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 : Réseau et télécommunication

Par :
- Tahar Mohamed
- Saidi Abdelnaceur Djillali

Sur le thème

Un effectif système de détection d'intrusion pour l'amélioration de la précision

Soutenu publiquement le .. /09 / 2021 à Tiaret devant le jury composé de :

Mr Nassan Samir	Grade Université MAA	Président
Mr Daoud Mohamed Amine	Grade Université MAA	Encadreur
Mr Alem Abdelkader	Grade Université MAA	Examinateur

2020-2021

Remerciement…

En premier lieu, nous remercions ***DIEU*** *Pour le tout puissant , maître des cieux et de la terre,*
Qui nous a éclairé le chemin et permis de mener à bien ce travail.
Nous tenons également à exprimer toute notre reconnaissance à notre Promoteur Monsieur ***DAOUD Mohamed Amine,*** *qui s'est toujours montré disponible et à l'écoute durant toute la réalisation de ce présent mémoire et qui a su guider et structurer nos idées grâce à ses précieux conseils.*
Nos profonds remerciements s'adressent aux membres de jury qui nous font honneur en acceptant d'évaluer notre travail.
Un énorme merci à nos familles et amis pour leurs éternel soutient et la confiance qu'ils ont en nos capacité.
Enfin, nous remercions tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce modeste travail.

Je dédie ce travail à …✍

Ma chère mère et mon père pour l'éducation qui l'on prodigue avec tous les moyens et au prix de tous les sacrifices

Aux personnes dont j'ai bien aimé la présence dans ce jour à tous mes frères et mes sœurs

A ceux qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, qui m'ont accompagné durant mon chemin d'études supérieures.

A tous ceux qui, d'une manière ou d'une autre, ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire

Nasro et Mohamed

Table des matières

Table de figures

Table des tableaux

Résumé :

La tâche principale d'un système de détection d'intrusion (IDS) est de détecter les comportements anormaux à la fois au système et les réseaux, et il y a eu de plus en plus d'études appliquant l'apprentissage automatique dans ce domaine. Les limites de l'utilisation d'un seul classificateur dans la classification du trafic normal et des anomalies (attaques) ont conduit à l'idée de construire des modèles hybrides ou d'ensemble plus compliqués mais offrant une plus grande précision et un taux de fausses alarmes (FAR) plus faible. Le but est d'améliorer les performances de l'IDS en utilisant des méthodes d'ensemble et la sélection de caractéristiques. Les modèles d'ensemble ont été construits sur la base des deux techniques d'ensemble, Bagging et Boosting,. Les modèles proposés ont ensuite été évalués à l'aide des ensembles de données CICIDS-2018. Le choix un modèle d'ensemble d'ensachage avec un classificateur de base pourrait produire des meilleures performances en termes de précision de classification lorsque l'on travaillait avec le sous-ensemble de caractéristiques sélectionnées

تتمثل المهمة الرئيسية لنظام كشف التسلل (IDS) في اكتشاف السلوك غير الطبيعي في كل من النظام والشبكات ، وهناك المزيد والمزيد من الدراسات التي تطبق التعلم الآلي في هذا المجال. أدت حدود استخدام المصنف الفردي في تصنيف حركة المرور العادية والشذوذ (الهجمات) إلى فكرة بناء نماذج هجينة أو مجمعة أكثر تعقيدًا ولكنها توفر دقة أكبر ومعدل أعلى. إنذارات كاذبة أقل (بعيد). الهدف هو تحسين أداء IDS باستخدام طرق التجميع واختيار الميزات. تم بناء نماذج المجموعة على أساس تقنيتي المجموعات ، التعبئة والتعزيز .. ثم تم تقييم النماذج المقترحة باستخدام مجموعات البيانات CICIDS-2018. قد يؤدي اختيار نموذج مجموعة تعبئة مع مصنف أساسي إلى أداء أفضل من حيث دقة التصنيف عند العمل مع مجموعة فرعية من الخصائص المحددة.

The main task of an intrusion detection system (IDS) is to detect abnormal behavior in both the system and the networks, and there have been more and more studies applying machine learning in this area. . The limits of the use of a single classifier in the classification of normal traffic and anomalies (attacks) have led to the idea of building hybrid or ensemble models that are more complicated but offer greater precision and a higher rate. lower false alarms (FAR). The aim is to improve the performance of the IDS using ensemble methods and feature selection. The ensemble models were built on the basis of the two ensemble techniques, Bagging and Boosting ,. The proposed models were then evaluated using the CICIDS-2018 datasets. Choosing a bagging set model with a basic classifier might produce better performance in terms of classification accuracy when working with the subset of selected characteristics.

INTRODUCTION GENERALE

Introduction Générale

Ces dernières années, les activités malveillantes telles que l'accès illégal aux données, l'usurpation d'identité, la modification de données, l'intrusion se sont répandues dans le cybermonde à une cadence alarmante. Les stratégies d'atténuation et de protection de la sécurité devraient être considérées comme obligatoires. Les méthodes et les outils de sécurité existantes ne suffisent pas d'assurer une protection complète contre des cyber-attaques sophistiquées et complexes qui évoluent et rendent le processus de détection plus compliqué et plus difficile.

Un éventuel mécanisme de protection tel que la détection d'intrusion est indispensable car elle s'agit d'une action préventive permettant de se débarrasser de tout acte malin au sein d'un environnement à sécuriser.

Les techniques de Machine Learning apprennent de manière autonome à effectuer une tâche ou à réaliser des prédictions à partir de données et améliorent leurs performances au fil du temps. Les limites de l'utilisation d'un seul classificateur dans la classification du trafic normal et des anomalies (attaques) ont conduit à l'idée de construire des modèles hybrides ou d'ensemble plus compliqués mais offrant une plus grande précision et un taux de fausses alarmes (FAR) plus faible.

Notre travail consiste à proposer une approche proposée pour un IDS en utilisant les algorithmes des machines Learning afin d'améliorer les mesures de performance des IDS, dans notre cas, on s'intéressé a l'Accuracy afin de régler les problèmes tels que le nombre élevé de faux positifs et les faux négatifs.

- ✓ La première partie de ce manuscrit présente un état de l'art qui est composée de deux partie très intéressante celles des systèmes de détection d'intrusion et celle du machine Learning.
- ✓ La deuxième partie concerne la proposition du notre approche basée sur un ensemble des expériences et une sous partie sur l'implémentation et résultats obtenues
- ✓ Enfin, on conclut par une conclusion générale

PARTIE I

Recherche bibliographique (état de l'art)

Chapitre 01

Systèmes de détection d'intrusion

- **Chapitre 01 : les systèmes de détection d'intrusion**

1. Introduction

La sécurité des systèmes informatiques se cantonne généralement à garantir les droits d'accès aux données et ressources d'un système en mettant en place des mécanismes d'authentification et de contrôle permettant d'assurer que les utilisateurs des dites ressources possèdent uniquement les droits qui leur ont été octroyés. **[01]**

Un système de détection d'intrusion (IDS) est une technologie de sécurité qui protège les entreprises contre les cybers attaques en surveillant le trafic réseau pour détecter les activités suspectes et en envoyant des alertes lorsque des failles de sécurité sont identifiées. Les technologies IDS et IPS (Intrusion Prevention System) les plus avancées exploitent l'analyse comportementale en temps réel et le Machine Learning pour détecter les intrusions.**[02]**

2. Définition d'un système de détection d'intrusions

Les IDS sont des outils permettant de détecter les attaques/intrusions du réseau sur lequel il est placé. C'est un outil complémentaire aux firewall, scanneurs de failles et anti-virus.**[3]**

Un système de détection d'intrusions (« Intrusion Détection System » ou IDS) est un appareil ou une application qui alerte l'administrateur en cas de faille de sécurité, de violation de règles ou d'autres problèmes susceptibles de compromettre son réseau informatique.**[4]**

Les systèmes de détection des intrusions (IDS) analysent le trafic réseau pour détecter des signatures correspondant à des cyberattaques connues.**[5]**

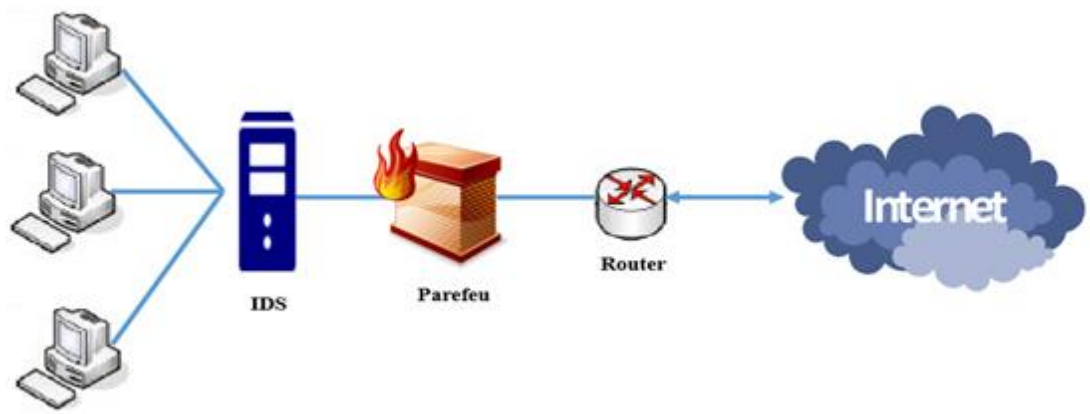

Figure 1 : Système de détection d'intrusions

3. Architecture Intrusion Détection System

Plusieurs schémas ont été proposés pour décrire les composants d'un système de détection d'intrusions. Parmi eux, nous avons retenu celui issu des travaux d'Intrusions Détection exchange format Working Group (IDWG) de l'Internet Engineering Task Force (IETF) comme base de départ, car il résulte d'un large consensus parmi les intervenants du domaine. **[6]**

L'objectif des travaux du groupe IDWG est la définition d'un standard de communication entre certains composants d'un système de détection d'intrusions. La figure (2) illustre ce modèle et permet d'introduire un certain nombre de concepts :

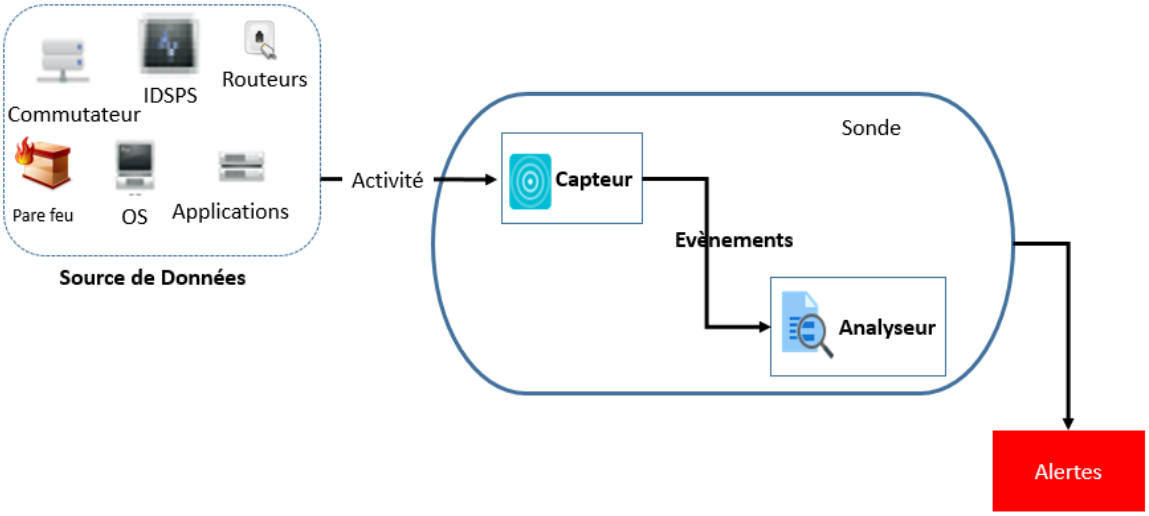

Figure 2 : Modèle générique de la détection d'intrusions proposé par l'IDWG

L'architecture IDWG d'un système de détection d'intrusions contient des capteurs qui envoient des événements à un analyseur. Les capteurs couplés avec un analyseur forment une sonde, cette dernière envoie des alertes qui la notifient à un opérateur humain. Les différents éléments de cette architecture sont :

- ✓ **Source de données :** dispositif générant de l'information sur les activités des entités du système d'information.
- ✓ **Capteur :** génère des événements en filtrant et formatant les données brutes provenant d'une source de données.

- ✓ **Evénement :** message formaté et renvoyé par un capteur. C'est l'unité élémentaire utilisée pour représenter une étape d'un scénario d'attaques connu.
- ✓ **Analyseur :** c'est un outil logiciel qui met en œuvre l'approche choisie pour la détection (comportementale ou par scénarios), il génère des alertes lorsqu'il détecte une intrusion.
- ✓ **Sonde :** un ou des capteurs couplés avec un analyseur.
- ✓ **Alerte :** message formaté émis par un analyseur s'il trouve des activités intrusives dans une source de données.

Dans ce modèle qui représente le processus complet de la détection ainsi que l'acheminement des données au sein d'un IDS. L'administrateur configure les différents composants (capteur(s), analyseurs(s)) selon une politique de sécurité bien définie. Les capteurs accèdent aux données brutes, les filtrent et les formatent pour ne renvoyer que les événements intéressants à un analyseur. Les analyseurs utilisent ces événements pour décider de la présence ou non d'une intrusion et envoient dans le cas échéant une alerte, qui notifie l'opérateur humain, une réaction éventuelle peut être menée automatiquement ou manuellement. **[07]**

4. Classification des systèmes de détections d'intrusion :

La classification adoptée selon différents critères qui ne sont pas forcément mutuellement exclusifs n'est pas, elle présente tour à tour et au même niveau les catégories caractérisant chaque IDS, et utilise les critères suivants (figure 3). **[07]**

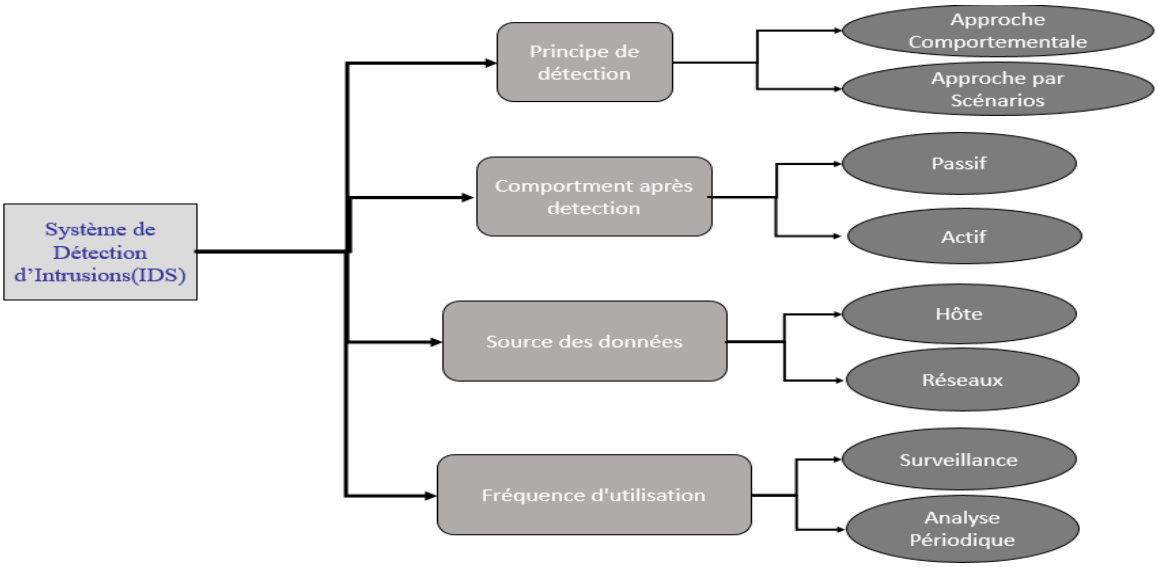

Figure 3 : Taxonomie des systèmes de détection d'intrusions.

La méthode de détection utilisée (principe).

• Le comportement après détection

• La source des données à analyser.

• La fréquence de l'analyse

5. Types de systèmes de détections d'intrusion

a. Système de détection d'intrusion basé sur un hôte

Les IDS systèmes (Host IDS) analysent le fonctionnement et l'état des machines sur lesquels ils sont installés afin de détecter les attaques en se basant sur des démons. L'intégrité des systèmes est alors vérifiée périodiquement et des alertes peuvent être levés. Par nature, ces IDS sont limités et ne peuvent détecter les attaques provenant des couches réseaux (tels que les attaques de type DOS).**[3]**

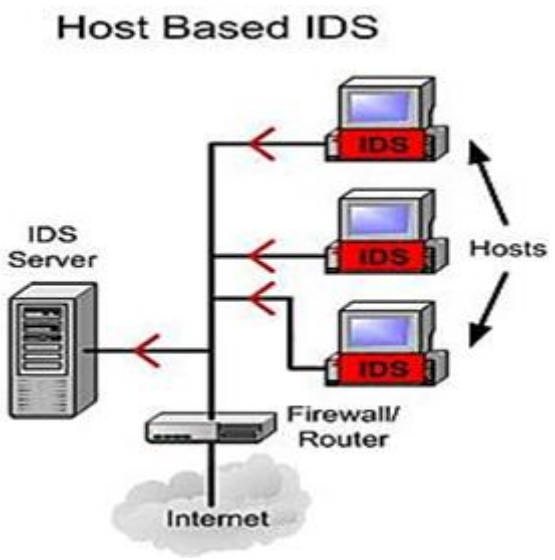

Figure 4 : IDS basé sur un Hôte

i. Avantages de HIDS :

- constate l'impact d'une attaque et donc mieux réagir.

_ observation des activités sur l'hôte avec précision.

_ détection des attaques impossibles à détecter avec des IDS réseau puisque le trafic y est souvent crypté

ii. Inconvénients de HIDS :

_ moins de facilité à détecter les scans.

_ l'analyse des traces d'audit du système est très contraignante en raison de la taille de ces dernières ils consomment beaucoup de ressources CPU.**[8]**

b. Système de détection d'intrusion basé sur le réseau

Les IDS réseaux (Network IDS), quant à eux, analysent en temps réel le trafic qu'ils aspirent à l'aide d'une sonde (carte réseau en mode "promiscuous"). Ensuite, les paquets sont décortiqués puis analysés. En cas, de détection d'intrusion, des alertes peuvent être envoyées.**[3]**

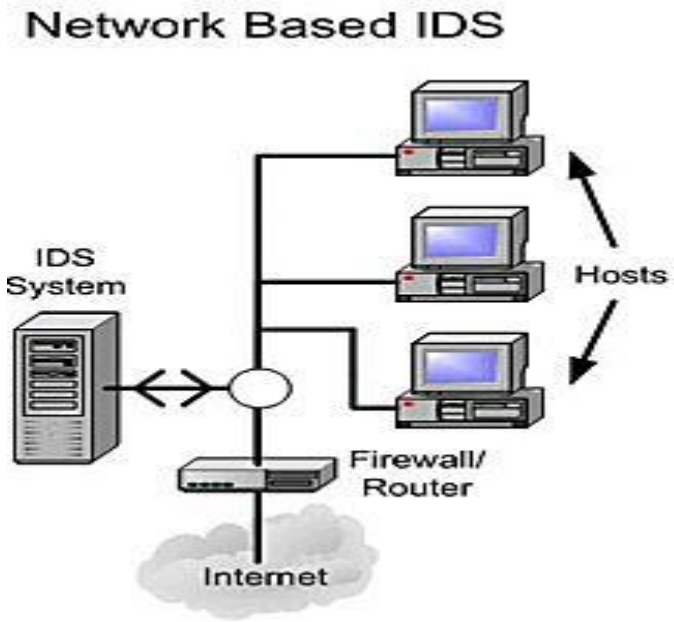

Figure 5 : IDS basé sur un Réseau

i. Avantages de NIDS

_ les capteurs peuvent être bien sécurisés puisqu'ils se "contentent" d'observer le trafic.

_ Détection facile - grâce aux signatures.

ii. Inconvénients de NIDS :

_ la probabilité de faux négatifs (attaques non détectées comme telles) est élevée et il est difficile de contrôler le réseau entier.

_ ils doivent principalement fonctionner de manière cryptée d'où une complication de l'analyse des paquets.

_ à l'opposé des IDS basés sur l'hôte, ils ne voient pas les impacts d'une attaque**.[8]**

6. Les techniques de détection d'IDS

Deux techniques de détection d'intrusion sont généralement mises en œuvre par les IDS courants.

a. Détection par signature (scénario)

Généralement, les IDS réseaux se basent sur un ensemble de signatures qui représentent chacune le profil d'une attaque. Cette approche consiste à rechercher dans l'activité de l'élément surveillé (un flux réseau) les empreintes d'attaques connues, à l'instar des antivirus. Une signature est habituellement définie comme une séquence d'événements et de conditions relatant une tentative d'intrusion. La reconnaissance est alors basée sur le concept de "pattern Matching" (analyse de chaînes de caractères présente dans le paquet, à la recherche de correspondance au sein d'une base de connaissance). Si une attaque est détectée, une alarme peut être remontée (si l'IDS est en mode actif, sinon, il se contente d'archiver l'attaque).**[3]**

i. Avantage de l'approche par scénarios

La prise en compte des comportements exacts des attaquants potentiels est possible.

ii. Inconvénients de l'approche par scénarios

La base de règles doit être bien construite, ce qui est parfois délicat.

Les performances du système expert sont limitées par celles de l'expert humain qui a fourni les règles.**[9]**

b. Détection par comportement

Les IDS comportementaux ont pour principale fonction la détection d'anomalie. Leur déploiement nécessite une phase d'apprentissage pendant laquelle l'outil va apprendre le comportement "normal" des flux applicatifs présents sur son réseau. Ainsi, chaque flux et son comportement habituel doivent être déclarés ; l'IDS se chargera d'émettre une alarme, si un flux anormal est détecté, et ne pourra bien entendu, spécifier la criticité de l'éventuelle attaque.

Les IDS comportementaux sont apparus bien plus tard que les IDS à signature et ne bénéficient pas encore de leur maturité. Ainsi, l'utilisation de tels IDS peut s'avérer délicate dans le sens où les alarmes remontées contiendront une quantité importante de fausses alertes. Ce problème peut être résolus en généralisant la déclaration des flux mais cette opération peut entrainer une transparence de l'IDS face à la détection de certaines attaques.**[3]**

i. Avantage de l'approche comportementale

La détection d'intrusion inconnue est possible.

ii. Inconvénients de l'approche comportementale

Le choix des différents paramètres du modèle statistique est assez délicat et soumis à expérience de l'officier de sécurité.

L'hypothèse d'une distribution normale des différentes mesures n'est pas prouvé que le choix des mesures à retenir pour un système cible donné est délicat.**[9]**

7. Conclusion

Ce chapitre nous a permis de constater que les IDS sont de plus en plus fiables, d'où le fait qu'ils soient souvent intégrés dans les solutions modernes de sécurité. Les avantages qu'ils présentent par rapport aux autres outils de sécurité les favorisent. Il nous a également permis de comprendre que ces derniers sont indispensables aux fournisseurs du Cloud afin d'assurer leur sécurité Cloud. Dans le chapitre qui suit nous présenterons les différentes techniques de l'apprentissage automatique.

Chapiter 02

Machine Learning

Chapitre 02 : Machine Learning

1. Introduction

L'intelligence artificielle a récemment attiré l'attention du monde entier, et elle est devenue le dialogue le plus important sur les tables de discussion sur ce que le monde cherche à réaliser en termes de développement technologique et de progrès sans précédent, et en fait cet intérêt n'a pas été vain. De nombreux modèles sont apparus qui ont confirmé que l'intelligence artificielle était proche de la concurrence avec l'intelligence humaine, et en créant des voitures autonomes, le robot Sophia et bien d'autres, et avec la réussite après le succès, le domaine d'intérêt s'est davantage accru dans ce que l'on appelle l'apprentissage automatique et son développement pour avancer vers les succès

2 Machine Learning

L'apprentissage automatique, appelé ML, le concept d'apprentissage automatique peut être simplifié comme l'une des branches émergeant de la science de l'intelligence artificielle (IA) basée sur la programmation d'ordinateurs de différentes formes pour pouvoir effectuer des tâches et mettre en œuvre les commandes qui leur sont assignées en s'appuyant sur les données dont ils disposent et en les analysant tout en limitant l'intervention humaine. En le dirigeant ou complètement absent. Il est à noter que le terme d'apprentissage automatique est apparu à la demande du pionnier de l'intelligence artificielle Arthur Samuel en 1959 dans le cadre des travaux des laboratoires IBM, et il est à noter que la machine dans ce cas doit s'appuyer sur l'analyse des données saisies à l'avance pour répondre aux commandes et aux tâches qui lui sont demandées, donc l'intervention humaines et très petite **[11]**

La machine aura également la responsabilité de prendre des décisions en cas de besoin et de déterminer quelles tâches doivent être effectuées, quand, comment et pourquoi sans aucune assistance humaine, car cela contribuera inévitablement à l'achèvement des tâches le plus rapidement possible par rapport au temps que les gens consomment pour accomplir les tâches. **[11]**

2.1 Deep Learning

La couche immédiatement inférieure est occupée par le **Deep Learning (DL)**, l'une des nombreuses approches du machine Learning. **[12]** Le Deep Learning ou apprentissage profond c'est une technique de machine Learning reposant sur le modèle des réseaux neurones : des dizaines voire des centaines de couches de neurones sont empilées pour apporter une plus grande complexité à l'établissement des règles **[13]**

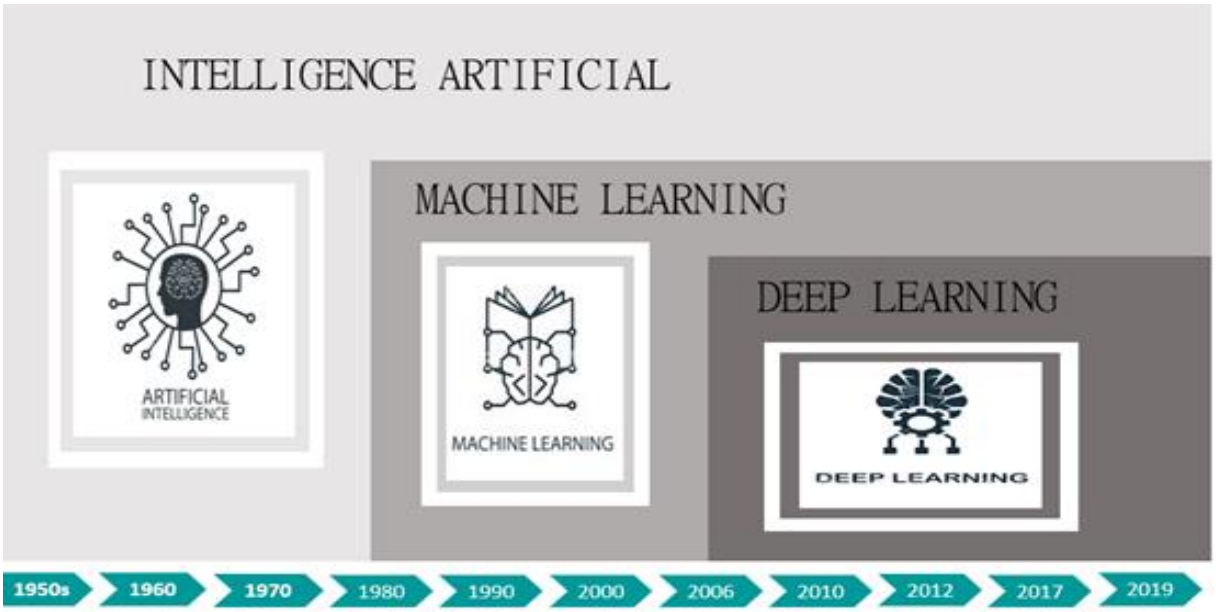

Figure 6 : Intelligence Artificielle

2.2 La relation entre l'intelligence artificielle, apprentissage autonome, données

Les dimensions de la relation entre l'IA, l'apprentissage automatique et l'exploration de données peuvent être dessinées dans 3 parachutes de taille variable; Là où la science de l'intelligence artificielle est le plus grand parapluie qui inclut directement sous elle le parapluie de l'apprentissage automatique, tandis que ce dernier embrasse le parapluie de l'exploration et de l'extraction de données, et ainsi il est conclu que l'intelligence artificielle est au sommet de la pyramide et son rôle est de chercher à programmer des machines et des ordinateurs en s'appuyant sur plusieurs méthodes pour correspondre à l'intelligence humaine**.** En fin de compte, cela dépend de différents styles de prise de décision et de réflexion. Quant au Machine Learning, il représente le rôle de la couche qui suit le sommet de la pyramide, et son rôle est de mettre en œuvre la tâche d'automatisation et de programmation, et d'apprendre aux machines à utiliser les données dont elles disposent pour prendre des décisions, et ici le rôle du Data Mining apparaît en recherchant des données pertinentes et en les employant dans l'exécution de la tâche. **[11]**

2.3 Catégories de Machine Learning

Le Machine Learning n'est pas une nouvelle technologie. Le premier réseau neuronal artificiel, appelé « Perceptron », a été inventé en 1958 par le psychologue américain Frank Rosenblatt. **[12]**

Au départ, Perceptron devait être une machine, et non un algorithme. En 1960, il a été utilisé pour le développement de la machine de reconnaissance d'images « Mark 1 Perceptron ». Mark 1 Perceptron a été le premier ordinateur à utiliser des réseaux neuronaux artificiels (ANN) pour simuler la réflexion humaine et apprendre par essais et erreurs

Le Machine Learning est de plus en plus utilisé en raison de la multiplication des bibliothèques et des Framework open source et de la multiplication par plusieurs milliards de fois de la puissance de traitement des ordinateurs entre 1956 et 2018. Aujourd'hui, le machine Learning est partout : des transactions boursières à la protection contre les logiciels malveillants en passant par la personnalisation du marketing. **[12]**

Quelle que soit sa simplicité ou sa complexité, le Machine Learning peut être classé en grandes catégories :

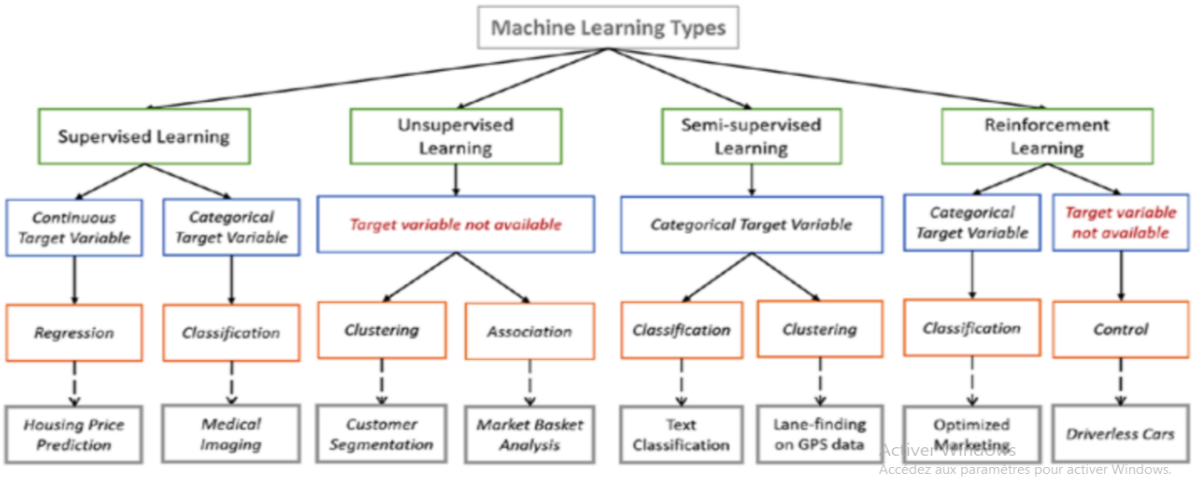

Figure 7 : Catégories de Machine Learning

2.3.1 Machine Learning avec supervision

Le machine Learning avec supervision est une technologie élémentaire mais stricte. Les opérateurs présentent à l'ordinateur des exemples d'entrées et les sorties souhaitées, et l'ordinateur recherche des solutions pour obtenir ces sorties en fonction de ces entrées. Le but recherché est que l'ordinateur apprenne la règle générale qui mappe les entrées et les sorties.**[12]**

Le machine Learning avec supervision peut être utilisé pour faire des prédictions sur des données indisponibles ou futures on parle de « modélisation prédictive ». L'algorithme essaie de développer une fonction qui prédit avec précision la sortie à partir des variables d'entrée – par exemple, prédire la valeur d'un bien immobilier (sortie) à partir d'entrées telles que nombre de pièces, année de construction, surface du terrain, emplacement,

etc.**[12]**
Le machine Learning avec supervision peut se subdiviser en deux types :

***/- Classification :** La variable de sortie est une catégorie.

***/- Régression** : La variable de sortie est une valeur spécifique. **[12]**

2.3.2 Machine Learning sans supervision

Dans le Machine Learning sans supervision, l'algorithme est laissé à lui-même pour déterminer la structure de l'entrée (aucun label n'est communiqué à l'algorithme). Cette approche peut être un but en soi (qui permet de découvrir des structures enfouies dans les données) ou un moyen d'atteindre un certain but. **[12]**

Cette approche est également appelée « apprentissage des caractéristiques » (feature learning). Un exemple de Machine Learning sans supervision est l'algorithme de reconnaissance faciale prédictive de Facebook, qui identifie les personnes sur les photos publiées par les utilisateurs. Il existe deux types de Machine Learning sans supervision :

***/- Clustering :** L'objectif consiste à trouver des regroupements dans les données.

***/- Association :** L'objectif consiste à identifier les règles qui permettront de définir de grands groupes de données. **[12]**

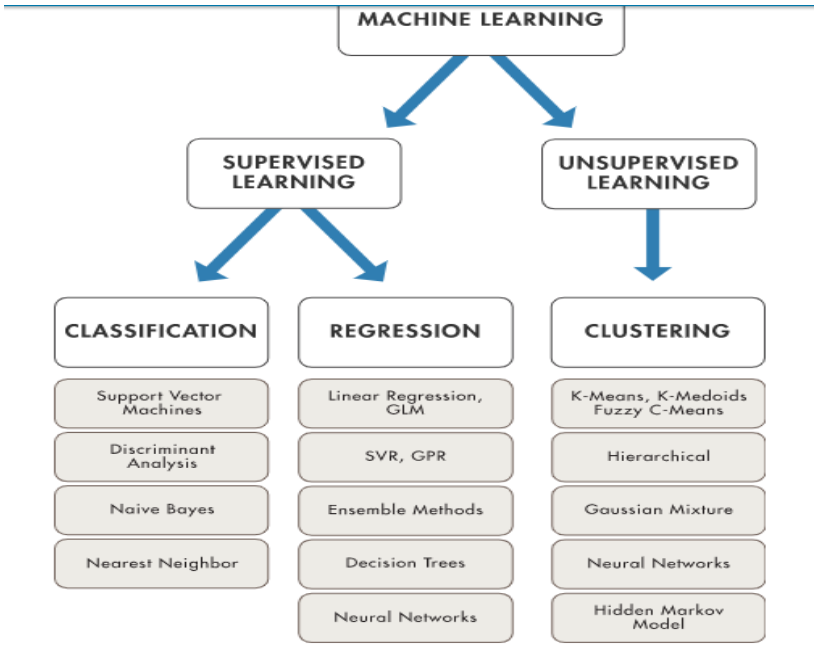

Figure 8 : Algorithmes des ML

2.3.3 Machine Learning par renforcement

Dans le Machine Learning par renforcement, un programme informatique interagit avec un environnement dynamique dans lequel il doit atteindre un certain but, par exemple conduire un véhicule ou affronter un adversaire dans un jeu. **[12]**

Le programme-apprenti reçoit du feedback sous forme de « récompenses » et de « punitions » pendant qu'il navigue dans l'espace du problème et qu'il apprend à identifier le comportement le plus efficace dans le contexte considéré. **[12]** En 2013, c'était déjà un algorithme de Machine Learning par renforcement (Q-Learning) qui s'était rendu célèbre en apprenant comment gagner dans six jeux vidéo Atari sans aucune intervention d'un programmeur. **[12]**

Il existe deux types de machine Learning par renforcement :

***/- Monte Carlo :** Le programme reçoit ses récompenses à la fin de l'état « terminal ».

***/- Machine Learning par différence temporelle (TD) :** Les récompenses sont évaluées et accordées à chaque étape. **[12]**

2.4 Algorithmes de Machine Learning

2.4.1 Algorithme de Classification

2.4.1.1 L'arbre de décision

L'arbre commence par une racine (où on a toute nos observations) puis une série de branches dont les intersections s'appellent des nœuds et termine par des feuilles qui correspondent chacune à une des classes à prédire. On parle de profondeur de l'arbre comme étant le nombre maximum de nœuds avant d'atteindre une feuille. **[14]** 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:

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 1 : Data set

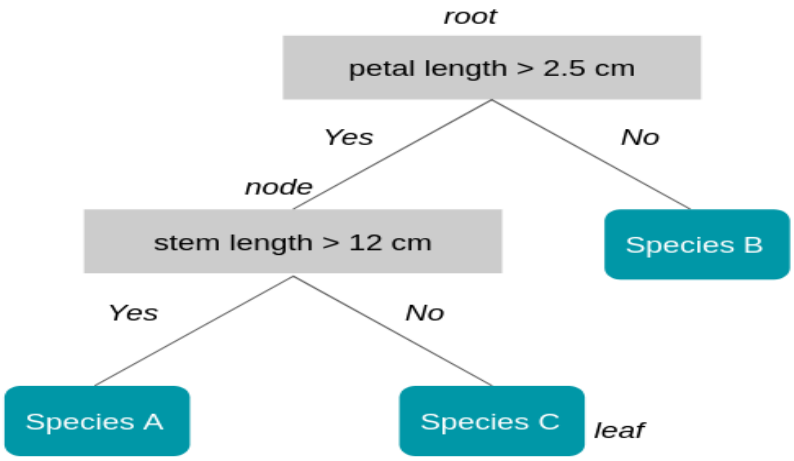

Figure 9 : Arbre de Décision

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). **[14]**

2.4.1.2 Le KNN

K plus proches voisins:

La méthode des K plus proches voisins (KNN) a pour but de classifier des points cibles (classe méconnue) en fonction de leurs distances par rapport à des points constituant un échantillon d'apprentissage (c'est-à-dire dont la classe est connue a priori).

KNN est une approche de classification supervisée intuitive, souvent utilisée dans le cadre du machine learning. Il s'agit d'une généralisation de la méthode du voisin le plus proche (NN). NN est un cas particulier de KNN, où k = 1.

L'approche de classification KNN se base sur l'hypothèse que chaque cas de l'échantillon d'apprentissage est un vecteur aléatoire issu de Rn. Chaque point est décrit comme x =< a1(x), a2(x), a3(x),.., an(x) > où ar(x) correspond à la valeur I du rème attribut. ar(x) peut être soit une variable quantitative soit une variable qualitative. **[15]**

2.4.1.3 Le Naïve Bayes

Naïve Bayes Classifier est un algorithme populaire en Machine Learning. C'est un algorithme du Supervised Learning utilisé pour la classification. Il est particulièrement utile pour les problématiques de classification de texte. Un exemple d'utilisation du Naïve Bayes est celui du filtre anti-spam.

Le naïve Bayes classifier se base sur le théorème de Bayes. Ce dernier est un classique de la théorie des probabilités. Ce théorème est fondé sur les probabilités conditionnelles. **[16]**

2.4.1.4 Machines à Vecteurs de Support

Aussi connu sous le nom de "SVM" (Support Victor Machine) cet algorithme sert principalement à des problèmes de classification même s'il a été étendu à des problèmes de régression **[14]**

Une machine à vecteurs de support, traduction littérale pour Support Victor Machine, est un algorithme d'apprentissage automatique supervisé qui peut être utilisé à des fins de classification et de régression. Les SVM sont plus généralement utilisés dans les situations de classification.**[17]**. Les SVM reposent sur l'idée de trouver un hyperplan qui divise au mieux un jeu de données en deux classes **[17]**

Le SVM est utilisé pour les problèmes de classification de texte telles que l'attribution de catégorie, la détection du spam ou encore l'analyse des sentiments. Ils sont également couramment utilisés pour les problèmes de reconnaissance d'image, particulièrement en reconnaissance de forme et en classification de couleur. SVM joue également un rôle essentiel dans de nombreux domaines de la reconnaissance manuscrite des symboles, tels que les services d'automatisation postale.**[17]**

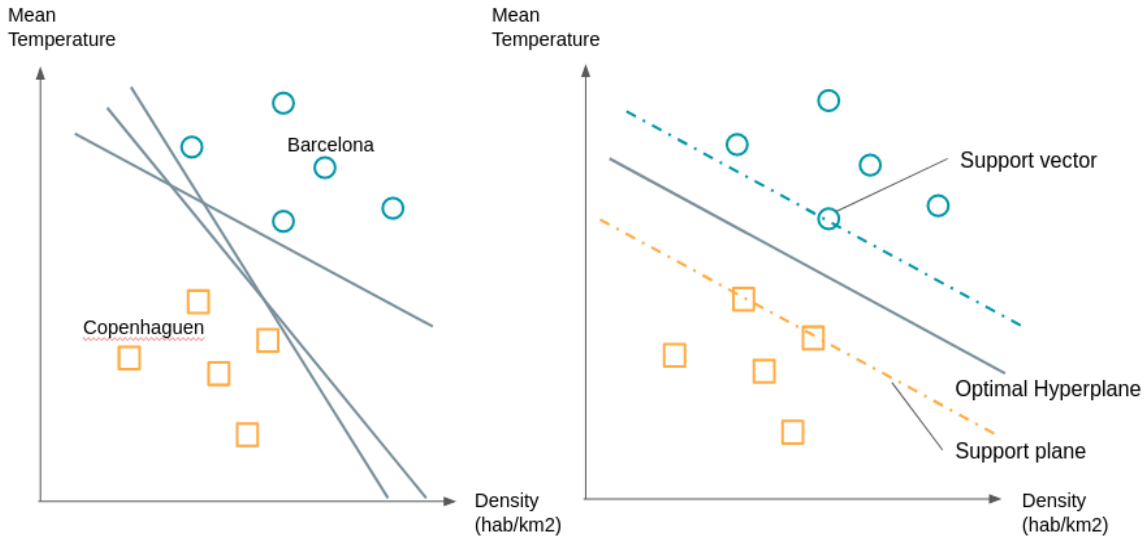

Figure 10 : Machines à Vecteurs de Support

2.4.1.5 Le AdaBoost

L'algorithme AdaBoost, proposé par Freund et Schapire **[9]**, est une méthode itérative qui produit une règle de classification performante (« forte ») en combinant plusieurs règles de précision modérée (apprenants ou classifieurs faibles). Il s'appuie sur la théorie de l'apprentissage PAC (Probably Approximately Correct) **[18]**

Un classificateur AdaBoost est un méta-estimateur qui commence par ajuster un classificateur sur l'ensemble de données d'origine, puis ajuste des copies supplémentaires du classificateur sur le même ensemble de données, mais où les poids des instances mal classées sont ajustés de sorte que les classificateurs suivants se concentrent davantage sur les cas difficiles. **[19]**

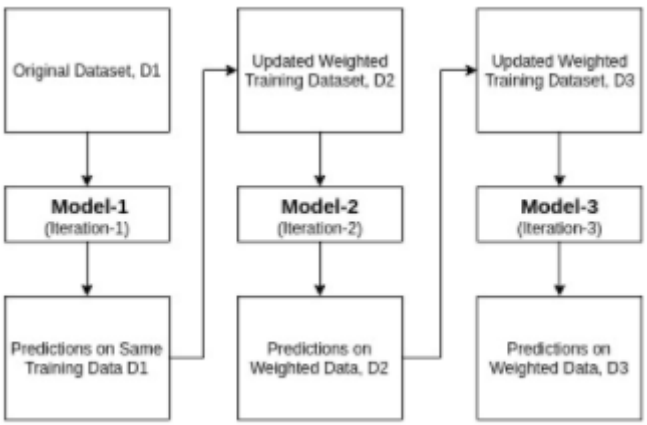

Figure 11 : Algorithme Adabooste

2.4.2 Linéaire régression

La régression est une mesure statistique utilisée en finance, mais aussi dans d'autres disciplines scientifiques pour tenter de déterminer la force de la relation entre une variable dépendante (habituellement conçue par Y) et une série d'autres variables changeantes (appelées variables indépendantes) .

La forme générale de la régression linéaire est la suivante : Y = a*X + b + epsilon avec a et b deux constantes. Y est la variable à prédire, X la variable utilisée pour prédire, a est la pente de la régression et b est l'intercept, c'est-à-dire la valeur de Y lorsque X est égal à zéro.

Dans le cas d'une régression linéaire multiple, il y a plusieurs variables changeantes et sur écrit Y = a*X1 + b*X2 + … + z + epsilon.

Epsilon, supposé très petit, correspond au terme d'erreur de régression. Il donne la possibilité de ne pas être tout à fait exact dans l'estimation. Son espérance mathématique est égale à zéro. **[20]**

2.4.3 Le Clustering

Cet algorithme a été conçu en 1957 au sein des Laboratoires Bell par Stuart P.Lloyd comme technique de modulation par impulsion et codage(MIC) . Il n'a été présenté au grand publique qu'en 1982. En 1965 Edward W.Forgy avait déjà publié un algorithme quasiment similaire c'est pourquoi le K-means est souvent nommé algorithme de Lloyd-Forgy. **[22]**

Les champs d'application sont divers : segmentation client, analyse de donnée, segmenter une image, apprentissage semi-supervisé….

Le clustering est une discipline particulière du Machine Learning ayant pour objectif de séparer vos données en groupes homogènes ayant des caractéristiques communes. C'est un domaine très apprécié en marketing, par exemple, où l'on cherche souvent à segmenter les bases clients pour détecter des comportements particuliers. L'algorithme des K-moyennes (K-means) est un algorithme non supervisé très connu en matière de Clustering. **[22]**

Clustering (ou partitionnement des données) : Cette méthode de classification non supervisée rassemble un ensemble d'algorithmes d'apprentissage dont le but est de regrouper entre elles des données non étiquetées présentant des propriétés similaires. Isoler ainsi des schémas ou des familles permet aussi de préparer le terrain pour l'application ultérieure d'algorithmes d'apprentissage supervisé (comme le KNN). **[6]**

Le clustering est utilisé notamment lorsqu'il est coûteux d'étiqueter le données. C'est néanmoins un problème mal défini mathématiquement : différentes métriques et/ou différentes représentations des données aboutiront à différents regroupements sans qu'aucun ne soit nécessairement meilleur qu'un autre. Ainsi la méthode de clustering doit être choisie avec soin en fonction du résultat attendu et de l'utilisation prévue des données.**[6]**

Fonctionnement simplement :

K-means définit par McQueen est un des plus simples algorithmes de classification automatique des données.

L'idée principale et de choisir aléatoirement un ensemble de centres fixé a priori et de chercher itérativement la partition optimale. Chaque individu est affecté au centre le plus proche, après l'affectation de toutes les données la moyenne de chaque groupe est calculée, elle constitue les nouveaux représentants des groupes, lorsqu'ont abouti à un état stationnaire (aucune donnée ne change de groupe) l'algorithme est arrêté**.[7]**

2.4.1.1 PCA

Est un algorithme d'apprentissage automatique non supervision qui tente de réduire la dimensionnalité (nombre de fonctions) au sein d'un ensemble de données tout en conservant autant d'informations que possible. Cette action s'effectue en recherchant un nouvel ensemble de variables appelées *composantes*, qui constituent les composés des caractéristiques originales décarrelées les unes les autres. Les composants sont également contraints de telle sorte que le premier composant représente la plus grande variabilité possible dans les données, le deuxième composant la deuxième variabilité la plus importante, et ainsi de suite. **[21]**

2.5. Les techniques de la sélection features

La sélection de caractéristique (ou sélection d'attribut ou de variable) est un processus utilisé en apprentissage automatique et en traitement de données. Il consiste, étant donné des données dans un espace de grande dimension, à trouver un sous-ensemble de variables pertinentes. C'est-à-dire que l'on cherche à minimiser la perte d'information venant de la suppression de toutes les autres variables. C'est une méthode de réduction de la dimensionnalité.**[23]**

2.5.1 Les Algorithmes Génétiques

L'algorithme génétique (AG) est un algorithme de recherche base sur les mécanismes de la sélection naturelle et de la génétique. Il combine une stratégie de "survie des plus forts" avec un échange d'information aléatoire mais structure. Pour un problème pour lequel une solution est inconnue, un ensemble de solutions possibles est créé aléatoirement.**[24]**

On appelle cet ensemble la population. Les caractéristiques (ou variables à déterminer) sont alors utilisées dans des séquences de gènes qui seront combinées avec d'autres gènes pour former des chromosomes et par après des individus. Chaque solution est associée à un individu, et cet individu est évalué et classifie selon sa ressemblance avec la meilleure, mais encore inconnue, solution au problème. Il peut être démontre qu'en utilisant un processus de sélection naturelle inspire de Darwin, cette méthode convergera graduellement à une solution.**[24]**

Comme dans les systèmes biologiques soumis à des contraintes, les meilleurs individus de la population sont ceux qui ont une meilleure chance de se reproduire et de transmettre

une partie de leur héritage génétique ´ a la prochaine génération. Une nouvelle population, ou génération, est alors créée en combinant les gènes des parents. On s'attend à ce que certains individus de la nouvelle génération possèdent les meilleures caractéristiques de leurs deux parents, et donc qu'ils seront meilleurs et seront une meilleure solution au problème. Le nouveau groupe (la nouvelle génération) est alors soumis aux même critères de sélection, et par après génère ses propres rejetons. Ce processus est répété plusieurs fois, jusqu'a ce que tous les individus possèdent le même héritage génétique. Les membres de cette dernière génération, qui sont habituellement très différents de leurs ancêtres, possèdent de l'information génétique qui correspond ´ a la meilleure solution au problème. L'algorithme génétique de base comporte trois opérations simples qui ne sont pas plus **[24]** compliquées que des opérations algébriques :

– Sélection

– Reproduction

– Mutation

L'algorithme génétique fut développée par Holland **[24]**

2.5.2 Particle Swarm Optimization (PSO)

L'optimisation par essaims particulaires (Particle Swarm Optimization) est une méthode qui s'inspire de la biologie pour résoudre des problèmes d'optimisation. **[25]** L'optimisation de l'essaim de particules (PSO) est un algorithme d'optimisation stochastique basé sur la population motivée par le comportement collectif intelligent de certains animaux tels que comme des volées d'oiseaux ou des bancs de poissons. Depuis sa présentation en 1995, il a connu une multitude d'améliorations. En tant que chercheurs ont appris la technique, ils ont dérivé de nouvelles versions visant à différentes demandes, développé de nouvelles applications dans un nombreux domaines, ont publié des études théoriques sur les effets des divers paramètres et proposé de nombreuses variantes de l'algorithme. Cet article présente son origine et son contexte et effectue l'analyse théorique du PSO. Ensuite, nous analysons sa situation actuelle de recherche et d'application en algorithme structure, sélection de paramètres, structure topologique, discret

Algorithme PSO et algorithme PSO parallèle, multi-objectif optimisation PSO et ses applications d'ingénierie. Finalement, les problèmes existants sont analysés et les directions de recherche futures sont présentées. **[26]**

APPRENTISSAGE	ALGORITHMES	AVANTAGES		INCONVENIENTS
		AdaBoostClassifier	1.facile à mettre en œuvre. 2.Il corrige de manière itérative les erreurs et améliore la précision. 3.peut utiliser de classificateurs de base avec AdaBoost. [27]	1.AdaBoost est sensible aux données de bruit. 2.AdaBoost est plus lent que XGBoost. [27]

Supervisé	Classification	SVM	1.Capacité à traiter de grandes dimensionnalités. 2.Traitement des problèmes non linéaires avec le choix des noyaux 3. Non paramétrique 4. Souvent performant dans les comparaisons avec les autres approches 5. Paramétrage permet de la souplesse [28]	1.Difficulté à identifier les bonnes valeurs des paramètres. 2. Problème lorsque les classes sont bruitées (multiplication des points supports) 3.Pas de modèle explicite pour les noyaux non linéaires (utilisation des points supports) 3. Difficulté d'interprétations (ex. pertinence des variables) 4. Le traitement des problèmes multi-classes reste une question ouverte.[28]
		Arbre de décision	1.Facilité à manipuler des données . 2. variables d'amplitudes très différentes 3.Multi-classe par nature 4.Classification très efficace (en particulier sur inputs de grande dimension) [29]	1.Sensibilité au bruit et points aberrants 2. Stratégie d'élagage délicate [29]
		Naïve Bayes	1.Lafacilité et la simplicité de leur implémentation. 2. Leur rapidité. 3. Les méthodes Naïve Bayes donnent de bons résultats.[30]	1. ce type d'algorithmes permet de faire le même travail de classification que les autres algorithmes qui existent déjà, mais ces performances sont limitées quand il s'agit d'une grande quantité de lexiques à traiter[30]
		KNN	1.L'algorithme est simple et facile à mettre en œuvre. 2.L'algorithme est polyvalent. Il peut être utilisé pour la classification, la régression.[31]	1.L'algorithme devient beaucoup plus lent à mesure que le nombre d'exemples d'apprentissage augmente. 2.Le choix de la méthode de calcul de la distance ainsi que le nombre de voisins K peut ne pas être évident. 3.L'étape de prédiction peut-être lente. La complexité est de l'ordre de O(n) avec (k << n) [31]
	REGRESSION	Linéaire	1. Simplicité d'interprétation. 2. facilité de calcul [47]	Elle ne traite pas les valeurs manquantes de variables continues sensible aux valeurs hors norme de variables continues [47]
	REDUCTION DES DIMENSIONS	PCA	1.Simplicité mathématique 2.Simplicité des résultats 3.Puissance 4. Flexibilité[47]	1.l'ACP n'a pas réellement 2.s'applique simplement sur des cas précis 3.Perte d'information par l'emploi fréquent de la 1ère composante principale uniquement. [47]
	CLUSTERING		1.Simple 2. Flexible 3. Efficace 4. Complexité temporelle	1.Ensemble non optimal de clusters 2.Manque de cohérence 3. Limitation des calculs

Non Supervisé		K-means	[32]	4. Spécifiez les valeurs k[32]

Conclusion :

Ce chapitre présente les concepts de base de l'apprentissage automatique. Premièrement, sa définition et ses types ont été abordés pour donner une image claire, et deuxièmement, les algorithmes ont été expliqués en détail, en particulier les algorithmes.

PARTIE 2

Chapitre 03

Approche proposée et l'implémentation

- **Chapitre 3 : L'approche proposé et l'implémentation**

1 Introduction

Dans ce chapitre, nous présentons l'approche proposée de notre IDS. 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 L'approche proposée

2.1 Matériel :

Le matériel réalisé est un PC personnel Dell I3 avec une capacité mémoire de 4GB, un processeur CPU i3-6006U cadencé à 2.00 GHz, tournant sous Windows 10 64 bits.

2.2 Le jeu de données utilisé

Le jeu de données CSE-CIC-IDS2018 comprend les captures du trafic réseau et les journaux système de chaque machine, ainsi que 80 fonctionnalités extraites du trafic capturé à l'aide de CICFlowMeter-V3 qui génère des flux bidirectionnels (Biflow) qui seront décrit dans le tableau suivant, où le premier paquet détermine les directions avant (source vers destination) et arrière (destination vers source), d'où les 83 caractéristiques statistiques telles que la durée, le nombre de paquets, le nombre d'octets, la longueur des paquets, etc. sont également calculés séparément dans le sens avant et arrière. La sortie de l'application est le format de fichier CSV avec six colonnes étiquetées pour chaque flux, à savoir Flow ID, Source IP, Destination IP, Source Port, Destination Port et Protocol avec plus de 80 fonctionnalités de trafic réseau dans ce tableau nous présentons chaque fonction avec sa description.**[33]**

L'ensemble de données CSE-CIC-IDS2018 comprend sept scénarios d'attaque différents : Brute-force, Heartbleed, Botnet, Dos, DDoS, attaques Web et infiltration du réseau de l'intérieur.

L'infrastructure attaquante comprend 50 machines et l'organisation victime compte 5 départements et comprend 420 machines et 30 serveurs.**[34]**

Nom de la fonction	Description
Dst Port	Port de destination de la connexion.
Protocol	Protocole utilisé lors de la connexion
Timestamp	Temp à laquelle la connexion a eu lieu
Flow Duration	Durée de la connexion.

Tot Fwd Pkts	Nombre total de paquets de transfert.
Tot Bwd Pkts	Nombre total de paquets vers l'arrière.
TotLen Fwd Pkts	Longueur totale des paquets de transfert.
TotLen Bwd Pkts	Longueur totale des paquets vers l'arrière.
Fwd Pkt Len Max	Taille maximale des paquets de transfert.
Fwd Pkt Len Min	Taille minimale des paquets de transfert.
Fwd Pkt Len Mean	Taille moyenne du paquet vers l'avant
Fwd Pkt Len Std	Taille de l'écart type du paquet vers l'avant
Bwd Pkt Len Max	Taille maximale du paquet vers l'arrière
Bwd Pkt Len Min	Taille minimale du paquet vers l'arrière
Bwd Pkt Len Mean	Taille moyenne du paquet vers l'arrière
Bwd Pkt Len Std	Taille de l'écart type du paquet vers l'arrière
Flow Byts/s	Nombre d'octets de flux par seconde
Flow Pkts/s	Nombre de paquets de flux par seconde
Flow IAT Mean	Temps moyen entre deux paquets envoyés dans le flux
Flow IAT Std	Temps d'écart type entre deux paquets envoyés dans le flux
Flow IAT Max	Temps maximum entre deux paquets envoyés dans le flux
Flow IAT Min	Temps minimum entre deux paquets envoyés dans le flux
Fwd IAT Tot	Temps total entre deux paquets envoyés dans le sens direct
Fwd IAT Mean	Temps moyen entre deux paquets envoyés dans le sens direct
Fwd IAT Std	Temps d'écart type entre deux paquets envoyés dans le sens direct
Fwd IAT Max	Temps maximum entre deux paquets envoyés dans le sens direct
Fwd IAT Min	Temps minimum entre deux paquets envoyés dans le sens direct

Bwd IAT Tot	Temps total entre deux paquets envoyés vers l'arrière
Bwd IAT Mean	Temps moyen entre deux paquets envoyés vers l'arrière
Bwd IAT Std	Temps d'écart type entre deux paquets envoyés vers l'arrière
Bwd IAT Max	Temps maximum entre deux paquets envoyés vers l'arrière
Bwd IAT Min	Temps minimum entre deux paquets envoyés vers l'arrière
Fwd PSH Flags	Nombre de fois que le drapeau PSH a été mis en paquets dans le sens aller (0 pour UDP)
Bwd PSH Flags	Nombre de fois que le drapeau PSH a été défini dans des paquets se déplaçant vers l'arrière (0 pour UDP)
Fwd URG Flags	Nombre de fois que le drapeau URG a été défini dans des paquets se déplaçant vers l'avant (0 pour UDP)
Bwd URG Flags	Nombre de fois où le drapeau URG a été défini dans des paquets se déplaçant vers l'arrière (0 pour UDP)
Fwd Header Len	Nombre total d'octets utilisés pour les en-têtes dans le sens direct
Bwd Header Len	Nombre total d'octets utilisés pour les en-têtes vers l'arrière
Fwd Pkts/s	Nombre de paquets de transfert par seconde
Bwd Pkts/s	Nombre de paquets en arrière par seconde
Pkt Len Min	Longueur minimale d'un paquet
Pkt Len Max	Longueur maximale d'un paquet
Pkt Len Mean	Longueur moyenne d'un paquet
Pkt Len Std	Longueur d'écart type d'un paquet
Pkt Len Var	Longueur de variance d'un paquet
FIN Flag Cnt	Nombre de paquets avec FIN
SYN Flag Cnt	Nombre de paquets avec SYN
RST Flag Cnt	Nombre de paquets avec RST
PSH Flag Cnt	Nombre de paquets avec PUSH
ACK Flag Cnt	Nombre de paquets avec ACK

URG Flag Cnt	Nombre de paquets avec URG
CWE Flag Count	Nombre de paquets avec CWE
ECE Flag Cnt	Nombre de paquets avec ECE
Down/Up Ratio	Ratio de téléchargement et de téléchargement
Pkt Size Avg	Taille moyenne des paquets
Fwd Seg Size Avg	Taille moyenne observée vers l'avant
Bwd Seg Size Avg	Nombre moyen d'octets débit en masse dans le sens direct
Fwd Byts/b Avg	Nombre moyen d'octets en vrac dans le sens direct
Fwd Pkts/b Avg	Nombre moyen de paquets en vrac dans le sens aller
Fwd Blk Rate Avg	Nombre moyen de taux en vrac vers l'avant
Bwd Byts/b Avg	Nombre moyen d'octets en vrac dans le sens retour
Bwd Pkts/b Avg	Nombre moyen de paquets en vrac dans le sens arrière
Bwd Blk Rate Avg	Nombre moyen de taux en vrac vers l'arrière
Subflow Fwd Pkts	Le nombre moyen de paquets dans un sous-flux dans le sens direct
Subflow Fwd Byts	Le nombre moyen d'octets dans un sous-flux dans le sens direct
Subflow Bwd Pkts	Le nombre moyen de paquets dans un sous-flux vers l'arrière
Subflow Bwd Byts	Le nombre moyen d'octets dans un sous-flux vers l'arrière
Init Fwd Win Byts	Le nombre total d'octets envoyés dans la fenêtre initiale dans le sens direct
Init Bwd Win Byts	Le nombre total d'octets envoyés dans la fenêtre initiale vers l'arrière
Fwd Act Data Pkts	Nombre de paquets avec au moins 1 octet de charge utile de données TCP dans le sens aller
Fwd Seg Size Min	Taille minimale de segment observée vers l'avant
Active Mean	Durée moyenne pendant laquelle un flux était actif avant de devenir inactif
Active Std	Écart type de temps un flux a été

Active Max	Durée maximale pendant laquelle un flux était actif avant de devenir inactif
Active Min	Durée minimale pendant laquelle un flux était actif avant de devenir inactif
Idle Mean	Temps moyen pendant lequel un flux était inactif avant de devenir actif
Idle Std	Temps d'écart type pendant lequel un flux était inactif avant de devenir actif
Idle Max	Durée maximale pendant laquelle un flux était inactif avant de devenir actif
Idle Min	Temps minimum pendant lequel un flux était inactif avant de devenir actif

Tableau 2 : fonctionnalités de trafic réseau avec la description

3 Approche détaillée

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

Expérience 1 : en applique le classifieur AdaBoost sur une data qui contient 78 feature. Et le résultat que nous obtenons est 92 %.

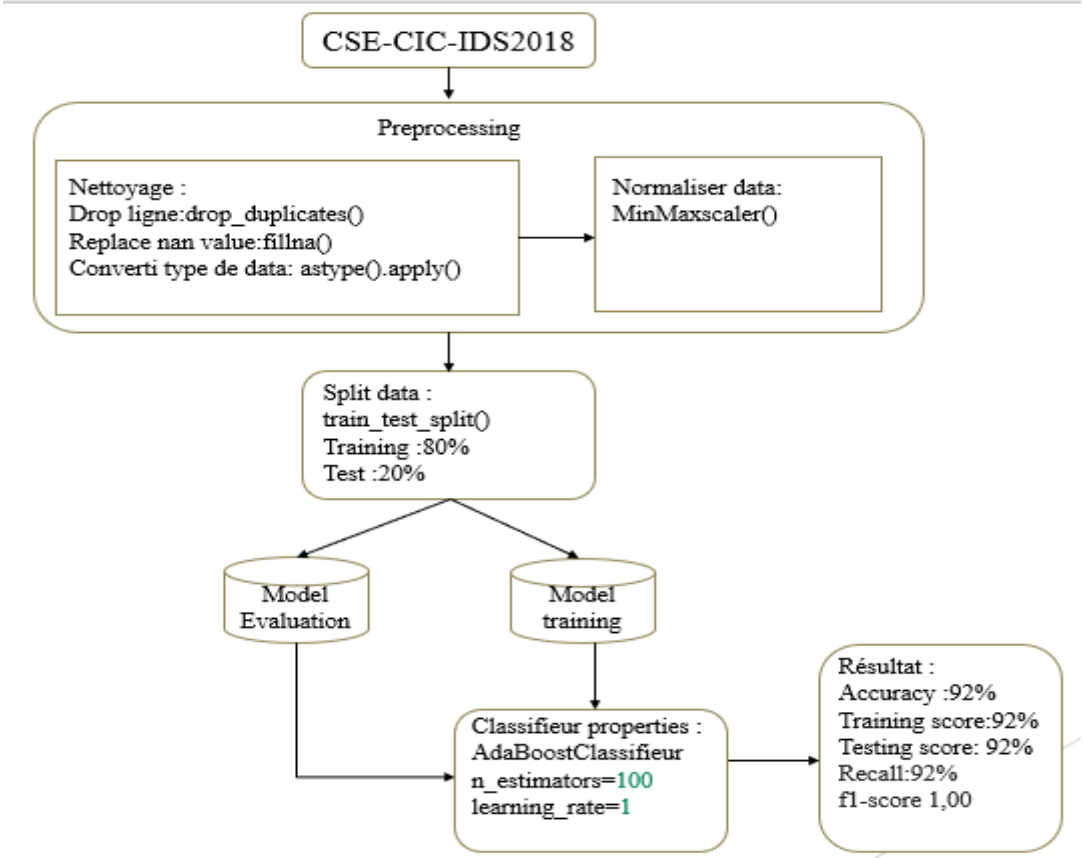

Figure 12 : Schéma de la méthode de la conception

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 « **AdaboostClassifieure** » et voici le résultat final obtenu :

	Accuracy	Training Score	Testing Score	Recall	f1-Scoor
Expérience 1	92 %	92 %	92 %	92 %	1.00

Tableau 3 : Résultat obtenu pour l'expérience 1

Expérience 2 : pour améliorer l'Accuracy de notre model, on a effectué une technique de Sélection de fonctionnalité « features sélection » SelectPercentile. et le résultat que nous obtenons est 96 %.

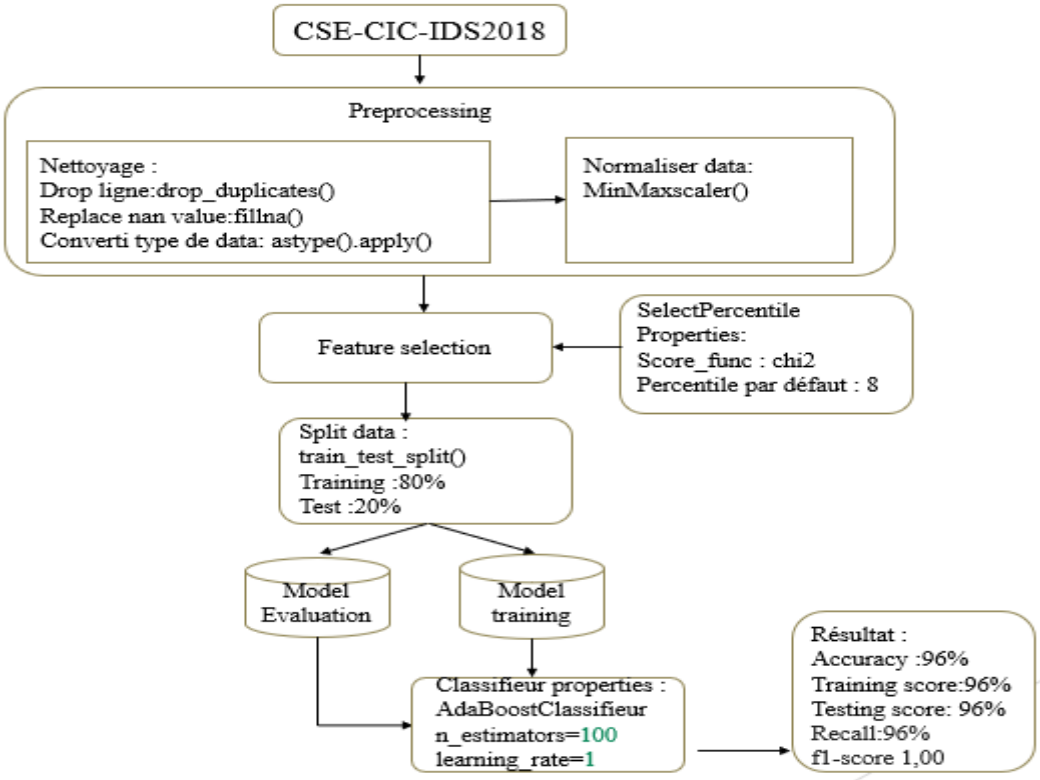

Figure 13 : Schéma de la méthode de la conception

Dans cette expérience, nous utilisant une technique de features sélection qui s'appelle « **SelectPercentil** » pour obtenu les features nécessaire, après nous donnes notre résulta a la technique « **Train_Test_Split()** » pour faire division de dataset , après nous appliquons l'algorithmes « **AdaboostClassifieur** » et voici le résultat obtenu :

	Accuracy	Training Score	Testing Score	Recall	f1-Scoor
Expérience 2	96 %	96 %	96 %	96 %	1.00

Tableau 4 : Résultat obtenu pour l'expérience 2

Nous remarquons que on a une amélioration entre l'expérience 01 et l'expérience 02 malgré les deux expériences donne nous des bons résultats

Expérience 3 : en appliquant la technique « Swarm – PSO » pour améliorer l'accuracy plus l'algorithme Adaboost Classifieur, alors lors de notre travail on a rencontré un problème de performance de notre machine à cause de la caractéristique de la machine .

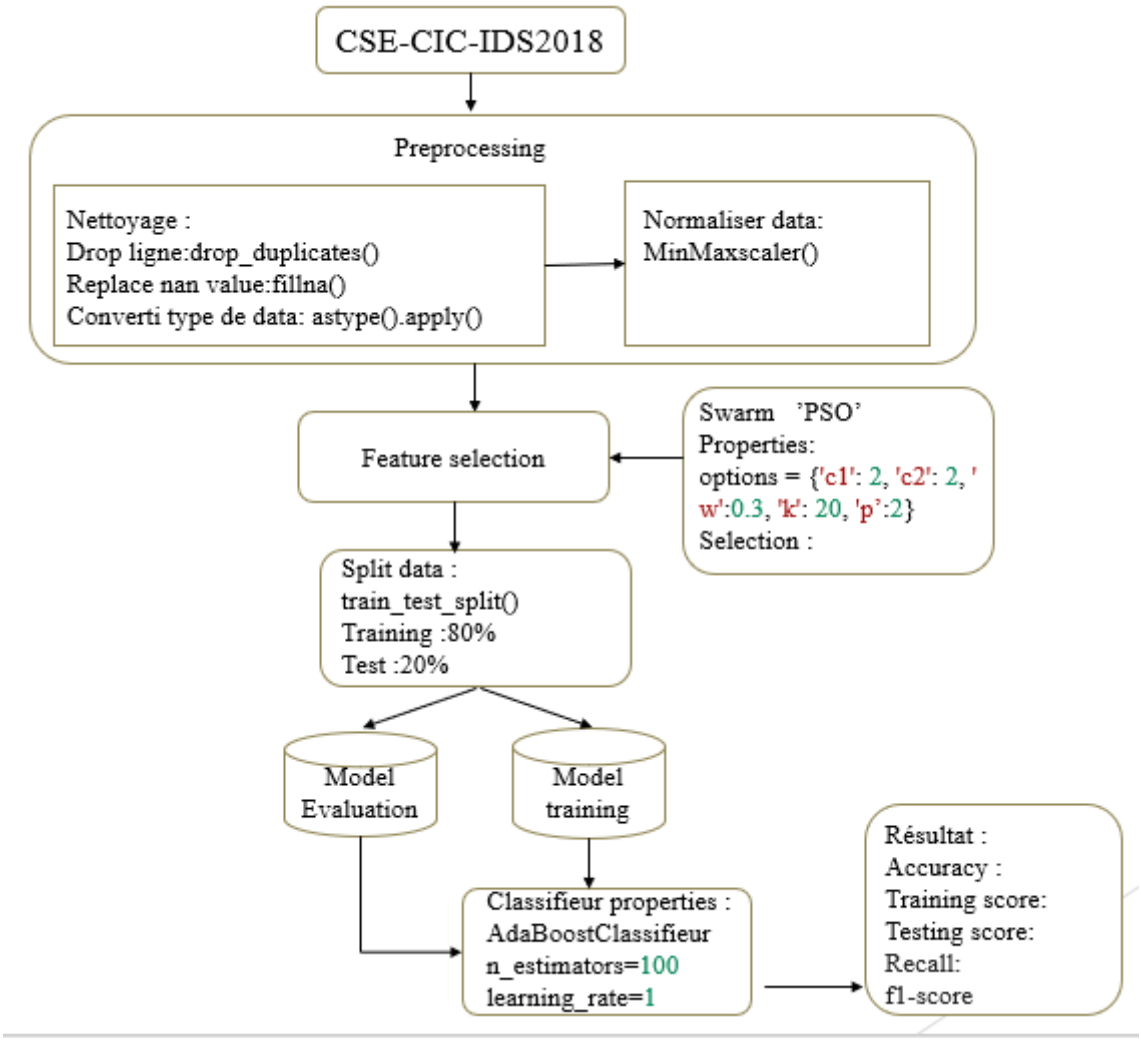

Figure 14 : Schéma de la méthode de la conception

Dans cette Expérience nous appliquant une technique de « **SWARM-PSO** » pour améliorer l'accuracy ,nous donne le résultat obtenu à la technique « **Train_Test_Split ()** » pour faire la division des donnée mais malheureusement lord le performance de notre machine nous n'avons pas obtenu de résultat :

	Accuracy	Trainning Sccore	Testing Scoore	Recall	f1-Scoor
Expérience 2	-	-	-	-	-

Tableau 5 : Résultat obtenu pour l'expérience 3

4 L'implémentation

4.1 Le Langage Python

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages **[35]**

Figure 15 : Logo Python

4.2. Colaboratory ' Colab '

est un produit de Google Research. Colab permet à n'importe qui d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. C'est un environnement particulièrement adapté au machine Learning, à l'analyse de données et à l'éducation. En termes plus techniques, Colab est un service hébergé de notebooks Jupyter qui ne nécessite aucune configuration et permet d'accéder gratuitement à des ressources informatiques, dont des GPU **[36]**

Figure 16 : logo Colaboratory et jupyter

4.3. Bibliothèques Supplémentaires :

Afin d'atteindre les objectifs de ce projet, nous avons utilisé d'autres bibliothèques externes pour effectuer certaines tâches spécifiques. En plus de celles fournies par la bibliothèque standard de Python.

- **Matplotlib**

C'est une librairie qui permet de tracer des graphes **[37]** est une bibliothèque Python open source permettant de créer des visualisations de données **[38]** est un module de Python qui permet de dessiner des courbes en deux et trois dimensions. Il s'agit d'une bibliothèque très riche qui génère des figures que vous pouvez enregistrer sous les formats **[39]**

- **Scikit-learn**

Est une bibliothèque développée en Python, un langage de programmation de haut niveau. Elle est dédiée à l'apprentissage statistique (machine Learning) et peut être utilisée comme middleware, notamment pour des tâches de prédiction **[40]**

- **NumPy**

Est le package fondamental pour le calcul scientifique en Python. Il s'agit d'une bibliothèque Python qui fournit un objet tableau multidimensionnel **[41]**

- **Pandas**

Est une librairie Python qui a pour objectif de vous faciliter la vie en matière de manipulation de données. Les structures de données gérées par Pandas peuvent contenir tout type d'éléments à savoir (dans le jargon Pandas) des Séries et Data Frame et des Panel. Dans le cadre de nos expérimentations on utilisera plutôt les Data frame car ils offrent une vue bidimensionnelle des données (comme un tableau Excel), et c'est exactement ce que l'on va chercher à utiliser pour nos modèles. **[42]**

- **glob**

En Python, le module glob est utilisé pour récupérer des fichiers / noms de chemin correspondant à un modèle spécifié. Les règles de modèle de glob suivent la règle standard d'extension de chemin Unix. Il est également prédit que selon les repères, il est plus rapide que les autres méthodes de faire correspondre les noms de chemin dans les répertoires. **[43]**

4.4. Chargement des données du Dataset

Pour commencer, il faudra lire et charger les données contenues dans le fichier csv. Python propose via sa librairie Pandas des classes et fonctions pour lire divers formats de

fichiers, la fonction read_csv est une fonction pandas importante pour lire les fichiers csv et effectuer des opérations dessus.

```
def read_data(dataroot,file_ending='*.csv'):
    if file_ending==None:
       print("please specify file ending pattern for glob")
       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('/content/drive/MyDrive/dataset/',file_ending='*.csv')
datasetdataset
```

Figure 17 : Chargement de données .

4.5. Nettoyage des données

Nettoyage de données est l'étape la plus importante avant d'analyser ou modéliser des données mais elle peut être très fastidieuse **[44]**

Lors de l'utilisation de la base CSE-CIC-IDS2018, nous avons constaté qu'elle contient quelques données obsolètes qui peuvent nuire au bon fonctionnement de notre système.

Dans CSE-CIC-IDS2018 il y a des lignes répétées il faut supprimer c'est lignes.et des colonnes qui contient des cellules avec le même nom de la colonne il faut remplacer par la valeur nan et en corrigés c'est valeur par la moyenne des colonnes.

```
data = data.drop_duplicates()
print('Les lignes répétées ont été supprimées')
```

```
for i in range(0,79):
    colm = data.columns.values[i]
    data = data.replace(colm , np.nan)
```

```
data.fillna(data.mean(), inplace=True)
print('Les valeurs est charger')
```

Figure 18 : nettoyage de données

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

```
data = data.astype(object).apply(pd.to_numeric)
print('La conversion numérique est faite ')
```

Figure 19 : convertir le type de donnes

4.6. 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.**[45]**

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] **[45]**

La normalisation peut- être effectuée par la technique du **Min-Max Scaling.** La transformation se fait grâce à la formule suivante **[45]**

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

Avec :

- X_{min}: la plus petite valeur observée pour la feature X
- X_{max}: la plus grande valeur observée pour la feature X
- X : La valeur de la feature qu'on cherche à normaliser

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

```
newdata =Normaliser(X)
```

Figure 20 : normalisation des donnes

4.7. 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.

Expérience 2 :

On a utilisé une technique de feature selection SelectPercentile

```
X_new =SelectPercentile(chi2).fit_transform(newdata, y)
X_new.shape
```

```
(1869101, 8)
```

Figure 21 : feature selection

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_cd, X_test_cd , y_train_cd , y_test_cd = train_test_split(newdata, y, test_size=0.20, random_state=42)
```

Figure 22 : séparation des donnes

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

```
#classifier with values of data complte
ada_sp = AdaBoostClassifier(n_estimators=100,learning_rate=1)
# Train Adaboost Classifer
model_sp = ada_sp.fit(X_train_sp, y_train_sp)
```

Figure 23 : Application le classifieur AdaBoostClassifier.

```
#Predict the response for test dataset
y_pred_sp = model_sp.predict(X_test_sp)
```

Figure 24 : Classification du données de test.

Après la prédiction on calculer le taux de précision avec la métrique accuracy, notre modèle à atteindre vers 92.22 % 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 25 : calcul de la précision.

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

```
print('Model test Score: %.3f, ' %model_sp.score(X_test_sp, y_test_sp),
      'Model training Score: %.3f' %model_sp.score(X_train_sp, y_train_sp))
```

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

Figure 26 : évaluation de model

4.8. 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.

```
Confusion matrix, without normalization
[[319679    688      8      0  11828]
 [     0   7953      0      0    251]
 [    99   1000    441      0    527]
 [     0      0      2   7919      0]
 [     0      6      0      3  23417]]
```

Figure 27 : MATRICE DE CONFUSION .

4.9. La courbe ROC :

Les courbe ROC sont créé, En traçant le vrai taux positif par rapport au taux de faux positifs à différents paramètres de seuil pour chaque classe, pour montre la capacité de diagnostic du notre classifieur par le calcule (TPR) et (FPR). On a le résultat suivant qui montre que pour chaque class la courbe est au-dessus de la diagonale, et pour chaque classe on a :

- Seuil: En bas à gauche, point (0,0)
- Taux de faux positifs (FPR): 0. Le classificateur n'a identifié aucun échantillon négatif réel comme positif
- True Positive Rate (TPR): 0 . Le classificateur n'a pu attraper aucun des échantillons True Positive
- Seuil2: En haut à droite, point (1.0, 1.0) (la barre de maintien est en bas)
- FPR: 1,0 .Le classificateur a identifié tous les échantillons négatifs réels comme positifs TPR: 1.0 Classifier a montré une bonne performance sur la capture de tous les positifs réels

Ainsi, le point idéal est donc le coin supérieur gauche du graphique : les faux positifs sont proches de 0 et les vrais positifs sont proches de 1 .

Cependant, pour qu'il y ait discrimination, il est nécessaire que la courbe monte très vite. Il faut avoir simultanément forte sensibilité et forte spécificité. Une mesure du pouvoir discriminante est obtenue à l'aide de l'aire sous la courbe ROC **[46]**

Si ROC = 0.5	Pas de discrimination
Si 0.5 < ROC <0.7	Discrimination insuffisante
Si 0.7 = ROC <0.8	Discriminante acceptable
Si 0.8 = ROC <0.9	Discriminante excellente
Si 0.9 = ROC < 1	Discriminante exceptionnelle

Tableau 6 : Mesures de discrimination.

Pour notre modèle, sa valeur est éloignée vers 0.8 . Elle traduit là une discrimination excellente

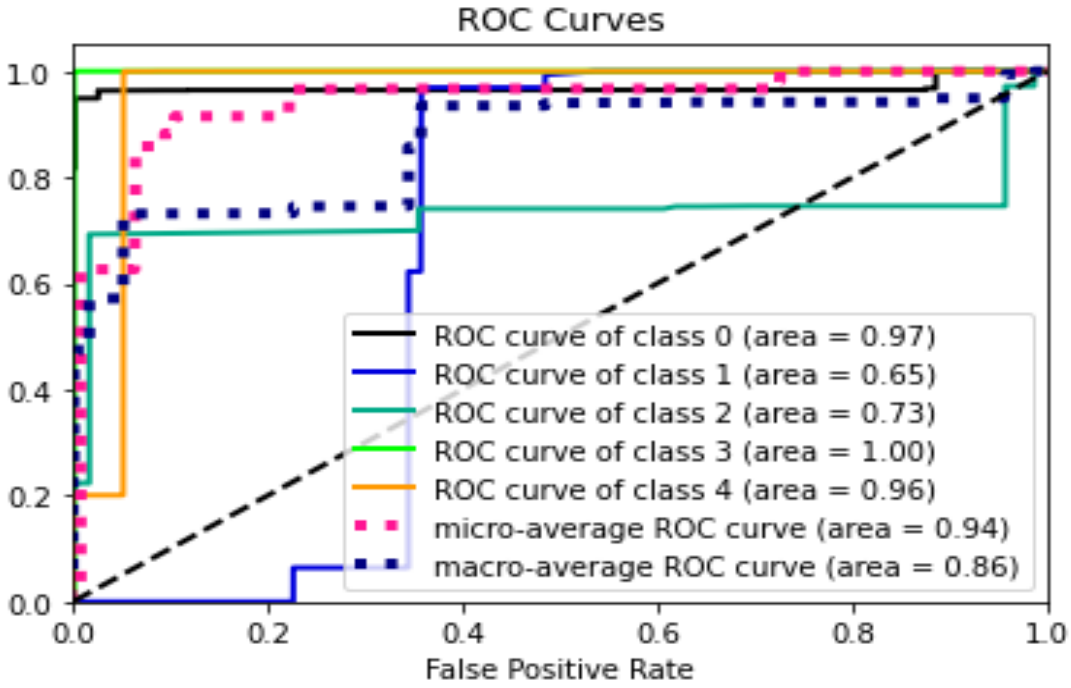

Figure 28 : Courbe ROC

Ce travail montre à quel point le jeu de données CSE-CIC-IDS2018 est très utile pour tester différents classificateurs, les travaux se concentrent sur la phase de prétraitement de CSE-CIC-IDS2018 afin de préparer des expériences fiables et des données de test indépendantes randomisées. Parmi les techniques de classification, le classifier Adabooste a atteint le taux de précision le plus élevé pour la détection et la classification de tous les types d'attaques de jeux de données CSE-CIC-IDS2018

4.10. Rapport de classification (Classification report) :

Le rapport de classification est utilisé pour mesurer la qualité des prévisions de notre classifieur. Combien de prédictions sont vraies et combien sont fausses. Plus précisément, les vrais positifs, les faux positifs, les vrais négatifs et les faux négatifs sont utilisés pour prédire les mesures d'un rapport de classification, comme indiqué ci-dessous :

	precision	recall	f1-score	support
0	1.00	0.96	0.98	332203
1	0.82	0.97	0.89	8204
2	0.98	0.21	0.35	2067
3	1.00	1.00	1.00	7921
4	0.65	1.00	0.79	23426
accuracy			0.96	373821
macro avg	0.89	0.83	0.80	373821
weighted avg	0.97	0.96	0.96	373821

Figure 29 : rapport de classification

- Le rappel signifie "combien de cette classe vous trouvez sur le nombre entier d'éléments de cette classe".
- La précision sera "combien sont correctement classés dans cette classe"
- Le score f1 est la moyenne harmonique entre précision et rappel
- Le support est le nombre d'occurrences de la classe donnée dans votre ensemble de données.

4.11. Définition des termes :

- Vrai positif (TP) : l'observation est positive et devrait être positive.
- Faux négatif (FN) : L'observation est positive, mais prédite négative.
- Vrai négatif (TN) : l'observation est négative et devrait être négative.
- Faux positif (FP) : L'observation est négative, mais prédite positive.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+F}$$

La précision est le rapport où est le nombre de vrais positifs et le nombre de faux positifs. La précision est intuitivement la capacité du classificateur à ne pas étiqueter comme positif un échantillon négatif

$$\text{Précision} = \frac{TP}{TP+FN}$$

Le rappel est le rapport où est le nombre de vrais positifs et le nombre de faux négatifs. Le rappel est intuitivement la capacité du classificateur à trouver tous les échantillons positifs

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{Score F1} = 2^* \frac{\text{Precession} * \text{recall}}{\text{Precession} + \text{recall}}$$

Le support est le nombre d'occurrences de chaque classe dans y_true.

Résutat / Expérience	Accur %	Training score %	Evaluation Score %	Recall %	F1-score %
1	92	92	92	92	92
2	96	96	96	96	96
3	-	-	-		

Tableau 7 : tableau général du résultats obtenus

5. Conclusion :

Dans ce chapitre nous avons, en premier lieu, présenté les différents outils et langages que nous avons utilisés pour implémenter notre model. Ainsi une approche hybride de (SelectPercentile + AdaboostClassifer) a été utilisée.
Ce travail montre à quel point le jeu de données CSE-CIC-IDS2018 est très utile pour tester différents classificateurs, les travaux se concentrent sur la phase de prétraitement de CSE-CIC-IDS2018 afin de préparer des expériences fiables et des données de test indépendantes randomisées. Parmi les techniques de classification.

Conclusion Générale

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 SelectPercentil et Adabooste pour scruter les informations en profondeur de ce processus.

Les techniques utilisées (SelectPercentil+Adabooste) présentent une grande capacité de modélisation pour IDS et une grande précision dans la classification utilisant les techniques d'apprentissage automatique. Dans le cadre de la tâche de classification multi classée sur le jeu de données CSE-CIC-IDS2018, le modèle peut effectivement améliorer à la fois la précision de la détection d'intrusion et la capacité de reconnaître le type d'intrusion. La combinaison du SelectPercentile et de Adabooste offre de meilleures performances que les systèmes de détection d'intrusion existants en atteint une précision de 96.00 % avec 62 de caractéristiques sélectionnées.

Notre projet a été une opportunité pour approfondir nos connaissances dans le domaine de Machine Learning et d'apprendre ses différents modèles et leurs applications. Il est important pour nous de dire que l'un des avantages majeurs de ce travail est de familiariser avec la compréhension des articles et la maîtrise de plusieurs bibliothèques où nous avons vu et les exploiter pour la création des modèles.

Annex

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. Algorithme Adaboost :

Initialization:

1. Given training data from the instance space $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y} = \{-1, +1\}$.
2. Initialize the distribution $D_1(i) = \frac{1}{m}$.

Algorithm:

for $t = 1, \ldots, T$: **do**

Train a weak learner $h_t : \mathcal{X} \rightarrow \mathrm{R}$ using distribution D_t.

Determine weight α_t of h_t.

Update the distribution over the training set:

$$D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$$

where Z_t is a normalization factor chosen so that D_{t+1} will be a distribution.

end for

Final score:

$f(x) = \sum_{t=0}^{T} \alpha_t h_t(x)$ and $H(x) = sign(f(x))$

Annex

3. Algorithme de k-means

Entrée

- Un ensemble de données D, où chaque instance X_i est décrite par un vecteur de d dimensions et par une classe $Y_i \in \{1, \ldots, J\}$.
- Le nombre de clusters souhaité, noté K.

Début

1) Prétraitement des données.
2) Initialisation des centres.

Pour un nombre fixé de partitions, noté **R faire**

Répéter

3) *Affectation :* générer une nouvelle partition en assignant chaque instance X_i au groupe dont le centre est le plus proche.

$$X_i \in C_k \forall j \in 1, \ldots, K \qquad k = min_j \parallel X_i - \mu_j \parallel$$

avec μ_k est le centre de gravité du cluster C_k.

4) *Représentation :* calculer les centres associés à la nouvelle partition

$$\mu_k = \frac{1}{N_k} \sum_{X_i \in C_k} X_i$$

jusqu'à ce que (convergence de l'algorithme)

Fin Pour

5) Choix de la meilleure partition parmi les R partitions.
6) Attribution des classes aux clusters formés.
7) Prédiction de la classe des nouvelles instances.

Fin

Sortie

- Chaque cluster est représenté par un prototype qui possède la même prédiction de classe.
- Chaque cluster est associé à une description donnée par le biais de langage B.
- L'inertie intra-clusters est minimale (l'homogénéité des instances est maximale).
- L'inertie inter-clusters est maximale (la similarité entre les clusters est minimale).
- Le taux de bonnes classifications est maximal.

4. Algorithme SelectPercentile

```
# Import Libraries
from sklearn.feature_selection import SelectPercentile
from sklearn.feature_selection import chi2 , f_classif
#--------------------------------------------------

#Feature Selection by Percentile
print('Original X Shape is ' , X.shape)
FeatureSelection = SelectPercentile(score_func = chi2, percentile=20) # 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

[01] : https://web.maths.unsw.edu.au/~lafaye/CCM/secu/secuintro.htm

[02] : https://www.f-secure.com/fr/business/resources/intrusion-detection-systems

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

[04]:https://www.lemagit.fr/definition/Systeme-de-detection-dintrusions

[05]:https://blog.varonis.fr/ids-et-ips-en-quoi-sont-ils-differents/

[06]: Hervé Debar, Benjamin Morin, FrédéricCuppens, Fabien Autrel, LudovicMé, Bernard Vivinis Salem Benferhat, MireilleDucassé, RodolpheOrtalo, Détectiond'intrusions : corrélationd'alertes. Article de synthèse, Caen, France, 2004

[07]: Cédric Michel, Langage de description d'attaques pour la détection d'intrusions par corrélation d'événements ou d'alertes en environnement réseau hétérogène, thèse de doctorat de l'Université de Rennes1,16 Décembre 2003

[08]:MÜLLER, Klaus, ALIAS'SOCMA, Klaus Müller, et TARBOURIECH, Georges. IDS-Systèmes de Détection d'Intrusion, Partie I. LinuxFocus article, 2003, no 292.

[09]:Alanou, Ludovic Mé—Véronique. "Détection d'intrusion dans un système informatique: méthodes et outils." SUPELEC BP 2835511 (1996).

[10]:Müller, K. (2003). IDS– Système de Détection d'Intrusion, Partie II.

[11] : https://www.arageek.com/l/ما-هو-تعلم-الآلة

[12] :https://www.talend.com/fr/resources/what-is-machine-learning/

[13] : https://datascientest.com/deep-learning#:~:text=Le%20Deep%20learning%20ou%20apprentissage%20profond%20est%20l%27une%20des,des%20réseaux%20de%20neurones%20artificielles.

[14]:https://datakeen.co/8-machine-learning-algorithms-explained-in-human-language/

[15] : https://www.xlstat.com/fr/solutions/fonctionnalites/k-nearest-neighbors-knn

[16] : https://mrmint.fr/naive-bayes-classifier

[17] : https://analyticsinsights.io/les-svm-support-vector-machine/

[18] : https://projet.liris.cnrs.fr/imagine/pub/proceedings/RFIA-2010/pdf/4B_P26-Prevost.pdf

[19] : https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html?fbclid=IwAR0a1fjAbseo5YINqUfKHiVqP5f9m0cwEi1lkbOR53bKY2Hs0PMEBpXPbIE

[20] : https://blog.nalo.fr/lexique/regression-lineaire/

[21] : https://docs.aws.amazon.com/fr_fr/sagemaker/latest/dg/pca.html

[22] : https://datascientest.com/algorithme-des-k-means

Bibliographie

[23] : https://fr.wikipedia.org/wiki/Sélection_de_caractéristique

[24] : http://www8.umoncton.ca/umcm-cormier_gabriel/SystemesIntelligents/AG.pdf

[25] : https://dataanalyticspost.com/Lexique/particle-swarm-optimization-pso/

[26] : https://kpfu.ru/staff_files/F_1407356997/overview.pdf

[27] : https://ichi.pro/fr/classificateur-adaboost-en-python-155257329026137

[28] : https://eric.univ-lyon2.fr/~ricco/cours/slides/svm.pdf

[29] : https://people.minesparis.psl.eu/fabien.moutarde/ES_MachineLearning/Slides/coursFM_AD-RF.pdf

[30] : http://depot-e.uqtr.ca/id/eprint/1201/1/030110265.pdf

[31] : https://www.isnbreizh.fr/nsi/activity/algoRefKnn/index.html

[32] : Hilali, h., application de la classification textuelle pour l'extraction des règles d'association maximales. thèse de maitrise en informatique, université du québec à trois-rivières, trois-rivières, 2009.

[33] : https://www.unb.ca/cic/datasets/ids-2018.html

[34] : https://www.unb.ca/cic/datasets/ids-2018.html

[35] : https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/

[36] : https://research.google.com/colaboratory/faq.html?hl=fr

[37] : http://www.python-simple.com/python-matplotlib/matplotlib-intro.php

[38] : https://datascientest.com/matplotlib-tout-savoir

[39] : https://cahier-de-prepa.fr/mp1-janson/download?id=526

[40] : https://www.inria.fr/fr/lancement-de-linitiative-scikit-learn?fbclid=IwAR1r89W0NsQHyu7BN31qRQJq5YEUS0iORwj37i51Zj0ds35stAwHCL-8N8c

[41] : https://numpy.org/doc/stable/user/whatisnumpy.html

[42] : https://www.datacorner.fr/pandas-1/

[43] : https://www.geeksforgeeks.org/how-to-use-glob-function-to-find-files-recursively-in-python/

[44] : https://moncoachdata.com/blog/nettoyage-de-donnees-python/

[45] : https://mrmint.fr/data-preprocessing-feature-scaling-python

[46] : https://www.memoireonline.com/12/19/11370/m_Facteurs-explicatifs-de-linadequation-professionnelle10.html

Bibliographie

[47] : « BOUROUBA Hadjer , CHAOUCHE Ouidad / Optimisation des IDS du Cloud Computing par les techniques de machines Learning / word/univ-tiaret – ibn khaldoun/ 2019-2020 »