République algérienne démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique

Université 20 Aout 1955-SKIKDA

Faculté des Mathématiques, d'Informatique des Sciences

Département d'informatique

Mémoire Fin D'étude

En vue de l'obtention du diplôme de Master professionnel en Informatique

Option : Réseaux et Systèmes Distribues (RSD)

Thème

Système de détection d'intrusions basé sur L'apprentissage automatique.

Réalisé par :

- ***BOUFFEENECHE Yasmina***
- ***MESSALLAOUI Amina***

Encadré par :

- ***Pr: MAZOUZI Smaine***

Année Universitaire: 2023/2024


بسم الله الرحمن الرحيم

Remerciements

Nous tenons à remercier tout d'abord Allah de nous avoir donné la force, le courage et la patience et la santé pour réaliser ce modeste travail.

A notre Encadrant

Nous exprimons nos plus chaleureux et profonds remerciements à notre directeur de recherche, Monsieur MAZOUZI Smaine, pour ses précieux conseils, ses encouragements et ses orientations. Nous sommes également reconnaissants pour l'aide et le temps qu'il a constamment offerts tout au long de l'élaboration de ce mémoire.

A notre membres du Jury

Nous remercions également les membres du jury d'avoir accepté d'honorer notre travail par leur jugement.

A notre collaboration

Nous souhaitons exprimer nos sincères remerciements pour notre collaboration exceptionnelle tout au long de cette expérience académique.Avoir eu l'opportunité de travailler ensemble sur ce projet passionnant a été un véritable privilège. Notre étroite coopération et nos échanges d'idées ont ouvert de nouvelles perspectives, nous permettant d'accomplir bien plus que ce que nous avions espéré.

A nos Famille

Nous tenons à exprimer notre profonde gratitude à nos parents qui nous ont soutenu tout au long de ce projet.

Dédicaces

A mes Parents

À ma source de joie, ceux qui ont toujours veillé sur mon bonheur, qui ont sacrifié pour me voir réussi et qui m'ont comblé tant d'amour et de tendresse, à mes chers parents . Ils ont été toujours présents à mes côtés par leurs sacrifices et leurs prières. Que Dieu leur procure une longue vie avec une bonne santé.

A ma Famille

Je dédie ce travail à ma famille, mon marie, mes chères enfants J'espère être à la hauteur de leurs attentes et ne jamais les décevoir.

A mon Encadrant

A mon encadrant Monsieur MAZOUZI Smaine, vous m'avez beaucoup aidédans cette recherche cordialement je vous souhaite longue vie.

À tous ceux qui, par un mot, m'ont donné la force de continuer…

À tous ceux qui m'aiment et que j'aime…

BOUFFENECH Yasmina

Dédicaces

A mes chères parentes

Je voudrais vous exprimer ma profonde gratitude et mes sincères remerciements pour tout le soutien et l'encouragement que vous m'avez apportés tout au long de mon parcours académique. Grâce à vous et à votre amour ainsi qu'à votre confiance en moi, j'ai pu surmonter les défis et réaliser cette grande réussite dans ma vie.

A ma famille :

Je dédie ce modeste travaille à ma famille, qui a été mon roc et ma source inépuisable de soutien. Vous encouragements constants et votre amour inconditionnel m'ont permis d'atteindre ce jalon. Cette thèse est dédiée à vous tous." Spécialement mon enfant ***Djoud****, parti trop tôt mais dont l'amour continue de ma guider chaque jour. Tu vis éternellement dans le cœur de ta tante. Qu'Allah ait pitié de vous*

A mon professeur :

À mon encadrant monsieur Mazouzi smaine , pour son soutien indéfectible, sa sagesse inspirante et ses conseils éclairés qui ont guidé chaque pas de ce travail.

Merci pour votre mentorat précieux.".

Mon mémoire de maîtrise n'est pas juste un document académique, c'est le fruit de mes efforts et le résultat d'un soutien inestimable de votre part. merci a tout mes amis , À tous ceux qui, affection et mon soutien morale .

Avec toute ma gratitude à tout.

AMINA Messallaoui

Résumé

Les attaques deviennent de plus en plus fréquentes, ciblant les données stockées sur Internet comme jamais auparavant. Assurer la protection des informations sensibles est donc d'une importance cruciale. Dans ce cadre, notre projet vise à développer une méthode de détection des intrusions réseau en utilisant l'apprentissage automatique. Plus précisément, nous proposons de mener une étude comparative entre les algorithmes KNN et SVM afin d'identifier et prévenir les attaques quotidiennes sur les réseaux. Cette approche innovante représente une solution prometteuse pour détecter et atténuer efficacement les menaces de cybersécurité. Pour notre étude, nous utilisons la base de données **NLS-KDD** comme référence.

Mots-clés :

Sécurité des Réseaux, IDS, apprentissage automatique, KNN, SVM, NLS-KDD.

Summary

Attacks are becoming increasingly frequent, targeting data stored on the Internet more than ever before. Ensuring the protection of sensitive information is therefore of crucial importance. In this context, our project aims to develop a network intrusion detection method using machine learning. Specifically, we propose to conduct a comparative study between the KNN and SVM algorithms to identify and prevent daily network attacks. This innovative approach offers a promising solution for effectively detecting and mitigating cybersecurity threats. For our study, we use the **NLS-KDD** dataset as a reference.

Keywords:

Network Security, IDS, Machine Learning, KNN, SVM,NLS- KDD.

ملخص

تزداد الهجمات تواترًا، مستهدفة البيانات المخزنة على الإنترنت أكثر من أي وقت مضى. لذلك، فإن ضمان حماية المعلومات الحساسة أمر ذو أهمية قصوى. في هذا السياق، يهدف مشروعنا إلى تطوير طريقة لكشف التسلل إلى الشبكة باستخدام التعلم الآلي. بشكل محدد، نقترح اجراء دراسة مقارنة بين خوارزية KNN و SVMلتحديد ومنع الهجمات اليومية على الشبكات. هذه المقاربة المبتكرة تمثل حلاً واعدًا للكشف عن تهديدات الأمن السيبراني والتخفيف منها بفعالية. نستخدم قاعدة بيانات**NLS-KDD**كمرجع لدراستنا.

الكلمات الرئيسية:

أمن الشبكات، كشف التسلل، التعلم الآلي، KNN، SVM، NLS-KDD

Table des matières

Liste des Figures

Liste des Tableaux

Liste des abréviations :

IDS	*intrusions détection system.*
DOS	*Denial of Service.*
U2R	*User to Root.*
R2L	*Remote to Local.*
VPN	*Virtual Private Network.*
KNN	*K Nearest Neighbors.*
SVM	*Support Vector Machine.*
NSL-KDD	*Network Security Layer-Knowledge Discovery in Databases.*
TN	*True Negative.*
FN	*False Negative.*
FP	*False Positive.*
TP	*True Positiv*

Introduction Général

Introduction Générale

Internet est devenu un outil indispensable et une source d'information majeure dans notre monde actuel. Il joue un rôle essentiel dans les domaines éducatif, créatif et commercial. Toutefois, avec cette dépendance croissante à Internet, la protection des données contre les intrusions est devenue une priorité absolue.

La sécurité sur Internet est l'une des principales préoccupations de notre époque, en raison des nombreuses menaces auxquelles elle est exposée. Pour cette raison, des systèmes de détection d'intrusion (IDS) ont été développés pour protéger les données et les utilisateurs. Les administrateurs réseau personnalisent ces systèmes pour empêcher les attaques malveillantes, rendant ainsi ces outils essentiels à la gestion de la sécurité.

Il existe diverses méthodes pour détecter les anomalies et les comportements malveillants. Parmi elles, l'apprentissage automatique se distingue comme une approche clé, utilisant des étapes séquentielles de traitement de l'information pour identifier des motifs et apprendre des caractéristiques ou des représentations.

L'apprentissage automatique joue un rôle crucial dans la détection des intrusions. Il est largement appliqué dans plusieurs domaines, tels que la reconnaissance de la parole, la modélisation de graphes, la surveillance des motifs, la vision par ordinateur, le traitement du langage naturel et le traitement du signal. Les avancées en matière d'algorithmes d'apprentissage offrent un potentiel significatif pour améliorer les capacités des systèmes de détection d'intrusion, augmenter les taux de détection et réduire les fausses alertes. Cependant, il est important de noter que l'implémentation de l'apprentissage automatique dans les opérations de détection d'intrusion présente certaines limitations qu'il convient de prendre en compte.

Dans ce mémoire de master, nous étions orienter pour étudier les techniques d’apprentissage automatique, et leurs utilisation pour la détection d’intrusions aux systèmes informatiques et aux réseaux. Nous avons choisir deux modèles classiques, mais de grande utilité pour la détection

d’intrusion même avec des ensembles de données d’apprentissage restreints. D’autres modèles profonds sont certainement plus performants, mais hélas nécessitent des data-sets volumineuses, et inaccessible au public sur internet. L’objectif de notre étude est de comparer les modèles étudiés, notamment en considérant une base d’apprentissage bien précise, dans notre cas, la base NSL-KDD.

Dans le premier chapitre, nous présenterons les notions de base sur la sécurité informatique, incluant sa définition, ses objectifs, les différentes attaques possibles et les techniques de sécurisation des systèmes informatiques, notamment les systèmes de détection d'intrusions.

Le deuxième chapitre explorera le domaine de l'intelligence artificielle et ses différentes branches, avec un accent particulier sur l'apprentissage automatique des données. Nous examinerons ses concepts fondamentaux, les diverses catégories de classification, ainsi que les multiples techniques et algorithmes utilisés pour découvrir des schémas au sein de vastes ensembles de données, générant ainsi des informations exploitables.

Dans le troisième chapitre, nous détaillerons nos modèles de détection d'intrusion basé sur l'apprentissage automatique, en présentant les algorithmes utilisés (KNN et SVM). Nous décrirons ensuite la conception de notre système, le jeu de données employé et les métriques d'évaluation. Le chapitre suivant sera consacré à notre contribution et aux résultats obtenus grâce aux techniques d'apprentissage automatique décrites précédemment .

Chapitre I : Sécurité informatique et Système de détection d'intrusion

Introduction

La sécurité des systèmes informatique est un problème sensible et préoccupant. Le progrès spectaculaire des technologies de l'information et de la communication offre actuellement des facilités incontournables en matière de transfert de fichiers, messagerie et bien d'autres formes d'échanges d'information. Le développement de l'échange d'information s'est accompagné malheureusement du développement d'activités malveillantes dont les motivations sont aussi nombreuses que dangereuses et évoluent dans le temps. Plusieurs outils et mécanismes de sécurité existent, leur principal rôle est de sécuriser les ressources et les données du système contre tout accès indiscernable et malveillant visant à compromettre la sécurité et la sûreté du système en question. On peut citer entre autres les pare-feux, les anti-virus, les scanners de vulnérabilité, les systèmes de détection d'intrusion, etc. Ce chapitre présente un état de l'art des derniers outils mentionnés : les systèmes de détection d'intrusion. Ce chapitre est subdivisé en deux parties. La partie une est réservée à la sécurité informatique La partie deux arbore une vue générale des IDS, leurs historique, définition et architecture.

I. Sécurité informatique

I.1 Définition :

Avec l'essor d'internet, et l'utilisation par la majorité des entreprises et des organisations de processus informatisés, les menaces visant les systèmes d'informations n'ont cessé d'augmenter et de se sophistiquer, faisant aujourd'hui de la sécurité informatique une nécessité pour tous les types de structure. Donc, **si quoi la sécurité informatique ?**

Définition 1 : La sécurité informatique est l'ensemble des moyens techniques, organisationnels, juridiques et humains pour protéger l'intégrité et la confidentialité des informations stockées dans un système informatique

Définition 2 : La sécurité informatique c'est la protection système informatique contre toute violation, intrusion, dégradation ou vol de données au sein du système d'information.

I.2 Les objectifs et principes fondamentaux de la sécurité informatique :

Les objectifs de protection de la sécurité de l'information sont les point clés élémentaires de la

protection de l'information .l'information représente une valeur économique importante pour toutes les entreprises et pas seulement depuis aujourd'hui.elle est le fondement de leur et constitue donc une condition essentielle à la réussite de leur activités .il est donc évident-ou du moins souhaitable que l'information doit être protégée. Cependant, il existe encore un large fossé entre le désir et le réalité pour y parvenir, les entreprises et les organisations doivent d'abord se pencher sur les objectif de protection fondamentaux de la sécurité de l'information suivant :[1]

- **L'authentification**

L'authentification est l'objectif de la sécurité informatique visant à garantir que seules les entités légitimes ont accès aux ressources et aux services protégés. Elle consiste à vérifier l'identité déclarée d'une entité (utilisateur, système, périphérique) afin de s'assurer qu'elle est autorisée à accéder aux informations ou aux fonctionnalités qu'elle demande. En garantissant l'authenticité des entités qui interagissent avec un système informatique, l'authentification contribue à renforcer la confidentialité, l'intégrité et la disponibilité des données et des services.

- **Intégrité**

La sécurité informatique doit permettre à l'entreprise de garder ses données intactes, et veiller à ce qu'elles ne subissent aucun dommage ou destruction volontaire ou accidentelle.

On distingue généralement deux types d'intégrité, dont les processus et méthode peuvent varier

✓ **L'intégrité physique :** protection de données uniques et exactes lorsqu'elles sont stockées et récupérées.

✓ **L'intégrité logique :** conservation inchangée des données au cours de leurs manipulations multiples au niveau d'une base de données relationnelle

- **La confidentialité**

  La confidentialité consiste à s'assurer que les informations diffusées ne sont accessibles qu'aux personnes qui disposent d'un droit d'accès. Elle constitue un axe important de la sécurité des systèmes d'information

- **La disponibilité (availability)**

La disponibilité est l'une des pierres angulaires de la sécurité informatique. Elle désigne la capacité des collaborateurs à utiliser l'ensemble des données qui leur sont nécessaires pour l'accomplissement d'une tâche de manière sécurisée.

- **La non-répudiation**

La non-répudiation des informations a pour but d'assurer que l'émetteur d'une information (quelle qu'elle soit) ne soit pas en mesure de nier qu'il est bien à l'origine de celle-ci. Pour répondre à cet objectif de la sécurité informatique et qu'il se concrétise, on utilise la signature des emails, des documents ou des certificats. Ainsi, seul l'utilisateur possédant une clé privée peut apposer sa signature sur un email. Cette personne ne peut donc pas nier en être l'émetteur.

II. Les Attaques informatiques

Les attaques informatiques, définies comme des tentatives de compromettre des systèmes informatiques, sont menées par des individus ou des groupes pour des raisons variées telles que le gain financier, des convictions politiques, ou la curiosité. La dépendance croissante aux réseaux informatiques a révélé des vulnérabilités, c'est-à-dire des failles dans les systèmes dues à des budgets limités, un manque de temps, un personnel insuffisamment qualifié, ou des politiques de sécurité inadéquates. Les coûts de réparation après une attaque peuvent être élevés, incitant les entreprises à investir dans la sécurité informatique.

Plutôt que de viser une sécurité parfaite et irréaliste, il est crucial de prévenir, détecter et réagir aux attaques. La prévention utilise des techniques comme l'authentification, le chiffrement et le camouflage pour dissuader les attaquants. Étant donné que la prévention ne peut pas tout arrêter, la détection sert à identifier les activités suspectes qui violent les politiques de sécurité[2].

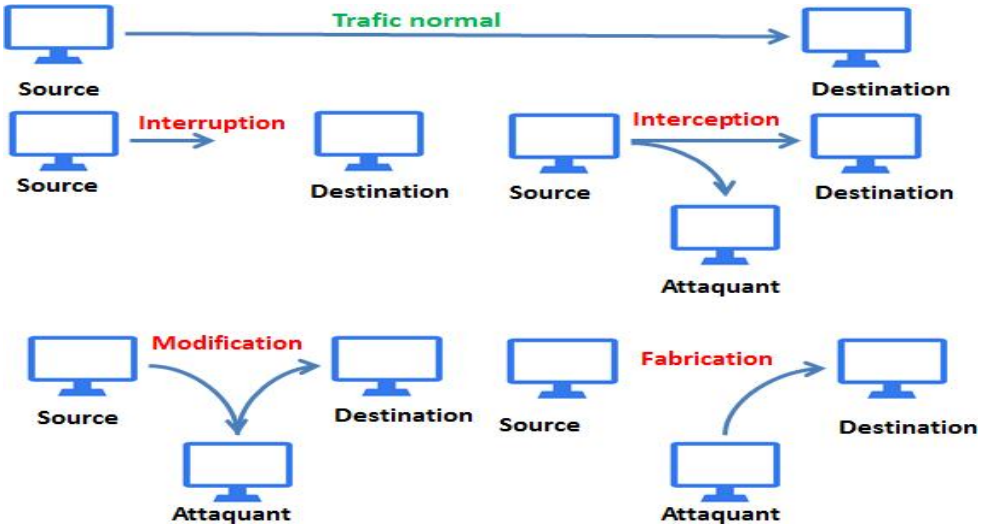

Figure1. 1: Les objectifs des attaques informatiques**.** [2]

II.1 Schéma d'une attaque :

Un schéma d'attaque illustre les différentes étapes impliquées dans une tentative de violation des objectifs de la sécurité informatique. Voici une approche en six points :[3]

- **Collecte d'informations :** L'attaquant commence par recueillir des informations sur le système ciblé, telles que les vulnérabilités, les configurations, les points d'accès, etc.
- **Intrusion :** Une fois que suffisamment d'informations ont été collectées, l'attaquant utilise ces données pour infiltrer le système, exploitant les vulnérabilités découvertes ou en utilisant des techniques telles que l'ingénierie sociale pour accéder aux ressources protégées.
- **Établissement d'un accès persistant :** Après avoir réussi à s'introduire dans le système, l'attaquant peut mettre en place des mécanismes permettant une ré-intrusion future, tels que l'installation de logiciels malveillants persistants ou la création de portes dérobées.
- **Propagation :** L'attaquant peut chercher à propager son intrusion à d'autres systèmes connectés, créant ainsi un réseau d'attaques distribuées qui augmentent sa portée et sa capacité à causer des dommages.
- **Paralysie du système :** Une fois qu'il a atteint ses objectifs, l'attaquant peut déployer des actions visant à paralyser le système cible, entraînant des dysfonctionnements, des temps d'arrêt ou des pertes de données.
- **Effacement des traces :** Pour éviter d'être détecté et poursuivi, l'attaquant efface méticuleusement toutes les traces de son intrusion, telles que les journaux d'activité, les fichiers de configuration modifiés, etc., afin de rendre son intrusion aussi discrète que possible. [4]

II.2 Les types d'attaque :

Il existe plusieurs formes d'attaque et Parmi les plus connues on a :

- **Attaque réseaux**

IP spoofing : Technique utilisée afin de tenter d'obtenir un accès non autorisé sur une machine. L'intrus envoi des messages à un ordinateur cible en utilisant une Adresse IP semblant indiquer que le message provient d'une machine de confiance. [5]

Les scans : c'est la première et la plus importante étape de l'attaquant est d'obtenir suffisamment d'informations pour préparer une attaque plus efficace.Cette technique consiste à rapporter des informations sur les machines scannées, et en particulier le système d'exploitation et les services

installés. On peut donc déterminer avec précision les failles de sécurités et donc les types d'attaques possibles sur la machine en question.

Sniffing: Il permet de surveiller et d'analyser le trafic réseau afin d'obtenir des informations pertinentes pour que l'attaquant peut avoir une bonne modélisation des attaques ultérieures.

Déni de service : visent à inonder les systèmes, les réseaux ou les serveurs d'un trafic massif, rendant ainsi le système incapable de répondre aux demandes légitimes. Les attaques peuvent également utiliser plusieurs dispositifs infectés pour lancer une attaque sur le système cible. C'est ce qu'on appelle une attaque par déni de service distribué (DDoS).[6]

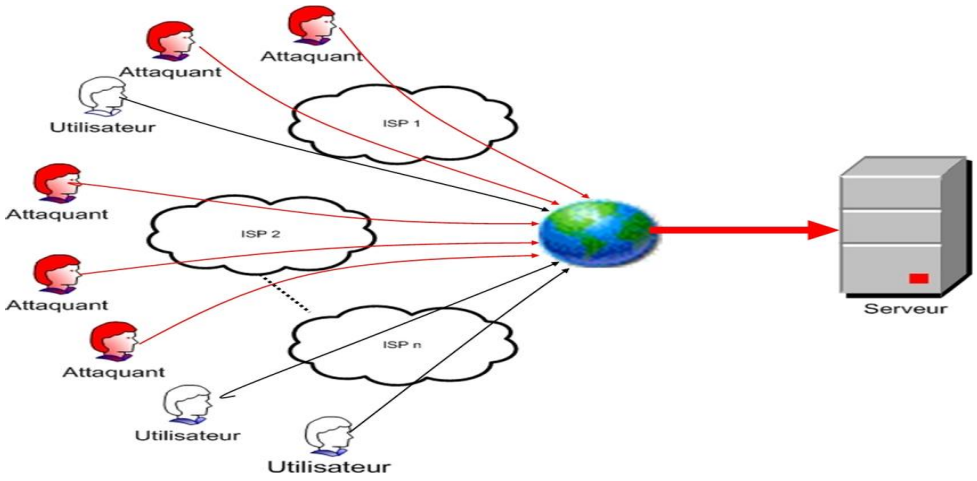

Figure1. 2: Exemple d'attaque déni de service. [6]

II.3 Attaque applicative

- **Les injections SQL :** Tout comme les attaques de scripts, les injections SQL profitent de paramètres d'entrée non vérifiés. Comme leur nom l'indique, 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. [7]

- **Man in the middle :**(MITM) est un terme général désignant le moment où un auteur se positionne dans une conversation entre un utilisateur et une application, soit pour écouter

clandestinement, soit pour se faire passer pour l'une des parties, ce qui donne l'impression qu'il s'agit d'un échange normal d'informations. Est en cours.[8]

De plus, il peut être utilisé pour prendre pied à l'intérieur d'un périmètre sécurisé pendant la phase d'infiltration d'une attaque contre une menace persistante avancée (MPA).

D'une manière générale, une attaque MITM équivaut à un facteur ouvrant votre relevé bancaire, notant les détails de votre compte, puis refermant l'enveloppe et la livrant à votre porte. [9]

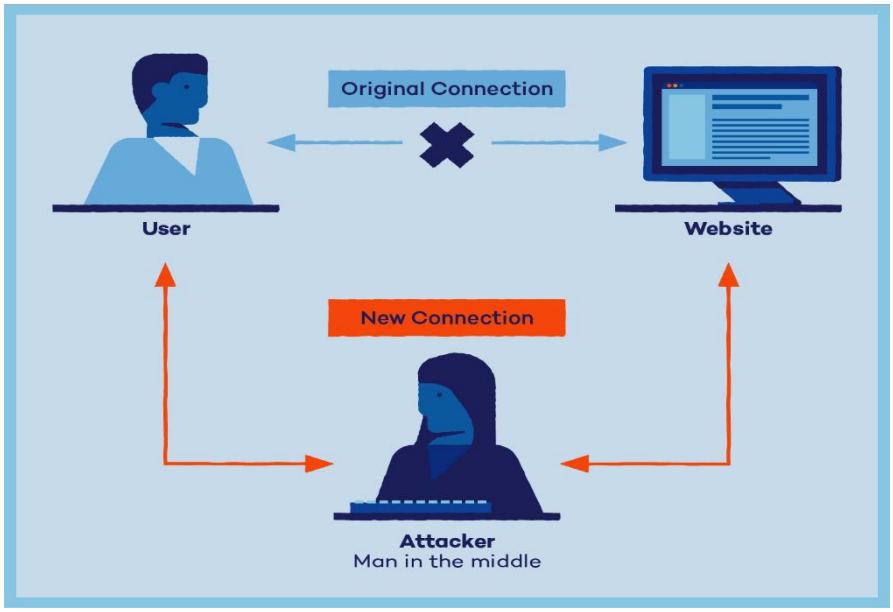

Figure1. 3: Man in the middle attack [7]

II.4 Les attaques données

- **logiciel malveillant :**Un logiciel malveillant ou maliciel, aussi dénommé logiciel nuisible ou programme malveillant ou pourriciel (de l'anglais malware), est un programme développé dans le but de nuire à un système informatique, sans le consentement de l'utilisateur dont l'ordinateur est infecté. Il existe plusieurs méthodes utilisées par les pirates pour infecter un ordinateur, comme le phising (hameçonnage par e-mail) ou le téléchargement automatique d'un fichier par exemple.

De nos jours, le terme « virus » est souvent employé, à tort, pour désigner toutes sortes de logiciels malveillants. En effet, les maliciels englobent les virus, les vers, les chevaux de Troie, ainsi que

d'autres menaces. La catégorie des virus informatiques, qui a longtemps été la plus répandue**.[10]**

- **Ver :** C'est un malware qui circule d'une façon autonome entre les systèmes en utilisant un des protocoles de communication, typiquement la messagerie électronique. Il fait de sorte de se coller comme pièce jointe à un message séduisant bien préparé, puis il s'envoie automatiquement à tous les contacts de l'utilisateur, à son insu. Tout destinataire qui ouvre la pièce jointe provoque le même processus, et le ver peut se propager ainsi chez des millions d'utilisateurs en quelques jours. Il peut aussi être téléchargé à partir d'une page web piégée**.[11]**
- **Virus :** Un virus est un code malveillant qui se propage à l'aide d'un autre programme ou fichier qui joue le rôle d'hôte du virus et sans lequel le virus ne peut pas se propager.

En général, pour infecter un système, un virus il se présente sous la forme de quelques lignes de code en langage machine binaire qui se greffent sur un programme utilisé sur le système cible, afin d'en modifier le comportement.

- **Cheval de Troie :** Une fonctionnalité malveillante ou un code malicieux peuvent être intégrés à une application d'apparence non nuisible. Quand l'application est utilisée le code malveillant est activé pour exécuter les actions programmées par le hacker et qui causent les préjudices au système cible.
- **Bombe logique:** C'est un code dormant dans le système et qui ne se déclenche Qu'à une date et un temps bien précis, et dont l'action peut être nuisible pour le Système lui-même, ou pour d'autres systèmes. Dans ce cas, le système affecté Sera utilisé comme machine d'attaque. Ceci est réalisé dans les attaques par déni de service : Dos et DDoS
- **Porte dérobée :** Une porte dérobée, également appelée Remote Administration Tool (RAT), est une application qui permet à certains utilisateurs (administrateurs systèmes ou cybercriminels) d'accéder au système d'un ordinateur sans que l'utilisateur ne l'autorise ou ne le sache.
- **Logiciel espion (spyware) :** aussi appelé mouchard, est un logiciel qui peut secrètement enregistrer vos activités sur votre ordinateur. Le plus souvent, il s'installe par le biais d'un téléchargement gratuit ou d'une pièce jointe d'e-mail infectée ; Un simple clic suffit pour installer un logiciel espion sur votre ordinateur, sans que vous le sachiez. Une fois installé, le logiciel espion Il est donc difficile à détecter et encore plus à désinstaller.

- **Pourriel (spam) :** Un spam (pourriel, junk mail, courrier indésirable) est l'envoi massif de courrier électronique à des destinataires ne l'ayant pas sollicité. Les spammeurs collectent généralement les adresses électroniques sur internet (dans les forums, sur les sites internet, dans les groupes de discussion, …). . en général d 'envois en grande quantité effectués à des fins publicitaires

III. Le mécanisme de défense et de sécurité

III.1 Un antivirus :

Est un logiciel dont l'objectif est de détecter des menaces de types virus (mais également vers informatiques, chevaux de Troie etc) et de les stopper ou les mettre en quarantaine de façon à protéger la machine ou le système informatique concerné

Il existe différents types de logiciels antivirus, du logiciel antivirus grand public, gratuit ou payant, au logiciel antivirus professionnel, davantage destiné à l'usage des entreprises.

III.2 Pare feu :

Un firewall (ou pare-feu) est outil informatique (matériel et/ou logiciel) est une passerelle filtrante qui protège un ordinateur ou un réseau des intrusions venues d'Internet. Il est parfois traduit comme coupe-feu, barrière de sécurité ou garde-barrière. Il est doté au moins de deux interfaces, l'un destiné au réseau interne et l'autre au réseau externe.[12]

Voici une figure qui montrer tout ça :

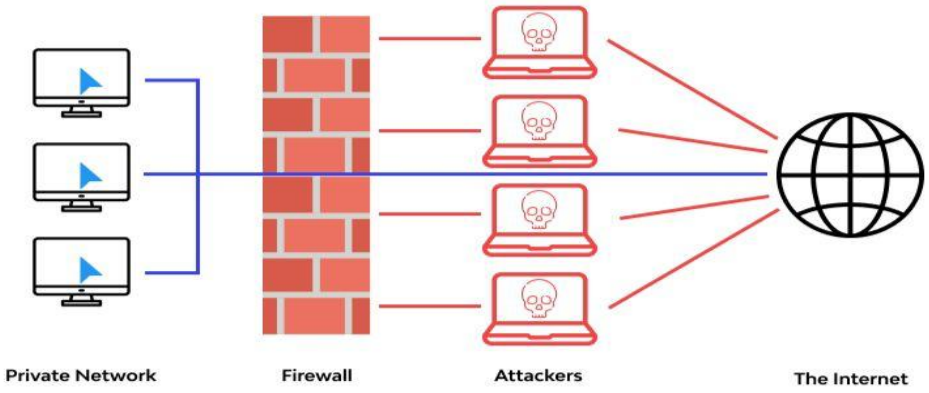

Figure1. 4:**Pare-feu(Firewall).** [12]

III.3 Cryptographie:

La cryptographie est une méthode de protection des informations et des communications par l'utilisation de codes, de sorte que seuls les destinataires des informations puissent les lire et les traiter. En informatique, la cryptographie désigne des techniques d'information et de communication sécurisées dérivées de concepts mathématiques et d'un ensemble de calculs basés sur des règles, appelés algorithmes, pour transformer les messages de manière difficile à déchiffrer. Ces algorithmes déterministes sont utilisés pour la génération de clés cryptographiques, la signature numérique, la vérification pour protéger la confidentialité des données.**[13]**

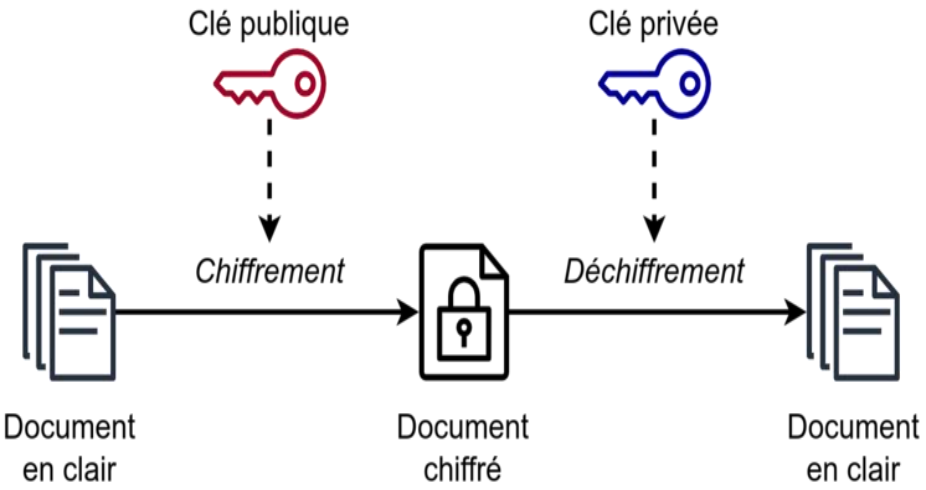

Figure1. 5: **la cryptographie** [14]

III.4 VPN (Virtual Privat Network) :

Dans les réseaux informatiques, le réseau privé virtuel (Virtual Privat Network, abrégé en VPN) est une technique permettant aux postes distants de communiquer de manière sûre, tout en empruntant des infrastructures publiques (internet).

Un VPN repose sur un protocole, appelé protocole de tunnelisation, c'est-à-dire un protocole permettant aux données passant d'une extrémité à l'autre du VPN d'être sécurisées par des algorithmes de cryptographie.

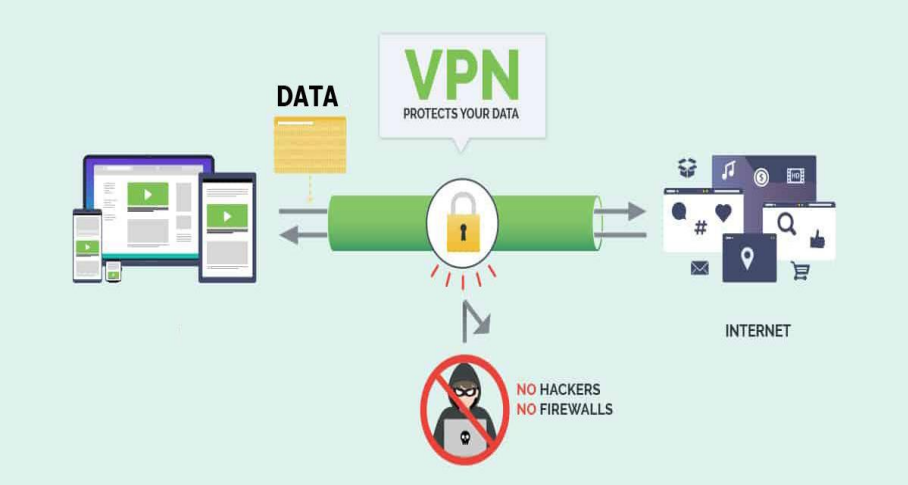

Figure1. 6 : Réseau privé virtuel (VPN) [15]

Système de détection d'intrusion (IDS) : La détection des intrusions est un mécanisme de cyber sécurité courant dont la tâche est de détecter les activités malveillantes dans des environnements hôte et ou réseau. La détection des activités malveillantes permettent de réagir en temps opportun, par exemple pour arrêter une attaque en cours. Vu l'importance de détection des intrusions, les milieux de la recherche et de l'industrie ont conçu et développé une variété de systèmes de détection d'intrusion.**[16]**

IV. Système de Détection D'intrusion(IDS)

IV.1 Introduction :

Avec le développement rapide de la technologie des réseaux et en particulier les réseaux sans fil, la sécurité de ces réseaux ainsi que ses terminaux connectés contre diverses menaces intentionnelles ou

accidentelles, est devenue un problème crucial.

Toutes les informations concernées par les technologies Internet, les informations stockées dans des bases de données et qui sont transmises sur le réseau doivent être protégées. Les intrusions sont des véritables menaces qui peuvent être des activités non autorisées ou des utilisations malveillantes des ressources d'information qui offensent les politiques de sécurité.

Les systèmes et les techniques traditionnelles de prévention des intrusions comme les pare-feu, le cryptage et le contrôle d'accès sont la plupart du temps inefficaces face à ces nouvelles menaces. C'est pour pallier ce manque que sont apparus récemment des nouveaux composants de sécurité appelés les systèmes de détection d'intrusions.Ces derniers permettent de déjouer les attaques attendues sur le réseau en générant des alertes, des avertissements là ou il existe des menaces externes ou internes, ce qui aide à réduire le temps et l'effort fournit par l'administrateur et permet d'avoir la confiance des clients et garder une bonne réputation sur l'organisation.

IV.2 Définition d'un (IDS) :

La détection d'intrusion consiste à surveiller les événements survenant dans un ordinateur ou un réseau et à les analyser pour repérer des signes d'intrusions **.**Ces intrusions sont définies comme des tentatives de compromission de la confidentialité, de l'intégrité ou de la disponibilité des systèmes, ou de dépassement des conditions de sécurité d'un ordinateur ou d'un réseau. Un système de détection d'intrusion (IDS) est un dispositif ou une application qui alerte l'administrateur en cas de faille de sécurité, de violation de règles ou d'autres problèmes pouvant compromettre le réseau informatique.**[17]**

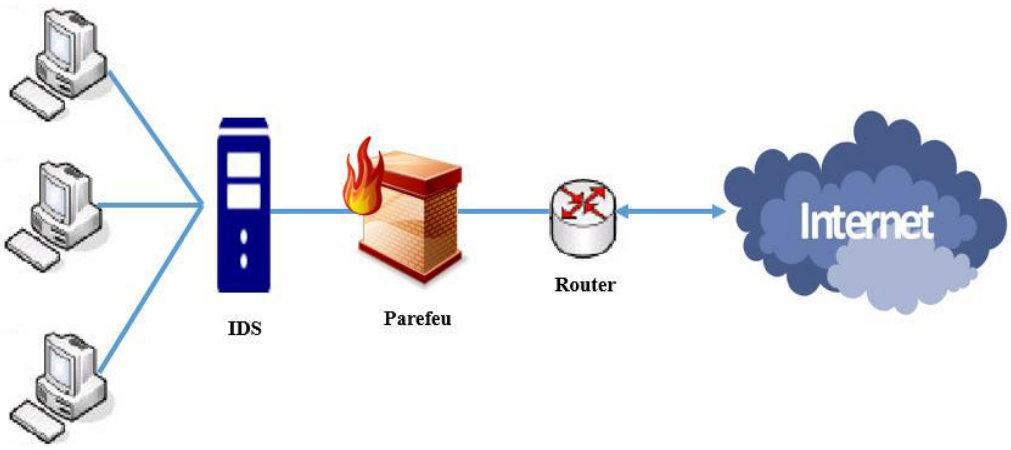

Figure 1.7 : Système de détection d'intrusions **[18]**

IV.3 Objectifs des (IDSs) :

Un système de détection d'intrusion (IDS) analyse les événements provenant de différents systèmes en temps réel ou en différé pour détecter et prévenir les attaques. Ses objectifs incluent : **[19]**

- ✓ Collecter des informations sur les intrusions.
- ✓ Centraliser la gestion des alertes.
- ✓ Effectuer un premier diagnostic sur la nature de l'attaque.
- ✓ Permettre une réponse rapide et efficace.
- ✓ Réagir activement à l'attaque pour la ralentir ou la stopper.

IV.4 Architecture fonctionnelle des(IDSs):

Nous décrivons dans ce qui suit les composants constituants un système de détection d'intrusions.

- **Capteur [20]**

Le détecteur surveille l'activité du système en utilisant une source de données et transmet à l'analyseur une série d'événements qui fournissent des informations sur l'évolution de l'état du système. Bien que le détecteur puisse simplement transmettre ces données brutes, il est généralement procédé à un prétraitement. Traditionnellement, trois types de détecteurs sont distingués en fonction des sources de données utilisées pour observer l'activité du système : les détecteurs système, les détecteurs réseau et les détecteurs applicatifs.

- **Analyseur :**

L'objectif de l'analyseur est de déterminer si le flux d'événements fourni par le capteur contient des éléments caractéristiques d'une activité malveillante.

- **Manager (gestionnaire) :**

Le gestionnaire recueille les alertes générées par le capteur, les organise et les présente à l'opérateur. Il peut également être responsable de déterminer la réponse à adopter, laquelle peut inclure :

- ✓ Isolation de l'attaque, visant à limiter ses effets.
- ✓ Élimination de l'attaque, visant à arrêter l'attaque.
- ✓ Rétablissement, consistant à restaurer le système dans un état sain.
- ✓ Diagnostic, qui est la phase d'identification du problème.

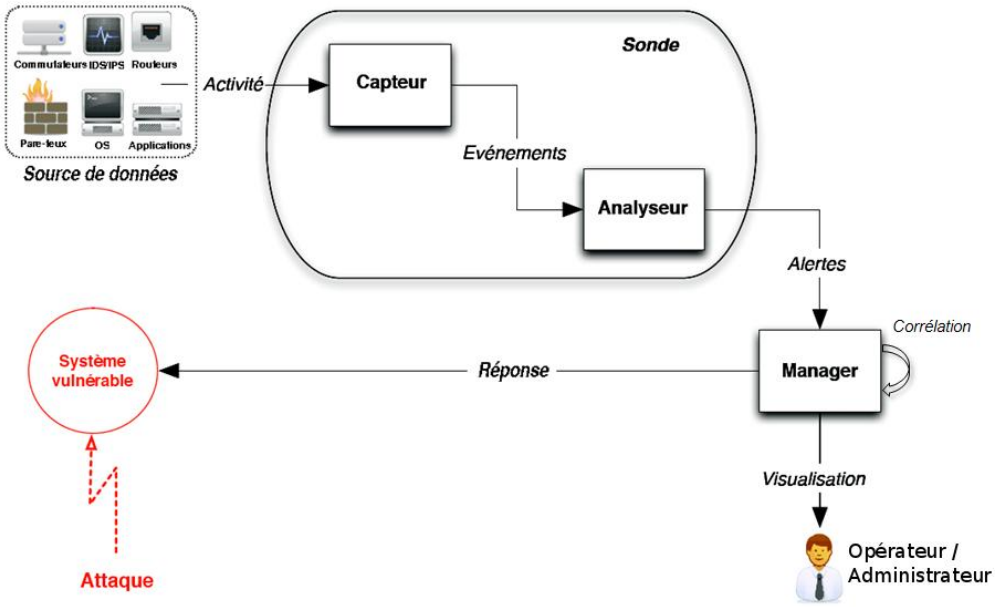

Figure1. 7: **Architecture de base d'un IDS** [8]

IV.5 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 la 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 la figure 1.8)

- L'emplacement d'IDS
- Les méthodes de détection
- Les types de réponse
- Fréquence d'utilisation

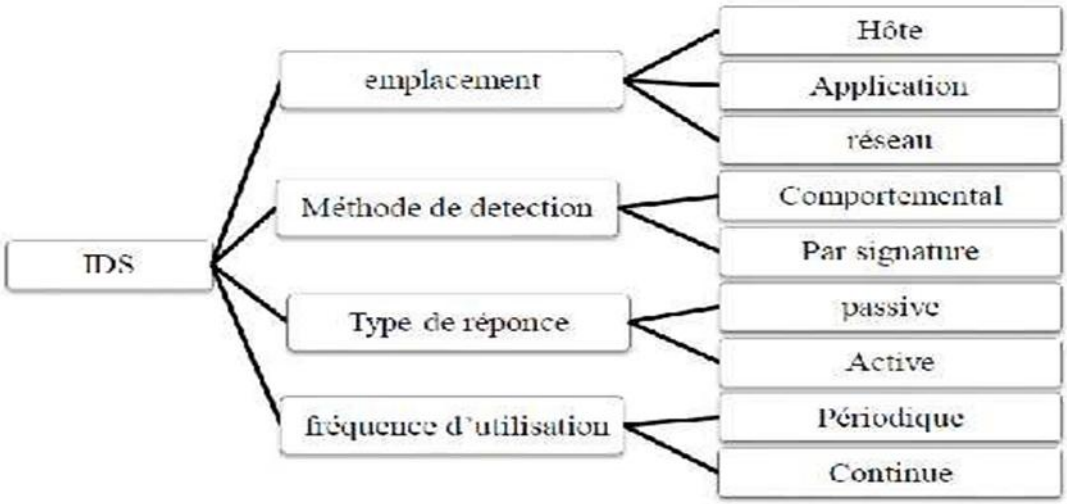

Figure1. 8: **Classification des IDS.**[21]

IV.5-1 Emplacement d'un IDS :

Les IDS se classent en trois catégories principales :

Les Systèmes de détection d'intrusions de type "hôte" (HIDS) : Ces systèmes surveillent un ordinateur spécifique sur lequel ils sont installés. Ils fonctionnent comme des agents qui analysent et détectent toute activité interne ou externe contournant la politique de sécurité du système.**[19]**

Les Systèmes de détection d'intrusions "réseaux" (NIDS) : Ces outils analysent le trafic réseau à l'aide d'une sonde qui "écoute" le segment de réseau à surveiller, détectant les anomalies par rapport à un modèle de référence.

Les Systèmes de détection d'intrusions "hybrides" (NIDS+HIDS) : Ces systèmes combinent les caractéristiques des NIDS et des HIDS. Ils surveillent à la fois le réseau et les terminaux. Les sondes, placées en des points stratégiques, agissent comme des NIDS et/ou des HIDS selon leur emplacement. Toutes les alertes sont centralisées et les informations de diverses origines sont agrégées et corrélées par une machine dédiée.

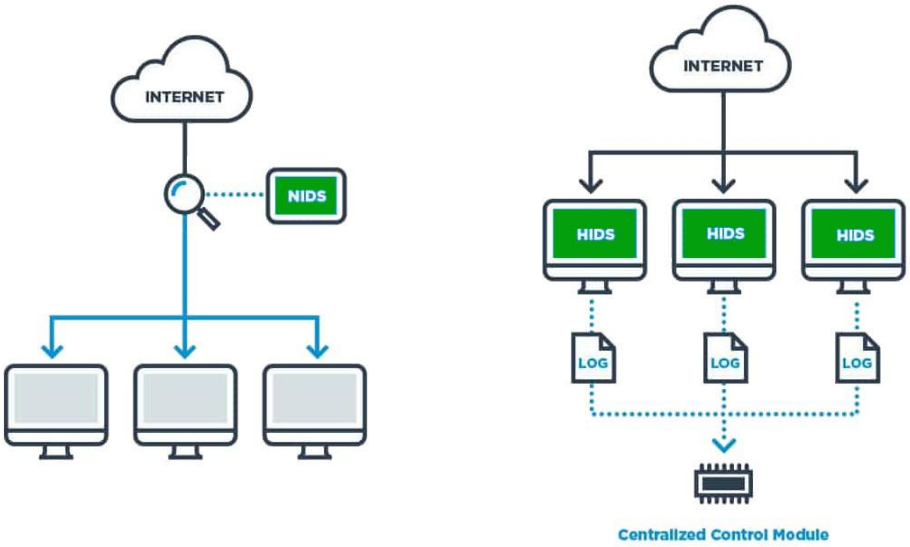

Figure1. 9 : : Emplacement d'un IDS.[19]

IV.5-2 Les Méthodes de détection :

Deux approches de détection ont été proposées :

L'approche comportementale modélise le comportement normal des utilisateurs, du système informatique et de l'activité réseau. Ensuite, toute déviation par rapport à la normale constitue un événement suspect. Tandis que l'approche par signature, elle, s'appuie sur un modèle constitué des sections interdites dans le système d'informatique, ce modèle s'appuie sur la connaissance des techniques employées par les attaquants : on tire des scénarios d'attaque et on recherche dans les traces d'audit leur éventuelle survenue.

- **L'approche par scénario ou signature** :

Dans cette approche, les détecteurs d'intrusion utilisent une base de motifs représentant des scénarios d'attaque connus. Cette base est utilisée en temps réel pour analyser les informations fournies par les sondes de détection. Il s'agit d'un système de reconnaissance de motifs qui identifie la présence d'intrusions connues grâce à une base de signatures. L'efficacité de ce système dépend donc fortement de la précision de cette base de signatures.

Il est possible de créer des signatures plus génériques pour détecter les variantes d'une même attaque, ce qui nécessite une bonne connaissance des attaques et du réseau pour éviter de perturber le trafic normal. Une signature définit les caractéristiques d'une attaque, soit au niveau des paquets (jusqu'à

TCP ou UDP), soit au niveau des protocoles (HTTP, FTP, etc.).

Au niveau des paquets, l'IDS analyse les différents paramètres de tous les paquets en transit et les compare aux signatures d'attaques connues. Au niveau des protocoles, l'IDS vérifie si les commandes envoyées sont correctes et ne contiennent pas d'attaque, une fonctionnalité particulièrement développée pour HTTP. Cependant, plus il y a de signatures à tester, plus le temps de traitement est long. L'utilisation de signatures plus élaborées peut donc offrir un gain de temps appréciable. **[22]**

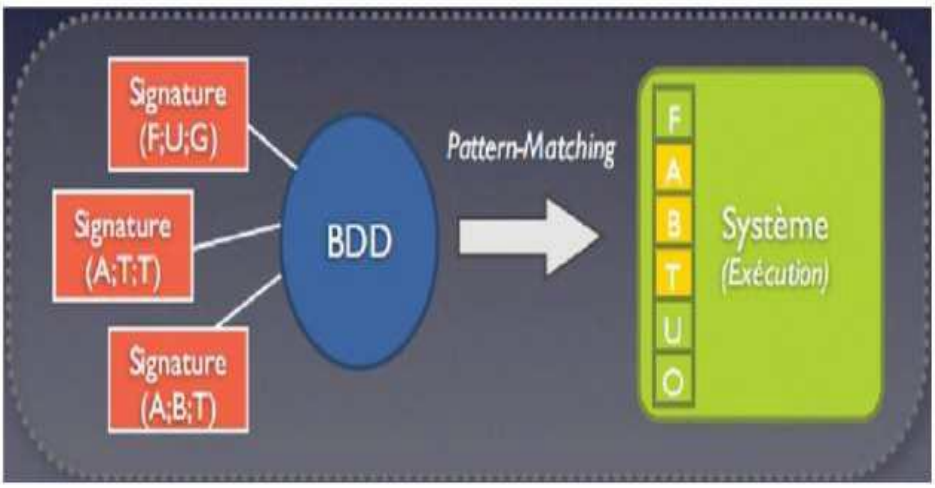

Figure1. 10: **Approche par scénario** [22]

- **L'approche comportementale :**

Les détecteurs d'intrusions comportementaux fonctionnent en créant un modèle de référence qui représente le comportement normal de l'entité surveillée. Ce modèle sert ensuite lors de la phase de détection pour identifier les déviations comportementales. Le comportement actuel de l'entité est comparé à ce modèle de référence, et une alerte est déclenchée lorsqu'une déviation significative (au-delà d'un certain seuil) est détectée. Cette approche considère que tout comportement déviant du modèle de comportement normal est une anomalie, potentiellement symptomatique d'une intrusion ou d'une tentative d'intrusion.

IV.5-3 Types de réponses :

Une autre façon de classer les systèmes de détection d'intrusions consiste à les classer par type de réaction lorsqu'une attaque est détectée **: [22]**

- **Réponse passive :**

Un IDS passif est un système conçu uniquement pour surveiller et analyser le trafic réseau, puis alerter un opérateur en cas de vulnérabilités ou d'attaques potentielles. Il ne peut pas, à lui seul, exécuter des fonctions de protection ou de correction.

Lorsqu'une attaque est détectée, l'IDS passif génère une alerte et notifie l'administrateur système par e-mail, message dans une console, ou même par bippeur. C'est ensuite à l'administrateur de prendre les mesures nécessaires. Les principaux avantages des systèmes IDS passifs sont leur facilité et rapidité de déploiement.

- **Réponse Active :**

La réponse active à une attaque vise à l'interrompre dès sa détection. Deux techniques principales sont utilisées pour cela : la reconfiguration du pare-feu et l'interruption d'une connexion TCP.

- **Reconfiguration du pare-feu :**

Cette technique bloque le trafic malveillant en fermant le port utilisé ou en interdisant l'adresse de l'attaquant au niveau du pare-feu. La capacité de reconfiguration dépend du modèle de pare-feu, car tous ne permettent pas cette fonctionnalité par un IDS. La reconfiguration est limitée par les capacités du pare-feu.

- **Interruption de la connexion TCP :**

L'IDS peut interrompre une session entre un attaquant et la cible pour empêcher le transfert de données ou la modification du système attaqué. Pour cela, l'IDS envoie un paquet TCP reset (avec le flag RST) aux deux extrémités de la connexion, faisant croire à chaque extrémité que l'autre s'est déconnectée, ce qui interrompt l'attaque.

Dans le cadre d'une réponse active, il est crucial de s'assurer que le trafic détecté comme malveillant l'est réellement, afin d'éviter de déconnecter des utilisateurs légitimes. Généralement, les IDS ne réagissent activement qu'aux alertes confirmées comme étant des attaques. L'analyse des fichiers

d'alertes générés est donc nécessaire pour évaluer toutes les attaques détectées.**[23]**

Cependant, l'automatisation de ces réponses peut être dangereuse, car elle peut entraîner des dénis de service provoqués par l'IDS lui-même. Un attaquant peut tromper l'IDS en usurpant des adresses du réseau local, que l'IDS interprétera alors comme la source de l'attaque. Il est donc préférable de proposer une réaction facultative à un opérateur humain, qui prendra la décision finale.

Enfin, l'utilisation d'un IDS peut se faire de manière continue (en ligne) ou périodique (hors ligne).

IV.5-4 Fréquence d'utilisation :

Il existe deux façons de mesurer, d'évaluer et d'estimer la fréquence d'utilisations des systèmes de détection d'intrusions, la surveillance continue et l'analyse périodique. [24]

- **La surveillance périodique :**

est la surveillance régulière et l'analyse systématique du système de manière périodique, afin de détecter toute intrusion potentielle ou anomalie survenue dans le passé. Cela se traduit par des examens récurrents à des intervalles déterminés par l'administrateur, représentant une (observation périodique dans le temps).

- **La surveillance continue :**

Revient à de surveiller et d'analyser le système de manière permanente, sans arrêt et en temps réel (observation dans le temps du système).

Conclusion

En conclusion, les IDS représentent un pilier crucial dans la défense informatique, offrant une protection indispensable contre les attaques potentielles, étant des outils de première importance pour détecter les menaces sans nécessiter de connaissances préalables. Cependant, leur efficacité dépend de la sélection appropriée des meilleurs classifieurs en fonction des données d'apprentissage disponibles. Dans le prochain chapitre, nous examinerons de près les méthodes de classification liées à l'apprentissage automatique, dans le but d'améliorer la précision des analyseurs d'IDS et de renforcer ainsi la sécurité des systèmes informatiques.

Chapitre II : Apprentissage Automatique

Introduction

Les avancées significatives dans le domaine de la cybersécurité ont engendré un intérêt croissant pour les stratégies de détection des attaques, notamment grâce à l'utilisation de techniques d'apprentissage automatique (ML). Ces techniques révolutionnaires ont pour objectif premier d'analyser les vastes ensembles de données collectées afin d'identifier des schémas et des comportements suspects, pouvant indiquer la présence potentielle d'activités malveillantes au sein des systèmes informatiques cibles ou des réseaux.

Ce chapitre se penche sur l'approche et les méthodes de l'intelligence artificielle, en particulier l'apprentissage automatique, employées dans le domaine de la détection d'intrusion, en mettant l'accent sur leur pertinence et leur efficacité.

I. L'intelligence artificielle

L'intelligence artificielle (IA) représente un champ interdisciplinaire de l'informatique et des mathématiques, englobant un ensemble de techniques algorithmiques et de théories visant à créer des machines capables de simuler l'intelligence humaine. Son objectif principal est de reproduire les capacités cognitives humaines afin de résoudre des problèmes complexes. Pour ce faire, l'IA cherche à modéliser l'intelligence humaine en tant que phénomène, tout comme cela se fait dans d'autres disciplines telles que la physique, la chimie ou la biologie. Ce domaine en pleine expansion trouve ses fondements théoriques et ses applications dans divers domaines tels que la théorie des probabilités, les neurosciences, la robotique, la théorie des jeux, la santé et les transports.

L'intelligence artificielle est généralement catégorisée en deux types[25] :

- L'IA forte vise à développer des machines dotées de capacités cognitives proches de celles de l'humain, leur permettant de résoudre des problèmes inconnus de manière autonome. Bien que cela reste un domaine de recherche actif, l'IA forte n'existe pas encore.
- L'IA faible se concentre sur le développement de machines capables de résoudre des problèmes spécifiques en utilisant des données et une assistance humaine, sans toutefois avoir une compréhension profonde des phénomènes en jeu.

I.1 Définition de l'intelligence artificielle (IA):

Figure 2. 1:Intelligence Artificiel AI *[Online, 26]*

- (**John McCarthy**): Le but de intelligence artificielle est l'étude de la structure de l'information et de la structure de processus de résolution de problèmes, indépendamment des applications et indépendamment d'une réalisation.
- **(Marvin Minsky**): L'IA a pour but la construction de programmes informatiques qui s'adonnent a des taches qui sont, pour l'instant, accomplies de façon plus satisfaisantes par des êtres humains car elles demandent des processus mentaux de haut niveau tels que : l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critique
    - (**Allen Newell**): Une présupposition essentielle pour agir intelligemment d'une manière générale est la capacité de produire et de manipuler des structures symboliques.
    - (**John Shore**) : Reproduction des activités cognitives jugées intelligentes pour l'homme.
    - (**Jean-Louis Laurière**) : Etude des activités intellectuelles de l'homme pour lesquelles aucune méthode n'est a priori connue. (Tout ce qui n'a pas encore été fait en Informatique - quand on sait le faire, ce n'est plus de l'IA ...) [27]

I.2 Définition d'un système intelligent:

Un système intelligent est un système doté d'une intelligence artificielle, composé d'une partie matérielle et d'une partie informatique. Ces systèmes sont capables de traiter différentes informations et de prendre des décisions en conséquence.

I.3. Domaines de l'Intelligence Artificiel:

Le domaine de l'intelligence artificielle est subdivisé en plusieurs sous-domaines interconnectés, tels que représentés dans la Figure 2.2

L'apprentissage automatique, par exemple, consiste à acquérir des connaissances à partir de l'expérience ou de bases de données afin de résoudre des problèmes donnés. Il repose particulièrement sur l'analyse statistique des données d'entraînement et comprend une variété d'algorithmes utilisant différents modèles mathématiques, parmi lesquels les réseaux de neurones, largement répandus dans divers domaines. L'apprentissage profond, quant à lui, exploite les réseaux de neurones pour résoudre des problèmes complexes, notamment dans le traitement d'images et de séries temporelles comme la reconnaissance vocale.

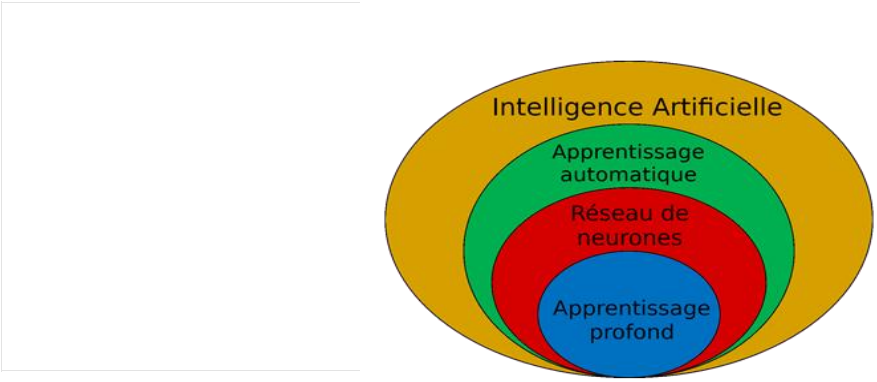

Figure 2. 2: Domaines de l'Intélligence Artificiel[Online, 28]

Il est à noter que toutes les techniques d'intelligence artificielle ne relèvent pas nécessairement de l'apprentissage automatique. Par exemple, il est possible de programmer directement des règles de prédiction dans une machine sous forme de conditions "si-alors". Ces systèmes, appelés systèmes experts, ont été largement utilisés dans les années 1980 pour accomplir des tâches spécifiques avec une grande précision. Cependant, ce chapitre se concentrera uniquement sur les techniques d'apprentissage automatique.

II. Apprentissage automatique

L'apprentissage automatique englobe un large éventail de techniques et sert à des fins variées. Avant d'entrer dans les détails, il est nécessaire de clarifier ce qu'est le AA.

II.1 Définition d'Apprentissage Automatique :

L'apprentissage, ou Learning en anglais, c'est le processus de construire un modèle général à partir de données (observations) particulières du monde réel. Ainsi, le but est double :

- Prédire un comportement face à une nouvelle donnée.
- Approximer une fonction ou une densité de probabilité.

Deux (02) branches d'apprentissage existent :

- Apprentissage symbolique, issue de l'IA.
- Apprentissage numérique, issue des statistiques.

Dans la pratique, le mot entraînement (Training, en anglais) est souvent synonyme de : apprentissage. Ainsi, en science cognitive, l'apprentissage est définit comme étant la capacité à améliorer les performances au fur et à mesure de l'exercice d'une activité.

Exemple, un joueur de jeu d'échec : assimile (par expérience, s'entraîne) et raisonne (ceci lui procure une certaine intelligence ou puissance de raisonnement pour qu'il puisse progresser) –c'est le cas pour un algorithme intelligent [29].

(Arthur Samuel) a proposé une première définition en 1959 : "Champ d'études donnant aux ordinateurs la capacité d'apprendre sans être explicitement programmé." Cette définition, en apparence simple, reste complexe à saisir en dehors de son contexte initial, qui était principalement axé sur le jeu d'échecs.

Une autre définition, plus technique, a été avancée par (**Tom Mitchell)** en 1997 : "On dit d'un programme informatique qu'il apprend de l'expérience E concernant une certaine classe de tâches T et d'une mesure de performance P, si sa performance aux tâches dans T, telle que mesurée par P, s'améliore avec l'expérience E."[30]

Cette définition transpose cette idée à la détection d'intrusions. Elle énonce qu'un système de détection d'intrusions réseau utilisant une technique d'apprentissage automatique est un

programme informatique qui apprend à partir des données enregistrées sur un réseau. Il réalise la tâche de classification en identifiant le trafic normal ou les attaques, et sa performance de classification s'améliore avec l'expérience.

Concernant les méthodes d'apprentissage et les types de tâches, il est possible de les catégoriser selon divers critères, tels que le niveau de supervision apporté à la création du modèle ou l'interaction entre le modèle et son environnement.

La Figure 2.3 donne une représentation partielle de ces différentes méthodes, en se basant sur le type d'apprentissage et les types de tâches associés. Son contenu est suffisant pour comprendre nos travaux de recherche.

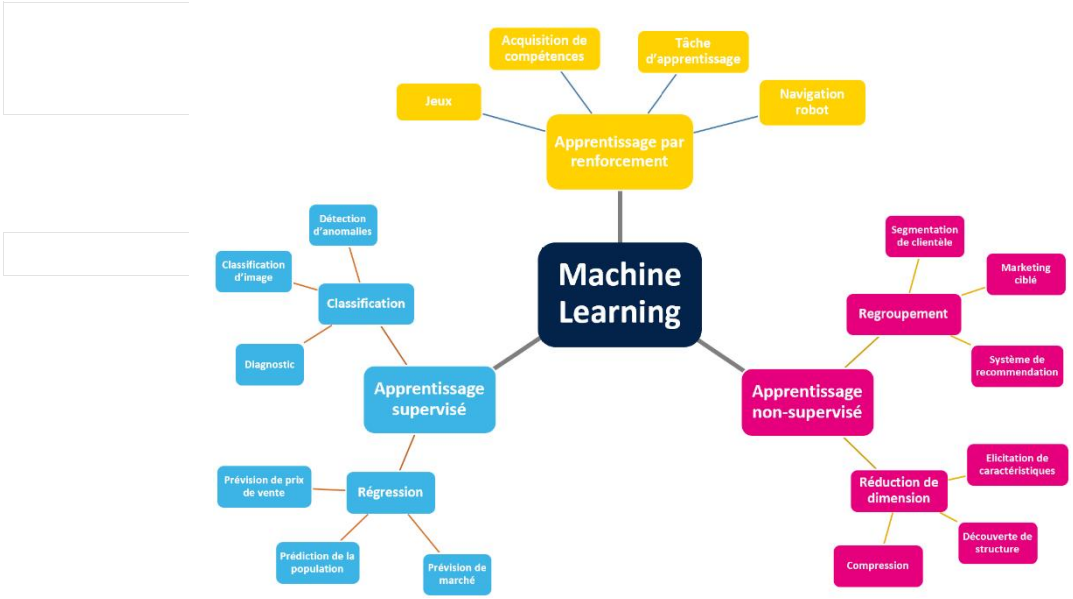

Figure 2. 3: Catégories d'apprentissage automatique avec exemples[30]

II.2 Types d'apprentissage :

Dans notre cas, nous nous concentrerons sur les types d‟‟apprentissage à savoir supervisé et non-supervisé.

II.2.1 Apprentissage supervisé :

La forme la plus courante d'apprentissage automatique est l'apprentissage supervisé.

Dans l'apprentissage supervisé, l'ordinateur reçoit des exemples d'entrées déjà étiquetés avec

les sorties souhaitées. L'objectif est que l'algorithme puisse "apprendre" en comparant ses prédictions avec les sorties attendues, ajustant ainsi le modèle en conséquence. Ce processus utilise des modèles pour prédire les valeurs des étiquettes sur de nouvelles données non étiquetées.

Par exemple, dans un scénario d'apprentissage supervisé, un algorithme peut être entraîné avec des images étiquetées de requins (étiquetées comme "Poisson") et des images océaniques (étiquetées comme "Océan"). Après avoir été formé sur ces données, l'algorithme devrait être capable d'identifier ultérieurement des images non étiquetées de requins comme des poissons et des images non étiquetées d'océans comme des océans. Cela peut être appliqué dans divers contextes, tels que la prédiction d'événements futurs basée sur des données historiques, comme les fluctuations du marché boursier ou la filtration des courriers indésirables.

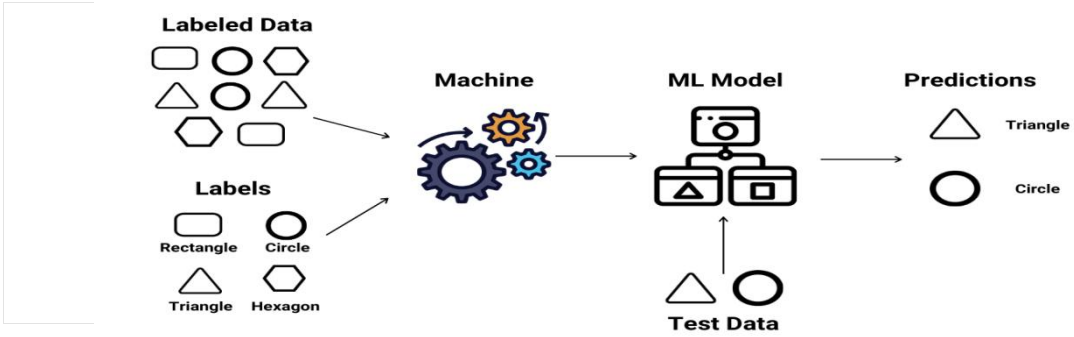

Figure 2. 4: L'Apprentissage supervisé[Online, 31]

- **Formulation [29]**

Supposons qu'un jeu de données d'apprentissage est formulé par N variables (descripteur de données à N valeurs) le décrivant (par exemple, le cas des images en forme de matrices d'entiers).

Nous disposons alors, pour un ensemble donné, de deux types d'informations : un **vecteur de valeurs** $X = x_1 , \dots , x_N$ prises par chaque variable, et une valeur de sortie Y appelée valeur supervisée ou réponse supervisée (qui peut être une classe si l'on prend un problème de classification).

Si nous formalisons le problème d'apprentissage décrit précédemment, nous pouvons le

représenter comme un ensemble de couples entrée-sortie $\boldsymbol{X_i}, \boldsymbol{Y_i}$, avec $i \in 1, n$, n étant le nombre d'exemples ou d'échantillons disponibles.

On appelle alors ***fonction d'apprentissage*** la fonction notée : $\boldsymbol{l} : X \rightarrow Y$ qui associe un résultat (valeur) supervisé à chaque vecteur d'entrée.

Le but d'un algorithme d'apprentissage supervisé sera donc d'approcher cette fonction $\boldsymbol{l}$, uniquement à partir des exemples d'apprentissage.

- **Types de Tâches:**

L'apprentissage supervisé englobe deux principales catégories de tâches : la classification et la régression.

- **Classification**

La classification implique la recherche ou la découverte d'un modèle ou d'une fonction permettant de séparer les données en plusieurs catégories discrètes. Dans ce processus, les données sont attribuées à différentes étiquettes en fonction de paramètres spécifiques en entrée, puis des étiquettes sont prédites pour de nouvelles données. Ce processus traite des problèmes où les données peuvent être divisées en étiquettes discrètes binaires ou multiples en fonction de certains critères.telles que la classification d'images en différentes classes (chats, chiens, oiseaux). Par exemple, un modèle peut être entraîné sur un ensemble de données d'images d'animaux étiquetées pour reconnaître automatiquement les différents types d'animaux.

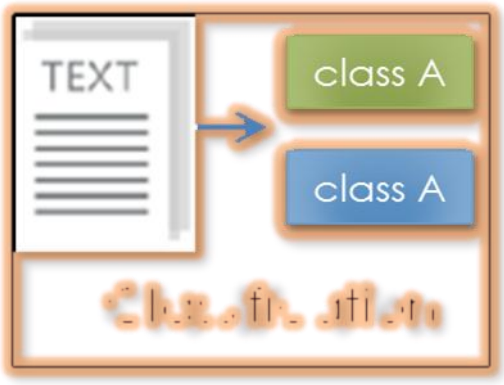

Figure 2. 5: La tache de Classification [32]

- **Régression**

La régression est un processus consistant à trouver un modèle ou une fonction permettant de distinguer les données en valeurs réelles continues au lieu d'utiliser des classes ou des valeurs discrètes. La régression peut également identifier le mouvement de distribution en fonction des données historiques.

En d'autres termes, la régression sert à trouver la relation d'une variable par rapport à une ou plusieurs autres, dans le but d'estimer une valeur (numérique) de sortie à partir des valeurs d'un ensemble de caractéristiques en entrée. Par exemple, estimer le prix d'une maison en se basant sur sa surface, nombre des étages, son emplacement · · · etc.

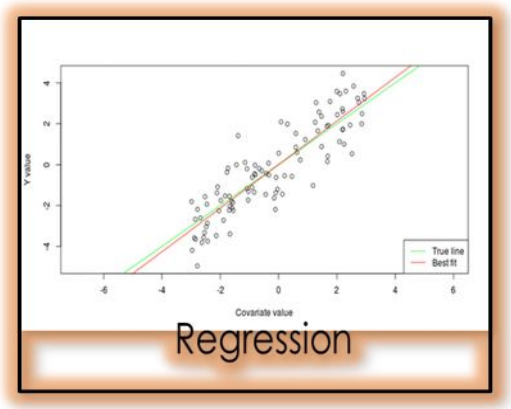

Figure 2. 6: La tache de Regression[32]

II.2-2 L'apprentissage Non Supervisé

L'apprentissage non supervisé implique l'utilisation de données non étiquetées, ce qui permet à l'algorithme d'identifier de manière autonome des similarités entre ces données. Cette approche est particulièrement bénéfique étant donné que les données non étiquetées sont souvent plus abondantes que les données étiquetées. Les objectifs de l'apprentissage non supervisé peuvent varier, allant de la détection de modèles cachés à la création de représentations significatives des données brutes pour la classification automatique. Cette méthode est fréquemment employée dans des domaines tels que l'analyse de données transactionnelles, où elle permet de découvrir des patterns subtils, comme l'identification de clients susceptibles d'être enceintes en se basant sur leurs achats. De plus, les techniques d'apprentissage non supervisé sont utilisées pour la détection d'anomalies, comme la fraude par carte de crédit, ainsi que dans les systèmes de recommandation. Par exemple, dans le

domaine de la vision par ordinateur, cet apprentissage peut être utilisé pour regrouper des images de chiens similaires sans avoir besoin d'étiquettes préalables.

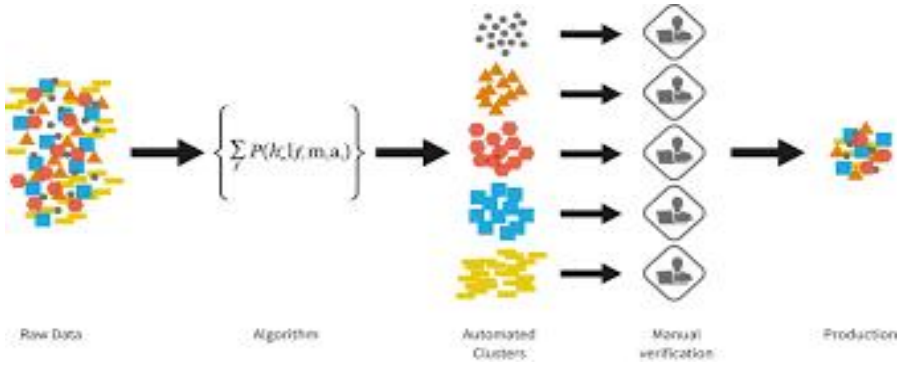

Figure 2. 7: L'Apprentissage non supervisé*[Online, 33]* **.**

- **Types de Tâches:**

Les principales catégories d'algorithmes d'apprentissage non supervisé comprennent le regroupement (clustering) et l'association.

- **Regroupement (Clustering)**

possèdent une mesure de similarité plus élevée que les données de n'importe quel autre cluster.Ils permettent de découvrir des structures inhérentes aux données sans étiquettes préalables.

L'analyse de cluster est L'analyse de cluster implique l'application d'un ou plusieurs algorithmes de clustering avec pour objectif de trouver les modèles ou les groupements cachés dans un Dataset. Les algorithmes de clustering permettent de former des groupements ou des clusters de manière à ce que les données d'un clustert utilisée en bio-informatique pour les analyses de séquences et les regroupements génétiques, en Data Mining pour l'extraction de séquences et de modèles, en imagerie médicale pour les segmentations d'images.

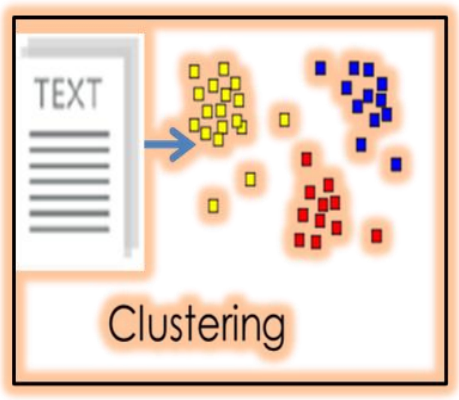

Figure 2. 8: La tache de Regroupement(Clustering)[32].

- **Association**

Les algorithmes d'association sont utilisés pour découvrir des règles ou des relations intéressantes entre les variables dans un ensemble de données. Ils permettent de mettre en évidence des tendances ou des corrélations qui peuvent être utilisées pour prendre des décisions ou faire des recommandations. Par exemple, dans le domaine du commerce électronique, un algorithme d'association peut être utilisé pour identifier les produits qui sont fréquemment achetés ensemble, ce qui peut aider à proposer des recommandations de produits croisés aux clients.

II.3.Table de Comparaison:

Paramètres	Apprentissage automatique supervisé	Apprentissage automatique non supervisé
Données d'entrée	Les algorithmes sont formés à l'aide de données étiquetées.	Les algorithmes sont utilisés contre des données qui ne sont pas étiquetées.
Complexité informatique	Une méthode plus simple	Complexe sur le plan informatique
Précision	Haute précision	Moins précis
Nombre de classes	Le nombre de classes est connu	Le nombre de classes n'est pas forcément connu

Tableau 2. 1 : Différence entre l'apprentissage supervisé et non supervisé.

III. IDS par les méthodes d'Apprentissage

Ce qui suit est une liste non exhaustive des algorithmes d'Apprentissage qui pourraient être envisagées pour la mise en place des IDS.

III.1 Arbre De décision [34] :

Cette technique divise le problème de classification en sous-problèmes. Il crée un arbre de décision qui est utilisé pour développer un modèle utilisé aux fins de la classification. Effectuer une classification est relativement simple. Il suffit de répondre aux questions en descendant dans l'arbre de décision.

Les arbres de décision sont des modèles non-paramétriques et trouvent des règles en général assez puissantes. Ils peuvent traiter des très grands data-set et ils peuvent aussi utiliser des prédicateurs mixtes (catégoriques et nombres). Les variables redondantes sont éliminées, parce que l'arbre ne les prend même pas en compte.A chaque nœud, on doit donc trouver deux choses : quelle features utiliser et quel est le point de séparation des deux zones, de sorte à minimiser l'erreur quadratique pour la régression et l'impureté pour la classification. On fait ainsi grandir l'arbre de la manière la plus grande possible.

Les forêts aléatoires sont donc un ensemble d'arbres de décisions entraînés individuellement, légèrement différents les uns des autres. Pour prédire une nouvelle valeur, on effectue la classification pour chaque arbre de cette forêt. La forêt choisit la valeur ayant le plus de votes parmi tous ses arbres.

Pour résumer, les forêts aléatoires sont nommées ainsi parce qu'on injecte de l'aléatoire dans la création de chaque arbre. Premièrement avec la création d'échantillons aléatoires (bootstraping). Deuxièmement, avec la sélection des features sur lesquels effectuer le test de séparation.

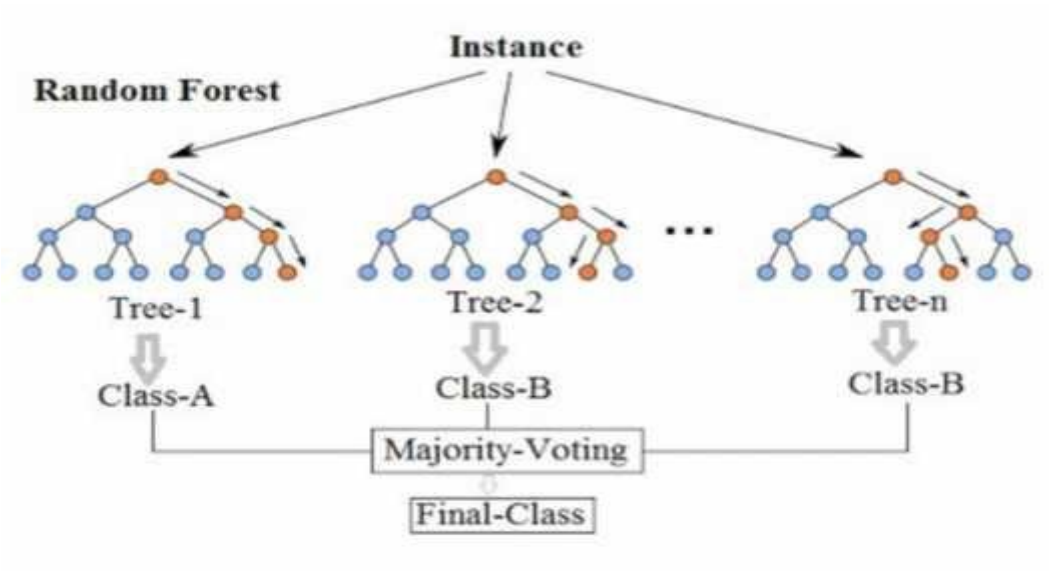

Figure 2. 9 : algorithme des arbres de décision [34].

III.2 L'algorithme des K plus proche voisin (KNN):

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

Cette méthode supervisée et non-paramétrique est souvent performante. De plus, son apprentissage est assez simple, car il est de type apprentissage par coeur. Cependant, le temps de prédiction est très long, car il nécessite le calcul de la distance avec tous les exemples, mais il existe des heuristiques pour réduire le nombre d'exemples à prendre en compte[35]

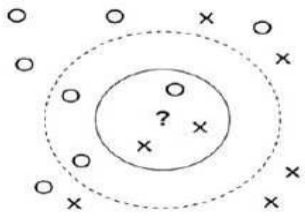

Figure 2. 10: Algorithme KNN[34]

III.3 Naïve bayes [34] :

C'est une technique de classification basée sur le théorème de Bayes avec une hypothèse d'indépendance parmi les prédicateurs. En termes simples, un classificateur Naïve Bayes suppose que la présence d'une fonctionnalité particulière dans une classe n'est pas liée à la présence d'une autre fonctionnalité. Par exemple, un fruit peut être considéré comme une pomme s'il est rouge, rond et d'environ 3 pouces de diamètre. Même si ces caractéristiques dépendent les unes des autres ou de l'existence d'autres caractéristiques, toutes ces propriétés contribuent indépendamment à la probabilité que ce fruit soit une pomme et c'est pourquoi il est appelé «naïf».

Le modèle Naïve Bayes est facile à construire et particulièrement utile pour les très grands

ensembles de données.

Outre sa simplicité, Un de ces points fort est le besoin d'une faible quantité d'informations pour la phase d'apprentissage. Naïve Bayes est réputé a surpasser même les méthodes de classification les plus sophistiquées

Bayes theorem:

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

Équation 1 : *Theorem de Naive Bayes.*

Le terme P (A \ B) se lit **:** la probabilité que l'événement A se réalise sachant que l'événement B s'est déjà réalisé. https://mrmint.fr/naive-bayes-classifier

$$P(A \quad \text{e} \quad A$$
$$c_1 \qquad t_1$$

$$P(B \cap A \quad) = \frac{C_1 \quad (B \cap A)}{C_1 \quad t_1}$$

Note:

- le cardinal d'un ensemble est le nombre d'éléments dans ce dernier
- cardinal total représente l'ensemble total des instances

Cas ou on a plusieurs paramètres / caractéristiques :

$$P(y|x_1,...,x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

Équation 2 *:Naive bayes a plusieurs caractéristiques.*

- **Avantage :**

✓ Le Naïve Bayes Classifier est très rapide pour la classification : en effet les calculs de probabilités ne sont pas très coûteux.
✓ La classification est possible même avec un petit jeu de données

- **Inconvénients :**

Contre intuitivement, malgré la violation de la contrainte d'indépendance des variables, Naïve Bayes donne de bons résultats de classification.

III.4 SVM (support a vecteurs 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[36].

III.5 Régression logistique

La régression logistique est essentiellement un algorithme de classification supervisée. Dans un problème de classification, la variable cible (ou sortie), y, ne peut prendre que des valeurs discrètes pour un ensemble donné de caractéristiques (ou entrées), X. Le modèle construit un modèle de régression pour prédire la probabilité qu'une entrée de données donnée appartienne à la catégorie numérotée "1". Tout comme la régression linéaire suppose que les données suivent une fonction linéaire, la régression logistique modélise les données à l'aide de la fonction sigmoïde. La régression logistique ne devient une technique de classification que lorsqu'un seuil de décision est introduit dans l'image. La fixation de la valeur du seuil est un aspect très important de la régression logistique et dépend du problème de classification lui même. La décision concernant la valeur du seuil est principalement affectée par les valeurs de précision et de rappel. Idéalement, nous souhaitons que la précision et le rappel soient égaux à 1, mais c'est rarement le cas.

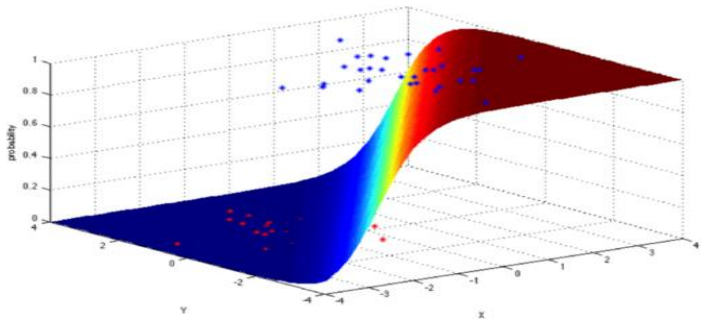

Figure 2. 11: Regression Logistique[36].

III.6 Régression linéaire:

La régression linéaire est un algorithme d'apprentissage automatique supervisé utilisé pour effectuer des tâches de régression. Elle modélise une valeur cible prédictive basée sur des variables indépendantes et est principalement utilisée pour découvrir la relation entre ces variables et faire des prévisions. Les différents modèles de régression varient selon le type de relation qu'ils considèrent entre les variables dépendantes et indépendantes, ainsi que selon le nombre de variables indépendantes utilisées. La régression linéaire vise à prédire la valeur d'une variable dépendante (y) en fonction d'une variable indépendante donnée (x), établissant ainsi une relation linéaire entre x (entrée) et y (sortie). Par exemple, si x représente l'expérience professionnelle et y le salaire, la ligne de régression représente la meilleure ligne d'ajustement pour ce modèle.

Lors de l'apprentissage du modèle, on utilise :

x : données d'entrée pour l'apprentissage (univarié - une seule variable d'entrée).

y : étiquettes des données (pour l'apprentissage supervisé).

Le modèle ajuste une ligne de régression optimale pour prédire y à partir de x en trouvant les meilleures valeurs pour les paramètres du modèle. Une fois ces valeurs déterminées, la ligne d'ajustement est utilisée pour prédire y pour toute nouvelle valeur de x. Ainsi, la régression linéaire permet de prévoir la valeur de la variable dépendante en fonction de la variable indépendante en utilisant la meilleure ligne d'ajustement déterminée lors de l'apprentissage du modèle[37].

III.7 Analyse discriminante linéaire:

L'analyse discriminante linéaire ou analyse discriminante normale ou analyse de la fonction discriminante est une technique de réduction de la dimensionnalité couramment utilisée pour les problèmes de classification supervisée. Elle est utilisée pour modéliser les différences entre les groupes, c'est-à-dire pour séparer deux ou plusieurs classes. Elle est utilisée pour projeter les caractéristiques d'un espace de dimension supérieure dans un espace de dimension inférieure. Par exemple, nous avons deux classes et nous devons les séparer efficacement. Les classes peuvent avoir plusieurs caractéristiques. L'utilisation d'une seule caractéristique pour les classer peut entraîner un certain chevauchement, comme le montre la figure ci-dessous. Nous allons donc continuer à augmenter le nombre de caractéristiques pour une classification correcte[37].

III.8 K-Means

L'algorithme K-Means est un algorithme de clustering, qui regroupe l'ensemble de données non étiquetées en différents clusters. Ici, K définit le nombre de clusters qui doivent être créés dans le processus, comme si K = 2, il y aura deux clusters, et pour K=3, il y aura trois clusters, et ainsi de suite. Il nous permet de regrouper les données en différents groupes et constitue un moyen pratique de découvrir les catégories de groupes dans l'ensemble de données non étiquetées sans avoir besoin d'entraînement. Il s'agit d'un algorithme basé sur les centroïdes, où chaque groupe est associé à un centroïde. L'objectif principal de cet algorithme est de minimiser la somme des distances entre le point de données et les clusters correspondants. L'algorithme prend l'ensemble de données non étiquetées comme entrée, divise l'ensemble de données en un nombre k de clusters, et répète le

processus jusqu'à ce qu'il ne trouve pas les meilleurs clusters. La valeur de k doit être prédéterminée dans cet algorithme.

Conclusion

Dans cette seconde section, nous avons exploré le domaine de l'Intelligence Artificielle ainsi que ses diverses branches. Nous avons particulièrement mis en lumière l'Apprentissage Automatique des Données, en examinant ses concepts fondamentaux, les différentes catégories de classification, ainsi que les multiples techniques et algorithmes utilisés pour découvrir des schémas au sein de vastes ensembles de données, générant ainsi des informations exploitables.

Ces connaissances, à un niveau avancé, s'avéreront précieuses dans la conception de notre système de détection d'intrusion.

Chapitre III : Méthodes d'apprentissage Pour la détection d'intrusion

Introduction

Les systèmes de détection d'intrusions (IDS) sont devenus un élément essentiel pour sécuriser les réseaux contre les attaques croissantes. Un défi majeur pour les évaluateurs d'IDS est de garantir leur bon fonctionnement, en s'assurant qu'ils déclenchent des alertes en cas d'intrusion réelle tout en évitant les fausses alertes.

Les IDS basés sur la classification visent à classifier le trafic réseau en deux catégories : "normal" et "intrusion". Cette classification nécessite un processus d'apprentissage pour assurer une précision optimale, réduisant ainsi les faux positifs (cas normaux classés comme intrusions) et les faux négatifs (intrusions classées comme normales).

Ce chapitre présente notre solution, comprenant deux grandes étapes. Tout d'abord, nous décrivons l'implémentation de notre IDS pour rendre les informations des paquets réseau compatibles avec notre jeu de données, qui est le NLS- KDD. Ensuite, nous utilisons des algorithmes de classification, tels que KNN et SVM, pour prédire si un paquet réseau est normal ou une attaque.

Enfin, nous expliquons les différentes mesures de performance que nous utiliserons pour comparer ces algorithmes et évaluer l'efficacité de notre système de détection d'intrusions.

I. Les Algoritmes utilisées:

I.1 L'algorithme des K plus proche voisin (KNN):

L'algorithme des k-plus proches voisins (KNN) est une méthode d'apprentissage à base d'instances. Il ne comporte pas de phase d'apprentissage en tant que telle. Cette méthode enregistre toutes les classes qui lui sont fournies à l'aide de données d'apprentissage, définit et classe les nouvelles instances en fonction d'une mesure de similarité. Ses k plus proches voisins sont alors considérés : on observe leur classe/catégorie, et celle qui revient le plus (La classe majoritaire) parmi les voisins est affectée aux nouvelles instances à classer.

La distance peut, en général, être n'importe quelle mesure métrique: la distance euclidienne standard est le choix le plus courant. Les méthodes basées sur les voisins sont connues sous le nom de méthodes d'apprentissage automatique non généralisables, car elles «se souviennent» simplement de toutes ses données d'apprentissage

I.1.1 Données en entrée :

un ensemble de données D.

une fonction de définition distance d

Un nombre entier K

Pour une nouvelle observation X dont on veut prédire sa variable de sortie y Faire :

- Calculer toutes les distances de cette observation X avec les autres observations du jeu de données D.
- Retenir les K observations du jeu de données D les proches de X en utilisation la fonction de calcul de distance.
- Prendre les valeurs de y des K observations retenues.

1. Si on effectue une régression, calculer la moyenne (ou la médiane) de y Retenues.
2. Si on effectue une classification, calculer le mode de y retenues.

- Retourner la valeur calculée dans l'étape 3 comme étant la valeur qui a été prédite par
- K-NN pour l'observation X.

I.1.2 Calcul de similarité dans l'algorithme K-NN

Il existe plusieurs fonctions de calcul de distance, notamment, la distance euclidienne, la distance de Manhattan, la distance de Minkowski …etc. On choisit la fonction de distance en fonction des types de données qu'on manipule. Ainsi pour les données quantitatives (exemple: poids, salaires, taille, montant de panier électronique etc.…) et du même type, la distance euclidienne est un bon candidat. Quant à la distance de Manhattan, elle est une bonne mesure à utiliser quand les données (input variables) ne sont pas du même type (exemple : âge, sexe, longueur, poids etc.…).

Il est inutile de coder, soi-même ces distances, généralement, les librairies de Machine Learning comme Scikit Learn, effectue ces calculs en interne. Il suffit juste d'indiquer la mesure de distance qu'on souhaite utiliser.

I.1.3 Calcul de distance :

a) La distance euclidienne:

- distance qui calcule la racine carrée de la somme des différences carrées entre les coordonnées de deux points :

$$D_e(x,y) = \sqrt{\sum_{j=1}^{n}(x_j - y_j)^2}$$

Équation 1: *la distance euclidienne.*

b) La distance de Manhattan :

- la distance de Manhattan: calcule la somme des valeurs absolues des différences entre les coordonnées de deux points :

$$D_m(x,y) = \sum_{i=1}^{k} | x_i - y_i |$$

Équation 2: *Distance de Manhattan.*

I.1.4 Choix de K :

Le choix de la valeur de **k** dans l'algorithme des k-plus proches voisins **(k-NN)** est crucial pour la performance du modèle. Un petit **k** (par exemple, k=1) peut entraîner un sur-apprentissage (overfitting), où le modèle s'adapte trop précisément aux données d'entraînement, capturant ainsi le bruit et les particularités qui ne se généralisent pas bien aux nouvelles données. En

revanche, un **k** trop grand peut mener à un sous-apprentissage (underfitting), où le modèle devient trop rigide et lisse trop les prédictions, manquant ainsi des motifs importants dans les données. La clé est de trouver un bon équilibre entre ces deux extrêmes. La validation croisée est une technique couramment utilisée pour déterminer la valeur optimale de **k**. Elle consiste à diviser les données en plusieurs sous-ensembles, à entraîner le modèle sur certains de ces sous-ensembles et à le tester sur les autres, en répétant le processus pour différentes valeurs de **k**.

Le **k** qui minimise l'erreur de validation croisée est généralement choisi comme la meilleure valeur pour les prédictions futures. En somme, un choix judicieux de **k** permet de trouver un compromis entre flexibilité et généralisation, améliorant ainsi l'efficacité du modèle **k-NN**.

Exemple : dans l'exemple suivant, si K=3 il (le ?) sera classé avec les triangles, si K=5 il sera classé avec les carrés[38].

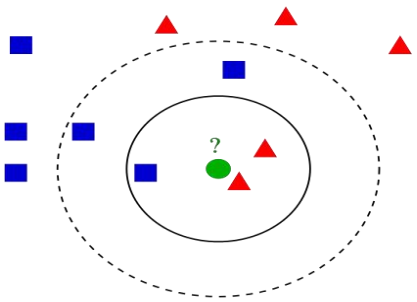

Figure 3. 1: **Algorithme KNN [39]**

I.2 SVM (support a vecteurs machine)

Les machines à vecteurs de support (SVM) constituent une catégorie d'algorithmes d'apprentissage automatique essentiels, utilisés pour résoudre une gamme variée de problèmes tels que la classification, la régression ou la détection d'anomalies. Originellement développées dans les années 1990, elles se distinguent par leur capacité à trouver un hyperplan optimal dans un espace de dimension N, où N représente le nombre de caractéristiques des données. Cet hyperplan agit comme une frontière de décision qui sépare efficacement les différents points de données dans l'espace, permettant ainsi une classification précise et une généralisation robuste. En d'autres termes, les SVM visent à trouver la meilleure séparation linéaire entre les différentes classes de données, ce qui en fait un outil puissant et polyvalent pour la

modélisation et l'analyse de données.

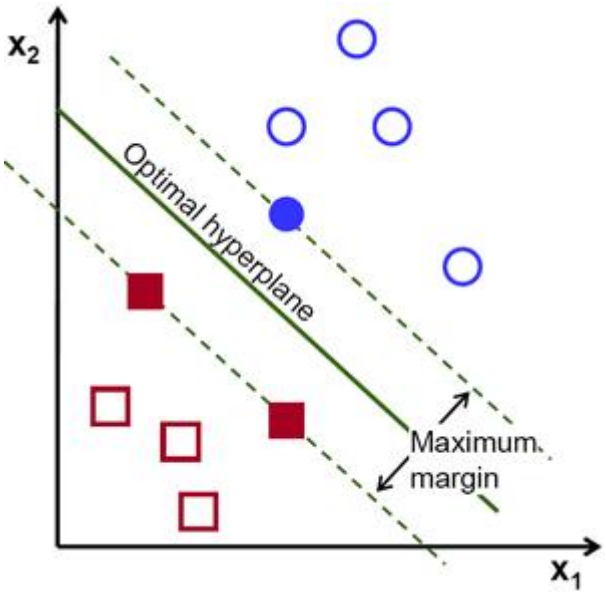

Figure 3. 3 Hyperplan optimal [39]

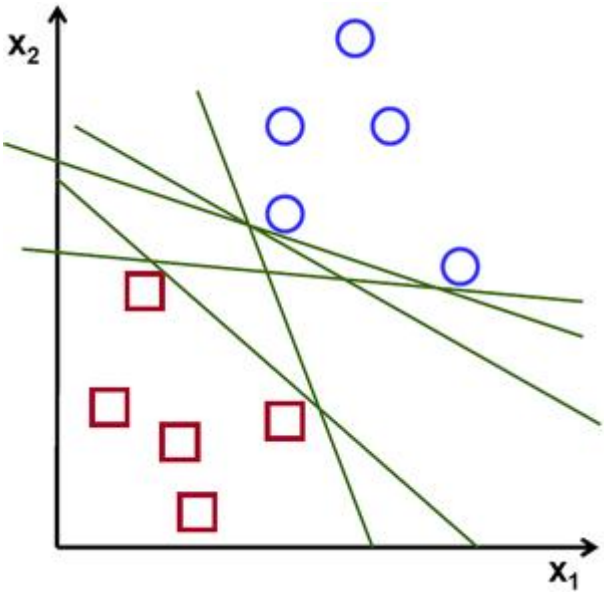

Figure 3. 2: Hyperplans possibles[39]

Lorsqu'il s'agit de séparer les deux classes de points de données, il est important de noter qu'il existe une multitude d'hyperplans possibles qui pourraient être sélectionnés. Notre objectif fondamental est donc de trouver un plan qui offre une marge maximale, ce qui correspond à la plus grande distance possible entre les points de données des deux classes.

Cette maximisation de la marge présente un avantage crucial : elle fournit une certaine marge d'erreur, permettant ainsi une meilleure généralisation et une classification plus confiante des futurs points de données.

En d'autres termes, en choisissant un hyperplan avec une marge maximale, nous obtenons une frontière de décision plus robuste, capable de mieux séparer les différentes classes et de fournir une meilleure capacité de prédiction pour de nouvelles données.

I.2.1 Intuition de la grande marge :

Dans l'algorithme d'apprentissage SVM (Support Vector Machine), l'intuition de la grande marge est cruciale. Contrairement à la régression logistique, qui utilise une fonction sigmoïde pour comprimer les sorties dans la plage [0,1] et attribuer des étiquettes basées sur un seuil de 0,5, le SVM utilise des seuils de 1 et -1 pour classer les exemples. Les valeurs comprises entre -1 et 1 définissent une marge autour de l'hyperplan séparateur. L'objectif du SVM est de maximiser cette marge, c'est-à-dire de trouver l'hyperplan qui sépare les classes avec la plus grande distance possible entre les points les plus proches de chaque classe et l'hyperplan. En maximisant la marge, le SVM améliore la généralisation du modèle, rendant celui-ci moins sensible aux variations dans les données d'entraînement et plus robuste face aux

nouvelles données.

Cette approche contraste avec celle de la régression logistique, qui se concentre sur l'ajustement des probabilités dans une plage fixe pour la classification.

En résumé, l'intuition de la grande marge dans le SVM permet de créer un modèle robuste et généralisable en maximisant la distance entre les classes.

I.2.2 Fonction de coût et mises à jour du gradient

Dans l'algorithme SVM, nous cherchons à maximiser la marge entre les points de données et l'hyperplan. La fonction de perte qui permet de maximiser la marge est Hing Loss :

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

Le coût est de 0 si la valeur prédite et la valeur réelle sont de même signe. Si elles ne le sont pas, nous calculons alors la valeur de la perte. Nous ajoutons également un paramètre de régularisation à la fonction de coût. L'objectif du paramètre de régularisation est d'équilibrer la maximisation de la marge et la perte. Après avoir ajouté le paramètre de régularisation, les fonctions de coût se présentent comme suit.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

Maintenant que nous avons la fonction de perte, nous prenons les dérivées partielles par rapport aux poids pour trouver les gradients. En utilisant les gradients, nous pouvons mettre à jour nos poids.

$$\frac{\delta}{\delta w_k} \lambda \parallel w \parallel^2 = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k} (1 - y_i \langle x_i, w \rangle)_+ = \begin{cases} 0, & \text{if } y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases}$$

Lorsqu'il n'y a pas d'erreur de classification, c'est-à-dire que notre modèle prédit correctement la classe de notre point de données, nous devons seulement mettre à jour le gradient du paramètre de régularisation.

$$w = w - \alpha \cdot (2\lambda w)$$

Lorsqu'il y a une mauvaise classification, c'est-à-dire que notre modèle se trompe dans la

prédiction de la classe de notre point de données, nous incluons la perte avec le paramètre de régularisation pour
effectuer la mise à jour du gradient[39]

$$w = w + \alpha \cdot (y_i \cdot x_i - 2\lambda w)$$

I.2.3 Algorithme SVM

Cet algorithme décrit les étapes de base pour entraîner et utiliser un SVM pour la classification. Les détails peuvent varier légèrement en fonction des spécificités du problème et des implémentations logicielles utilisées.

1.Initialisation :

Représentez les données dans un espace de caractéristiques de dimension n.

2.Sélection des Hyperparamètres :

Choisissez le type de noyau (linéaire, polynomial, RBF, etc.) si nécessaire.
Définissez les paramètres du noyau et la valeur de la pénalité pour la régularisation.

3. Formulation du Problème d'Optimisation :

Formulez un problème d'optimisation pour trouver l'hyperplan qui maximise la marge entre les classes tout en minimisant les erreurs de classification.

4. **Résolution du Problème d'Optimisation :**

Utilisez des méthodes d'optimisation pour résoudre le problème et trouver les paramètres de l'hyperplan.

5. Détermination des Vecteurs de Support :

Identifiez les vecteurs de support, qui sont les points de données les plus proches de l'hyperplan.

6. Construction de l'Hyperplan :

Construisez l'hyperplan optimal en utilisant les vecteurs de support.

7. **Prédiction :**

Pour une nouvelle donnée, calculez la valeur de la fonction de décision en utilisant les paramètres de l'hyperplan.

8. **Classez la nouvelle donnée en fonction du signe de la fonction de décision :**

Si la valeur est positive, la donnée appartient à une classe.
Si la valeur est négative, la donnée appartient à l'autre classe.

9. Utilisation du Noyau (si nécessaire) :

Si un noyau non linéaire est utilisé, transformez les données d'origine dans un espace de

caractéristiques de plus haute dimension à l'aide de la fonction noyau.

II. L'architecture de modèle d'apprentissage automatique :

Nous devons tout d'abord passer par les étapes de machine Learning, au début on télécharge le jeu de donnée qui contient jeu de donne de test et jeu de donnée de train, après on fait le prétraitement (la normalisation et encodage de donnée) ensuite on crée des modèles d'apprentissage automatique à la fin on teste les modèles avec le jeu de donnée de test.
Dans ce qui suit on va détailler ces étapes d'apprentissage pour mieux comprendre le processus de son fonctionnement:

II.1 Collecte des Données:

Tout d'abord, des données de trafic réseau sont collectées à partir des divers points de surveillance dans le réseau. Ces données peuvent inclure des informations telles que les adresses IP source et destination, les ports utilisés, les protocoles de communication, les volumes de données échangées, etc.

II.2 Prétraitement des Données:

Les données collectées sont ensuite prétraitées pour les rendre utilisables par l'algorithme. Cela peut inclure des étapes telles que la normalisation des valeurs, le filtrage du bruit, la réduction de la dimensionnalité, etc. L'objectif est de préparer les données de manière à ce qu'elles puissent être efficacement analysées par l'algorithme d'apprentissage automatique.

II.3 Entraînement du Modèle:

Une fois les données prétraitées, l'algorithme utilise un ensemble d'apprentissage pour entraîner un modèle de comportement normal du réseau. Ce modèle est basé sur les schémas typiques de trafic observés dans le réseau au fil du temps. Différentes techniques d'apprentissage automatique peuvent être utilisées, telles que les réseaux de neurones, les arbres de décision, les machines à vecteurs de support (SVM), etc.

II.4 Détection des Anomalies:

Une fois que le modèle est entraîné, il est utilisé pour surveiller en temps réel le trafic réseau entrant. L'algorithme compare les caractéristiques du trafic observé avec le modèle de comportement normal. Toute déviation significative par rapport à ce modèle est considérée

comme une anomalie potentielle et peut déclencher une alerte.

II.5 Gestion des Alertes:

Lorsqu'une anomalie est détectée, l'algorithme génère une alerte pour signaler le problème. Cette alerte peut être transmise à un administrateur système ou à un système de gestion des incidents de sécurité pour enquête et réponse appropriée. La gestion des alertes peut également inclure des actions automatisées, telles que le blocage du trafic suspect ou la mise en place de contre-mesures de sécurité.

II.6 Adaptation et Mise à Jour:

Pour maintenir sa précision au fil du temps, le modèle conçu peut être périodiquement réentraîné avec de nouvelles données. Cela permet à l'algorithme de s'adapter aux changements dans les schémas de trafic et aux nouvelles menaces émergentes.

Voici Le fonctionnement globale de notre IDS comme le montre la **figure 3.4**

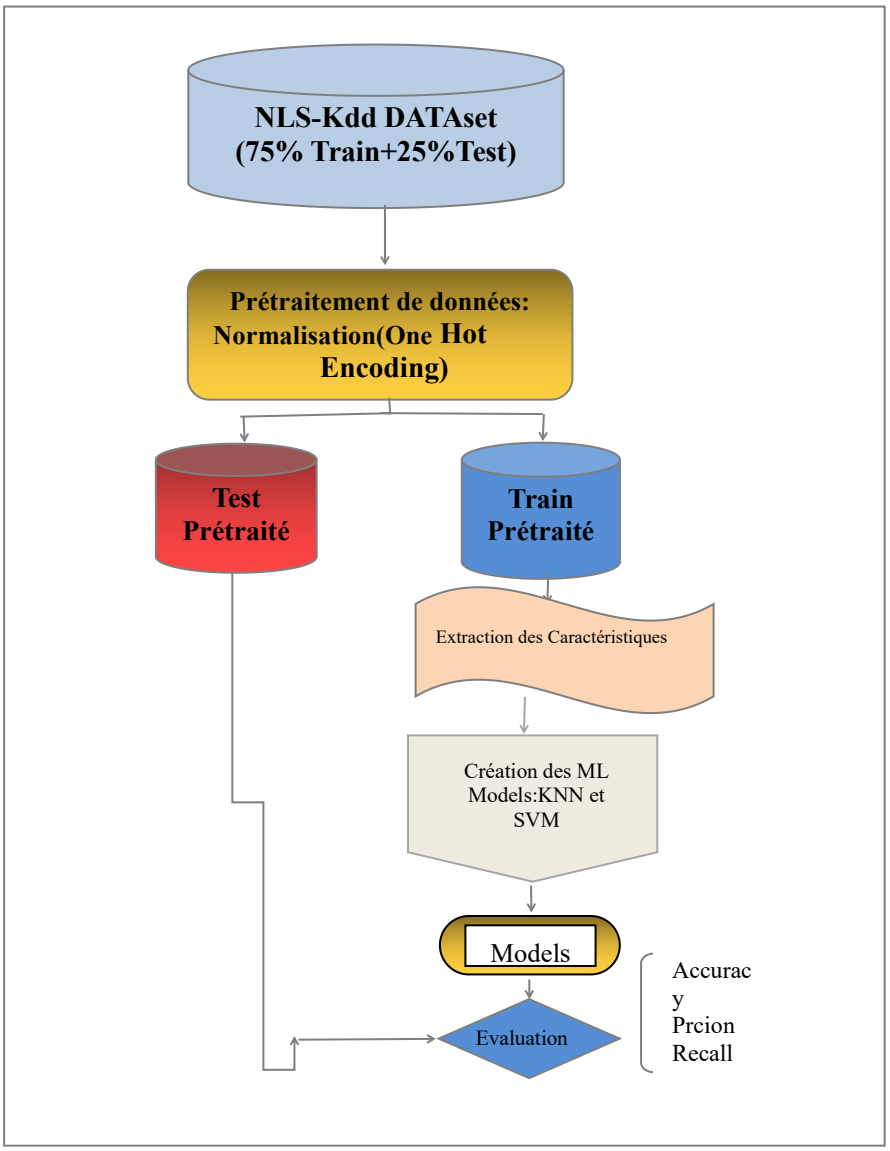

Figure 3. 4: L'architecture de modèle d'apprentissage automatique

III. Le choix du Data Set

Les systèmes intelligents de détection d'intrusion nécessitent un ensemble de données efficace pour être construits. Un ensemble de données riche en informations et qui imite le trafic en temps réel est essentiel pour former et évaluer la performance de ces systèmes. Le jeu de données NSL-KDD est une version améliorée du jeu de données KDD 99 original. Dans ce projet, nous analysons et utilisons le jeu de données NSL-KDD pour étudier l'efficacité de deux algorithmes de classification différents dans la détection des anomalies dans le trafic réseau.

Le jeu de données NSL-KDD comporte 42 attributs, représentant une amélioration notable par rapport au jeu de données KDD 99, dont les instances dupliquées ont été éliminées. Cela permet

d'éviter les résultats de classification biaisés qui peuvent découler de la présence de doublons. Les différentes configurations de cet ensemble de données varient par le nombre d'instances, mais conservent toujours les 42 attributs. L'attribut numéro 42, nommé «class», est particulièrement important car il indique si une instance donnée représente une connexion normale ou une attaque[40].

En utilisant cet ensemble de données, nous pouvons entraîner et tester divers algorithmes de classification pour déterminer leur efficacité dans la détection des intrusions et des anomalies réseau. L'objectif est d'identifier les algorithmes les plus performants pour améliorer la sécurité des systèmes informatiques en détectant et en répondant rapidement aux menaces.

Les classes d'attaques présentes dans l'ensemble de données **NLS-KDD** sont regroupées en quatre catégories :

III.1 Catégorie d'attaques dans le Dataset [41]:

- **DOS:**

Le déni de service est une catégorie d'attaque, qui épuise les ressources de la victime, l'empêchant ainsi de traiter des demandes légitimes. -par exemple syn. inondation. Caractéristiques pertinantes: "source bytes" and "percentage of packets with errors"

- **Objet de probation:**

L'objectif de la surveillance et vérification des autres attaques est d'obtenir des informations sur la victime distante, par exemple. Balayage de port.

Fonctionnalités pertinentes: "duration of connection" and "source bytes"

- **U2R:**

L'accès non autorisé aux privilèges super utilisateur (root) locaux est un type d'attaque, par lequel un attaquant utilise un compte normal pour se connecter au système victime et tente de gagner du temps. Les privilèges root / administrateur en exploitant une vulnérabilité de la victime, par exemple tampons par débordement attaqués. Caractéristiques pertinantes: "number of file creations" and "number of shell prompts invoked.

- **R2L :**

Accès non autorisé depuis une machine distante, l'attaquant s'introduit dans une machine distante et

obtient un accès local à la machine victime. Par exemple : Deviner le mot de passe. Caractéristiques

pertinentes : Network level features –"duration of connexion" and "service requested" and host level features –"number of failed login attempts"

Chaque types d'attaques et subdivisée en sous attaques présenté dans le **Figure 3.5**

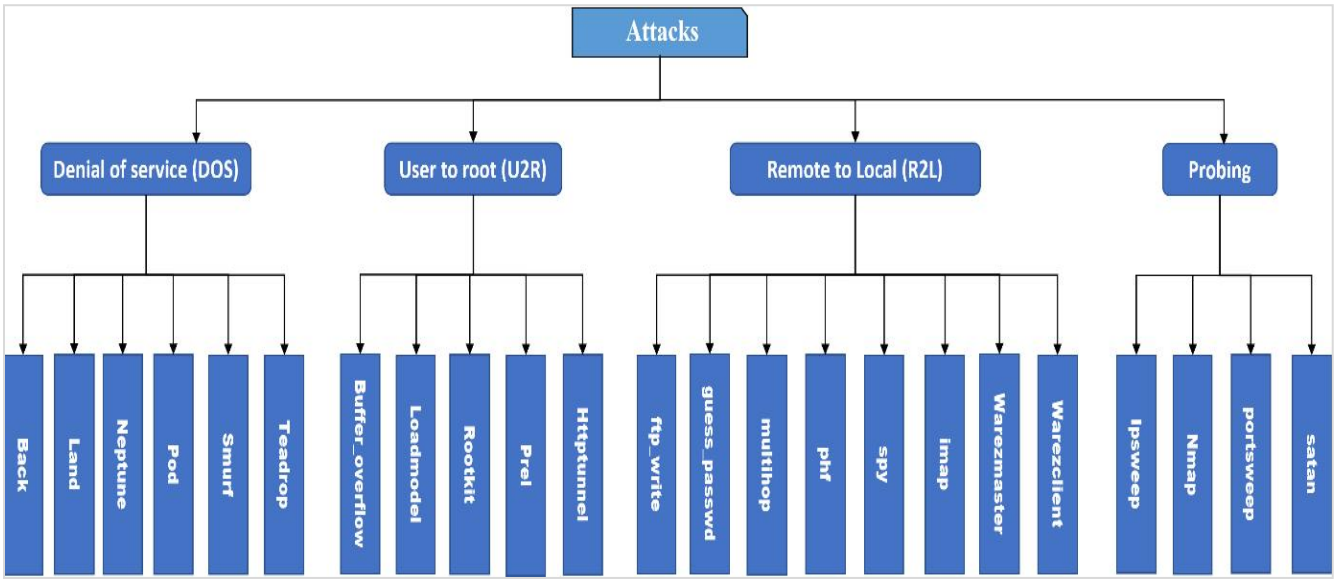

Figure 3. 5: les catégories et les types d'attaques[42]

La répartition des données est très déséquilibrée, avec une majorité de connexions normales (97,06 %) et une minorité de connexions malveillantes. L'attaque DoS est le type d'attaque le plus fréquent, représentant 80 % des connexions malveillante comme la montre le **Figure 3.6.**

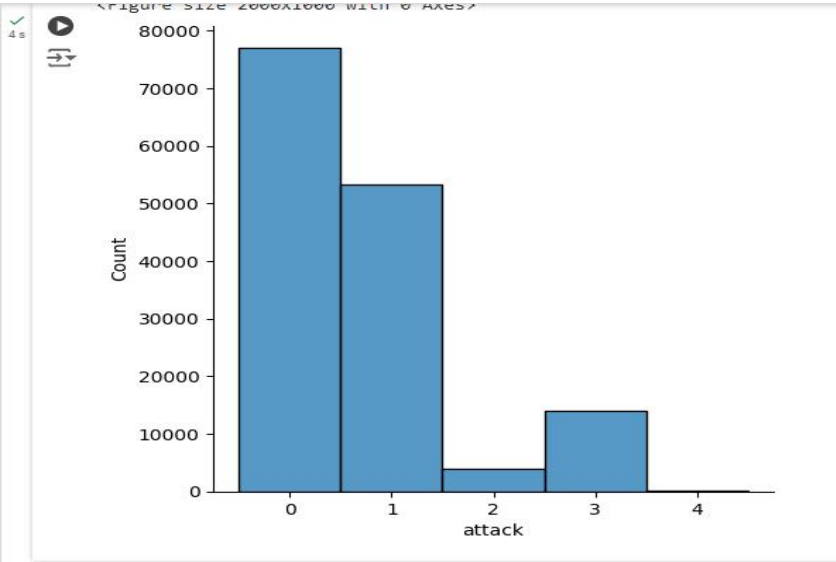

Figure 3. 6: Distribution des données dans la base NLS-KDD

III.2 Attributs de la base NLS-KDD

Dans la base NLS-KDD, les attributs peuvent être classés en quatre catégories comme l'illustre le tableau 3-1

Les attributs de base :qui sont les attributs de base des connexions TCP, telles que la durée, les hôtes source et destination, port et flag.

Les attributs du trafic :qui sont les attributs calculés à l'aide d'une fenêtre de temps de deux secondes, tels que le nombre de connexions vers la même machine.

Les caractéristiques du contenu :ces attributs sont construits à partir de la charge utile (Data) des paquets du trafic tels que le nombre d'échec de connexion et le nombre d'accès aux fichiers de contrôle.

Les caractéristiques de l'hôte :ce sont les attributs conçus pour évaluer les attaques qui durent plus de deux secondes.

Caractéristique de DoS, Probe, R2L, and U2R on NSL-KDD	
DoS	'dst_host_same_srv_rate' ; 'dst_host_serror_rate' ; 'num_compromised' ; 'same_srv_rate' ;'diff_srv_rate' ; 'dst_host_count' ; 'dst_host_srv_serror_rate' ; 'ecr_i' ; 'RSTR' ; 'wrong_fragment' ; 'dst_bytes' ; 'src_bytes.'
Prob	'src_bytes' ; 'telnet' ; 'smtp' ; 'private' ; 'http' ; 'ftp_data' ; 'finger' ; 'dst_host_rerror_rate' ; 'dst_host_same_src_port_rate' ; 'dst_host_diff_srv_rate' ; 'dst_host_same_srv_rate' ; 'rerror_rate' ; 'dst_bytes.
R2L	'duration' ; 'imap4' ; 'ftp_data' ; 'dst_host_srv_diff_host_rate' ; 'dst_host_same_src_port_rate' ; 'dst_host_same_srv_rate' ; 'dst_host_srv_count' ; 'dst_host_count' ; 'num_access_files' ; 'num_failed_logins' ; 'hot' ; 'dst_bytes' ; 'src_bytes.'
U2R	'duration' ; 'dst_host_srv_diff_host_rate' ; 'dst_host_count' ; 'srv_count' ; 'num_shells' ; 'num_file_creations' ; 'root_shell' ; 'dst_bytes' ; 'dst_host_same_srv_rate' ; 'hot' ; 'src_bytes.'

Tableau 3. 1: Les caractéristiques des données dans la base NLS-KDD[39]

- **Continu :** Ils représentent des entités avec des valeurs numériques qui peuvent prendre n'importe quelle valeur dans une plage. Les exemples incluent la durée d'une connexion réseau, les octets transférés ou le temps entre les paquets.

Discret : Ils représentent des fonctionnalités avec un ensemble limité de valeurs ou de catégories. Les exemples incluent le type de protocole (TCP, UDP, ICMP), le type de service (http, FTP, SSH) ou le type de connexion réseau (normal, suspect, attaque).

```
[ ]  pd.set_option('display.max_columns', None)
     print(fn)
```

+ Code + Texte

```
                      duration  continuous
0                protocol_type    symbolic
1                      service    symbolic
2                         flag    symbolic
3                    src_bytes  continuous
4                    dst_bytes  continuous
5                         land  continuous
6               wrong_fragment  continuous
7                       urgent  continuous
8                          hot  continuous
9               num_failed_logins  continuous
10                   logged_in  continuous
11             num_compromised  continuous
12                  root_shell  continuous
13               su_attempted  continuous
14                   num_root  continuous
15           num_file_creations  continuous
16                num_shells  continuous
17             num_access_files  continuous
18           num_outbound_cmds  continuous
19               is_host_login  continuous
20               is_guest_login  continuous
21                      count  continuous
22                  srv_count  continuous
23               serror_rate  continuous
24           srv_serror_rate  continuous
25               rerror_rate  continuous
26           srv_rerror_rate  continuous
27             same_srv_rate  continuous
28             diff_srv_rate  continuous
29         srv_diff_host_rate  continuous
30             dst_host_count  continuous
31         dst_host_srv_count  continuous
32     dst_host_same_srv_rate  continuous
33     dst_host_diff_srv_rate  continuous
34 dst_host_same_src_port_rate  continuous
35 dst_host_srv_diff_host_rate  continuous
36         dst_host_serror_rate  continuous
37     dst_host_srv_serror_rate  continuous
38         dst_host_rerror_rate  continuous
39     dst_host_srv_rerror_rate  continuous
```

Figure 3. 7: Les attributs de la base NLS-KDD

III.3 Distribution des connexions réseau dans la base NLS-KDD :

Type	*Base d'apprentissage*		*Base de test*	
	Nbr connexions	*Pourcentage*	*Nbr connexions*	*Pourcentage*
Normal	67343	53.46%	9711	43.07%
DoS	45927	36.46%	7591	33.67%
Probe	11656	09.25%	2421	10.74%
R2L	995	00.79%	2754	12.22%
U2R	52	00.04%	67	00.30%
Total	125973	100.0%	22544	100.0%

Tableau 3. 2: La distribution de connections dans la base NLS- KDD[43]

III.3 Fractionnement de l'ensemble de donnée : Le Fractionnement de l'ensemble de données en ensembles d'apprentissage et de test :

X_train : X irrévocable utilisé pour s'adapter au modèle d'apprentissage automatique.

X_test : X irrévocable utilisé pour évaluer l'ajustement du modèle d'apprentissage automatique.

Y_train : Y irrévocable utilisé pour s'adapter au modèle d'apprentissage automatique.

Y_test : Y irrévocable utilisé pour évaluer l'ajustement du modèle d'apprentissage automatique.

IV. Métriques d'évaluation

De nombreuses métriques sont utilisées pour évaluer les méthodes d'apprentissage automatique et d'apprentissage profond. Les modèles optimaux sont sélectionnés à l'aide de ces métriques

IV.1 Accuracy :

Est une mesure appropriée lorsque l'ensemble de données est équilibré. Dans des environne- ments réseau réels ; cependant, les échantillons normaux sont beaucoup plus abondants que les échantillons anormaux ; par conséquent, Accuracy peut ne pas être une mesure appropriée.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

IV.2 Précision (P) :

Est défini comme le rapport des échantillons positifs réels aux échantillons positifs prédits, il représente la confiance de la détection d'attaque.

$$P = \frac{TP}{TP + FP}$$

IV.3 Rappel (R) :

Est défini comme le rapport des vrais échantillons positifs au total des échantillons positifs et également appelé le taux de détection. Le taux de détection reflète la capacité du modèle à reconnaître les attaques, qui est une mesure importante dans les systèmes de détection d'intrus

$$R = \frac{TP}{TP + FN}$$

IV.4 Mesure-F (F) :

Est définie comme la moyenne pondérée de la précision et du rappel.

$$F = \frac{2 * P * R}{P+R}$$

IV.5 Le taux de faux négatifs (FNR) :

est défini comme le rapport des échantillons faussement négatifs au total des échantillons positifs.

Dans la détection d'attaque, le FNR est également appelé le taux d'alarme manquée.

$$FNR = \frac{FN}{TP + FN}$$

IV.6 Le taux de faux positifs (FPR) :

est défini comme le rapport des échantillons faux positifs aux échantillons positifs prédits. Dans la détection d'attaque, le FPR est également appelé le taux de fausse alarme, et il est calculé comme suit :

$$FPR = \frac{FP}{TN+FP}$$

Où :

TP (True Positive) est le nombre d'échantillons correctement distingués comme malveillants.

FP (False Positive) est le nombre d'échantillons identifiés à tort comme malveillants.

TN (True Negative) est le nombre d'échantillons correctement distingués comme bénins.

FN (False Negative) est le nombre d'échantillons identifiés à tort comme bénins[41].

IV.7 Matrice de confusion :

Est une disposition de tableau spécifique permettant de visualiser les performances d'un algorithme ML pour un problème de classification, elle est connue sous le nom de matrice d'erreur.

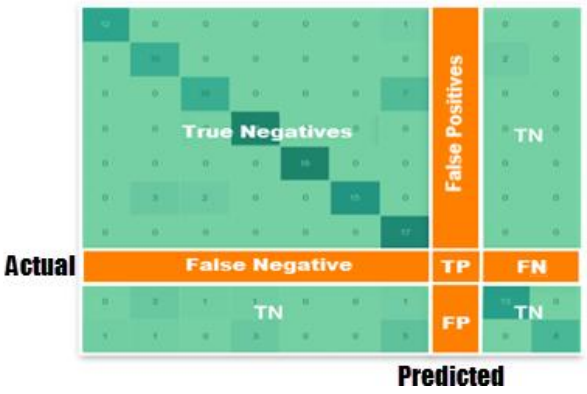

Figure 3. 8 : Illustration d'une matrice de confusion[44].

Conclusion :

Nous avons détaillé notre système de détection d'intrusion basé sur l'apprentissage automatique, en commençant par présenter les algorithmes utilisés (KNN et SVM). Nous avons ensuite décrit la conception de notre système, ainsi que le jeu de données employé et les métriques d'évaluation. Le chapitre suivant sera consacré à notre contribution et aux résultats obtenus en utilisant les techniques d'apprentissage automatique précédemment décrites.

Chapitre IV: Implémentation et Tests

Introduction :

Dans ce chapitre, nous allons examiner en détail les outils de développement que nous avons utilisés pour notre étude, notamment Python et Google Colab. Ces plates-formes

ont été essentielles pour la mise en œuvre et l'exécution de nos algorithmes d'apprentissage automatique. Python, grâce à ses bibliothèques robustes comme Scikit-learn, Pandas et NumPy, offre une flexibilité et une efficacité remarquables pour le traitement de données et la construction de modèles. Google Colab, de son côté, facilite l'exécution de scripts Python dans un environnement en ligne, offrant des ressources de calcul puissantes sans nécessiter de configuration locale complexe.

Nous présenterons les résultats obtenus à l'aide de deux algorithmes de classification populaires : le K-Nearest Neighbors (KNN) et les Support Vector Machines (SVM). Chacun de ces algorithmes sera évalué à travers des mesures de performance standards telles que l'accuracy, la précision, le rappel et le score F1. Ces métriques nous permettront de comprendre la performance et la pertinence de chaque modèle dans le contexte de notre jeu de données spécifique.

Enfin, nous comparerons les résultats obtenus par les deux modèles. Cette comparaison sera basée sur les différentes mesures d'évaluation mentionnées précédemment. En analysant les forces et les faiblesses de chaque algorithme, nous pourrons tirer des conclusions sur le modèle le plus adapté à notre problématique.

I. Les outils de développement:

I.1 Présentation de Python :

Python est un langage de programmation de haut niveau, interprété et polyvalent, qui a gagné en popularité au fil des années en raison de sa simplicité, sa lisibilité et sa robustesse. Créé par Guido van Rossum et publié pour la première fois en 1991, Python est conçu pour être facile à lire et à écrire, ce qui le rend particulièrement adapté aux débutants tout en étant suffisamment puissant pour les développeurs expérimentés.

Python est souvent loué pour sa syntaxe claire et concise, qui permet de réduire le nombre de lignes de code nécessaires pour accomplir une tâche. Cette simplicité améliore non seulement la productivité des développeurs, mais aussi la maintenabilité du code.

Python est un langage multiplate-forme, ce qui signifie que le code écrit en Python peut être exécuté sur divers systèmes d'exploitation tels que Windows, macOS et Linux sans modification significative.

Python bénéficie d'une communauté active et en croissance constante, qui contribue à l'amélioration continue du langage et de ses bibliothèques. Cette communauté offre également un support considérable sous forme de documentation, forums, et didacticiels**. [45]**

Figure4. 1: Logo de langage python [45]

I.1.1 Applications de Python:

Python est utilisé dans une multitude de domaines, notamment :

- **Développement Web :** Frameworks tels que Django et Flask permettent de créer des applications web robustes et scalables. [46]
- **Science des Données :** Python est l'outil de prédilection pour les data scientists, grâce à ses bibliothèques puissantes comme Pandas, NumPy et Matplotlib.[47]
- **Apprentissage Automatique et Intelligence Artificielle :** Avec des bibliothèques comme Scikit-learn, TensorFlow et Keras, Python est largement utilisé pour développer et déployer des modèles de machine learning et deep learning.[47]
- **Automatisation et Scripting :** La simplicité de Python le rend idéal pour écrire des scripts qui automatisent des tâches répétitives.
- **Développement de Jeux :** Bibliothèques comme Pygame permettent le développement de jeux vidéo simples.[47]

I.1.2 Bibliothèque python :

Une bibliothèque Python est un ensemble de modules ou de packages pré-écrits qui fournissent des fonctionnalités spécifiques. Ces bibliothèques sont conçues pour être réutilisées par les développeurs afin d'éviter de réinventer la roue à chaque fois qu'ils ont besoin de fonctionnalités communes. Python a une vaste collection de bibliothèques pour diverses tâches, allant du développement web à l'analyse de données en passant par l'apprentissage automatique et bien plus encore.

Par exemple, la bibliothèque standard de Python offre des fonctionnalités de base comme la manipulation de fichiers, le traitement de chaînes de caractères, les opérations réseau, etc. Ensuite, il existe des bibliothèques tierces comme NumPy pour le calcul numérique, Pandas pour l'analyse de données, TensorFlow pour l'apprentissage automatique, Django pour le développement web, et ainsi de suite.

Les bibliothèques Python sont généralement distribuées via le Python Package Index (PyPI), ce qui permet aux développeurs d'installer et de gérer facilement les dépendances de leurs projet

Voici quelques exemples supplémentaires de bibliothèques Python populaires et leurs domaines d'application :

- **Pandas :** Pandas est une excellente bibliothèque pour importer vos tableaux Excel

(autres formats) dans Python dans le but de tirer des statistiques et de charger votre Data-set dans Sklearn

- **Seaborn :** Seaborn est une bibliothèque permettant de créer des graphiques statistiques en Python. Elle est basée sur Matplotlib, et s'intègre avec les structures Pandas.
- **NumPy :**(Numerical Python) est une bibliothèque open-source de Python qui supporte des tableaux et des matrices multidimensionnels, ainsi qu'une grande collection de fonctions mathématiques de haut niveau pour opérer sur ces tableaux. Elle est largement utilisée pour des calculs numériques, des manipulations de données.
- **Matplotlib :** Une bibliothèque de traçage 2D et 3D qui permet de créer des graphiques et des visualisations de données.
- **Beautiful Soup :** Une bibliothèque pour l'analyse HTML et XML, souvent utilisée pour extraire des données à partir de pages web.

Requests : Une bibliothèque HTTP élégante et simple pour effectuer des requêtes HTTP en Python.

- **Scikit-learn :** Une bibliothèque d'apprentissage automatique très utilisée qui offre une variété d'algorithmes pour la classification, la régression, le clustering, etc.
- **NLTK (Natural Language Toolkit) :** Une bibliothèque pour le traitement du langage naturel en Python, offrant des fonctionnalités telles que la tokenisation, la lemmatisation, la reconnaissance d'entités nommées, etc.
- **Pygame :** Une bibliothèque de développement de jeux en Python, offrant des fonctionnalités pour la création de jeux 2D
- **PyTorch :** Un framework d'apprentissage automatique développé par Facebook, qui offre une grande flexibilité et des performances élevées pour la création de modèles d'apprentissage profond.

Ces bibliothèques, parmi de nombreuses autres, sont largement utilisées dans la communauté Python pour accélérer le développement de logiciels et la résolution de problèmes dans divers domaines. [48]

I.2 Présentation de Google colab :

Google Colab, ou Google Colaboratory, est un service en ligne gratuit offert par Google qui permet aux utilisateurs de rédiger et d'exécuter des scripts Python directement dans leur

navigateur. Lancé en 2017, Google Colab est particulièrement apprécié par la communauté des développeurs et des chercheurs pour sa facilité d'utilisation et ses ressources de calcul puissantes.

Figure4. 2: Logo de google colab [50]

I.2.1 Les caractéristiques de Google colab :

Voici quelques caractéristiques importantes de Google Colab :

- **Environnement de Développement Basé sur Jupyter :** Colab utilise les notebooks Jupyter, qui sont des documents interactifs combinant code, texte, images, et visualisations. Cette interface intuitive permet de documenter et de partager facilement le travail, rendant la collaboration plus efficace**.[49]**

- **Accès Gratuit aux Ressources de Calcul :**Google Colab offre un accès gratuit à des environnements de calcul dotés de CPU, GPU et TPU. Cela permet aux utilisateurs d'exécuter des algorithmes d'apprentissage automatique et des simulations computationnelles sans avoir à investir dans du matériel coûteux.**[49]**

- **Intégration avec Google Drive :** Les notebooks Colab peuvent être directement sauvegardés et ouverts depuis Google Drive, facilitant l'accès et la gestion des fichiers. De plus, les utilisateurs peuvent importer et exporter des notebooks en différents formats, y compris .ipynb et .py. **[51]**

- **Partage et Collaboration en Temps Réel :** Google Colab permet de partager des notebooks avec d'autres utilisateurs, qui peuvent commenter ou éditer en temps réel. Cela facilite grandement la collaboration sur des projets de groupe ou des recherches conjointes.

- **Installation de packages supplémentaires :** Colab permet d'installer des packages Python supplémentaires via pip directement à partir du notebook, ce qui facilite l'utilisation de bibliothèques tierces**. [52]**

I.2.2 Applications de Google Colab :

- **Apprentissage Automatique et Deep Learning :** Grâce à l'accès gratuit aux GPU et TPU, Colab est idéal pour entraîner des modèles de machine learning et deep learning de grande taille.
- **Analyse de Données :** Les notebooks interactifs facilitent l'exploration et la visualisation des données, ce qui est essentiel pour les projets de data science.
- **Recherche et Développement :** Les chercheurs peuvent utiliser Colab pour prototyper rapidement des algorithmes et partager leurs résultats avec la communauté scientifique.
- **Enseignement et Formation :** Les éducateurs peuvent créer des tutoriels interactifs et des exercices de programmation que les étudiants peuvent exécuter directement dans leur navigateur.

En résumé, Google Colab est un outil puissant et accessible qui présente de nombreux avantages. Avec son accès gratuit aux ressources de calcul, son interface conviviale et ses fonctionnalités de collaboration, il constitue une plate-forme idéale pour les projets de recherche et d'analyse de données.

II.Extraits de codes :

Nous avons développé deux modèles de classification, l'un basé sur KNN et l'autre sur SVM, pour prédire ou classifier un paquet réseau en tant que paquet normal ou attaque. Tout d'abord, nous avons importé les bibliothèques nécessaires, notamment numpy, pandas et matplotlib pour les graphiques, comme illustré dans la figure suivante.

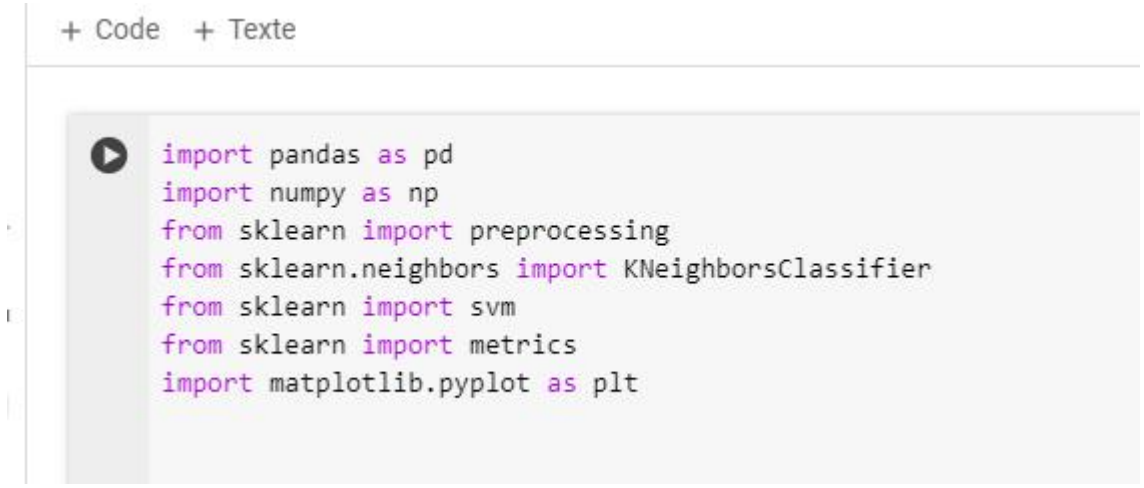

Figure4. 3: importation des librairies python.

II.1 Chargement des données du data-set:

Ensuite Pour commencer, il est nécessaire de lire et de charger les données à partir du fichier texte. Python, via sa bibliothèque Pandas, offre des classes et des fonctions permettant de lire divers formats de fichiers. comme le montre la figure:

```
df_train = pd.read_csv('/content/drive/MyDrive/KDD/NSL_KDD_Train (1).csv')
df_test = pd.read_csv('/content/drive/MyDrive/KDD/NSL_KDD_Test (1).csv')
```

Figure4. 4 : Chargement du Data-set

Nous avons séparé les données en sous bases de données , une sous base pour l'entraînement et une sous base de test, en va prendre que 75% pour l'entraînement 25% pour le test

```
## le dataset contient vers les 160 miles points, 75% entrainement et 25% trtest
X_train = X[:120000]
X_test = X[120000:]
y_train = y[:120000]
y_test = y[120000:]
```

Figure4. 5: Partitionnement du Data-set

II.2 La mise en échelle et la standardisation :

Pour appliquer les algorithmes de classification, il est crucial que les variables prédictives du

modèle soient de la même échelle. Pour cela, nous utiliserons une technique appelée mise à l'échelle des caractéristiques (features scaling).

La bibliothèque Scikit-learn de Python offre plusieurs classes et méthodes pour le pré-traitement des données (data preprocessing) pour les algorithmes de machine learning. Le module sklearn.preprocessing propose la classe StandardScaler, qui permet de normaliser nos variables prédictives.

Une fois les données correctement préparées, nous pouvons commencer à les explorer. Cette étape est essentielle pour mieux comprendre les comportements des données et saisir le phénomène sous-jacent. Il est crucial de ne pas négliger cette phase : les meilleurs data scientists ne sont pas ceux qui maîtrisent les algorithmes les plus complexes, mais ceux qui comprennent parfaitement leurs données et qui ont soigneusement préparé leur terrain en amont.

```
# normal:0, dos:1, r2l:2, probe;3, ur2:4
change_label(df)
y=df['attack']
y = y.astype(int)
```

Figure4. 6: la mise en échelles des attributs.

II.3 Apprentissage et test :

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

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn import svm
```

Figure4. 7 : Importation des Algorithmes.

- **KNN (KNeighborsClassifier) :**

Nous avons utilisé le modèle KNN (K-Nearest Neighbors) pour la classification. Tout d'abord, nous avons initialisé le modèle KNN en utilisant la classe KNeighborsClassifier de la bibliothèque Scikit-learn, en spécifiant le nombre de voisins à considérer (dans cet exemple, n_neighbors=1). Ensuite, nous avons entraîné ce modèle avec les données d'entraînement, comprenant les caractéristiques (X_train) et les étiquettes correspondantes (y_train), en utilisant

la méthode fit. Une fois le modèle entraîné, nous l'avons utilisé pour prédire les étiquettes des données de test (X_test) à l'aide de la méthode predict.

```
knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(X_train, y_train)
y_pred_knn = knn.predict(X_test)
```

Figure4. 8: Algorithme KNN.

- **Machine à vecteur de support (SVM) :**

Tout d'abord, nous avons initialisé le modèle SVM en utilisant la classe SVC de la bibliothèque Scikit-learn. Ensuite, nous avons entraîné ce modèle avec les données d'entraînement, comprenant les caractéristiques (X_train) et les étiquettes correspondantes (y_train). Une fois le modèle entraîné, nous l'avons utilisé pour prédire les étiquettes des données de test (X_test) à l'aide de la méthode predict.

```
svmclassif = svm.SVC()
svmclassif.fit(X_train,y_train)
y_pred_svm = svmclassif.predict(X_test)
```

Figure4. 9: Algorithme SVM.

II.4 Calcul des métriques :

Enfin, pour évaluer les performances du modèle KNN, SVM, nous avons utilisé une fonction de mesure de performance (metric) qui compare les étiquettes prédites aux étiquettes réelles des données de test.

III. Présentation des résultats :

Voici les métriques du rapport de classification (accurcy, recall, précision, et F1 score) ainsi que la matrice de confusion de chaque modèle.

KNN:

les Classes / Les métriques	Normal	Dos	R2L	Probe	Ur2
Récall	0,74%	0,96%	0,50%	0,77%	0,30%
Précision	0,95%	0,85%	0,07%	0,77%	0,33%
F_1- Score	0,83%	0,90%	0,13%	0,77%	0,31%
Accuracy	0,8035069261792039 %				

Tableau 4. 1: Rapport de classification KNN (accurcy, recall, précision, et F1 score).

- **SVM:**

Les Classes / Les Métriques	Normal	Dos	R2L	Probe	Ur2
Récall	0,71%	0,97%	0,95%	0,77%	0,7%
Précision	0,95%	0,84%	0,06%	0,76%	0,10%
F_1- Score	0,81%	0,90%	0,11%	0,77%	0,18%
Accuracy	0,7972295283184289 %				

Tableau 4. 2: **Rapport de classification Svm (accurcy, recall, précision, et F1 score).**

IV. Comparaison et Discutions :

En comparant les résultats de classification KNN et SVM :

La précision globale de KNN est légèrement supérieure à celle de SVM (0.8035 contre 0.7972).

En ce qui concerne le rappel (recall), les performances varient entre les classes, mais en général, les résultats de KNN sont plus élevés pour certaines classes tandis que SVM obtient de meilleurs résultats pour d'autres.

Les scores de précision (precision) montrent également des variations entre les classes, avec des performances parfois meilleures pour Knn et parfois pour SVM.

Les scores F1 (qui combinent précision et rappel) indiquent une performance globalement supérieure de Knn par rapport à SVM.

Une matrice de confusion est un tableau qui compare les valeurs prédites par un modèle (y-prédit) aux valeurs réelles (true labels). En analysant les matrices de confusion du KNN (K-Nearest Neighbors) et du SVM (Support Vector Machine), nous pouvons observer des différences de performance entre les deux modèles.

KNN :

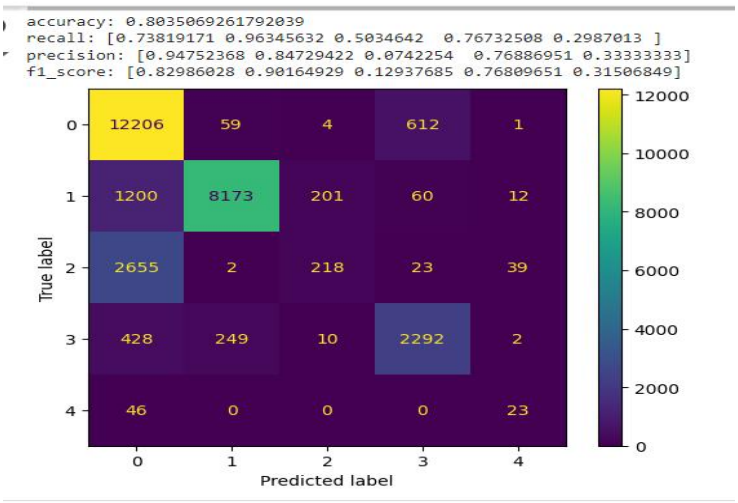

Figure4. 10: matrice de confusion pour KNN.

- **SVM :**

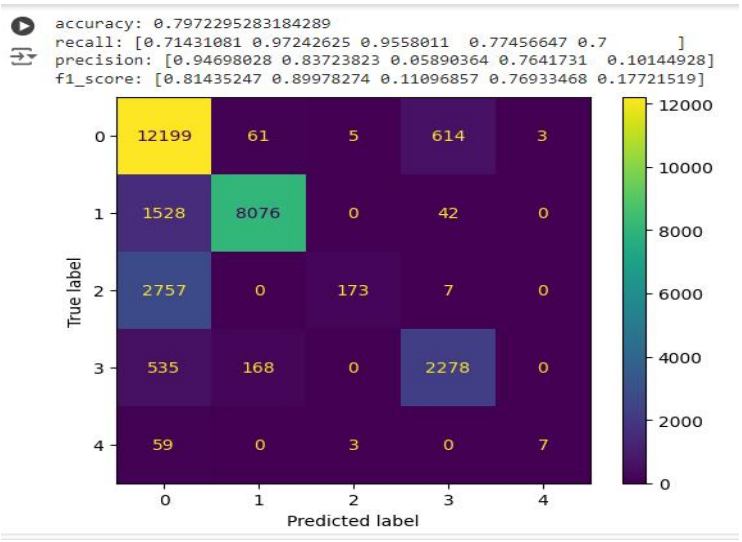

Figure4. 11: matrice de confusion pour SVM.

Les cellules de la matrice de confusion sont colorées pour indiquer le nombre de classifications correctes ou incorrectes : plus la couleur est jaune, plus le nombre de classifications est élevé ; à l'inverse, plus la couleur est bleu foncé, moins il y a de classifications.

Pour la classe 0, qui représente les données normaux, le modèle KNN réalise 12 206 classifications correctes, tandis que le SVM en réalise 12 199. Cela signifie que KNN est légèrement meilleur pour identifier correctement les données normales.

Pour la classe 1, qui correspond aux attaques de type DOS (Denial of Service), le KNN effectue 8 173 classifications correctes contre 8 076 pour le SVM. Ici aussi, KNN montre une meilleure performance.

Cependant, pour la classe 2, qui représente les attaques de type R2L (Remote to Local), le SVM dépasse le KNN avec 2 757 classifications correctes contre 2 655 pour le KNN.

En résumé, bien que le SVM soit plus performant pour la détection des attaques R2L, le KNN montre une performance globale supérieure, étant plus robuste et précis dans la plupart des cas, notamment pour les classes 0 et 1. Cette analyse montre que le choix du modèle peut dépendre du type de classe que l'on cherche à prédire avec le plus de précision.

Conclusion :

En conclusion, le chapitre "Conception et réalisation" s'est concentré sur le développement d'un IDS basé sur KNN et SVM en utilisant l'ensemble de données NLS-KDD. Les principaux sujets abordés comprenaient l'établissement de l'environnement d'exécution, la sélection et la préparation des ensembles de données, la création de modèles KNN et SVM, la formation, l'évaluation et les tests et la comparaison des résultats.

Le chapitre a démontré une efficacité légèrement meilleur du modèle KNN en le comparant avec SVM pour détecter avec précision les intrusions et différencier le trafic réseau normal. Ces résultats fournissent des informations précieuses pour de nouvelles recherches sur la détection des intrusions et contribuent aux progrès de la cybersécurité.

Conclusion Générale

Conclusion Générale

En conclusion, cette recherche sur les systèmes de détection d'intrusion réseau (IDS) a apporté des informations précieuses dans ce domaine. L'étude a couvert les concepts fondamentaux, les principes de l'apprentissage automatique, ainsi que la conception, la mise en œuvre et l'évaluation d'un outil IDS.

Les résultats soulignent l'importance cruciale d'un IDS pour atténuer et répondre aux menaces potentielles dans les environnements réseau. En utilisant des techniques d'apprentissage automatique, un outil IDS robuste a été développé, démontrant des performances fiables dans la détection et l'alerte contre les intrusions.

Dans notre travail, nous avons commencé par la sélection des données, en choisissant de travailler avec un ensemble de données nommé NLS-KDD pour la détection des anomalies. Nous avons décidé de tester deux algorithmes généralement bien adaptés à nos objectifs :

1. K-Nearest Neighbors (KNN)
2. Support Vector Machine (SVM)

Notre objectif était de trouver l'algorithme offrant la meilleure précision possible pour une utilisation pratique. Plusieurs mesures de performance peuvent être utilisées, les plus courantes étant : "Accuracy", "Precision", "Recall", "F1-score" et la matrice de confusion.

Les résultats obtenus montrent que les deux algorithmes donnent des résultats assez proches, mais que le KNN présente globalement une performance supérieure, étant plus robuste et précis dans la plupart des cas. Nous pensons que la précision de ces deux algorithmes pourrait encore être améliorée en personnalisant la configuration du réseau et le taux d'apprentissage. Il serait également pertinent, dans de futurs travaux, d'examiner plus attentivement la configuration de chaque algorithme, dans le but de trouver les meilleurs paramètres permettant d'améliorer la précision tout en réduisant le temps d'exécution.

Bibliographie

BIBLIOGRAPHIE

[1]- https://www. global.com/fr-lu/apprendre/blog/les-objectifs-de-protection de la sécurité l'information-et-leur-signification

[2]- https://www.cyberjobs.fr/actualites-articles/zoom-sur-les-5-objectifs-de-la-securite-informatique

[3]- Jonathan Krier,< **Les systèmes de détection d'intrusions, document réalisé dans le cadre d'un sujet d'initiation >** à la recherche en Master, publié sur le site web Developpez. com, 21 juillet 2006

[4]- Przemysiam Kazienko & Piotr Dorosz « **Intrusion Detection Systems (IDS) Part I – (network intrusion; attack symptoms; IDS tasks; and IDS architecture)** », 2004.

[5]- http://www.dicodunet.com/definitions/reseaux/ip-spoofing.htm

[6]- https://www.cloudflare.com/fr-fr/learning/ddos/glossary/denial-of-service

[7]- https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/

[8]-BOUGHELIT Meriem, MOUSSOUS Somia < ***Développement d'un Système d'Accès Intelligent En Utilisant les Méthodes d'Apprentissage Automatique*** >Mémoire de Master en Informatique **U**NIVERSITÉ **S**AAD **D**AHLAB- **B**LIDA1, 2019, p6

[9]- https://fr.scribd.com/document/649629472/Support-Cours-SERE-1

[10]- https://fr.wikipedia.org/wiki/Logiciel_malveillant

[11]- https://elearn.univ-tlemcen.dz/mod/resource/view.php?id=15171

[12]- https://kinsta.com/fr/blog/qu-est-ce-qu-un-pare-feu/

[13]- https://elearning.centre-univ-mila.dz/a2024/course/view.php?id=338

[14]- https://actualiteinformatique.fr/cryptomonnaie/definition-cryptographie

[15]-CHEFRI Sarra **<*Détection d'intrusions via des réseaux de neurones optimisés par des méta heuristiques*** >Mémoire de Master Recherche Informatique, Université Mohamed Seddik Ben Yahia – Jijel, 2019,p13

[16]- Ni Gao, Ling Gao, Quanli Gao, and Hai Wang. An ***intrusion detection model based on deep belief networks***. In 2014 Second International Conference on Advanced Cloud and Big Data, pages 247–252. IEEE, 2014.

[17]- AMANDA, M., & NSIRI, M. (2011). ***Etude d'un système de détection d'intrusion comportementale pou l'analyse du trafic aéroportuaire*** .Rapport de Projet LENAC.

[18]- S. AGGOUN, S. BELKACEM, ***Mise en oeuvre d'une solution de sécurité basée sur les IDS Cas d'étude : entreprise Cevital,*** Mémoire de Master 02 en Informatique, Université ABDERAHMANE MIRA de Bejaia, 2013.

[19] -K. BELKHATMI et O. BENAMARA, **Mise en place d'un système de détection et de prévention d'intrusion**, Mémoire de Master 02 Université A/Mira de Béjaïa, 2016.

[20]- Lmis Hawarah **une approche probabiliste pour le classement d'objets incomplément connus dans un arbre de décision THESE** Université Joseph Fourier.

[21] Benbrahim Embarka., Amiche Selyna **Mise en place d'une solution de détection d'intrusion**, Mémoire Master académique Réseaux et Télécommunications université Mouloud Mammeri de Tizi-Ouzou ,2017

[22]- Debar, H., Dacier, M., Wespi, A, A Revised **Taxonomy for Intrusion-Detection Systems**, **Annals des Télécommunications**, Vol. 55, No. 7-8, 2000. James Anderson. "**Computer security threat monitoring and surveillance**" . Technical report, James Anderson company, For Washington, Pennsylvania, April 1980.

[23]-SLIMANI Ahmed, **Application des systèmes immunitaires artificiels à la détection d'intrusion** Mémoire Magister, Université Mohamed Boudiaf USTO-MB- d'Oran ,2010

[24]- Lynda SELLAMI, **Approche Data Mining pour la Détection d'Intrusions** Mémoire de Magistère En Informatique, Université Abderrahmane Mira de Bejaia

[25]- Ludovic DE MATTEIS Steeven JANNY - Solal NATHAN – Wenqi SHU-QUARTIER **Introduction à l'apprentissage automatique Édité le** (24/05/2022)

[26]-https://elwatan-dz.com/consequences-du-developpement-de-lintelligence-artificielle-sur-leconomie-mondiale

[27]- Chapitre 1 du cours Intelligence Artificielle présenté par **M. Bougamouza** université 20 Aout de skikda

[28]-https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/dossier-intelligence-artificielle].

[29]- Mokhtar TAFFAR Université de Jijel INITIATIONAL APPRENTISSAGE

AUTOMATIQUE Ya Ouli Al-Albab Université de Jijel Faculté des Sciences Exactes et de l'Informatique Département Informatique Support de Cours pour étudiants en Master en Intelligence Artificielle.

[30]- Arnaud Rosay. **Détection d'intrusions dans les objets connectés par des techniques d'apprentissage automatique :** étude dans les domaines de l'éducation et des voitures connectées. Réseau de neurones [cs.NE]. Le Mans Université, 2022. Français.
ffNNT : 2022LEMA1044ff. fftel-03937132f

[31]https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.datatransitionnumerique.com%2Fmachinelearningpython%2F&psig=AOvVaw2eWhjJ2y_qtZbvtG3jWQ_4&ust=17172283

[32]- Mr. MIMOUNE Zakarya. Développement d'une Architecture Basée sur l'Apprentissage Profond (Deep Learning) pour la Détection d'Intrusion dans les Réseaux (2018/2019).

[33]https://www.google.com/imgres?q=image%20apprentissage%20non%20supervis%C3%A9

[35]- Université Saad Dahlab De BLIDA1 Faculté Des Sciences Département D'informatique. Mémoire réalisé par :**Douid Rania: Système de détection d'intrusion réseau basé sur L'algorithme de Classification KNN: 2018-2019**
https://di.univ-blida.dz/jspui/bitstream/123456789/4069/1/

[36]- BAH DIDI EL MOKHTAR SALEM.**Système de détection d'intrusion avec une approche D'apprentissage automatique** Université Saad Dahlab De BLIDA1
http://di.univ-blida.dz:8080/jspui/handle/123456789/9421

[37]- BOUKERTOUTA Mohammed **Mémoire de Fin d'études MasterDétection des intrusions basée sur l'apprentissage automatique dans les systèmes IdO(Internet des Objets)Juin 2022**
https://dspace.univguelma.dz/jspui/bitstream/123456789/13426/1/BOUKERTOUTA_MOHAMMED%20AMIN_F5.pdf

[38]- Université Saad Dahlab De BLIDA1 Faculté Des Sciences Département D'informatique.Mémoire réalisé par :**Douid Rania: Système de détection d'intrusion réseau basé sur L'algorithme de Classification KNN: 2018-2019**
https://di.univ-blida.dz/jspui/bitstream/123456789/4069/1/

[39]- Tighzer Yanis Mouzaia Raouf Mémoire Fin de cycle master: **Apprentissage automatique pour la détection d'intrusion dans un système informatique:(2020-2021)**

https://www.univbejaia.dz/xmlui/bitstream/handle/123456789/18335/Apprentissage

[40]- « Zahaf Mohamed El-Bachir » RAPPORT DE MINI-PROJET: **Système de Détection d'Intrusion Réseau Basé sur l'Apprentissage Automatique(2022-2023).**

http://ebiblio.univmosta.dz/bitstream/handle/123456789/26139/MINF381.pdf?sequence=1&isAllowed=y

[41]- Melaoui taki eddine.**Mémoire de fin d'études Pour une détection intelligente de télé intrusion:(11/07/2021).**

http://dspace.univtebessa.dz:8080/jspui/bitstream/123456789/927/1/m%C3%A9moire-fin-d%C3%A9tudeMalaoui%20%281%29.pdf

[42]- BAH DIDI EL MOKHTAR SALEM.Système de détection d'intrusion avec une approche D'apprentissage automatique Université Saad Dahlab De BLIDA1

http://di.univ-blida.dz:8080/jspui/handle/123456789/9421

[43]- CHEFRI Sarra <Détection d'intrusions via des réseaux de neurones optimisés par des méta heuristiques > Mémoire de Master Recherche Informatique, Université Mohamed Seddik Ben Yahia – Jijel, 2019,p13

[44]- Hamouda Djallel Mémoire de Fin d'études Master

Filière : Informatique **Un système de détection d'intrusion pour la cybersécurité(Octobre 2020).**

[45]- **https://python.sdv.univ-paris-diderot.fr/01_introduction/**

[46]- **https://www.python.org/about/apps/**

[47]- **https://www.geeksforgeeks.org/python-applications-in-real-world/**

[48]- **https://aws.amazon.com/fr/what-is/python/**

[49]- **https://datascientest.com/google-colab-tout-savoir**

[50]- **https://x.com/GoogleColab/header_photo**

[51]-https://www.analyticsvidhya.com/blog/2020/03/google-colab-machine-learning-deep-learning/

[52]- https://academy.hsoub.com/apps/productivity/