REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHEMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Réseaux et Télécommunications

Par :

Mezili Houcine

Sur le thème

Vers une amélioration de la détection d'intrusion par les méthodes de sélection des fonctionnalités à l'aide des arbres de décision

---

Soutenu publiquement le 10 / 10 / 2021 à Tiaret devant le jury composé de :

Mr Mostefaoui sidahmed mokhtar	Grade	Université MCB	Président
Mr Daoud mohamed amine	Grade	Université MAA	Encadreur
Mr Mokhtari ahmed	Grade	Université MAA	Examinateur

2020-2021

Remerciements

Tout d'abord, nous remercions **ALLAH** qui nous aide et nous donne la patience et le courage durant ces années d'étude.

Nous souhaitons d'adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire.

Ces remerciements vont au corps professoral et administratif de département d'informatique de l'université d'Ibn Khaldoun de Tiaret pour la richesse et la qualité de leurs enseignements.

Ensuite nous tenons à remercier notre encadreur
Mr Daoud Mohamed Amine
pour l'orientation, la confiance, la patience qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené. Qu'il trouve dans ce travail un hommage vivant à sa haute personnalité.

Nous tenons aussi à remercier les membres du jury qui ont accepté d'examiner notre mémoire.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours soutenue et encouragé au cours de la réalisation de ce mémoire.

Merci à tous.

DEDICACE

Je dédie cet humble travail :

À mon cher père et à ma chère mère Que Dieu les protège et leurs offre la chance et le bonheur.
À mes Frères qui je souhaite un avenir radieux plein de réussite
À toute ma famille.
À mes Amis qui me sont chers
Je remercie également tous mes professeurs et surtout mon encadreur Mr Daoud Mohamed Amine
En un mot à tous les gens qui contribué ma réussite de près ou de loin.
Puisse Dieu vous donne santé, bonheur, courage et surtout réussite

(Mezili **houcine &** Ladjal **hocine**)

TABLE DES MATIÈRES

CHAPITRE 01 : LE SYSTÈME DE DÉTECTION D'INTRUSION

CHAPITRE 02 : MACHINE LEARNING ET SELECTION DES CARACTERISTIQUES

Chapitre 03 : L'approche proposée

Chapitre 04 : Implémentations

Liste des figures

Liste des tableaux

Résumé

L'utilisation des systèmes de détection d'intrusions (IDS) est l'un des moyens d'offrir un environnement sécurisé et rassurant pour les Utilisateurs des systèmes informatiques. Des mises à jour et des améliorations de ces systèmes sont recommandées en raison de l'apparition régulière de nouvelles vulnérabilités. Les recherches ont montré le rôle important du Machine Learning (ML) dans la construction et la réalisation de nouvelles techniques plus satisfaisante et qui peuvent prédire les nouvelles attaques plus rapidement et efficacement pour préparer la contre-mesure la mieux adaptée.

Les modèles proposés ont ensuite été évalués à l'aide des ensembles de données CICDDoS2019. Grâce à l'utilisation du Machine learning, des solutions efficaces peuvent être réalisé en vue de renforcer la capacité de détection des systèmes de détection d'intrusions.

يعد استخدام أنظمة كشف التسلل (IDS) إحدى وسائل توفير بيئة آمنة ومطمئنة لمستخدمي أنظمة الكمبيوتر. يوصى بإجراء تحديثات وتحسينات لهذه الأنظمة نظرًا للظهور المنتظم للثغرات الأمنية الجديدة.

أظهرت الأبحاث الدور المهم للتعلم الآلي (ML) في بناء وتنفيذ تقنيات جديدة أكثر إرضاءً ويمكن أن تتنبأ بهجمات جديدة بشكل أسرع وأكثر كفاءة للتحضير للإجراء المضاد الأكثر ملاءمة. ثم تم تقييم النماذج المقترحة باستخدام مجموعات البيانات.CICDDoS2019 .

من خلال استخدام التعلم الآلي، يمكن تحقيق حلول فعالة من أجل تعزيز قدرة الكشف عن أنظمة الكشف عن التسلل.

The use of Intrusion Detection Systems (IDS) is one of the ways to provide a secure and reassuring environment for users of computer systems.

Updates and improvements to these systems are recommended due to the regular appearance of new vulnerabilities.

Research has shown the important role of Machine Learning (ML) in the construction and realization of new techniques that are more satisfactory and can predict new attacks more quickly and efficiently to prepare the most appropriate countermeasure.

The proposed models were then evaluated using the CICDDoS2019 datasets.

Through the use of machine learning, effective solutions can be realized to enhance the detection capability of intrusion detection system.

Introduction générale

Introduction générale

La venue d'Internet, et sa politique démocratique permettant à toute machine d'être connectée au réseau, et à toute personne d'en bénéficier ainsi que de proposer ses propres services.

Le contrôle de l'accès à ces informations et le suivi des opérations effectuées doivent être assurés. Pour améliorer la sécurité des réseaux, les administrateurs disposent de nombreux outils, dont les systèmes de détection d'intrusions.

Au cours des deux dernières décennies, le système de détection automatique des intrusions a été un point d'exploration important. Un système de détection d'intrusion (IDS) permet d'aider à surveiller et à analyser le trafic réseau comme normal ou anormal. D'après de nombreuses solutions possibles, des problèmes existent, [1] En raison de similitudes structurelles dans l'ensemble de données de trafic réseau, l'attaque est considérée comme normale (corrélation élevée) et [2] La précision de base et le taux de détection doivent être améliorés avec des performances efficaces.

La reconnaissance des formes et l'exploration de données sont les techniques qui permettent d'acquérir des informations significatives à partir de données à grande échelle. Ces techniques sont largement utilisées, car il y a une quantité et un type de données en une augmentation constante. Pour l'ensemble de données obtenu, les algorithmes de réduction de données sont nécessaires pour le filtrage, le tri prioritaire et la fourniture de mesures redondantes pour détecter la sélection de caractéristiques. La sélection des fonctionnalités est une étape importante avant la classification et l'analyse des données. Elle est une sorte de problème d'optimisation NP complet. En utilisant des algorithmes tels que les algorithmes génétiques ou Swarm, des données de qualité sont obtenues, ce qui augmente à son tour la qualité des systèmes de détection ou le succès de la reconnaissance.

Les techniques d'apprentissage automatique sont l'ensemble des algorithmes évolutifs qui apprennent avec l'expérience, ont amélioré les performances dans les situations qu'ils ont déjà rencontrées. L'arbre de décision est utilisé comme classificateur dans le but de minimiser son erreur de classification en utilisant un sous-ensemble de l'ensemble de données de détection d'intrusion.

Objectif :

Le but de ce travail est détecter et de réduire l'erreur des systèmes de détection d'intrusion, aussi, de tester l'effet de l'élimination des caractéristiques sans importance et obsolètes des ensembles de données sur le succès de la classification, en utilisant le classifieur arbres de décision. L'implémentation et le développement de l'approche est utilisée dans la classification des attaques. Dans l'ensemble, les mesures de performance tel que le Recall, nous permet de prendre des bonnes décisions concernant le taux de détection.

La structure du mémoire :

Afin de répondre à cet objectif, ce mémoire est structuré de la façon suivante :

- Le premier chapitre est une présentation de la notion des systèmes de détection d'intrusions.
- Le deuxième chapitre présente les sélections de fonctionnalités et techniques d'apprentissage automatique.
- Le troisième chapitre est pour présenter l'approche proposée.
- Le quatrième chapitre est pour faire des implémentations et des tests concernant l'approche proposée.

PARTIE 1

Recherche bibliographique (État de l'art)

Chapitre 01

Le système de détection d'intrusion

1. Introduction

Les technologies Internet ont été évoluées et révolutionné nos modes de vie et de travail. Cette évolution affecte toutes les organisations, grandes et petites. Les vagues de changements technologiques sont fréquentes et s'accélèrent, demandant une adaptation constante des entreprises et de leurs employés.

Les systèmes et réseaux informatiques contiennent diverses formes de vulnérabilité, donc la sécurité est, de nos jours devenue un problème majeur dans la gestion des réseaux d'entreprise ainsi que pour les particuliers toujours plus nombreux à se connecter à Internet.

Les pirates informatiques développent rapidement de nouvelles techniques, plus complexes, pour mener une cyberattaque. La nécessité de montrer une meilleure résilience et d'avoir des solutions de protection puissantes est donc essentielle. Les intrusions sont causées par des attaquants qui accèdent aux systèmes par Internet, qui sont des utilisateurs autorisés par le système et qui tentent de gagner plus de privilèges pour lesquels ils ne sont pas autorisés, ou des utilisateurs autorisés qui abusent des privilèges qui leurs sont attribués.

Pour faire face à ces problèmes de sécurité informatique, différents mécanismes ont été mis en place pour prévenir toute sort d'attaque comme les pares-feux, antivirus, qui s'avèrent limités face au développement rapide des techniques de piratage, d'où la nécessité de mettre en place un système de détection d'intrusion.

Le système de détection d'intrusion, est une technique permettant de détecter les intrusions de les prévenir, ce système nous aide à prévoir, surveiller ou à identifier toute activité non autorisée dans un réseau. Il est essentiel pour une infrastructure informatique sécurisée de manière optimale. Ces mesures vous aideront à identifier les failles de sécurité et à les prévenir.

Les IDS ont obtenu l'acceptation comme un complément nécessaire à l'infrastructure de sécurité de chaque organisation. Même si, la technologie de détection d'intrusion ne puisse pas offrir une protection complète contre les attaques, ils améliorent l'approche de défense en profondeur, qui est la tendance à la mode de la sécurité réseau.

Ces systèmes sont devenus pratiquement nécessaires en raison de l'augmentation continue du nombre et du risque d'attaques réseau ces dernières années.

2. Définition IDS

Le premier modèle de détection d'intrusion est développé en 1984 par Dorothy Denning et Peter Neuman, qui s'appuie sur des règles d'approche comportementale. Ce système appelé IDES

IDS est un ensemble de composants logiciels et/ou matériels destiné à repérer des activités anormales ou suspectes sur la cible analysée, un réseau ou un hôte, son rôle est de surveiller les données qui transitent sur ce système. Il permet ainsi d'avoir une action d'intervention sur les risques d'intrusion. Afin de détecter les attaques que peut subir un système ou réseau informatique [4] .

Le système de détection d'intrusion est un mécanisme conçu ou des systèmes logiciels capables de détecter les activités anormales ou suspectes et de surveiller les événements qui se produisent dans un réseau ou sur la cible analysée (réseau ou hôte). Pour être en mesure d'informer l'administrateur système de toute trace d'activité anormale sur ce dernier, l'administrateur décidera s'il faut interdire l'activité.

Les systèmes d'IDS comparent l'activité réseau en cours avec une base de données d'attaques connues afin de détecter divers types de comportements tels que les violations de la politique de sécurité, les malwares et les scanners de port.

Pourquoi utiliser IDS ? :

- Pour fournir les informations possibles sur les intrusions et les tentatives qui ont eu place, permettant l'amélioration du diagnostic, la récupération, et la correction des facteurs causals.
- Pour agir et contrôler la qualité de la sécurité et l'administration, en particulier dans les grandes entreprises.
- Pour déceler les objectifs des attaques.
- Pour éviter les problèmes, on augmente la protection des risques qui sont découverts, et infliger une punition pour ceux qui voudraient attaquer ou autrement abuser du système.
- Pour détecter les attaques et les violations de sécurité qui ne sont pas prévus par d'autres mesures de sécurité [5].

3. Architecture des IDS

Nous décrivons les composants qui constituent classiquement un système de détection d'intrusion. La Figure ci-dessous illustre les interactions entre ces composants [6].

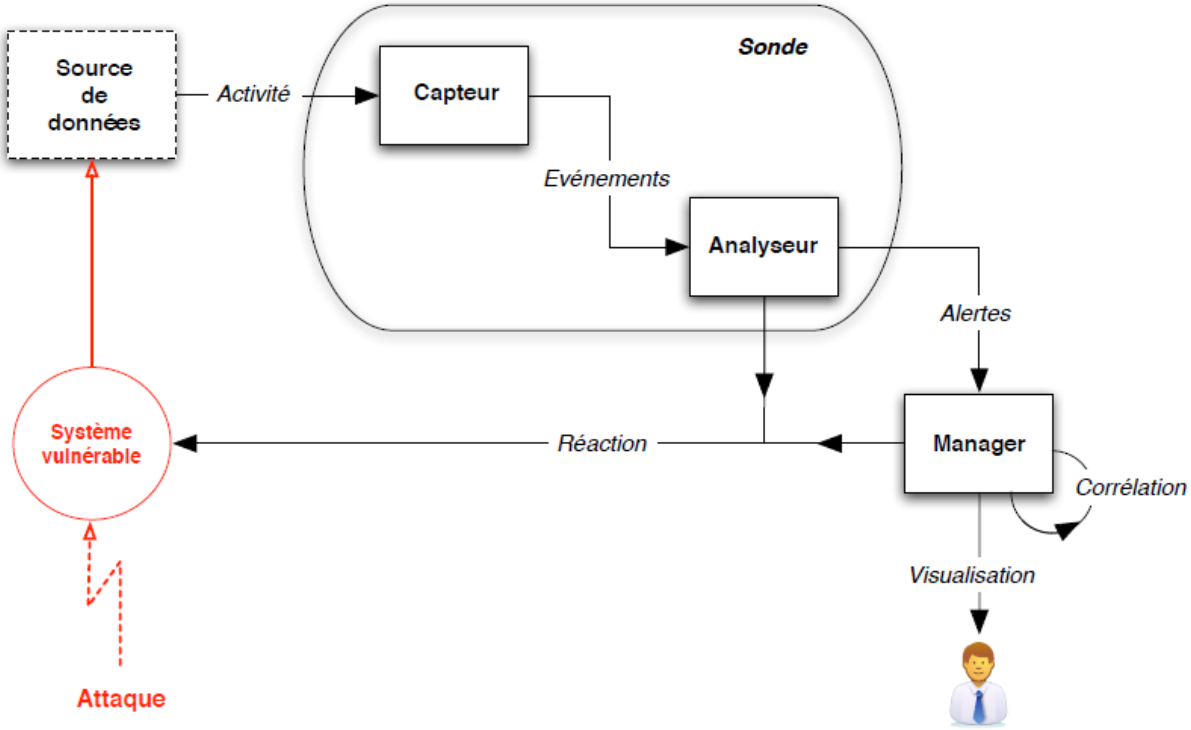

Figure 1 : architecture fonctionnelle d'un IDS [6]

- **Capteur :**

Le capteur observe l'activité du système par le biais d'une source de donnée et fournit à l'analyseur une séquence d'événements qui renseignent de l'évolution de l'état du système.

Le capteur peut se contenter de transmettre directement ces données brutes, mais en général un prétraitement est effectué. Pour cela, on distingue trois types de capteurs en fonction des sources de données utilisées pour observer l'activité du système : les capteurs système, les capteurs réseau et les capteurs 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 :**

Le manager collecte les alertes produites par le capteur, les met en forme et les présente à l'opérateur. Eventuellement, le manager est chargé de la réaction à adopter qui peut être :

- Isolement de l'attaque, qui a pour but de limiter les effets de l'attaque.
- Suppression d'attaque, qui tente d'arrêter l'attaque.
- Recouvrement, qui est l'étape de restauration du système dans un état sain.

Diagnostic, qui est la phase d'identification du problème.

4. Classification des systèmes de détection d'intrusion

Les différents systèmes de détection d'intrusion disponibles peuvent être classés selon plusieurs critères qui sont [7]:

- Emplacement.
- La méthode de détection.
- Le types de réponse.
- La fréquence d'utilisation.

La figure ci-dessous illustre les détails de chaque critère.

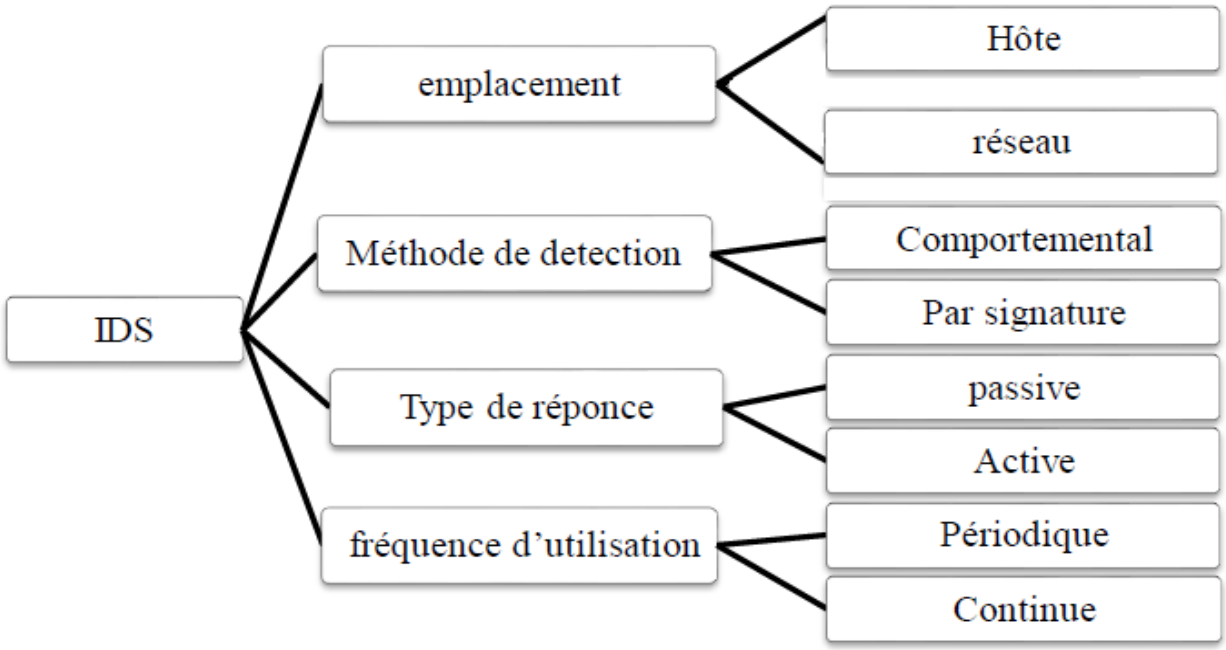

Figure 2 : classification d'un système de détection d'intrusion

4.1 L'emplacement d'IDS

La façon la plus commune pour classifier les IDS est de les regrouper par emplacement de l'information source où ils opèrent. Les sources des informations primaires sont : les paquets réseau capturés à partir des réseaux ou des segments de réseau local ou les systèmes d'exploitation et les fichiers critiques.

4.1.1 Types de détection intrusion

Depuis [8] [9], plusieurs techniques pour la détection des intrusions ont été étudiées. Plusieurs nouvelles méthodes de détection ont été mises en place. Plusieurs efforts de recherche ont été lancés et des résultats efficaces ont été obtenus.

Les systèmes de détection d'intrusion peuvent être classifiés selon leurs sources de données. Les classifications les plus communes des IDSs sont : IDS basés sur le réseau (NIDS) et IDS basés sur les hôtes (HIDS) :

- Les N-IDS (*Network Based Intrusion Détection System*), ils assurent la sécurité au niveau du réseau.
- Les H-IDS (*Host Based Intrusion Détection System*), ils assurent la sécurité au niveau des hôtes [10].

4.1.1.1 La détection d'intrusion basée sur l'hôte (HIDS)

Les systèmes de détection d'intrusion basés sur l'hôte (HIDS) sont des systèmes qui se trouvent aux points de terminaison de service plutôt que dans les points de transit du réseau tels que NIDS. Le premier type d'IDS largement implémenté, Host IDS, est installé sur les serveurs et se concentre davantage sur l'analyse des système d'exploitation et fonctionnalité d'application résidant sur l'hôte HIDS. Les HIDS sont souvent essentiels pour détecter les attaques internes dirigées contre les serveurs d'une organisation tels que les serveurs DNS, de messagerie et Web. HIDS peut détecter une variété de situations d'attaque potentielles telles que les changements d'autorisation de fichier et les demandes client-serveur mal formées. Vérificateurs d'intégrité des fichiers et de fichiers journaux Les agents de vérification de l'intégrité des fichiers et des fichiers journaux sont une forme de HIDS qui se concentre sur les fichiers binaires du système d'exploitation et les fichiers journaux normalement produits par des mécanismes de sécurité basés sur le système d'exploitation, tels que les journaux de connexion.

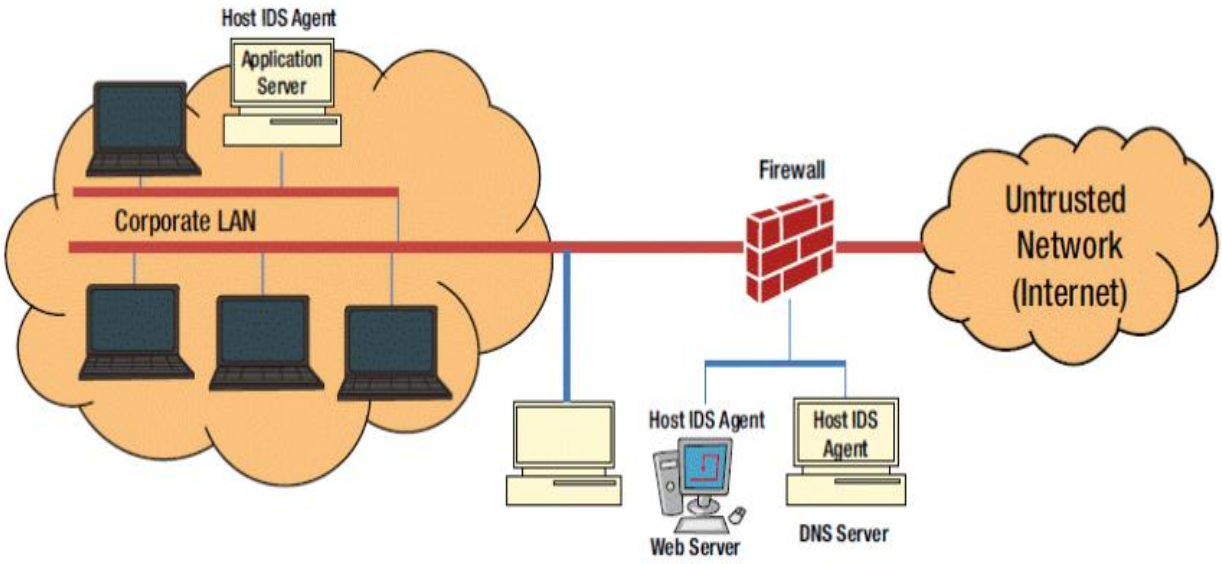

Figure 3 : Système de détection d'intrusion basé sur l'hôte [11]

4.1.1.2 La détection d'Intrusion basée sur l'réseau (NIDS)

Les systèmes de détection d'intrusion basés sur le réseau (NIDS) sont des appareils intelligemment distribués au sein de réseaux qui inspectent passivement le trafic traversant les appareils sur lesquels ils se trouvent. Les NIDS peuvent être des systèmes matériels ou logiciels. Souvent, les NIDS ont deux interfaces réseau. L'un est utilisé pour écouter les conversations réseau en mode promiscuité et l'autre est utilisé pour le contrôle et la création de rapports. Avec l'avènement de la commutation, qui isole les conversations unicast vers les ports de commutation d'entrée et de sortie, les fournisseurs d'infrastructures réseau ont mis au point des techniques de mise en miroir de ports pour répliquer tout le trafic réseau vers le NIDS. Les NIDS sont des systèmes basés sur des signatures ou des anomalies. Les deux sont des mécanismes qui séparent le trafic bénin de ses frères malveillants. Les problèmes potentiels avec NIDS incluent la surcharge de données du réseau à haut débit, les difficultés de réglage, le chiffrement et le retard de développement de la signature.

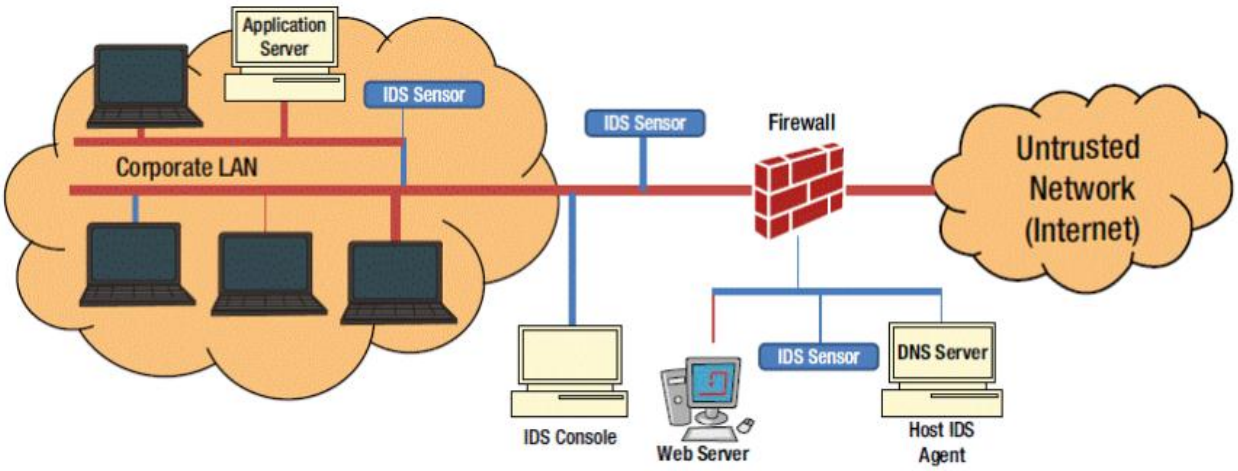

Figure 4 : Détection et système d'intrusion basés sur le réseau [11]

- **Avantages et Inconvénients de (NIDS/HIDS)**

Dans [12] un tableau récapitulatif, mettant en avant les avantages, désavantages de chacun de ces IDS. Ils discutent également dans ce tableau de la responsabilité du déploiement et de mise à jour de ces systèmes.

Type d'IDS	NIDS	HIDS
Avantages	- Détecte-les instruisons en surveillant le trafic réseau - Besoin d'être placé sur le réseau (physiquement) - Les captures peuvent être bien - Assurer la sécurité contre les attaques puisqu'il est invisible.	- Détecte les intrusions en surveillant les fichiers, appels système ou événements réseau de l'hôte - Pas besoin d'équipement en plus - Détecter des attaques qui inaccessibles à détecter avec des IDS réseau puisque le trafic est souvent crypté.
Inconvénients	- Difficile de détecter des instruisons provenant de contenu chiffré - Ne peut pas détecter les attaques ne transitant pas par le NIDS	- Besoin de l'installer sur chaque machine - Ils ont moins de facilité à détecter les scans. - Détection d'attaques locales uniquement
Placement	Réseau physique ou virtuel	Machine virtuelle ou physique
Dépilement & Responsabilité	Administrateur	Utilisateur & administrateur

Tableau 1 : la comparaison entre NIIDS et HIDS

4.2 Les méthodes de détection

Ils utilisent des techniques de détection basées sur des signatures numériques ou de détection d'anomalie.

4.2.1 Approche par scénario ou par signature

La forme la plus répandue de détection d'intrusion est la correspondance de signature. Appelés IDS basés sur les signatures, ces systèmes surveillent le réseau ou le serveur et comparent les attributs de trafic de paquets à un ensemble de listes d'attaques ou de signatures prédéterminées. Si une conversation réseau particulière correspond à une signature configurée sur l'IDS, le système alerte les administrateurs ou entreprend une autre action préconfigurée.

Ce mécanisme protège contre les menaces connues. Une signature est un modèle connu de menace, tel que :

- Un e-mail avec une pièce jointe contenant un malware connu avec un sujet intéressant.
- Une « connexion à distance » par un utilisateur administrateur, qui est une violation manifeste de la politique d'une organisation.

Ils ne peuvent détecter que les menaces connues et ne sont donc pas efficaces pour détecter les menaces inconnues. Pour détecter une attaque, la correspondance de signature doit être

précise, sinon, même si l'attaque présente une petite variation par rapport à la signature de menace connue, le système ne sera pas en mesure de détecter. Par conséquent, il est très facile pour les attaquants de compromettre et de pénétrer dans le réseau de confiance. [11]

Les IDS basés sur les signatures peuvent être assez efficaces dans la surveillance de la sécurité, mais ils présentent plusieurs inconvénients. Pour détecter la plupart des attaques potentielles, la base de données des signatures sur l'IDS doit être volumineuse. À mesure que la vitesse des réseaux augmente, il est difficile pour les IDS basés sur les signatures suivre le rythme du trafic réseau. En règle générale, les IDS basés sur les signatures doivent être désaccordés en supprimant certaines des signatures de la base de données active avant utilisation. Bien que cela permette à l'IDS de fonctionner correctement, il le fait au risque de manquer des attaques potentielles. De même, comme ces IDS alertent uniquement les administrateurs des attaques potentielles pour lesquelles il a une signature, les nouvelles vulnérabilités et exploits ne seront pas détectés tant que les fournisseurs ou les administrateurs n'auront pas développé de nouvelles signatures.

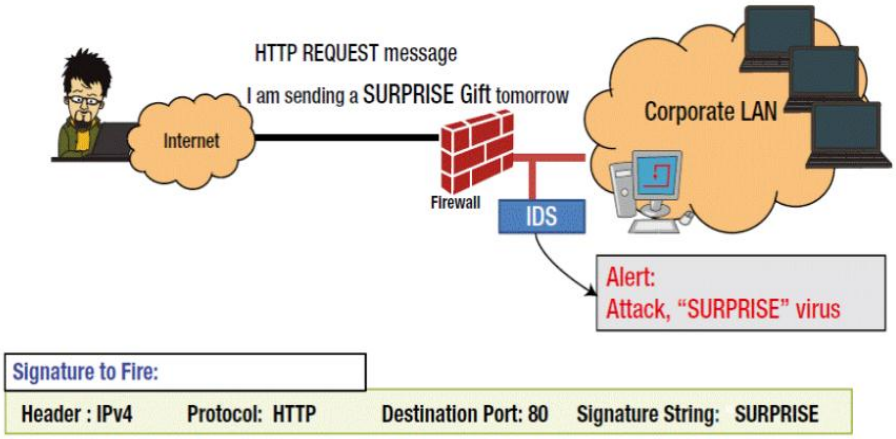

Figure 5 : illustration de l'approche signature

4.2.2 L'approche comportementale

La détection par comportement consiste à considérer comme hostile tout ce qui n'est pas normal, au sens où on cherchera plutôt à bien définir ce qui est un comportement normal sur le système pour pouvoir y opposer toute déviation, que l'on considérera comme étant une attaque : « si ce n'est pas normal, alors c'est dangereux ».

Cette technique consiste à détecter une intrusion en fonction du comportement de l'utilisateur ou d'une application, autrement dit c'est créer un modèle basé sur le comportement habituel du système et surveiller toute déviation de ce comportement.

Plusieurs paramètres sont possibles : la charge CPU, le volume de données échangées, la durée et l'heure de connexion sur des ressources, la répartition statistique des protocoles et applications utilisés...etc.

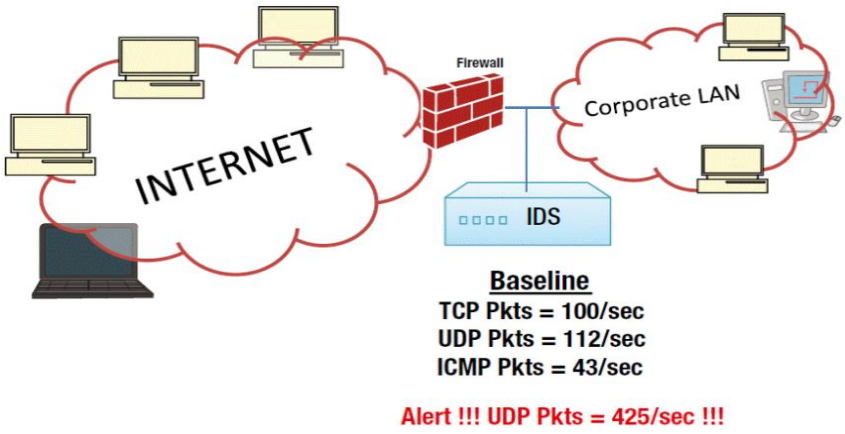

Figure 6 : illustration de l'approche comportementale

- **Avantages et Inconvénients des techniques**

Les techniques de détection	Les avantages	Les inconvénients
Par signatures	• Très efficace pour détecter des attaques sans produire un grand nombre de fausses alarmes. • Peut rapidement et sûrement diagnostiquer l'utilisation d'un outil spécifique ou une technique d'attaque. • Ceci peut aider les responsables de sécurité à donner la priorité aux mesures correctives.	• Le système de détection doit être constamment mis à jour avec les signatures des nouvelles attaques. • Un nombre limité de signatures qui peuvent être définis, ce qui les empêchent de détecter des variantes de ces attaques.
Par Anomalies	• Détectent le comportement peu commun, et ils ont ainsi la capacité de détecter des symptômes des attaques connues et inconnues sans la connaissance spécifique des détails. • Produire l'information utile pour la définition des signatures pour les systèmes de détection d'intrusion à base de signatures.	• Le grand nombre de fausses alarmes dues aux comportements imprévisibles des utilisateurs du réseau. • Les systèmes basés sur cette approche doivent être dotés d'une certaine intelligence pour raison d'apprentissage automatique.

Tableau 2 : les avantages et les inconvénients des techniques de détection

4.3 Les types de réponse

Lors de la détection d'une attaque, un système de détection d'intrusions, peut adopter plusieurs comportements. Il existe deux types de réponses, suivant les IDS utilisés. La réponse passive est disponible pour tous les IDS, la réponse active est plus ou moins implémentée. Une liste des réponses actives et passives est présentée dans le tableau 3 :

4.3.1 Réponse active

La réponse active au contraire a pour but de stopper une attaque au moment de sa détection. Pour cela on dispose de deux techniques : la reconfiguration du firewall et l'interruption d'une connexion TCP. Par exemple un IDS peut réagir en reparamétrant un par- feu, pour mettre en place des règles de blocage temporaire de certains flux réseau anormaux, comme ils peuvent se restreindre à des réponses passives en diffusant un alerte identifiant l'attaque détectée. [13]
La reconfiguration du firewall permet de bloquer le trafic malveillant au niveau du firewall, en fermant le port utilisé ou en interdisant l'adresse de l'attaquant. Cette fonctionnalité dépend du modèle de firewall utilisé, tous les modèles ne permettant pas la reconfiguration par un IDS. De plus, cette reconfiguration ne peut se faire qu'en fonction des capacités du firewall.

4.3.2 Réponse passive

La réponse passive d'un IDS consiste à enregistrer les intrusions détectées dans un fichier de log qui sera analysé par le responsable de sécurité. Certains IDS permettent d'enregistrer l'ensemble d'une connexion identifiée comme malveillante. Ceci permet de remédier aux failles de sécurité pour empêcher les attaques enregistrées de se reproduire, mais elle n'empêche pas directement une attaque de se produire.

Réponse passive	Réponse active
-Emmètre un rapport -Générer une alarme -Activer un archivage plus détaillé -Activer un archivage à distance -Créer des fichiers de sauvegarde	-Bloquer le compte d'un utilisateur -Suspendre des processus malveillants -Terminer une session -Bloquer une adresse IP -Déconnecter la machine du réseau -Mettre hors service les ports et les services attaquées -Tracer l'origine de la connexion -Forcer une nouvelle authentification -Restreindre les activités d'un utilisateur

Tableau 3 : Réponses aux attaques des systèmes de détection d'intrusion [13]

4.4 Fréquence d'utilisation

Une autre caractéristique des systèmes de détection d'intrusions, c'est leur fréquence d'utilisation: continue (online) ou périodique (offline). [5]

4.4.1 Utilisation continue

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

4.4.2 Utilisation périodique

Certains systèmes de détection d'intrusions, analysent périodiquement les fichiers d'audit à la recherche d'une éventuelle intrusion ou anomalie passée et on ne voit que le résultat d'attaque, Cela peut être suffisant dans des contextes peu sensibles, par exemple du fait qu'un HIDS analyse des fichiers traces transmis seulement toutes les heures. Elle est préférable pour avoir une défense plus fiable du point de vue du temps de calcul que pour la première utilisation. Contrairement à l'IDS online, l'attaque ne peut pas être détectée le plus tôt possible pour l'éviter. Plus une attaque est détectée tardivement, les dommages sont importants.

5. Mesures d'évaluations (performances) des systèmes de détection d'intrusions

La matrice de confusion est utilisée pour visualiser, pour chaque classe de modèle, les vraies classifications et les classifications prédites.

On classe les résultats en 4 catégories :

- **True Positive (TP) :** la prédiction et la valeur réelle sont positives.

Exemple : Une personne malade et prévu malade.

- **True Negative (TN) :** la prédiction et la valeur réelle sont négatives.

Exemple : Une personne saine et prévu saine.

- **False Positive (FP) :** la prédiction est positive alors que la valeur réelle est négative.

Exemple : Une personne saine et prévu malade.

- **False Negative (FN) :** la prédiction est négative alors que la valeur réelle est négative.

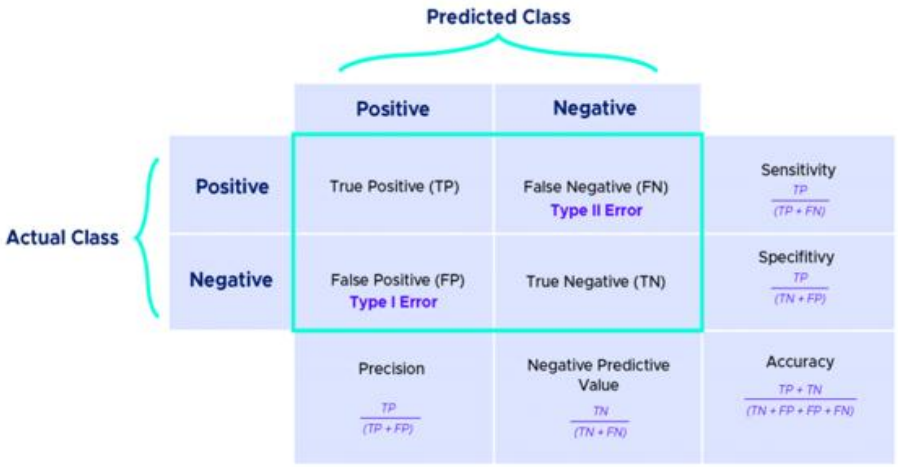

Tableau 4 : Matrice De Confusion [14]

6. Critères de choix d'un IDS

Les systèmes de détection d'intrusion sont devenus indispensables lors de la mise en place d'une infrastructure de sécurité opérationnelle. Ils s'intègrent donc toujours dans un contexte et dans une architecture imposante des contraintes très diverses. Certains critères imposant le choix d'un IDS peuvent être dégagés :

- **Fiabilité** : Les alertes générées doivent être justifiées et aucune intrusion ne doit pouvoir lui échapper.
- **Réactivité** : Un IDS doit être capable de détecter les nouveaux types d'attaques le plus rapidement possible ; pour cela il doit rester constamment à jour. Des capacités de mise à jour automatique sont indispensables.
- **Performance** : la mise en place d'un IDS ne doit en aucun cas affecter les performances des systèmes surveillés. De plus, il faut toujours avoir la certitude que l'IDS a la capacité de traiter toute l'information à sa disposition (par exemple un IDS réseau doit être capable de traiter l'ensemble du flux pouvant se présenter à un instant donné sans jamais supprimer de paquets) car dans le cas contraire il devient trivial de masquer les attaques en augmentant la quantité d'information.

7. Conclusion

Les IDS reposent sur un ensemble de mécanismes et d'algorithmes permettant de détecter, de manière optimale, des intrusions ou menaces dans un réseau informatique.

La détection d'intrusion fait appel à plusieurs champs de recherche tels que la reconnaissance des formes, l'apprentissage automatique, la classification, etc. pour but d'améliorer les capacités de contrôle et de protections des IDS et de trouver de nouvelles solutions de détection, de filtrage ou de réaction après alerte.

Dans le chapitre qui suit nous présenterons les différentes techniques de l'‘apprentissage automatique.

Chapitre 02

Machine Learning Et La sélection des caractéristiques

Chapitre 02 : Machine learning et la sélection des caractéristiques

1. Introduction générale

Dans le contexte de la détection d'intrusion, le terme "classification" est souvent utilisé pour exprimer une "distinction" ou "identification" des attaques. Dans le but de faciliter l'analyse et l'identification, certains ont proposé de répertorier et classifier les attaques pour aider à gérer les incidents et pour traiter les informations d'audit [15] Néanmoins, ce même argument qui a permis un déploiement massif des IDS, pose de sérieux problèmes pour les évaluateurs de tels systèmes.

Pour bien traiter ce problème, il nous semble nécessaire de commencer par une analyse approfondie de l'algorithme de classification choisis, et de mettre ainsi en évidence les détails de son application au domaine de détection de l'intrusion.

L'apprentissage automatique en anglais machine learning (ML) est une forme d'intelligence artificielle (IA) qui permet au système d'apprendre à partir de données plutôt que par une programmation explicite. Au quotidien, machine learning est particulièrement présente et est appliquée dans différents domaines. Plusieurs cas nous permettent d'illustrer l'utilité et les effets de cette technologie dans notre vie quotidienne, cette technologie apparaît comme une composante essentielle pour les entreprises qui souhaitent améliorer la connaissance de leurs clients et ainsi répondre aux besoins des clients. L'objectif du machine learning est donc d'entraîner un algorithme pour traiter de manière pertinente et efficace les différentes données à disposition. Le machine learning se divise en deux principales phases : apprentissage et prédiction.

Dans ce chapitre, nous avons présenté l'apprentissage automatique et ses types, ainsi que les principales séries d'algorithmes pour chaque type.

2. Définition des concepts

2.1 Intelligence artificielle

L'intelligence Artificielle se définirait comme étant « l'ensemble de théories et de techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence. »

Ce serait, des ordinateurs ou des machines dotées de programmes capables de performances similaires à l'intelligence humaine, ou même, amplifiées par la technologie.

Ces machines sont en mesure de :

- Raisonner
- Traiter de grandes quantités de données

- Discerner des modèles indétectables par l'œil d'un humain
- Comprendre et analyser ces modèles
- Interagir avec l'Homme
- Apprendre progressivement
- Améliorer continuellement ses performances

Selon Harry Shum, Président Exécutif de Microsoft, l'IA fonctionne seulement s'il y a présence « d'une vaste quantité de data ; d'une puissance informatique extraordinaire, notamment grâce au cloud ; et des algorithmes révolutionnaires, basés sur le deep learning ». [16]

Les utilisations de l'IA aujourd'hui peuvent être regroupées en 3 catégories principales : l'identification, la prédiction et la génération de données.

Parmi ces applications de l'IA, on retrouve notamment les notions de machine learning, ainsi que celles de deep learning [17].

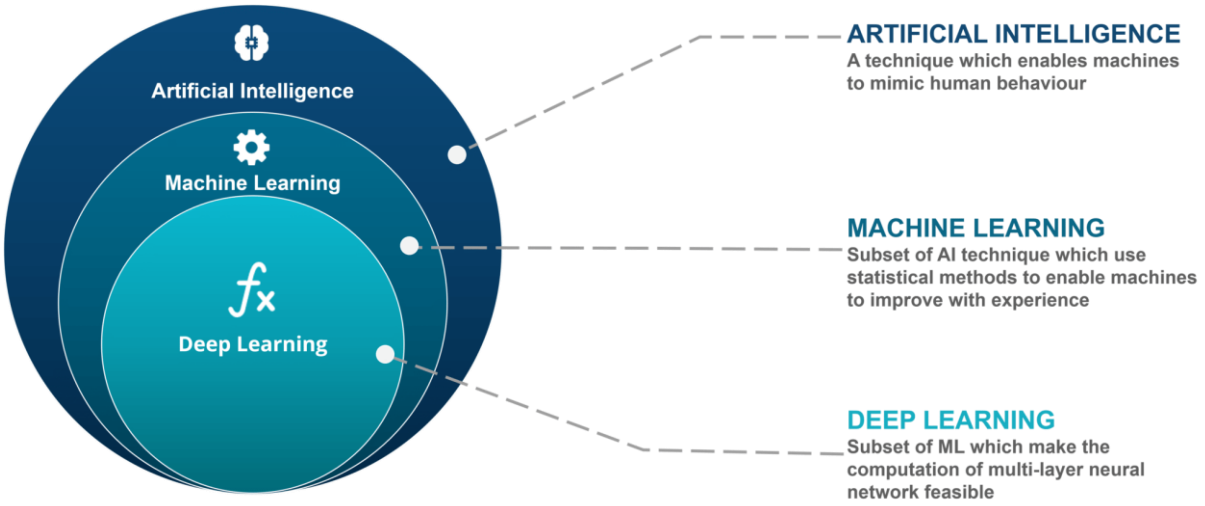

Figure 7 : Représentation des différents types d'intelligence artificielle

2.2 Machine Learning

L'apprentissage automatique n'est pas une nouvelle technologie. Le premier réseau de neurones artificiels, appelé « perceptron », a été inventé en 1958 par le psychologue américain Frank Rosenblatt [18]. L'apprentissage automatique est une technologie d'intelligence artificielle qui permet aux ordinateurs d'apprendre sans programmation explicite.

Cependant, pour apprendre et se développer, les ordinateurs ont besoin de données pour l'analyse et la formation. L'apprentissage automatique comprend une variété de méthodes pour créer automatiquement des modèles à partir de données. Ces méthodes sont en fait des

algorithmes. Son but est de permettre aux machines ou aux ordinateurs de fournir des solutions à des problèmes complexes en traitant de grandes quantités d'informations. Cela offre ainsi une possibilité d'analyser et de mettre en évidence les corrélations qui existent entre deux ou plusieurs situations données, et de prédire leurs différentes implications [19] .

2.3 Deep Learning

L'apprentissage profond en anglais Deep Learning est basé sur les mêmes principes d'apprentissage que l'apprentissage automatique, mais la densité de neurones analytiques est beaucoup plus élevée. Cela aide à éliminer la couche de perception. Dans ce cas, le réseau de neurones lui-même reconnaîtra les caractéristiques discriminantes du problème. C'est pourquoi l'apprentissage en profondeur est souvent appelé « réseau de neurones profonds ». En référence aux couches que possèdent ces réseaux de neurones, cela s'appelle "profondeur".

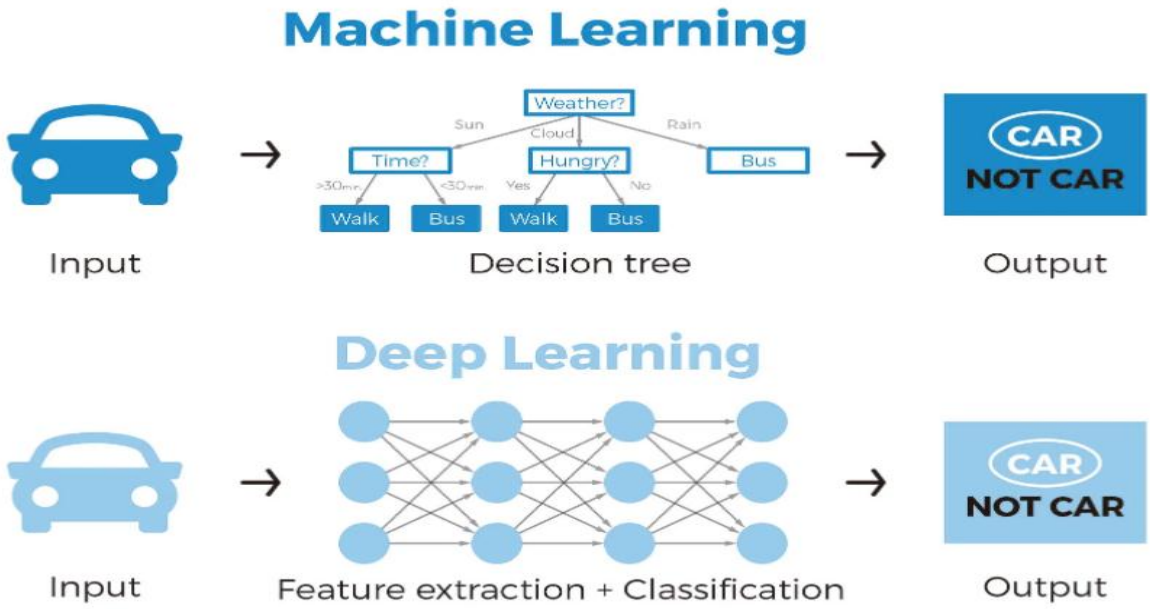

Figure 8 : L'apprentissage automatique vs L'apprentissage profond [20].

Grâce à l'apprentissage profond, l'IA a un bel avenir devant elle. En effet, l'apprentissage profond a permis de nombreuses applications pratiques de l'apprentissage automatique et, par extension, dans le domaine général de l'IA. L'apprentissage profond décompose les tâches d'une manière qui rend possible toutes sortes d'assistance à la machine. Des voitures sans conducteur, de meilleurs soins de santé préventifs, voire de meilleures recommandations de films, sont tous présents aujourd'hui ou à l'horizon. L'IA est le présent et l'avenir. Avec l'aide de l'apprentissage profond, l'IA pourrait même atteindre cet état de science-fiction que nous avons si longtemps imaginé [21]. Des algorithmes de réseaux de neurones sont apparus

récemment, et s'inspirent de la structure biologique de notre cerveau et des interconnexions entre les neurones. Cependant, contrairement aux cerveaux biologiques où n'importe quel neurone peut être combiné avec n'importe quel autre neurone, ces réseaux neuronaux sont liés les uns aux autres dans une direction spécifique de propagation des données.

3. Machine Learning

L'apprentissage automatique est un sous-domaine de l'intelligence artificielle (IA). En général, l'objectif de l'apprentissage automatique est de comprendre la structure des données et de les intégrer dans des modèles qui peuvent être compris et utilisés par les tout le monde.

3.1 Les types du Machine Learning

Dans le domaine du Machine Learning il existe deux principaux types d'apprentissages : supervisées et non supervisées. La principale différence entre ces deux types est que l'apprentissage supervisé se fait sur la base d'un fait. En d'autres termes, nous avons une connaissance préalable de ce que devrait être la valeur de sortie de l'échantillon.

L'objectif de l'apprentissage supervisé est d'apprendre une fonction à partir d'échantillons de données et de résultats attendus qui se rapproche le plus de la relation entre l'entrée et la sortie observables dans les données. En revanche, l'apprentissage non supervisé n'a pas de résultats étiquetés. Son objectif est de dériver la structure naturelle qui existe dans un ensemble de points de données [22] .

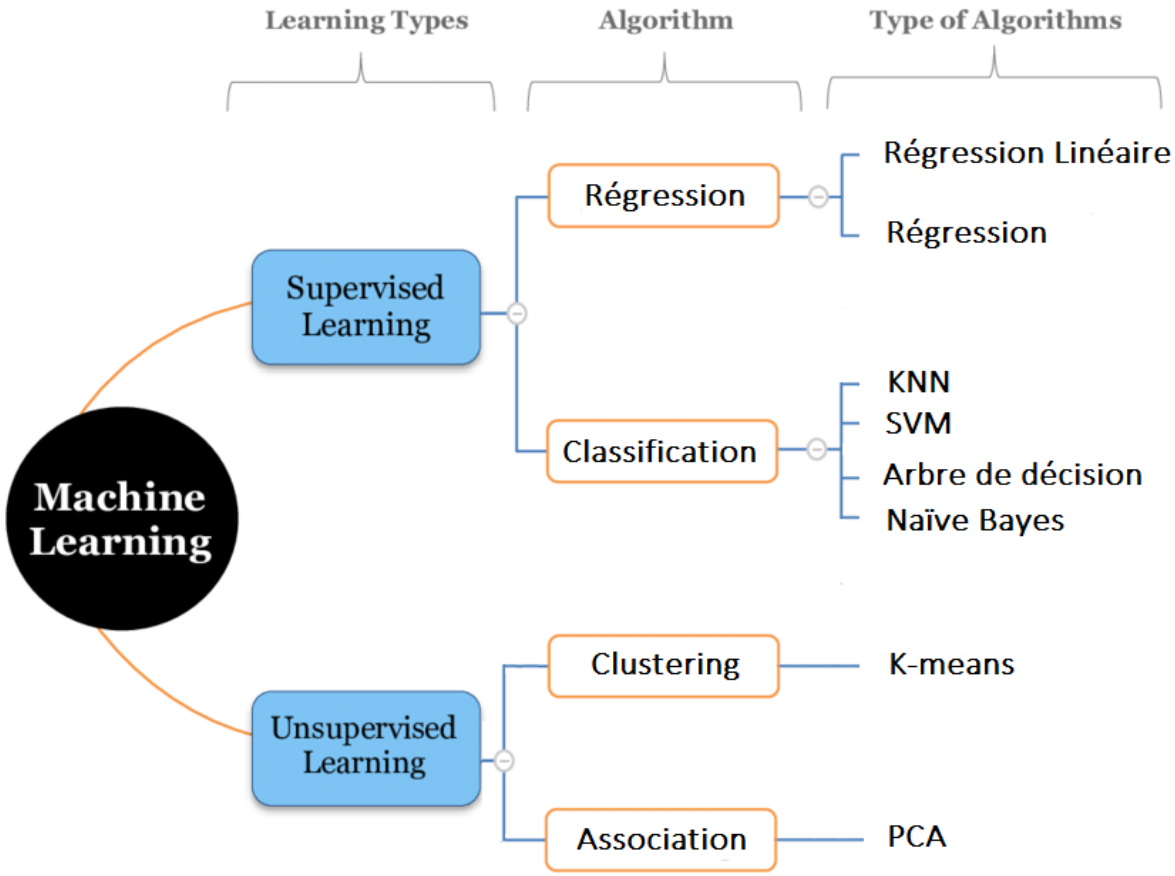

Figure 9 : Les types de Machine Learning

3.1.1 L'apprentissage supervisé

La majorité des apprentissages automatiques utilisent un apprentissage supervisé.

L'apprentissage supervisé consiste en des variables d'entrée (x) et une variable de sortie (Y). Vous utilisez un algorithme pour apprendre la fonction de mapping de l'entrée à la sortie.

$$Y = f(X)$$

Le but est d'appréhender si bien la fonction de mapping que, lorsque vous avez de nouvelles données d'entrée (x), vous pouvez prédire les variables de sortie (Y) pour ces données.

C'est ce qu'on appelle l'apprentissage supervisé, car le processus d'un algorithme tiré de l'ensemble de données de formation (training set) peut être considéré comme un enseignant supervisant le processus d'apprentissage. Nous connaissons les réponses correctes, l'algorithme effectue des prédictions itératives sur les données d'apprentissage et est corrigé par l'enseignant. L'apprentissage s'arrête lorsque l'algorithme atteint un niveau de performance acceptable. L'apprentissage supervisé est généralement effectué dans le contexte de la **classification** et de la **régression**.

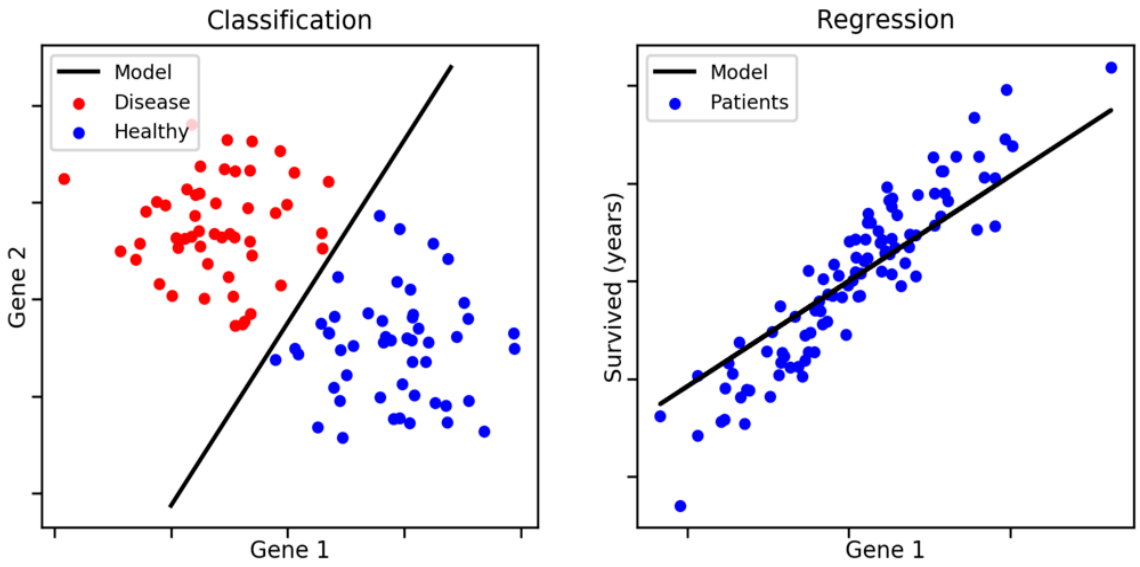

Figure 10 : La Classification et de la Régression

❖ **Classification :** Un problème de classification survient lorsque la variable de sortie est une catégorie, telle que « rouge », « bleu » ou « maladie » et « pas de maladie ».

Exemples :

- En finance et dans le secteur bancaire pour la détection de la fraude par carte de crédit (fraude, pas fraude).
- Détection de courrier électronique indésirable (spam, pas spam).
- Dans le domaine du marketing utilisé pour l'analyse du sentiment de texte (heureux, pas heureux).
- En médecine, pour prédire si un patient a une maladie particulière ou non.

❖ **Régression : Un problème de régression se pose lorsque la variable de sortie est une valeur réelle, telle que « dollars » ou « poids ».**

Exemples :

- Prédire le prix de l'immobilier
- Prédire le cours de bourse

Certains types courants de problèmes fondés sur la classification et la régression incluent la prévision et la prévision de séries temporelles, respectivement.

3.1.2 L'apprentissage non supervisé

L'apprentissage non supervisé (Unsupervised Learning) consiste à ne disposer que de données d'entrée (X) et pas de variables de sortie correspondantes.

L'objectif de l'apprentissage non supervisé est de modéliser la structure ou la distribution sous-jacente dans les données afin d'en apprendre davantage sur les données. On l'appelle apprentissage non supervisé car, contrairement à l'apprentissage supervisé ci-dessus, il n'y a pas de réponse correcte ni d'enseignant. Les algorithmes sont laissés à leurs propres mécanismes pour découvrir et présenter la structure intéressante des données. L'apprentissage non supervisé comprend deux catégories d'algorithmes : Algorithmes de **regroupement** et d'**association**.

❖ Regroupement ou Clustering :

Le clustering est le processus de séparation ou de division des ensembles de données en plusieurs groupes afin que les ensembles de données appartenant au même groupe se ressemblent davantage que les ensembles de données d'autres groupes. En termes simples, l'objectif est de séparer les groupes ayant des caractéristiques similaires et de les affecter à des clusters. Voyons avec un exemple. Supposons que vous soyez le gérant d'un magasin de location et que vous souhaitiez comprendre les préférences des clients pour développer votre entreprise. Vous pouvez diviser tous les clients en 10 groupes en fonction de leurs habitudes d'achat et utiliser une stratégie distincte pour chaque groupe de clients de ces 10 groupes. C'est ce que nous appelons le regroupement.

❖ Association :

L'association consiste à découvrir des relations intéressantes entre des variables dans de grandes bases de données. Par exemple, les personnes qui achètent une nouvelle maison ont aussi tendance à acheter de nouveaux meubles. Il découvre la probabilité de co-occurrence d'éléments dans une collection.

En résumé, le clustering consiste à grouper des points de données en fonction de leurs similitudes, tandis que l'association consiste à découvrir des relations entre les attributs de ces points de données.

3.2 Les algorithmes des machines Learning

3.2.1 Algorithmes de régression

C'est un processus de recherche d'un modèle ou d'une fonction permettant de distinguer les données en valeurs réelles continues au lieu d'utiliser des classes ou des valeurs discrètes. il existe 2 types de cet algorithme :

3.2.1.1 La régression Linéaire

La régression linéaire est l'un des algorithmes d'apprentissage supervisé les plus populaires. Il est aussi simple et parmi les mieux compris en statistique et en apprentissage automatique.

La régression linéaire est un type d'analyse prédictive de base. Le concept général de la régression est d'étudier deux questions :

- Un ensemble de variables prédictives permet-il de prédire une variable de résultat ?
- Quelles sont les variables les plus significatives et ont le plus d'impact sur la variable de résultat ?

Les algorithmes de régression linéaire modélisent la relation entre des variables prédictives et une variable cible. La relation est modélisée par une fonction mathématique de prédiction. Le cas le plus simple est la régression linéaire uni variée. Elle va trouver une fonction sous forme de droite pour estimer la relation. La régression linéaire multivariée intervient quand plusieurs variables explicatives interviennent dans la fonction de prédiction.

Finalement, la régression polynomiale permet de modéliser des relations complexes qui ne sont pas forcément linéaires.

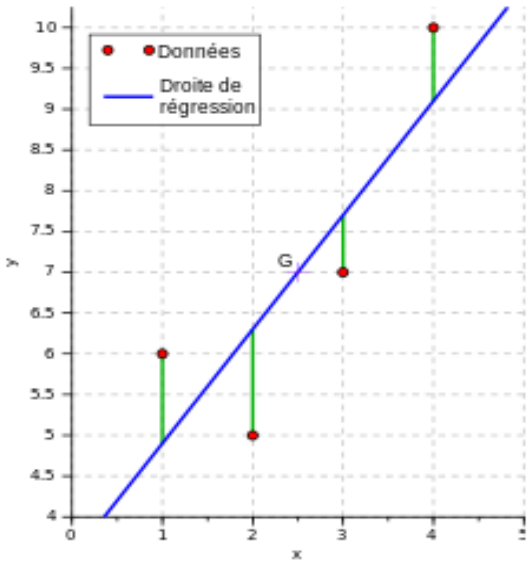

Figure 11 : Régression linéaire [23].

3.2.1.2 La régression logistique

La régression logistique est devenue un outil important dans la discipline de l'apprentissage automatique. Cette approche permet d'utiliser un algorithme dans l'application d'apprentissage automatique pour classer les données entrantes en fonction des données historiques. Plus il y a de données pertinentes en entrée, plus l'algorithme est en mesure de prédire des classifications au sein des jeux de données. La régression logistique ou modèle logit est un modèle de régression binomiale. Comme pour tous les modèles de régression binomiale, il s'agit de modéliser au mieux un modèle mathématique simple à des observations réelles nombreuses. En d'autres termes d'associer à un vecteur de variables aléatoires (x1,…,xk) une variable aléatoire binomiale génériquement notée y. La régression logistique constitue un cas particulier de modèle linéaire généralisé [23] .

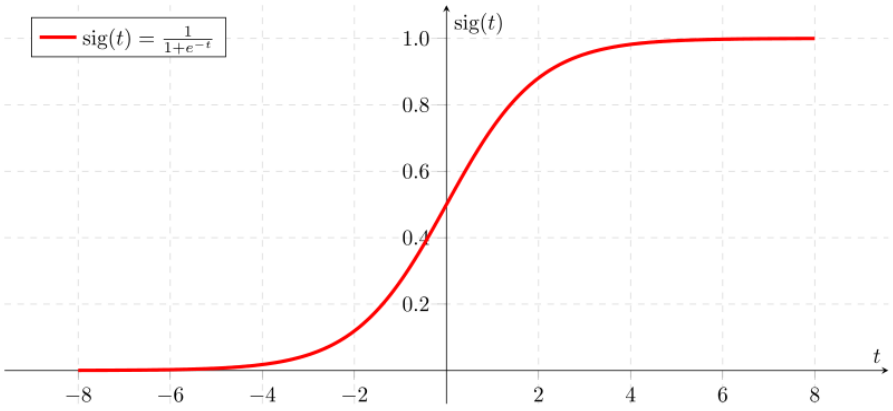

Figure 12 : Graphe et expression de la fonction sigmoïde

3.2.2 Algorithme de Classification

La classification automatique est le processus qui permet d'analyser et d'organiser un ensemble de données, selon leurs caractéristiques, dans des classes de similarité. Elle se base principalement sur des représentations classiques de données dont les limites de traitement sont connues et, qui dans la plupart du temps, demande un temps de calcul énorme [24] .

3.2.2.1 K plus proches voisins (KNN)

K Nearest **N**eighbor (**PPV P**lus **P**roches **V**oisins) est une méthode dédiée à la classification et peut être étendue aux tâches d'estimation. La méthode PPV est une méthode de raisonnement par cas. Cela commence par l'idée de prendre une décision en trouvant un ou plusieurs cas similaires qui ont été résolus en mémoire. Aucune étape de formation n'est incluse pour créer un modèle à partir d'échantillons de formation. L'échantillon d'apprentissage est associé à la

fonction de distance et à la fonction de sélection de classe en tant que fonction de classe voisine la plus proche pour former un modèle.

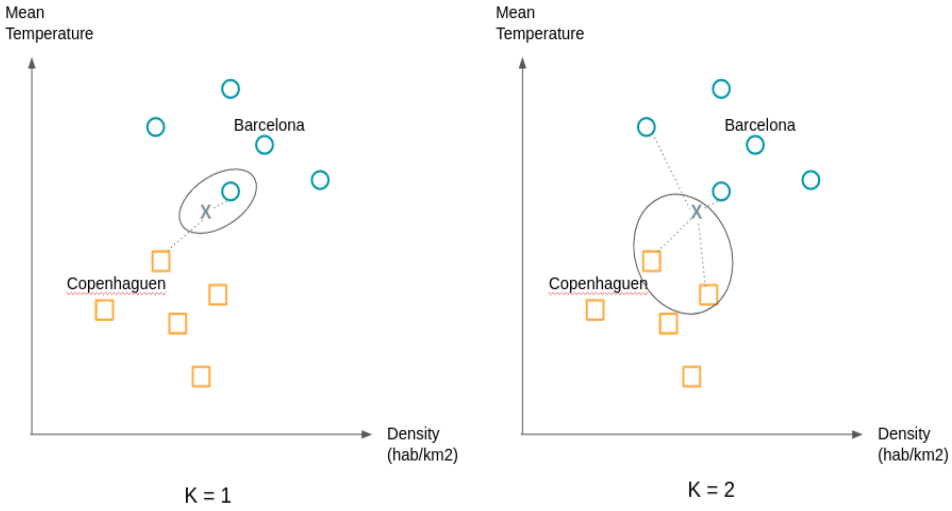

Figure 13 : K nearest neighbours

On affecte à une observation la classe de ses K plus proches voisins. " C'est tout ?!" me direz-vous. Oui c'est tout, seulement comme l'exemple suivant le montre le choix de K peut changer beaucoup de choses. On cherchera donc à essayer différentes valeurs de K pour obtenir la séparation la plus satisfaisante [25].

3.2.2.2 Algorithme les machines à support de vecteurs (SVM)

SVM (Support Vector Machine ou Machine à vecteurs de support) : Les SVMs sont une famille d'algorithmes d'apprentissage automatique qui permettent de résoudre des problèmes tant de classification que de régression ou de détection d'anomalie. Ils sont connus pour leurs solides garanties théoriques, leur grande flexibilité ainsi que leur simplicité d'utilisation même sans grande connaissance de data mining.

Les SVMs ont été développés dans les années 1990. Comme le montre la figure ci-dessous, leur principe est simple : il ont pour but de séparer les données en classes à l'aide d'une frontière aussi « simple » que possible, de telle façon que la distance entre les différents groupes de données et la frontière qui les sépare soit maximale. Cette distance est aussi appelée « marge » et les SVMs sont ainsi qualifiés de « séparateurs à vaste marge », les « vecteurs de support » étant les données les plus proches de la frontière [26] .

Reprenons notre exemple de destinations idéales de vacances. Pour la simplicité de notre exemple considérons seulement 2 variables pour décrire chaque ville : la température et la densité de population. On peut donc représenter les villes en 2 dimensions.

Nous représentons par des ronds des villes que vous avez beaucoup aimé visiter et par des

carrés celles que vous avez moins apprécié. Lorsque vous considérez de nouvelles villes vous souhaitez savoir de quelle groupe cette ville se rapproche-t-elle le plus.

Comme nous le voyons sur le graphique de droite il existe de nombreux plans (des droites lorsque nous n'avons que 2 dimensions) qui sépare les deux groupes.

On va choisir la droite qui est à la distance maximale entre les deux groupes. Pour le construire nous voyons déjà que n'avons pas besoin de tous les points, il suffit de prendre les points qui sont à la frontière de leur groupe on appelle ces points ou vecteurs, les vecteurs supports. Les plans passant par ces vecteurs supports sont appelés plans supports. Le plan de séparation sera celui qui sera équidistant des deux plans supports.

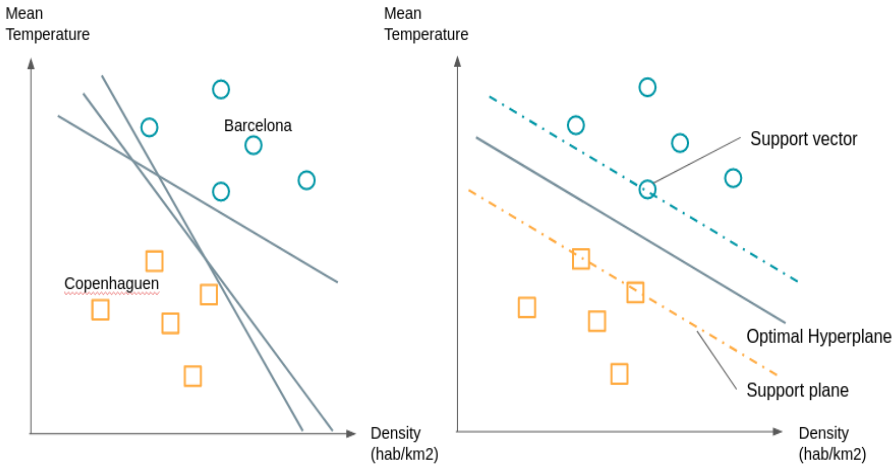

Figure 14 : SVM Exemple

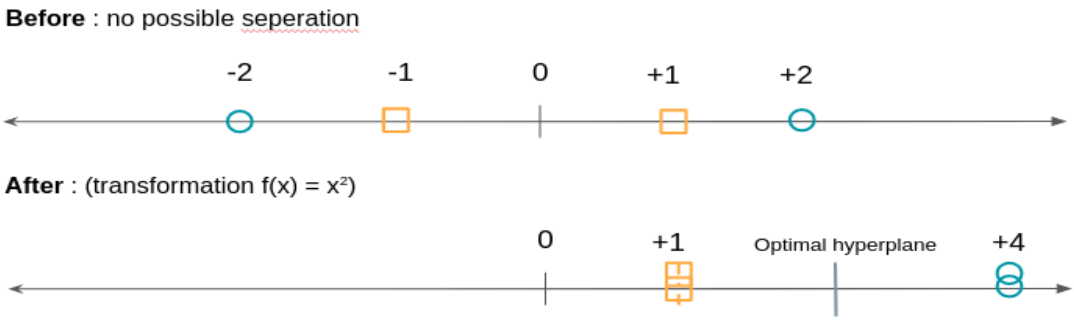

Figure 15 : SVM Exemple 2

L'algorithme SVM va donc consister à chercher à la fois l'hyperplan optimal ainsi que de minimiser les erreurs de classification [25] .

3.2.2.3 Algorithme arbre de décision (DT)

Un arbre de décision est un algorithme qui peut être utilisé dans de nombreux environnements: commerce de détail, finance, pharmaceutique, etc. La machine dessine

simplement un arbre de divers résultats qui peuvent ou non se produire, et suit chaque événement jusqu'à sa conclusion naturelle tout en calculant toutes les probabilités d'événements possibles [27]. Étant donné un corpus d'observations étiquetées, un arbre de décision est utilisé pour classer les observations futures. C'est le cas dans notre exemple de botanique, où nous avons classé 100 observations en espèces A, B et C. L'arbre commence par la racine (on a toutes les observations), puis une série de branches dont les intersections sont appelées nœuds et se terminent par des feuilles, chaque feuille correspondant à une classe à prédire. Nous appelons la profondeur de l'arbre le nombre maximum de nœuds avant d'atteindre la feuille. Chaque nœud de l'arbre représente une règle (exemple : longueur du pétale supérieure à 2,5 cm). Parcourir l'arbre c'est donc vérifier une série de règles. L'arbre est construit de telle sorte que chaque nœud correspond à la règle (type de mesure et seuil) qui divisera le mieux l'ensemble d'observations de départ.

Exemple [25]:

Données :

Observation	Petal length	Stem length	Species
1	2.7 cm	20 cm	A
2	2.6 cm	12 cm	C
3	2.1 cm	21 cm	B
4	1.9 cm	20 cm	B

Data

Tableau 5 : Arbre data exemple

Arbre :

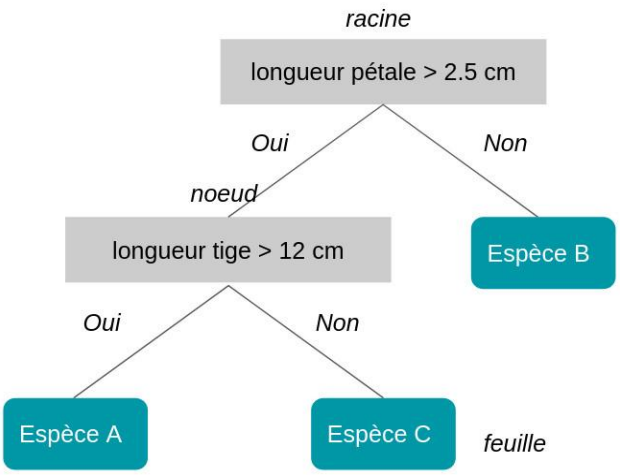

Figure 16 : Arbre exemple

L'arbre à une profondeur de 2 (un nœud plus la racine). La longueur du pétale est la première mesure qui est utilisée car elle sépare le mieux les 4 observations selon l'appartenance aux classes (ici à la classe B).

- **Les algorithmes d'induction des arbres de décision :**

Nom de l'algorithme	Développeur	Année
CHAID	Kass	1980
CART	Breiman, et al.	1984
ID3	Quinlan	1986
C4.5	Quinlan	1993
SLIQ	Agrawal, et al.	1996
SPRINT	Agrawal, et al.	1996

Tableau 6 : Les algorithmes d'inductions des arbres de décision [28].

Les algorithmes pour créer un arbre de décision sont : ID3, C4.5, CART.

Pour classifier un exemple, il faut partir de la racine de l'arbre et descendre jusqu'à une feuille en respectant les prédicats.

3.2.2.4 L'algorithme de Naïve Bayes

La catégorisation manuelle des pages Web, des documents, des e-mails ou de toute annotation de texte volumineux est difficile, voire impossible. C'est là que l'algorithme d'apprentissage automatique du classificateur naïf de Bayes entre en jeu. Le classificateur est une fonction qui attribue la valeur de l'élément global à l'une des catégories disponibles. Par exemple, le filtrage anti-spam est une application populaire de l'algorithme Naïve Bayes. Filtre anti-spam ici, est un classificateur qui attribue une étiquette « Spam » ou « Pas de spam » à tous les emails.

Le classificateur naïf de Bayes est l'une des méthodes d'apprentissage supervisé les plus populaires dans les méthodes de similarité, qui est basée sur le théorème de probabilité bayésien populaire. Surtout pour la prédiction des maladies et la classification des documents. Il s'agit d'une classification de mots simple basée sur le théorème de probabilité bayésien pour l'analyse de contenu subjective [23].

Théorème de Bayes : Le théorème de Bayes est une formule mathématique simple utilisée pour calculer les probabilités conditionnelles [29].

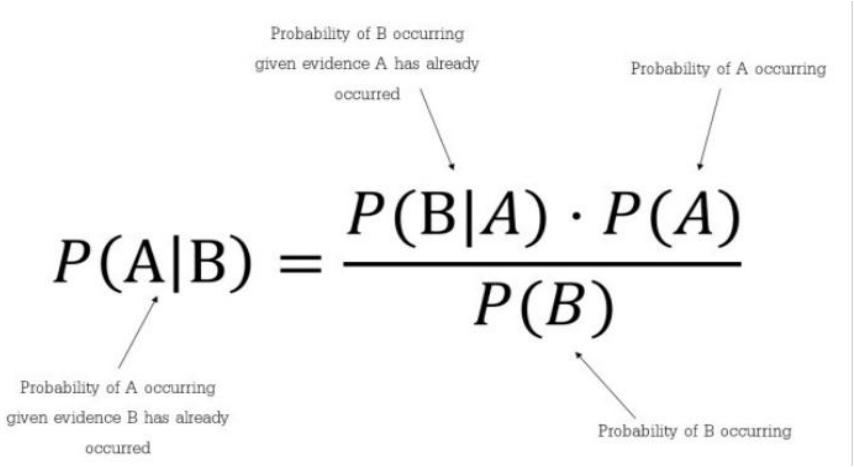

Figure 17 : Théorème de Bayes

3.2.3 Clustering

L'algorithme de clustering est un algorithme d'apprentissage automatique très utile pour identifier des groupes de comportements similaires. L'algorithme peut traiter les données et les classer. Les algorithmes de clustering, peuvent découvrir automatiquement le comportement de groupes de clients (ou clusters) ayant des caractéristiques similaires. Cela permet donc de regrouper les téléspectateurs en fonction de comportements similaires, d'identifier facilement les « électrons libres » qui n'appartiennent pas à un certain groupe, et même de découvrir des comportements qui ne sont pas a priori inconnus. Cela facilite la personnalisation des services et offre une expérience client de qualité [30].

3.2.3.1 K-means (K-moyen)

Les K-Means sont des algorithmes de machine learning sans supervision qui sont utilisés pour résoudre des problèmes de clustering. Ils divisent et classent un ensemble de points de données non affectés d'un label (sans classification externe) en un groupe appelé « cluster » (sans rapport avec les clusters de serveurs). Chaque itération de l'algorithme assigne chaque point à un groupe présentant des caractéristiques similaires. Les points de données peuvent être suivis dans le temps pour détecter les changements qui se produisent dans les clusters.

Les algorithmes K-Means peuvent confirmer des hypothèses sur les types de groupes qui existent dans un dataset spécifique, ou être utilisés pour découvrir des clusters inconnus. Parmi les cas d'usage commerciaux, citons le regroupement de l'inventaire par activité commerciale et la détection d'anomalies dans des données – par exemple, les données brutes collectées par un bot (robot logiciel) [27].

3.2.4 Réduction de dimensions

La réduction de dimensionnalité (ou réduction de (la) dimension) est un processus étudié en mathématiques et en informatique, qui consiste à remplacer des données dans un espace de grande dimension par des données dans un espace de plus petite dimension. Pour que l'opération soit utile, les données de sortie doivent représenter correctement les données d'entrée.

3.2.4.1 PCA

L'analyse en composantes principales est un algorithme d'apprentissage non supervisé utilisé pour la réduction de la dimensionnalité dans l'apprentissage automatique . Il s'agit d'un processus statistique qui convertit les observations de caractéristiques corrélées en un ensemble de caractéristiques linéairement non corrélées à l'aide d'une transformation orthogonale.

Ces nouvelles fonctionnalités transformées sont appelées les composants principaux. C'est l'un des outils populaires utilisés pour l'analyse exploratoire des données et la modélisation prédictive. C'est une technique pour dessiner des modèles forts à partir de l'ensemble de données donné en réduisant les variances. Certaines applications réelles de la PCA sont le traitement d'images, le système de recommandation de films, l'optimisation de l'allocation de puissance dans divers canaux de communication**.** C'est une technique d'extraction de caractéristiques, elle contient donc les variables importantes et supprime la variable la moins importante. L'algorithme PCA est basé sur certains concepts mathématiques tels que :

- Variance et covariance
- Valeurs propres et facteurs propres [31]

- **Avantages et inconvénients les algorithmes de machine learning**

Apprentissage	Type	Algo	Avantages	Inconvénients
Supervisé	**Classification**	**KNN**	• Facile à implémente. • Efficace. [32] • L'algorithme est polyvalent [33]	• Calculer chaque fois la similarité entre les k. [34] • Grande capacité de stockage. • Utilise de nombreuses données de références pour classifier les nouvelles entrées. [32]
		SVM	• Leur capacité à manipuler de grandes quantités de données • Le faible nombre d'hyper paramètres. • Elles sont bien fondées théoriquement. [35]	• Complexes pour la classification des corpus. • Demande un temps énorme pendant les phases de test. [36]
		Arbre de décision	• Faciles à comprendre. • Ils permettent de sélectionner l'option la plus appropriée parmi plusieurs. • Il est facile de les associer à d'autres outils de prise de décision. [37]	• Instables. [37] • Certains concepts sont difficiles à exprimer à l'aide d'arbres de décision (comme XOR). [38]
		Naïve Bayes	• La facilité et la simplicité de leur implémentation. • Leur rapidité. • Les méthodes Naïve Bayes donnent de bons résultats. [39]	• Faire le même travail de classification. [40] [41]
	Régression	**Linéaire**	• Simplicité d'interprétation. • Facilité de calcul [42]	• Elle ne traite pas les valeurs manquantes de variables continues sensible

				aux valeurs hors norme de variables continues [43]
Non Supervisé	**Association**	**PCA**	• Simplicité mathématique • Simplicité des résultats • Puissance • Flexibilité [44]	• L'ACP n'a pas réellement • S'applique simplement sur des cas précis • Perte d'information par l'emploi fréquent de la 1ère composante principale uniquement. [44]
	Clustering	**K-means**	• Simple • Flexible • Efficace • Complexité temporelle. [45]	• Ensemble non optimal de clusters • Manque de cohérence • Limitation des calculs • Spécifiez les valeurs k [45]

Tableau 7 : Avantages et inconvénients les algorithmes de machine learnin

4. Sélection des caractéristiques

La sélection des caractéristiques est l'une des étapes cruciales du cycle de vie de tout projet de science des données. Lorsque nous formons notre modèle d'apprentissage automatique, toutes les fonctionnalités ne contribuent pas de la même manière. Certaines fonctionnalités sont importantes et d'autres n'aident même pas du tout à la formation. Nous devons donc supprimer ces types d'entités de notre ensemble de données. Par conséquent, fonction de sélection est le processus d'élimination ou de réduire les caractéristiques d'entrée / variables qui, à son tour réduit la complexité, rend le processus de formation plus rapide et augmente également la précision du modèle. [46] La sélection des fonctionnalités est essentielle à la création d'un modèle performant pour plusieurs raisons. La première raison est que la sélection des fonctionnalités implique une certaine réduction de cardinalité pour imposer une limite du nombre d'attributs pouvant être pris en compte lors de la création d'un modèle. Les données contiennent presque toujours plus d'informations que nécessaire pour générer le modèle ou elles contiennent un type d'informations inapproprié. Par exemple, vous pouvez avoir un dataset de 500 colonnes qui décrivent les caractéristiques des clients. Toutefois, si certaines colonnes contiennent des données éparses, cela n'est pas très utile de les ajouter au modèle et, si certaines colonnes sont en double, leur utilisation peut rendre le modèle inexact. La sélection des fonctionnalités améliore la qualité du modèle, tout en optimisant le processus de modélisation. Si vous incluez des colonnes inutiles pour générer un modèle, davantage d'UC et de mémoire sont consommées pendant le processus d'apprentissage, et il faut plus d'espace de stockage pour le modèle terminé. Indépendamment du problème des ressources, vous avez intérêt à effectuer la sélection des fonctionnalités et à identifier les colonnes les plus pertinentes, car l'utilisation de colonnes inutiles peut diminuer la qualité du modèle de plusieurs façons :

- Des données parasites ou redondantes rendent plus difficile la découverte de séquences significatives.
- En présence d'un jeu de données multidimensionnel, la plupart des algorithmes d'exploration des données nécessitent un jeu de données d'apprentissage beaucoup plus important.

Pendant le processus de sélection des fonctionnalités, l'analyste ou l'outil de modélisation ou l'algorithme sélectionne ou ignore de façon active les attributs en fonction de leur utilité pour l'analyse. L'analyste peut effectuer l'ingénierie des fonctionnalités pour ajouter des fonctionnalités et supprimer ou modifier des données existantes, tandis que l'algorithme

d'apprentissage automatique calcule généralement les scores des colonnes pour valider leur utilité dans le modèle.

Pour résumer, la sélection des fonctionnalités permet à la fois d'éviter d'avoir trop de données qui présentent peu d'intérêt ou de n'avoir pas assez de données utiles. Votre objectif, en utilisant la sélection des fonctionnalités, est d'identifier le nombre minimum de colonnes de la source de données qui sont importantes pour le modèle à créer [47]

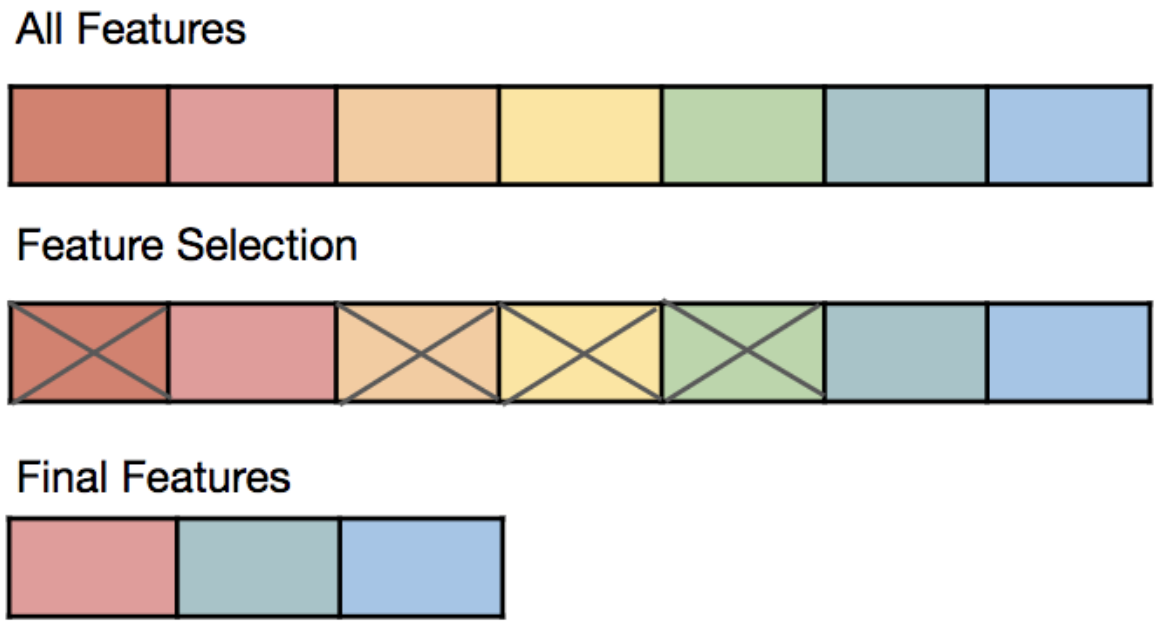

Figure 18 : sélection des caractéristiques [48]

4.1 Les algorithmes génétiques (AG)

Algorithme génétique (AG) est une métaheuristique inspirée du processus de sélection naturelle qui appartient à la classe plus large des algorithmes évolutionnaires (EA). Les algorithmes génétiques sont couramment utilisés pour générer des solutions de haute qualité aux problèmes d' optimisation et de recherche en s'appuyant sur des opérateurs d'inspiration biologique tels que la mutation , le croisement et la sélection . [49]

Les algorithmes génétiques ont montré leur efficacité pour résoudre des problèmes d'optimisation dont l'espace de recherche est de grande dimension. L'application des AGs sur des problèmes de sélection de sous-ensembles de caractéristiques a été mise en place en 1993 par Ferri et al. [50] Qui ont montré que l'utilisation des AGs est bien adaptée pour la sélection sur des ensembles de caractéristiques de taille moyenne (20 _a 49 caractéristiques) pour lesquels la plupart des méthodes classiques nécessitent un temps de calcul _énorme pour réaliser une sélection.

En 2000, Kudo et Sklansky [51] ont montré la possibilité d'utiliser les AGs pour la sélection sur des ensembles de grande _échelle (50 caractéristiques et plus) en ajustant les paramètres de l'AG (le nombre de générations, la taille de la population et les probabilités des opérations génétiques) d'un côté et la fonction d'évaluation de l'autre côté. Une fois que les paramètres ont été bien fixé et la fonction d'évaluation bien définie, ils ont montré que les résultats de l'AG, sont meilleurs que ceux des méthodes basées sur la recherche séquentielle. La procédure de sélection par un algorithme génétique est illustrée par la figure 22.

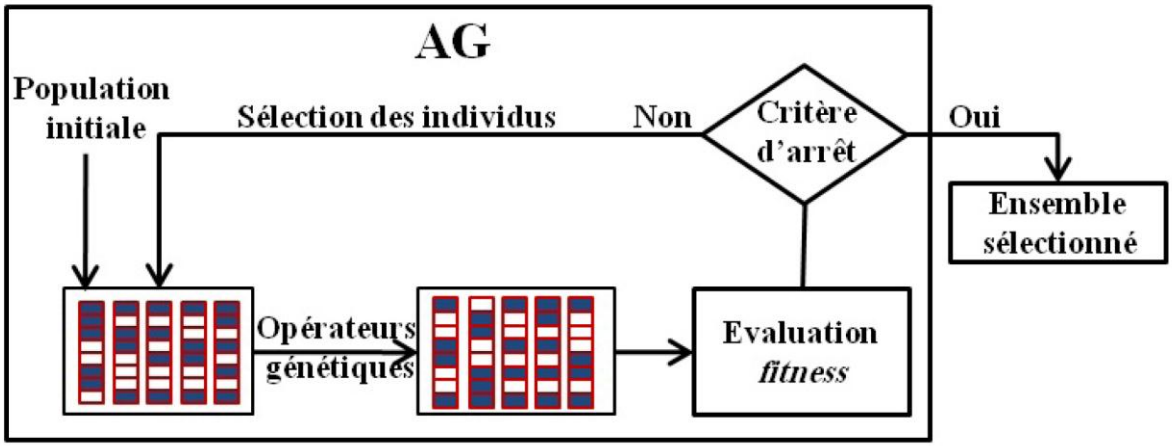

Figure 19 : Sélection de caractéristiques par un algorithme génétique

En termes de sélection de caractéristiques, chaque chromosome représentera un sous-ensemble de caractéristiques, et il sera représenté avec un codage binaire : 1 signifie **« choisir »** une caractéristique donnée et 0 signifie **« ne pas choisir »** une caractéristique.

Après cela, nous effectuons de nombreuses itérations pour créer une nouvelle génération (nouveau sous-ensemble de fonctionnalités) de solutions possibles à partir de la génération actuelle à l'aide de quelques opérateurs :

- **Sélection :** filtre de manière probabiliste les solutions peu performantes tout en choisissant les solutions les plus performantes à exploiter.
- **Cross Over :** c'est le moyen utilisé par GA pour explorer de nouvelles solutions et échanger des informations entre les chaînes. Cet opérateur sera appliqué pour sélectionner des paires de chromosomes de manière aléatoire, avec une probabilité égale à un taux de croisement donné. De cette façon, nous générons de nouveaux chromosomes qui, espérons-le, conserveront les bonnes caractéristiques des générations précédentes.
- **Mutation :** Cet opérateur protège les GA contre la perte irrémédiable des bonnes

fonctionnalités de la solution. Il modifie un symbole de certains chromosomes avec une probabilité égale à un taux de mutation donné très faible afin de restaurer le matériel génétique perdu.

Voici les étapes de la sélection des caractéristiques avec des algorithmes génétiques :

a) Initialisez une population avec des individus générés aléatoirement (dans notre cas, cela signifie différents sous-ensembles de caractéristiques) et créez un algorithme d'apprentissage automatique.
b) Évaluez l'adéquation de chaque sous-ensemble de fonctionnalités avec une métrique d'évaluation de votre choix, en fonction de l'algorithme choisi.
c) Reproduire des chromosomes de haute condition physique (sous-ensemble de caractéristiques) dans la nouvelle population.
d) Supprimer les chromosomes de mauvaise condition physique (sélection).
e) Construire de nouveaux chromosomes (croisement).
f) Récupérer les fonctionnalités perdues (mutation).
g) Répétez les étapes 2 à 6 jusqu'à ce qu'un critère d'arrêt soit atteint (peut être le nombre d'itérations, par exemple).

4.2 Swarm Intelligence (SI)

Un système d'intelligence en essaim (SI) est généralement constitué d'une population d'agents non compliqués qui interagissent à proximité les uns des autres et avec leur environnement. L'inspiration provient régulièrement de la nature, principalement des systèmes biologiques. Les agents suivent des procédures assez simples, et bien qu'il n'y ait pas de structure de gestion centralisée indiquant comment les agents individuels doivent se comporter. L'émergence d'un comportement "intelligent" à l'échelle mondiale peut être due en outre aux interactions entre les agents. Les algorithmes SI tels que l'optimisation par colonies de fourmis (ACO), l'optimisation par colonies d'abeilles artificielles (ABC), l'optimisation par essaims de particules (PSO) et d'autres systèmes ont prouvé leur puissante capacité dans les questions biologiques. Il a été prouvé que ces algorithmes fournissent des résultats appropriés dans un grand nombre d'applications du monde réel [52].

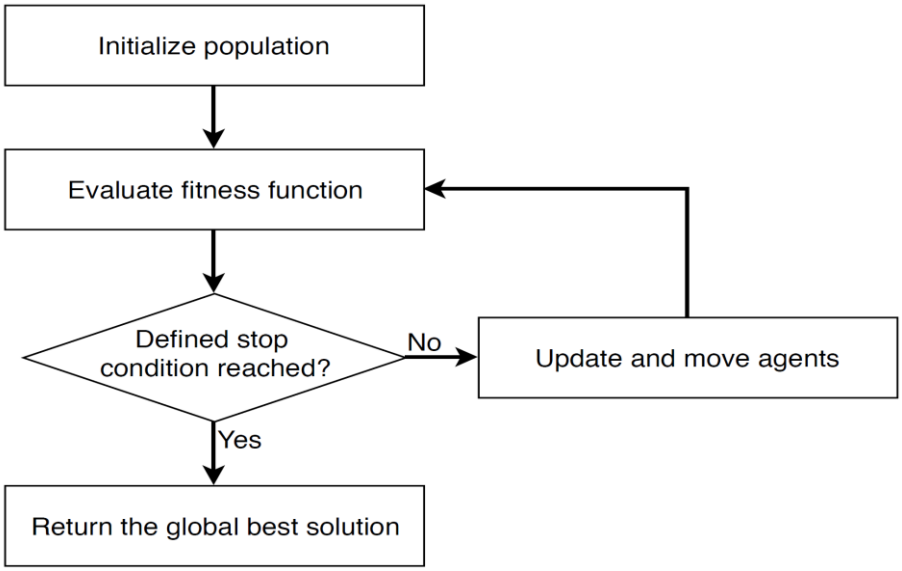

Figure 20 : Cadre de l'intelligence en essaim [53]

La figure 20 a présenté le cadre général des algorithmes SI, dans lequel la recherche de la meilleure solution est effectuée via un essaim d'agents. Chaque agent détient une solution candidate. La recherche commence par une initialisation aléatoire des agents en fonction du sujet. Ceci peut être suivi à l'aide d'une évaluation de la qualité des solutions candidates proposées par les agents. La troisième étape est celle de la nouvelle génération de sous-ensemble qui diffère d'un algorithme à l'autre. La génération de nouveaux sous-ensembles candidats suit les sources d'inspiration des algorithmes [53].

4.3 Artificial Bee Colony (ABC)

ABC est l'un des algorithmes d'optimisation par recherche en essaim les plus récents. Il a été proposé pour la première fois par Dervis Karaboga en 2005 [54]. Il s'agit d'un algorithme d'optimisation inspiré de la conduite intelligente de recherche de nourriture d'un essaim d'abeilles pour trouver une solution ultime. Cet algorithme est considéré comme aussi simple et facile à mettre en œuvre que PSO et DE [55].

Les caractéristiques moins distinctives de l'ensemble des données affectent négativement la classification. De telles données diminuent notamment la vitesse et les performances du système de manière significative. Avec le système proposé, en utilisant l'algorithme de sélection des caractéristiques, les caractéristiques avec des données moins discriminantes ont été éliminées. L'ensemble de données réduit a augmenté le succès du test du classificateur et la vitesse du système. D'après la figure 21, le système proposé comporte deux phases. Dans la première phase, comme critère de sélection, le clustering avec l'algorithme ABC a été utilisé pour la sélection des caractéristiques, et, ainsi, une méthode de sélection des caractéristiques

plus efficace a été constituée. Ainsi, il a été possible de sélectionner les caractéristiques connexes dans un laps de temps plus court et de réduire la dimension du vecteur de caractéristiques. Dans un deuxième temps, les données réduites obtenues sont fournies au classificateur DT pour déterminer les taux de précision. La validation croisée -fold a été utilisée pour améliorer la fiabilité du classificateur.

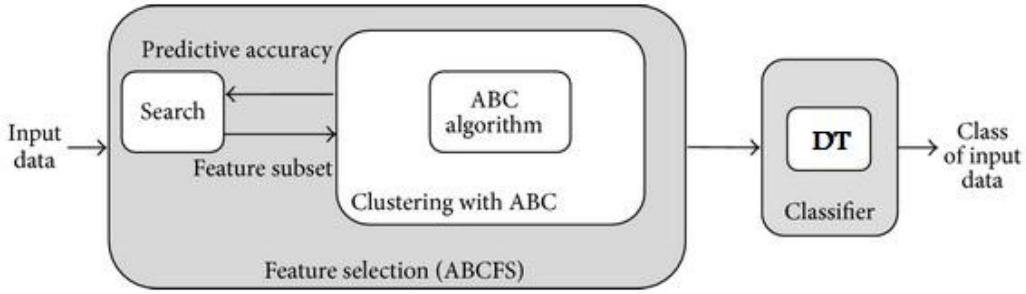

Figure 21 : système proposé comporte ABC + DT

Algorithm 1 ABC optimization approach

1: Initialization Phase
2: **repeat**
3: Employed Bee Phase
4: Onlooker Bee Phase
5: Scout Bee Phase
6: Memorize the best solution achieved so far
7: **until** (Cycle = Maximum Cycle Number or a Maximum CPU time)

Figure 22 : pseudocode pour l'approche d'optimisation ABC [56]

5. Conclusion

Dans ce chapitre, nous avons présenté les algorithmes de machine learning servaient à deux choses : classifier et prédire et qu'ils se divisaient en algorithmes supervisés et non supervisés. Il existe de nombreux algorithmes possibles ainsi que les avantages et les inconvénients de chaque méthode.

On a présenté les algorithmes de sélection des caractéristiques, et nous discutons de 3 techniques ou astuces pour sélectionner le meilleur sous-ensemble de fonctionnalités à partir d'un ensemble de données. On peut utiliser ces hacks dans votre modèle de science des données pour sélectionner le meilleur sous-ensemble de fonctionnalités et former un modèle robus.

PARTIE 2

Approche proposée et Implémentation

Chapitre 03

L'approche proposée

1. Introduction

Dans ce chapitre, nous présentons notre approche proposée de notre IDS, et nous décrivons et révélons toutes les étapes de la mise en œuvre de notre approche, Ainsi, une mise en œuvre du modèle est présentée et effectuée en utilisant le langage Python.

En commençant tout d'abord par une présentation l'ensemble de données utilisés et le matériel réalisé. Ensuite, nous présentons des captures d'écran de l'exécution de notre application.

2. Performance Evaluation

Tous les tests ont été effectués à l'aide d'un ordinateur portable avec les détails de configuration matérielle et logicielle indiqués dans le tableau 8.

Fabricant	*Toshiba*
Précision du modèle	Satellite L50-B
Type de système	Système d'exploitation 64 bits, processeur x64
Système opérateur	Windows 7 Professional
Type de processeur	Intel® Core i3-4005U
La vitesse du processeur	1.70 GHz
Mémoire installée (RAM)	8 GB
Nombres de cœurs	2
Le nombre de fils	4
Outil d'apprentissage automatique	Anaconda3-2021.05

Tableau 8 : La configuration matérielle et logicielle

3. Ensemble de données d'évaluation de détection d'intrusion (CICDDoS2019)

L'ensemble de données choisi pour cette étude c'est CIC-DDoS-2019, qui contient les attaques DDoS courantes bénignes et les plus récentes, qui ressemblent aux vraies données du monde réel. Il comprend également les résultats de l'analyse du trafic réseau à l'aide de CICFlowMeter-V3 avec des flux étiquetés en fonction de l'horodatage, des adresses IP source et de destination, des ports source et destination, des protocoles et de l'attaque (fichiers CSV). [57]

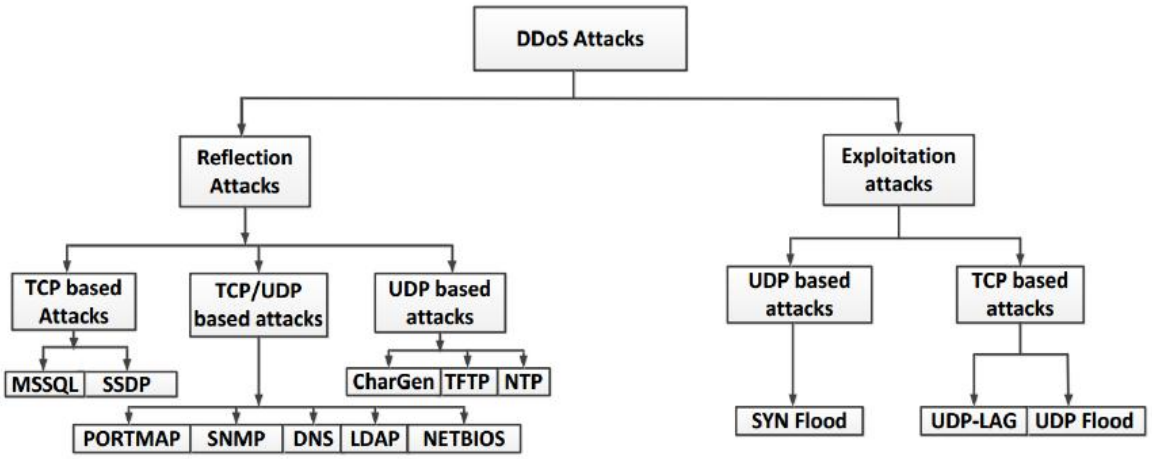

Figure 23 : DDoS Taxonomie d'attaque

Ainsi, notre dataset contient plusieurs types d'attaques voici le tableau suivant :

Type ID	Attack type	Attack time
1	PortMap	9 : 43–9 : 51
2	NetBIOS	10 : 00-10 : 09
3	LDAP	10 : 21-10 : 30
4	MSSQL	10 : 33-10 : 42
5	UDP	10 : 53-11 : 03
6	UDPLag	11 : 14-11 : 24
7	SYN	11 : 28-17 : 35

Tableau 9 : Heure d'attaque DDoS le 3 novembre

Type ID	Attack type	Attack time
1	NTP	10 : 35–10 : 45
2	DNS	10 : 52–11 : 05
3	LDAP	11 : 22–11 : 32
4	MSSQL	11 : 36–11 : 45
5	NetBIOS	11 : 50–12 : 00
6	SNMP	12 : 12–12 : 23
7	SSDP	12 : 27–12 : 37
8	UDP	12 : 45–13 : 09
9	UDPLag	13 : 11–13 : 15
10	TFTP	13 : 35–17 : 15

Tableau 10 : Heure de l'attaque DDoS le 1er décembre

Le tableau fournit une brève description des attaques DDoS basées sur la réflexion et l'exploitation.

Classe	Description
SYN Flood	SYN flood est une attaque par déni de service dans laquelle un attaquant envoie une succession de requêtes SYN à un système cible dans le but de consommer les ressources du serveur afin que le système ne réponde pas au trafic légitime [58].
WebDDoS	Le WebDDoS est une attaque visant à mettre hors service le site Web cible ou à le ralentir en inondant le réseau, le serveur ou l'application de faux trafic [59].
TFTP	Une attaque TFTP exploite la vulnérabilité de débordement de tampon dans un serveur TFTP (Trivial File Transfer Protocol) [60].
DNS	Une attaque DNS exploite les vulnérabilités du DNS [61].
PORTMAP	PORTMAP est une attaque sur le port TCP ou UDP 111, qui est un service utilisé pour diriger les clients vers le numéro de port approprié afin qu'ils puissent communiquer avec le service RPC (Remote Procedure Call) demandé [62].
MSSQL	L'injection de Microsoft Structured Query Language (MSSQL) est une attaque qui permet d'exécuter des instructions SQL malveillantes [63].
LDAP	L'injection LDAP est une attaque utilisée pour exploiter les applications Web qui construisent des déclarations LDAP sur la base des entrées utilisateur [64].
NETBIOS	Une faille de sécurité dans le système d'entrée/sortie de base du réseau (NetBIOS) permet à un attaquant de voir des informations dans la mémoire d'un ordinateur sur un réseau [65].
NTP	NTP est une attaque par amplification dans laquelle l'attaquant exploite des serveurs NTP accessibles au public pour submerger la cible de trafic UDP [66].
SSDP	Une attaque SSDP exploite les protocoles de réseau Universal Plug and Play (UPnP) afin d'envoyer une grande quantité de trafic à une victime pour submerger ses ressources informatiques [67].
SNMP	Une attaque SNMP (Simple Network Management Protocol) génère une grande quantité de trafic qui est dirigé vers les victimes à partir de plusieurs réseaux [68].
UDP	L'inondation du protocole UDP (User Datagram Protocol) est une attaque dans laquelle un grand nombre de paquets UDP sont envoyés à une victime dans le but de dépasser sa capacité de traitement et de réponse. Le pare-feu protégeant le serveur cible est alors épuisé [69].
UDP-Lag	UDP-Lag est une attaque qui perturbe la connexion entre le client et le serveur [70].
CharGEN	L'inondation par le protocole de génération de caractères (CharGEN) est une attaque qui consiste à envoyer de petits paquets portant l'adresse IP usurpée de la victime à des dispositifs Internet exécutant CharGEN afin

d'épuiser les ressources informatiques [71].

Tableau 11 : fournit une brève description des attaques DDoS basées sur la réflexion et l'exploitation

L'ensemble de données est organisé par jour. Pour chaque jour, les données brutes, y compris le trafic réseau (Pcaps) et les journaux d'événements (journaux d'événements Windows et Ubuntu) ont été enregistrées pour chaque appareil. Dans le processus d'extraction des caractéristiques des données brutes, CICFlowMeter-V3 a été utilisé et plus de 80 caractéristiques de trafic ont été extraites et enregistrées sous forme de fichier CSV pour chaque appareil.

Fonctionnalité	Description
Fwd Packet Length Max	Taille maximale des paquets dans le sens direct (sortant)
Fwd Packet Length Min	Taille minimale des paquets dans le sens direct
Min Packet Length	Longueur minimale d'un paquet
Max Packet Length	Longueur maximale d'un paquet
Average Packet Size	Taille moyenne d'un paquet
FWD Packets/s	Nombre de paquets transmis par seconde
Fwd Header Length	Longueur de l'en-tête d'un paquet transféré
Fwd Header Length 1	Nombre d'octets dans un en-tête dans le sens direct
Min_Seg_Size_Forward	Taille minimale du segment dans le sens de la marche
Total Length of Fwd Packet	Taille des paquets dans le sens direct
Fwd Packet Length Std	Écart-type d'un paquet dans le sens direct.
Flow IAT Min	Temps minimum entre deux paquets dans le flux
Subflow Fwd Bytes	Nombre moyen d'octets dans un sous-flux dans le sens direct
Destination Port	Adresse pour recevoir les paquets TCP ou UDP
Protocol	TCP ou UDP pour la transmission des données
Packet Length Std	Écart-type de la longueur des paquets
Flow Duration	Durée de l'écoulement en µs
Fwd IAT Total	Temps total entre deux paquets dans le sens direct.
ACK Flag Count	Nombre de paquets avec ACK
Init_Win_Bytes_Forward	Nombre d'octets dans la fenêtre initiale dans la direction avant
Flow IAT Mean	Temps moyen entre deux paquets dans le flux
Flow IAT Max	Temps maximum entre deux paquets dans le flux
Fwd IAT Mean	Temps moyen entre deux paquets dans le sens direct
Fwd IAT Max	Temps maximum entre deux paquets dans le sens de l'avance

Tableau 12 : fonctionnalités de trafic réseau avec la description [72].

L'ensemble de données utilisé au cours de cette recherche est accessible à partir de : https://www.unb.ca/cic/datasets/ddos-2019.html

4. Approche détaillée

Dans cette approche de l'élaboration de notre problème, nous présentons la description et le détail de toutes les étapes de la réalisation de notre approche

On explique La méthode proposée évalue par un ensemble de données CICDDoS2019, qui sont prétraités pour convenir à l'application des techniques d'apprentissage automatique.

IDS traite une énorme quantité de données, qui contient non pertinent et redondant fonctionnalités. Les fonctionnalités étrangères peuvent rendre plus difficile la détection de comportements suspects, entraînant un processus de formation et de test lent, une consommation de ressources plus élevée et un faible taux de détection.

La sélection des fonctionnalités est l'un des sujets clés d'IDS, elle améliore les performances de classification en recherchant le sous-ensemble de fonctionnalités, qui classe le mieux les données d'entraînement. La suppression de ces fonctionnalités redondantes ou non pertinentes est très importante.

L'apprentissage par arbre de décision (DT) est une méthode en apprentissage automatique. Son but est de créer un modèle qui prédit la valeur d'une variable-cible depuis la valeur de plusieurs variables d'entrée.

DT est appliqué pour la réduction de dimension de caractéristique. Pour CICDDoS2019 ensemble de données.

Une autre des fonctions de sélection des caractéristiques de sklearn, SelectKBest.

SelectKBest fonctionne en conservant les k premières caractéristiques de X avec les scores les plus élevés.

Nous évaluons l'ensemble de données CICDDoS2019 et nous proposons IDS basé sur DT, Voir la figure suivante représentant la mise en œuvre de l'algorithme DT :

Notre objectif est assuré une amélioration des mesures de performance de Accuracy. Pour cela on a réalisé deux expériences en base sur le classifieur arbre de décision.

Expérience 1 :

En applique le classifieur arbre de décision sur une data. Le résultat que nous obtenons est 96 %.

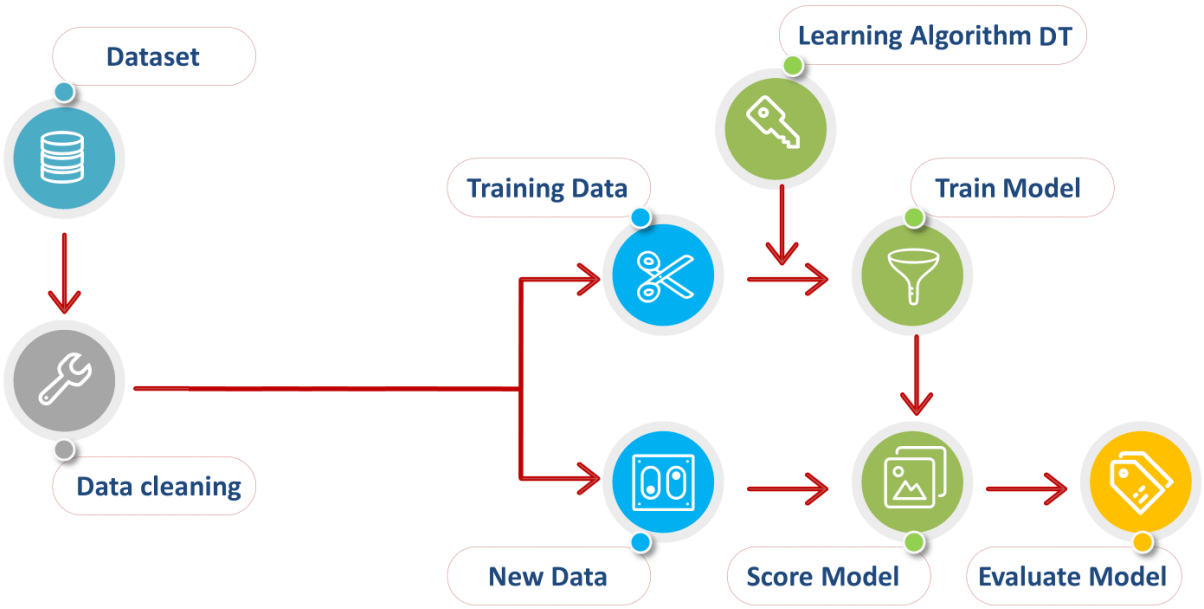

Figure 24 : Schéma de la méthode de la conception

Dans cette expérience, nous utilisant notre Dataset complète, alors nous utilisant aussi la technique **« Train_teste_Split() »** pour faire la division de la dataset, après nous donne le résultat obtenu pour l'algorithme da classification **« DT »** et voici le résultat final obtenu :

	Accuracy	Recall	f1-Scoor
Expérience 1	96 %	96 %	1.00

Tableau 13 : Résultat obtenu pour l'expérience 1

Expérience 2 :

Pour notre model, Pour avoir des meilleurs résultats que la première expérience, on a pensé la selection des meilleures fonctionnalités, pour cela on a choisi l'algorithme SelectKBEST avec le classifieur de l'arbre de décision.

Après 30 jours d'exécutions, on n'a pas u avoir des résultats souhaites

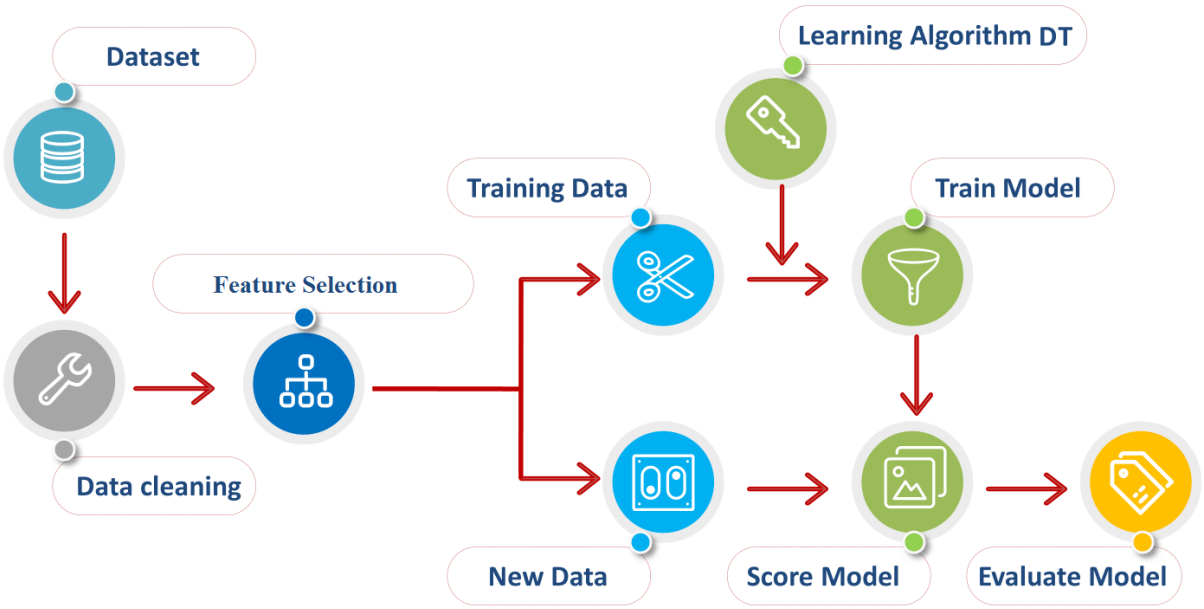

Figure 25 : Schéma de la méthode de la conception

Chapitre 04

Implémentations

1. Les outils de développement

1.1 Définition du langage Python en informatique

Python est un langage de programmation général et de haut niveau. Il est gratuit, open source et multiplateforme. Il prend également en charge les types de données tels que les valeurs numériques, les chaînes, les listes, les n-uplets et les dictionnaires. De plus, c'est un langage basé sur interprète. L'interprète lit le code source ligne par ligne. Par conséquent, il s'agit d'un langage lent à comparer aux langages basés sur le compilateur tels que C, C++. La syntaxe de ce langage est simple et facile à apprendre. De plus, Python supporte des bases de données telles que MySQL, MSSQL. Globalement, Python est un langage polyvalent qui permet de créer diverses applications. Il est populaire parmi les débutants et les développeurs. [73]

Figure 26 : Logo Python

1.2 Définition de l'anaconda

Anaconda est une plate-forme informatique gratuite. Il est possible de l'installer en fonction du système d'exploitation Windows, Linux, MacOs. Il se compose des distributions Python et R et du gestionnaire de paquets appelé conda. Anaconda fournit un ensemble de bibliothèques et de packages préinstallés. Certains d'entre eux sont NumPy, SciPy, Pandas, Scikit Learn, Nltk et Jupiter.

Figure 27 : Logo Anaconda

1.3 Définition jupyter

Jupyter est une application web utilisée pour programmer dans plus de 40 langages de

programmation, dont Python, Julia, Ruby, R, ou encore Scala. C'est un projet communautaire dont l'objectif est de développer des logiciels libres, des formats ouverts et des services pour l'informatique interactive. Jupyter est une évolution du projet IPython. Jupyter permet de réaliser des calepins ou notebooks, c'est-à-dire des programmes contenant à la fois du texte en mark down et du code en Julia, Python, R... Ces calepins sont utilisés en science des données pour explorer et analyser des données. [74]

Figure 28 : Logo Jupyter

2. Bibliothèques essentielles pour l'apprentissage automatique en Python

Bibliothèque de programmes ou bien librairie logicielle est un ensemble de fonctions utilitaires, regroupées et mises à disposition afin de pouvoir être utilisées sans avoir à les réécrire. Les fonctions sont regroupées de par leur appartenance à un même domaine conceptuel (mathématique, graphique, tris, etc.) [75]. La bibliothèque standard de Python est très grande, elle offre un large éventail d'outils [76]. Dans ce qui suit, nous allons définir les bibliothèques utilisées dans notre implémentation :

Pandas

Bibliothèque open source, sous licence BSD (Berkeley Software Distribution), qui fournit des structures de données et des outils d'analyse de données performants et faciles à utiliser pour le langage de programmation Python [77].

Matplotlib

Bibliothèque complète pour la création de visualisations statiques, animées et interactives en Python [78].

Seaborn

Bibliothèque de visualisation de données en Python basée sur matplotlib. Elle fournit une interface de haut niveau pour dessiner des graphiques statistiques attrayants et informatifs [79].

NumPy

Paquet fondamental pour le calcul scientifique en Python. C'est une bibliothèque Python qui fournit un objet de tableau multidimensionnel, divers objets dérivés (tels que des tableaux et

des matrices masqués), et un assortiment de routines pour des opérations rapides sur des tableaux, y compris des opérations mathématiques, logiques, de manipulation de formes, de tri, de sélection, d'entrées/sorties, de transformées de fourrier discrètes, d'algèbre linéaire de base, d'opérations statistiques de base, de simulation aléatoire et bien plus encore [80].

Sklearn

Un module Python intégrant des algorithmes classiques d'apprentissage machine dans le monde étroitement lié des paquets scientifiques Python (numpy, scipy, matplotlib) [81].

3. Étapes du prétraitement des données

3.1 Importation des bibliothèques requises

Nous devrons importer Numpy et Pandas. Numpy est une bibliothèque qui contient des fonctions mathématiques et est utilisée pour le calcul scientifique tandis que Pandas est utilisé pour importer et gérer les ensembles de données.

Ici, nous importons les bibliothèques pandas et Numpy et attribuons respectivement un raccourci "pd" et "np".

```
# packages to import
import pandas as pd
import numpy as np
import glob
import joblib
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler ,MinMaxScaler
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split,cross_val_score
from sklearn.utils.multiclass import unique_labels
from os.path import join
from sklearn .preprocessing import LabelEncoder
from sklearn.feature_selection import SelectKBest, chi2
from sklearn import model_selection
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix
import seaborn as sns
from sklearn import metrics
```

Figure 29 : Importation des bibliothèques

3.2 Importation de l'ensemble de données

Les ensembles de données sont disponibles au format .csv. Un fichier CSV stocke les données tabulaires en texte brut. Chaque ligne du fichier est un enregistrement de données. Nous utilisons la méthode read_data de la bibliothèque pandas pour lire un fichier CSV local en tant que **dataframe**.

```
def read_data(dataroot,file_ending='*.csv'):
        if file_ending==None:
            exit()
        print(join(dataroot,file_ending))
        filenames = [i for i in glob.glob(join(dataroot,file_ending))]
        combined_csv = pd.concat([pd.read_csv(f,dtype=object) for f in filenames],sort=False)
        return combined_csv
dataset=read_data('/Users/toshiba/data/',file_ending='*.csv')
dataset
```

Figure 30 : importation l'ensemble de données

3.3 Nettoyage des données

```
dataset = data.drop_duplicates()
print('les lignes repeter supprimees')
for i in range(0,72):
    colm= data.columns.values[i]
    data=data.replace(colm,np.nan)
data.fillna(data.mean(),inplace= True)
print('les valeurs est charger')
```

Figure 31 : nettoyer les données

Notre type de dataset définie Object nous utilisant le command ci-dessus qui faire la conversion Object ver numérique

```
dataset=data.astype(object).apply(pd.to_numeric)
print('la conversion numerique est faite')
```

Figure 32 : convertir le type de donnes

3.4 Normalisation des données

La plupart du temps, en machine Learning, les Data Set proviennent avec des ordres de grandeurs différents. Cette différence d'échelle peut conduire à des performances moindres. Pour pallier cela, des traitements préparatoires sur les données existent. Pour ramener nos variables au même ordre de grandeur, nous appliquerons un procédé qui s'appelle : features scaling. Le package sklearn. Preprocessing propose la classe MinMaxdardScaler où peut- être appliqué quand les données varient dans des échelles différentes. A l'issue de cette transformation, les features seront comprises dans un intervalle fixe [0,1] La normalisation peut- être effectuée par la technique du **Min-Max Scaling.** La transformation se fait grâce à la formule suivante

$$X_{normalise} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Avec :

- $Xmin$: la plus petite valeur observée pour la feature X
- $Xmax$: la plus grande valeur observée pour la feature X

▪ X : La valeur de la feature qu'on cherche à normaliser

```
Entrée [223]: ##cette fonction pour Normaliser les X
              def Normaliser(data):
                  x = data
                  sc = MinMaxScaler()
                  sc.fit(x)
                  newdata = sc.transform(x)
                  print(newdata)
                  return newdata

Entrée [224]: newdata =Normaliser(X)
```

Figure 33 : normalisation des données

3.5 Mise à l'échelle des fonctionnalités

La plupart des algorithmes d'apprentissage automatique utilisent la **distance euclidienne** entre deux points de données dans leurs calculs. Pour cette raison, **les entités de magnitude élevée pèseront plus** dans les calculs de distance **que les entités de faible magnitude**. Pour éviter cela, la normalisation des fonctionnalités ou la normalisation du score Z est utilisée. Ceci est fait en utilisant la classe "StandardScaler" de "sklearn.preprocessing".

```
de sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
```

```
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
```

Figure 34 : mise à l'échelle des fonctionnalités

4. Définir le model

Avant diviser les données en des ensembles d'apprentissage et de test. On applique des techniques de feature sélection pour améliorer la performance d'Accuracy, 80% des données de chacun des mélanges sont utilisés pour entrainer le classificateur sur la relation contrainte déformation réelle, puis 20% des données ont été utilisés pour valider le model.

```
Entrée [332]: #use SelectKBest to feature selection
              X_new = SelectKBest(chi2, k=20).fit_transform(newdata, y)
              X_new.shape

Out[332]: (191694, 20)
```

Figure 35 : feature selection techniques SelectKBest

SelectKBest : Il conserve les fonctionnalités de notation top-k.

On utiliser la fonction train_test_split pour effectuer la séparation des donnes on donne dataset normaliser complet. Le test_size = 0,2 à l'intérieur de la fonction indique le

pourcentage des données qui doivent être conservées pour le test.

```
#train and test with complte data
X_train, X_test , y_train , y_test = train_test_split(newdata, y, test_size=0.20, random_state=42)
```

Figure 36 : diviser des données

Maintenant, pour construire nos ensembles d'entraînement et de test, nous allons créer 4 ensembles —

1. **X_train** (partie formation de la matrice de caractéristiques),
2. **X_test** (partie test de la matrice de fonctionnalités),
3. **Y_train** (formation d'une partie des variables dépendantes associées aux X trains, et donc aussi des mêmes indices),
4. **Y_test** (tester une partie des variables dépendantes associées aux X ensembles de test, et donc également les mêmes indices).

Nous leur attribuerons le test_train_split, qui prend les paramètres - tableaux (X et Y), test_size (Spécifie le rapport dans lequel diviser l'ensemble de données).

5. Arbre de décision

En applique classifieur arbre de décision avec ensemble de donner qui est réduit Et le résultat que nous obtenons est 96% donc on a une amélioration pour notre model.

```
DecisionTreeClassifierModel = DecisionTreeClassifier(criterion='gini',max_depth=3,random_state=33)
#criterion can be entropy
DecisionTreeClassifierModel.fit(X_train, y_train)
```

Figure 37 : Application le classifieur arbre de décision.

```
#Calculating Prediction
y_pred = DecisionTreeClassifierModel.predict(X_test)
y_pred_prob = DecisionTreeClassifierModel.predict_proba(X_test)
```

Figure 38 : Classification du données de test

Après la prédiction on calculer le taux de précision avec la métrique accuracy, notre modèle à atteindre vers 96 % de taux de précision dans la classification de l'ensemble de données.

```
# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test_sp, y_pred_sp))

Accuracy: 0.9614467887036844
```

Figure 39 : calcul de la précision

Nous pouvons voir que le modèle à de meilleures performances sur l'ensemble de données d'apprentissage et l'ensemble de données de test.

```
Entrée [350]: print('Model test Score: %.3f ' %DecisionTreeClassifierModel.score(X_test, y_test),
                    'Model training Score: %.3f ' %DecisionTreeClassifierModel.score(X_train, y_train))

              Model test Score: 0.961,  Model training Score: 0.961
```

Figure 40 : évaluation de model

5.1 Matrice de confusion

Également connue sous le nom de matrice d'erreur, est une disposition de tableau spécifique permettant de visualiser les performances d'un algorithme.

```
DecisionTreeClassifierModel Train Score is :  1.0
DecisionTreeClassifierModel Test Score is :  1.0
Predicted Value for DecisionTreeClassifierModel is :  [1 1 1 0 1 1 1 1 1 1]
Prediction Probabilities Value for DecisionTreeClassifierModel is :  [[0. 1.]
 [0. 1.]
 [0. 1.]
 [1. 0.]
 [0. 1.]
 [0. 1.]
 [0. 1.]
 [0. 1.]
 [0. 1.]
 [0. 1.]]
Confusion Matrix is :
 [[  930     0]
 [    0 37409]]
```

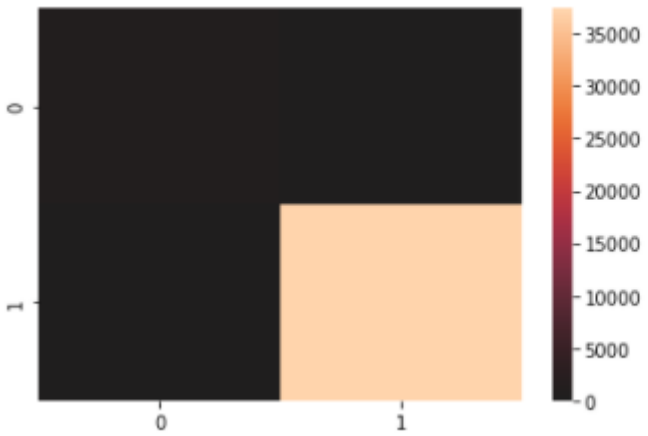

Figure 41 : MATRICE DE CONFUSION

6. Conclusion

Dans ce chapitre, nous avons d'abord présenté les différents outils et langages que nous avons utilisés pour implémenter notre modèle.

Initialement, nous avons prétraité les données utilisées CICDDoS2019 afin de préparer des essais fiables et des données de tests randomisés indépendants. Parmi les techniques de classement.

Dans la première étape, il s'agissait d'appliquer l'algorithme d'arbre de décision et le résultat obtenu est de 96%.

Quant à la deuxième étape, nous avons ajouté la fonctionnalité de sélection technique à l'aide de selectKBest

Malheureusement après 30 jours d'exécution, nous n'avons pas obtenu les résultats souhaités car à ce stade nous avons trouvé cela difficile à mettre en œuvre en raison des caractéristiques de l'ordinateur utilisé.

1. Conclusion générale

L'amélioration des systèmes de détection d'intrusions existants conduit sur une réflexion des techniques d'apprentissage automatique.

Pour cela, nous avons sélectionné les techniques les plus adaptées à notre choix technique, et avons principalement appliqué les techniques SelectKBest et DT pour scruter les informations en profondeur de ce processus.

La combinaison du SelectKBest + DT n'a pas pu avoir des bons résultats à cause des performances du PC par un plantage du RAM.

Notre projet est l'occasion d'approfondir nos connaissances dans le domaine du machine learning et d'apprendre ses différents modèles et leurs applications.

Pour nous, il est important de dire que l'un des principaux avantages de ce travail est de se familiariser avec la compréhension des articles et de maîtriser plusieurs bibliothèques que nous avons vues, et de les utiliser pour créer des modèle

1. Algorithme PCA :

1: **Input**: a D-dimensional training set $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ and the new (lower) dimensionality d (with $d \leq D$)
2: Compute the mean $\bar{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i$
3: Compute the covariance matrix $\mathrm{Cov}(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)^T$
4: Find the spectral decomposition of $\mathrm{Cov}(\boldsymbol{x})$, obtaining the eigenvectors $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_D$ and their corresponding eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_D$. Note that the eigenvalues are sorted, such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D \geq 0$
5: For any $\boldsymbol{x} \in \mathbb{R}^D$, its new lower dimensional representation is:

$$\boldsymbol{y} = \left(\boldsymbol{\xi}_1^T(\boldsymbol{x} - \bar{\boldsymbol{x}}), \boldsymbol{\xi}_2^T(\boldsymbol{x} - \bar{\boldsymbol{x}}), \ldots, \boldsymbol{\xi}_d^T(\boldsymbol{x} - \bar{\boldsymbol{x}})\right)^T \in \mathbb{R}^d,$$

and the original $\boldsymbol{x}$ can be approximated as

$$\boldsymbol{x} \approx \bar{\boldsymbol{x}} + (\boldsymbol{\xi}_1^T(\boldsymbol{x} - \bar{\boldsymbol{x}}))\boldsymbol{\xi}_1 + (\boldsymbol{\xi}_2^T(\boldsymbol{x} - \bar{\boldsymbol{x}}))\boldsymbol{\xi}_2 + \cdots + (\boldsymbol{\xi}_d^T(\boldsymbol{x} - \bar{\boldsymbol{x}}))\boldsymbol{\xi}_d$$

2. PSO

```
from datetime import datetime as dt
import time
import pyswarms as ps
from pyswarms.utils.functions import single_obj as fx
from pyswarms.utils.plotters.plotters import  plot_cost_history
start = dt.now()
print("Started at: ", str(start))
particleScore = list()
particleSize = list()
#mySubsets = list()
# Initialize swarm, arbitrary
options = {'c1': 2, 'c2': 2, 'w':0.3, 'k': 20, 'p':2}
# Call instance of PSO
dimensions = newdata.shape[1] # dimensions should be the number of features
optimizer = ps.single.GlobalBestPSO(n_particles=20, dimensions=dimensions, options=options)
optimizer.optimize(fx.sphere, iters=100)
plot_cost_history (optimizer.cost_history)
plt.show()
end = dt.now()
print("Finished at: ", str(end))
total = end-start
print("Total time spent: ", total)
```

3. Algorithme arbre de décision

Algorithme d'apprentissage par arbre de décision

Donnée : un échantillon *S* de *m* enregistrements classés (*x*, *c*(*x*))
Initialisation : *A* := arbre vide ;
Nœud _courant := racine ;
Echantillon_courant := *S*
Répéter
Décider si le nœud courant est terminal
Si (Nœud _courant est terminal) **alors**
Etiqueter le nœud courant par une feuille
Sinon
Sélectionner un test :
Créer les fils
Définir les échantillons sortants du nœud
Finsi
Nœud _courant := un nœud non encore étudie de *A*
Echantillon_courant := échantillon atteignant Nœud _courant
Jusque (Nœud _courant = ø)
Elaguer l'arbre de décision *A* obtenu
Sortie : l'arbre A élagué

4. Formule Générale (DT)

```
#Import Libraries
from sklearn.tree import DecisionTreeClassifier
sklearn.tree.DecisionTreeClassifier(criterion='gini', splitter='best',
max_depth=None,min_samples_split=2,

min_samples_leaf=1,min_weight_fraction_leaf=0.0,max_features=None,
                    random_state=None,
max_leaf_nodes=None,min_impurity_decrease=0.0,
                    min_impurity_split=None, class_weight=None,presort=False)
DecisionTreeClassifierModel =
DecisionTreeClassifier(criterion='gini',max_depth=3,random_state=33) #criterion can be
entropy
DecisionTreeClassifierModel.fit(X_train, y_train)
#Calculating Details
print('DecisionTreeClassifierModel Train Score is : ' ,
DecisionTreeClassifierModel.score(X_train, y_train))
print('DecisionTreeClassifierModel Test Score is : ' ,
DecisionTreeClassifierModel.score(X_test, y_test))
```

```
print('DecisionTreeClassifierModel Classes are : ' , DecisionTreeClassifierModel.classes_)
print('DecisionTreeClassifierModel feature importances are : ' ,
DecisionTreeClassifierModel.feature_importances_)
#Calculating Prediction
y_pred = DecisionTreeClassifierModel.predict(X_test)
y_pred_prob = DecisionTreeClassifierModel.predict_proba(X_test)
print('Predicted Value for DecisionTreeClassifierModel is : ' , y_pred[:10])
print('Prediction Probabilities Value for DecisionTreeClassifierModel is : ' ,
y_pred_prob[:10])
```

5. Algorithme SelectkBest

```
#Import Libraries
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2 , f_classif
#Feature Selection by KBest
#print('Original X Shape is ' , X.shape)
FeatureSelection = SelectKBest(score_func= chi2 ,k=3) # score_func can = f_classif
X = FeatureSelection.fit_transform(X, y)
#showing X Dimension
#print('X Shape is ' , X.shape)
#print('Selected Features are : ' , FeatureSelection.get_support())
```

Bibliographie

[1] M. &. M. Zamani, «Machine learning techniques for intrusion detection,» arXiv preprint arXiv:13 12.2177., 2013.

[2] A. K. J. W. S. D. &. S. A. GulzanEsther, «Efficient intrusion detection using machine learning tec hniques».

[3] [En ligne]. Available: http://mrproof.blogspot.com/2010/11/les-systemes-de-detectionsdintrusions

[4] D. Imane, «Etude et mise en place d'un système de détection/prévention d'intrusion (ids/ips) résea
u etude de cas snort,» 2013-2014.

[5] C. Soumia, *Système de détection d'intrusion basé sur la,* UNIVERSITE DE M'SILA Départemen t d'Informatique, 28/06/2012.

[6] J. Timmis, «Artificial immune systems,» A novel data analysis technique inspired by the immune network theory, 1999.

[7] H. Debar, «a revised taxonomy for intrusion detectionsystems,» Wespi, 1999.

[8] Cisco: the Science of Intrusion Detection System Attack Identification, Cisco, January 2003.

[9] S. Axelsson, The Base Rate Fallacy and its Implications for the Difficulty of Intrusion Detection, 1999.

[10] Cisco Security Professional's Guide to Secure Intrusion Detection Systems.

[11] [En ligne]. Available: https://ebrary.net/26721/computer_science/use_ids.

[12] D. P. B. B. H. P. A. P. a. M. R. Chirag Modi, «A survey of intrusion detection techniques in cloud ,» Journal of Network and Computer Applications, 2013.

[13] T. Abbes, *Classification du trafic et optimisation des règles de filtrage pour la détection d'intrusi ons,* l'université Henri Poincaré.Nancy1, 2004.

[14] yohan-c, «https://datascientest.com/,» 16 02 2021. [En ligne]. Available: https://datascientest.com /danielcomment-lire-et-exploiter-une-matrice-de-confusion. [Accès le 17 09 2021].

[15] D. L. Lough, A Taxonomy of Computer Attacks with Applications to Wireless Networks, Virgini a Polytechnic Institute and State University: PhD thesis, 2001.

[16] [En ligne]. Available: https://experiences.microsoft.fr/articles/intelligence-artificielle/comprendre -utiliser-intelligence-artificielle/.

[17] [En ligne]. Available: https://www.natsystem.fr/comment-integrer-lia-dans-vos-projets.

[18] [En ligne]. Available: https://www.talend.com/fr/resources/what-is-machine-learning/#:~:text=Le%20premier%20r%C3%A9seau%20neuronal%20artificiel,images%20%C2%AB%20Mark%201%20Perceptron%20%C2%BB..

[19] [En ligne]. Available: https://digitalinsiders.feelandclic.com/machine-learning-definition.

[20] [En ligne]. Available: https://blog.bismart.com/en/difference-between-machine-learning-deep-lear ning. [Accès le 17 09 2021].

[21] «What's the difference between artificial intelligence, machine learning and deep learning?».

[22] Z. ISMAILI. [En ligne]. Available: https://analyticsinsights.io/apprentissage-supervise-vs-non-supervise/.

[23] H. Issarane, «https://analyticsinsights.io/,» [En ligne]. Available: https://analyticsinsights.io/5-apprentissage-supervise/.

[24] H. a. L. S. Mathian, «Les méthodes de classification de données spatiales,» 2006.

[25] G. Bonnardot, LUNDI 6 NOVEMBRE 2017. [En ligne]. Available: https://datakeen.co/8-machine -learning-algorithms-explained-in-human-language/.

[26] «https://dataanalyticspost.com/,» [En ligne]. Available: https://dataanalyticspost.com/Lexique/svm

[27] «https://www.talend.com/fr/,» [En ligne]. Available: https://www.talend.com/fr/resources/what-is-machine-learning/.

[28] N. V. S. PRABHU, «Data Mining and Warehousing,» New Age International (P) Ltd, New Delhi, 2007.

[29] N. S. Chauhan, «Algorithme Naïve Bayes : tout ce que vous devez savoir,» 2020 » juin.

[30] 05 01 2018. [En ligne]. Available: https://invenis.co/blog/3-algorithmes-de-machine-learning-bien-utiles-business/.

[31] [En ligne]. Available: https://www.javatpoint.com/principal-component-analysis.

[32] B. Fertil, «Reconnaissance des formes : Classement d'ensembles d'objets,» 2006.

[33] [En ligne]. Available: http://tpe-intelligence-artificielle-2013.e-monsite.com/pages/definition-de-l-intelligenceartificielle.html.

[34] C. T. Quang, «Classification automatique des textes vietnamiens Hanoi, Institut de la Francophonie pour l'informatique,» 2005.

[35] S.-T. John, «Support Vector Machines and other kemel-based leaming methods,» Cambridge University Press, 2000.

[36] H. a. B. F. Mohamadally, «SVM : Machines à vecteurs de support ou séparateurs à vastes marges, » Versailles St Quentin, Versailles St Quentin, 2006.

[37] [En ligne]. Available: https://www.supinfo.com/articles/single/8381-reseaux-quand-ips-ids-smele?fbclid=IwAR1DL5ATCcrakiGIEFWswFh9AXMSqb0SRKVXXXkcjkKtj6dQQ_0lRhusbE.html.

[38] [En ligne]. Available: https://www.oracle.com/fr/cloud/deep-learning-intelligence-artificielle.html.

[39] J. Graham-Cumming, «Interview de John Graham-Cumming, l'auteur du logiciel,» 2006.

[40] L. Denoue, «Classification supervisée de documents,» 2003.

[41] K. Zeitouni, «Analyse et extraction de connaissances des bases de données,» 2006.

[42] [En ligne]. Available: https://digitalinsiders.feelandclic.com/construire/definition-quest-machine-learning.

[43] T. Stéphane, «Data Mining et statistique décisionnelle: L'intelligence des données».

[44] [En ligne]. Available: https://www.stat4decision.com/fr/voila-dashboards-a-partir-de-vos-jupyter-notebooks/.

[45] h. Hilali, «application de la classification textuelle pour l'extraction des règles d'association maximales,» université du québec à trois-rivières, trois-rivières, 2009.

[46] [En ligne]. Available: https://ichi.pro/fr/selection-des-fonctionnalites-avec-une-approche-pratique-21896594170991. [Accès le 17 08 2021].

[47] [En ligne]. Available: https://docs.microsoft.com/fr-fr/analysis-services/data-mining/feature-selection-data-mining?view=asallproducts-allversions. [Accès le 17 08 2021].

[48] N. Shah. [En ligne]. Available: https://medium.datadriveninvestor.com/feature-selection-techniques-1a99e61da222. [Accès le 01 10 2021].

[49] M. Mitchell, «Une introduction aux algorithmes génétiques,» Cambridge, MA : MIT Press, Cambridge, 1996.

[50] F. J. K. V. e. K. J. Ferri, «Feature subset search using genetic algorithms In Workshop on Natural Algorithms in Signal Processing,» IEE. Press, 1993.

[51] M. e. S. J. Kudo, «Comparison of algorithms that select features for pattern classi,» Pattern Recognition, 2000.

[52] J. Kennedy, «Swarm intelligence,» Handbook of nature-inspired and innovative computing ,Springer, Boston, MA, Springer, Boston, MA, 2006.

[53] L. e. a. Brezočnik, «"Swarm Intelligence Algorithms for Feature Selection: A Review,» 2018.

[54] K. D, «An idea based on honey bee swarm for numerical optimization,» Erciyes University, 2005.

[55] K. D, «Artificial bee colony algorithm,» Scholarpedia, 2010.

[56] B. G. C. O. N. K. D Karaboga, «A comprehensive survey:artificial bee colony (ABC) algorithm and applications,» Artif. Int. Rev., 2012.

[57] [En ligne]. Available: https://www.unb.ca/cic/datasets/ddos-2019.html.

[58] O. C. Ibe, Fundamentals of Data Communication Networks, USA: Hoboken, NJ, USA: Wiley, Nov 2019.

[59] Cloudflare, «What is a distributed denial-of-service (DDoS) attack? [Online]».

[60] FortiGuard, TFTP server buffer overflow.

[61] «The most popular types of DNS attacks».

[62] D. Smith, «Portmapper is preying on misconfigured servers to amplify attacks,» Sep. 2015.

[63] Akamai, «Attackers using new MS SQL reflection techniques,» Feb. 2015.

[64] R. B. M. B. a. A. G. J. M. Alonso, «LDAP injection techniques,» in IEEE Singapore International Conference on Communication Systems,, Guangzhou, China, Nov 2008.

[65] Microsoft, «MS03-034: Flaw in NetBIOS could lead to information disclosure,» Sep 2019. [En ligne]. Available: https://support.microsoft.com/en-us/help/824105/ms03-034-flaw-in-netbios-could-lead-to-information-disclosure.

[66] Cloudflare, «NTP amplification DDoS attack».

[67] M. Majkowski, «Stupidly simple DDoS protocol (SSDP) generates 100 Gbps DDoS,» Jun. 2017.

[68] Imperva, «SNMP reflection/amplification».

[69] S. R. M. S. a. L. T. F. Lau, «"Distributed denial of service attacks,» IEEE International Conference on Systems, Man and Cybernetics,, Nashville, TN, USA, Oct.2000.

[70] A. H. L. S. H. a. A. A. G. I. Sharafaldin, «Developing distributed denial of service (DDoS) attack dataset and taxonomy,",» in International Carnahan Conference on Security Technology, Chennai, India, Oct. 2019,.

[71] JavaPipe, «35 types of DDoSattacks (that hackers will use against you in 2020),,» Jun. 2019.

[72] C. I. f. Cybersecurity, «Network traffic flow analyzer».

[73] «fr.sawakinome.com,» [En ligne]. Available: https://fr.sawakinome.com/articles/programming/difference-between-anaconda-and-python-programming.html.

[74] [En ligne]. Available: https://fr.wikipedia.org/wiki/Jupyter#cite_note-kernels_community-2.

[75] [En ligne]. Available: https ://www.techno-science.net/definition/1470.html.

[76] [En ligne]. Available: https ://docs.python.org/.

[77] [En ligne]. Available: https ://pandas.pydata.org/docs/.

[78] [En ligne]. Available: https ://matplotlib.org/.

[79] [En ligne]. Available: https ://seaborn.pydata.org/.

[80] [En ligne]. Available: https ://numpy.org/doc/stable/user/whatisnumpy.html.

[81] [En ligne]. Available: https ://www.kite.com/python/docs/sklearn.

[82] M. D. a. H. Liu, «Feature selection for classification,» 1997.

[83] P. Mahé, «Noyaux pour graphes et Support Vector Machines pour le criblage virtuel de molécules,» DEA MVA 2002/2003, Septembre 2003.

[84] F. B. Mohamadally Hasan, «SVM machine à vecteurs de support ou séparateur à vaste marge,» BD Web, ISTY3, Versailles St Quentin, France, janvier 2006.

[85] S. B. e. E. Loper, «Natural Language Processing with Python : analyzing text with the natural language toolkit,» O'Reilly Media, Inc, 2009.