République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

UNIVERSITÉ IBN KHALDOUN DE TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHEMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Réseaux et Télécommunication

Par :
BEHTANI saadia kamilia
BAHLOUL zoulikha bochra

Sur le thème

La proposition d'un système de détection d'intrusion efficace basée sur les anomalies du trafic réseau déséquilibré

Soutenu publiquement le 04 / 07 / 2023 à Tiaret devant le jury composé de :

Mr Benaouda Hbib	MAA Université de Tiaret	Président
Mr Daoud Mohamed Amine	MAA Université de Tiaret	Encadreur
Mr Meghazi lhadj madani	MAA Université de Tiaret	Examinateur

2022-2023

Dédicace

Je dédie ce modest travail à …✍

A ma très chère mère, qui me donne toujours
l'espoir de vivre et qui n'a jamais cessé de prier pour moi
A mon très cher père, pour ses encouragement, son
soutien, surtout pour son amour et son sacrifice afin que
rien n'entrave le déroulement de mes études.
A mes Sœurs , A tous les Amis en particulier bochra
Tout qui ma m'aide et compluse ce modest travail.

Kamilia.

Je dédie ce modest travail à …✍

A l'étre le plus cher de ma vie, ma mère.
A celui qui m'a fait moi une femme, mon père.
A mes chers Frères.
A tous les amis en particulier kamilia et,
A tous ceux qui ont participé a ma réussite

Bochra.

Remerciements…✍

Nous avons à commencer par exprimer notre gratitude à Dieu, qui est le tout puissant, le maître des cieux et de la terre, qui nous a donné la direction et nous a permis de réaliser ce projet.

Nous voulons aussi exprimer notre gratitude à notre promoteur, Monsieur DAOUD Mohamed Amine, qui s'est toujours montré disponible et écouté tout au long de la réalisation de ce mémoire présent et qui a su guider et structurer nos idées grâce à ses précieux conseils.

Nos sincères remerciements aux membres du jury qui ont accepté d'évaluer notre travail.

Un grand merci à nos amis et familles pour leur soutien continu et leur confiance en nos capacités.

Enfin, nous remercions tous ceux qui ont contribué de près ou de loin à ce modeste projet.

Résumé

Les systèmes de détection d'intrusion (IDS) jouent un rôle essentiel dans l'identification des comportements anormaux dans les environnements système et réseau. Les techniques d'apprentissage automatique ont gagné une importance dans ce domaine. Cependant, la présence de données déséquilibrées pose un défi important pour obtenir des résultats de détection précis et fiables. Cette étude se concentre sur l'optimisation des mesures de performance des IDS pour un dataset récent déséquilibré CICIDS 2017. Plus précisément, la recherche met l'accent sur l'impact des données déséquilibrées et explore les limites des approches à un classificateur unique pour classer efficacement le trafic normal et les anomalies (attaques). Pour y remédier, l'étude propose la construction d'un modèle plus sophistiqué qui combine plusieurs classificateurs basés sur les modèles d'ensemble, (Bagging et Boosting), il est évalué à l'aide de l'ensemble de données CICIDS-2017. Les résultats montrent que l'IDS est plus robuste et plus efficace avec mesure de performance de F1-score de 99.97% pour la combinaison du modèle SMOTE avec Bagging.

Abstract

Intrusion detection systems (IDS) play an essential role in identifying anomalous behavior in system and network environments. Machine learning techniques have gained importance in this field. However, the presence of unbalanced data poses a significant challenge to obtaining accurate and reliable detection results. This study focuses on optimizing IDS performance metrics for a recent unbalanced dataset CICIDS 2017. Specifically, the research focuses on the impact of unbalanced data and explores the limitations of single-classifier approaches to effectively classify normal traffic and anomalies (attacks). To address this, the study proposes the construction of a more sophisticated model that combines several classifiers based on ensemble models, (Bagging and Boosting), it is evaluated using the CICIDS-2017 dataset. The results show that the IDS is more robust and efficient, with an F1-score performance measure of 99.97% for the model combination SMOTE with Bagging.

الملخص

تلعب أنظمة الكشف عن التسلل (IDS)دورًا حاسمًا في تحديد السلوك غير الطبيعي في بيئات النظام والشبكة. اكتسبت تقنيات التعلم الآلي أهمية في هذا المجال. غير أن وجود بيانات غير متوازنة يشكل تحديا كبيرا أمام الحصول على نتائج كشف دقيقة وموثوقة. تركز هذه الدراسة على تحسين مقاييس أداء IDS لمجموعة بيانات CICIDS لعام 2017 غير المتوازنة. على وجه التحديد، يركز البحث على تأثير البيانات غير المتوازنة ويستكشف قيود نهج التصنيف الفردي لتصنيف حركة المرور العادية والشذوذ (الهجمات) بكفاءة. لعلاج ذلك، تقترح الدراسة إنشاء نموذج أكثر تطوراً يجمع بين العديد من المصنفات بناءً على نماذج المجموعات (التعبئة والتغليف)، ويتم تقييمه باستخدام مجموعة البيانات .CICIDS-2017 وتبين النتائج أن مؤشر التنمية المستدامة أكثر قوة وكفاءة F1- score قياس الأداء بنسبة99.97% في المائة لمزيج النموذج SMOTE مع Bagging .

Table des matières

Liste des abréviations

IDS : Intrusion detection system.

HIDS: Host Intrusion Detection System.

NIDS: Network Intrusion Détection System.

SMOTE: Synthetic Minority Oversampling Technique.

IA : Intelligence Artificielle.

ML : Machine learning.

RL: Renforcement Learning.

K-PPV : K-Plus Proches Voisins.

SVM :Machine vectorielle de soutien.

Liste des Figures

Liste des Tableaux

Introduction générale

Les systèmes de détection d'intrusion (IDS) sont cruciaux pour prévenir les attaques malveillantes sur les réseaux informatiques. Les modèles d'apprentissage automatique peuvent être utilisés pour améliorer les systèmes de détection d'intrusion en permettant la détection automatique des attaques. Cependant, les performances de ces modèles peuvent être affectées par un problème de données déséquilibrées. Pour résoudre ce problème et améliorer la précision des IDS, il est donc crucial de sélectionner et de prétraiter soigneusement les données.

De nombreux ensembles de données des systèmes de détection d'intrusion (IDS) rencontrent le problème des données déséquilibrées, où le nombre d'instances de certaines classes d'attaques est considérablement plus faible que d'autres. Par exemple, dans le célèbre ensemble de données CICIDS001, CICIDS2017 et CICIDS2018 Les chercheurs ont proposé diverses méthodes pour résoudre ce problème de déséquilibre des données et améliorer les performances des modèles d'apprentissage automatique dans la détection d'attaques.
Cependant, les IDS sont confrontés à des défis des données déséquilibrées, où la distribution du trafic réseau normal et des activités malveillantes est fortement asymétrique. On parle généralement d'une classe du traffic normal est plus dominante qu'une autre classe, qui représente les anomalies ou les attaques. Ce déséquilibre des classes pose un problème aux algorithmes traditionnels d'apprentissage automatique, car ils ont tendance à privilégier la classe majoritaire et peinent à détecter efficacement les instances de la classe minoritaire.

Les techniques de Machine Learning permettent aux systèmes d'apprendre de manière autonome à effectuer des tâches et à faire des prédictions à partir de données, en améliorant leurs performances au fil du temps. Cependant, lorsqu'il s'agit de classifier le trafic normal et les anomalies (attaques), l'utilisation d'un seul classificateur présente des limites

La combinaison des techniques des données déséquilibrées et de l'apprentissage automatique permet d'améliorer la précision et l'efficacité de la détection des intrusions. En relevant les défis posés par les données déséquilibrées, l'apprentissage automatique peut permettre aux IDS de mieux identifier les menaces potentielles et d'y répondre renforçant ainsi la sécurité des réseaux et des systèmes.

Ce travail est structuré en deux parties principales. La première partie est composée de trois chapitres, qui sont organisés comme suit :
Dans le premier chapitre, nous avons parlé des systèmes de détection d'intrusion qui sont des dispositifs de sécurité permettant de détecter les tentatives d'intrusion dans un réseau informatique ou dans un système

d'information. Nous avons consacré cette section à la présentation de ces systèmes et de leur fonctionnement, ainsi qu'aux différents types d'IDS et à leurs méthodes de détection.
Le deuxième chapitre a présenté l'apprentissage automatique, ainsi que ses différents types et ses algorithmes. Nous avons également discuté des concepts fondamentaux de cette technologie et de sa relation avec l'IA.
Le troisième chapitre traite des données déséquilibrées, présentant les différents types de déséquilibre de classe dans les ensembles de données et les problèmes associés. De plus, nous avons appliquer des méthodes pour gérer les données déséquilibrées telles que le suréchantillonnage, le sous-échantillonnage et SMOTE, ainsi que des méthodes d'ensemble pour améliorer les performances des modèles de machine learning sur ces données.

Dans la deuxième partie, nous avons les deux autres chapitres organisés comme suit :

Le chapitre 4 de notre mémoire, la deuxième partie, contient la proposition du notre modèle. L'ensemble de données d'évaluation de la détection d'intrusion CICIDS2017, ainsi que le modèle proposé et les étapes que nous avons suivies pour le développer, ont été détaillés.

Dans le dernier chapitre,nous avons présenté les outils, les bibliothèques et les langages de programmation que nous avons utilisés pour développer notre modèle, ainsi que les étapes de prétraitement des données En outre, nous avons présenté diverses expériences menées et leurs résultats correspondants. L'objectif de ce chapitre était d'analyser les résultats et de discuter de leur pertinence par rapport à l'état de l'art.

A la fin, ce travail se termine par une conclusion générale.

PARTIE 1

Recherche bibliographique (État de l'art)

Chapitre 1

Les systèmes de détection d'intrusion

Chapitre 1 : Système de détection d'intrusion

1.1. Introduction

La sécurité des systèmes informatiques généralement limité 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 possèdent uniquement les droits qui leur ont été octroyés.
Un système de détection d'intrusion (IDS) est une technologie de sécurité qui protège les environnements contre les cybers attaques en surveillant le trafic réseau pour détecter les activités suspectes et en envoyant des alertes lorsque les intrusions sont identifiées.

1.2. Définition La sécurité informatique

La sécurité informatique est l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires à la maintenance, la restauration et la sécurisation des systèmes informatiques. Elle est intrinsèquement liée à la sécurité de l'information et des systèmes d'information.
La raison principale de l'existence de la sécurité informatique est que les produits et services informatiques ne sont pas intrinsèquement sécurisés. Si les ordinateurs étaient protégés contre les virus, il n'y aurait pas besoin de produits antivirus. Si le mauvais trafic réseau ne pouvait pas être utilisé pour attaquer les ordinateurs, alors aucun on s'inquiéterait de l'achat d'un pare-feu... Cependant, nous n'avons pas besoin de dépenser des milliards chaque année pour les rendre plus sécurisés. [1]

1.3. Les propriétés de sécurité

Les principales propriétés de sécurité sont :

1.3.1.1 Confidentialité

Permet d'assurer que l'information sur le système ne puisse être lue que par les personnes autorisées.

1.3.1.2 Intégrité

L'intégrité permet de certifier que l'information sur le système ne puisse être modifiée que par les personnes autorisées (pas de divulgation à des tiers non autorisés).

1.3.1.3 Disponibilité

Demande que l'information sur le système soit disponible aux personnes autorisées. [9]

1.4. Intrusion

Action (ou tentative d'action) qui a pour conséquence de compromettre l'intégrité, la confidentialité ou la disponibilité d'une ressource (violation de la politique de sécurité). [8]

1.5. Détection d'intrusions

Les techniques de détection d'intrusion tentent de faire la distinction entre une utilisation normale du système et une tentative d'intrusion, tout en émettant des alertes. En règle générale, les données d'audit du système sont analysées à la recherche d'intrusions connues ou de signatures comportementales inhabituelles. La détection peut se faire en temps réel, auquel cas un programme (IDS) peut déclencher une alerte et un personnel qualifié peut tenter de remédier à l'intrusion en coupant la liaison ou en se mettant en piste. [3]

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

Un IDS, (**Intrusion Detection System**), est un système logiciel ou matériel conçu pour être capable de surveiller automatiquement un réseau ou une machine particulière pour les événements qui se produisent, et pour pouvoir signaler à l'administrateur système toute trace d'activité inhabituelle dans ce dernier ou sur la machine surveillée. IDS est un système de détection passif. [4]

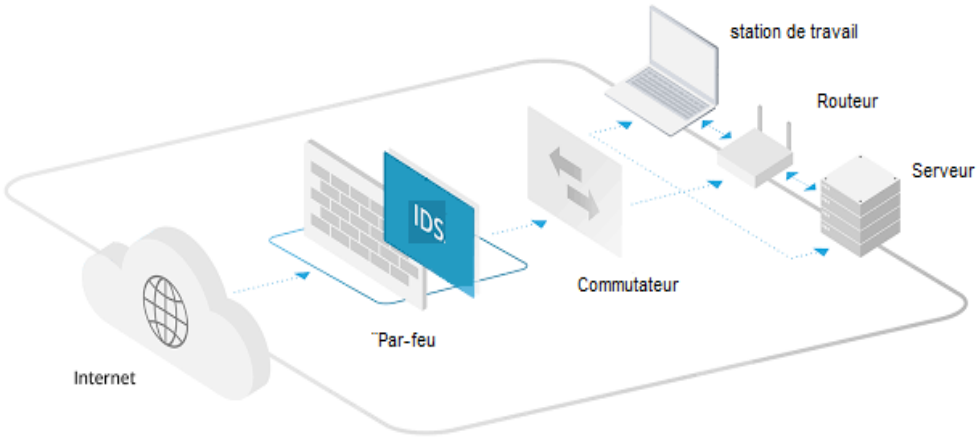

Figure 1: L'emplacement d'un IDS

1.7. Logiciels existants

1.7.1. Bro

Bro est apparu au même moment que Snort, avec les mêmes intentions : offrir un moyen de décrire des signatures pour détecter le trafic malicieux et également se baser sur des modules spécialisés. Pour décrire ces signatures, Bro propose un langage dédié qui se rapproche d'un langage impératif classique et permet de décrire les motifs des attaques. Sa communauté est moins importante que celle de Snort, mais il faut remarquer que Bro est la première solution à considérer que le système d'analyse puisse être la cible d'une attaque.

Bro propose des mécanismes pour contrer ces différents types d'attaques. [6]

Figure 2 : Logo Bro

1.7.2. Snort

Fonctionne à la fois comme NIDS et HIDS ainsi que comme IPS. L'analyseur NIDS de Snort tente de comparer chaque paquet qu'il reçoit avec un ensemble de règles définies par la configuration de Snort. Des règles peuvent être spécifiées pour détecter certains contenus dans la charge utile du paquet ou d'autres caractéristiques trouvées dans le paquet ou d'autres caractéristiques trouvées dans l'en-tête du paquet. Lorsqu'une règle correspond à un paquet, Snort peut prendre des mesures telles que l'alerte, la journalisation du paquet, l'ignorance du paquet ou l'abandon du paquet. [5]

Figure 3: Logo Snort

1.7.3. Suricata

Suricata fonctionne de la même manière que Snort, mais offre une plus grande évolutivité grâce à la prise en charge du multithreading dans son analyseur. Il est également plus rapide que Snort, bien qu'il consomme plus de ressources. [5]

Figure 4: Logo Suricata

1.7.4. Zeek

Anciennement connu sous le nom de Bro, est un moniteur de sécurité réseau qui fournit la fonctionnalité IDS de sécurité réseau. Zeek fonctionne différemment de Snort et Suricata car l'analyseur de Zeek inclut un moteur d'événements qui déclenche différents événements dans le trafic. Les événements sont transmis à la plate-forme de script de

Zeek, où les caractéristiques du trafic sont analysées plus en détail à l'aide de ses propres scripts. L'analyse est effectuée à l'aide du propre langage de script de Zeek, qui génère des fichiers journaux et des notifications sur les activités suspectes ou intéressantes. Le langage de script est Turing complet, faisant de Zeek une plateforme d'analyse de trafic hautement personnalisable. [5]

Figure 5: Logo ZEEK

1.8. Objectifs des systèmes de détection d'intrusions

Un IDS a plusieurs objectifs parmi lesquels :

- La détection de toutes violations liées à la politique de sécurité **:** L'IDS surveille le réseau ou le système pour détecter tout comportement anormal ou toute violation des règles de sécurité prédéfinies. Son but est de détecter toute activité qui ne correspond pas à la politique de sécurité établie.
- La signalisation des attaques : L'IDS est conçu pour identifier et signaler les attaques potentielles ou réelles. Il utilise des techniques telles que l'analyse des signatures, l'analyse comportementale et la corrélation des événements pour détecter des modèles d'activité suspects associés à des attaques connues ou à des comportements malveillants.
- L'analyse du trafic à tous les niveaux, liaison de données, réseau, transport et application : L'IDS examine le trafic réseau à différents niveaux, notamment la liaison de données, le réseau, le transport et l'application. Il peut ainsi détecter des anomalies ou des activités. [7]

1.9. Architecture des IDS et Principes de fonctionnement

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.

L'objectif des travaux du groupe IDWG est de définir un format de communication standard entre les différents composants d'un système de détection d'intrusion. La figure ci-dessous illustre ce modèle et introduit plusieurs concepts :

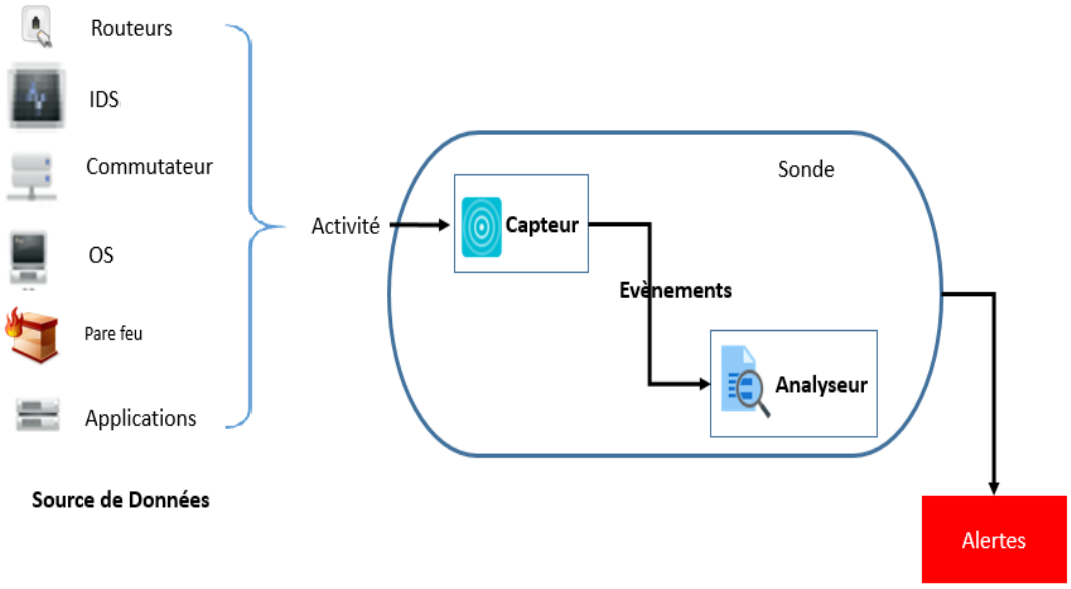

Figure 6 :Modèle générique de la détection d'intrusions proposé par l'IDWG [8]

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 :

1.9.1. Source de données

Dispositif générant de l'information sur les activités des entités du système d'information.

1.9.2. Capteur

Génère des événements en filtrant et formatant les données brutes provenant d'une source de données.

1.9.3. Evénement

Message formaté et renvoyé par un capteur. C'est l'unité élémentaire utilisée pour représenter une étape d'un scénario d'attaques connu.

1.9.4. 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.

1.9.5. Sonde

Un ou des capteurs couplés avec un analyseur.

1.9.6. Alerte

Message formaté émis par un analyseur s'il trouve des activités intrusives dans une source de données. [8]

1.10. Classification des systèmes de détections d'intrusion

La classification des IDS basée sur divers critères qui ne sont pas nécessairement liés les uns aux autres ne présentent pas tour à tour les catégories caractérisant chaque IDS, et elle utilise les critères suivants.

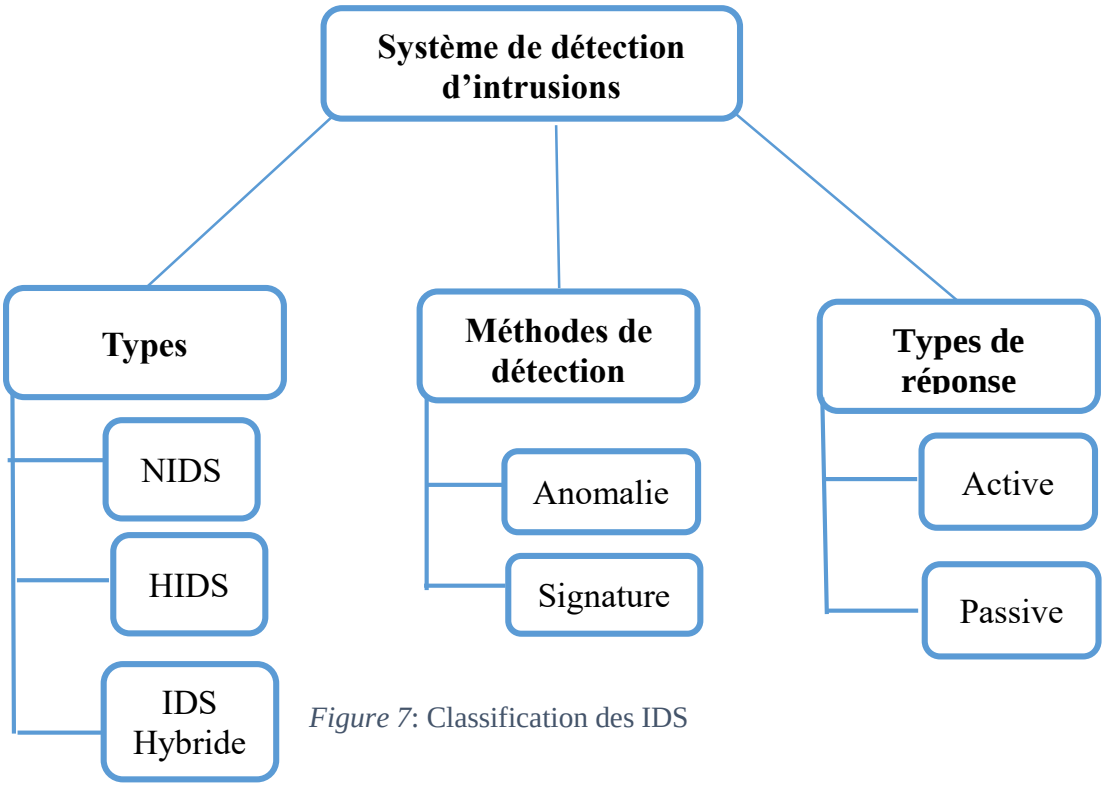

Figure 7: Classification des IDS

1.11. Les types d'un système de détection d'intrusion

1.11.1. Systèmes de détection d'intrusion réseaux (NIDS)

L'IDS réseau ou (NIDS : Network Intrusion Détection System) surveille le trafic réseau. Il se place sur un segment réseau et écoute le trafic. Ce trafic sera ensuite analysé afin de détecter les signatures d'attaques ou les différences avec le fonctionnement de référence. [9]

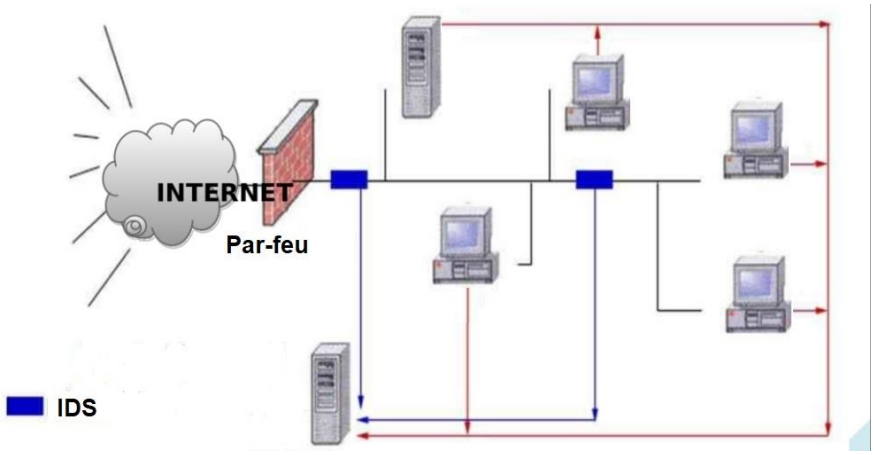

Figure 8: NIDS

1.11.2. Systèmes de détection d'intrusion sur hôte (HIDS)

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 levées. [10]

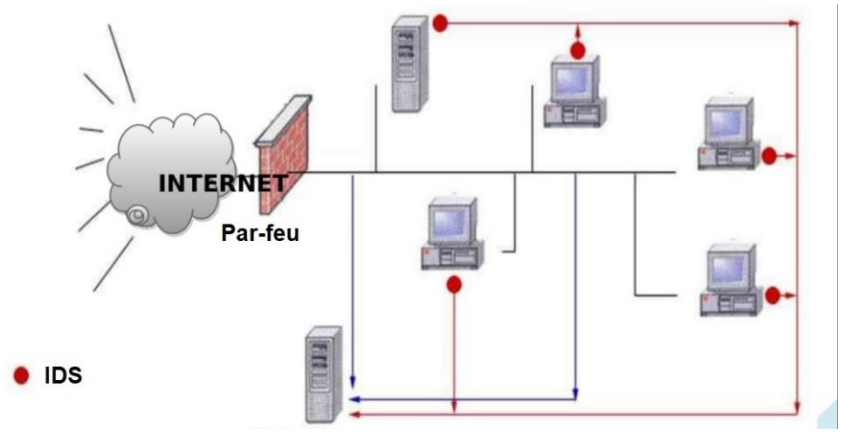

Figure 9: HIDS

1.11.3. Un IDS hybrides

Les IDS hybrides sont généralement utilisés dans des environnements décentralisés. Ils collectent des informations à partir de différentes sondes placées sur le réseau et peuvent fonctionner à la fois comme un NIDS (système de détection des intrusions sur le réseau) et/ou un HIDS (système de détection des intrusions sur l'hôte), en fonction de leur emplacement. Le terme "hybride" fait référence au fait que ces systèmes sont capables de collecter des informations provenant à la fois des systèmes HIDS et NIDS. Toutes ces sondes HIDS et NIDS remontent alors les alertes à une machine qui va centraliser le tout, et agréger/lier les informations d'origines multiples. [11]

	NIDS	HIDS
Avantages	-Le NIDS peut surveiller un grand réseau (un grand nombre d'hôte). [9] - Assurer la sécurité contre les attaques puisqu'il est invisible. [13] - les capteurs peuvent être bien sécurisés puisqu'ils se "contentent" d'observer le trafic. [12] -Détection facile grâce aux signatures. [12]	-Analyse des flux cryptés (ce que ne peut réaliser un NIDS). [9] -Aucun équipement supplémentaire requis. -Surveille les intrusions qui s'appli- quent uniquement à l'hôte. [10] -Permet de constater l'impact d'une attaque et pout donc mieux réagir. [12] -Observation des activités sur l'hôte avec précision. [12]
Inconvénients	-Difficile de détecter des intrusions provenant de contenu chiffré. [14] -Faible devant les attaques de dénis de services. [10] -La probabilité de faux négatifs (attaques non détectées comme telles) est élevée et il est difficile de contrôler le réseau entier. [12]	-Besoin de l'installer sur chaque machine. [14] -Détection d'attaques locales uniquement. [14] - Besoin de HIDS spécifique pour des systèmes spécifiques. -Ils consomment beaucoup de ressources CPU. [13]

Placement	-Réseau physique ou virtuel.	-Machine virtuelle ou physique.
Déploiement & responsabilité	-Administrateur.	-Administrateur & utilisateur.

Figure 10 : comparaison entre NIDS et HIDS

1.12. Méthodes de détection

1.12.1. Détection par Anomalie

Les IDS basés sur la détection d'anomalies reposent sur l'idée que le comportement d'un attaquant diffère du comportement normal d'un utilisateur légitime. Ces systèmes aident à détecter les attaques en évolution, c'est-à-dire celles qui n'ont pas encore été répertoriées dans les bases de signatures connues.

Les IDS basés sur des anomalies analysent le comportement normal du système et continuent de le mettre à jour pendant un certain temps. Par exemple, chaque connexion réseau est caractérisée par un ensemble de fonctionnalités telles que le protocole, le service, le nombre de tentatives de connexion, les paquets par flux, les octets par flux, l'adresse source, l'adresse de destination, le port source, le port de destination, etc. Les statistiques comportementales de ces fonctionnalités sont enregistrées sur une période. Tout écart anormal dans les valeurs des fonctionnalités pour tout flux de connexion sera marqué comme anormal par le moteur de détection d'anomalies. [9]

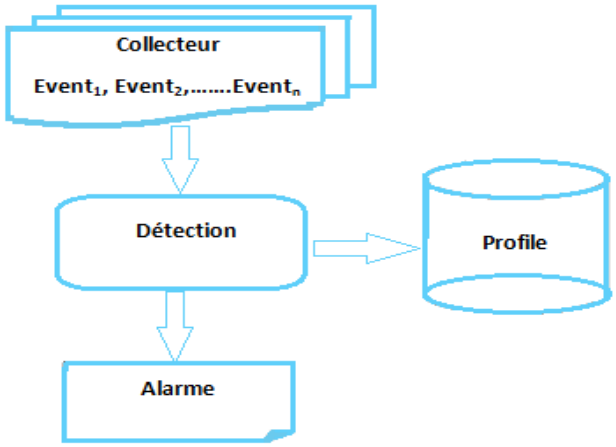

Figure 11: Modèle de détection par l'approche comportementale.

1.12.2. Détection par Signature

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. Chaque signature est habituellement définie comme une séquence d'événements et de conditions qui décrit une tentative d'intrusion. La détection repose sur le concept de "Pattern Matching" (correspondance de motifs), où l'analyse des chaînes de caractères présentes dans les paquets est effectuée afin de trouver des correspondances avec une base de connaissances de signatures. 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). [4]

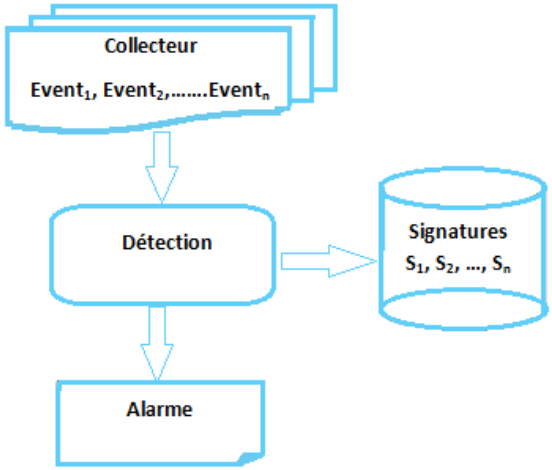

Figure 12: Modèle de détection pour l'approche par signature.

	Signature	**Anomalie**
Avantages	-L'analyse basée sur la connaissance est très efficace pour la détection des attaques, avec un taux très faible de fausses alarmes positives. -Les alarmes générées peuvent être significatives en termes de faux positifs et de faux négatifs.	-Efficace pour détecter de nouvelles vulnérabilités dans des situations imprévues. -Faciliter la détection des abus de privilège

Inconvénients	-L'analyse basée sur la connaissance permet uniquement la détection des attaques préalablement connues. Par conséquent, la base de connaissances doit être régulièrement mise à jour avec les signatures des nouvelles attaques. -Le risque que l'attaquant peut influencer sur la détection après la reconnaissance des signatures.	-Faible précision des profils en raison des événements observés. -Non disponible lors de la reconstruction des profils de comportement. -Difficile de déclencher des alertes en temps réel.

Tableau 1: les avantages et les inconvénients des méthodes de détection. [1]

1.13. Comportement après la détection

Après la détection d'intrusion 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.

- Réponse passive

Lorsqu'une attaque est détectée, le système de détection d'intrusions ne prend aucune action, il génère seulement une alarme.

- Réponse active

La réponse active consiste à répondre directement à une attaque. [15]

1.14. Les avantages d'utilisation des IDS [9]

- Déjouer les attaques attendues sur le réseau

Les IDS protègent les systèmes contre les attaques réseaux par : détection de porte dérobée, détection d'usurpation d'adresse IP, Dos, les vers, les chevaux de Troie, virus, Botnet, rootkit, Spyware, et autres menaces qui pourraient nuire au réseau, Les IDS actifs prennent des mesures automatiques contre les menaces de sécurité et les risques auxquels font face.

- Avertis Administrateur réseau d'alerte pour les événements de sécurité potentiels

La fonction de base des systèmes de détection d'intrusions est de générer des avertissements là où existent des menaces externes, internes ou de violations de la politique de sécurité réseau, et aussi de fournir à l'administrateur des informations détaillées sur le mouvement des données au sein du réseau.

- Gagnez du temps

L'utilisation des IDS fournit beaucoup de temps et d'effort pour connaitre de ce qui se passe dans le réseau, peut aussi tourner en permanence sans superviseur humain.

- Contrôle des Programmes utilisés par les employés pour surveiller l'Internet

IDS peut aider à découvrir les programmes qui traitent de l'internet, cela permet de mieux contrôler et de protéger le réseau.

- Avoir la confiance des clients

Les IDS aident les organisations de protéger les données de ses clients contre le vol et la violation de la sécurité, Cela permet d'avoir la confiance des clients et partenaires et garder une bonne réputation sur l'organisation.

➢ Économisez de l'argent

Grace aux IDS les organisations peuvent déterminer les mouvements suspects dans le réseau et signaler les responsables pour prendre des mesures proactives en protéger le réseau et gagner l'argent qui sera dépensé si la violation de la sécurité est arrivée dans le réseau ou si le vol de renseignements personnels a eu lieu.

1.15. Mesures performances des IDS [8]

Les mesures performances sont utilisées pour évaluer l'efficacité de ces systèmes dans la détection des activités suspectes ou malveillantes sur un réseau ou un système informatique.

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

		Réalité	
		Négative :0	**Positive:1**
Prédiction	**Négative :0**	Vrai négative (VN)	Faux positive (FP)
	Positive :1	Faux négative (FN)	Vrai positive (VP)

Tableau 2: Matrice de confusion [8].

- Faux positif

Une alerte provenant d'un IDS mais qui ne correspond pas à une attaque réelle. [15]

- Faux négatif

On parle de faux négatif lorsqu'une tentative d'intrusion n'est pas détectée.

- Vrais positifs

L'entrée est une attaque qui est détectée par IDS comme une attaque.

- Vrais négatifs

L'entrée est normale, ce qui est détecté par IDS comme un trafic normal. [16]

1.15.1. Accuracy

Dans les problèmes de classification, la précision est le nombre de prédictions correctes sur l'ensemble des prédictions effectuées. C'est une bonne mesure lorsque les classes de variables cibles sont presque équilibrées. [34]

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

1.15.2. Précision

La précision indique, sur l'ensemble des observations positives, le nombre d'observations positives prédites correctement. Plus le nombre de prédictions correctes est élevé, plus le classificateur est performant. La prédiction peut être donnée à l'aide de l'équation suivante : [34].

Précision=TP/(TP+FP)

1.15.3. Le rappel

Indique, sur l'ensemble des observations de la classe actuelle, le nombre d'observations positives prédites correctement. Comme pour la précision, plus le nombre de prédictions correctes est élevé, plus le classificateur est performant. Le rappel peut être calculé à l'aide de l'équation suivante : [34]

Rappel= TP/(TP+FN)

1.15.4. F1Score [35]

Le F1-score est une métrique pour évaluer la performance des modèles de classification à 2 classes ou plus. Il est particulièrement utilisé pour les problèmes utilisant des données déséquilibrées comme la détection de fraudes ou la prédiction d'incidents graves.

Le F1-score permet de résumer les valeurs de la précision et du rappel en une seule métrique. Mathématiquement, le F1-score est défini comme étant la moyenne harmonique de la précision et du ra ppel, ce qui se traduit par l'équation suivante :

$$F1Score = \frac{2*Précision*Rappel}{Précision+Rappel} = \frac{2*TP}{2*TP+FP+FN}$$

1.16. Conclusion

Dans ce chapitre, nous avons présenté divers concepts liés à la sécurité informatique. Nous sommes particulièrement intéressés aux systèmes de détection des intrusions, car ils jouent un rôle complémentaire aux méthodes de sécurité conventionnelles. Notre objectif principal dans ce chapitre est de mettre en évidence l'importance des systèmes de détection des intrusions Afin que les IDS soient performantes en détection, il faut que le modèles doit être choisi attentivement, sur la base des techniques de machines Learning. Dans le prochain chapitre, on présente les techniques de machine Learning.

Chapitre 2

L’apprentissage automatique

Chapitre 2 : L’apprentissage automatique

2.1. Introduction

L'apprentissage automatique est une branche de l'intelligence artificielle qui permet aux ordinateurs d'apprendre à partir de données et d'améliorer leurs performances sans être explicitement programmés. Il implique le développement d'algorithmes et de modèles capables d'identifier automatiquement des modèles, de faire des prédictions et de prendre des décisions basées sur des données. L'apprentissage automatique révolutionne les industries et fait progresser des domaines tels que la reconnaissance d'images, le traitement du langage naturel et les recommandations personnalisées.

2.2. Définition de L'intelligence artificielle

L'intelligence artificielle, souvent abrégée en IA, a été définie par l'un de ses créateurs, Marvin Lee Minsky, comme : "La construction de programmes informatiques qui permettent aux humains d'effectuer des tâches de manière plus satisfaisante car elles nécessitent un niveau de processus mental plus élevé", tels que : l'apprentissage perceptif, organisation de la mémoire et raisonnement critique [20]. Il est utilisé dans différents domaines d'application et utilisations potentielles telles que : compréhension du langage naturel, reconnaissance visuelle, robotique, systèmes autonomes. [21]

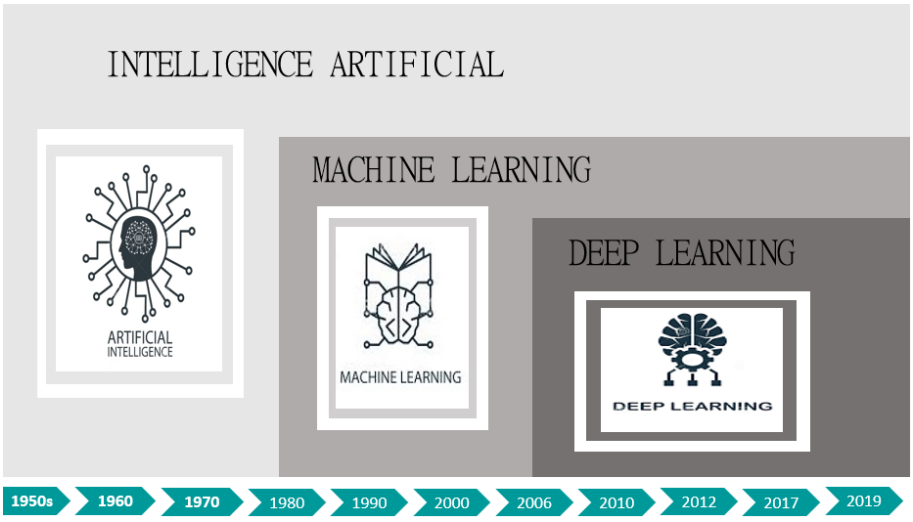

Figure 13: Intelligence Artificielle

2.3. Définition de L'apprentissage automatique

Apprentissage automatique, connu sous le nom de ML, le concept d'apprentissage automatique peut être réduit à l'une des branches émergentes de la science de l'intelligence artificielle (IA), qui repose sur la programmation d'ordinateurs de différentes formes pour pouvoir effectuer des tâches et exécuter les commandes assignées Ils s'appuient sur eux, s'approprient les données et les analysent tout en limitant l'intervention humaine. Il est important de noter que le terme apprentissage automatique a été inventé en 1959 à la demande du pionnier de l'intelligence artificielle Arthur Samuel dans le cadre de travaux dans les laboratoires IBM, et il est important de noter que les machines dans ce contexte doivent s'appuyer sur des données analytiques pré-saisies. [22]

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. [22]

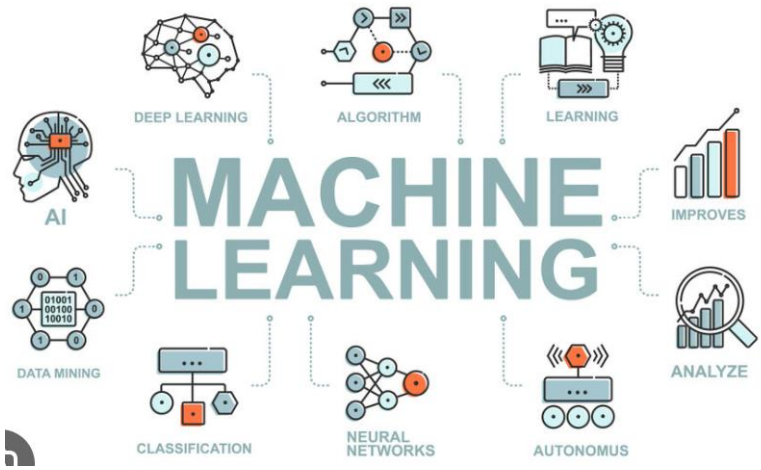

Figure 14: l'apprentissage automatique.

2.4. Types d’apprentissage

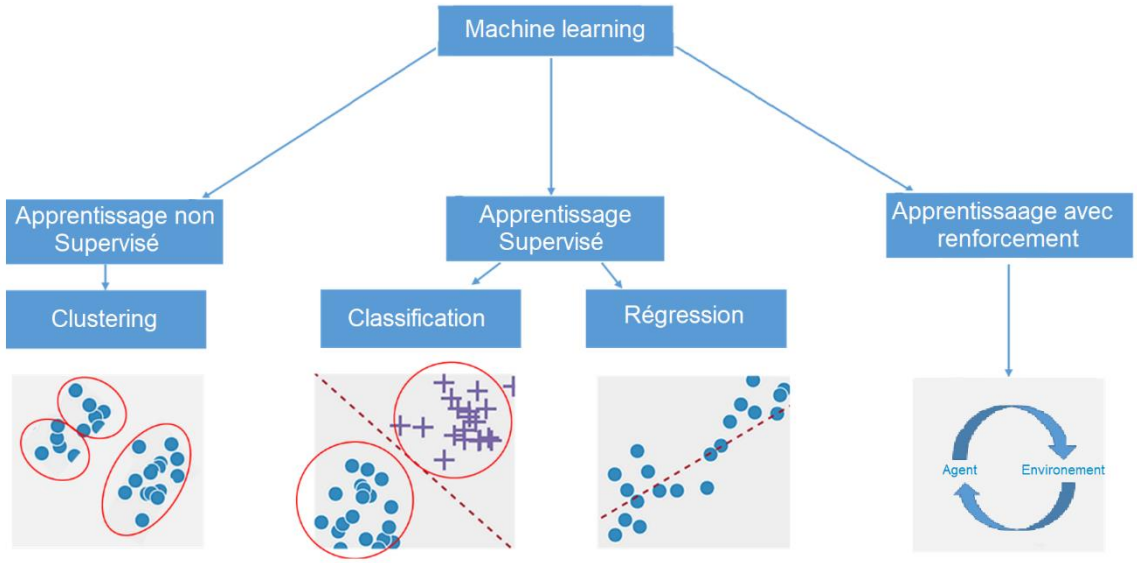

Figure 15 : Les types d’apprentissage automatique.

2.4.1. L’apprentissage Supervisé

Les machines auront 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 aidera inévitablement à terminer les tâches le plus rapidement possible. Les gens consomment pour accomplir des tâches. [22]

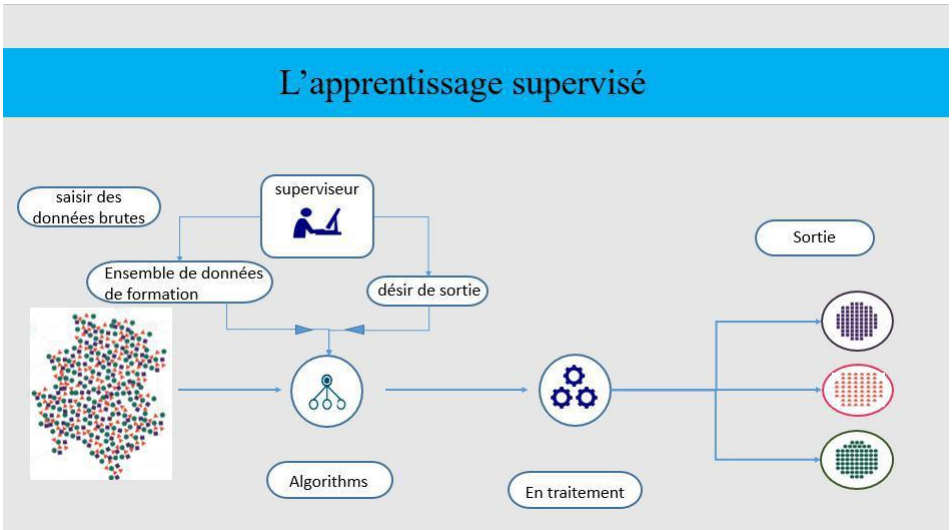

Figure 16: L’apprentissage Supervise.

L’apprentissage supervisé peut être utilisé pour deux types de problèmes qui sont :

2.4.1.1 Classification

La classification est le processus d'analyse et d'organisation d'un ensemble de données en classes similaires en fonction des caractéristiques des données. Il repose principalement sur des représentations classiques de données dont les limites de traitement sont connues et nécessitent dans la plupart des cas des temps de calcul importants. C'est au début des années 1960 que les méthodes de classification ont connu de nouveaux développements méthodologiques avec l'avènement des ordinateurs, qui ont permis l'émergence d'algorithmes d'analyse de données et de classification automatique. [24]

2.4.1.2 Régression

La régression est le processus de recherche d'un modèle ou d'une fonction qui distingue les données comme des valeurs réelles continues plutôt que d'utiliser des catégories ou des valeurs discrètes. Il peut également identifier les mouvements de distribution sur la base de données historiques. Étant donné que les modèles de prévision de régression

prédisent des quantités, la compétence du modèle doit être signalée comme une erreur dans ces prédictions. [28]

2.4.2. L’apprentissage non Supervisé :

Dans l'apprentissage non supervisé, les données ne sont pas étiquetées, de sorte que l'algorithme d'apprentissage découvre par lui-même les points communs entre ses données d'entrée. Les méthodes d'apprentissage automatique qui facilitent l'apprentissage non supervisé sont particulièrement utiles car les données non étiquetées sont plus abondantes que les données étiquetées.

L'objectif de l'apprentissage non supervisé peut être aussi simple que de découvrir des modèles cachés dans un ensemble de données, mais il peut aussi avoir un objectif d'apprentissage des caractéristiques, qui permet à la machine intelligente de découvrir automatiquement les représentations nécessaires pour classer les données brutes. [10]

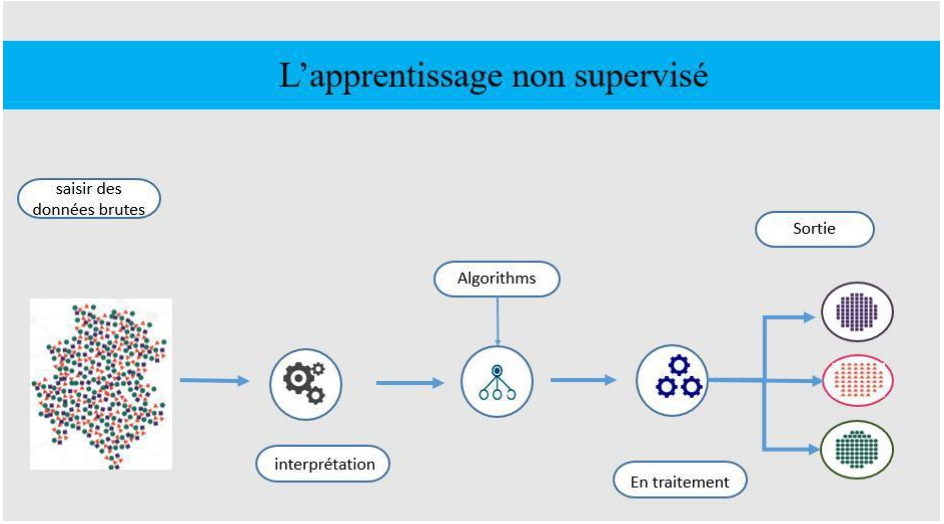

Figure 17: L’apprentissage Non Supervise.

L’apprentissage non supervisé peut être utilisé pour deux approches : {le regroupement (clustering) et l’association.}

2.4.2.1 Clustering

L'algorithme de clustering est un outil très utile en apprentissage automatique pour identifier des groupes de comportements similaires. Il permet de traiter et de classifier les données de manière à comprendre les différents types de clients en fonction de leurs

réactions à diverses stimulations externes, telles que l'envoi d'offres ou d'alertes. Grâce à un algorithme de clustering, il devient possible de découvrir automatiquement des groupes (ou clusters) de clients partageant des comportements similaires. Cela facilite le regroupement des audiences en fonction de leurs comportements similaires, de repérer facilement les « électrons libres » qui n’appartiennent pas à un groupe et même de découvrir les comportements inconnus a priori. Ainsi, il devient plus facile de personnaliser les services offerts et de fournir une expérience de qualité à ses clients. [32]

2.4.3. L’apprentissage par renforcement

L'apprentissage par renforcement (RL) est un domaine lié aux situations où un algorithme doit prendre des décisions où ces décisions ont des conséquences. Il intervient lorsque vous présentez à l'algorithme des exemples non étiquetés, comme dans l'apprentissage non supervisé. Toutefois, il est possible d'accompagner ces exemples d'une rétroaction positive ou négative en fonction des solutions proposées par l'algorithme.

Un exemple intéressant d'utilisation du RL se produit lorsque des ordinateurs apprennent à jouer à des jeux vidéo de manière autonome. Dans ce cas, l'algorithme explore différentes actions et observe les récompenses (positive ou négative) associées à ces actions pour ajuster ses stratégies et améliorer ses performances au fil du temps. Le RL permet ainsi à l'algorithme de développer des compétences et de prendre des décisions optimales dans des environnements complexes et interactifs. [11]

2.5. Les algorithmes des machines Learning

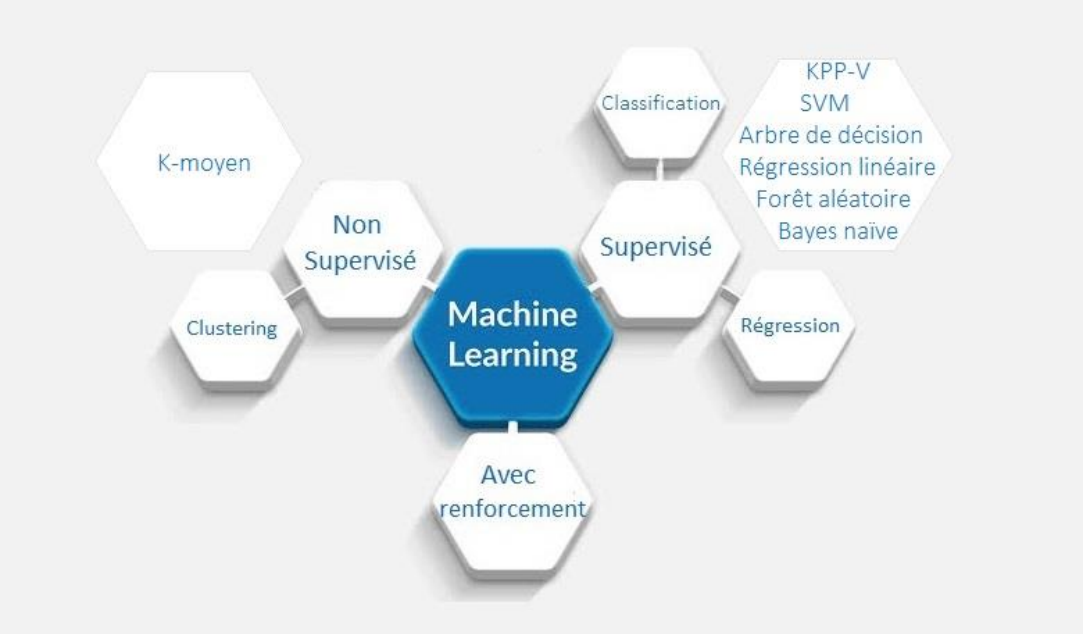

Figure 18: Les algorithmes da l’apprentissage automatique.

2.5.1. Algorithme de Classification

2.5.1.1 K-Plus Proches Voisins (K-PPV)

L'algorithme de k-plus proche voisin est un modèle de reconnaissance de modèle qui peut être utilisé pour la classification et la régression. Souvent abrégé en k-NN, le k est un entier positif, typiquement petit. Dans la classification ou la régression, l'entrée consistera en les k exemples d'entraînement les plus proches dans un espace.

L'algorithme KNN est l'un des plus simples de tous les algorithmes d'apprentissage automatique. Il est un type d'apprentissage basé sur l'apprentissage paresseux (lazy Learning). En d'autres termes, il ne nécessite pas de phase d'entraînement explicite, ou celle-ci est très limitée. Cela signifie que la phase d'entraînement est généralement rapide. [11]

Le fonctionnement de KPP-V se résume dans les étapes suivantes :

1. Charger les données

2. Initialiser **k** au nombre de plus proches voisins choisi

3. Pour chaque exemple dans les données : 3.1 Calculer la distance entre notre requête et l'observation itérative actuelle de la boucle depuis les données. 3.2 Ajouter la distance et l'indice de l'observation concernée à une collection ordonnée de données

4. Trier cette collection ordonnée contenant distances et indices de la plus petite distance à la plus grande (dans ordre croissant).

5. Sélectionner les k premières entrées de la collection de données triées (équivalent aux k plus proches voisins)

6. Obtenir les étiquettes des **k** entrées sélectionnées

7. Si **régression**, retourner la moyenne des **k** étiquettes

8. Si **classification**, retourner le mode (valeur la plus fréquente/commune) des **k** étiquettes. [25]

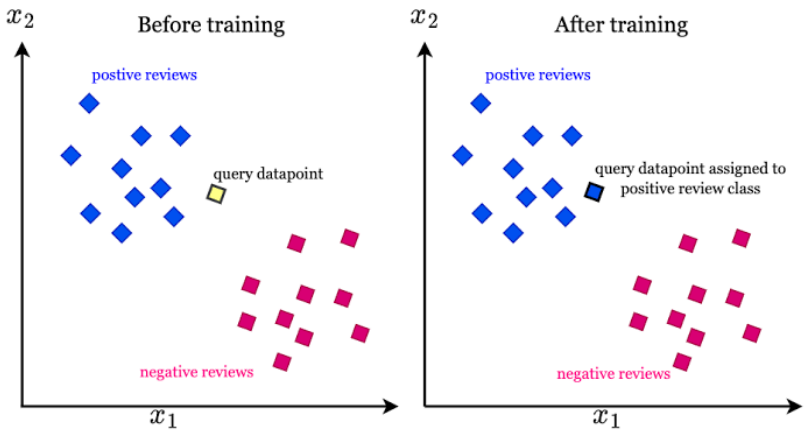

Figure 19 : K-Plus Proches Voisins.

2.5.1.2 Bayes Naïve

Les méthodes Naïve Bayes sont considérées parmi les modèles probabilistes les plus connus. Elles se basent principalement sur le théorème de Bayes (Bayes, 1963).

Ces algorithmes sont fréquemment utilisés pour la catégorisation et la classification de documents. Ils permettent d'estimer la probabilité de chaque classe parmi les exemples,

en fonction des caractéristiques d'un document donné, et attribuent ensuite à ce dernier la classe la plus probable. Ce processus est communément appelé "probabilités a priori" ou "prior probabilities". [26]

Étapes :

1. Lire l'ensemble de données d'apprentissage ;

2. Calculer la moyenne et l'écart type des variables prédictives dans chaque classe

3. Répéter le calcul de la probabilité de fi à l'aide de l'équation de densité de Gauss dans chaque classe ; jusqu'à ce que la probabilité de toutes les variables prédictives ait été calculée.

4. Calculer la vraisemblance pour chaque classe ;

5. Obtenir la plus grande vraisemblance. [27]

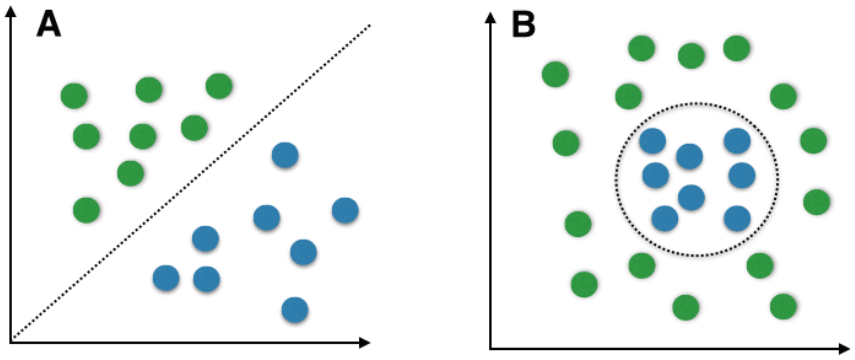

Figure 20 : Bayes Naïve

2.5.1.3 Machine vectorielle de soutien (SVM)

La machine à vecteur de support (SVM) est une autre technique d'apprentissage automatique de pointe très largement utilisée. Dans le domaine de l'apprentissage automatique, les machines à vecteurs de support sont des modèles d'apprentissage supervisés avec des algorithmes d'apprentissage associés qui analysent les données utilisées pour la classification et l'analyse de régression. En plus de permettre la

classification linéaire, les SVM peuvent également réaliser efficacement une classification non linéaire en utilisant ce qu'on appelle l'astuce du noyau. Cette méthode permet de cartographier implicitement les entrées dans des espaces de caractéristiques à haute dimension. L'objectif principal des SVM est de tracer des marges entre les différentes classes. Ces marges sont tracées de manière à maximiser la distance entre la marge et les classes, ce qui réduit au minimum les erreurs de classification. [27]

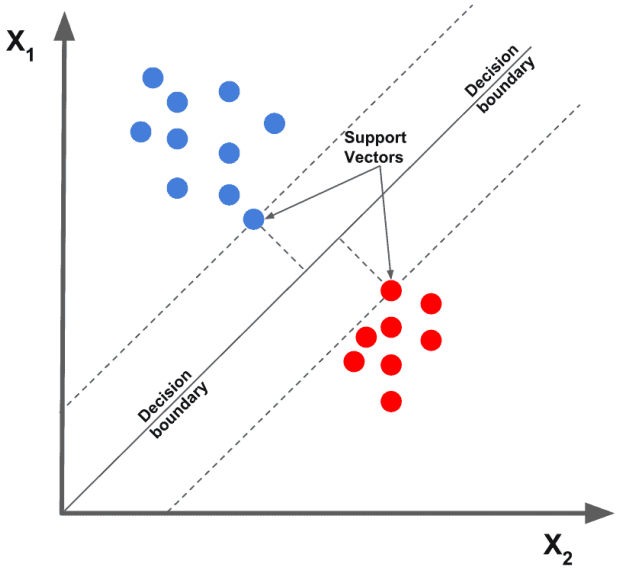

Figure 21: Support Vector Machines (SVM).

2.5.1.4 L’arbre de décision

Un arbre de décision est une représentation graphique des résultats potentiels d'une série de choix interconnectés. Il permet à une personne ou à une organisation d'évaluer différentes actions possibles en fonction de leurs coûts, probabilités et bénéfices. Il peut être utilisé pour faciliter une discussion informelle ou pour créer un algorithme qui détermine mathématiquement le meilleur choix.

Un arbre de décision commence généralement par un nœud initial à partir duquel plusieurs résultats possibles se déroulent. Chacun de ces résultats conduit à d'autres nœuds, qui à leur tour engendrent d'autres possibilités. La structure ainsi formée rappelle la forme d'un arbre, d'où le nom "arbre de décision". [31]

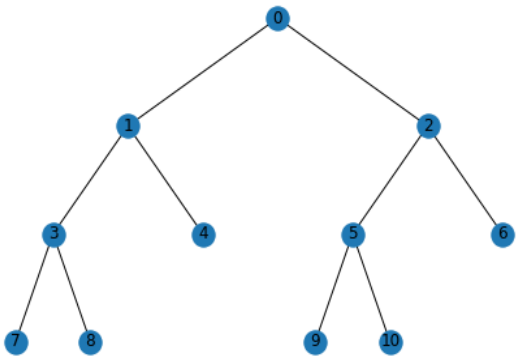

Figure 22 : Arbre de décision.

2.5.2. Algorithme de régression

2.5.2.1 La régression Linéaire

Les algorithmes de régression linéaire sont utilisés pour modéliser la relation entre des variables prédictives et une variable cible. Cette relation est représentée par une fonction mathématique de prédiction. Dans le cas le plus simple, appelé régression linéaire univariée, l'algorithme cherche à trouver une fonction sous forme de droite pour estimer la relation entre les variables. Cependant, dans le cas de la régression linéaire multivariée, plusieurs variables explicatives sont prises en compte dans la fonction de prédiction. Finalement, la régression polynomiale permet de modéliser des relations complexes qui ne sont pas forcément linéaires. [29]

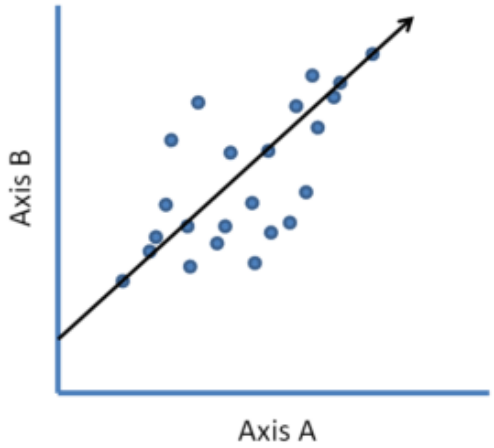

Figure 23 : La régression Linéaire .

2.5.3. Algorithme de Clustering

2.5.3.1 K-means (K-moyen)

L'un des premiers algorithmes d'apprentissage non supervisé est K-means. Il utilise une approche itérative pour produire un résultat final. Les entrées de l'algorithme sont le nombre de clusters souhaité et l'ensemble de données non étiquetées. L'ensemble de données se compose de caractéristiques pour chaque point de données. Les algorithmes commencent par les estimations initiales pour les K centroïdes, peuvent être générés de manière aléatoire ou directement à partir du dataset. [33]

Le fonctionnement de K-moyen se résume dans les étapes suivantes :

1. On choisit k objets au hasard qu’on considère comme des centres pour les classes initiales.

2. On affecte chaque objet au centre le plus proche pour obtenir une partition de k classes.

3. On recalcule les centres de chaque classe.

4. La répétition des étapes 2 et 3 jusqu’à la stabilité des centres. [8]

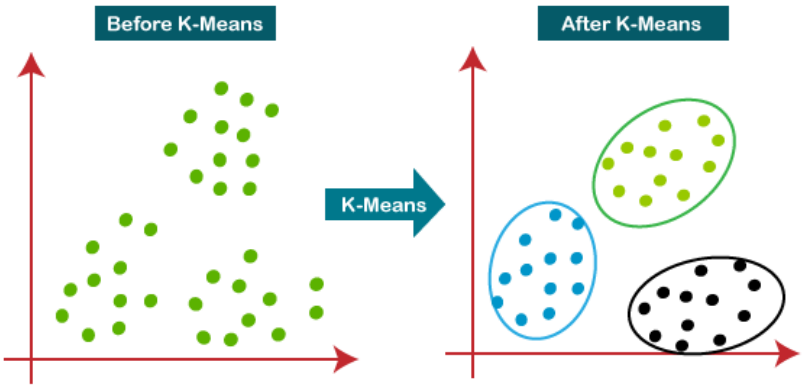

Figure 24 :K-moyen.

APPRENTISSAGE	ALGORITHMES		AVANTAGES	INCONVENIENTS
Supervisé	Classification	KNN	1. facile à implémente. 2. efficace. 3. L'algorithme est polyvalent	1. Calculer chaque fois la similarité entre les k. 2. grande capacité de stockage. 3. utilise de nombreuses données de références pour classifier les nouvelles entrées
		SVM	1. Leur capacité à manipuler de grandes quantités de données 2. Le faible nombre d'hyper paramètres. 3. Elles sont bien fondées théoriquement.	1. complexes pour la classification des corpus. 2. demande un temps énorme pendant les phases de test.
		Arbre de Décision	1. faciles à comprendre. 2. Ils permettent de sélectionner l'option la plus appropriée parmi plusieurs. 3. Il est facile de les associer à d'autres outils de prise de décision.	1. instables. 2. Certains concepts sont difficiles à exprimer à l'aide d'arbres de décision (comme XOR).
		Naïve Bayes	1. La facilité et la simplicité de leur implémentation. 2. Leur rapidité. 3. Les méthodes Naïve Bayes donnent de bons résultats.	1. faire le même travail de classification.
	REGRESSION	Linéaire	1. Simplicité d'interprétation. 2. facilité de calcul	Elle ne traite pas les valeurs manquantes de variables continues sensible aux valeurs hors norme de variables continue
Non Supervisé	CLUSTERING	K-means	1. Simple 2. Flexible 3. Efficace 4. Complexité temporelle.	1. Ensemble non optimal de clusters 2. Manque de cohérence 3. Limitation des calculs 4. Spécifiez les valeurs k

Tableau 3: Avantage et inconvénients des algorithmes de ML[10]

2.6. Conclusion

Ce chapitre a abordé les concepts fondamentaux de l'apprentissage automatique. On a tout d'abord défini l'apprentissage automatique et présenté ses différents types pour donner une vision globale de ce domaine. Ensuite, les algorithmes ont été expliqués en détail, en mettant l'accent sur certains algorithmes clés. Cette exploration des algorithmes a permis de comprendre leur fonctionnement, leur utilité et les problèmes qu'ils peuvent résoudre.

Chapitre 3

Données déséquilibrées

Chapitre 3 : Données déséquilibrées

3.1. Introduction

Les données déséquilibrées se réfèrent à une situation où la répartition des classes ou catégories dans un ensemble de données est fortement asymétrique, avec une classe étant nettement plus prédominante que les autres. Cette disparité des données pose des défis lors de la création de modèles prédictifs précis, car les algorithmes ont tendance à favoriser la classe majoritaire et à obtenir de faibles performances avec la classe minoritaire. Pour remédier à cela, des techniques spécialisées sont nécessaires pour assurer une représentation et un apprentissage équitable, ce qui conduit à des applications d'analyse de données et d'apprentissage automatique plus robustes.

3.2. Définition des données déséquilibrées

Les données déséquilibrées sont un terme utilisé pour caractériser un type particulier d'ensemble de données, ce qui pose des défis importants liés aux problèmes de classification. Il peut être utilisé pour diverses applications telles que le secteur financier, médical et public. Il n'y a pas de définition stricte, mais elle fait référence à des scénarios dans lesquels le nombre d'échantillons associés à chaque classe varie considérablement. [17]

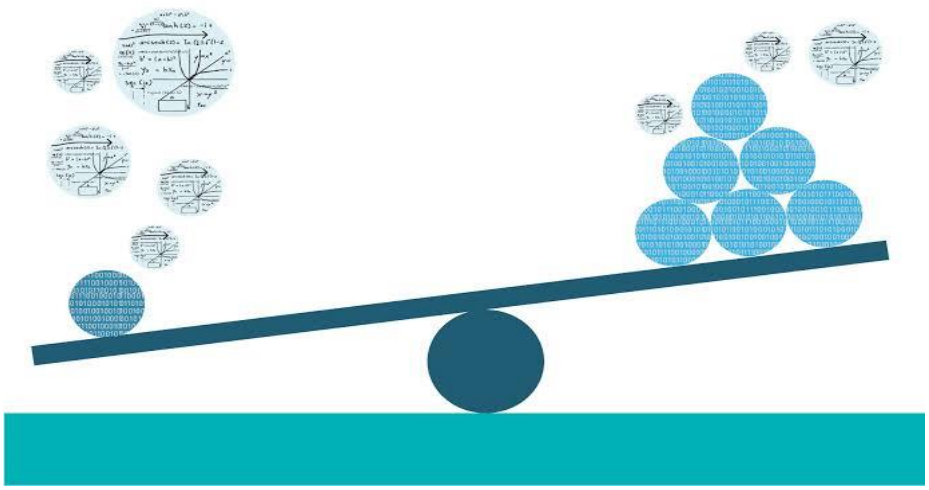

Figure 25: Les données déséquilibrée.

3.3. Comment utiliser les données déséquilibrées en Machine Learning ?

Une littérature abondante a émergé sur la correction des déséquilibres depuis le début des années 2000, et de nombreuses méthodes sont disponibles. Ces méthodes fonctionnent à trois niveaux.Au niveau des données (Solution au niveau des données), au niveau de l'algorithme (Solution au niveau de l'algorithme) ou au niveau de l'erreur de classification. [18]

3.3.1. Solutions au niveau des données

Ces solutions consistent à ré-échantillonner les données utilisées pour entraîner les algorithmes d'apprentissage automatique. Le but est de rééquilibrer la classe et de faciliter l'apprentissage. Il existe trois méthodes principales :

3.3.1.1 Le sous-échantillonnage

Les méthodes de sous-échantillonnage consistent à sélectionner ou à générer un ensemble S'0 à partir de l'ensemble d'apprentissage original de la classe majoritaire S0 tel que |S'0|<|S0|. Dans le cas de la sélection d'ensemble, les instances sont sélectionnées à partir des données initiales, où S'0 ⊂ S0. Cependant, lors de la génération d'ensemble, la méthode mise en œuvre réduit la classe majoritaire et génère synthétiquement l'ensemble d'apprentissage majoritaire, par exemple S′0⊄ S0.

Exemple de méthode de sélection

- Sous-échantillonnage aléatoire : un nombre prédéterminé d'instances de la classe majoritaire est sélectionné au hasard à partir de S0. Cette approche non heuristique est considérée comme naïve car aucune hypothèse n'est faite sur les données. Le risque de perte d'informations est élevé. [47]

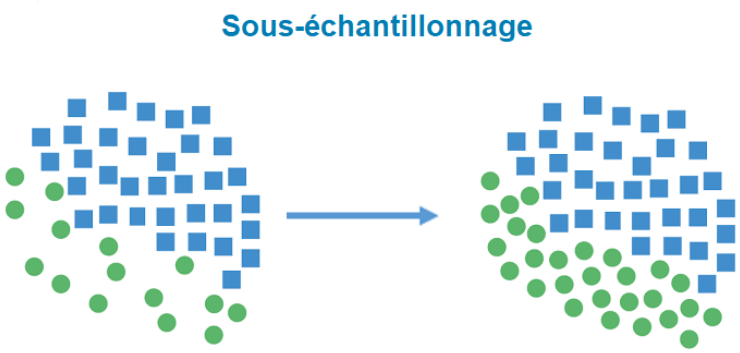

Figure 26: Sous-échantillonnage.

3.3.1.2 Le sur-échantillonnage

Le sur-échantillonnage consiste à générer des instances synthétiques de la classe minoritaire afin d'augmenter sa cardinalité. Il en résulte l'ensemble S'1 contenant l'ensemble d'apprentissage original pour la classe minoritaire : S1 ⊂ S'1. Les méthodes les plus courantes sont les suivantes :

- **exemple de méthode**

Le sur-échantillonnage aléatoire consiste à dupliquer aléatoirement des instances de la classe minoritaire en fonction du niveau d'équilibre souhaité. Cette approche peut provoquer ou augmenter le surajustement du classifieur. [47]

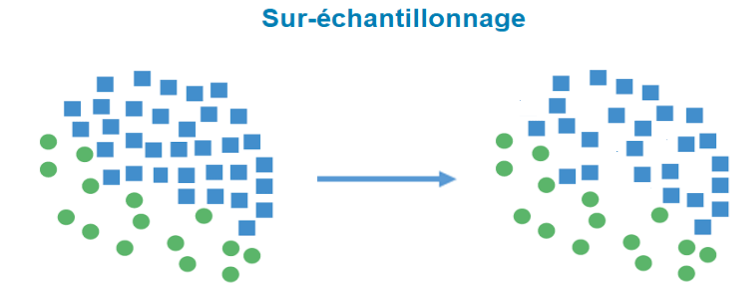

Figure 27: Sur-échantillonnage .

3.3.1.3 SMOTE (Synthetic Minority Oversampling Technique)

SMOTE est un algorithme de suréchantillonnage qui crée des observations synthétiques à partir d'observations existantes de la classe minoritaire. En fonction de la quantité de suréchantillonnage requise, SMOTE calcule les k voisins les plus proches pour créer des exemples synthétiques. [49]

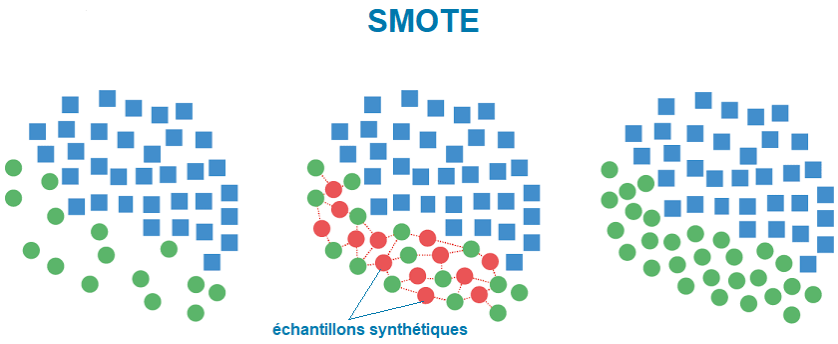

Figure 28: SMOTE

	Sous-échantillonnage	**Sur-échantillonnage**
Avantages	- Réduire la taille de l'ensemble des données. - Faible besoin de stockage. - Économise les coûts de calcul. - nécessite moins de temps d'exécution. - Aide à équilibrer l'ensemble de données	- Aucune perte d'informations utiles. - Ajouter aléatoirement plus d'observations minoritaires par réplication. - Aucune perte d'information.

Inconvénients	-Suppression aléatoire des bserv-ations de la classe majoritaire. -Supprimera des informations utiles tout en sous échantillonnant. - Les observations éliminées peuvent contenir des informations importantes peut entraîner une perte. - L'échantillonnage aléatoire peut être un échantillon équitable.	- Peut introduire du bruit supplémen-ntaire dans les données. -Risque de surajustement en raison de la copie des mêmes informations.
Quand utiliser	- Lorsque nous avons un bon nombre d'enregistrements dans notre classe minoritaire.	- Quand nous avons trop peu d'observation dans la classe minoritaire

Tableau 4:Comparaison entre sous-échantillonnage et le sous-échantillonnage.

3.4. Méthodes d'ensemble

Les méthodes d'ensemble sont des algorithmes d'apprentissage qui construisent un ensemble de classifieurs et effectuent une agrégation de leurs prédictions (voir figure 17). Cette idée suit le comportement de la nature humaine, qui tend à rechercher plusieurs opinions avant de prendre des décisions importantes (l'union fait la force) .

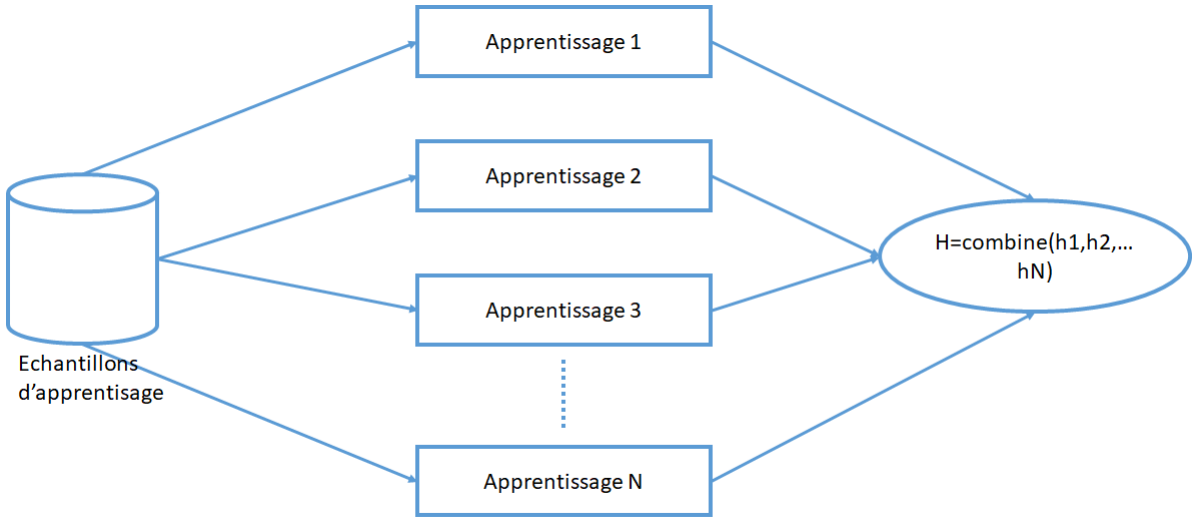

Figure 29: Principe général des méthodes d'ensemble.[19]

Les méthodes d'ensembles ont classées selon leur principe de fonctionnement en deux catégories : les méthodes d'ensemble séquentielles qui fonctionnent de manière itérative, et les méthodes d'ensemble parallèles qui fonctionne simultanément sur un ensemble de classificateurs

3.4.1. Les méthodes d'ensemble parallèle

Les méthodes d'ensembles parallèles correspondent à l'agrégation d'un ensemble de classificateurs qui fonctionnent simultanément.

➢ Bagging

La méthode du Bagging a été introduite par Breimen. Le mot bagging désigne la contraction des mots bootstrap et agrégation.

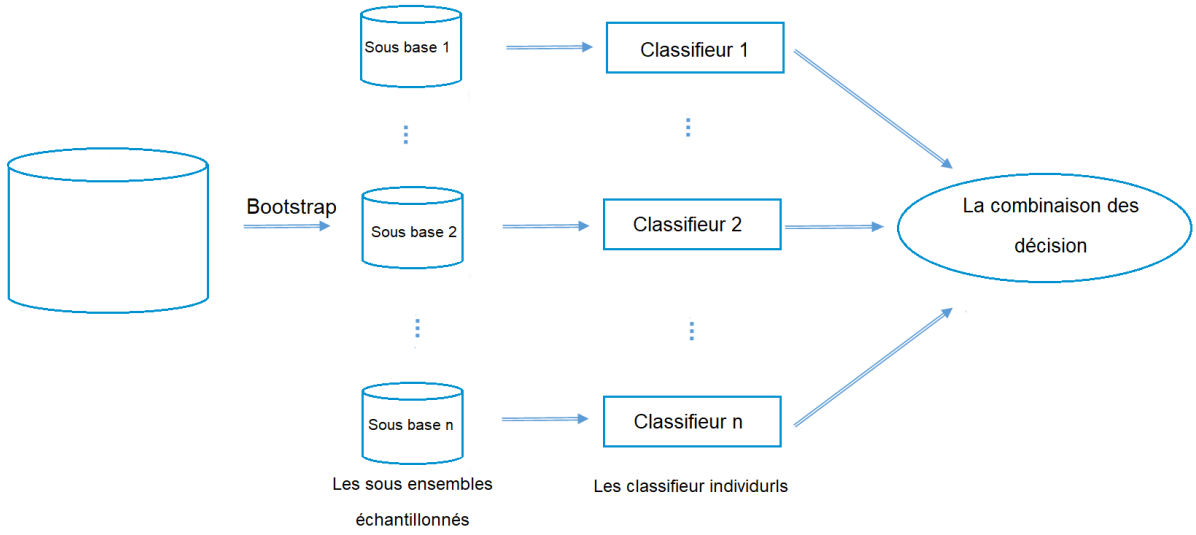

Figure 30: Principe de Bagging [19]

➢ Le Bagging

S'appuyer sur les méthodes bootstrap pour améliorer les prédictions du classifieur. Par conséquent, son principe est d'échantillonner uniformément les observations et de créer de nouveaux échantillons de manière actualisée, et de construire un classificateur sur

chaque échantillon. La prédiction finale est obtenue par vote majoritaire au sein des classifications intermédiaires (Voir figure 18).

Le principal avantage du bagging est de réduire l'instabilité du classificateur. Dont l'application du classificateur de type arbre de décision, car de petits changements dans la base apprise peuvent entraîner des changements importants dans la structure de l'arbre et donc dans ses performances de généralisation. Dans ce cas, la réduction de l'instabilité peut rendre les prédictions plus fiables et améliorer les performances de généralisation.

Il existe un autre point qui fait la force du bagging, ce sont les mesures out-of-bag. La mesure des Out-Of-Bag est un paramètre qui permet d'ajouter les instances non tiré par l'échantillonnage aléatoire avec remise. Ce paramètre introduit par la méthode bootstrap permet l'évaluation interne du classifeur et l'estimation de l'importance des variables pour la sélection de variables.

Les agorithmes du bagging

Bagging par arbre de décision et Bagging avec des forêts aléatoires, Bagging avec les SVM.

➢ Le Booststrap

Booststrap est un principe de rééchantillonnage statistique traditionnellement utilisé pour estimer des quantités ou des propriétés statistiques. L'idée du bootstrap est d'utiliser plusieurs ensembles de données rééchantillonnés à partir de l'ensemble de données observé et d'utiliser un tirage aléatoire avec remplacement (voir Figure 19). [19]

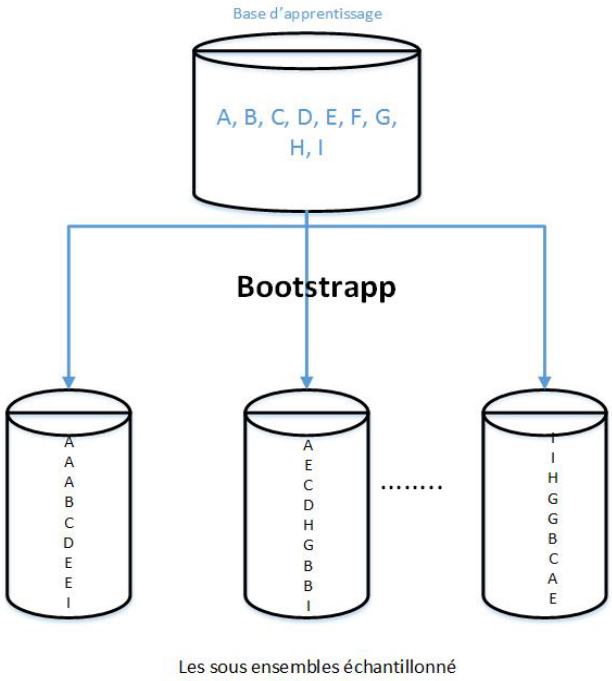

Figure 31: Principe Bootstrap *[19]*

3.4.2. Les méthodes d'ensembles séquentiels

Le boosting fonctionne de manière itérative et cyclique et est l'une des méthodes d'intégration séquentielle les plus réussies.

- Boosting

Le boosting est une méthode d'ensemble proposée par Schapire, dont le but est d'améliorer les performances des algorithmes d'apprentissage. En théorie, le Boosting peut considérablement améliorer les performances de tout algorithme d'apprentissage avec des fonctionnalités faibles, c'est-à-dire qu'il ne garantit que de renvoyer des classificateurs à haut risque mais moins de 50 %.

Le Boosting repose sur le même principe que le Bagging (voir Figure 20) : il construit un ensemble de classificateurs, qui sont ensuite agrégés par une moyenne pondérée des résultats. Cependant, dans le cas du Boosting, l'ensemble des classificateurs est construit de manière récursive et itérative. Plus précisément, chaque classificateur est

une version adaptée du classificateur précédent, donnant plus de poids aux observations mal prédites. [19]

- **Les algorithmes du boosting :**

AdaBoost et Gradient Boosting, XCBoost

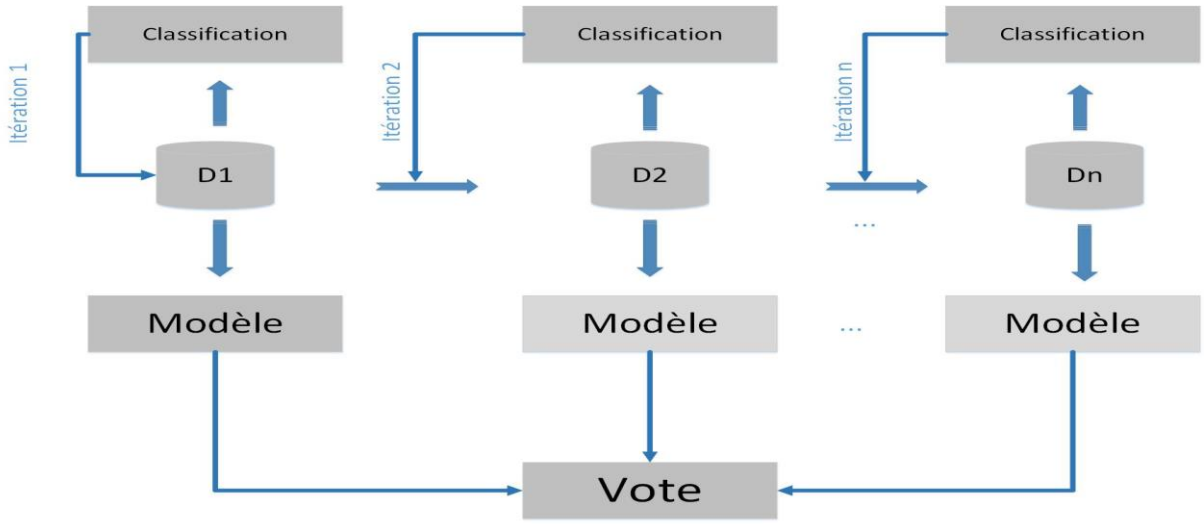

Figure 32: Principe de Boosting [19]

Bagging	Boosting
Aléatoire.	Adaptatif et généralement déterminist.
Utilise des échantillons Bootstrap.	Utilise échantillon initial au complet.
Les modèles ont le même poids.	Les modèles pondérés selon leur qualité d'ajustement.

Tableau 5: Bagging vs Boosting [19]

3.5. Conclusion

En conclusion, il est impératif de relever le défi posé par les données déséquilibrées pour développer des modèles prédictifs précis et fiables. En mettant en œuvre des techniques spécialisées pour garantir une représentation équitable et un apprentissage impartial, nous pouvons surmonter les limites imposées par les données déséquilibrées, ce qui se traduit par des applications plus robustes et plus efficaces de l'analyse des données et de l'apprentissage automatique.

PARTIE 2

Chapitre 4

Proposition du modèle

Chapitre 4 : proposition du modèle

4.1. Introduction

Ce chapitre présente notre approche du développement du problème il décrit et clarifie toutes les étapes pour réaliser notre approche. Il détaille le principe de fonctionnement et toutes les difficultés rencontrées dans l'application des techniques. La méthode proposée est évaluée à l'aide de l'ensemble de données CICIDS2017. L'ensemble de données est prétraité pour le rendre adapté à l'application de techniques d'apprentissage automatique.

4.2. Ensemble de données d'évaluation de détection d'intrusion (CICIDS2017)

L'ensemble de données CICIDS2017 contient des attaques bénignes et les attaques courantes les plus récentes, qui ressemblent aux véritables données du monde réel. Il comprend également les résultats de l'analyse du trafic réseau à l'aide de CICFlowMeter avec des flux étiquetés en fonction de l'horodatage, des IP source et de destination, des ports source et de destination, les protocoles et les l'attaques.

CICFlowMeter génère des flux bidirectionnels (Biflow), 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 FlowID, SourceIP, DestinationIP, SourcePort, DestinationPort et Protocol avec plus de 80 fonctionnalités de trafic réseau, dans ce tableau nous présentons chaque fonction avec sa description. [11]

Depuis la création de l'ensemble de données CICIDS2017, celui-ci a commencé à attirer les chercheurs pour l'analyse et le développement de nouveaux modèles et algorithmes. CI-CIDS2017, l'ensemble de données s'étend sur huit fichiers différents contenant cinq

jours de données de trafic normal et d'attaques de l'Institut canadien de cyber sécurité. Une brève description de tous ces fichiers est présentée dans le tableau 7. [36]

Nom des fichiers	Activité de jour	Attaques trouvées
Monday-WorkingHours.pcap_ISCX.csv	Monday	Benign (Normal human activities)
Tuesday-WorkingHours.pcap_ISCX.csv	Tuesday	Benign, FTP-Patator, SSH-Patator
Wednesday-workingHours.pcap_ISCX.csv	Wednesday	Benign, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, Heartbleed
Thursday-WorkingHours-Morning-WebAttacks.pcap_ ISCX.csv	Thursday	Benign, Web Attack – Brute Force, Web Attack – Sql Injection, Web Attack – XSS
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ ISCX.csv	Thursday	Benign, Infiltration

Friday-WorkingHours-Morning.pcap_ISCX.csv	Friday	Benign, Bot
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv	Friday	Benign, PortScan
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	Friday	Benign, DDoS

Tableau 5: Description des fichiers contenant l'ensemble de données CICIDS2017.

Le tableau 7 montre que l'ensemble de données contient des informations sur les attaques sous la forme de données de trafic sur cinq jours. Les données du jeudi après-midi et du vendredi conviennent bien à la classification binaire. De même, les données du mardi, du mercredi et du jeudi matin conviennent mieux à la conception d'un modèle de détection multiclasse. Il convient toutefois de noter qu'un modèle de détection optimal doit être capable de détecter les attaques de tout type. Par conséquent, pour concevoir un tel IDS typique, les données de trafic de tous les jours doivent être fusionnées pour former un seul ensemble de données à utiliser par l'IDS. C'est exactement ce que nous avons fait pour fusionner ces fichiers.

En fusionnant les fichiers présentés dans le tableau 1, nous avons trouvé la forme complète d'un ensemble de données qui contient 3119345 instances et 83 caractéristiques contenant 15 étiquettes de classe (1 normale + 14 étiquettes d'attaque). En outre, l'examen des instances des fichiers combinés a révélé que l'ensemble de données contient 288602 instances dont l'étiquette de classe est manquante et 203 instances dont l'information est manquante. En supprimant ces instances manquantes, nous avons obtenu un ensemble de données combiné de CI-CIDS2017 contenant 2830540 instances. À ce stade, nous avons recherché d'éventuelles instances redondantes. De manière surprenante, aucune instance redondante n'a été trouvée. Les

caractéristiques de l'ensemble de données combiné et l'occurrence détaillée par classe sont présentées dans le tableau 8 et le tableau 9. [36]

Remarque :

Nom de la fonction : les noms des fonctionnalités (features) présentes dans le jeu de données CICIDS2017.

Numéro des classes : le nombre de classes uniques présentes dans chaque colonne.

Nom de la fonction	Description	Numéro des classes
Flow ID	un identifiant unique attribué aux flux de réseau.	1085071
Source IP	L’adresse IP de source de la connexion.	17005
Source Port	Port de source de la connexion.	64640
Destination IP	L’adresse IP de destination de la connexion.	19112
Destination Port	Port de destination de la connexion.	53805
Protocol	Protocole utilisé lors de la connexion	3
Timestamp	Temps à laquelle la connexion a eu lieu	27965
Flow Duration	Durée du flux en microsecondes	1050899
Total Fwd Packets	Total des paquets dans le sens direct	1432
Total Backward Packets	Total des paquets dans le sens inverse	1747
Total Length of Fwd Packets	Taille totale des paquets dans le sens direct	17928
Total Length of Bwd Packets	Taille totale du paquet dans le sens inverse	64698
Fwd Packet Length Max	Taille maximale du paquet dans la direction avant	5279
Fwd Packet Length Min	Taille minimale du paquet dans la direction avant	384
Fwd Packet Length Mean	Taille moyenne du paquet dans la direction avant	109091

Fwd Packet Length Std	taille du paquet dans la direction avant	254384
Bwd Packet Length Max	Taille maximale du paquet en sens inverse	4838
Bwd Packet Length Min	Taille minimale du paquet dans le sens inverse	583
Bwd Packet Length Mean	Taille moyenne du paquet en sens inverse	154284
Bwd Packet Length Std	taille du paquet dans le sens inverse	249206
Flow Bytes/s	Nombre d'octets de flux par seconde	1595244
Flow Packets/s	Nombre de paquets de flux par seconde	1242273
Flow IAT Mean	Temps moyen entre deux paquets envoyés dans le flux	1170377
Flow IAT Std	Écart-type du temps entre deux paquets envoyés dans le flux	1057046
Flow IAT Max	Temps maximum entre deux paquets envoyés dans le flux	580289
Flow IAT Min	Temps minimum entre deux paquets envoyés dans le flux	136316
Fwd IAT Total	Temps total entre deux paquets envoyés dans le sens direct	493098
Fwd IAT Mean	Temps moyen entre deux paquets envoyés dans le sens direct	738963
Fwd IAT Std	Écart-type entre deux paquets envoyés dans le sens direct	700372
Fwd IAT Max	Temps maximum entre deux paquets envoyés dans le sens direct	437316
Fwd IAT Min	Temps minimum entre deux paquets envoyés dans le sens direct	110631
Bwd IAT Total	Temps total entre deux paquets envoyés en sens Inverse	414928

Bwd IAT Mean	Temps moyen entre deux paquets envoyés en sens inverse	671985
Bwd IAT Std	Écart-type du temps entre deux paquets envoyés dans le sens inverse	709055
Bwd IAT Max	Temps maximum entre deux paquets envoyés dans le sens inverse	368285
Bwd IAT Min	Temps minimum entre deux paquets envoyés dans le sens inverse	66074
Fwd PSH Flags	Nombre de fois où l'indicateur PSH a été activé dans les paquets circulant dans le sens direct (0 pour UDP)	2
Bwd PSH Flags	Nombre de fois où l'indicateur PSH a été activé dans les paquets voyageant dans le sens inverse (0 pour UDP)	1
Fwd URG Flags	Nombre de fois où l'indicateur URG a été activé dans des paquets circulant dans le sens direct (0 pour UDP)	2
Bwd URG Flags	Nombre de fois où l'indicateur URG a été activé dans les paquets voyageant dans le sens inverse (0 pour UDP)	1
Fwd Header Length	Total des octets utilisés pour les en-têtes dans le sens direct	3771
Bwd Header Length	Nombre total d'octets utilisés pour les en-têtes dans le sens inverse	3945
Fwd Packets/s	Nombre de paquets aller par seconde	1222680
Bwd Packets/s	Nombre de paquets en sens inverse par seconde	1109941
Min Packet Length	Longueur minimale d'un paquet	215
Max Packet Length	Longueur maximale d'un paquet	5708
Packet Length Mean	Longueur moyenne d'un paquet	226511

Packet Length Std	Longueur de l'écart-type d'un paquet	413401
Packet Length Variance	Variance de la longueur d'un paquet	408238
FIN Flag Count	Nombre de paquets avec FIN	2
SYN Flag Count	Nombre de paquets avec SYN	2
RST Flag Count	Nombre de paquets avec RST	2
PSH Flag Count	Nombre de paquets avec PUSH	2
ACK Flag Count	Nombre de paquets avec ACK	2
URG Flag Count	Nombre de paquets avec URG	2
CWE Flag Count	Nombre de paquets avec CWR	2
ECE Flag Count	Nombre de paquets avec ECE	2
Down/ Patio	Taux de téléchargement et de téléversement	31
Average Packet Size	Taille moyenne des paquets	212207
Avg Fwd Segment Size	Taille moyenne observée dans le sens direct	99719
Avg Bwd Segment Size	Taille moyenne observée dans le sens inverse	147611
Fwd Header Length.1	Nombre total d'octets utilisés pour les en-têtes dans le sens direct	3771
Fwd Avg Bytes /Bulk	Nombre moyen d'octets en vrac dans le sens direct	1
Fwd Avg Packets /Bulk	Nombre moyen de paquets en vrac dans le sens direct	1
Fwd Avg Bulk Rate	Nombre moyen de paquets en vrac dans le sens direct	1
Bwd Avg Bytes /Bulk	Nombre moyen d'octets en vrac dans le sens inverse	1
Bwd Avg Packets /Bulk	Nombre moyen de paquets en vrac dans le sens inverse	1
Bwd Avg Bulk Rate	Nombre moyen d'octets en masse dans le sens inverse	1
Subflow Fwd Packets	Nombre moyen de paquets dans un sous-flux dans le sens direct	1432

Subflow Fwd Bytes	Nombre moyen d'octets dans un sous-flux dans le sens direct	17928
Subflow Bwd Packets	Nombre moyen de paquets dans un sous-flux dans le sens inverse	1747
Subflow Bwd Bytes	Le nombre moyen d'octets dans un sous-flux dans le sens inverse.	64738
Init_Win_bytes_forward	Nombre total d'octets envoyés dans la fenêtre initiale dans le sens direct	12151
Init_Win_bytes_backward	Nombre total d'octets envoyés dans la fenêtre initiale dans le sens inverse.	13112
act_data_pkt_fwd	Nombre de paquets contenant au moins 1 octet de données TCP dans le sens direct	1093
min_seg_size_forward	Taille minimale d'un segment observée dans le sens direct	28
Active Mean	Durée moyenne d'activité d'un flux avant qu'il ne devienne inactif	326583
Active Std	Écart-type du temps d'activité d'un flux avant l'inactivité	202829
Active Max	Durée maximale d'activité d'un flux avant qu'il ne devienne inactif	299565
Active Min	Durée minimale pendant laquelle un flux a été actif avant de devenir inactif	175670
Idle Mean	Durée moyenne d'inactivité d'un flux avant qu'il ne devienne actif	222137
Idle Std	Écart-type de la durée d'inactivité d'un flux avant qu'il ne devienne actif	197617
Idle Max	Durée maximale d'inactivité d'un flux avant qu'il ne devienne actif	149737
Idle Min	Durée minimale d'inactivité d'un flux avant qu'il ne devienne actif	223888

Label	représente la classification du trafic réseau en tant que normal ou lié à une intrusion ou à une attaque particulière.	15

Tableau 6: fonctionnalités de trafic réseau avec la description.

Encodage	Normal/ Attaque Labels	Numéro d'instances
1	BENIGN	2273097
2	Bot	1966
3	DDoS	128027
4	DoS GoldenEye	10293
5	DoS Hulk	231073
6	DoS Slowhttptest	5499
7	DoS slowloris	5796
8	FTP-Patator	7938
9	Heartbleed	11
10	Infiltration	36
11	Port Scan	158930
12	SSH-Patator	5897
13	Web Attack – Brute Force	1507
14	Web Attack – Sql Injection	21
15	Web Attack – XSS	652

Tableau 7: Occurrence des instances par classe dans l'ensemble de données CICIDs2017.

4.3. Le modèle proposé

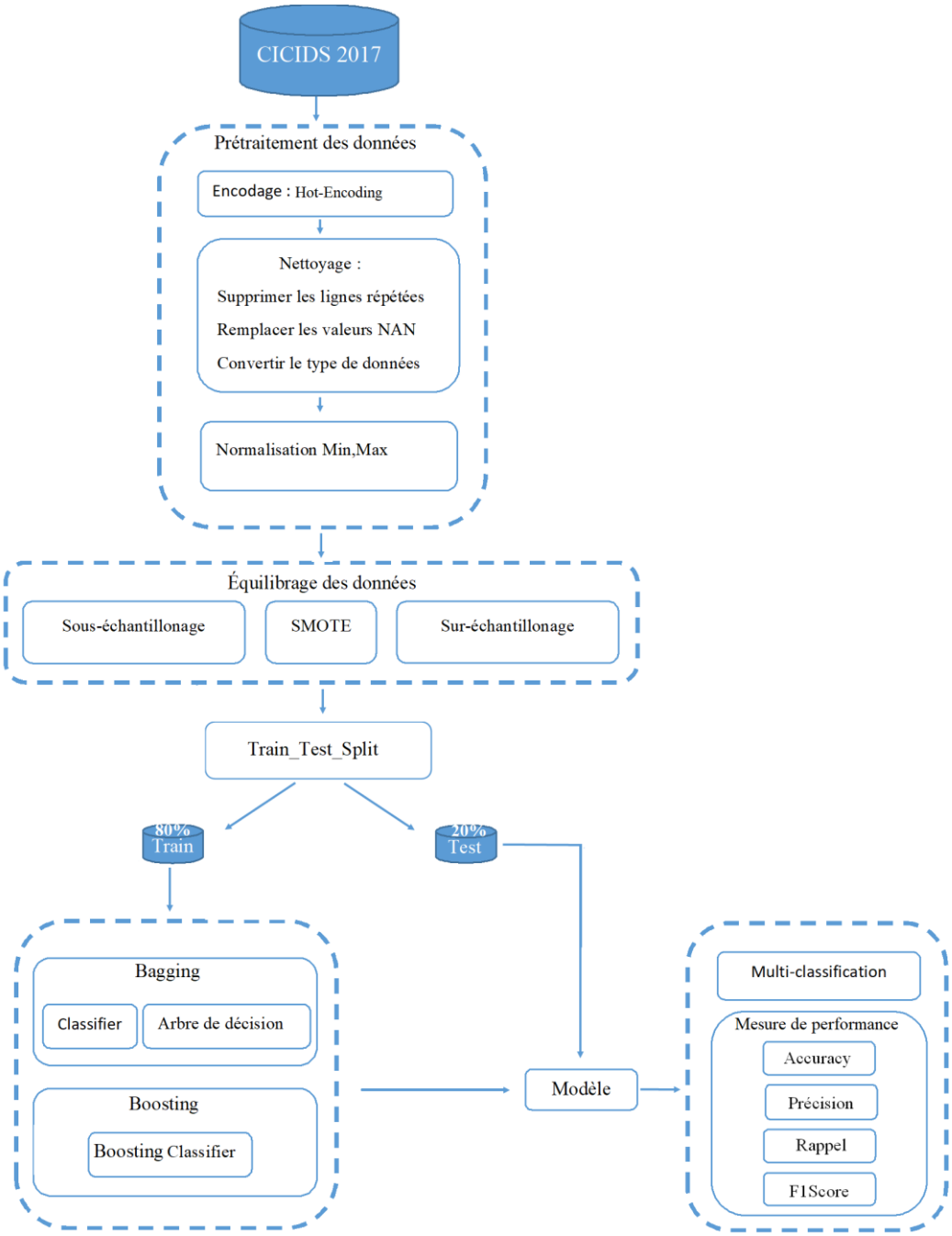

Figure 33: Schéma de la méthode de la conception.

4.4. Prétraitement

Le prétraitement des données est une technique d'exploration des données qui transforme les données brutes en formats compréhensibles, en s'attaquant à des problèmes tels que l'incomplétude, l'incohérence et les erreurs dans les grands ensembles de données. [48]

4.4.1. Encodage :

Hot encoding Le codage à chaud est une technique que nous utilisons pour représenter les variables catégorielles sous forme de valeurs numériques dans un modèle d'apprentissage automatique. [44]

4.4.1.1 Nettoyage des données

Assurer un nettoyage efficace des données est une étape cruciale avant l'analyse ou la modélisation des données, même si cette tâche peut s'avérer laborieuse. Nous avons commencé par supprimer les lignes en double en tant qu'étape initiale du processus de nettoyage des données. En outre, nous avons supprimé les espaces et remplacé les valeurs manquantes (NaN) par les moyennes des colonnes et convertir les colonnes "Object" en valeurs numériques.

4.4.1.2 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 palier à cela, des traitements préparatoires sur les données existent. Notamment le **Feature Scaling** qui comprend la **Standardisation** et la **Normalisation**.

Min-Max Scaling 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]. Le but d'avoir un tel intervalle restreint est de réduire l'espace de variation des valeurs d'une feature et par conséquent réduire l'effet de valeurs aberrantes.

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 [45]

4.4.2. Ré-équilibrage

En complément du choix d'un critère pertinent, il peut être intéressant de tenter de ré-équilibrer l'échantillon pour aider les algorithmes à mieux détecter les individus de la classe minoritaire. Les méthodes classiques consistent à créer de nouvelles observations de la classe minoritaire (oversampling) et/ou supprimer des individus de la classe minoritaire (undersampling).et SMOTE pour Synthetic Minority Over-Sampling Technique consiste à sur-échantillonner en se basant sur les proches voisins de la classe minoritaire. [46]

4.4.3. Étape de Split

Dans cette étape, nous divisons l'ensemble de données en 80 % de données d'entraînement et 20 % de données de Test.

4.4.4. Modèle

Dans notre modèle, nous avons utilisé des méthodes d'ensemble telles que Bagging et Boosting. Ces méthodes sont souvent utilisées sur les données déséquilibrées car elles peuvent améliorer la performance des modèles de prédiction en combinant plusieurs modèles individuels.

4.5. Conclusion

Dans ce chapitre nous avons détaillé le modèle généré pour l'apprentissage automatique, en se basant sur un jeu de données CICIDS2017 déséquilibrées.

Chapitre 5

Implémentation

Chapitre 5 : Chapitre implémentation

5.1. Introduction

Dans ce chapitre, nous commençons par introduire les outils et langages que nous avons utilisés pour mettre en œuvre notre modèle. Ensuite, nous détaillons les différentes étapes du travail sur la base de données, notamment le nettoyage et la normalisation des données. Enfin, nous avons présenté les expériences menées et fourni des captures d'écran illustrant l'exécution de notre application.

5.2. Matériel et logiciels utilisés

➢ Définition du langage Python en informatique

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. [37]

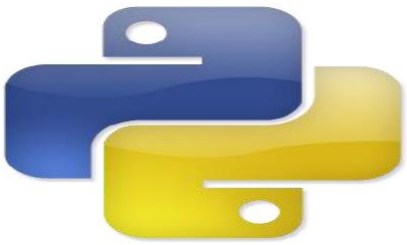

Figure 34: Logo Python.

➢ Définition de l'anaconda

Anaconda est un outil dont la distribution est **libre et open source.** Il est destiné à la programmation dans un environnement Python et R. Anaconda est largement utilisé en sciences de données, en intelligence artificielle ou Machine Learning. Cette distribution scientifique de Python renferme de nombreux packages nécessaires à l'analyse de données. Anaconda est également un gestionnaire d'environnement open source. [38]

Figure 35: Logo Anaconda.

➢ Définition jupyter

Le notebook est devenu un outil de travail extrêmement populaire dans les cercles académiques et scientifiques, en particulier pour gérer le code à l'état de prototype. Parmi les différents systèmes de notebook disponibles, Jupyter se distingue comme étant le plus largement adopté dans le monde de la science des données et au-delà. Son lancement initial a eu lieu en 2015, ce qui signifie qu'en l'espace de 5 ans seulement, cet outil s'est fermement établi comme un composant indispensable de la boîte à outils du scientifique des données. [39]

Figure 36: Logo Jupyter.

5.3. Bibliothèques Supplémentaires

➢ Pandas

La librairie Pandas est une librairie Python qui a pour objectif de vous faciliter la vie en matière de manipulation de données. C'est donc un élément indispensable qui faut maîtriser en tant que data scientiste. Les structures de données gérées par Pandas peuvent contenir tout type d'éléments à savoir (dans le jargon Pandas) des Séries et DataFrame et des Panel. Dans le cadre de nos expérimentations on utilisera plutôt les Dataframe car ils offres une vue bidimentionnelle des données (comme un tableau excel), et c'est exactement ce que l'on va chercher à utiliser pour nos modèles. [40]

➢ Matplotlib

Matplotlib est une bibliothèque complète permettant de créer des visualisations statiques, animées et interactives en Python. Matplotlib rend les choses faciles et les choses difficiles possibles.

1. Créez des graphiques de qualité professionnelle.
2. Créez des figures interactives qui peuvent être zoomées, panoramiques, mises à jour.
3. Personnaliser le style visuel et la mise en page.
4. Exporter vers de nombreux formats de fichiers.
5. Intégrer dans JupyterLab et les interfaces utilisateur graphiques.
6. Utiliser un large éventail de logiciels tiers basés sur Matplotlib [41]

➢ imblearn

Imbalanced-learn (importé sous le nom d'imblearn) est une bibliothèque open source, sous licence MIT, qui s'appuie sur scikit-learn (importé sous le nom de sklearn) et fournit des outils pour traiter la classification avec des classes déséquilibrées. [42]

➢ Scikit-learn :

Scikit-learn, une bibliothèque Python, fournit une large gamme d'algorithmes d'apprentissage supervisé et non supervisé. Elle s'appuie sur des technologies familières telles que NumPy, pandas et Matplotlib. Les fonctionnalités offertes par scikit-learn sont les suivantes :

- Régression, y compris la régression linéaire et logistique.

- Classification, y compris K-Nearest Neighbors pour la classification des voisins les plus proches.

- Le regroupement, y compris les algorithmes K-Means et K-Means++.

- Sélection de modèles.

- Prétraitement, y compris la normalisation Min-Max. [43]

5.4. Implémentation

5.4.1. Importation des bibliothèques

Pour effectuer les opérations décrites sur le jeu de données "cicids2017", nous pouvons utiliser des bibliothèques de manipulation de données populaires en Python. Voici une approche pour accomplir ces tâches :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import AdaBoostClassifier
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn import metrics
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
```

Figure 37. Importer les bibliothèques néccessaires.

5.4.2. Concaténation des tableaux

Pour charger le jeu de données CICIDS2017 dans un DataFrame pandas, nous pouvons utiliser la fonction appropriée de la bibliothèque pandas, en nous assurant que nous disposons du fichier de données dans un format compatible avec pandas et on faire la concaténation de la base de données. Voici comment procéder :

```
dataset1=pd.read_csv('Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv')
dataset2=pd.read_csv('Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv')
dataset3=pd.read_csv('Friday-WorkingHours-Morning.pcap_ISCX.csv')
dataset4=pd.read_csv('Monday-WorkingHours.pcap_ISCX.csv')
dataset5=pd.read_csv('Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX.csv')
dataset6=pd.read_csv('Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv')
dataset7=pd.read_csv('Tuesday-WorkingHours.pcap_ISCX.csv')
dataset8=pd.read_csv('Wednesday-workingHours.pcap_ISCX.csv')
```

```
db= pd.concat([dataset1,dataset2,dataset3,dataset4,dataset5,dataset6,dataset7,dataset8],axis="rows")
```

Figure 38 : Chargement de données .

5.4.3. Nettoyage des données

```
db=Dataset.drop_duplicates()
print('les lignes répétées ont été supprimées')
```

```
db.columns = db.columns.str.replace(' ','')
print('les espaces blancs ont été supprimées')
```

```
db.fillna(db.mean(), inplace=True)
print('les valeurs est charger')
```

```
assert db.isnull().sum().sum()==0 ,"pas de null valeur "
```

Figure 39: Nettoyage des données..

Pour convertir les colonnes "Object" en valeurs numériques, nous pouvons utiliser la fonction astype de la bibliothèque pandas. Voici comment procéder :

```
db=db.astype(float).apply(pd.to_numeric)
print('la conversion numérique est faite')
```

Figure 40: La conversion numérique des données..

5.4.4. Normalisation des données

```
from sklearn import preprocessing
db.replace([np.inf, -np.inf], np.nan, inplace=True)
scaler = preprocessing.MinMaxScaler(feature_range=(0, 1))
norm = scaler.fit_transform(db)
norm_df = pd.DataFrame(norm,columns=[db.columns])
print('Donnée original \n',db.head())
print('Données normalisées par MinMaxScaler() \n',norm_df.head())
```

Figure 41: Normalisation des données..

5.4.5. Équillibrage des données

Le sur-échantillonnage aléatoire consiste à sélectionner des exemples aléatoires de la classe minoritaire avec remplacement et à compléter les données d'apprentissage avec plusieurs copies de cette instance, de sorte qu'il est possible qu'une instance unique soit sélectionnée plusieurs fois.

Néanmoins, le sur-échantillonnage est une solution tout à fait décente et doit être testé.

Voici comment nous pouvons l'implémenter en Python.

```
undersampler = RandomUnderSampler()
X_resampled, y_resampled = undersampler.fit_resample(X, y)
```

Figure 42: Sous-échantillonnage aléatoire.

Une approche pour le sur-échantillonnage consiste à générer de nouveaux échantillons pour la classe minoritaire en effectuant un échantillonnage avec remplacement. La bibliothèque imblearn propose le RandomOverSampler, qui offre cette fonctionnalité.

Néanmoins, le sous-échantillonnage est une solution tout à fait décente et doit être testé.

Voici comment nous pouvons l'implémenter en Python

```
oversampler = RandomOverSampler()
X_resampled, y_resampled = oversampler.fit_resample(X, y)
```

Figure 43 : Sur-échantillonnage aléatoire.

SMOTE (Synthetic Minority Over-sampling Technique) est un algorithme populaire utilisé pour générer des exemples synthétiques d'échantillons de classes minoritaires, ce qui permet d'équilibrer l'ensemble de données. Pour utiliser SMOTE dans Jupyter Notebook.

```
# Apply SMOTE to the training set
smote = SMOTE()
x_resampled, y_resampled = smote.fit_resample(x,y)
```

Figure 44 : Algorithme SMOTE .

5.4.6. Étape de Split

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=1)
```

Figure 45 :Séparation des données.

5.5. Les expériences

Pour améliorer les mesures de performance telles que l'exactitude, la précision, le rappel et le score F1, nous avons mené trois expériences sur notre ensemble de données.

- Expérience 1 :

Dans la première expérience, nous avons appliqué une technique de données déséquilibrées appelée souséchantillonnage avec un arbre de décision pour obtenir des mesures de performance, et voici les résultats :

	Accuracy	precision	recall	f1_score
Expérience 1	83.33%	88.09%	88.09%	84.76%

Tableau 8: Sous-échantillonnage avec arbre de décision.

- Expérience 2 :

Dans la deuxième expérience, nous avons appliqué le sur-échantillonnage à l'arbre de décision et obtenu les résultats suivants :

	Accuracy	precision	recall	f1_score
Expérience 2	99.99%	99.99%	99.99%	99.99%

Tableau 9: Sur-échantillonnage avec arbre de décision.

- Expérience 3 :

Dans la troisième expérience, nous avons travaillé sur l'application de la technique SMOTE avec l'arbre de décision et voici les résultats obtenus.

	Accuracy	precision	recall	f1_score
Expérience 3	99.97%	99.97%	99.97%	99.97%

Tableau 10: SMOTE avec arbre de décision.

- Expérience 4 :

Dans cette expérience, nous avons appliqué le sous-échantillonnage avec le classificateur Bagging et nous avons obtenu les résultats suivants :

	Accuracy	precision	recall	f1_score
Expérience 4	94.44%	95.23%	96.42%	95.10%

Tableau 11: Sous-échantillonnage avec bagging classifier.

- Expérience 5 :

Dans cette expérience, nous avons également appliqué le sous-échantillonnage avec le classificateur Boosting et nous avons obtenu des résultats différents :

	Accuracy	precision	recall	f1_score
Expérience 5	83.33%	88.09%	84.52%	82.72%

Tableau 12: Sous-échantillonnage avec boosting classifier.

- Expérience 6 :

Dans cette expérience, nous avons utilisé la technique SMOTE avec le classificateur Bagging et voici les résultats de cette expérience :

	Accuracy	precision	recall	f1_score
Expérience 6	99.97%	99.97%	99.97%	99.97%

Tableau 13 : SMOTE avec bagging classifier.

Dans les trois expériences suivantes, nous n'avons pas obtenu de résultats car l'exécution n'a pas été terminée avec succès.

- Expérience 7 :

	Accuracy	precision	recall	f1_score
Expérience 7	/	/	/	/

Tableau 14 :SMOTE avec boosting classifier.

- Expérience 8 :

	Accuracy	precision	recall	f1_score
Expérience 8	/	/	/	/

Tableau 15: Sur-échantillonnage avec bagging classifier.

- Expérience 9 :

	Accuracy	precision	recall	f1_score
Expérience 9	/	/	/	/

*Tableau 16 :*Sur-échantillonage avec Boosing classifier.

Enfin, nous comparons les performances de différents modèles d'apprentissage automatique sur les tâches de détection d'intrusion sur la base de données CICIDS2017. Les résultats de nos tests montrent que parmi les modèles testés, le sous-échantillonnage d'arbre de décision obtient le score F1 le plus élevé. Cependant, il convient de noter que les performances du modèle peuvent varier en fonction du type d'attaque, et certains types d'attaques rares peuvent être difficiles à détecter en raison de la nature incomplète

des données. Par conséquent, afin d'augmenter l'efficacité des systèmes de détection d'intrusion, les données doivent être soigneusement sélectionnées et prétraitées, et des techniques appropriées sont utilisées pour traiter les problèmes d'équilibre des données.

5.6. Conclusion

Dans ce chapitre, nous avons d'abord présenté les différents outils, langages et bibliothèques que nous avons utilisés pour mettre en œuvre notre modèle. Ensuite, nous avons présenté les résultats des expériences sur le data set, et l'expérience la plus réussie qui a donné un bon résultat est le Sous-échantillonnage avec arbre de décision.

Conclusion générale

Dans ce mémoire, nous avons proposé un système de détection d'intrusion visant à détecter les attaques en abordant le défi des données déséquilibrées sur un jeu de données récent appelé CICIDS2017, où les classes rares peuvent être sous-représentées.

Nous avons réalisé neuf expériences au cours desquelles nous avons appliqué diverses techniques de gestion des données déséquilibrées, telles que le sur-échantillonnage, le sous-échantillonnage et SMOTE, afin d'améliorer les mesures de performance. De plus, nous avons combiné ces techniques avec des méthodes d'ensemble, ce qui a conduit à une amélioration des résultats. Toutefois, certaines expériences n'ont pas donné de résultats probants en raison de contraintes matérielles.

Sur la base de l'ensemble de ces expériences, nous avons démontré que la combinaison des techniques de traitement des données déséquilibrées avec les méthodes d'ensemble a produit des résultats remarquables en termes de mesure de performance, avec un score F1 de 99,99%.

Bibliographie

[1] Bouras, Ikram, et Khadra Fethallah. Un Système de Détection D'Intrusion pour les Smart Grids. Universite laarbi tebessi tebessa, 2017. dspace.univ-tebessa.dz:8080, http://dspace.univtebessa.dz:8080/jspui/handle/123456789/http//localhost:8080/jspui/handle/123456789/171.

[2] Guy Pujolle, « Les réseaux », Eyrolles ; 2008.

[3] D.E Denning. ‖An intrusion dection model‖ In :proccedings of the IEEE Transactions on software engineering, Septembre 2007 .

[4] Mimoune, Zakarya, et Abdelwhab /. promoteur Ouahab. Développement d'une Architecture Basée sur l'Apprentissage Profond (Deep Learning) pour la Détection d'Intrusion dans les Réseaux. Université Ahmed
Draïa -Adrar, 1 juillet 2019.https://dspace.univadrar.edu.dz/jspui/handle/123456789/3175 Accède le 30/01/2023

[5] https://www.diva-portal.org/smash/get/diva2:1324795/FULLTEXT01.pdf

[6] Vern Paxson. Bro : a system for detecting network intruders in real-time. Computer Networks,1999. Accède le 30/01/2023

[7] https://d1n7iqsz6ob2ad.cloudfront.net/document/pdf/5385dfe5c5ee2.pdf

[9] Abdelhalim Zaidi. Recherche et détection des patterns d'attaques dans les réseaux IP _a hauts débits. Réseaux et télécommunications [cs.NI]. Université d'Evry-Val d'Essonne, 2011.

[8] Bourouba, Hadjer,et Ouidad Chaouche.Optimisation des IDS du Cloud Computing par les techniques de machines Learning .Université Ibn Khaldoun-Tiaret-,2020 .dspace.univ-tiaret.dz,http://dspace.univ-tiaret.dz :80/handle/123456789/5364

[9] Abid, Malika, et Mustapha Mohamed Amine Beneddine. Conception d'un IDS basé sur le Deep Learning etRBN. Université Ibn Khaldoun -Tiaret-, 2021. dspace.univ-tiaret.dz, https://dspace.univtiaret.dz:80/handle/123456789/5522. Accède le 30/01/2023

[10] Dahmani, Mostefa, et Kamel Ouardani. Proposition d'un outil d'assistance pour la construction des systèmes de détection d'intrusion. Université Ibn Khaldoun -Tiaret-, 2020. dspace.univ-tiaret.dz, http://dspace.univ-tiaret.dz:80/handle/123456789/5359. Accède le 31/05/2023

[11] Bourouba, Hadjer,et Ouidad Chaouche.Optimisation des IDS du Cloud Computing par les techniques de machines Learning .Université Ibn Khaldoun-Tiaret-,2020 .dspace.univ-tiaret.dz,http://dspace.univ-tiaret.dz :80/handle/123456789/5364

[12] H. GUILLAUME, Détection d'intrusions paramétrée par la politique de sécurité, soupélec, campus de rennes, équipe SSIR, 7 février 2005.

[13] Belkhatmi, Keltouma, and Ouarda Benamara. Mise en place d'un systéme de détection et de prévention d'intusion. Diss. Université de Bejaia, 2016

[14] Modi, Chirag, et al. « A Survey of Intrusion Detection Techniques in Cloud ». Journal of Network and Computer Applications, vol. 36, no 1, janvier 2013, p. 42 57. ScienceDirect, https://doi.org/10.1016/j.jnca.2012.05.003. Accède le 30/01/2023

[15] https://dbprog.developpez.com/securite/ids/#LIII-A-1 Accède le 04/06/2023

[16] Ferhat, Ikram. ALGORITHME NSGA-III pour la sélection Des fonctionnalités utilisée dans un problème de classification déséquilibrée. juin 2020. archives.univ-biskra.dz, http://archives.univ-biskra.dz:80/handle/123456789/15781. 04/06/2023

[17] TurinTech. « What Is Imbalanced Data and How to Handle It? » TurinTech AI, 4 janvier 2022, https://www.turintech.ai/what-is-imbalanced-data-and-how-to-handle-it/. Accède le 04/06/2023

[18] Tremblay, Charles. « Imbalanced data et Machine Learning ». Kobia, 19 janvier 2022, https://kobia.fr/imbalanced-data-et-machine-learning/. Accède le 04/06/2023

[19] Hadji, Zahra. Apprentissage des données en distribution déséquilibrée par les méthodes d'ensembles. 14 juin 2015. dspace.univ-tlemcen.dz, http://dspace.univ-tlemcen.dz//handle/112/10926.

[20] Https://www.oracle.com/fr/cloud/deep-learning-intelligence-artificielle.html Accès le 16/01/2020

[21] http://tpe-intelligence-artificielle-2013.e-monsite.com/pages/definition-de-l-intelligence-artificielle.html Accès le 16/01/2020 à 11 :52,»

[22] Tahar, Mohamed, et Abdelnaceur Djillali Saidi. Un effectif système de détection d'intrusion pour l'amélioration de la précision. Université Ibn Khaldoun -Tiaret-, 2021. dspace.univ-tiaret.dz, http://dspace.univ-tiaret.dz:80/handle/123456789/5503.

[23] Kateb, Nabila, et Taher Guerram. Une Approche multi agents pour les datd mining. 2011. bib.univ-oeb.dz:8080, http://localhost:8080/xmlui/handle/123456789/6755. Accès le 06/06/2023

[24] https://depot-e.uqtr.ca/1201/1/030110265.pdf Accès le 06/06/2023

[25] « Machine Learning : l'algorithme des k plus proches voisins ». MonCoachData, 19 septembre 2019, https://moncoachdata.com/blog/algorithme-des-k-plus-proches-voisins/. Accède le 06/06/2023

[26] https://depot-e.uqtr.ca/1201/1/030110265.pdf
06/06/2023

[27] https://www.researchgate.net/publication/344717762_Machine_Learning_Algorithms_-A_Review. Accède le 06/06/2023

[28] ML | Classification vs régression – StackLima. 5 juillet 2022, https://stacklima.com/ml-classification-vs-regression/. Accède le 06/06/2023

[29] Benzaki, Younes. « 9 Algorithmes de Machine Learning que chaque Data Scientist doit connaitre ». Mr. Mint : Apprendre le Machine Learning de A à Z, 30 juin 2017, https://mrmint.fr/9-algorithmes-de-machine-learning-que-chaque-data-scientist-doit-connaitre.

[30] Algorithme N°4 - La régression linéaire pour comprendre les grands principes du Machine Learning ». Devoteam France, https://france.devoteam.com/paroles-dexperts/algorithme-n4-la-regression-lineaire-pour-comprendre-les-grands-principes-du-machine-learning/. Accède le 29 mai 2023.

[31] « Qu'est-ce qu'un arbre de décision ? » Lucidchart, https://www.lucidchart.com/pages/fr/arbre-de-decision. Accède le 6 juin 2023.

[32] https://invenis.co/blog/3-algorithmes-de-machine-learning-bien-utiles-business/

[33] Zeradna, Rim, et Imen Chorfi. L'utilisation de l'apprentissage automatique pour la détection des attaques Déni de Service (DOS) dans les réseaux de capteurs sans fil. Université Ibn Khaldoun -Tiaret-, 2022. dspace.univ-tiaret.dz, http://dspace.univ-tiaret.dz:80/handle/123456789/5720.

[34] Akhras, Mousa, et al. « Using Machine Learning to Build a Classification Model for IoT Networks to Detect Attack Signatures ». International Journal of Computer Networks &

Communications (IJCNC), vol. 12, nº 6, novembre 2020, p. 99. aircconline.com, https://doi.org/10.5121/ijcnc.2020.12607.

[35] Tremblay, Charles. « F1-score & F-beta score, compromis entre Precision et Recall en classification ». Kobia, 17 novembre 2021, https://kobia.fr/classification-metrics-f1-score/.

[36] « Ijet 22797 | PDF | Denial Of Service Attack | Statistical Classification ». Scribd, https://www.scribd.com/document/403095164/IJET-22797. Accède le 6 juin 2023.

[37] «https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/Accède le 18/05/2023.

[38] Anaconda pour Python - Présentation et installation | Jedha. https://www.jedha.co/formation-python/ananconda-python. Accède le le 29 mai 2023.

[39] Aubert, Alan. « Qu'est-ce que Jupyter et comment faire plus avec vos notebooks ? » Saagie, 20 novembre 2020, https://www.saagie.com/fr/blog/quest-ce-que-jupyter-et-pourquoi-est-il-un-outil-incontournable/.

[40] Cayla, Benoit. « Python Pandas - Tuto (Partie N°1) ». datacorner par Benoit Cayla, 4 mai 2018, https://datacorner.fr/pandas-1/. Accède le 29/05/2023

[41] https://matplotlib.org/ Accède le 29/05/2023

[42] imbalanced-learn documentation — Version 0.10.1. https://imbalanced-learn.org/stable/. Accède le 29 mai 2023

[43] What Is Scikit-Learn? » Codecademy, https://www.codecademy.com/article/scikit-learn. Accède le 29 mai 2023.

[44] « One Hot Encoding in Machine Learning ». *GeeksforGeeks*, 12 juin 2019, https://www.geeksforgeeks.org/ml-one-hot-encoding-of-datasets-in-python/.

[45] «Data Preprocessing : Feature Scaling avec Python et Sickit Learn ». Mr. Mint : Apprendre le Machine Learning de A à Z, 12 octobre 2017, https://mrmint.fr/data-preprocessing-feature-scaling-python.

[46] Rouvière, Laurent. Chapitre 7 Données déséquilibrées | Machine learning. lrouviere.github.io, https://lrouviere.github.io/TUTO_ML/dondes.html. Consulté le 20 juin 2023.

[47] Ajakan, N. (2022). Ingénierie de la représentation des variables pour la classification binaire à partir des données déséquilibrées (Doctoral dissertation, Université Laval).

[48] Johnny. « Prétraitement des données dans l'apprentissage automatique ». Blog ARC Optimizer, 7 octobre 2022, https://blog.arcoptimizer.com/pretraitement-des-donnees-dans-lapprentissage-automatique.

[49] Hamouda, Djallel. Un système de détection d'intrusion pour la cybersécurité. Working Paper,2020.dspace.univ-guelma.dz,http://dspace.univ-guelma.dz/jspui/handle/123456789/10125.