الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DR. YAHIA FARES DE MEDEA

FACULTE DES SCIENCES

Département du Mathématiques et de l'Informatique

Projet de fin d'études

Présenté pour l'obtention du Diplôme
De Master en Informatique

Spécialité : Ingénierie des Systèmes et Technologie de Web

THEME

Approche Automatique Pour La Classification Du Trafic Réseau

Proposé et dirigé par :

Mr. KHELDOUN Ahmed

Présenté par :

- Mestoui Ammar
- Sakhri karim

Année Universitaire : **2021/2022**

Résumé :

Les avancées technologiques et l'évolution rapide d'internet et des réseaux informatiques, y compris les réseaux d'entreprises, rendent nécessaire l'implémentation d'une stratégie de gestion de réseau afin de pouvoir optimiser l'utilisation des ressources, planifier le dimensionnement des infrastructures et opérer les systèmes de qualité de service et les mécanismes de sécurité. Ainsi, et pour atteindre cet objectif, il est indispensable de connaître le trafic véhiculant à travers le réseau.

De plus, il est très important pour les fournisseurs de services Internet « ISPs » de savoir quels types d'applications réseau circulent sur le réseau. La classification du trafic réseau est la première étape de l'analyse et de l'identification des différents types d'applications circulant dans un réseau

Il existe de nombreuses façons traditionnelles de classer le trafic réseau. Cependant, avec le nombre croissant d'applications sur Internet, il est devenu difficile de classer le trafic à l'aide de ces techniques.

L'objectif de ce mémoire est d'appliquer les techniques d'apprentissage automatique pour la classification du trafic réseau, pour celà, nous avons utilisé 5 algorithmes d'apprentissage automatiques à savoir : Arbre de décision, Forets Aléatoires, SVM, Naïve Bayes et K-Nearest Neighbors « KNN ». Enfin, nous avons effectué une comparaison entre ces algorithmes et la meilleur performance obtenue était de 99% en utilisant Forest Aléatoires.

Mots clés: trafic réseau, apprentissage automatique, apprentissage supervise, classification.

Abstract:

Technological advances and the rapid evolution of the Internet and computer networks, including business networks, make it necessary to implement a network management strategy in order to be able to optimize the use of resources, plan the sizing infrastructure and operate quality of service systems and security mechanisms. Thus, and to achieve this objective, it is essential to know the traffic conveying through the network.

In addition, it is very important for Internet Service Providers « ISPs » to know what types of network applications are circulating on the network. Network traffic classification is the first step in analyzing and identifying the different types of applications circulating in a network.

There are many traditional ways to classify network traffic. However, with the growing number of applications on the Internet, it has become difficult to classify traffic using these techniques.

The objective of this thesis is to apply machine learning techniques for the classification of network traffic, for this, we used 5 machine learning algorithms namely: Decision Tree, Random forest, SVM, Naive Bayes and K-Nearest Neighbors. Finally, we made a comparison between these algorithms and the best performance obtained was 99% using Random Forest.

Keywords: Network traffic, Machine Learning « ML », Supervised Learning, Classification.

ملخص:

يجعل التقدم التكنولوجي والتطور السريع للإنترنت وشبكات الكمبيوتر ، بما في ذلك شبكات الأعمال ، من الضروري تنفيذ استراتيجية لإدارة الشبكة من أجل التمكن من تحسين استخدام الموارد وتخطيط البنية التحتية و مساعدة مزودو خدمة الانترنت وكذلك فيما يتعلق بأمن المعلومات. ولتحقيق هذا الهدف ، من الضروري معرفة حركة البيانات و التطبيقات عبر شبكة الانترنت .

تصنيف تدفق البيانات في الشبكة هي الخطوة الأولى في تحليل وتحديد أنواع مختلفة من التطبيقات المتداولة في الشبكة.

هناك العديد من الطرق التقليدية لتصنيف حركة مرور البيانات عبر الشبكة. ومع ذلك ، مع تزايد عدد التطبيقات على الإنترنت ، أصبح من الصعب تصنيف حركة البيانات باستخدام هذه التقنيات. الهدف من هذا المشروع هو تطبيق تقنيات التعلم الآلي لتصنيف حركة مرور الشبكة ، لذلك استخدمنا 5 خوارزميات للتعلم الآلي وهي: decision tree، Random forest ، SVM، Naive bayse, KNN.

أخيرًا ، قمنا بإجراء مقارنة بين هذه الخوارزميات وكان أفضل أداء تم الحصول عليه 99٪ باستخدام Random Forest.

الكلمات الرئيسية: تدفق البيانات في الشبكة,التعلم ألي,التعلم الخاضع لالشراف,التصنيف.

Remerciements

Avant tout, nous remercions Dieu le tout puissant qui nous donnons la force et la volonté pour finir ce travail.
nous tiens à remercier énormément notre encadreur

DR. ***Kheldoun Ahmed,***

pour avoir accueilli et proposé ce sujet passionnant, etsurtout pour ses précieux conseils.

nous remercions vivement les ***membres de jury****, d'avoir accepté d'évaluer notre travail.*

Je désire aussi remercier les enseignants de l'université de médea,

qui m'ont fourni les outils nécessaires à la réussite de mes études universitaires
nous souhaitons remercier nous amis et notre collègues, notre familles qui sont toujours su être là dans tous les moments, son soutien et sa générosité et sa volonté ont toujours mérité mon plus profond respect.

Dédicace 1

Pour tous les moments de la vie, de l'échec et de la réussite.
Je dédie le fruit de notre travail à mes chers parents les
plus proches de notre cœur,
à leurs sacrifices et à leur
présence. Ils ont porté sur leurs frêles épaules toutes les
responsabilités et ont dépensé leur temps,
leur argent et leur
énergie sans compter.
Nous demandons à Dieu de protéger ma mère,
et d'avoir pitié de l'âme pure de mon père et de le faire entrer dans
ses vastes jardins.
A tous nos petits membres de la famille.
A tous nos amis proches, merci pour tous les moments
inoubliables.
Un grand merci pour tous les moments qu'on a passé
ensemble.

Karim.

Dédicace 2

*A **ma tendre Mère** : Tu représentes pour moi la Source de tendresse et l'exemple de dévouement qui n'a pas Cessé de m'encourager. Tu as fait plus qu'une mère puisse Faire pour que ses enfants suivent le bon chemin dans leur vie et leurs études.*

*A **mon très cher Père Ahmed** : Aucun dédicace Ne serait exprimé l'amour, l'estime et le respect que j'ai Toujours pour toi. Rien au monde ne vaut les efforts Fournis jour et nuit pour mon éducation et mon bien être.*

Ce travail est le fruit de tes sacrifices que tu as consenti Pour mon éducation et ma formation le long de ces années.

A mes très chers Frères, sœurs et beaux-frères.

A mes copines et toute ma famille.

A Mes meilleures amies

*A mon binôme **Karim***

Je ne peux nommer ici toutes les personnes qui de près ou deLoin m'ont aidés et encouragés mais je les en remercie vivement.

Ammar

Table of Contents

Résumé : II

Liste des figures XII

Liste des tables XIII

Liste des abréviations XIV

Introduction générale 1

Partie 01 : Etat de l'art **3**

Chapitre 01 : Classification du trafic réseau **4**

1 Introduction 4

2 Trafic réseau 4

3 Flux de réseau 5

3.1 Transmission de l'information 5

3.2 Modèle OSI 6

3.3 Modèle TCP/IP 8

3.3.1 Couche hôte réseau 8

3.3.2 Couche internet 9

3.3.3 Couche transport 9

3.3.4 Couche application 9

4 Classification de trafic réseau 9

4.1 Classification basée sur les ports 10

4.2 Classification par l'inspection de charge des paquets « DPI » 10

4.3 Classification basée sur l'analyse statistique des attributs 11

5 Conclusion 15

Chapitre 02 : Apprentissage automatique **16**

1 Introduction 16

2 Définition 17

3 Histoire d'apprentissage automatique 17

4 Les types d'apprentissages automatiques 18
4.1 Apprentissage supervisé 18
4.1.1 Classification 19
4.1.1.1 Classification binaire : 19
4.1.1.2 Classification multi-catégorie : 19
4.1.2 Régression 20
4.1.2.1 Régression linéaire 20
4.1.2.2 Régression logistique 21
4.2 Apprentissage non supervisé 21
4.3 Apprentissage semi-supervisé 22
4.4 Apprentissage par renforcement 22
5 Les algorithmes d'apprentissages automatiques 23
5.1 Les machines à vecteurs supports « SVM » 23
5.1.1 Définition 23
5.1.2 Principe de fonctionnement général 24
5.1.2.1 Hyperplan 24
5.1.2.2 Vecteur support 24
5.1.2.3 Marge 25
5.1.2.4 Noyaux 25
5.2 Naïve bayse 26
5.2.1 Théorème de bayes 26
5.2.2 Classificateur bayésien naïf 27
5.3 Forets aléatoires 28
5.4 K-Nearest Neighbor « KNN » 29
5.5 Arbre de décisions 30
6 Conclusion 31
Partie 02 : Conception et Implémentation 32

Chapitre 03: Conception 33

1 Introduction..... 33

2 Architecture..... 33

2.1 DataSet..... 35

2.2 Pre-Processing..... 35

2.2.1 Nettoyage des données..... 35

2.2.2 Feature extraction..... 36

2.2.2.1 Sélection des attributs manuellement : 36

4.2.2.2 Les méthodes warpper et filtre :..... 38

2.2.3 L'étiquetage de donnée..... 38

2.2.4 L'encodage des donnes « Encoder Label »..... 38

2.2.5 Normalisation..... 38

2.3 Validation du modèle..... 39

2.4 Les algorithmes de classification..... 39

2.7 Métriques d'évaluation : 40

3 Conclusion 41

Chapitre 04: Implémentation..... 42

1 Introduction..... 42

2 Outils et technologies utilisés 42

2.1 Outils:..... 42

2.2 Technologies :..... 42

3 DataSet..... 43

3.1 DataSet de moore:..... 43

3.2 Fichier « pcap » :..... 45

4 Résultats et discussion: 46

4.1 Première expérience :..... 46

4.2 Deuxième expérience :..... 53

4.2.1 la méthode warpper : 53
4.2.2 La méthode Filter : 55
4.2.3 La sélection des attributs manuellement : 56
4.2.3 Comparaison entre les méthodes : 57
4.3 Comparaison avec les approches proposées dans la littérature 58
5 Conclusion 58
Conclusion et perspectives : 59
Perspectives : 60
Références bibliographiques 61

Liste des figures

Figure 1.1 : Transmission des données 6

Figure 1.2 : Couches de Modèle OSI 7

Figure 1.3 : Couches de modèle TCP /IP 8

Figure 2.1 : Types d'apprentissage automatique 18

Figure 2.2 : Exemple de régression linéaire 20

Figure 2.3 : La fonction logistique (sigmoïde) 21

Figure 2.4 : Hyperplan H que sépare les deux ensembles des points 24

Figure 2.5 : Hyperplan H optimale, vecteurs supports et marge maximale 25

Figure 2.6 : Principe de l'algorithme des forêts aléatoires 29

Figure 2.7 : Algorithme k-Nearest Neighbor 30

Figure 3.1 : Architecteur de système utilisée pour la classification de trafic réseau .. 34

Figure 4.1 : Distribution de classe dans l'ensemble de donnée 46

Figure 4.2 : Comparaison entre l'accuracy de tous les algorithmes de classification . 47

Figure 4.3 : Les bon K pour la classification « Erreur rate » 48

Figure 4.4 : Comparaison Précision entre les algorithmes de classification. 51

Figure 4.5 : Comparaison Recall entre les algorithmes de classification. 52

Figure 4.6 : Comparaison F1-score entre les algorithmes de classification. 52

Figure 4.7 : Performances de mesure pour tous les algorithmes de classification, en utilisant la méthode du forward warrper 53

Figure 4.8 : Performances de mesure pour tous les algorithmes de classification, en utilisant la méthode du Backward warrper. 54

Figure 4.9 : Performances de mesure pour tous les algorithmes de classification, en utilisant la méthode du filtre. 55

Figure 4.10 : Performances de mesure pour tous les algorithmes de classification, en utilisant la méthode de sélection des attributs manuellement. 56

Figure 4.11 : Comparaison entre tous les résultats des articles. 58

Liste des tables

Table 1.1 : Quelques protocoles utilisés par les couches..... 7

Table 1.2 : Le numéro de port attribué par (IANA) à certaines applications..... 10

Table 1.3 : Résume et détaille des caractéristiques et les limites des différentes méthodes de classification..... 12

Table 3.1 : Les attribut utilisée dans notre étude (Feature sélection). 37

Table 3.2 : Type et paramètre d'algorithme d'apprentissage. 39

Table 4.1 : Les catégories du premier DataSet et leur application. 44

Table 4.2 : Nombre d'occurrence de classes. 44

Table 4.3 : Nombre d'occurrence de classes. 45

Table 4.4 : Nombre de classes du notre ensemble de donnée..... 45

Table 4.5 : Résultats d'accuracy entre tous les algorithmes de classification.. 47

Table 4.6 : Les performance de mesure d'algorithme SVM.. 49

Table 4.7 : Les performance de mesure d'algorithme DT. 49

Table 4.8 : Les performance de mesure d'algorithme RF. 50

Table 4.9 : Les performance de mesure d'algorithme NB. 50

Table 4.10 : Les performance de mesure d'algorithme KNN. 51

Table 4.11 : Performances de mesure pour tous les algorithmes de classification, en utilisant la méthode du Forward warpper 53

Table 4.12 : Performances de mesure pour tous les algorithmes de classification, en utilisant la méthode du Backward warpper. 54

Table 4.13 : Performances de mesure pour tous les algorithmes de classification, en utilisant la méthode du filtre. 55

Table 4.14 : Performances de mesure pour tous les algorithmes de classification, en utilisant la méthode de sélection des attributs manuellements 56

Liste des abréviations

QoS : Quality of service.

OSI : Open Système Interconnection.

DPI : L'inspection des paquets de données.

ISO : L'international Standards Organisation.

TLS : Transport Layer Sécurité.

CSV : Comma-Separated Value.

TCP/IP : Transmission Control Protocol /Internet Protocol.

IP : Internet protocol.

P2P : Peer to Peer.

SVM : Support Vector Machine.

KNN : k-nearest neighbor.

NB : Naïve bayse.

DT : Decision tree.

RF : Random forest.

IA : Intelligence Artificielle.

ML : Machine Learning.

TP : True Positive.

TN : True Negative.

FP : False Positive.

FN : False Negative.

IANA : Internet assigned numbers authority.

Introduction générale

Depuis leur apparition dans les années 60, les réseaux d'interconnexions ont connu une évolution rapide autant au niveau de la taille qu'au niveau de la complexité.

Plusieurs types de réseaux ont vu le jour, entre autres, les réseaux d'entreprise et de centres de données. L'émergence de ces réseaux, que ce soit à grandes échelles comme les centres de données ou les réseaux moyens comme les réseaux d'entreprise engendre de nouveaux défis concernant la sécurité, l'optimisation d'utilisation des ressources, les performances réseau, la qualité de service, etc. Par conséquent, la surveillance et le contrôle des réseaux deviennent indispensables, surtout avec l'abondance et la diversité des données y transitant. Cependant, une méconnaissance du trafic occasionne sans doute une mauvaise gestion des réseaux. Alors La classification du trafic réseau est une tâche importante dans les réseaux de communication modernes.

Au cours des deux dernières décennies, de nombreuses techniques de classification du trafic réseau ont été proposées pour classer les applications inconnues. La première est la technique basée sur le port. C'est une excellente technique pour la classification/identification du trafic réseau. Mais cette technique a échoué en raison de l'augmentation des applications Peer to Peer « P2P », qui utilisent des numéros de port dynamiques. Ensuite, la deuxième est la technique basée sur la charge utile.

Cette technique donne des résultats précis dans la classification du trafic réseau. Cette technique est également appelée technique d'inspection approfondie des paquets « DPI ». Mais il y a un problème dans cette technique. Le problème est qu'il ne peut pas être utilisé pour les applications de réseau de données cryptées car de nombreuses applications de réseau utilisent des techniques cryptées pour protéger les données contre la détection. Ainsi, cette technique a également échoué en raison de l'utilisation d'un flux crypté d'applications. Par la suite, les chercheurs ont proposé une autre méthode appelée apprentissage automatique pour classer le trafic internet ainsi que pour savoir quel type d'applications circulent dans le réseau. La technique d'apprentissage automatique donne des résultats précis très prometteurs dans la

classification du trafic réseau. Cette technique est basée sur la formation et le test d'ensembles de données pour classer des applications inconnues.

Cette mémoire se compose de quatre chapitre :

Chapitre 1 : Il décrit les concepts de trafic réseau et les déférents méthodes de classification du trafic réseau.

Chapitre 2 : Nous passons en revue un aperçu sur l'apprentissage automatique et les déférents type d'apprentissage et les algorithmes utilise dans la classification.

Chapitre 3 : nous présenterons le concepts et l'architecture de notre système et les étapes utilise dans notre étude.

Chapitre 4 : nous présenterons les résultats obtenus à l'aide des tableaux des performances et des graphiques base sur des statistiques, nous allons faire les discussions et nous basons sur les tests statistique bien connus. Enfin, nous concluons en résumant les contributions de ce mémoire, les limites et les travaux future.

Partie 01 : *Etat de l'art*

Dans cette partie, on a expliqué les notions de base qui permettant de compris les idées développées dans ce mémoire. Elle se compose en deux chapitre. Dans le premier chapitre, on a passée en revue de la classification du trafic réseau et la notion du trafic réseau ainsi que le flux de donnée et aussi les principes de classification du trafic réseau.

Dans le deuxième chapitre nous allons présentons certain concept de l'apprentissage automatique et faire une description des composants des machine learning avec la classification du trafic réseau.

Chapitre 01 : Classification du trafic réseau

1 Introduction

Le trafic réseau et ses données sont une source d'informations très précieuse et très intéressante. Les entreprises, les universités ou les pays utilisent différentes approches et solutions de sécurité pour protéger les données et les utilisateurs sur le réseau.

La classification du trafic réseau peut être décrite comme le processus d'attribution d'instances ou d'éléments de trafic à les applications ou les types d'applications qui les ont générées

La classification du trafic réseau peut être décrite comme le processus d'attribution d'instances ou des éléments de trafic à des applications ou aux types d'applications qui les ont créées. La classification précise et opportune des données de trafic réseau est d'une importance capitale pour les fournisseurs de services et les opérateurs de réseau, offrant plusieurs avantages, notamment : la surveillance de la sécurité du réseau, l'évitement de la congestion, la gestion du protocole Internet (IP), l'application de la qualité de service (QoS), la gestion de la bande passante [1].

Ce chapitre est organisé comme suit: La section 2 présente la définition de trafic réseau. En suite dans La section 3 on a décrire la définition de flux de donnée et comment les données sont-elles transférées. Et à la fin dans la section 4, nous allons présenté la définition de classification du trafic réseau et aussi les méthodes de la classification du trafic.

2 Trafic réseau

Le trafic réseau est la quantité de données circulant sur un réseau informatique à un moment donné. Le trafic réseau, également appelé trafic de données, est décomposé en paquets de données et envoyé sur un réseau avant d'être réassemblé par l'appareil ou l'ordinateur récepteur. [2]

Le trafic réseau a deux flux directionnels, nord-sud et est-ouest. Le trafic affecte la qualité du réseau car une quantité de trafic inhabituellement élevée peut entraîner des vitesses de téléchargement lentes ou des connexions (Voix sur IP) inégales.

Le trafic est également lié à la sécurité, car une quantité de trafic inhabituellement pourrait être un signe d'une attaque. [2]

3 Flux de réseau

Le terme « flux » peut être considéré comme une conversation entre deux points d'extrémité sur un réseau. Ces deux points d'extrémité échangeront des paquets entre eux jusqu'à la fin de cette conversation. Les paquets se composent de deux sections, la section d'en-tête qui contient des informations sur le paquet (par exemple, l'adresse de destination et d'origine) et la section de charge utile qui contient le message qui sera remis au destinataire. Un flux individuel a été défini comme l'échange unidirectionnel ou bidirectionnel de paquets contenant les cinq mêmes propriétés clés : (Destination IP adresse, Source IP adresse, Destination port adresse, Source port adresse) [3].

3.1 Transmission de l'information

Avant de spécifier les modèles en vigueur sur les réseaux, on va découvrir de façon sommaire comment l'information est transmise, en tous les cas quels sont les principes qui ont été retenus.

Les modèles d'architecture réseau sont composés de couches, relativement indépendantes les unes des autres. On part des applications (haut niveau) à la transmission physique (bas niveau).

L'idée retenue a été de segmenter l'information à transmettre (on parlera de segments, paquets, trames), chaque couche ajoutant des informations

On voit que chaque donnée est englobée (encapsulée), à chaque étape. Le processus inverse se produit à la réception du message par la machine réceptrice**.** Comme illustré à la **Figure 1.1**.

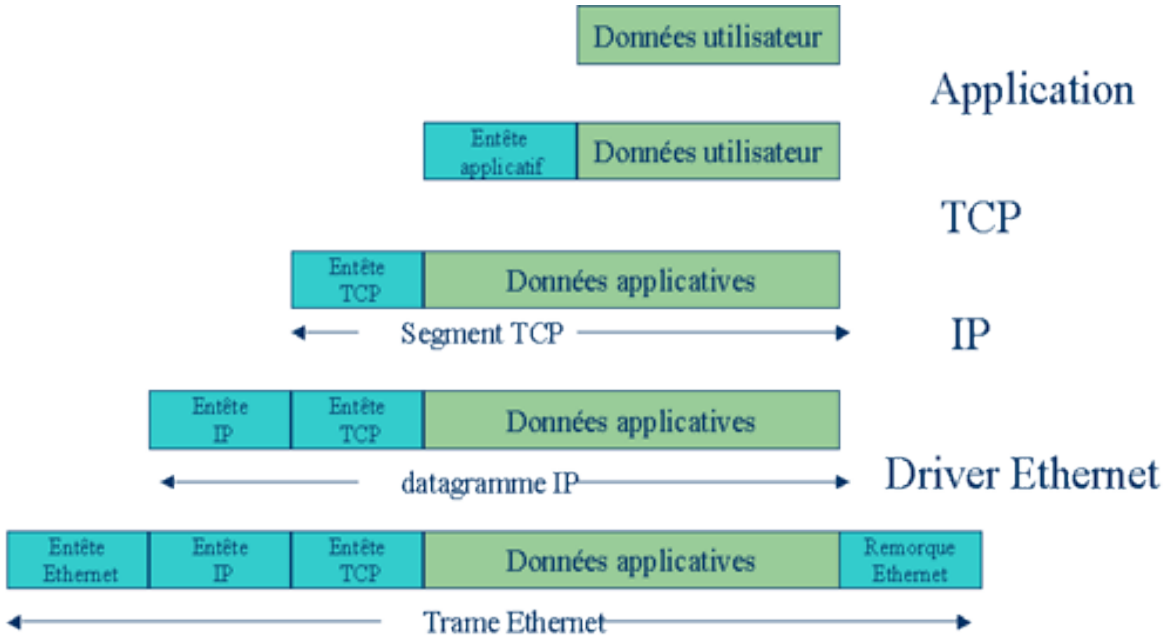


Figure 1.1 : Transmission des données. [4]

L'idée de découper l'information en petits segments est simple : la transmission est plus rapide, s'il y a une erreur ou une perte, il n'y a pas tout le message à renvoyer, il y a moins d'engorgement des capacités de transmission [4].

3.2 Modèle OSI

Pour uniformiser les protocoles utilisés lors des communications entre les différentes couches d'un réseau, l'International Standards Organisation « ISO » a fondé un modèle, le modèle OSI. La dernière révision de ce modèle date de 1995. [5]

Le modèle OSI n'est pas une architecture réseau, car il ne décrit pas explicitement les protocoles mis en jeu. Elle précise seulement le rôle de chaque couche. Ce modèle est composé de 7 couches comme illustré à la **Figure 1.2**.

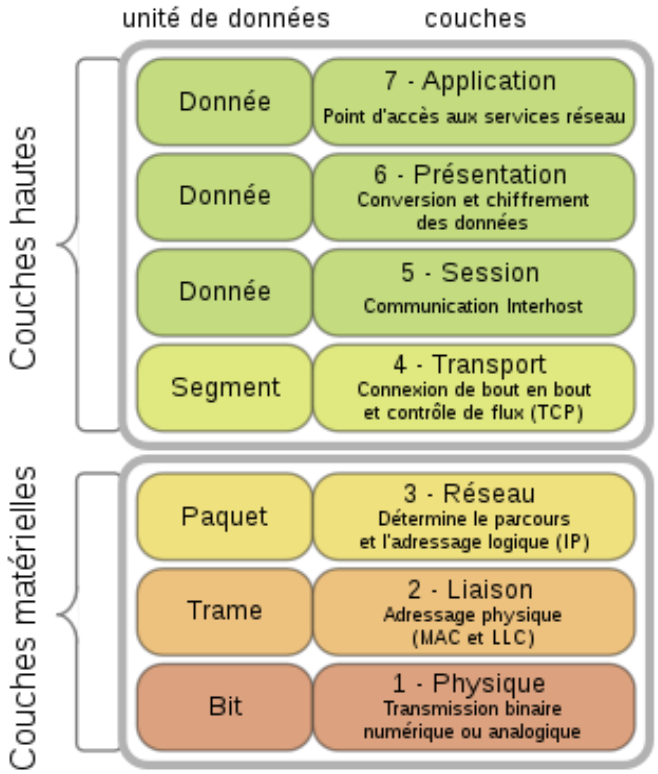


Figure 1.2 : Couches de Modèle OSI.

Table 1.1 : Quelques protocoles utilisés par les couches.

Application	FTP,SSH,SFTP,DNS,HTTP,IMAP NFS, POP3, Telnet, FIX….
Presentation	ASCII, Unicode
Session	RPC, NetBIOS
Transport	TCP,UDP,RTP
Réseau	IP, ICMP,IGMP,ARP
Liaison	Ethernet, ATM, Token ring, SLIP
Physique	Électronique , radio

Le modèle OSI a eu l'avantage d'imposer des standardisations et de prendre en compte l'hétérogénéité du monde informatique de l'époque. Cependant, il est postérieur (fin des années 70) au modèle TCP/IP (développé par Arpanet), qui ne présente que 4 (ou 5) couches, et dont les débuts (université de Berkeley en 74 sous Unix) ont été couronnés de succès. L'avènement d'internet lui assurera de devenir le modèle retenu. [6]

3.3 Modèle TCP/IP

Le modèle TCP/IP n'est pas vraiment éloigné du modèle OSI. Il ne présente cependant que 4 couches, même si un modèle hybride, séparant les couches Physique et Liaison peut faire autorité. [7]

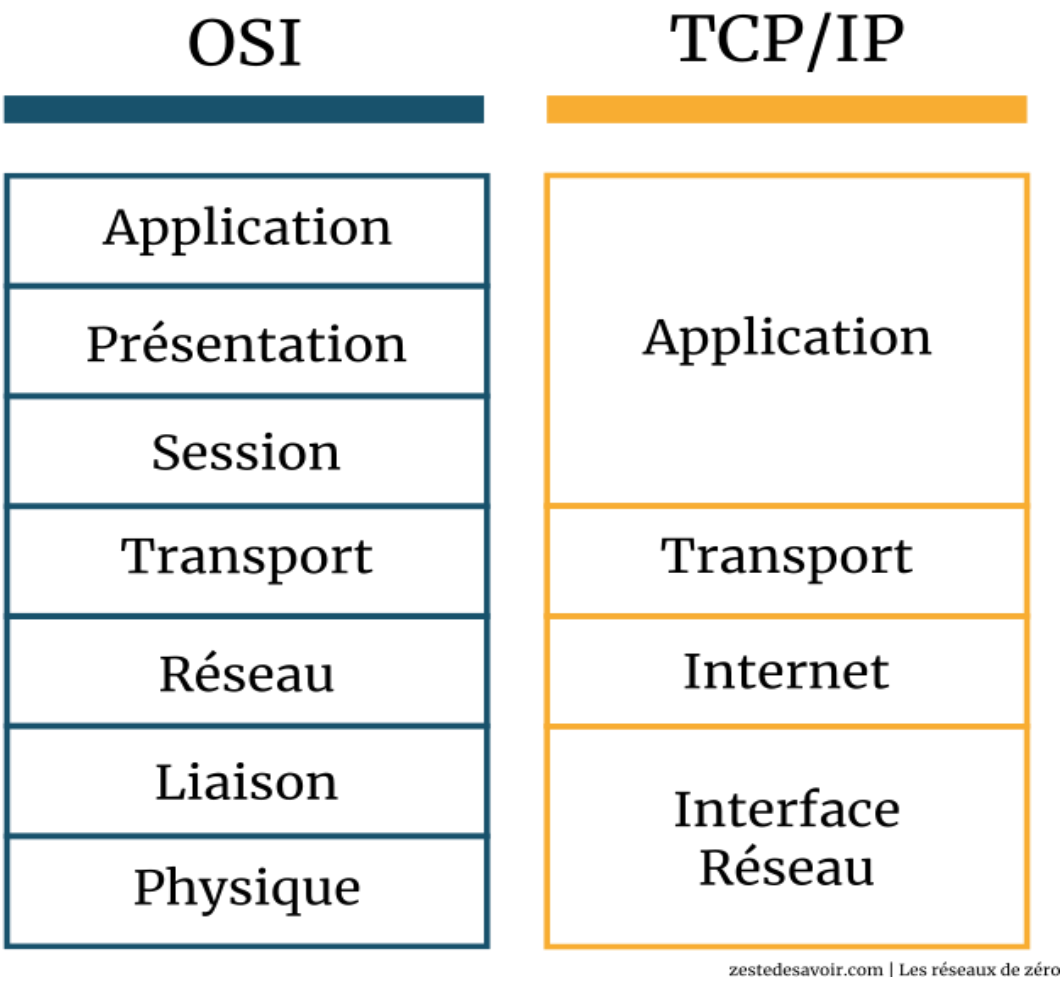


Figure 1.3 : Couches de modèle TCP /IP.

3.3.1 Couche hôte réseau

Elle est en fait composée de deux couches : Physique et Liaison.

- **Couche physique :**

Décrit les caractéristiques physiques de la connexion câbles, ondes…

- **Couche liaison :**

Spécifie le moyen utilisé pour acheminer les données sur la couche physique : Câble, Ethernet,Wi-FI.

3.3.2 Couche internet

Son rôle est l'injection de paquets dans n'importe quel réseau. Lorsque deux terminaux communiquent entre eux via ce protocole, aucun chemin pour le transfert des données n'est établi à l'avance

Il est dit que le protocole est « non orienté connexion ». Ainsi les paquets envoyés peuvent arriver dans le désordre car ils n'auront pas suivi la même route. C'est le protocole transport qui se chargera de remettre les paquets dans le bon ordre.

3.3.3 Couche transport

Son rôle est similaire à celui de la couche transport du modèle OSI. Les protocoles utilisés à ce niveau sont TCP et UDP.

TCP est fiable, acheminant sans erreur les paquets à destination, utilisant des services d'acquittement, de gestion du temps d'attente. UDP est non fiable mais plus rapide. Il est utilisé dans les liaisons voix IP, où l'on préfère perdre quelques données qu'attendre. Utilisé aussi pour le streaming ou la vidéoconférence.

Dans l'entête TCP, on retrouve le code du port source et du port destination. Par exemple 80 pour HTTP.

3.3.4 Couche application

Le Modèle TCP/IP est fondé sur le constat que les logiciels réseaux n'utilisent que très peu, ou pas, les couches session et présentation. Cette couche regroupe toute les protocoles de haut niveau (FTP, SMTP, HTTP, DNS…). Cette couche devra choisir un protocole de transport adapté au service demandé.

4 Classification de trafic réseau

La classification des flux réseau en fonction du type d'application est la technique de base pour de nombreux systèmes de surveillance et de contrôle réseau, notamment, les systèmes de gestion de la qualité de service et les systèmes de sécurité (détection d'anomalies et d'intrusions) [10].

Plusieurs techniques ont été appliquées pour classer les données de trafic réseau au cours des deux dernières décennies. Cette section examine les méthodes de classification et les divise en 3 catégories en fonction de leur évolution chronologique.

4.1 Classification basée sur les ports

Au début d'Internet, la classification et l'identification du trafic réseau n'étaient pas du tout un problème. La classification basée sur les ports impliquait l'identification d'une application basée sur l'inspection de l'en-tête du paquet et sa mise en correspondance avec le numéro de port TCP ou UDP enregistré auprès de l'IANA (Internet Assigned Numbers Authority) [11].

Tableau 1.2 montre les différents types d'applications et leurs numéros de ports attribués par l'IANA, par exemple les applications utilisent le numéro de port 25 (SMTP) pour envoyer et recevoir des e-mails, les applications web utilisent numéro 80 comme numéro de port…etc.

Tableau 1.2 : le numéro de port attribué par (IANA) à certaines applications.

Application	Port attribué
FTP Data	20
FTP	21
SSH	22
Telent	23
SMTP	25
http	80
POP3	110

Les développements historiques ont révélé l'imprécision et la fiabilité de cette technique traditionnelle. Le déclin de l'utilisation de cette technique est dû à plusieurs raisons, notamment : certaines applications modernes circulent avec des numéros de port non enregistrés (ports non standard) ou choisissent un port aléatoire, par exemple les applications Peer to Peer (P2P), Pire encore, d'autres applications (par exemple, des applications non légitimes) se cachent derrière des ports bien connus pour éviter d'être filtrées et par conséquent de contourner les restrictions de contrôle d'accès au système d'exploitation [11].

4.2 Classification par l'inspection de charge des paquets « DPI »

La technique basée sur la charge utile repose sur l'inspection du contenu des paquets et leur mise en correspondance avec un ensemble déterministe de signatures stockées (modèle). Les résultats de cette méthode de classification sont extrêmement précis.

L'inspection de la charge utile a été largement utilisé dans plusieurs outils commerciaux et open source. En outre, les classificateurs basés sur la charge utile sont souvent utilisés comme étape principale dans les systèmes de détection d'intrusion réseau (IDS) pour identifier les activités malveillantes sur le réseau. La fiabilité de cette méthode a été largement étudiée. Sen et al. [12]

Bien que l'inspection basé sur la charge utile soit considéré comme une technique fiable, il présente des inconvénients et des faiblesses importantes. Premièrement, la capacité du classificateur diminue lorsqu'il s'agit de trafic chiffré et d'obscurcissement ou d'encapsulation de protocole ; l'Inspection de paquet crypté de cette manière est impossible, ce qui signifie qu'une grande partie du trafic réseau reste non classifiée. En plus de cela et plus important encore, l'inspection du contenu d'un paquet pose des problèmes de confidentialité et cet acte peut représenter une violation des politiques et réglementations de confidentialité. De plus, ce procédé impose un coût et une charge de calcul élevés au dispositif de classification, car il nécessite plusieurs instances d'accès au contenu du paquet. Par conséquent, la classification basée sur la charge utile rencontre des difficultés pour faire face au grand nombre de flux et au débit élevé du trafic réseau. [13]

4.3 Classification basée sur l'analyse statistique des attributs

La classification statistique est une technique basée sur la justification qui exploite les caractéristiques statistiques du flux de trafic réseau pour identifier l'application. Cette méthode utilise un certain nombre de mesures de niveau de flux [14], par exemple, la durée du paquet, le temps d'inter-arrivée des paquets, la longueur des paquets et le temps d'inactivité du flux de trafic. Ces mesures sont uniques pour un type spécifique d'applications ; par conséquent, cela permet au classificateur de distinguer différentes applications les unes des autres.

Au début, les caractéristiques statistiques du trafic réseau ont été étudiées dans plusieurs études. Dans [13], les auteurs ont observé la corrélation entre la classe dans le trafic réseau et les caractéristiques statistiques du flux telles que les octets et la durée. Ils proposent des modèles expérientiels de caractéristiques de connexion pour un grand nombre d'applications TCP. De plus, dans [14], les auteurs ont utilisé les caractéristiques statistiques du flux du réseau, telles que la taille des paquets, l'heure d'inter-arrivée des paquets et la durée du flux, pour analyser les systèmes de chat sur Internet. Certains des travaux ultérieurs tels que [13], et ont observé les propriétés uniques du trafic réseau pour un certain nombre d'applications Internet. Les résultats

de ces études ont inspiré les chercheurs à travailler sur de nouvelles techniques de classification basées sur des propriétés statistiques. [15]

Pour effectuer la classification réelle basée sur des caractéristiques statistiques, les classificateurs doivent utiliser des techniques d'exploration de données, en particulier des algorithmes d'apprentissage automatique, car ils doivent traiter différents modèles de trafic à partir de grands ensembles de données.

Les algorithmes d'apprentissage automatique sont extrêmement légers et mathématiquement moins coûteux que les techniques de classification basées sur la charge, car ils utilisent des informations provenant de l'analyse des flux.

L'efficacité des travaux en classification statistique dépend des caractéristiques extraites du flux, qui nécessitent une connaissance approfondie en raison de sa complexité.

Cependant, ces techniques surpassent les techniques basées sur la charge utile car elles ne traitent pas le contenu des paquets et peuvent donc analyser le trafic chiffré sans aucune difficulté.

Donc, ces techniques surpassent les techniques basées sur la charge utile car elles ne traitent pas le contenu des paquets et peuvent donc analyser le trafic chiffré sans aucune difficulté.

Tableau 1.3: Résumé des caractéristiques et les limites des différentes méthodes de classification. [15]

S/N	Méthode de classification	Caractéristiques/Avantages	Désavantages
1	classification basée sur les portes	• Identification du trafic effectuée à l'aide des numéros de port attribués par l'IANA. • La méthode est rapide et consomme peu de ressources.	• En raison du nombre croissant d'application ,il y a tendance á utliser des numéros de port imprévisibles • La méthode peut ne pas convenir aux applications non enregistrées par l'IANA. • La méthode peut ne pas

		• Pris en charge par de nombreux périphériques réseau.	fonctionner correctement avec les applications qui utilisent des numéros de port alloués dynamiquement.
2	classification par l'inspection de charge des paquets (DPI)	• Inspecte la charge utile réelle du paquet. • Identification non basée sur le numéro de port. • La méthode fournit un résultat plus précis par rapport aux techniques basées sur le port. • La méthode est tout à fait adaptée au trafic P2P.	• La méthode est lente et nécessite beaucoup de puissance de traitement - coût de calcul élevé. • Les signatures doivent être tenues à jour, car les applications changent très fréquemment • Pas facile à appliquer au trafic crypté. • La méthode souffre également d'une violation des politiques réglementations de confidentialité
3	classification basée sur l'analyse statistique	• La méthode est basée sur l'analyse Des propriétés statistiques du flux comprenant les paquets. • Des attributs de flux tels que la taille des paquets, les temps entre	• Il est généralement nécessaire de prétraiter les données de trafic avant la classification • Les performances des Algorithmes de (ML) peuvent être affectées par un trop grand nombre de fonctionnalités, en

	des attributs	les arrivées des paquets, etc. peuvent être utilisés. • La méthode est rapide et consomme moins de puissance de traitement • Il peut également détecter la Classe d'application encore Inconnues • La méthode est également capable d'identifier le trafic crypté.	particulier lorsqu'elles ne sont pas pertinentes ou sont redondantes. La sélection de fonctionnalités est généralement utilisée pour sélectionner des ensembles de fonctionnalités optimaux.

5 Conclusion

Dans ce chapitre, nous avons présenté la classification du trafic réseau. Tous d'abord nous avons expliqué la notion du trafic réseau. Ensuit on a défini le flux de donnée, et enfin nous avons présenté les techniques utilises pour classification de trafic réseau.

Le trafic réseau est un domaine d'application très riche d'apprentissage automatique Pour cette raison nous allons introduire l'apprentissage automatique afin d'appliquer la classification de trafic réseau.

Chapitre 02 : Apprentissage automatique

1 Introduction

L'apprentissage est défini comme "le processus de changement et d'amélioration des comportements par l'exploration de nouvelles informations dans le temps". Lorsque « l'apprentissage » dans cette définition est effectué par les machines, on parle d'apprentissage automatique. [16]

L'apprentissage automatique est un sous-domaine de l'intelligence artificielle « IA », son objectif est généralement de comprendre la structure des données et d'intégrer ces données dans des modèles qui peuvent être compris et utilises par les gens.

L'apprentissage automatique aujourd'hui est largement utiliser pour faciliter notre vie, ces algorithmes permettre de résoudre des solutions dans des défèrent problème comme la classification du trafic réseau.

Ce chapitre est organisé comme suit : les sections 2 et 3 présente la définition et l'historique d'apprentissage automatique, la section 4 présente les déférents types d'apprentissage automatique, et la section 5 présente les déférents algorithmes utilise dans notre travaille, finalement on à faire une petite conclusion qui résume tous le chapitre.

2 Définition

L'apprentissage automatique ou Machine Learning, est l'utilisation d'algorithme permettant à des programmes informatiques de s'améliorer automatiquement par le biais de l'expérience. En d'autres termes, ces algorithmes permettent à un ordinateur d'apprendre à la manière d'un être humain. L'apprentissage automatique offre de nombreuses possibilités, et permet notamment de résoudre des problèmes très compliqués. On l'utilise notamment pour l'analyse d'imagerie médicale, pour les assistants numériques, la détection de fraude, la cyber sécurité ou encore les voitures autonomes. Il s'agit en réalité d'une véritable révolution industrielle. De manière générale, le Machine Learning exploite diverses techniques pour traiter de larges volumes de données et s'appuyer sur les résultats pour prendre des décisions ou effectuer des prédictions [17].

3 Histoire d'apprentissage automatique

Dans les années 1940, sur la base des études sur les crashs électriques de neurone, les scientifiques sont expliqué le mécanisme de prise de décision de l'homme par le canon et le feu, les années 1950, les recherches sur l'intelligence artificielle ont commencé [18].

Au cours de ces années, Alan Turin a exécuté le test de Turing afin de teste la capacité d'une machine à imite un humain, l'objectif du test de Turing était de mesure la capacité de la machine à entre en contact avec un humain lors d'un entretien. Si la machine fonctionnait moins bien qu'un humain, c'était un succès.

En 1956, le terme « Intelligence Artificielle » a été utilisé pour la première fois dans un école d'été organisée par Marvin Minsky du (Massachusetts Institute of Technology), John McCarthy de l'université de Stanford et Allen Newell et Herbert Simon de l'université Carnegie-Mellon. Jusque-là, le terme d'Alan Turing, « l'intelligence de la machine », avait été utilisé.

En 1959, Arthur Samul a créé le programme des dames, puis l'apprentissage automatique a fait son chemin. De ces développements aux années 1980, il y a eu quelques études sur l'esprit abstrait, les systèmes basés sur l'information, que l'on a appelés « l'hiver de l'intelligence artificielle ». Dans les années 1990, les études sur l'intelligence artificielle et l'apprentissage automatique se sont accélérées en raison de l'évolution des technologies des jeux. De nos jours, l'intelligence artificielle et

l'apprentissage automatique sont utilisés dans nombreux secteurs de recherche et de travail [19].

4 Les types d'apprentissages automatiques

Il existe plusieurs méthodes de faire apprendre à une machine **Figure 2.1**. Pour bien choisir une méthode il faut comprendre les fondements de ces méthodes et qui permet de distingue afin de détermine les modèles qui traitent au mieux un problème particulier.

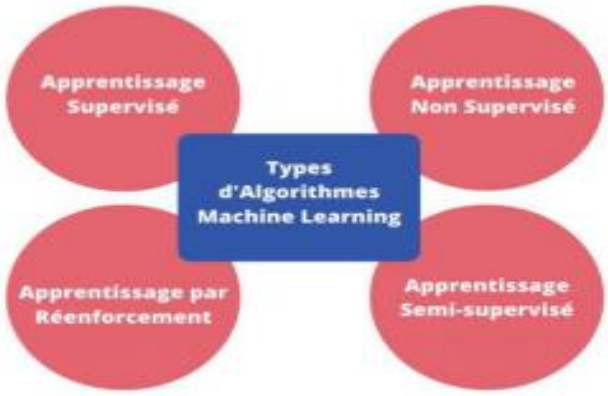


Figure 2.1 : les types d'apprentissage automatique.

L'apprentissage automatique peut être examiné en quatre parties comme suit :

- Apprentissage supervisé « Supervised Learnning ».
- Apprentissage non supervisé « Unsupervised Learnning ».
- Apprentissage semi-supervisé « Semi-Supervised Learnning ».
- Apprentissage par renforcement « Reinfirocemenc Learnning ».

4.1 Apprentissage supervisé

L'algorithme d'apprentissage supervisé s'appuie sur des classes ou des valeurs prédéterminées pour mettre en place un système de classement à partir d'observations déjà cataloguées. Il s'agit alors d'utiliser ces observations déjà classées pour apprendre un modèle qui permette d'associer à toute nouvelle observation rencontrée la classe la plus adaptée. C'est la méthode utilisée pour prédire le changement de note, c'est-à-dire qu'à partir d'un certain nombre d'obligations cataloguées comme ayant

changé de note ou non, l'algorithme mis en œuvre est capable de considérer les caractéristiques d'une nouvelle obligation et de lui attribuer sa classe de changement de note. [20].

Il existe deux types d'apprentissage supervisé : l'apprentissage par classification et l'apprentissage par régression:

- **Classification** : Répartition des données dans les catégories définies sur le jeu de données en fonctions de leurs spécificités.
- **Régression** : Prédire ou conclure les autres caractéristiques des données en fonction de certaines de ces caractéristiques disponibles. [16]

4.1.1 Classification

Les méthodes de classification s'appliquent lorsque l'ensemble des valeurs résultats est discret. Ceci revient à attribuer une classe « aussi appelée étiquette ou label » pour chaque valeur d'entrée. Les techniques de classifications peuvent être basées sur des hypothèses probabilistes « par exemple : naïf bayésien », des notions de proximité « par exemple : *k* plus proches voisins » ou des recherches dans des espaces d'hypothèses « par exemple : arbres de décisions ». Le choix de la technique convenable est important, il faut pouvoir choisir la méthode la plus adapté qui sera capable de séparer au mieux les données d'apprentissage [21].

Il existe deux types de classifications :

4.1.1.1 Classification binaire :

Il s'agit d'un processus ou d'une tâche de classification, dans laquelle une donnée est classée en deux classes. Il s'agit essentiellement d'une sorte de prédiction sur le groupe auquel appartient la chose [19]

Exemples de classification binaire :

- Détection de spam « Spam ou Non ».
- Prédiction de cohorte « Travail ou Non ».
- Espérance de conversion « Achat ou Non » …

4.1.1.2 Classification multi-catégorie :

La classification multi-catégorie consiste à classe les éléments dans différentes catégories. Contrairement au système binaire, il n'est pas limité à un nombre quelconque de classes.

Exemples de classification multi-catégories :

- Classification de trafic réseau.

- Classification des livres par thème.
- Classification des étudiants selon leur spécialité.
- Catégoriser les actualités dans différentes catégories etc.

Il existe plusieurs algorithmes on peut utilise dans la classification d'après elle :

- Support Vector Machine « SVM ».
- Naïve Bayes « NB ».
- Forest Aléatoire « Random forest ».
- Arbre de décision « DT ».
- K-Nearest Neighbors « KNN ».

4.1.2 Régression

Dans la machine, le problème de la régression elle produit tant que la valeur de sortie est une valeur réelle ou continue, par exemple : poids, salaire.

4.1.2.1 Régression linéaire

Technique consistant à approcher la dépendance entre deux variables X et Y par une droite Y=aX+b. [22]

Dans le cas le plus simple, cette technique tente d'approcher par une droite la relation entre deux variables. On peut l'appliquer à plusieurs variables pour essayer de déterminer une droite qui passe le plus proche possible de la courbe définie par les dites variables. L'utilisation d'une telle approche n'est pas évidente dans la mesure où rare sont les phénomènes qui sont linéaires. L'exemple suivant montre la dépendance entre le taux de la troponine dans le sang humain et le risque d'avoir un infarctus du myocarde. Les points de la « **Figure 2.2** » ne vérifient presque jamais la droite. Dans le cas de phénomènes linéaires, la régression linéaire reste simple et efficace. [22]

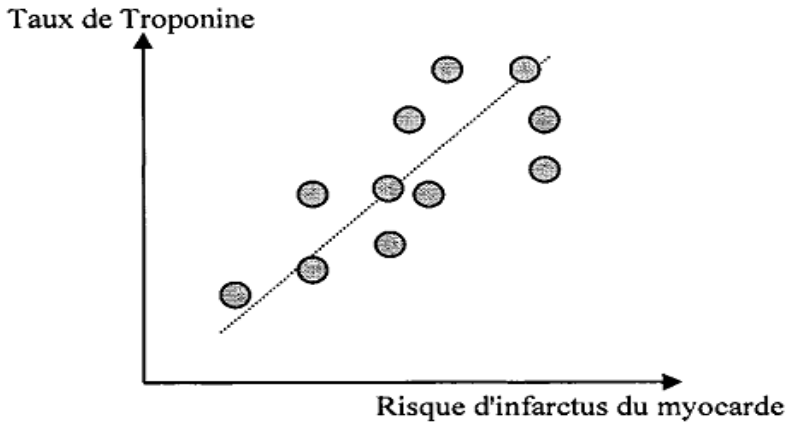


Figure 2.2 : Exemple de régression linéaire. [22]

4.1.2.2 Régression logistique

La régression logistique est une technique d'analyse d'un groupe de donnée qui comporte une ou plusieurs variable prédicteur « indépendant » et une variable cible « dépendant », les résultats sont plusieurs et elles sont des valeurs binaires. La régression logistique est rapide et relativement simple, Il appartient au groupe des classificateurs linéaires et ressemble quelque peu à la régression polynomiale et linéaire. Elle permet aussi de prédire la probabilité qu'un événement arrive ou non « valeur de 1 ou 0 » à partir de l'optimisation des coefficients de régression. Ce résultat varie toujours entre 0 et 1. Lorsque la valeur prédite est supérieure à un seuil, l'événement est susceptible de se produire, alors que lorsque cette valeur est inférieure au même seuil, il ne l'est pas.

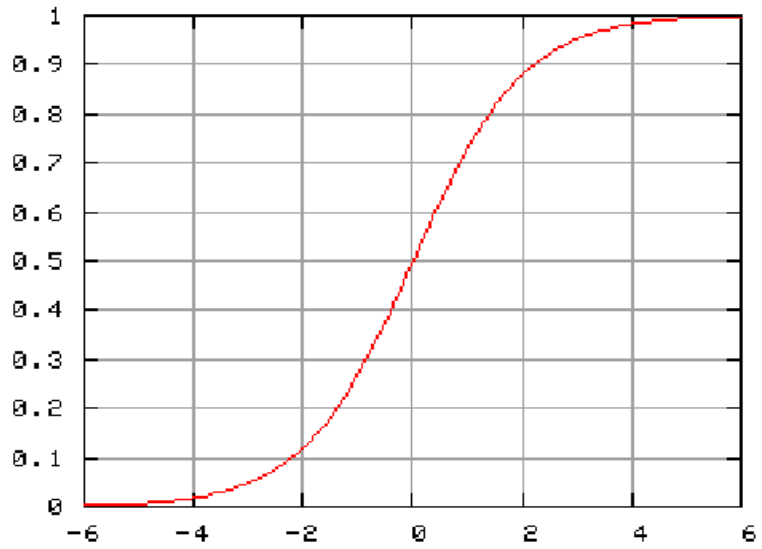


Figure 2.3 : La fonction logistique « sigmoïde ». [23]

4.2 Apprentissage non supervisé

Dans ce mode de fonctionnement du machine learning, il n'est pas question de s'appuyer sur des éléments prédéfinis, et la tâche de procéder à la catégorisation des données revient à la machine. Pour ce faire, le système va croiser les informations qui lui sont soumises, de manière à pouvoir rassembler dans une même classe les éléments

présentant certaines similitudes. Ainsi, en fonction du but recherché, il reviendra à l'opérateur ou au chercheur de les analyser afin d'en déduire les différentes hypothèses [32].

Il existe également deux types d'apprentissage non supervisé : le regroupement et l'association.

- **Clustering :**

  Trouve les regroupements de données qui sont similaires les uns aux autres lorsque les regroupements inhérents aux données ne sont pas connus.

- **Association :**

  Déterminer les relations et les connexions entre les données d'un même ensemble de données. [16]

  La différence entre l'apprentissage supervisé et non supervisé que, dans l'apprentissage non supervisé, les données de sortie ne sont pas fournies.

4.3 Apprentissage semi-supervisé

L'apprentissage supervisé et non supervisé est insuffisant lorsque les données étiquetées sont inférieures aux données non étiquetées.

Dans tels cas, des données non étiquetées, qui sont absolument insuffisantes, sont utilisées pour déduire des informations à leur sujet.

Cette méthode est appelée apprentissage semi-supervisé. La différence entre l'apprentissage semi-supervisé et supervisé est l'ensemble de données étiquetées. Dans l'apprentissage supervisé, les données étiquetées sont plus que les données auxquelles on peut s'attendre. En revanche, dans l'apprentissage semi-supervisé, les données étiquetées sont inférieures aux données qui seraient attendues. [24]

4.4 Apprentissage par renforcement

L'apprentissage par renforcement « RL pour Reinforcement Learning » fait référence à une classe de problèmes d'apprentissage automatique, dont le but est d'apprendre à partir d'expériences successives, ce qu'il convient de faire de façon à trouve la meilleure solution.

Dans un tel problème, on dit qu'un « agent » (l'algorithme, au sens du code et des variables qu'il utilise) interagit avec « l'environnement » pour trouve la solution optimale. L'apprentissage par renforcement diffère fondamentalement des problèmes supervisés et non supervisés par ce côté interactif et itératif : l'agent essaie plusieurs solutions (on parle « d'exploration »), observe la réaction de l'environnement et adapte son comportement « les variables » pour trouve la meilleure stratégie (il « exploite » le résultat de ses explorations). Un des concepts clés de ce type de problèmes est l'équilibre entre ces phases d'exploration et d'exploitation. Cette méthode est particulièrement adaptée aux problèmes

nécessitant un compromis entre la quête de récompenses à court terme et celle de récompenses à long terme. Parmi les exemples de problèmes traités de cette façon, on peut évoquer : apprendre à un robot à marche en terrain difficile, à conduire (cas de la voiture autonome) ou à accomplir une tâche spécifique (comme jouer au jeu de go), pilote un agent à travers un labyrinthe, etc. Les principales familles de problèmes d''apprentissage par renforcement sont les algorithmes de bandits, les problèmes de décisions (partiellement) markovien et les arbres de jeu. [25]

5 Les algorithmes d'apprentissages automatiques

À partir de cette section, on introduit les techniques d'apprentissage automatique, notamment l'apprentissage supervisé. Il existe une variété de méthodes (algorithmes) de classification, voici une brève description de chaque méthode de classification.

5.1 Les machines à vecteurs supports « SVM »

5.1.1 Définition

Les machines à vecteurs supports ou « Support Vector Machine » sont un ensemble de techniques d'apprentissage destinées à résoudre des problèmes de discrimination, c'est-à-dire décider à quelle classe appartient un échantillon, ou de régression, c'est-à-dire prédire la valeur numérique d'une variable.

Le succès de cette méthode est justifié par les solides de bases théoriques qui la soutiennent.

L'idée principale des SVM consiste à projete les données dans un espace de plus grande dimension appelé espace de caractéristiques, les données non linéairement séparables dans l'espace d'entrée deviennent linéairement séparables dans l'espace de caractéristiques.

Ce produit scalaire peut être exprimé, sous certaines conditions, par des fonctions définies dans l'espace d'entrée, qu'on appelle les noyaux.

Ce multiple choix de noyaux rend les SVM plus intéressantes et surtout plus riches puisqu'on peut toujours cherche de nouveaux noyaux qui peuvent être mieux adaptés à la tâche qu'on veut accomplir. Les trois noyaux les plus utilisés sont : le noyau linéaire, le noyau polynomial et le noyau gaussien noté aussi RBF (Radial Basis Fonction). [26]

5.1.2 Principe de fonctionnement général

Pour deux classes d'exemples donnés, le but de SVM est de trouve un classificateur qui va séparer les données et maximise la distance entre ces deux classes. Avec SVM, ce classificateur est un classificateur linéaire appelé hyperplan.

5.1.2.1 Hyperplan

Plaçons-nous dans le cas d'une classification binaire « i.e. les exemples à classifie réparties en 2 classes ». On appelle hyperplan séparateur un hyperplan qui sépare les deux classes **Figure 2.4**, en particulier il sépare leurs points d'apprentissage. Comme il n'est en général pas possible d'en trouve un, on se contentera donc de cherche un hyperplan discriminant qui est une approximation au sens d'un critère à fixer « maximiser la distance entre ces deux classes » [27].

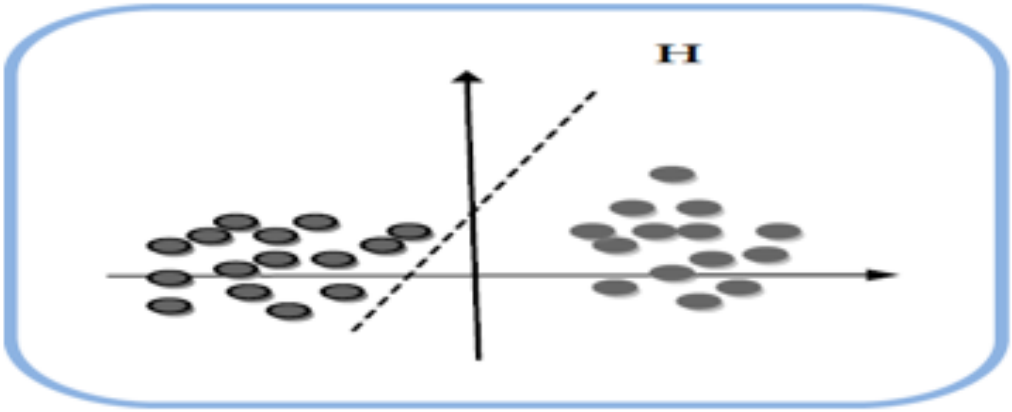


Figure 2.4 : Hyperplan H que sépare les deux ensembles des points.

5.1.2.2 Vecteur support

Pour une tache de détermination de l'hyperplan séparable de SVM, on utilise seulement les points les plus proches « i.e. les points de la frontière entre les deux classes des données » parmi l'ensemble total d'apprentissage, ces points sont appelés vecteurs supports **Figure 2.5**. [27]

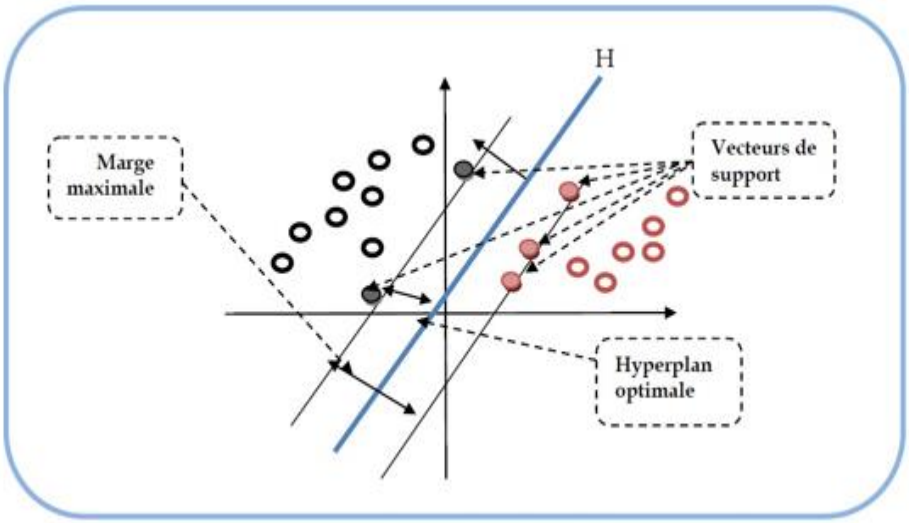


Figure 2.5 : Hyperplan H optimale, vecteurs supports et marge maximale.

5.1.2.3 Marge

Il existe une infinité d'hyperplans capable de sépare parfaitement les deux classes d'exemples. Le principe des SVM est de choisir celui qui va maximiser la distance minimale entre l'hyperplan et les exemples d'apprentissage « i.e. la distance entre l'hyperplan et les vecteurs supports », cette distance est appelée la marge **Figure2.4. [**26]

5.1.2.4 Noyaux

Un noyau est une fonction utilisée dans SVM pour aide à résoudre des problèmes. Ils fournissent des raccourcis pour évite des calculs complexes. La chose étonnante à propos du noyau est que nous pouvons aller à des dimensions plus élevées et effectuer des calculs fluides à l'aide de celui-ci. On peut montons jusqu'à un nombre infini de dimensions en utilisant des noyaux.

Parfois, on ne peut pas avoir des hyperplans pour certains problèmes. Ce problème se pose lorsque nous montons à des dimensions supérieures et essayons de forme un hyperplan.

Le noyau aide à forme l'hyperplan dans la dimension supérieure sans augmenter la complexité.

Exemple de noyaux

- Le noyau linéaire : est un simple produit scalaire :

$$k(x, z) = (x, z)$$

- Le noyau polynomial : permet de représenter des frontières de décision par des polynômes de degré d :

$$K(x, z) = (a * (x, z) + b)^d$$

- Le noyau RBF (Radial Basis Function) : noyau très utilisé dans la pratique qui s'évalue selon :

$$k(x, y) = \exp\left(-\alpha . \left|\left|x^{\beta} y\right|\right|^2\right)$$

5.2 Naïve bayse

Les classificateurs bayésiens sont des classificateurs statistiques. Ils peuvent prédire les probabilités d'appartenance à une classe, telles que la probabilité qu'un échantillon donné appartienne à une classe particulière.

Le classificateur bayésien est basé sur le théorème de Bayes. Les classificateurs bayésiens naïfs supposent que l'effet d'une valeur d'attribut sur une classe donnée est indépendant des valeurs des autres attributs.

Cette hypothèse est appelée indépendance conditionnelle de classe. Il est fait pour simplifie le calcul impliqué, et dans ce sens, est considéré comme « naïf ». [28]

5.2.1 Théorème de bayes

Soit X = {x1, x2, . . . , xn} un échantillon dont les composants représentent des valeurs faites sur un ensemble de n attributs.

En termes bayésiens, X est considéré comme une « preuve ». Soit H une hypothèse, par exemple que les données X appartiennent à une classe spécifique C. Pour les problèmes de classification, notre objectif est de déterminer P(H|X), la probabilité que l'hypothèse H soit vérifiée étant donné la « preuve », (c'est-à-dire l'échantillon de données observé X). En d'autres termes, nous recherchons la probabilité que l'échantillon X appartienne à la classe C, étant donné que nous connaissons la description de l'attribut de X. P(H|X) est la probabilité a posteriori de H conditionnée sur X.

$$P(H/X) = \frac{P(X|H)P(H)}{P(X)} \qquad [29]$$

P(H/X) : Probabilité postérieure de classe « y, cible » donnée prédicteur « x, attributs ».

P (X/H): La vraisemblance qui et la probabilité du prédicteur de la classe donnée.

P (H): Probabilité a priori de classe.

P (X): Probabilité a priori du prédicteur.

5.2.2 Classificateur bayésien naïf

Le classificateur bayésien naïf fonctionne comme suit :

1. Soit T un ensemble d'échantillon d'apprentissage, chacun avec ses étiquettes de classe. Il existe k classes : C1, C2, …. ,ck. Chaque échantillon est représenté par un vecteur à n dimensions, X = {x1, x2, . . . , xn}, représentant n valeurs mesurées des n attributs : A1, A2, . . . , A n, respectivement.

2. Étant donné un échantillon X, le classificateur prédira que X appartient à la classe ayant la probabilité a posteriori la plus élevée, conditionnée sur X. C'est-à-dire que X est prédit appartenir à la classe Ci si et seulement si : P (Ci |X) > P (Cj |X) for $1 \leq j \leq m, j \neq i$.

On trouve ainsi la classe qui maximise P(Ci|X). La classe Ci pour laquelle P(Ci|X) est maximisée et appelée hypothèse du maximum a posteriori. Par le théorème de Bayes :

$$P(Ci\ |X) = \frac{P(X|Ci)\ P(Ci)}{P(X)}$$

3.Comme P(X) est la même pour toutes les classes, seuls P(X|Ci) P(Ci) ont besoin être maximisée. Si les probabilités a priori de classe, P(Ci) sont n'est pas connu, alors on suppose généralement que les classes sont également probables, c'est-à-dire que : P(C1) = P(C2) = . . . = P(Ck), et on maximiserait donc P (X | Ci), Sinon on maximise P (X |Ci)P(Ci).

Notez que les probabilités a priori de classe peuvent être estimées par :

P(Ci) = freq (Ci, T) /|T|.

4.Étant donné des ensembles de données avec de nombreux attributs, il serait coûteux en calcul de P(X|Ci). Afin de réduire le calcul lors de l'évaluation de P(X|Ci) P(Ci), l'hypothèse naïve d'indépendance conditionnelle de classe est faite.

Cela suppose que les valeurs des attributs sont conditionnellement indépendantes les unes des autres, compte tenu de l'étiquette de classe de l'échantillon. Mathématiquement, cela signifie que : $P(X|Ci) \approx \prod_{k=1}^{n} P(xk|Ci)$.

Les probabilités : P(x1|Ci), P(x2|Ci), . . . , P(xn|Ci) peut facilement être estimée à partir de l'ensemble d'apprentissage.

Rappelons qu'ici xk fait référence à la valeur de l'attribut Ak pour l'échantillon X.

(a) Si Ak est catégoriel, alors P(xk|Ci) est le nombre d'échantillons de classe Ci dans T ayant la valeur xk pour l'attribut Ak, divisé par freq (Ci, T), le nombre d'échantillons de classe Ci en t.

(b) Si Ak est à valeur continue, alors nous supposons généralement que les valeurs ont une distribution gaussienne avec une moyenne μ et un écart type σ définis par

$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp - \frac{(x-\mu)^2}{2\sigma^2}$, Pour que : $p(xk|Ci) = g(xk, \mu Ci, \sigma Ci)$.

Nous devons calcule μCi et σCi, qui sont la moyenne et l'écart type des valeurs de l'attribut Ak pour les échantillons d'apprentissage de la classe Ci.

5. Afin de prédire l'étiquette de classe de X, P(X|Ci) P(Ci) est évalué pour chaque classe Ci. Le classifieur prédit que la classe étiquette de X est Ci si et seulement si c'est la classe qui maximise P(X|Ci) P(Ci). [30]

5.3 Forets aléatoires

Forets aléatoires ou Random Forest « algorithme des forets d'arbres décisionnels » est un algorithme d'apprentissage supervisée utilise pour les problèmes de classification et de régression introduites par Breiman en 2001 [31]. On peut connaitre le concept de « Forets » d'après son nom, le foret elle groupe plusieurs arbres de décision par la méthode bagging « bootstrap aggregating ».

L'idée de la méthode bagging n'est qu'une structure de modèles d'apprentissage augmente le résultat global.

Le principe des méthodes de bagging, et en particulier des forêts aléatoires, est de faire la moyenne des prévisions de plusieurs modèles indépendants pour réduire la variance et donc l'erreur de prévision. Pour construire ces différents modèles, on sélectionne plusieurs échantillons bootstrap, c'est à dire des tirages avec remises. En plus du principe de bagging, les forêts aléatoires ajoutent de l'aléa au niveau des variables.

Pour chaque arbre, on sélectionne un échantillon bootstrap d'individus et à chaque étape, la construction d'un nœud de l'arbre se fait sur un sous-ensemble de variables tirées aléatoirement.

La prédiction se base alors sur un sous-ensemble de la forêt. Pour un muplet « X1,…..,Xm » de variable explicative, on obtient un ensemble yi,yj ∈[0 ;k].

Avec k le nombre d'arbres de la forêt créée. On obtient plusieurs arbres et des prédictions différentes pour chaque individu. Lorsque le nombre d'arbres souhaité a été simulé, on peut obtenir la prédiction de deux manières différentes selon l'approche utilisée « classification ou régression ». [20]

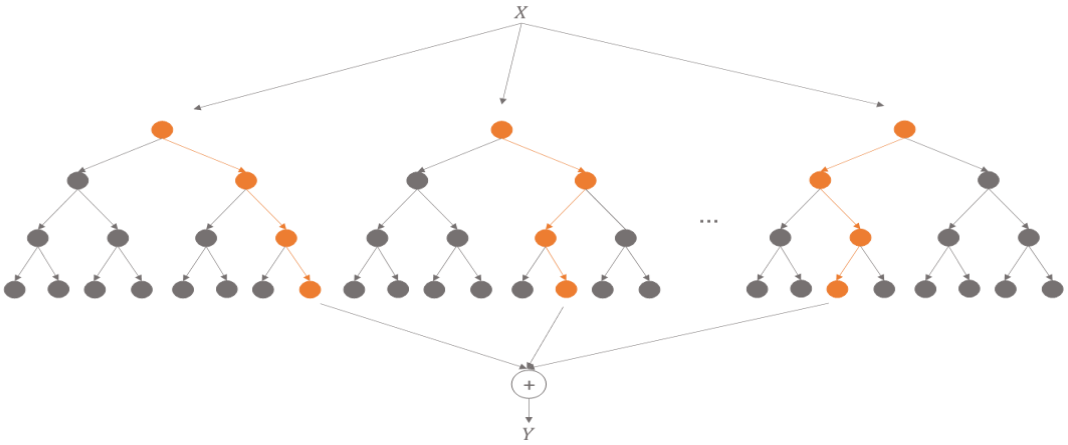


Figure 2.6: Principe de l'algorithme des forêts aléatoires. [20]

5.4 K-Nearest Neighbor « KNN »

K-NN est l'un des algorithmes de classification les plus simples à teste et le plus rapide à mettre en œuvre.

L'algorithme est basé sur un vote majoritaire des k-plus proches voisins, basé sur la distance Euclidienne dans l'espace de caractéristique pour prédire les classes de nouvelles données à partir de données étiquetées « données d'apprentissage », où k spécifie le nombre de voisins à utilise. Les exemples d'apprentissage sont des vecteurs dans un espace de caractéristiques multidimensionnel, chacun avec une étiquette de classe, cet algorithme ne nécessite pas d'étape d'apprentissage pour être effectué mais peut être réglé pour détermine la valeur optimale de k sur laquelle baser la classification.

Les données d'apprentissage ont été mises à l'échelle de sorte qu'elles aient une moyenne de 0 et un écart type de 1 [32].

En effet, la phase d'apprentissage de l'algorithme K-NN consiste seulement à stocke les vecteurs de caractéristiques et les étiquettes de classe des échantillons d'apprentissage. Dans la phase de classification, k est une constante définie par l'utilisateur et un vecteur non étiqueté « une requête ou un point de test » est classé en affectant l'étiquette la plus fréquente parmi les k échantillons d'apprentissage les plus proches de ce point de requête.

La distance Euclidienne est une mesure de distance couramment utilisée pour les variables continues. Pour les variables discrètes, telles que la classification de texte, une autre mesure peut être utilisée, telle que la mesure de chevauchement « ou distance de Hamming ».

La distance Euclidienne:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Où : x, y sont des vecteurs.

La distance de Minkowsky:

$$d(x,y) = \left(\sum_{i=1}^{n}|x_j - y_i|^p\right)^{1/p}$$

Où : x , y sont des vecteurs.

p : paramètre

La distance de Manhattan :

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

Où : x, y sont des vecteurs.

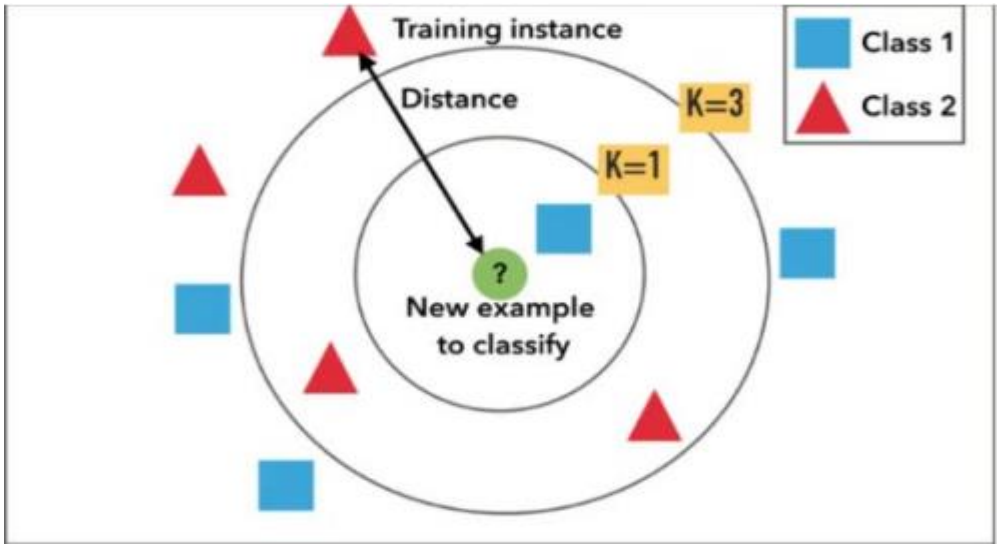


Figure 2.7: Algorithme k-Nearest Neighbor. [8]

5.5 Arbre de décisions

Les arbres de décision ou decision tree « en anglais » sont des méthodes d'apprentissages supervises utilise pour la régression et la classification.

Un arbre de décision est un enchaînement hiérarchique de règles logiques qui permettent de diviser la base d'exemples en sous-groupes, en fonction de la valeur des variables d'entrées. L'arbre est construit en recherchant à chaque niveau le paramètre

le plus discriminant pour classifier un exemple « Naffakhi, 2004 ». Différents algorithmes peuvent être utilisés pour développer l'arborescence. Les arbres de décision suscitent un large engouement en fouille de données, car ils sont simples et rapides tout en restituant de manière compréhensible les relations existantes entre les variables d'entrée et le phénomène à modéliser. [9]

6 Conclusion

L'apprentissage automatique est un paradigme important et largement utilisé pour de nombreux problèmes. Dans ce chapitre nous avons passé sur les concepts essentiels de la classification. En particulier nous avons présenté la machine learning. Ensuite on a défini les déférents algorithmes d'apprentissage utilise dans notre travail.

Partie 02 : Conception et Implémentation

La partie suivante présente la méthodologie qui nous avons proposé pour l'évaluation et la comparaison des modèles de classification du trafic réseau. Nous avons divisé cette partie en deux chapitres, le premier chapitre présente la méthodologie de travail, et le deuxième chapitre présente l'implémentation de la méthodologie proposée ainsi que les résultats obtenus.

Chapitre 03: Conception

1 Introduction

Dans ce chapitre, nous décrivons la méthodologie que nous avons suivie pour évalue et compare les modèles de la classification du trafic réseau. On a commencé par la présentation de notre architecture globale de système. Ensuite, nous allons présenter, en détails, les différentes étapes constituant cette architecture à savoir : le DataSet, le processus du prétraitement « Pre-Processing », les méthodes de validation, les algorithmes de classification et les métriques de l'évaluation de performance utilisées.

2 Architecture

L'objectif de cette étude est de propose un système de classification du trafic réseau à base de machine learning. Le système est capable d'identifier les déférents types du trafic réseau à savoir : FTP, WWW, MAIL…. etc. L'architecture proposée pour ce système de classification du trafic réseau est montrée dans la **Figure 3.1**. L'architecture est composée de plusieurs étapes qui seront présentées dans les sous-sections suivantes.

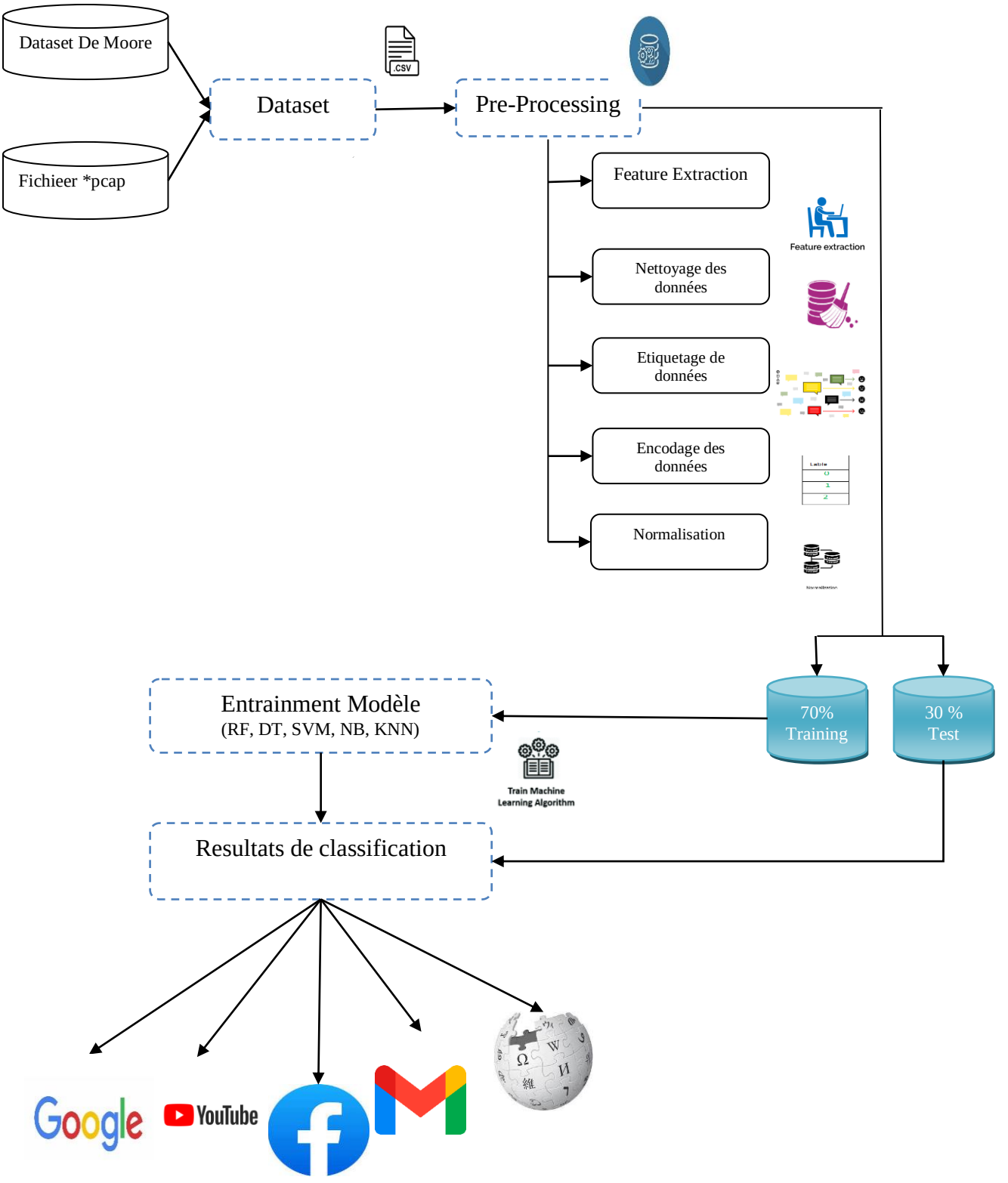


Figure 3.1 : Architecteur du système utilisée pour la classification du trafic réseau.

2.1 DataSet

- DataSet de Moore et Zuef :

  Dans cette étude, nous utilisons l'ensemble de données créé par Moore et Zuev, Il se compose de 10 ensembles de données surveillés à différents moments de la journée à partir d'un seul site internet [33].

- Ajout des catégories à partir d'un fichier *pcap :

  Nous ajoutons quelque classe aux premières DataSet à travers les données collectées dans la section réseau de l'Université du Cauca, « Popayan, Colombie, Unicauca-Version2-atts » en effectuant des captures de paquets à différents moments, matin et après-midi, pendant six jours, et converties et actuellement stockées dans un fichier CSV.

2.2 Pre-Processing

La plupart du donnée dans tous les ensembles de donnée contient des données manque, des donnée aberrantes, des données null ou elle contient plusieurs type catégoriel comme objet, entier, ….

Notre DataSet et comme tous les autres ensembles de donnée contient ces éléments et pour faire une bonne classification, la seule solution et appliquée la Pre-Processing.

Pre-Processing elle se compose de plusieurs étapes :

2.2.1 Nettoyage des données

Le nettoyage des données inclut la vérification de la qualité et la cohérence des données, la correction des erreurs et des données erronées, la suppression des valeurs manquantes, la suppression des valeurs extrêmes et excentrées.

- **Les valeurs manquantes**

  En pratique, il n’y a pas de données complètes. Malgré la quantité croissante de donnée disponible, la problématique de données manquantes reste existée dans différant forme comme « N/A », « null », « Empty », « ? », « none », « empty », « - », « NaN », « -999 », « inf », « -inf ».

- **Les valeurs aberrantes**

  La valeur aberrante est la valeur qui diffère de façon significative la tendance globale des autres observations quand on observe un ensemble de données ayant des caractéristiques communes. Les données aberrantes sont un grand problème pour de nombreux analyse, car le traitement de ces valeurs elle est complexe par exemple Les valeurs aberrantes ne sont pas forcément erronées.

2.2.2 Feature extraction

2.2.2.1 Sélection des attributs manuellement :

La sélection des attributs est une opération très importante avant l'utilisation des algorithmes d'apprentissage. Elle consiste de sélectionnée les attributs qui on va utilise pour faire la classification.

L'ensemble de donne de notre étude est compose de 248 Attribut différentes, chacune ayant une distribution variée dans les ensembles de données et associée à des coûts de collecte/calcul différents.

Maintenant, ce que nous voulons, c'est trouvé un sous-ensemble de cet ensemble de l'attributs mais contenant suffisamment d'informations qui conduisent à la précision souhaitée, Dans notre étude de l'ensemble de données et notre recherche sur le trafic réseau, nous avons sélectionné des attributs, qui selon nous pourraient affecte la précision de la classification **« Tableau 3.1 ».**

Tableau 3.1 : Les attributs utilisés dans notre étude « Feature extraction ».

Numero de feature	Nom de feature	Description
1	Serveur_port	Numéro de port du serveur.
2	Client_port	Numéro de port du client.
3	Push_packet_serv	Le nombre de tous les paquets vus avec le bit PUSH défini dans l'en-tête TCP.
4	Avg_seg_size_client	La taille moyenne du paquet.
5	Init_window_client	Le nombre de byte totale envoie du début de window « client vers serveur ».
6	Init_window_serveur	Le nombre de byte totale envoie du début de window «serveur vers client».
7	Mean_IAT	Le temps moyen entre deux paquets envoyés dans la direction « Forward » (temps relative).
8	Total_paket_a-b	Le nombre total de paquets du flux dans la direction « Forward » (client → serveur).

4.2.2.2 Les méthodes warpper et filtre :

- **La méthode Warpper :**

Les méthodes « wrapper » utilisent les algorithmes de classification pour générer et ensuite évaluer la qualité des ensembles candidats. Cette approche est généralement pertinente mais elle dépend de la représentativité de l'ensemble d'apprentissage.

Nous utiliserons deux méthodes pour faire la sélection des attributs notamment : forward warpper, Backward warpper.

- **La méthode Filtre :**

Les méthodes « filtres » sont basées sur une fonction critère pour mesurer la pertinence de l'information contenue dans les ensembles candidats. Elles réduisent considérablement le temps de calcul, alors que les méthodes « wrapper » permettent d'obtenir des meilleurs résultats.

2.2.3 L'étiquetage de donnée

Dans notre ensemble de donnes il existe 12 catégorie de trafic réseau notamment « ATTACK, DATABASE, FTP-CONTROL, FTP-DATA, FTP-PASV, INTERACTIVE, GAME, MAIL, MULTIMEDIA, P2P, SERVICE, WWW ».

Nous supprimons chaque catégorie contenant un petit nombre d'échantillons, et regroupons les catégories « FTP-PASSIVE, FTP-DATA, FTP-CONTROL » en une seule catégorie qui représente une catégorie Bull (FTP).

2.2.4 L'encodage des donnes « Encoder Label »

Lorsque vous travaillez avec un ensemble de données ayant des caractéristiques catégorielles, de nombreux algorithmes d'apprentissage automatique nécessitent que les données catégorielles « étiquettes » soient converties ou encodées dans le format numérique.

C'est une partie importante du prétraitement des données d'encoder les étiquettes « label » de manière appropriée sous forme numérique afin de s'assure que l'algorithme d'apprentissage interprète correctement les attributs.

2.2.5 Normalisation

La normalisation est une méthode de prétraitement des données qui permet de réduire la complexité des modèles. C'est également un préalable à l'application de certains algorithmes, nous utilisons dans notre travail Min-Max Normalisation.

2.3 Validation du modèle

Dans cette partie, on test et valide le modèle et ces paramètres selon des critères ce base sur ces résultats pour trouve les meilleurs modèles. La méthode qui on a travaillée avec lui est Hold-out.

- **Hold-out :**

  La méthode hold-out est le type de validation croisée le plus simple. L'ensemble de données est séparé en deux ensembles, appelés l'ensemble d'apprentissage et l'ensemble de test.

  Le training est sur quoi le modèle est formé, et le test pour voir dans quelle mesure ce modèle fonctionne sur les données, dans notre étude on choisir le training de 70% et le test 30%.

2.4 Les algorithmes de classification

Dans notre étude on a implémente en python par la bibliothèque Scikit learn 5 classificateur d'apprentissage machine supervise, comme décrit dans le « **Tableau 3.2** » pour comparer performances de chaque système. Ces algorithmes et sont paramètre sont défini dans le tableau ci-dessus.

Tableau 3.2 : Type et paramètre d'algorithme d'apprentissage.

Classifier	Paramètre	Valeur
K-Naerest Neighbor(KNN)	n_neighbor	1
Support Vector Machine (SVM)	Kernel Gamma	Rbf scale
Naïve Bayes(NB)	/	/
Random Forest(RF)	Max_depth n_estimators	100 100
Decision tree(DT)	Max_depth n_estimateur	100 100

Il est intéressant de souligne que dans l'algorithme de KNN ont prendre la valeur qui donnée la meilleur accuracy et pour cella on a défini une fonction qui donne les meilleurs valeurs « k » pour notre classification, la même idée pour l'algorithme SVM on a travaillée avec kernel « RBF » car elle donne les meilleurs résultats par rapport aux kernel linear, polynomial ou sigmoïde.

2.7 Métriques d'évaluation :

Pour évalue les performances des algorithmes d'apprentissage automatique, nous considérons diverses mesures : Accuracy, Précision, Recall, et F1-score.

- **Accuracy**

  L'accuracy est la ration entre les échantillons de flux du trafic correctement classés et le nombre total d'échantillons.

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

  Où

  TP : Vrais Positifs représentent le nombre de flux du trafic correctement classés dans la classe à laquelle ils appartiennent.

  TN : Vrais Négatifs représentent le nombre de flux du trafic correctement classés comme ne correspondant pas à une classe.

  FP: : Les Faux Positfs représentent le nombre de flux de trafic mal classés dans une classe.

  FN: Les Faux Négatifs représentent le nombre de flux de trafic classés à tort comme n'appartenant pas à une classe.

  Dans certains cas, lorsque l'ensemble de données à une classe représentant la majorité des valeurs d'échantillon, la valeur du score de accuracy il ne peut pas refléter précisément les performances du modèle de classifieur.

  Pour évite ce problème, nous utilisons également d'autres mesures de performance « précision, rappel et score F1 » pour présente les performances du modèle de classifieur.

- **Précision**

  Est une mesure du rapport entre les classes de trafic positives correctement prédites et le nombre total de classifications positives.

$$\mathbf{Précision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

- **Rappel**

  Mesure la ration entre les classes de trafic réellement positives et correctement prédites.

$$\mathbf{Rappel} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

- **Score F1-mesure**

  La moyenne de la précision et du rappel.

$$\mathbf{F1-Score} = \frac{2 * \text{Precision} * \text{Rappel}}{\text{Precision} + \text{Rappel}}$$

Aussi on a utilisé des autres métriques comme la matrice de confusion et le rapport de classification.

3 Conclusion

Dans ce chapitre, nous avons présente notre système de classification du trafic réseau en utilisant le concept d'apprentissage automatique. En effet, nous avons présente une description détaillée sur le DataSet utilisé. Ensuite, nous avons décrit le processus du

Pre-Processing et les différentes étapes de traitement d'ensemble de donné. Enfin, pour faire la classification, nous avons défini l'ensemble des algorithmes d'apprentissage utilisés et les métriques pour faire l'évaluation.

L'objectif de ce chapitre est faire l'implémentation des différents algorithmes d'apprentissages et faire une comparaison entre eux.

Chapitre 04: Implémentation

1 Introduction

Dans ce chapitre, on va présenter la partie pratique de notre étude. Premièrement nous avons faire une description des outils de développement, Puis, nous allons analyser et discute les résultats obtenus par les différents algorithmes d'apprentissage.

2 Outils et technologies utilisés

2.1 Outils:

- PC personnel : PC personnel Dell « Latitude e5440 » avec un processeur Intel® Core(TM) i5-4310U CPU@ 2.00GHz 2.60 et 4GB de RAM.

2.2 Technologies :

Le modèle proposé a été implémenté en visuale studio avec un environnement de programmation qui possède plusieurs bibliothèque d'apprentissage automatique :

- Python 3.9.1: Le choix du langage de programmation est Python. Il s'agit d'un langage orienté objet, facile à prendre en main, l'un des plus utilisé pour faire du machine learning. Python possède un grand nombre de librairies: d'analyse de données, de calcul scientifique, de visualisation et d'apprentissage automatique.
- Pandas: Pandas est une librairie Python qui nous a permis de manipuler facilement nos données pour l'analyse et le prétraitement.
- La bibliothèque Numpy : La bibliothèque Numpy permet d'effectuer des calculs numériques avec Python.
- La bibliothèque Matplotlib : Matplotlib est une bibliothèque Python qui permet de tracer et de visualiser des données sous plusieurs formes de graphiques.
- La bibliothèque seaborn : Est une librairie qui vient compléter certaines fonctionnalités non accessibles avec Matplotlib.
- La bibliothèque Scikit Learn : Il s'agit d'une librairies d'apprentissage automatique qui permet de faire de la classification, régression du clustering et

de la réduction de dimension, la sélection de modèle et du prétraitement de données.

- KNN, Forte Aleatoire, Naïve Bayse, Decision Tree et SVM sont utilisée comme

  Des classificateurs dans notre projet

Il existe plusieurs avantages pour l'utilisation du visuale studio code pour notre étude il contient la plupart des bibliothèques courantes nécessaires à l'apprentissage automatique telles que « Scikit Learn, Numpy, pandas, matplotlib, seaborn etc ».

3 DataSet

Durant cette étude on a utilisé une DataSet et ensuit on a ajouté des nouvelles catégories au premier DataSet d'âpre un autre fichier « pcap » pour trouve l'ensemble de donnée final pour faire la classification du trafic réseau.

3.1 DataSet de moore:

Dans cette étude, nous utilisons l'ensemble de données créé par Moore et Zuev, pour appliquer des algorithmes d'apprentissage automatique pour la classification du trafic. Un ensemble de données contient des caractéristiques du trafic, telles que la diversité des sources de données, des échantillons du trafic élevé et différents types de données. L'ensemble de données a été recueilli dans un laboratoire informatique de l'Université de Cambridge.

Il se compose de 10 ensembles de données surveillés à différents moments de la journée à partir d'un seul site internet, qui hébergeait environ 1 000 utilisateurs connectés à internet via une liaison gigabit bidirectionnelle. L'ensemble de données se compose de 248 attribut, telles que la durée du flux, le numéro de port TCP, le délai moyen de traitement des paquets et la variance. L'ensemble de données contient environ 377408 échantillons basés sur les flux de trafic TCP. Les catégories avec leurs applications correspondantes sont répertoriées dans le **Tableau 4.1**.

Tableau 4.1 : Les catégories du premier DataSet et leur application.

Classification	Example Application
BULK ftp	ftp
DATABASE	postgres, sqlnet oracle, ingres
INTERACTIVE	ssh, klogin, rlogin, telnet
MAIL	imap, pop2/3, smtp
SERVICES	X11, dns, ident, ldap, ntp
WWW	www
P2P	KaZaA, BitTorrent, GnuTella
ATTACK	Internet worm and virus attacks
MULTIMEDIA	Media Player, Real

Un bon modèle de classification du trafic réseau demande des bons paramètres, pour cella nous avons choisir les meilleur feature et élimine tous les features qui ne sont pas essentielle et aussi les très petites catégories comme Game, Interactive.

Lorsqu'on a terminé la sélection des features on a fusionne les dix ensembles de donne en un seul fichier et elle donne les statistiques suivant « **Tableau 4.2** » :

Tableau 4.2 : Nombre d'occurrence de classes.

Classe	Occurrence
BULK ftp	11539
DATABASE	2410
MAIL	28567
SERVICES	2099
WWW	328092
P2P	2094
ATTACK	1793
MULTIMEDIA	576
Total	377408

3.2 Fichier « pcap » :

On utilise le DataSet « Unicauca-Version2-atts » crée par l'Université du Cauca « Popayan, Colombie », qui sont transformé tous les fichiers de cette DataSet « pcap » au format « csv » et on a travaillé avec ce fichier, les nouveaux fichiers qui on a travaillé avec lui sont montre dans le « **Tableau 4.3 »**.

Tableau 4.3 : Nombre d'occurrence de classes.

Classe	Occurrence
Google	50703
Skype	8722

Pour faire un équilibrage du premier DataSet on a ajouté des nouveaux fichiers en format « csv » à partir du DataSet « Unicauca-Version2-atts », et on a obtenu des résultats montrons dans le « **Tableau 4.4 »**.

Tableau 4.4 : Nombre de classes du notre ensemble de donnée.

Classe	Occurrence
WWW	74476
GOOGLE	50703
MAIL	28567
BULK FTP	11539
SKYPE	8722
DATABASE	2648
SERVICE	2099
Totale	178754

On a représenté les applications d'ensemble de donnes aves une cercle « **Figure 4.1 »**.

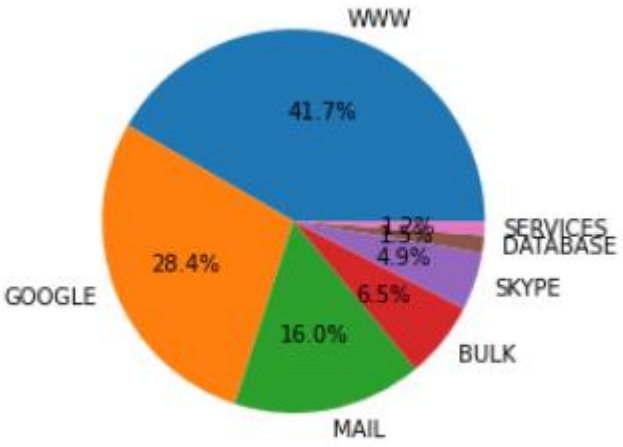


Figure 4.1 : Distribution de classe dans l'ensemble de donnée.

4 Résultats et discussion:

Cette section présente les résultats d'évaluation obtenus à partir des algorithmes d'apprentissages automatique pour la classification du trafic.

Nous avons fait deux expériences et nous présentons l'évaluation des algorithmes d'apprentissage automatique en termes de mesures de performance.

Ensuite, on compare notre meilleur algorithme d'apprentissage avec des autres articles.

4.1 Première expérience :

Dans cette étape, nous avons fait une expérimentation notre ensemble de données décrit dans (section 3), en utilisant la méthode de sélection des attributs sélectionnés manuellement.

Les mesures de performance obtenues se situent comme suit :

- **Par rapport au accuracy :**

Nous implémentons et comparons cinq algorithmes d'apprentissage automatique : SVM, DT, RF, NB « naïve bayse » et KNN. Nous avons défini différents paramètres de modèle pour chaque algorithme afin d'améliorer les mesures de performance.

Dans SVM, nous définissons un noyau RBF, RBF donnée des meilleures mesures de performance. L'accuracy de SVM a atteint 0.885729, démontrant ses hautes performances « **Figure 4.2** ».

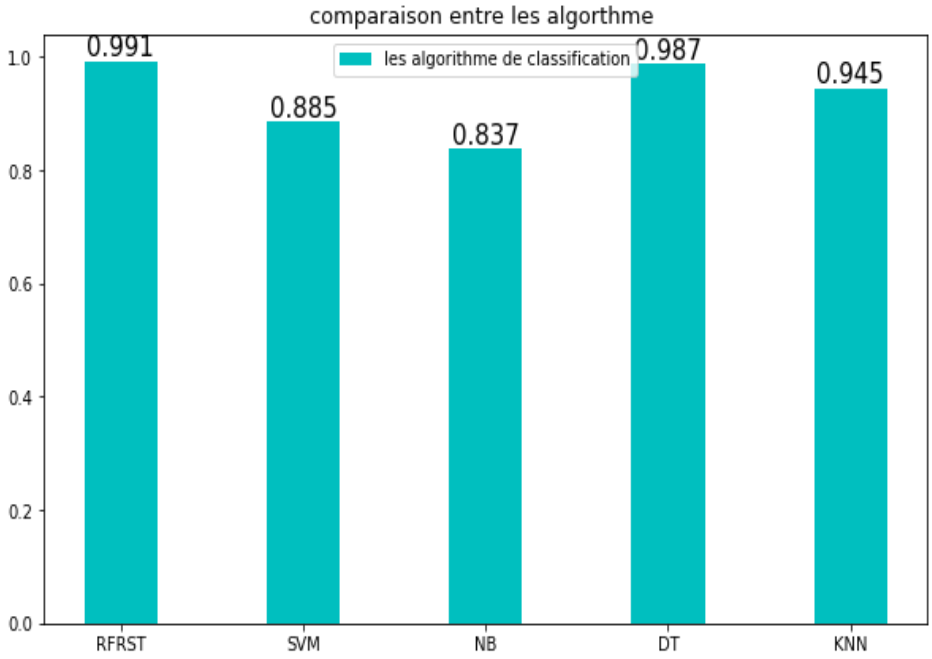


Figure 4.2 : Comparaison entre l'accuracy de tous les algorithmes de classification.

L'accuracy du trafic en utilisant SVM sont répertoriés dans le « **Tableau 4.5 »**.

Tableau 4.5 : Résultats d'accuracy entre tous les algorithmes de classification.

	Name	Accuracy	Temps
0	KNearest_Neighbors	0.9547	950ms
1	SVM	0.885729	4m49s
2	Decision_Tree	0.98718	1,36s
3	Random_Forest	0.991235	51,7s
4	Naive_Bayes	0.83737	151ms

Dans l'algorithme DT, nous avons utilisé un max_depth = 100, l'accuracy de DT a atteint 0.98718, qui est une performance remarquable « **Figure 4.2** ». L'accuracy du trafic en utilisant DT sont répertoriés dans le « **Tableau 4.5 »**.

Dans RF, on a utilisé un max_depth = 100 et n_estimators = 100. L'accuracy de RF a atteint 0.991235, ce qui est la plus grande accuracy parmi les algorithmes de classification évalués « **Figure 4.2** ». L'accuracy du trafic en utilisant RF sont répertoriés dans le « **Tableau 4.5 »**.

Dans NB, l'accuracy de NB égale à 0.83737 « **Figure 4.2 »**, L'accuracy du trafic en utilisant NB sont répertoriés dans le « **Tableau 4.5 »**.

Dans KNN, pour produit un bon algorithme K-Nearest Neighbor « KNN », on a implémenté une fonction appelée « **val_score »** qui donne le meilleur **K** pour la classification, et comme ça on obtenir des meilleures performances, Nous obtenons la meilleure valeur de k à partir d'accuracy obtenu pour k allant de 1 à 20, comme le montre la « **Figure 4.3 »,** Le résultat de cette fonction est dans « **Figure 4.3 »**.

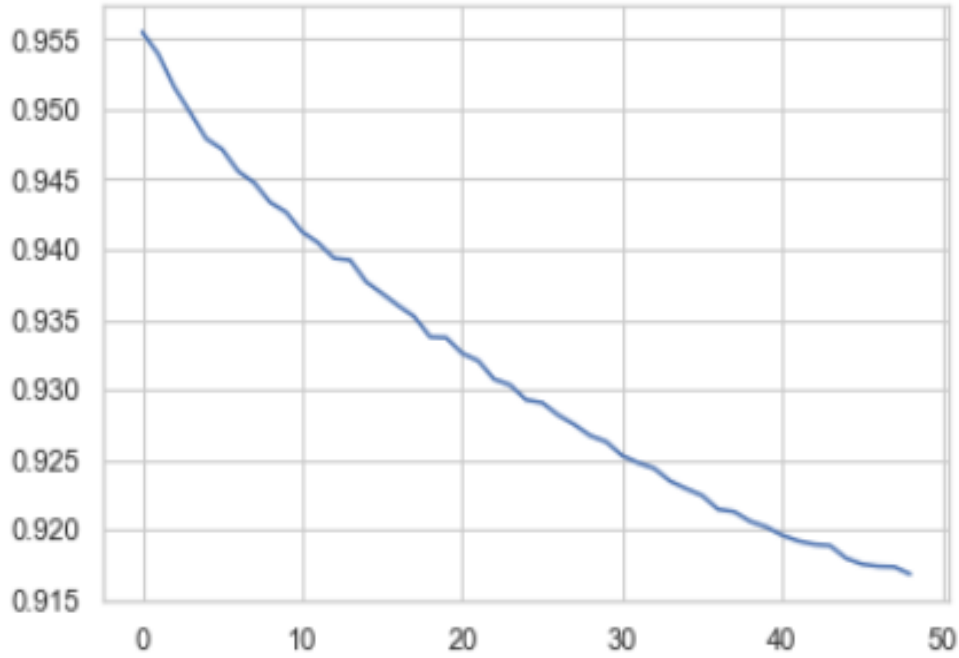


Figure 4.3 : Les bon K pour la classification « Erreur rate ».

KNN donne en pourcentage d'accuracy égale à 0.9547 « **Figure 4.2** », l'accuracy du trafic en utilisant KNN sont répertoriés dans le « **Tableau 4.5** ».

- **Par rapport au précision, recall, f1-score:**

Dans le SVM, La précision, le rappel et le score F1 par classe de trafic en utilisant SVM sont répertoriés dans le « **Tableau 4.6 »**. Les résultats démontrent que la classe WWW et la meilleur et la classes SERVICES et SKYPE ont les mesures les plus basses parmi les étiquettes de classification. La class WWW a une précision de 0.94, rappel de 0.98 et score F1 de 0.96. La classe SERVICES à une précision de 0.00, rappel de 0.00 et score F1 de 0.00, la classe SKYPE a une précision de 0.65, rappel de

0.11et score F1 de 0.19. Comme ces deux classes ont moins d'échantillons que les autres alors elle donnée ces mauvais résultats.

Tableau 4.6: Les performances de mesure d'algorithme SVM.

Traffic Classe	Précision	Recall	F1-score
BULK	0.83	0.62	0.71
DATABASE	0.97	0.69	0.80
GOOGLE	0.85	0.97	0.90
MAIL	0.80	0.87	0.84
SERVICES	0.00	0.00	0.00
SKYPE	0.65	0.11	0.19
WWW	0.94	0.98	0.96

L'algorithme decision tree « DT » donne des résultats qui sont montre dans le « **Tableau 4.7 »**. Les résultats démontrent que les classes WWW, BULK, MAIL, DATABASE, SERVICES sont les meilleurs et la classe SKYPE a la mesure la plus basses parmi les étiquettes de classification. La classe WWW, BULK, DATABASE, SERVICES et MAIL tous ces classes à une précision de 1.00, rappel de 1.00 et score F1 de 1.00. La classe SKYPE a une précision de 0.87, rappel de 0.88 et score F1 de 0.87.

Tableau 4.7 : Les performance de mesure d'algorithme DT.

Traffic Classe	Précision	Recall	F1-score
BULK	1.00	1.00	1.00
DATABASE	1.00	1.00	1.00
GOOGLE	0.98	0.98	0.98
MAIL	1.00	1.00	1.00
SERVICES	1.00	1.00	1.00
SKYPE	0.87	0.88	0.87
WWW	1.00	1.00	1.00

L'algorithme RF donne la plus grande accuracy parmi les algorithmes de classification évalués. Les classes BULK, DATABASE, MAIL, SERVICES et WWW sont les meilleurs, la classe SKYPE a la mesure la plus basses parmi les

étiquettes de classification. La classe BULK, DATABASE, MAIL, SERVICES et WWW tous ces classes à une précision de 1.00, rappel de 1.00 et score F1 de 1.00. La classe SKYPE a une précision de 0.96, rappel de 0.86 et score F1 de 0.91. Le « **Tableau 4.8 »** montre tous les résultats de classification.

Tableau 4.8 : Les performance de mesure d'algorithme RF.

Traffic Classe	Précision	Recall	F1-score
BULK	1.00	1.00	1.00
DATABASE	1.00	1.00	1.00
GOOGLE	0.98	0.99	0.99
MAIL	1.00	1.00	1.00
SERVICES	1.00	1.00	1.00
SKYPE	0.96	0.86	0.91
WWW	1.00	1.00	1.00

Dans NB, l'accuracy et l'inférieur à celui des autres algorithmes, la classe MAIL et la meilleur et la classes SKYPE a la mesure la plus basses parmi les étiquettes de classification. La class MAIL a une précision de 0.90, rappel de 0.99 et score F1 de 0.94. La classe SKYPE a une précision de 0.50, rappel de 0.15 et score F1 de 0.23. La précision, le rappel et le score F1 par classe de trafic en utilisant NB sont répertoriés dans le « **Tableau 4.9 »**.

Tableau 4.9 : Les performance de mesure d'algorithme NB.

Traffic Classe	Précision	Recall	F1-score
BULK	0.85	0.53	0.65
DATABASE	0.89	0.96	0.92
GOOGLE	0.85	0.71	0.78
MAIL	0.90	0.99	0.94
SERVICES	0.71	0.90	0.79
SKYPE	0.50	0.15	0.23
WWW	0.82	0.99	0.89

L'algorithme KNN donne des résultats qui sont montre dans le « **Tableau 4.10 »**. Les résultats démontrent que la classe DATABASE et la meilleur et la classe SKYPE a la mesure la plus basses parmi les étiquettes de classification. La classe DATABASE a une précision de 1.00, rappel de 1.00 et score F1 de 1.00, la classe WWW a une précision de 0.99, rappel de 0.99 et score F1 de 0.99. La classe SKYPE a une précision de 0.66, rappel de 0.59 et score F1 de 0.55.

Tableau 4.10 : Les performance de mesure d'algorithme KNN.

Traffic Classe	Précision	Recall	F1-score
BULK	0.97	0.97	0.97
DATABASE	1.00	1.00	1.00
GOOGLE	0.93	0.94	0.94
MAIL	0.98	0.98	0.98
SERVICES	0.93	0.98	0.96
SKYPE	0.66	0.59	0.55
WWW	0.99	0.99	0.99

Pour facilite la conception de tous les résultats on a faire une comparaison de tous les résultats de classification (Précision « **Figure 4.4 »**, Recall « **Figure 4.5 »**, F1-score « **Figure 4.6 »**) par des graphes comme ci-dessus.

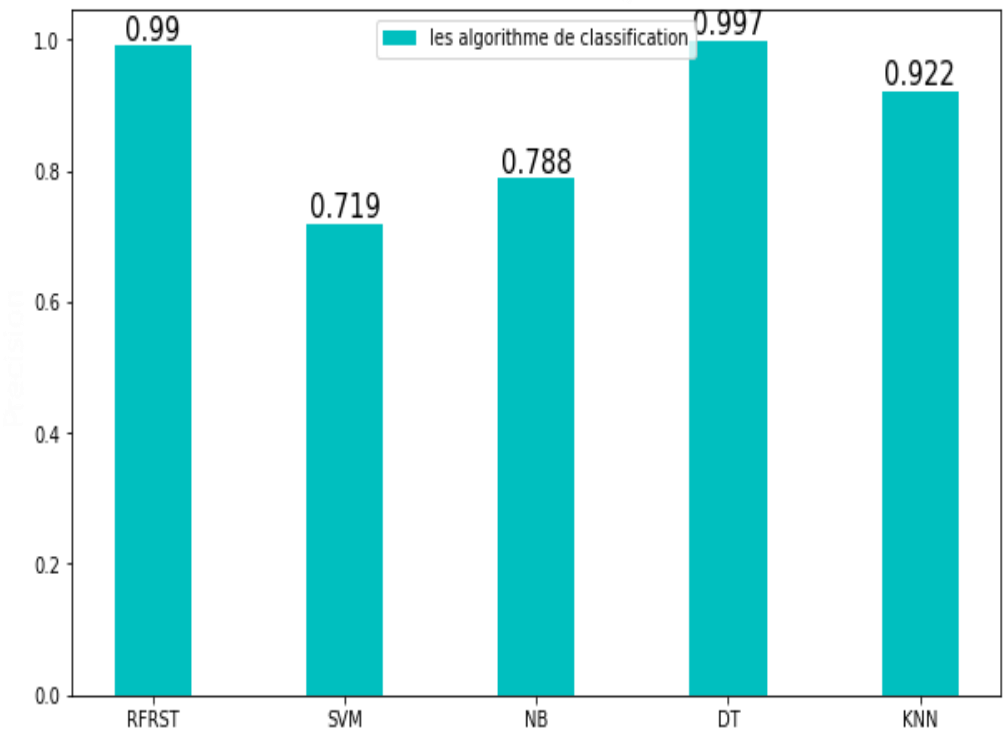


Figure 4.4 : Comparaison « Précision » entre les algorithmes de classification.

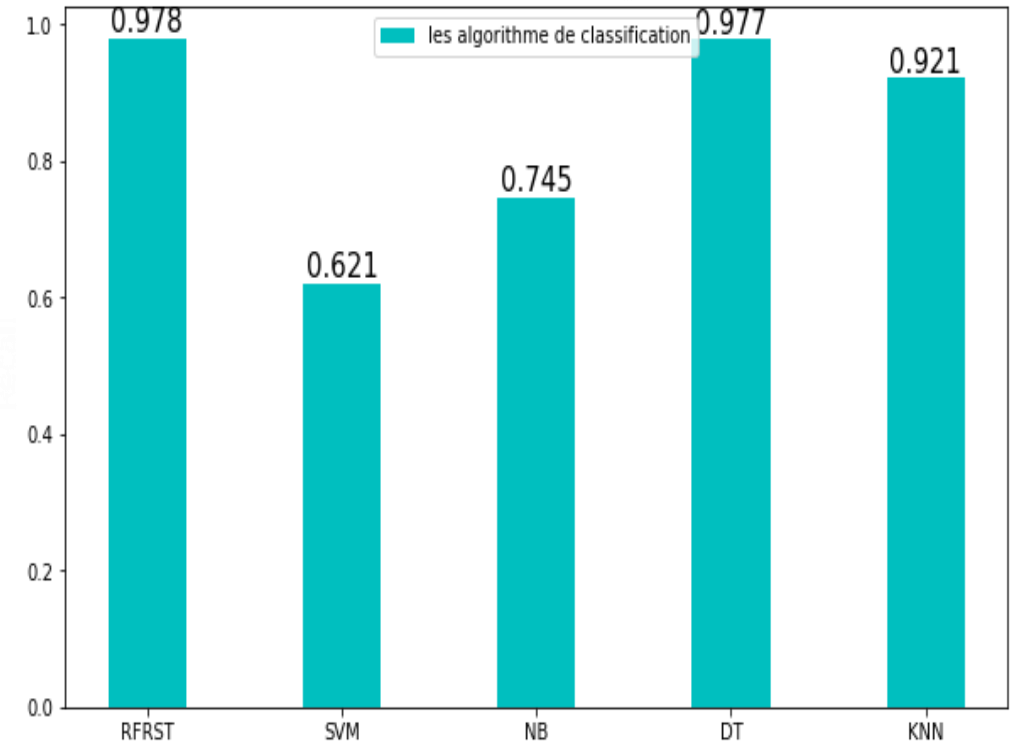


Figure 4.5 : Comparaison « Recall » entre les algorithmes de classification.

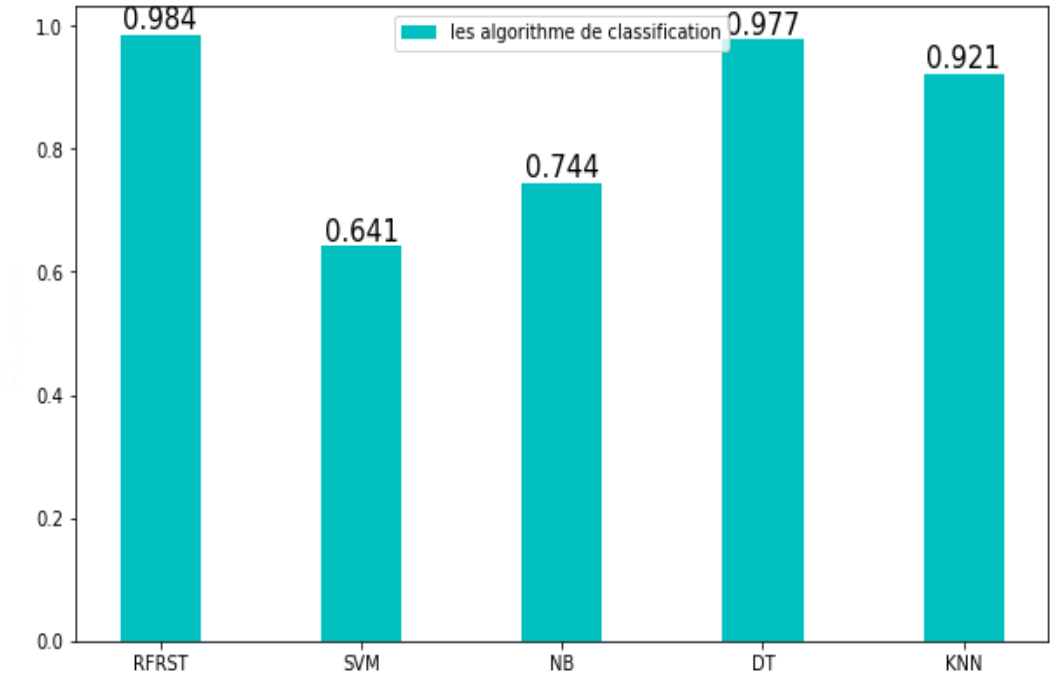


Figure 4.6 : Comparaison « F1_score » entre les algorithmes de classification.

4.2 Deuxième expérience :

Dans cette étape, nous avons fait une expérimentation **sur** ensemble de données de moore avec des méthodes de sélection des attributs (la méthode de warpper, la méthode de filter) et faire une comparaison avec les attributs que nous avons sélectionné manuellement.

4.2.1 la méthode warpper :

- **Forward warpper :**

Les mesures de performances obtenues sont situées dans le « **Tableau 4.11 »** suivant :

Tableau 4.11 : Performances de mesure pour tous les algorithmes de classification, en utilisant la méthode du Forward warpper.

	Name	Accuracy	précision	Recall	F 1_score
0	KNearest_Neighbors	0.993	0.978	0,976	0.977
1	svm	0.795	0.680	0.635	0.51
2	Decision_Tree	0.996	0.990	0.981	0.985
3	Random_Forest	0.9971	0.990	0.982	0.986
4	Naive_Bayes	0.291	0.5964	0.516	0.363

Le graphe illustré à la **« Figure 4.7 »** représente les performances de mesures.

Nous observons que le **RF** et **KNN** et **DT** présentent les meilleures performances avec une Accuracy de **99%.** Et **SVM** donner une Accuracy 79 %. En outre, l'accuracy de NB est très faible par rapport (**29%**).

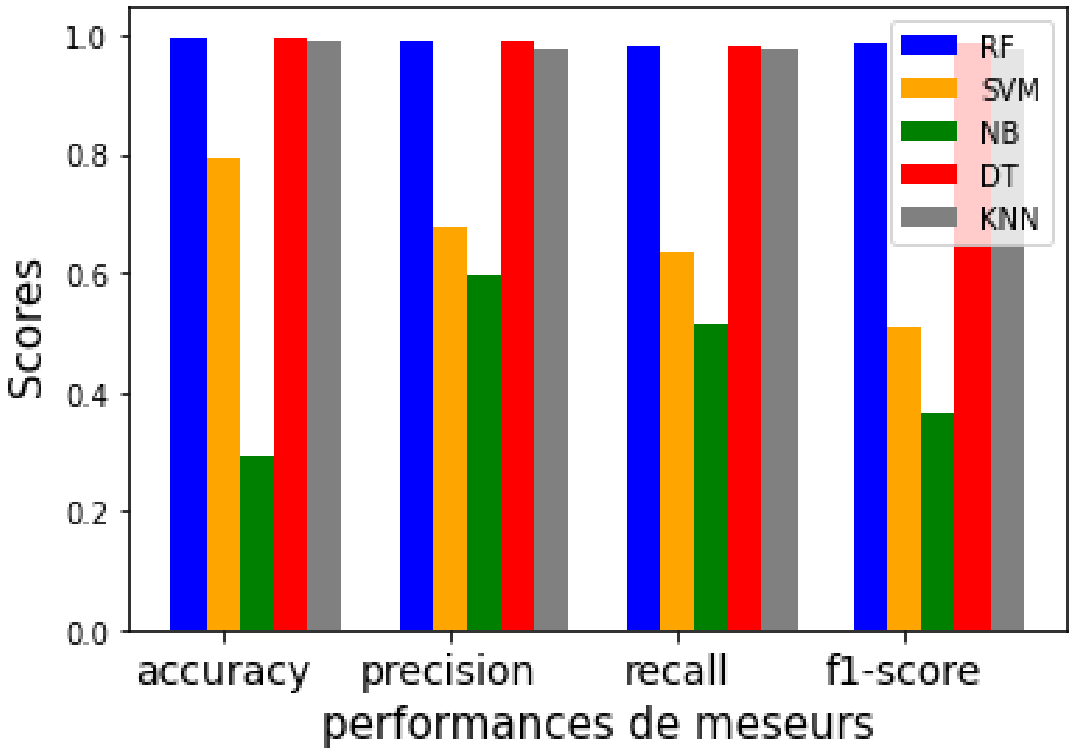


Figure 4.7 : Performances de mesure pour tous les algorithmes de classification, en utilisant la méthode du forward warpper.

- **Backward warpper :**

Les mesures de performances obtenues sont situées dans le « **Tableau 4.12 »** suivant :

Tableau 4.12 : Performances de mesure pour tous les algorithmes de classification, en utilisant la méthode du Backward warpper.

	Name	Accuracy	précision	Recall	F 1_score
0	KNearest_Neighbors	0.988	0.952	0,977	0.961
1	Svm	0.778	0.81	0.635	0.655
2	Decision_Tree	0.996	0.992	0.981	0.984
3	Random_Forest	0.996	0.991	0.982	0.984
4	Naive_Bayes	0.367	0.468	0.516	0.321

Le graphe illustré à la **« Figure 4.8 »** représente les performances de mesures.

Nous observons que le **RF** et **DT** présentent les meilleures performances avec une Accuracy de **99%.** Et **KNN** avec une accuracy **98%,** et **SVM** donner une Accuracy 77 %. En outre, l'accuracy de NB est très faible par rapport (**36%**).

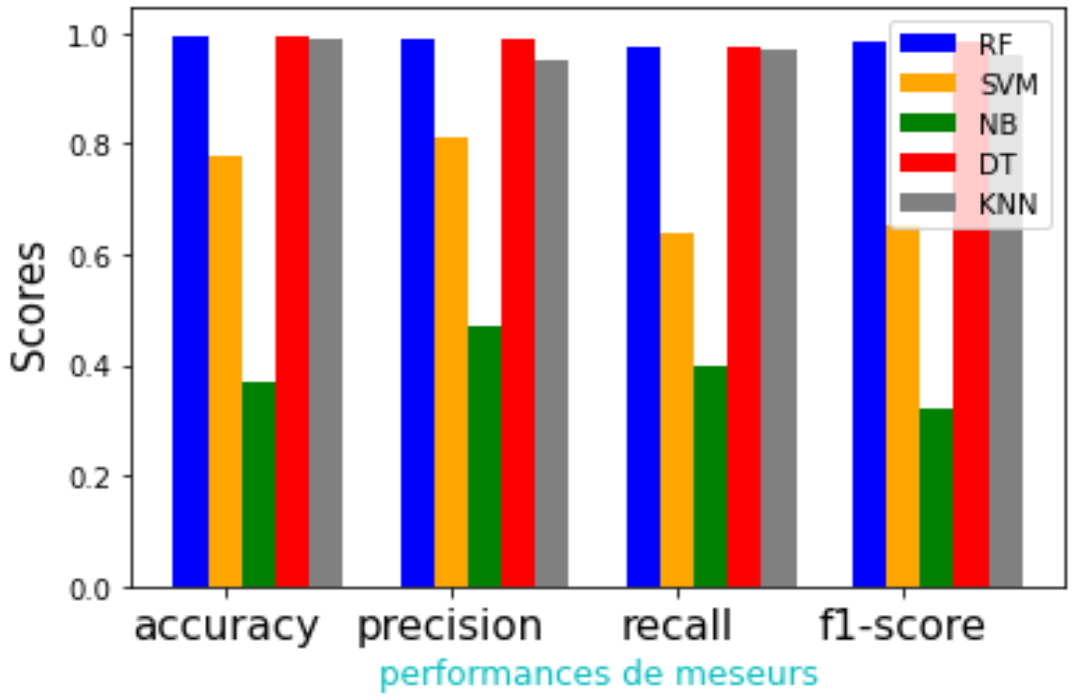


Figure 4.8 : Performances de mesure pour tous les algorithmes de classification, en utilisant la méthode du Backward warpper.

4.2.2 La méthode Filter :

Les mesures de performances obtenues sont situées dans le « **Tableau 4.13 »** suivant :

Tableau 4.13 Performances de mesure pour tous les algorithmes de classification, en utilisant la méthode du filtre.

	Name	Accuracy	précision	Recall	F 1_score
0	KNearest_Neighbors	0.996	0.990	0,985	0.987
1	Svm	0.981	0.961	0.995	0.995
2	Decision_Tree	0.998	0.992	0.991	0.9995
3	Random_Forest	0.999	0.995	0.995	0.995
4	Naive_Bayes	0.915	0.815	0.836	0.810

Le graphe illustré dans la **« Figure 4.9 »** représente les performances de mesures.

Nous observons que le **RF** et **DT** et **KNN** présentent les meilleures performances avec une Accuracy de **99%,** et **SVM** donner une Accuracy 98 %, et l'accuracy de NB est « **91%** ».

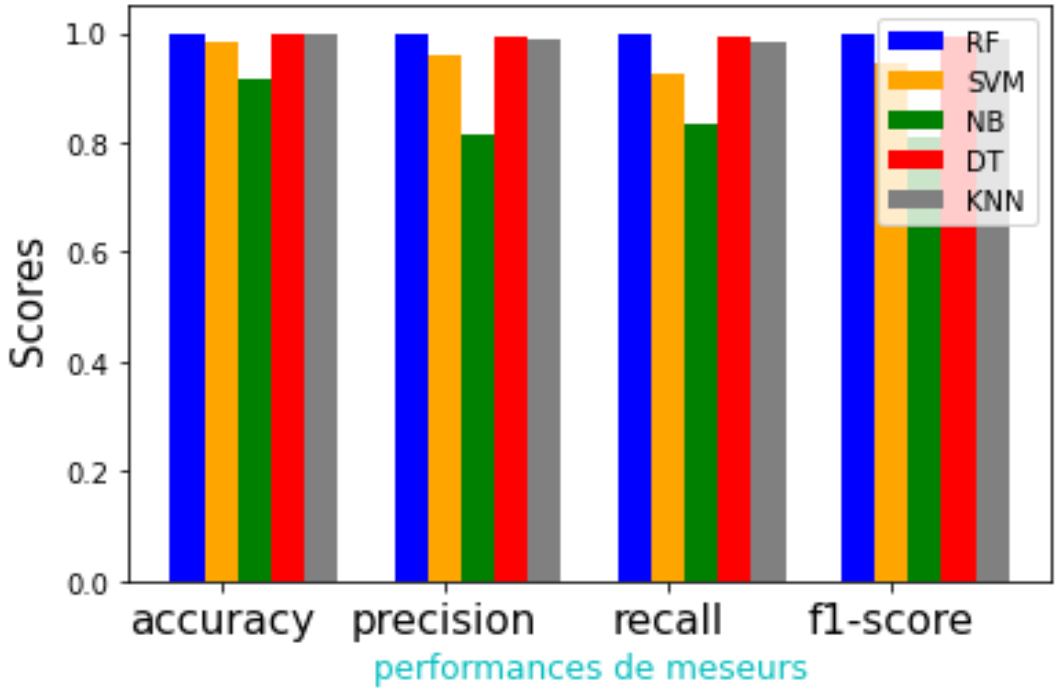


Figure 4.9 : Performances de mesure pour tous les algorithmes de classification, en utilisant la méthode du filtre.

4.2.3 La sélection des attributs manuellement :

Les mesures de performances obtenues sont situées dans le **tableau 4.14** suivant :

Tableau 4.14 : Performances de mesure pour tous les algorithmes de classification, en utilisant la méthode de sélection des attributs manuellement

	Name	Accuracy	précision	Recall	F 1_score
0	KNearest_Neighbors	0.996	0.990	0,985	0.987
1	Svm	0.995	0.951	0.960	0.943
2	Decision_Tree	0.999	0.999	0.999	0.99
3	Random_Forest	0.999999	0.999999	0.999999	0.999999
4	Naive_Bayes	0.995	0.954	0.948	0.951

Le graphe illustré à la figure 4.10 représente les performances de mesures.

Nous observons que tous les algorithmes présentent les meilleures performances avec une Accuracy de 99,99% pour le RF, et 99 % pour (SVM, NB, KNN, DT).

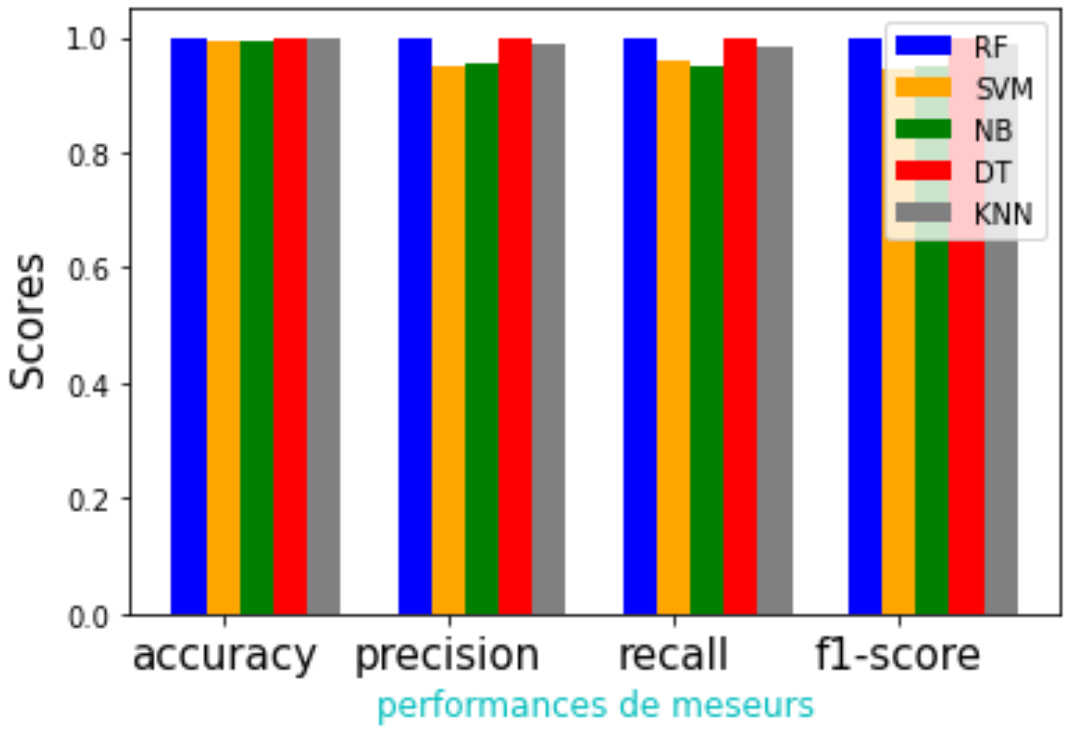


Figure 4.10 : Performances de mesure pour tous les algorithmes de classification, en utilisant la méthode de sélection des attributs manuellement.

4.2.3 Comparaison entre les méthodes :

Dans cette expérience, nous avons fait une comparaison entre différents types de sélection des attributs (méthode de filtre, méthode forward warpper, Backward warpper et la méthode de sélection des attributs manuellement).

Nous obtenons les meilleurs résultats dans la méthode de sélection des attributs manuellement pour tous les algorithmes utilisés et nous obtenons la meilleure performance dans l'algorithme RF avec une accuracy 99,99%.

Pour la méthode de filtre nous obtenons des bonnes performances avec les algorithmes (RF, DT, KNN) avec une accuracy 99%.

Pour les méthodes de warpper nous obtenons des résultats faibles par rapport aux méthode précédents, et pour les méthodes de warpper nous obtenons des résultats faibles par rapport aux méthode précédents, là que d'obtenir une accuracy de 29% pour le naïve bayse.

4.3 Comparaison avec les approches proposées dans la littérature

Dans cette partie on a comparé notre meilleur algorithme de classification par des autres résultats existants dans des autres études, pour cella on a choisie trois étude défèrent.

Notre meilleur algorithme de classification est Random Forest avec une accuracy égale à 0.991,Razan M. AlZoman et Mohammed J. F. Alenazi qui on trouves une accuracy égale à 0.9918 avec l'algorithme « decision tree », le deuxième et Cao etal qui trouvons par l'algorithme de classification « SVM » une accuracy égale à 0.9196 et le dernier est A. W. Moore et D. Zuev qui trouvons une accuracy égale à 0.95 avec l'algorithme « naïve bayse », les résultats de cette comparaison montre dans la « **Figure 4.11** ».

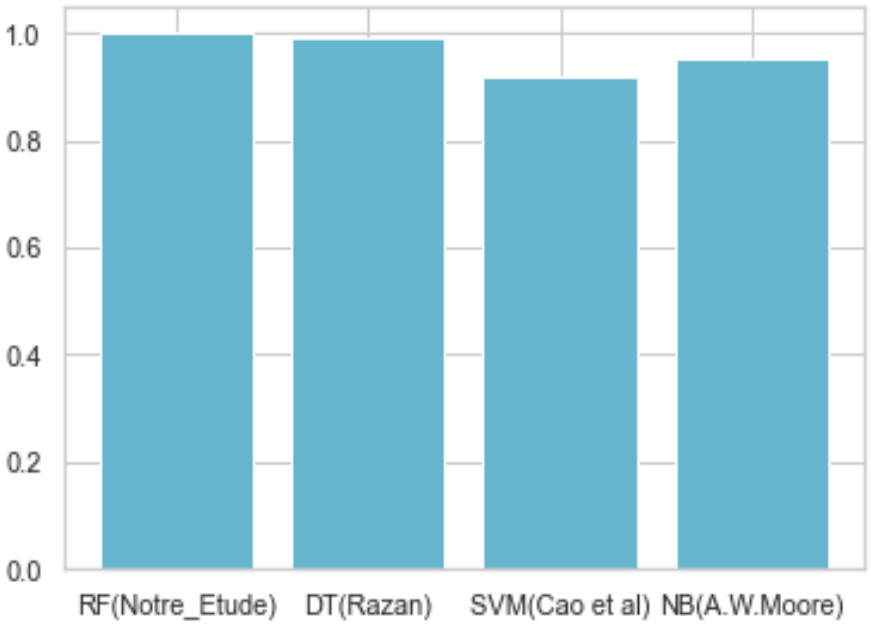


Figure 4.11 : Comparaison avec les approches proposées dans la littérature

5 Conclusion

Dans ce chapitre, nous avons implémenté et testé 5 algorithmes d'apprentissage à savoir : SVM, DT, RF, Naïve Bayse et KNN. De plus, nous avons effectué une comparaison entre ces différents algorithmes. Nous avons conclut que l'algorithme Random Forest présente la meilleur performance avec un taux de précision de l'ordre de 99%

Conclusion et perspectives :

La classification du trafic réseau joue un rôle important et fondamental dans la gestion et la sécurité des réseaux. Un des principaux défis dans ce domaine est de s'adapte à la nature dynamique du trafic internet. Chaque jour de nouvelles applications sont déployées sur internet, ce qui accroit le volume de trafic.

Dans ce travail, nous avons appliqué le concept d'apprentissage automatique pour résoudre ce problème de classification du trafic réseau. En effet, nous avons sélectionné 5 algorithmes à savoir : SVM, Random forest, Naïve Bayse, decision tree, KNN. Puis, nous avons comparé ces différents algorithmes avec le DataSet Moore connu dans la littérature. Nous avons constaté que les algorithmes SVM et Naïve Bayse présentent de mauvaises performances. Les principales raisons sont le fait qu'ils sont plus adaptés aux données catégorielles plutôt qu'aux données numériques.

Enfin, nous avons conclut que l'algorithme Random Forest présente une meilleure performance avec un taux de précision de l'ordre de 99%.

Perspectives :

Des travaux de futur pour enrichir ce travail sont les suivants:

- Nous devons étudier les méthodes de sélection de caractéristiques, telles que les méthodes basées sur les arbres et les méthodes basées sur l'analyse statistique, connaître leurs performances et choisir la meilleure méthode.
- La classification en utilisant l'apprentissage en profondeur « Deep learning » avec un ensemble de donnes très volumineux, et comparée les résultats avec notre propre expérience.
- Développer une plateforme de visualisation qui permet de classification de trafic réseau en temps réel et résoudre la problématique d'exploration des données réseau ou trafics IP pour la surveillance des réseaux.

Références bibliographiques

[1] Boutaba, R., Salahuddin, M., A., Limam, N., Ayou bi, S., Shahriar, N., Estr ada Solano, F., & Caicedo, O., M. (2018). A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. Journal of Internet Services and Applications, 9(1), 1-99.

[2] R,Zhong. «*Dynamic assignment, surveillance and controlfor traffic network with uncertainties*».2011.

[3] Benjamin Billet:<<Système de gestion de flux pour l'Internet des objets intelligents>>. L'université de Versailles Saint-Quentin-En-Yvelines.19 Mars 2015.

[4] laurentbloch. www.laurentbloch.org, Consulter le :28/02/2022.

[5] reseaux-telecoms. www.reseaux-telecoms.net, Consulter le 01/03/2022.

[6] Melle DIFALLAH Chahra :<<Contrôle de la température à travers un réseau Ethernet>>.université ferhat ABBASA de setif.2014.

[7] Siham zouaoui. « Implémentation et validation de la pile TCP/IP de Microchip sur un dsPIC ».2012.

[8] Aouissi. «Apprentissage automatique pour l'analyse des données massives : K-MEANS basé Spark vs K-MEANS classique». Université de annaba. 2020/2021.

[9] Fatima Ait Mahamed. « APPROCHES D'APPRENTISSAGE AUTOMATIQUE POUR LA DÉTECTION DU SPAM WEB : EXPLORATION DE DIVERSES CARACTÉRISTIQUES ». Université Du Québec à Montréal. Mars 2018.

[10] T. Nguyen, and G. Armitage, A Survey of Techniques for Internet Traffic Classification using Machine Learning, IEEE Survey and Tutorials, 10(4), pp. 56-76,2008.

[11] T. Karagiannis, A. Broido, N. Brownlee, K. C. Klaffy, and M. Faloutsos, "Is P2P dying or just hiding?" In IEEE Global Telecommun. Conf. 2004, Dallas, TX, vol. 3, pp. 1532 - 1538, 2004.

[12] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in - network identification of P2P traffic using application signatures," in 13th Int. Conf. World Wide Web 2004, New York, NY, pp. 512 - 521, May 2004.

[13] V. Paxson, "Empirically derived analytic models of wide- area TCP connections," IEEE/ACM Trans. Netw, vol. 2, no. 4, pp. 316– 336, 1994.

[14] C. Dewes, A. Wichmann, and A. Feldmann, “An analysis of Internet chat systems,” in ACM/SIGCOMM Internet Measurement Conf. 2003, Miami, FL, pp. 51 - 64, October 2003.

[15] Foremski, P. On different ways to classi fy Internet Traffic. A short review of selected publications. Theoretical and Applied Informatics, 25(2), 119-136. 2013.

[16] Sirmacek, B. Learning Algorithm Modeling for Mobile Robot with Fpga. Master's thesis, Yıldız Technical University, Istanbul. 2007.

[17] Emmanuel Ferreira : Apprentissage automatique en ligne pour un dialogue homme-machine situé.l’Université d’Avignon et des Pays de Vaucluse. [18] (Erdem, E. S. Emotion Recognition and Retrieval in Audio Signals. Master's thesis, Başkent University, Ankara.2014.

[19] Topal, C. Alan Turing's Sociological Thinking: Artificial Intelligence as a Social Dream. Journal of DTCF, 57(2), 1350-1352. 2017.

[20] Morgane LAUR. « Anticipation des changements de notes des obligations du portefeuille d’un assureur par méthode de machine learning ». Université Paris Dauphine. 2017.

[21] Mlle GUENNINECHE Amel. « Prédiction des propriétés des matériaux par apprentissage automatique ». UNIVERSITE ABOU-BEKR BELKAID – TLEMCEN. 2019.

[22] Zine-el-Abidine Soudani. « ÉTUDE COMPARATIVE DES ALGORITHMES DEDIES A LA CLASSIFICATION ». Université Du Québec. Décembre 2005.

[23] Arab Chaima. « Apprentissage automatique pour la classification du cancer du sein ». UNIVERSITE BADJI MOKHTAR – ANNABA. 2020/2021.

[24] Kızılkaya, Y. M. and Oğuzlar, A. (2018). Comparison of Supervised Learning Algorithms with R Programming Language. Black Sea, 37(37), 90-98

International journal of engineering research and technology, 2013.

[25] Matthieu Zimmer :Apprentissage par renforcement développemental. Department of Complex Systems, Artificial Intelligence & Robotics.

[26] Marref Nadia. <<Apprentissage Incrementale & Machines a vecteurs supports >>. Université HADJLA KHDAR–BATNA. 18/12/2013

[27] H.Mohamadally et B.Fomani, " SVM : Machines à Vecteurs de SupportouSéparateursàVastesMarges"VersaillesStQuentin,France.janvier 2006.

[28] k.ming leung ‘’Naive BAysian Classifier’’ polytechnic university –Depertement of Computer science /finnence and risk engineering.

[29] M. Kantardzic, Data Mining - Concepts, Models, Methods, and Algorithms, IEEE Press, Wiley-Interscience, 2003, ISBN 0-471- 22852-4.

[30] Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, Elsevier 2006, ISBN 1558609016. This part of the lecture notes is derived from chapter 6.4 of this book.].

[31] Robin Genuer. « Forets aléatoires : aspects théoriques, sélection de variables et applications ». Université de Jean-Michel Poggi. 24 Novembre 2010.

[32] Maneela Jain and Pushpendra Singh Tomar. Review of image classification methods and techniques.

[33] Moore, A.,Zuev, D.:Discriminators for usein flow-based classification. Technical report, Intel Research, Cambridge (2005).