La construction d'un arbre de décision s'effectue en 2 phases : le growing phase et le pruning phase. Pour arrêter la construction d'un arbre, un algorithme utilise des critères d'arrêts ou stopping criteria.

b. Growing phase

Elle correspond à la phase de croissance de l'arbre. Cette phase permet d'obtenir un arbre optimal. Elle est caractérisée par l'utilisation d'une fonction de sélection d'attribut appelée splitting criteria qui permet de choisir l'attribut à tester sur un noeud. La plupart des algorithmes d'arbres de décisions utilise une approche appelée « divide and conquer » qui effectue une partition récursive de la base de données à étudier.

c. Phase d'élagage « *pruning* phase »

Elle permet d'obtenir un arbre réduit qui classifie convenablement les nouveaux événements. Cet arbre est obtenu à partir d'un arbre optimal. Elle consiste à éliminer les feuilles et/ou les subtrees qui ne contribuent pas à la classification de nouveaux événements. Les principales tâches du pruning phase sont le subtree raising et le subtree replacement. Le subtree raising consiste à remplacer un subtree par un autre plus simple et plus précis et le subtree replacement consiste à remplacer un subtree par une feuille. Ces méthodes sont utilisées par C4.5.

Afin d'arrêter le processus de construction d'un arbre de décision, des conditions d'arrêt sont utilisées. Ces conditions empêchent la création de règles de décisions qui ne couvrent qu'un nombre restreint d'événements.

2.1.2.Avantages et inconvénients des arbres de décisions

a. Avantages

- Un arbre de décision offre une représentation simple qui facilite sa compréhension. Cette simplicité le rend accessible aux utilisateurs non experts.
- Un arbre de décision peut être construit à partir d'une la base de données contenant des attributs nominaux et/ou des attributs numériques continus et/ou discrets. De plus, un arbre de décision peut être construit malgré les imperfections de la base de données
- Un arbre de décision dépend du contenu de la base de données mais non de son format.

b. Inconvénients

- Certains algorithmes sont sélectifs car ils ont été conçus pour traiter uniquement des attributs de type nominaux ou des attributs discrets. C'est le cas de l'algorithme ID3.

- La méthode « divide and conquer » peut entrainer la réplication de subtrees dans un arbre. Cette réplication surcharge inutilement la représentation d'un arbre et réduit sa performance de classification.
- Un arbre de décision est sensible aux bruits. La modification des événements d'une base de données peut produire un arbre totalement différent. Les algorithmes d'arbres de décisions traditionnels sont par conséquent inadaptés pour les données variant dans le temps**[30].**

2.1.3. Algorithme :

En général, l'algorithme d'arbre de décision se présente de la façon suivante :

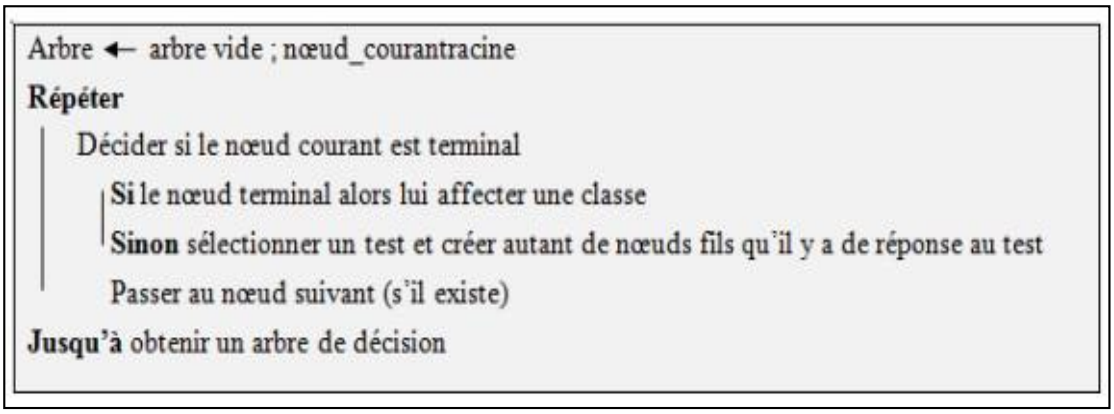


Figure 3.1 : L'algorithme d'arbre de décision [31]

2.1.4. Les forêts aléatoires

Une forêt aléatoire est essentiellement une collection d'arbres de décision, où chaque arbre est légèrement différent des autres. L'idée derrière les forêts aléatoires est que chaque arbre peut faire un assez bon travail de prévision, mais qu'il va probablement sur-apprendre une partie des données. Si nous construisons plusieurs arbres, qui fonctionnent tous bien et se surajoutent de différentes façons, nous pouvons réduire le nombre de surajouts en faisant la moyenne de leurs résultats. Cette réduction du surajustement, tout en conservant le pouvoir de prédiction des arbres, peut être démontrée par des calculs mathématiques rigoureux.

Pour mettre en œuvre cette stratégie, nous devons construire de nombreux arbres de décision. Chaque arbre doit permettre de prévoir l'objectif de manière acceptable et doit également être

différent des autres arbres. Les forêts aléatoires tirent leur nom de l'injection de hasard dans la construction des arbres pour s'assurer que chaque arbre est différent. Les arbres d'une forêt aléatoire sont randomisés de deux manières : en sélectionnant les points de données utilisés pour construire un arbre et en sélectionnant les caractéristiques de chaque test de division.

Les forêts aléatoires est l'algorithme le plus connu de la technique d'apprentissage ensembliste « bagging » ou « bootstrapaggregating » qui consiste à créer plusieurs entités d'un même modèle(plusieurs arbres de décisions) et d'entrainer chacun de cet arbre sur une portion aléatoire d'une collection de données, après avoir entrainer chaque arbre, nous pouvons regrouper les résultats de chaque arbre afin d'effectuer la prédiction **[32].**

2.2. La classification bayésienne :

Naive Bayes est un algorithme simple et puissant pour la modélisation prédictive. Le modèle comprend deux types de probabilités qui peuvent être calculées directement à partir des données d'apprentissage : (i) la probabilité de chaque classe et (ii) la probabilité conditionnelle pour chaque classe compte tenu de chaque valeur x. Une fois calculé, le modèle de probabilité peut être utilisé pour faire des prédictions pour de nouvelles données en utilisant le théorème de Bayes. Lorsque les données sont à valeurs réelles, il est courant de supposer une distribution gaussienne (courbe en cloche) afin que l'on puisse facilement estimer ces probabilités. Naive Bayes est appelé naïf car il suppose que chaque variable d'entrée est indépendante. Il s'agit d'une hypothèse forte et irréaliste pour des données réelles ; cependant, la technique est très efficace sur un large éventail de problèmes complexes. L'idée derrière la classification naïve de Bayes est d'essayer de classer les données en maximisant $P(O|C_i)P(C_i)$ en utilisant le théorème de probabilité a posteriori de Bayes (où O est l'objet ou le tuple dans un ensemble de données et "i" est un indice de la classe). Les étapes de mise en œuvre du classificateur de Bayes sont les suivantes

Étape 1 : Soit D un ensemble d'entraînement de tuples ou d'objets O et leurs étiquettes de classe associées. Chaque tuple est représenté par un vecteur d'attributs à n dimensions. O = (o_1, o_2, …, o_n) représente n mesures effectuées sur le tuple à partir des attributs A_1, A_2, A_3, …, An, respectivement.

Étape 2 : En supposant qu'il y a m classes, C_1, C_2, …, C_n, étant donné un tuple O, le classificateur prédira que O appartient à la classe ayant la probabilité a posteriori la plus

élevée conditionnée sur O. Cela signifie que le classificateur bayésien naïf prédit que tuple O appartient à la classe C_i si et seulement si

$P(C_i \mid O) > P(C_j \mid O)$pour $1 \leq j \leq n, j \neq i$.

Autrement dit, nous maximisons $P(C_i \mid O)$. La classe Ci pour laquelle $P(C_i \mid O)$ est maximisée est appelée hypothèse maximum a posteriori. En appliquant le théorème de Bayes,

$$P(C_i \mid O)=\frac{P(O \mid Ci)\, P(Ci)}{P(O)}$$

Étape 3 : Comme P(O) est constant pour toutes les classes, seul $P(O \mid C_i)P(C_i)$ doit être maximisé. Si les probabilités a priori des classes ne sont pas connues, alors on suppose que les classes sont équiprobables, c'est-à-dire que $P(C_1) = P(C_2) = \ldots = P(C_n)$ et on maximiserait donc $P(O \mid\mid C_i)$. Sinon, il faut maximiser $P(O \mid\mid C_i)P(C_i)$. De plus, les probabilités a priori de classe peuvent être estimées par $P(C_i) = |C_iD|/|D|$, où $|C_iD|$ est le nombre de tuples d'apprentissage de classe Ci dans D.

Étape 4 : Maintenant, compte tenu de l'ensemble de données avec de nombreux attributs, il serait coûteux en calcul de calculer $P(O \mid C_i)$. Pour réduire le calcul d'évaluation de $P(O \mid C_i)$, l'hypothèse naïve d'indépendance de classe est faite. Cela suppose que les valeurs d'attribut sont conditionnellement indépendantes les unes des autres, compte tenu de l'étiquette de classe de l'objet (c'est-à-dire qu'il n'y a pas de relations de dépendance entre les attributs). Ainsi,

$P(O \mid C_i)=P(o_1 \mid C_i)*P(o_2 \mid C_2)* \ldots *P(o_n \mid C_i)$

Étape 5 : Pour prédire l'étiquette de classe de O, $P(O \mid Ci)P(Ci)$ est évalué pour chaque classe Ci. Le classificateur prédit que l'étiquette de classe du tuple O est Ci si et seulement si

$P(O \mid C_i)\, P(C_i) > P(O \mid C_j)P(C_j)for1 \leq j \leq n, j \neq i$

On peut estimer les probabilités $P(O_1 \mid C_i)$, $P(O_2 \mid C_2)$, ..., $P(O_n \mid C_i)$ à partir des tuples d'apprentissage.

Par exemple, un fruit peut être considéré comme une pomme s'il est rouge, rond et d'environ 10 cm de diamètre. Un classificateur bayésien naïf considère chacune de ces caractéristiques

indépendamment de la probabilité que le fruit donné soit une pomme, indépendamment de toute corrélation possible entre la couleur, la forme et le diamètre.[33]

Parmi les avantages des méthodes Naïve Bayes on peut citer par exemple :

1. La facilité et la simplicité de leur implémentation.
2. Leur rapidité.
3. Les méthodes Naïve Bayes donnent de bons résultats **[34].**

3. Approche ensembliste

Les méthodes d'ensembles constituent une famille ou un ensemble d'algorithmes qui génèrent une collection de classifieurs et agrègent leurs prédictions, l'objectif visé est que le prédicteur final soit meilleur que chacun des prédicteurs individuels.

Au lieu d'essayer d'optimiser une méthode, les méthodes ensemblistes génèrent plusieurs règles de prédiction et mettent ensuite en commun leurs différentes réponses. L'heuristique de ces méthodes est qu'en générant beaucoup de prédicteurs, on explore grandement l'espace des solutions, et qu'en agrégeant toutes les prédictions, on récupère un prédicteur qui rend compte de toute cette exploration.

Pour qu'une méthode d'ensemble soit performante, elle doit réussir à construire une collection de prédicteur qui vérifie ces deux points :

- Chaque prédicteur individuel doit être relativement bon.
- Les prédicteurs individuels doivent être différents les uns des autres.

3.1. Quelques techniques ensemblistes

3.1.1. Bagging

Bagging est une méthode d'ensemble introduite par Breiman 3 en 1996 est basée sur les concepts de Bootstrapping.Le Bootstrap est un principe de rééchantillonnage statistique traditionnellement utilisé pour l'estimation de grandeurs ou de propriétés statistiques. Le bootstrapping est conçu pour générer au hasard et avec remise L copies indépendantes de S objets appelées bootstrap à partir de l'ensemble initial des échantillons d'apprentissage de taille S. Un objet de la base initiale peut être sélectionné plusieurs fois comme il peut être

absent dans les copies générées. Le même classifieur est appris sur chacune des copies. On obtient par la suite L classifieurs avec des performances différentes**[35]**.

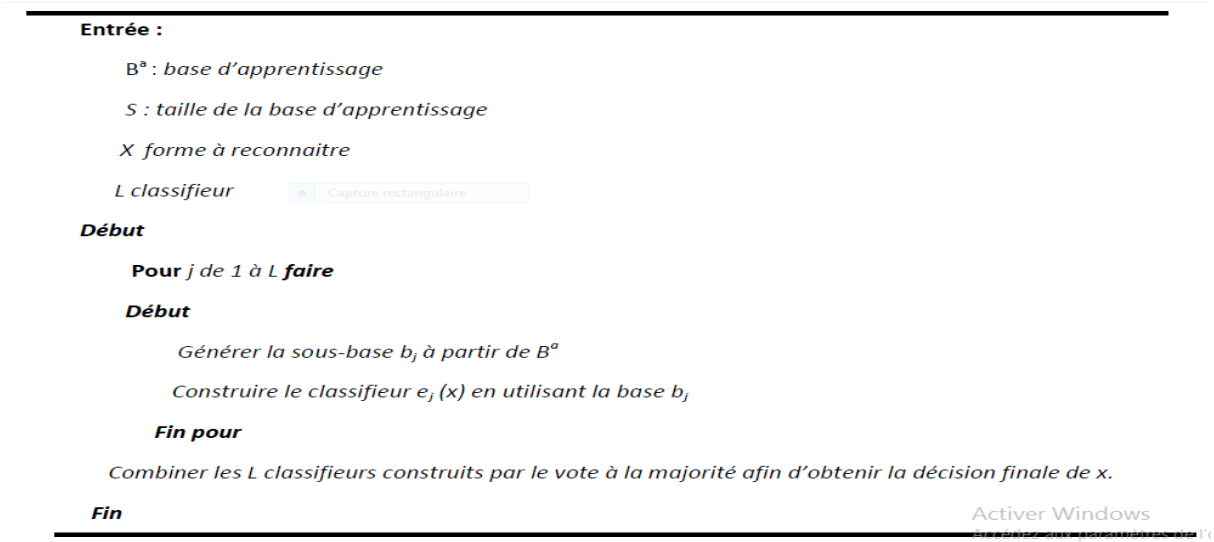
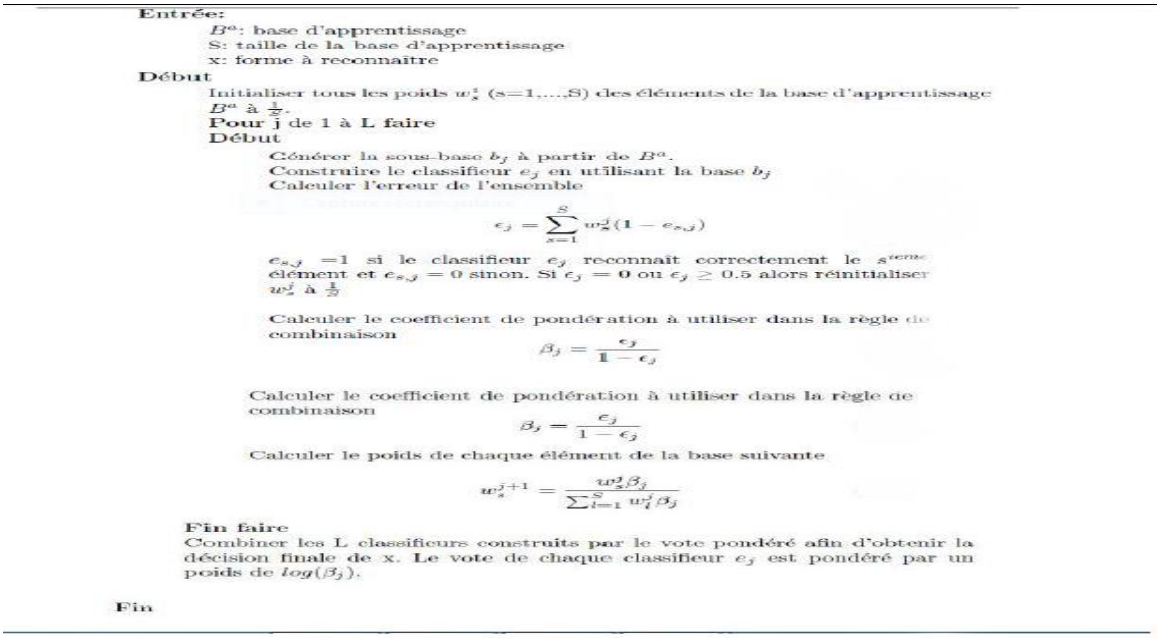


Figure 3.2 : Algorithme la méthode bagging**[35]**

3.1.2. Boosting

Le "Boosting" désigne un principe général d'apprentissage permettant d'améliorer la précision d'un algorithme d'apprentissage donné. Le principe général est de combiner linéairement des résultats de classificateurs dits "faibles" afin de construire un classificateur "fort" d'apprentissage à partir de l'ensemble original et une méthode de combinaison de classificateurs construits à partir de chaque nouvel ensemble.

Pour définir sa nouvelle technique de "Boosting" se base sur l'idée que tout classificateur faible capable d'apprendre avec une certaine confiance et une erreur de classification inférieure à (0.5), peut être transformer en un classificateur plus confiant et avec une erreur de classification aussi petite que désirée. A chaque itération, l'algorithme cherche à trouver un classificateur faible qui peut corriger au mieux les erreurs des classificateurs obtenus aux itérations précédentes. Dans le principe de "Boosting", cet objectif est réalisé à l'aide d'une pondération des données d'apprentissage**[35]**.



Entrée:
B^a: base d'apprentissage
S: taille de la base d'apprentissage
x: forme à reconnaître

Début
Initialiser tous les poids w_s^1 (s=1,...,S) des éléments de la base d'apprentissage B^a à $\frac{1}{S}$.
Pour j de 1 à L **faire**
Début
Générer la sous-base b_j à partir de B^a.
Construire le classifieur e_j en utilisant la base b_j
Calculer l'erreur de l'ensemble

$$\epsilon_j = \sum_{s=1}^{S} w_s^j (1 - e_{s,j})$$

$e_{s,j}$ =1 si le classifieur e_j reconnait correctement le $s^{ième}$ élément et $e_{s,j} = 0$ sinon. Si $\epsilon_j = 0$ ou $\epsilon_j \geq 0.5$ alors réinitialiser w_s^j à $\frac{1}{S}$

Calculer le coefficient de pondération à utiliser dans la règle de combinaison

$$\beta_j = \frac{\epsilon_j}{1 - \epsilon_j}$$

Calculer le coefficient de pondération à utiliser dans la règle de combinaison

$$\beta_j = \frac{\epsilon_j}{1 - \epsilon_j}$$

Calculer le poids de chaque élément de la base suivante

$$w_s^{j+1} = \frac{w_s^j \beta_j}{\sum_{i=1}^{S} w_i^j \beta_j}$$

Fin faire
Combiner les L classifieurs construits par le vote pondéré afin d'obtenir la décision finale de x. Le vote de chaque classifieur e_j est pondéré par un poids de $log(\beta_j)$.

Fin

Figure 3.3 : Algorithme la méthode boosting**[35]**

3.1.2.1. Adaboost

AdaBoost (Adaptive Boosting) est une méthode algorithmique connue dans le domaine de l'apprentissage. Cette méthode a été initialement conçue par Robert Schapire et Yoav Freund en 1995. Cette approche est la dérivée la plus pratiquée de la méthode du Bossting qui vise à stimuler la performance de l'algorithme d'apprentissage. Adaboost consiste à transformer, d'une maniéré efficace, un classifieur faible en un classifieur fort en réduisant les taux d'erreur.

L'algorithme d'Adaboost, appelle à chaque itération, un algorithme d'apprentissage qui en trainent les instances à classifier. Par la suite, Adaboost définit une nouvelle distribution de probabilité pour les instances d'apprentissage en fonction des résultats del'algorithme à l'itération précédente tout en augmentant le poids des instances qui ne sont pas correctement classées. A la fin, Adaboost combine les données faibles par un vote pondéré pour en déduire un classifieur fort. Dans cette optique, AdaBoost essaye, de trouver un classifieur optimal à partir de la combinaison d'un ensemble de données d'apprentissage faible **[35].**

Avantages :

_ Rapide ,simple et facile à implémenter.

_ Applicable a de nombreux domaines par un bon choix de classifieur faible.

_ Pas de sur apprentissage par la maximisation de la marge.

Inconvénients :

_ Parfois particulièrement sensible au bruit présent dans les données.

_ Importance du choix des classifieurs faibles, en particulier il ne faut pas que chaque classifieur faible pris individuellement soit trop "bon".

Algorithme : Adaboost

Entrées : Une base d'exemple $(x_1,y_1)\ldots(x_m, y_m)$, avec des labels $y_i \in \{1,1\}$

Pour *t=1,.....T faire*

-On trouve *le classifieur faible minimisant un critére d'erreur*
-*On calcule son poids*
-*On calcule une distribution D(i) des exemples*

FinPour

Sorties : On construit le classifieur fort en effectuant une somme pondérée des classifieurs séléctionnés

Figure 3.4 : Algorithme approche d'Adaboost**[35]**

3.1.3.Pourquoi Le choix d'Adaboost:

Adaboost est plus performant que le bagging et qu'il est moins risqué au cas du sur-apprentissage, car Adaboost essaye directement d'optimiser les votes pondérés. Cette théorie est prouvée par :

1. La diminution exponentielle de l'erreur empirique d'adaboost sur l'échantillon d'apprentissage avec le nombre d'itération.
2. la diminution de l'erreur en généralisation qui continue à baisser même lorsque l'erreur empirique a atteint son minimum.

Contrairement au Bagging qui demande un grand nombre d'itération pour que l'erreur en généralisation se stabilise, l'erreur de généralisation du Boosting ne diminue pas lorsque l'erreur en apprentissage a atteint son minimum. C'est ainsi que le Bagging est plus gourmand que le Boosting du point de vue d'espace mémoire puisque chaque classifieurs doit être stocké pour la prédiction d'un nouvel exemple.

Conclusion :

Dans ce chapitre, nous choisissons deux classifieurs pour tester afin de décider quel est le classifieur le plus approprié avec l'algorithme d'ensemble adaboost.

Au chapitre suivant nous montrons les aspects pratiques de l'expérimentation, à savoir les outils utilisés et quelques résultats de test avec l'ensemble des données KDD.

Chapitre 4

Test et implémentation

Introduction :

Après que nous avons présenté dans le chapitre précédent l'ensemble des classifieurs utilisés pour le test. Dans ce chapitre, nous allons tout d'abord présenter les outils utilisés pour la réalisation de notre méthodologie. Par la suite, nous définissons la base de test (KDDcup99), puis nous passons aux résultats expérimentaux et à leur comparaison avec quelques interfaces des résultats obtenus.

1. Outils de réalisation :

Nous allons présenter les principaux outils utilisés pour la mise en place de notre projet

1.1. Choix de langage de programmation python :

Python est un langage de programmation généraliste et très riche avec de nombreuses Fonctionnalités. C'est l'un des principaux objectifs d'un style de programmation connu Sous le nom de programmation orientée objet. Il s'agit d'un langage de programmation interprété de haut niveau [36].

1.2. Plateforme et environnement de développement :

Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur [39].

1.3. Bibliothèque Utilisé :

Nous avons utilisé scikit-learn qui est une bibliothèque d'apprentissage statique en python, c'est le moteur de beaucoup d'application de l'intelligence artificielle et de la science des données, elle est construite à base de :

- **Numpy** : est une bibliothèque Python open source pour le calcul scientique. NumPy vous permet de travailler avec des tableaux et des matrices de manière naturelle. La bibliothèque contient une longue liste de fonctions mathématiques utiles, dont certaines pour l'algèbre linéaire, la transformation de Fourier et les routines de génération de numéros [36].
- **Pandas** : est une bibliothèque Python open source pour l'analyse de données hautement spécialisées [36].
- **Matplotlib** : est la bibliothèque Python qui est actuellement la plus populaire pour produire des tracés et autres visualisations de données en deux dimensions. Comme l'analyse des données nécessite des outils de visualisation, elle est très répandue dans les milieux scientifiques et techniques [36].

2. Méthodologie de recherche :

Les étapes suivies / outils utilisés dans le cadre de la méthodologie de recherche sont les suivantes :

-Un ordinateur LENOVO Intel ® Core ™ CPUN5005ayant4GodeRamestutilisépourla démonstration de ce projet

-Python 3 est utilisé comme langage de programmation

-Le jeu de données KDD 99 est sélectionné.

-Scikit learn est choisi pour la simulation

-J48, naive Bayes et adaboost sont utilisés comme classifier dans notre projet et classent les instances en attaque ou en normale

-Le prétraitement du fichier de training et de test avec 42 attributs est effectué afin d'être utilisé pour les calculs associés à chaque algorithme de classification.

3. Choix du Dataset :

Il s'agit de l'ensemble de données utilisé pour le troisième concours international d'outils de découverte de connaissances et d'exploration de données, qui s'est tenu conjointement avec KDD-99, la cinquième conférence internationale sur la découverte de connaissances et l'exploration de données. La tâche du concours consistait à construire un détecteur d'intrusion réseau, un modèle prédictif capable de distinguer les « mauvaises » connexions, appelées intrusions ou attaques, et les « bonnes » connexions normales. Cette base de données contient un ensemble standard de données à auditer, qui comprend une grande variété d'intrusions simulées dans un environnement de réseau militaire [41].

Classes d'attaques de l'ensemble KDD99

Les principales classes d'attaques de l'ensemble de données KDD99 sont

- Probing: surveillance et sondage.
- DOS (Denial Of Service) : déni de service.
- R2L (Remote to User): accès non autorisé à partir d'une machine distante.
- U2R (User to Root): accès non autorisé pour avoir le privilège d'un administrateur[38].

Déni de service – « Denial-of-service (DOS) » :

Il s'agit d'empêcher par tous les moyens les utilisateurs de se servir des ressources disponibles en temps normal. Ces attaques à but purement « destructeur » et sont souvent très simple à mettre en place et donnent une sensation de puissances à l'attaquant, ce qui expliquent leur fréquence. Un exemple d'attaques DOS et le Smurf qui provoque un déni de service via des requêtes d'écho ICMP manipulées à une adresse de diffusion d'un réseau [37].

« Remote TO Local Acces »(R2L):

Ce type d'attaque essaye d'exploiter la vulnérabilité du système afin de contrôler la machine distante. Comme exemple d'attaque R2L, il y a celle qui visent les failles des protocoles IMAP (Internet Message Access Protocole). Ces protocoles permettent à des utilisateurs d'accéder à leurs comptes de courrier depuis des réseaux internes ou externes [37].

« User To Root Attacks »(U2R):

Ou l'attaquant essaye d'avoir les droits d'accès à partir d'un poste afin d'accéder aux systèmes. Un exemple d'attaque U2R est Rootkit, qui après avoir obtenu un accès ROOT pour l'intrus, remplace les commandes systèmes afin qu'il puisse revenir quand il le souhaite en tant que root (administrateur)[37].

Reconnaissance-probing :

Ces actions ne sont pas vraiment des attaques puisqu'elles ne sont pas « de structrice» au sens ou 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. Un exemple d'outils de reconnaissances probing est Satan (Security Administrateur Tool for Analyzing Net-work), qui est un analyseur de ports TCP/IP qui recherche sur des hôtes distants les failles de sécurité et les défauts de configuration courants [37].

4. Fonctionnement :

L'analyse du jeu de données KDD 99 est réalisée à l'aide de divers algorithmes de classification disponibles dans l'outil d'exploration de données **scikit-learn.** L'ensemble de données KDD 99 est analysé et classé en 2 groupes différents.de données : données normales et les données représentant des attaques. Une étude analytique approfondie est réalisée sur le jeu de données de test et de training.

L'ensemble de données KDD99 (apprentissage et test) est fourni sous forme de de fichier csv. Ce qui suit est un schéma montrant les étapes de notre Expérimentations, que nous expliquerons un peu plus tard

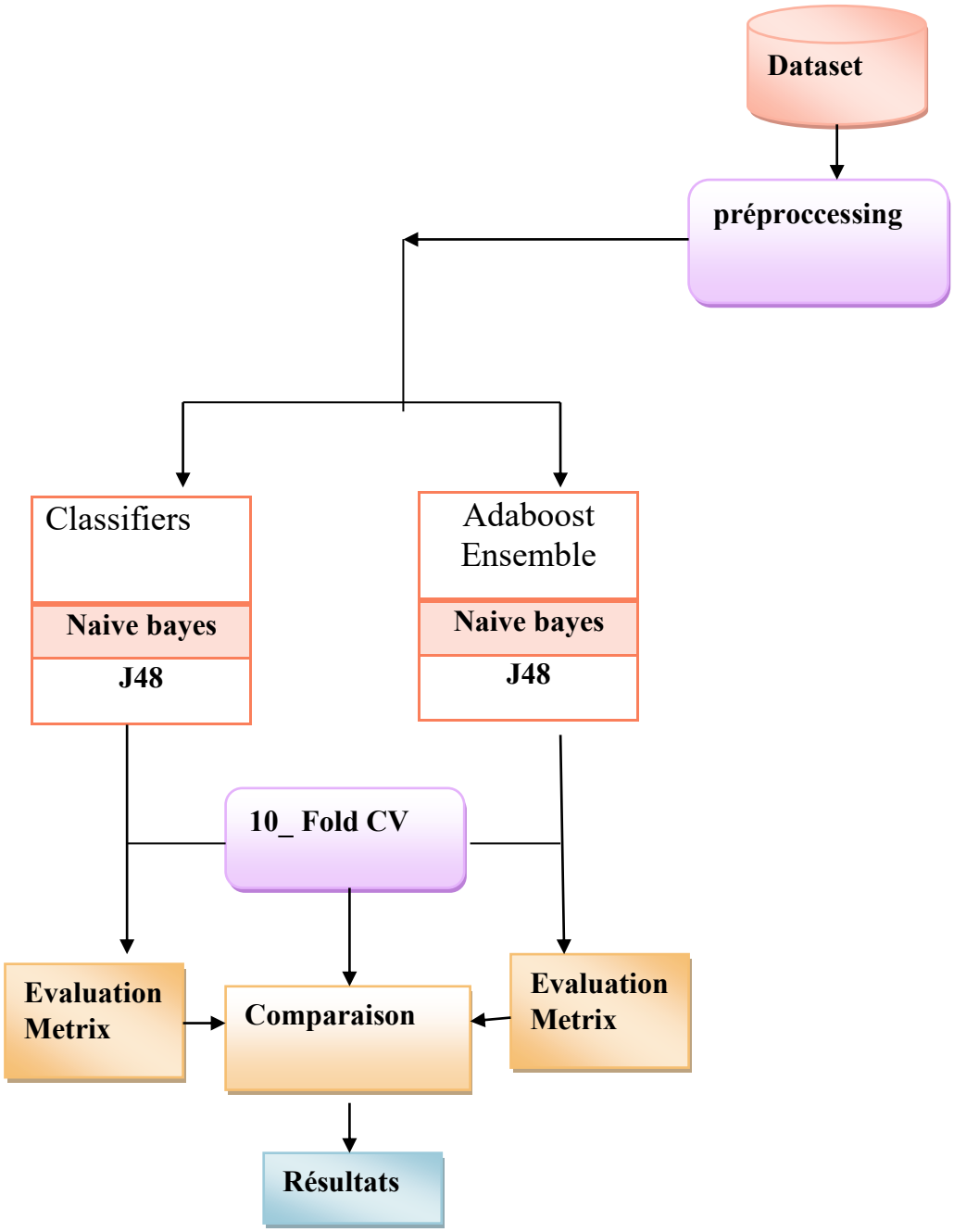


Figure 4.1 : Schéma de fonctionnement général du système

4.1. Chargement des données du dataset :

Pour commencer, il faudra lire et charger les données contenues dans le fichier csv. Python propose via sa librairie Pandas des classes et fonctions pour lire divers formats de fichiers notamment les fichiers .csv La fonction *read_csv ()*, renvoie un DataFrame contenant les données du dataset.

```
dataset = pd.read_csv('/kd99.csv', names=features, header=None)
dataset.head(10)
```

Figure 4.2 : Chargement des données.

Après on affiche les premières lignes (10 premières par exemple)chargées depuis notre fichier csv(a comma-separated values)

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_c
0	0	tcp	http	SF	181	5450	0	0	0	0	...	
1	0	tcp	http	SF	239	486	0	0	0	0	...	
2	0	tcp	http	SF	235	1337	0	0	0	0	...	
3	0	tcp	http	SF	219	1337	0	0	0	0	...	
4	0	tcp	http	SF	217	2032	0	0	0	0	...	
5	0	tcp	http	SF	217	2032	0	0	0	0	...	
6	0	tcp	http	SF	212	1940	0	0	0	0	...	
7	0	tcp	http	SF	159	4087	0	0	0	0	...	
8	0	tcp	http	SF	210	151	0	0	0	0	...	
9	0	tcp	http	SF	212	786	0	0	0	1	...	

10 rows × 42 columns

Figure 4.3 : Affichage des 10 premières lignes du dataset.

4.2. Prétraitement de données :

Le rôle de ce module est de préparer les données pour qu'elles soient directement exploitables par les différents modules de traitement (apprentissage, validation et classification).

Les attributs de l'ensemble de données KDD99 sont un mélange d'attributs continus, discrets et symboliques avec d'intervalles de valeurs très variés. La plupart des techniques de classification ne sont pas capable de traiter un tel format de données. Pour cela le

prétraitement est nécessaire avant la construction du modèle de classification

Il est constitué de plusieurs étapes qui s'exécutent successivement (ou alternativement si nécessaire) :

- Élimination des enregistrements redondants :la base KDD99 contient un grand nombre d'enregistrements dupliqués.
- Numérisation des différents champs symboliques et transformation des enregistrements en un format de traitement adéquat.
- Scalage de données (scaling data).[38]

4 .2.1.Étape d'élimination de redondances :suppression des enregistrements dupliqués

```
print('Null values in the dataset are: ',len(dataset[dataset.isnull().any(1)]))
dataset.drop_duplicates(subset=features, keep='first', inplace=True)
dataset.shape
```

```
Null values in the dataset are:  0
(145585, 42)
```

Figure 4.4 : Suppression des enregistrements dupliqués.

4.2.2.Étape Numérisation de données

Cette étape consiste à convertir les attributs symboliques en numériques. La procédure est comme suit :

En premier, Les noms d'attaques (comme buffer_overflow, guess_passwd, etc.) sont rangés dans une des deux classes : 0 pour Normal, 1 pour attaque

Les attributs symboliques comme protocol_type (3 symboles différents), service (70 symboles différents) et flag (11 symboles différents) sont transformés à des valeurs entières de 0 à N-1 où N est le nombre de symboles.

```
dataset.loc[(dataset['label']!='normal','label')]='attack'
dataset.loc[(dataset['label']=='normal','label')]='normal'
```

```
input_cols = list(dataset.columns)[1:-1]
target_col = 'label'
numeric_cols = dataset.select_dtypes(include=np.number).columns.tolist()[:-1]
```

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaler.fit(dataset[numeric_cols])
```

```
dataset[numeric_cols] = scaler.transform(dataset[numeric_cols])
```

```
from sklearn.preprocessing import LabelEncoder,normalize

le = LabelEncoder()

target = dataset['label']
dataset['label'] = le.fit_transform(target)
dataset['protocol_type'] = le.fit_transform(dataset['protocol_type'])
dataset['service'] = le.fit_transform(dataset['service'])
dataset['flag'] = le.fit_transform(dataset['flag'])
```

Figure 4.5 : Numérisation de données

4.2.3. Normalisation des données et features Scaling :

Notre exemple comporte des variables prédictives avec **des ordres de grandeurs très différents**.

Pour appliquer les algorithmes de classification, il est nécessaire que les variables prédictives faisant partie du modèle prédictif soient **du même ordre de grandeur.** Pour ramener nos variables prédictives au même ordre de grandeur, nous appliquerons un procédé qui s'appelle: **features scaling.**

La librairie Scikit learn de Python propose plusieurs classes et méthodes pour faire de la préparation de données (Data pré-processing) pour les algorithmes de Machine Learning. Le package sklearn. preprocessing propose la classe Standard Scaler qui permettra de faire du features Scaling surtoutes nos variables prédictives.

Les données bien propres peuvent maintenant commencer à être explorées. Cette étape vouspermet de mieux comprendre les différents comportements et de bien saisir le phénomènesous-jacent.

C'est vraiment une étape à ne pas négliger, les meilleurs data Scientists ne sont pas ceux quiconnaissent les algorithmes les plus complexes mais ceux qui ont une très bonne connaissancedesdonnées, et ont préparéleterrainavecsoin en amont.[40]

```
#feature scaling
sc_x = StandardScaler()
train_in = sc_x.fit_transform(train_in)
test_in = sc_x.transform(test_in)
```

Figure 4.6 :featuresscaling.

4.3.Partition de la base KDD cup99 :

Nous avons partitionné KDD cup99 en un sous ensemble d'apprentissage et un sous ensemble detest selon les pourcentages suivants :

70% Pour l'apprentissage

30% Pour l'apprentissage

4.4. Représentationdesdonnées :

Pour mieux comprendre les données et voir quelques statistiques, il est souvent utile de lesvisualiser. On peut représenter les données dansunespaceavec la librairieMatplotlib,[39]voiciun exemple de représentation graphique pour la fréquence d'apparition des attaques dans ledatasettrain et dans labasedetest

```
plt.figure(figsize=(20,15))
class_distribution = dataset['label'].value_counts()
class_distribution.plot(kind='bar')
plt.xlabel('Class')
plt.ylabel('Data points per Class')
plt.title('Distribution of yi in train data')
plt.grid()
plt.show()
```

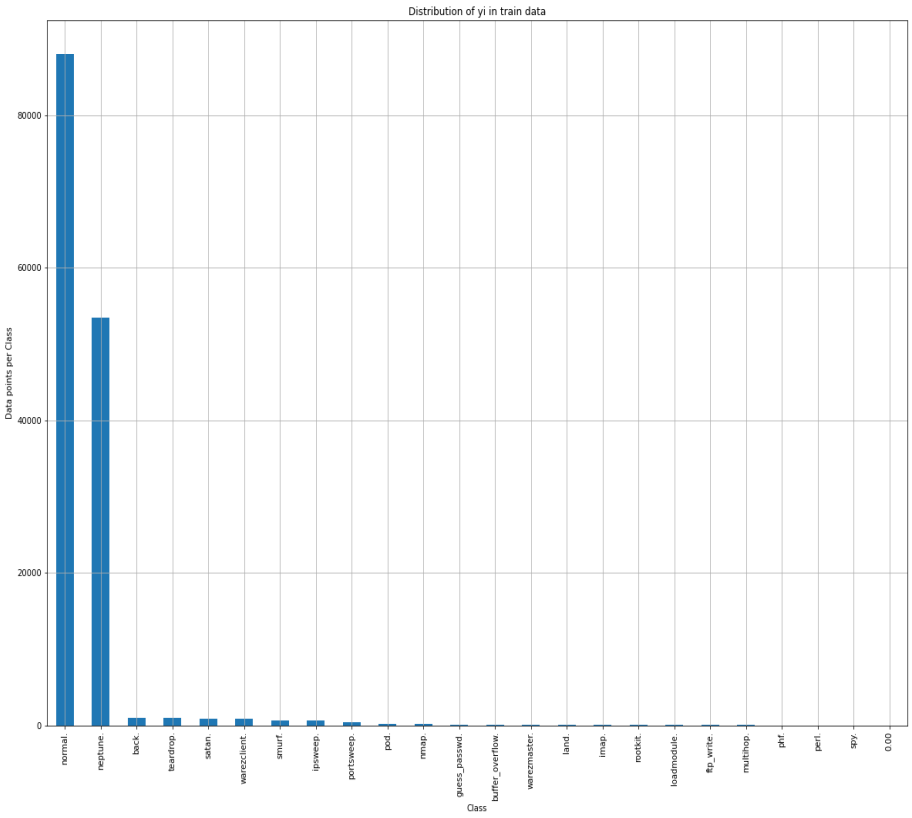


Figure 4.7 : Représentation des données.

4 .5.Apprentissage de l'algorithme par les données :

Nous allons enfin pouvoir appliquer nos algorithmes de classification, pour chaque algorithme nous allons montrer les différentes fonctions utilisées :

❖ **Arbre de décision(J48) :**

```
from sklearn.tree import DecisionTreeClassifier
import time
MBModel=DecisionTreeClassifier(criterion="entropy",max_depth=10,random_state=0)
train0 = time.time()
MBModel.fit(train_in,train_t)
train1 = time.time() - train0
```

Figure 4.8. Algorithme arbre de décision

❖ **Naive Bayes:**

```
from sklearn.naive_bayes import GaussianNB
import time
MBModel=GaussianNB()
train0 = time.time()
MBModel.fit(train_in,train_t)
train1 = time.time() - train0
```

Figure 4.9. Algorithme NaiveBayes

❖ **Arbre de décision(J48)+ Adaboost:**

```
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier

from sklearn import metrics
import time
nb=DecisionTreeClassifier(criterion="entropy",max_depth=10,random_state=0)
```

```
#create adaboost classifier object
abc=AdaBoostClassifier(base_estimator=nb,learning_rate=1 ,algorithm="SAMME",n_estimators=10)
```

Figure 4.10. Algorithme arbre de décision+Adaboost

❖ **NaiveBayes + Adaboost :**

```
from sklearn.ensemble import AdaBoostClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn import metrics
import time
nb=GaussianNB()
```

```
#create adaboost classifier object
abc=AdaBoostClassifier(base_estimator=GaussianNB(),learning_rate=1 ,algorithm="SAMME",n_estimators=10)
```

Figure 4.11. Algorithme NaiveBayes+Adaboost

4.6. Métriques utilisées :

4.6.1. Cross validation : Le moyen le plus simple d'utiliser la validation croisée consiste à appeler la fonction d'assistance cross_val_score sur l'estimateur et le jeu de données[40].

```
# Applying k-Fold Cross Validation
from sklearn.model_selection import cross_val_score
cross_val_score(AdaBoostClassifier(),train_in,train_t, cv = 10).mean()
```

Figure 4.12 : cross validation

4.6.2. Accuracy:

```
#model accuracy how ofen the classifier correct
print("accuracy:",metrics.accuracy_score(test_t,t_pred))
```

Figure 4.13 : Accuracy

4.6.3. Confusion matrix :

Par définition, une matrice de confusion est telle qu'elle est égale à un nombre d'observations Connues pour être dans le groupe mais prévues pour être dans le groupe.

Ainsi, dans la classification binaire, le nombre de vrais négatifs est C 0,0, les faux négatifssontC1, 0, les vrais positifs C1, 1 et les faux positifs C0, 1.

```
print(confusion_matrix(test_t,t_pred))
```

Figure 4.14 : Affichage de la matrice de Confusion

4.6.4. Classification Report:

Résumé textuel de la précision, rappel, score F1 pour chaque classe.[40]

```
print(classification_report(test_t,t_pred))
```

Figure 4. 15 : Affichage du Report de classification

4.7. Expérimentations :

4.7.1. Chaque classifieur unique

Classification par J48

Classifieur	Accuracy	Cross_Validation	Temps d'apprentissage	Temps de test
J48	0.998	0.998	0.327s	0.009s

Tableau 4 .1 :Résultats de Classification par J48

Classification par Naïve Bayes :

Classifieur	Accuracy	Cross_Validation	Temps d'apprentissage	Temps de test
Naïve Bayes	0.976	0.973	0.068s	0.022s

Tableau 4 .2:Résultats de Classification par Naïve Bayes

4.7.2. Chaque classifieur avec Adaboost

Classification par J48+ Adaboost

classifieur	Accuracy	Cross_Validation	Temps d'apprentissage	Temps de test
J48+adaboost	0.999	0.996	4.430s	0.069s

Tableau 4 .3:Résultats de Classification par J48+ Adaboost

Classification par Naïve Bayes + Adaboost

classifieur	Accuracy	Cross_Validation	Temps d'apprentissage	Temps de test
NaïveBayes +adaboost	0.976	0.996	1.424s	0.194s

Tableau 4 .4:Résultats de Classification par Naïve Bayes+adaboost

4.8.Tableaurécapitulatif :

	J48	Naïve Bayes	J48 + Adaboost	NaïveBayes +Adaboost
précision	**0.997**	**0.973**	**0.999**	**0.973**
recall	**0.999**	**0.987**	**0.999**	**0.987**
F1_score	**0.998**	**0.980**	**0.999**	**0.980**

Tableau 4 .5:Résultats de Classification de tous les cas

4.9. Discussion :

Selon les résultats des métriques obtenus par les classifieurs, le classifieur arbre de décision (J48) est le plus efficace par rapport au NaiveBayes que ce soit individuellement ou avec l'AdaBoost. Nous recommandons donc la combinaison des classifieurs J48 en utilisant l'algorithme d'AdaBoost.

Pour le temps d'exécution (spécialement le temps d'apprentissage) le classifieur Naive Bayes est plus rapide comparé avec J48 dans tous les cas.

4.10. Affichage des résultats :

```
1 print(confusion_matrix(test_t,t_pred))
2
```

```
[[17297    21]
 [   10 26348]]
```

```
1 print(classification_report(test_t,t_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     17318
           1       1.00      1.00      1.00     26358

    accuracy                           1.00     43676
   macro avg       1.00      1.00      1.00     43676
weighted avg       1.00      1.00      1.00     43676
```

Figure 4 .16: Classification par J48+ Adaboost

```
print(confusion_matrix(test_t,t_predicted))
```

```
[[17261    57]
 [   23 26335]]
```

```
print(classification_report(test_t,t_predicted))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     17318
           1       1.00      1.00      1.00     26358

    accuracy                           1.00     43676
   macro avg       1.00      1.00      1.00     43676
weighted avg       1.00      1.00      1.00     43676
```

Figure 4 .17: Classification par J48

```
1 print(confusion_matrix(test_t,t_pred))
2
```

```
[[16609   709]
 [  337 26021]]
```

```
1 print(classification_report(test_t,t_pred))
```

```
              precision    recall  f1-score   support

           0       0.98      0.96      0.97     17318
           1       0.97      0.99      0.98     26358

    accuracy                           0.98     43676
   macro avg       0.98      0.97      0.97     43676
weighted avg       0.98      0.98      0.98     43676
```

Figure 4 .18: Classification par Naive+ Adaboost

```
1 print(confusion_matrix(test_t,t_predicted))
```

```
[[16609   709]
 [  337 26021]]
```

```
1 print(classification_report(test_t,t_predicted))
```

```
              precision    recall  f1-score   support

           0       0.98      0.96      0.97     17318
           1       0.97      0.99      0.98     26358

    accuracy                           0.98     43676
   macro avg       0.98      0.97      0.97     43676
weighted avg       0.98      0.98      0.98     43676
```

Figure 4 .19: Classification par Naive

5. Conclusion :

Dans ce chapitre, Nous avons tester les deux classifieurs naive Bayes et J48 en présentant différentes métriques de performance et nous avons faire la comparaison de l'algorithme d'ensemble AdaBoost avec les deux classifieurs sur la base de test KDD99

Les résultats obtenus montrent que la qualité de classification dépend du choix de classifieurs de base et elle est meilleure avec le classifieur J48.

Conclusion générale et perspectives

Le travail présenté dans ce mémoire est lié au domaine de la sécurité des réseaux, plus précisément de la détection d'intrusions. Nous nous sommes intéressés à la détection d'intrusions partir des données du trafic réseau.
Nous avons abordé les méthodes d'ensemble pour la détection des intrusions. Plus précisément, nous avons adopté la méthode de boosting et particulièrement l'algorithme AdaBoost avec quelques techniques d'apprentissage automatique comme le NaiveBayes, les arbres de décision.

Le but de ce mémoire est d'obtenir le meilleur modèle pour les données de KDDcup99.Les résultats des expérimentations montrent que le classifieur J48 est meilleure que naive Bayes en termes de précision et que l'ensemble de classifieurs avec l'algorithme AdaBoost est la meilleure façon pour détecter les intrusions par rapport à l'utilisation d'un seul classifieur.

En perspective à ce travail, il est possible de tester l'algorithme d'adaboost en collaboration avec d'autres classifieurs, supervisés et non-supervisés, en utilisant la même base de données.

Bibliographie

[1] :https://fr.m.wikipedia.org

[2] : M. THIBAUT PROBST, «Evaluation et analyse des mécanismes de sécurité des réseaux dans les infrastructures virtuelles de cloud COMPUTING», thèse de doctorat, Institut National Polytechnique de Toulouse (INP Toulouse), 2015.

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

[4] : CherfiSarra, Détection d'intrusions via des réseaux de neurones optimisés par des métaheuristiques, Diplôme de master, Université Mohamed Seddik Ben Yahia de Jijel, 2019-2020.

[5] : EricFiliol, les virus informatiques, Springer Verlag, 2009.

[6] : MIHOUBI MOHAMED et MEDJANI NACER, Sécurisation d'une infrastructure LAN/WAN A base d'équipement Cisco ,MASTER ACADEMIQUE, **U**NIVERSITE **M**OULOUD **M**AMMERI DE **T**IZI-**O**UZOU, 2015.

[7] :laurent Bloch, CristophWolfhugel, Sécurité informatique principes et méthodes, Eyrolles, 2007.

[8] : Guy Pujolle, Les réseaux locaux, Eyrolles, 2003.

[9] : Mr. MIMOUNE Zakarya,Développement d'une Architecture Basée sur l'Apprentissage Profond (Deep Learning) pour la Détection d'Intrusion dans les Réseaux, diplôme de Master, Université Ahmed Draia – Adrar,2019.

[10] : Guillaume Desgeorge, La sécurité des réseaux, 2000.

[11] : Amakou M'BATA, Olivier PERSENT, Firewall ,pare-feux, Mur de feu, 2006.

[12] : Bruno M, la sécurité informatique CERAM, << Fondamentaux des sciences de l'information>>.

[13] : IdresLouiza ,Système de détection d'intrusion hybride et hiérarchique pour les réseaux de capteur sans fil,diplôme de master, Université Mouloud Mammeri de Tizi-Ouzou,20112012.

[14] : ChikoucheSoumia, système de détection d'intrusion basé sur la classification comportementale des processus, Université de M'Sila,2011/2012,21.

[15] : Mémoire online .Mise en place d'IDS en utilisant Snort . Disponible sur<https://www.memoireonline.com/01/17/9516/Mise-en-place-dun-IDS-en-utilisant-Snort.html>.

[16] : Liran LERMAN, Les systèmes de détection d'intrusion basés sur du machine learning, UNIVERSITÉ LIBRE DE BRUXELLES.

[17] : DABOUR Imane, Etude et mise en place d'un système de détection/prévention d'intrusion (IDS/IPS) réseau. Etude de cas SNORT,Diplôme de licence, Université Abou BakrBelkaid– Tlemcen,2013/2014.

[18] :http://www-igm.univ-mlv.fr/~dr/XPOSE2004/IDS/IDSSnort.html.

[19] :https://stringfixer.com/fr/Intrusion_Detection.html.

[20] :http://staff.univbatna2.dz/sites/default/files/benbrahim_meriem/files/chapitre-1-introduction-a-l-intelligence-artificielle.pdf

[21] :https://www.futura-sciences.com/tech/definitions/informatique-intelligence-artificielle-555/

[22] :https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501845-intelligence-artificielle-forte-definition/

[23] :https://docs.microsoft.com/fr-fr/azure/machine-learning/concept-deep-learning-vs-machine-learning

[24] : https://www.sas.com/fr_ca/insights/analytics/what-is-artificial- intelligence

[25] :https://digitalinsiders.feelandclic.com/machine-learning-definition

[26] : M^{elle}AISSAOUISihem, Apprentissage automatique et sécurité des systèmes d'information, Université d'Oran Es-senia ,diplôme de master, 2007-2008.

[27] :douidrania, système de détection d'intrusion réseau basé sur l'algorithme de classification KNN ,Université SAAD DAHLAB DE BLIDA ,diplôme de master ,2018-2019

[28] : Mlle GUENNINECHE Amel,Prédiction des propriétés des matériaux par apprentissage automatique, Université ABOU_BEKR BELKAID _TLEMCEN , diplôme de master.

[29]: Jamal Hussain, Samuel Lalmuanawma,AN INTELLIGENT HYBRID DECISION APPROACH WITH FEATURE SELECTION FOR ANOMALY NETWORK INTRUSION DETECTION SYSTEM, Mizoram University Department of Mathematics & Computer Science, Mizoram, Aizawl, 796004, India, 2014

[30] : RalambomahayaAndriamparanyValim-bavaka ,Conception d'un arbre de décision applique au data Mining , Ecole supérieure polytechnique d'antananarivo, diplôme d'ingénieur on éléctronique,2011/2012

[31] : AhemedZeggada,Rabah Moulai ,catégorisation automatique des textes arabe , Université SAAD DAHLEB de Blida,diplôme de Mater,2019/2020

[32] : Sami sadi ,Implémentation et évaluation d'un modèle d'apprentissage automatique pour l'estimation de la valeur marchande de propriétés immobilières,Université Mouloud Mammery de Tizi Ouzo ,diplôme de Mater , 2019/2020

[33] :https://www.sciencedirect.com/topics/mathematics/naive-bayes.

[34] :HASSANE HILALI, APPLICATION DE LA CLASSIFICATION TEXTUELLE POURL'EXTRACTION DES RÈGLES D'ASSOCIATION MAXIMALES, UNIVERSITÉ DU QUÉBEC,2009.

[35] :DERKAOUIAmina et DERKAOUI Soumia, ETUDE COMPARATIVE DES METHODES ENSEMBLISTES DE CLASSIFICATION DES DONNES MEDICALES, MASTER en GENIE BIOMEDICAL, Université Abou BakrBelkaïd de Tlemcen, 2016-2017.

[36] : Mlle. BELKHAMSA Amel, Mlle. YAHIAOUI Manel, <<La protection de la vie privée dans le Big Data>>, Master2 en Informatique, Universit_e AMO de Bouira, 2019/2020.

[37] : Mr. Ahmed ZEBOUDJ, Mr. nassim KHOBZI, <<CONCEPTION D'UN SYSTEME DEDETECTION D'INTRUSION BASE SUR UNARBRE DE DECISION>>, Diplôme de Master en informatique, UNIVERSITEMOULOUDMAMMERI DE TIZI-OUZOU, 2014/2015.

[38] :M^{elle}AISSAOUI Sihem, Apprentissage automatique et sécurité des systèmes d'information, Université d'Oran Es-senia ,diplôme de master, 2007-2008.

[39] :https://ledatascientist.com/google-colab-le-guide-ultime/

[40] : Douid Rania, système de détection d'intrusion réseau basé sur l'algorithme de classification KNN ,Université SAAD DAHLAB DE BLIDA ,diplôme de master ,2018-2019.

[41] :http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[Mehmet Bozuyla,21] : Mehmet Bozuyla,Pamukkale University, Faculty of Enginnering, Departmant of Electrical and Electronics Engineering, Denizli, Turkey, (ORCID: 0000-0002-7485-6106),2021.