RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUEET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Ghardaïa
Faculté des Sciences et de la Technologie
Département des Mathématiques et Informatique

MEMOIRE

Présenté pour l'obtention du diplôme de ***MASTER***

En : Informatique
Spécialité : Systèmes Intelligents pour l'Extraction de Connaissances (SIEC)

Par: BENRAZEK HAKIM

Sujet

CLASSIFICATION DU TRAFIC INTERNET A L'AIDE DE L'APPRENTISSAGE AUTOMATIQUE

BELLAOUAR SLIMANE	***M.C.***	***UNIVERSITE DE GHARDAIA***	***Président***
ADJILA ABDERRAHMANE	***M.A.***	***UNIVERSITE DE GHARDAIA***	***Examinateur***
BOUHANI ABDELKADER	***M.A.***	***UNIVERSITE DE GHARDAIA***	***Examinateur***
KERRACHE CHAKER ABDELAZIZ	***M.C.***	***UNIVERSITE DE GHARDAIA***	***Encadrant***

Année Universitaire 2019/2020

Dédicace

Je dédie ce modeste travail à :

Mes parents et mes grands-parents vos prières m'ont été d'un grand secoure pour mener a bien mes études , Aucune dédicace ne saurait être assez éloquente pour exprimer ce que vous méritez pour tous les sacrifices que vous n'avez cessés de me donner

Mes chères sœurs, mon frère , mes tantes et mes oncles.

En témoignage de l'attachement ,de l'amour et de l'affection que je porte pour vous, je vous dédie ce travail avec tous mes vœux de bonheur, de santé et de réussite A tous les autres membres de ma famille sans exception.

A mes amis et mes collègues dans et en dehors l'université.

BENRAZEK HAKIM

Remerciements

En préambule à ce mémoire, la grande louange à **Dieu** qui nous aide et nous donne la bonne santé, la patience et le courage durant l'élaboration de ce modeste travail.

Nous tenons à remercier tous ceux qui nous ont aidés, d'une manière ou d'une autre, pendant ce travail d'étude et de recherche.

Nous tenons d'abord à remercier très chaleureusement Monsieur **kerrache Chaker Abdelaziz** qui nous a permis de bénéficier de son encadrement. Les conseils qu'il nous a prodigué, la patience, la confiance qu'il nous a témoignés ont été déterminants dans la réalisation de notre travail de recherche.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre modeste travail et de l'enrichir par leurs propositions.

Nos remerciements s'étendent également à tous nos enseignants durant les années des études.

Merci à Tous.

الملخص

تصنيف حركة الإنترنت (CTI) تشمل عدة طرق التي تساهم في معالجة تصنيف حركة المرور وفقا للخصائص التي يتم ملاحظتها بشكل سلبي في حركة المرور نفسها. كما تحظى تقنيات (CTI)باهتمام كبير في مجال الإدارة والأمن المعلوماتي وتخطيط الشبكات، كما يوفر جودة عالية في تقديم الخدمات. وبمرور الزمن، أصبحت الأساليب التقليدية مثل التعريف القائم على الموانئ والتعريف القائم على أساس الحمولة أكثر صعوبة عند التنفيذ مع التطبيقات. لذلك تم استخدام تقنيات بديلة تعتمد على التعلم الآلي الذي يتيح استغلال إحصائية التدفق.

عملنا هذا يتضمن تصنيف حركة المرور على الإنترنت بواسطة التعلم الآلي، الذي استعملنا فيه خوازمية SVM . على الرغم من أن تصنيف حركة المرور بواسطة تطبيق SVM عرف تطورا ملحوظا، إلا أنه لا يزال هناك العديد من النقائص من أحل تحسين استخدام SVM في التعلم الآلي.

الكلمات المفتاحية: الإنترنت، تصنيف حركة المرور على الإنترنت، التعلم الآلي.

Resumé

La classification du trafic internet (CTI) comprend des méthodes qui traitent de la classification du trafic en fonction des caractéristiques qui sont négativement observées dans le trafic lui-même et éventuellement en fonction de certains objectifs de classification spécifiques. Les technologies (CTI) suscitent beaucoup d'attention dans les domaines de la gestion de réseau, de la sécurité et de la planification, ainsi que dans la fourniture de qualité de service.
Avec le temps, les méthodes traditionnelles telles que l'identification basée sur le port et l'identification basée sur la charge sont devenues plus difficiles lors de la mise en œuvre avec les applications. Par conséquent, les techniques alternatives sont basées sur l'apprentissage automatique, qui permettent d'exploiter les statistiques de flux.
Notre travail comprend la classification du trafic internet par l' apprentissage automatique, dans lequel nous avons utilisé l'algorithme SVM. Bien que la classification du trafic par l'application SVM ait connu un développement remarquable, mais ils existent encore plusieurs lacunes, afin d'améliorer l'utilisation de SVM dans l'apprentissage automatique.
Mots clés : Internet, Classification du trafic internet, Apprentissage automatique.

Abstract

Internet traffic classification (CTI) includes methods that deal with classifying traffic according to the characteristics that are negatively observed in the traffic itself and possibly according to some specific classification objectives. CTI technologies are of great interest in the areas of network management, security and planning, as well as in providing quality of service.
Over time, traditional methods such as port-based identification and load-based identification have become more difficult to implement in many applications. Therefore, alternative techniques based on machine learning have been used that make it possible to exploit the characteristics of flow statistics.
Our job is to categorize internet traffic using machine learning, where data flow in the internet is classified, and for this we use the SVM algorithm. Although some work on applying SVM to traffic classification has developed, many points remain open on how to improve its use in a machine learning problem.
Keywords : Internet, Internet traffic classification, machine learning.

Table des matières

Liste des tableaux

Table des figures

Introduction Générale

En raison de l'énorme développement qui a eu lieu sur internet au cours de la dernière décennie, l'utilisation d'internet est devenue indispensable pour tout le monde, le commerce électronique et les comptes bancaires sont les choses les plus sensibles, et l'énorme croissance des utilisateurs et des données sur internet pose un défi pour connaître le flux et le mouvement de l'information. Ainsi, le trafic internet est quelque chose dont il ne faut pas se passer. Dans le passé, il y a eu une évolution du nombre de flux de données. En parallèle, il y a eu une évolution des procédures et des outils de connaissance du flux et de la classification de ces données. De nombreuses recherches ont été effectuées dans ce domaine et diverses techniques ont été suggérées. Les plus connus sont les statistiques et l'exploration de données.
Le mouvement internet est devenu l'un des domaines les plus attrayants et les plus populaires pour les universitaires dans de nombreux domaines tels que : l'économie, l'industrie, etc., car il est en concurrence avec d'autres disciplines connexes. Ce qui a conduit à la classification de ce mouvement. Nous utilisons le terme classification du trafic pour décrire les moyens de classer le trafic en fonction des caractéristiques qui sont négativement observées dans le trafic et selon des objectifs de classification spécifiques. On pourrait n'avoir qu'un objectif de classification approximatif, c'est-à-dire s'il est orienté transaction, transfert en masse ou partage de fichiers d'égal à égal. Ou bien, on peut avoir un objectif de classification plus précis, c'est-à-dire l'application exacte que représente le trafic. Les fonctionnalités de trafic peuvent inclure le numéro de port, la charge utile de l'application, l'heure, la taille des paquets et les caractéristiques d'adressage du trafic. Les méthodes de classification incluent la correspondance exacte, par exemple, du numéro de port ou de charge, la détection heuristique ou l'apprentissage automatique (statistiques).
Cela nous amène à nous demander : comment le trafic internet est-il classé ?
Plus précisément : comment classer le trafic internet à l'aide de l'apprentissage

automatique ?
Dans notre mémoire, nous traiterons de la classification du trafic internet à l'aide d'apprentissage automatique parce que certains algorithmes d'apprentissage automatique conviennent pour classifier le flux de trafic internet à haut débit.
Le but de ce mémoire est, la classification du trafic internet à l'aide d'apprentissage automatique. Nous utilisons l'algorithme SVM pour cette opération.
Ce travail est organisé en trois chapitres.
• Chapitre 1 : Il s'agit d'un aperçu de l'apprentissage automatique, où il a été défini, puis abordé certaines de ses applications et son lieu d'utilisation en informatique, et à la fin nous avons mentionné ses différents types.
• Chapitre 2 : dédié aux techniques de classification du trafic internet utilisant l'apprentissage automatique. Il présente le trafic internet, sa classification et ses technologies inhérentes.
• Chapitre 3 : se concentre sur la réalisation de notre application, et inclut la conception architecturale en plus de l'environnement de développement, avec une discussion des résultats obtenus.

Chapitre 1

APPRENTISSAGE AUTOMATIQUE

Ce chapitre est consacré aux définitions et les concepts essentiels de l'apprentissage automatique.

1.1 Introduction

L'apprentissage automatique, comme sous-domaine de l'intelligence artificielle, permet une construction automatique des connaissances et de nos jours, il est particulièrement attrayant. Dans ce chapitre, cette discipline avec ses apports multiples est présentée comme une alternative à l'approche classique, pour le développement de systèmes informatiques complexes.

1.2 Définition

Dans cette section, nous abordons la définition d'apprentissage automatique et quelques algorithmes qui sont nécessaires à la compréhension de la suite du document.

Apprentissage Automatique

Traduction du terme en anglais (Machine learning), pour plusieurs équipes de recherche, l'intelligence artificielle est fortement liée à l'apprentissage automatique. Les chercheurs de l'université Carnegie Millon, tels que : Jaime Carbonell, Yves Kodratoff, Ryszard Michalski et Tom Mitchell, voient que pour définir ce que représente pour une machine le fait d'être intelligente, l'apprentissage est une question fondamentale. En effet, d'après [27] : «Pour pouvoir être considérée comme intelligente, une machine doit posséder les caractéristiques d'apprentissage et de créativité». Cette idée est pleinement soutenue par le fait que nous ne pouvons pas rendre une machine intelligente en y accumulant une masse de connaissances, mais plutôt qu'elle doit être dotée de capacités d'apprentissage à partir des événements observés. En plus de bénéficier de son expérience pour mieux se préparer à une future réponse à des événements similaires. Aussi, pour qu'une machine soit intelligente, elle doit avoir la capacité d'affiner et de créer de nouvelles connaissances, afin de s'adapter à de nouvelles situations.
Ainsi pour avoir un système informatique intelligent, il doit posséder la capacité d'apprentissage automatique lui permettant de s'auto-améliorer.

L'apprentissage automatique combine tellement de concepts différents et variés qu'il peut être difficile de fournir une définition unique. En effet, l'apprentissage automatique est multidisciplinaire et comprend des concepts et des techniques de plusieurs domaines dans lesquels s'inspirer et des outils.
L'apprentissage automatique représente un champ de recherche et d'application des plus foisonnants,, il se situe au carrefour de plusieurs disciplines incluant : l'intelligence artificielle, l'analyse des données et les statistiques, la philosophie, la psychologie, la théorie de l'information, la biologie, les sciences cognitives, la théorie de la complexité...
Alors pour ce domaine multidisciplinaire, nous présentons les définitions suivantes :

Définition 1

Du point de vue des machines, on dira qu'une machine apprend dès lors qu'elle change sa structure, son programme ou ses données en fonction de données en entrée ou de réponses à son environnement de sorte à ce que ses performances futures deviennent meilleures [34].

Définition 2

L'apprentissage dénote des changements dans un système qui lui permettent de faire la même tâche plus efficacement la prochaine fois [48].

Définition 3

L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentassions de méthodes qui permettent à une machine(au sens large) d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques [27].

Définition 4

Un programme informatique apprend à partir de l'expérience E par rapport à une classe de tâches T et une mesure de performance P, si sa performance à l'exécution de tâches de T, mesurée par P, s'améliore avec l'expérience E [36].
Donc, l'apprentissage automatique pour la machine est qu'avec l'ensemble de taches T que la machine doit réaliser, elle utilise l'ensemble d'expériences E telle que sa performance P est améliorée.
En d'autres termes, nous concevons des systèmes peuvent apprendre et se développer avec l'expérience et au fil du temps pour identifier un modèle pouvant être utilisé pour prédire les résultats de questions basées sur un apprentissage précé-

dent.

1.3 Applications d'apprentissage automatique

L'apprentissage automatique a été largement utilisé à l'heure actuelle, car en 1985, il n'existait pratiquement aucune application utilisé l'apprentissage automatique.
Au fil du temps, il a réalisé des progrès remarquables et de grands avantages en raison de ses applications d'une importance réaliste.
• **Reconnaissance vocale** : les systèmes commerciaux actuellement disponibles utilisent l'automatisation de la reconnaissance vocale. Il s'agit de la plus grande précision de reconnaissance du son lors d'une tentative programmée manuellement. Les systèmes de reconnaissance sonore commerciaux comportent deux étapes d'apprentissage distinctes : une avant l'installation du programme et la seconde après. L'utilisateur a acheté le programme pour plus de précision en utilisant la formation [26].
• **Vision par ordinateur** : de nombreux systèmes de vision ont été développés après l'apprentissage automatique, allant des systèmes de reconnaissance faciale aux systèmes de classification automatique des images cellulaires, car les systèmes résultants sont plus précis que les programmes portables [25].
• **Biomarquage** : les efforts du gouvernement utilisent l'apprentissage automatique pour détecter et suivre diverses épidémies, ce qui s'applique à l'utilisation de méthodes d'apprentissage automatiques pour l'achat de médicaments en vente libre [33].
• **Contrôle par robot** : des méthodes d'apprentissage automatiques ont été utilisées avec succès dans un certain nombre de systèmes Android, à l'instar de son utilisation dans les stratégies de contrôle pour un vol en hélicoptère et une aérobic stables [41].
• **Accélération des sciences expérimentales** : de nombreuses sciences basées sur les données ont tiré parti des méthodes d'apprentissage automatiques au cours de leurs découvertes scientifiques, qui ont transformé la pratique de nombreuses sciences expérimentales qui consomment beaucoup de données. Ateliers sur l'apprentissage automatique [36].

1.4 Rôle d'apprentissage automatique dans certaines applications

En examinant l'exemple des applications mentionnées ci-dessus, nous nous sommes interrogés sur le rôle futur de l'apprentissage automatique dans divers domaines d'application généraux et d'applications informatiques en particulier ? Les méthodes d'apprentissage automatiques constituent actuellement le meilleur moyen de développer certains types de programmes, notamment dans les applications qui :

• L'application est trop complexe pour permettre aux utilisateurs de concevoir manuellement l'algorithme. Ceci s'applique aux exemples précédents tels que la reconnaissance vocale et la vision par ordinateur. Chacun de nous peut facilement identifier les photographies contenant une image de sa mère, mais aucun d'entre nous ne peut écrire un algorithme pour effectuer cette tâche [36].

• L'application nécessite que le programme s'adapte à son propre environnement d'exploitation après utilisation. Par exemple, des systèmes de reconnaissance vocale sont attribués à l'utilisateur qui achète le logiciel. Dans ce cas, l'apprentissage automatique fournit un mécanisme d'adaptation qui a permis à la place de l'apprentissage automatique de croître rapidement dans le monde du logiciel [36].

À cet égard, les méthodes d'apprentissage automatiques jouent un rôle majeur dans le monde informatique, où les ordinateurs peuvent accéder aux données, et nous développons de plus en plus des algorithmes d'apprentissage automatique. Bien qu'il existe des applications logicielles pour lesquelles l'apprentissage automatique peut ne pas être aussi utile que l'écriture de programmes de multiplication matricielle, l'apprentissage automatique met l'accent sur la conception de systèmes d'autosurveillance et sur l'avantage d'un flux continu de données.

De même, l'apprentissage automatique permet de redéfinir le domaine de la statistique et de soulever des problèmes comme un apprentissage sans fin. L'informatique et les statistiques contribueront bien sûr à façonner «l'apprentissage automatique» et à introduire de nouvelles idées pour changer notre façon d'offrir l'éducation [36].

1.5 Types d'apprentissage automatique

Dans cette section, nous abordons les types d'apprentissage automatique et comprenons comment il apparaît dans les applications que nous utilisons.
L'apprentissage automatique est généralement divisé en quatre types, comme illustré par la Figure 1.1.

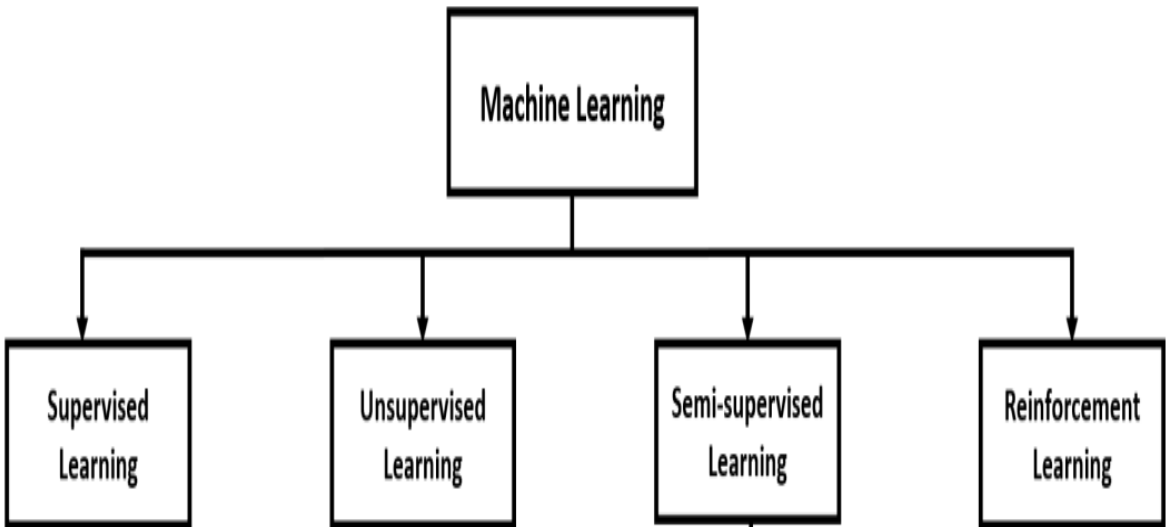

FIGURE 1.1 – Classes de l'apprentissage automatique [23].

1.5.1 Apprentissage supervisé

L'apprentissage supervisé est effectué lorsque des objectifs spécifiques sont définis pour atteindre un certain ensemble d'entrées. Pour ce type d'apprentissage, les données sont d'abord classées, suivies d'un entraînement avec des données structurées (qui ont des entrées et des sorties souhaitées). Il essaie d'identifier automatiquement les règles à partir des ensembles de données disponibles et de définir les différentes classes, et enfin de prédire l'appartenance des éléments à une classe donnée [23].

Les problèmes d'apprentissage supervisé sont classés en deux problèmes : la régression et la classification, comme montrer par la Figure 1.2.

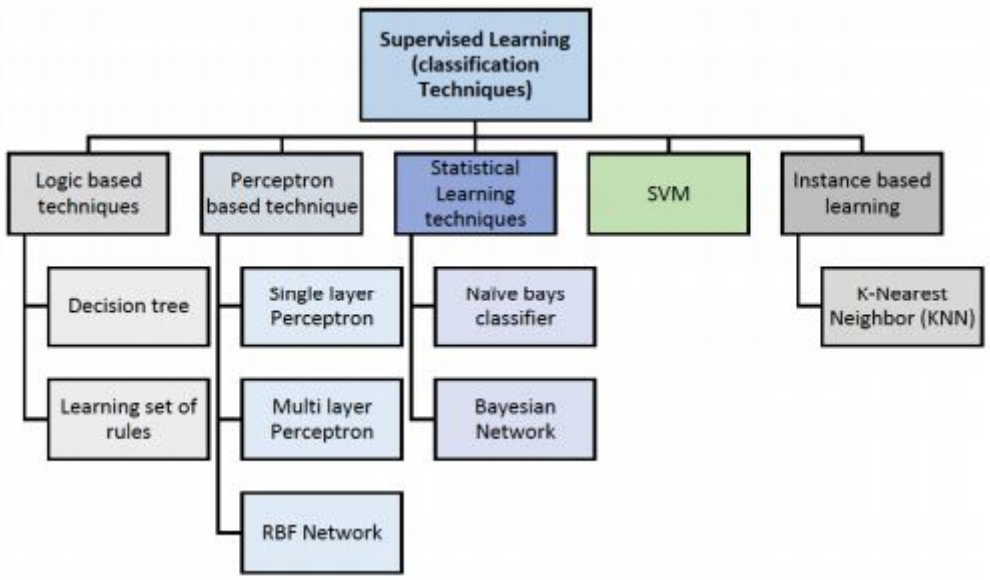

FIGURE 1.2 – apprentissage supervisée [35]

La régression

Le problème de la régression se pose lorsque la variable de sortie est réelle ou continue : par exemple, compte tenu des données relatives à la taille des maisons sur le marché immobilier, nous essayons de prédire son prix.

La classification

Le problème de classification survient lorsque la variable de sortie est une classe. Tels que : oui ou non, 0 ou 1, vrai ou faux. Par exemple, lorsque vous filtrez les messages "spam" ou "not spam".

Quelques techniques d'apprentissage supervisé

Il existe de nombreux algorithmes d'apprentissage supervisé tels que, réseaux de neurones, machines à vecteurs de support (SVM) , Naive Bayes ,la régression logistique,Linear Regression, l'arbre de décision et la forêt aléatoire [40].

-SVM (machines à vecteurs de support)

SVM est une méthode de classification binaire par apprentissage supervisé, elle fut introduite par Vapnik en 1995. Cette méthode est donc une alternative récente

pour la classification.
Elle repose sur l'existence d'un classificateur linéaire dans un espace approprié. Puisque c'est un problème de classification à deux classes, cette méthode fait appel à un jeu de données d'apprentissage pour apprendre les paramètres du modèle. Elle est basée sur l'utilisation de fonctions dites noyau (kernel) qui permettent une séparation optimale des données. Dans la présentation des principes de fonctionnements, nous schématiserons les données par des « points » dans un plan.
L'algorithme des machines à vecteurs de support a été développé dans les années 90 par le russe Vladimir Vapnik. Initialement, les SVM ont été développé comme un algorithme de classification binaire supervisée. Il s'avère particulièrement efficace de par le fait qu'il peut traiter des problèmes mettant en jeu de grands nombres de descripteurs, qu'il assure une solution unique (pas de problèmes de minimum local comme pour les réseaux de neurones) et il a fourni de bons résultats sur des problèmes réels [32].
L'algorithme sous sa forme initiale revient à chercher une frontière de décision linéaire entre deux classes, mais ce modèle peut considérablement être enrichi en se projetant dans un autre espace permettant d'augmenter la séparabilité des données. On peut alors appliquer le même algorithme dans ce nouvel espace, ce qui se traduit par une frontière de décision non linéaire dans l'espace initial [32].
-SVM principe de fonctionnement général
Notions de base : Hyperplan, marge et support vecteur Pour deux classes d'exemples donnés, le but de SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est un classificateur linéaire appelé hyperplan.
Dans le schéma qui suit (Figure 1.3), on détermine un hyperplan qui sépare les deux ensembles de points [21].

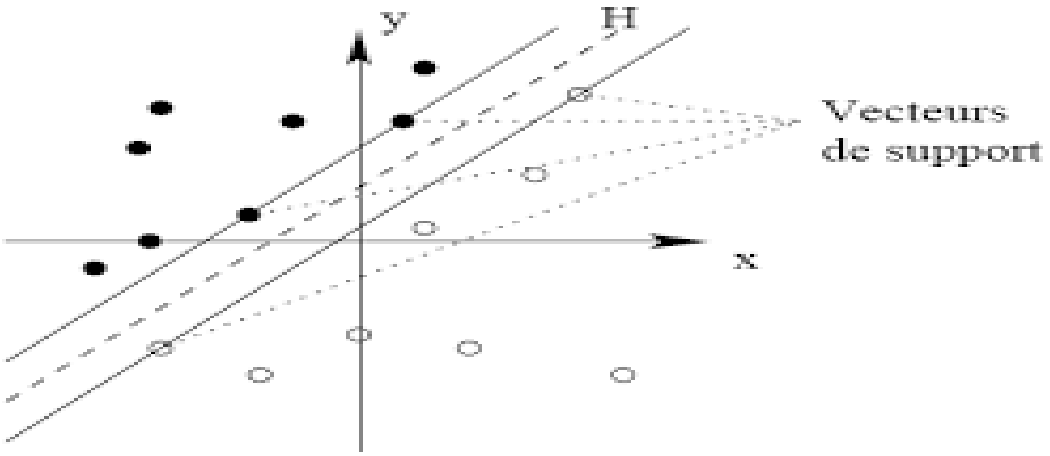

FIGURE 1.3 – Exemple de vecteurs de support [21]

.

Il est évident qu'il existe une multitude d'hyperplan valide mais la propriété remarquable des SVM est que cet hyperplan doit être optimal. Nous allons donc en plus chercher parmi les hyperplans valides, celui qui passe « au milieu » des points des deux classes d'exemples. Intuitivement, cela revient à chercher l'hyperplan le « plus sûr » [11].
En effet, supposons qu'un exemple n'ait pas été décrit parfaitement, une petite variation ne modifiera pas sa classification si sa distance à l'hyperplan est grande [11].
Formellement, cela revient à chercher un hyperplan dont la distance minimale aux exemples d'apprentissage est maximale On appelle cette distance « marge » entre l'hyperplan et les exemples. L'hyperplan séparateur optimal est celui qui maximise la marge (Figure 1.4). Comme on cherche à maximiser cette marge, on parlera de séparateurs à vaste marge [11].

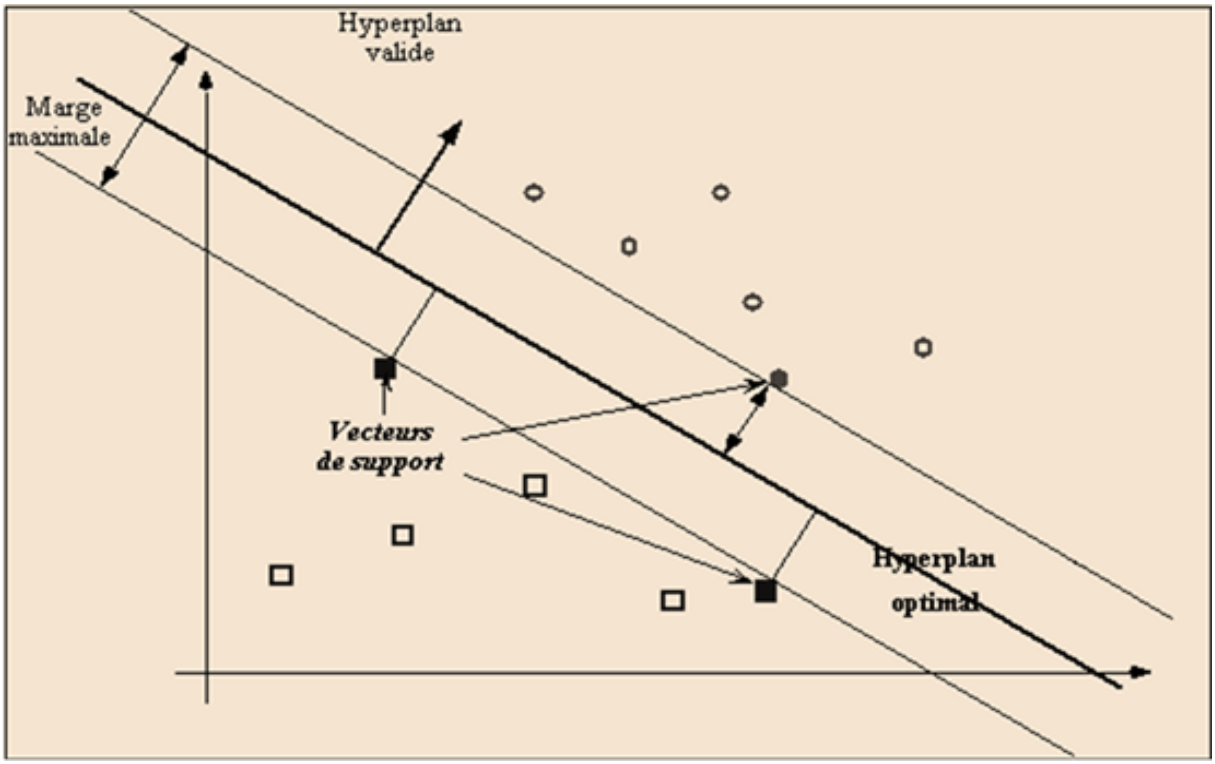

FIGURE 1.4 – Exemple de marge maximal (hyperplan valide) [10]

.

Pourquoi maximiser la marge ?

Intuitivement, le fait d'avoir une marge plus large procure plus de sécurité lorsque l'on classe un nouvel exemple. De plus, si l'on trouve le classificateur qui se comporte le mieux vis-à-vis des données d'apprentissage, il est clair qu'il sera aussi celui qui permettra au mieux de classer les nouveaux exemples. Dans le schéma qui suit,la première figure (Figure 1.5) nous montre qu'avec un hyperplan optimal, un nouvel exemple reste bien classé alors qu'il tombe dans la marge. On constate sur la deuxième figure (Figure 1.6) qu'avec une plus petite marge, l'exemple se voit mal classé [21]

.

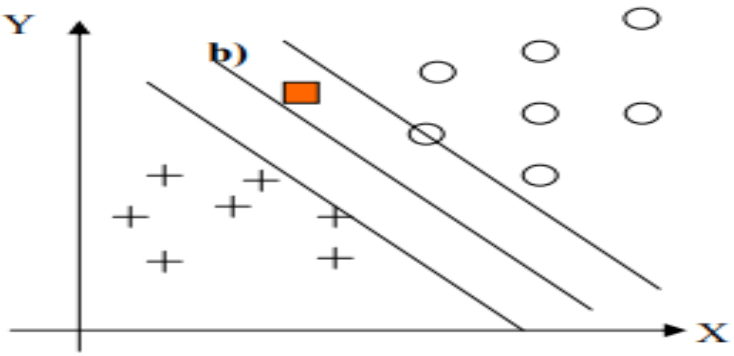

FIGURE 1.5 – Meilleur hyperplan séparateur [21].

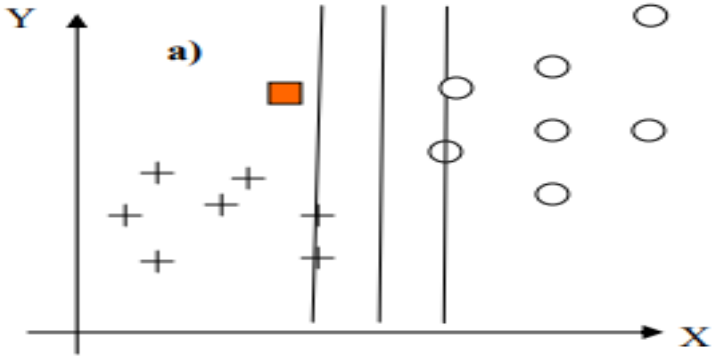

FIGURE 1.6 – Hyperplan avec faible marge [21].

En général, la classification d'un nouvel exemple inconnu est donnée par sa position par rapport à l'hyperplan optimal. Dans le schéma suivant (Figure 1.7), le nouvel élément sera classé dans la catégorie des « + ».

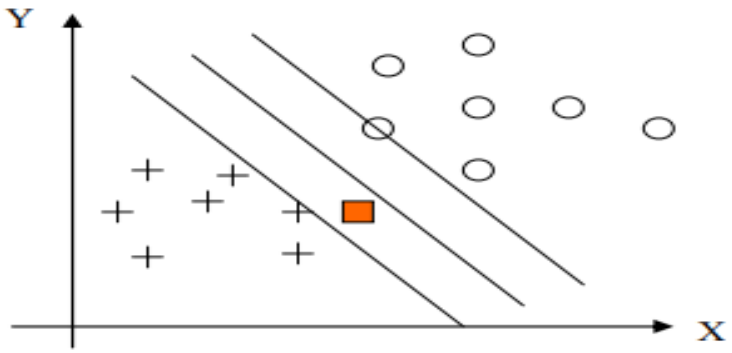

FIGURE 1.7 – Exemple de classification d'un nouvel élément.

Linéarité et non-linéarité

Parmi les modèles des SVM, on constate les cas linéairement séparable (Figure 1.8) et les cas non linéairement séparable (Figure 1.9). Les premiers sont les plus simples de SVM car ils permettent de trouver facilement le classificateur linéaire. Dans la plupart des problèmes réels il n'y a pas de séparation linéaire possible entre les données, le classificateur de marge maximale ne peut pas être utilisé car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables [21].

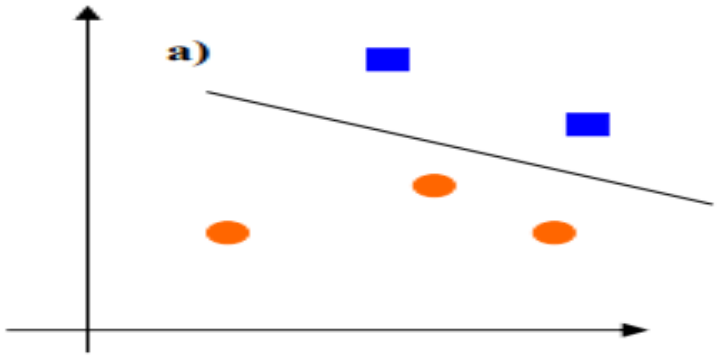

FIGURE 1.8 – Cas linéairement séparable [21].

.

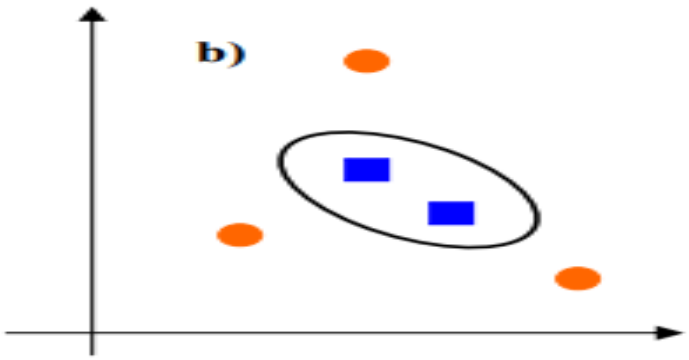

FIGURE 1.9 – Cas non linéairement séparable [21].

.

Cas non linéaire

Pour surmonter les inconvénients des cas non linéairement séparable, l'idée des SVM est de changer l'espace des données. La transformation non linéaire des données peut permettre une séparation linéaire des exemples dans un nouvel espace. On va donc avoir un changement de dimension. Cette nouvelle dimension est appelé « espace de re-description ».En effet, intuitivement, plus la dimension de l'espace de redescription est grande, plus la probabilité de pouvoir trouver un hyperplan séparateur entre les exemples est élevée [21]. Ceci est illustré par le schéma suivant (Figure 1.10) :

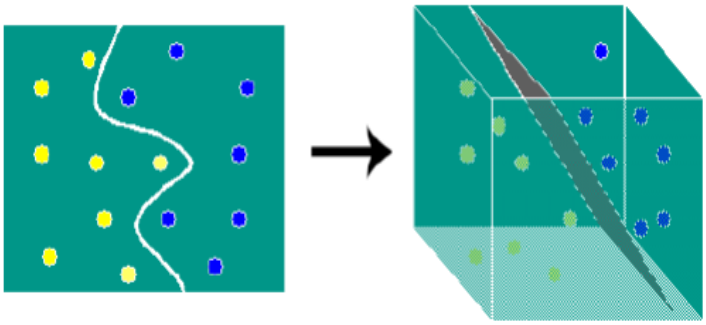

FIGURE 1.10 – Exemple de changement de l'espace de données [21].

On a donc une transformation d'un problème de séparation non linéaire dans l'espace de représentation en un problème de séparation linéaire dans un espace de redescription de plus grande dimension.Cette transformation non linéaire est

réalisée via une fonction noyau [21].
En pratique, quelques familles de fonctions noyau paramétrables sont connues et il revient à l'utilisateur de SVM d'effectuer des tests pour déterminer celle qui convient le mieux pour son application. On peut citer les exemples de noyaux suivants : polynomiale, gaussien, sigmoïde et laplacien [21].

Illustration de transformation de cas non linéaire : le cas XOR

Le cas de XOR n'est pas linéairement séparable, si on place les points dans un plan à deux dimensions, on obtient la figure suivante (Figure 1.11) :
Coordonnées des points : (0,0) ; (0,1) ; (1,0) ; (1,1)

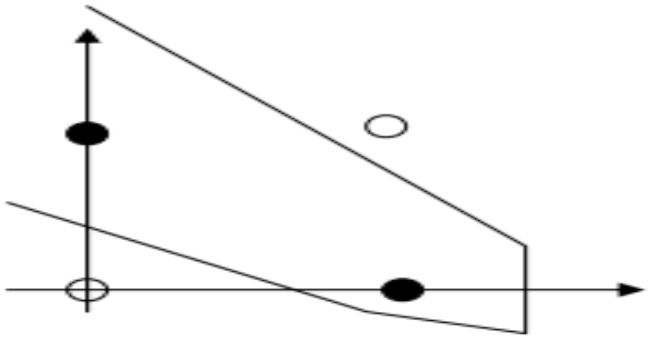

FIGURE 1.11 – Illustration de cas non linéairement séparable (le cas XOR) [21].

Si on prend une fonction polynomiale (x,y) -> (x,y,x.y) qui fait passer d'un espace de dimension 2 à un espace de dimension 3, on obtient un problème en trois dimensions linéairement séparable comme la Figure 1.12 :
(0,0) -> (0,0,0)
(0,1) -> (0,1,0)
(1,0) -> (1,0,0)
(1,1) -> (1,1,1)

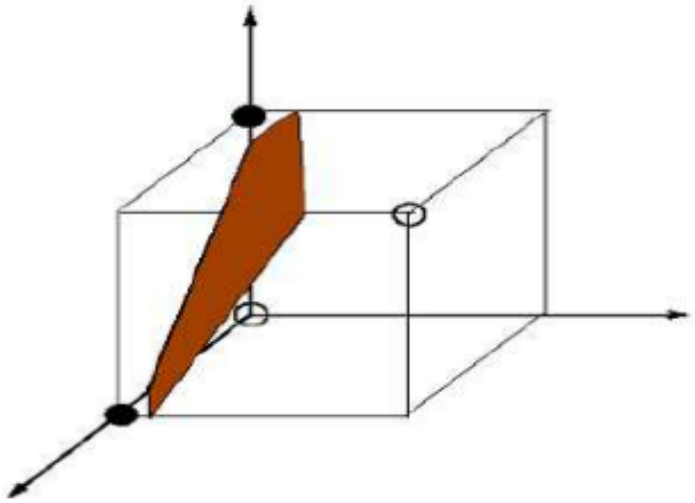

FIGURE 1.12 – Illustration de passage d'un espace 2D à un espace 3D [21].

Les domaines d'applications

SVM est une méthode de classification qui montre de bonnes performances dans la résolution de problèmes variés. Cette méthode a montré son efficacité dans de nombreux domaines d'applications tels que le traitement d'image, la catégorisation de textes ou le diagnostiques médicales et ce même sur des ensembles de données de très grandes dimensions.

La réalisation d'un programme d'apprentissage par SVM se ramène à résoudre un problème d'optimisation impliquant un système de résolution dans un espace de dimension conséquente. L'utilisation de ces programmes revient surtout à sélectionner une bonne famille de fonctions noyau et à régler les paramètres de ces fonctions. Ces choix sont le plus souvent faits par une technique de validation croisée, dans laquelle on estime la performance du système en la mesurant sur des exemples n'ayant pas été utilisés en cours d'apprentissage.

L'idée est de chercher les paramètres permettant d'obtenir la performance maximale. Si la mise en oeuvre d'un algorithme de SVM est en général peu coûteuse en temps, il faut cependant compter que la recherche des meilleurs paramètres peut requérir des phases de test assez longues [21].

- Les avantages et les inconvénients

Les avantages

-SVM est particulièrement efficace pour traiter des données de grande dimension

car la complexité de la forme ne dépend pas de l'espace de dimension de l'entité [37].
-La complexité du classeur peut être contrôlée par le paramètre C en rendant la marge plus lisse ou plus sinueuse au besoin [37].
-Il est possible de faire circuler le classeur même sans afficher une seule donnée de test.
De plus, l'utilisation de SVM linéaire est particulièrement utile pour trier les documents texte, car la plupart des problèmes de classification de texte sont séparables linéairement.
les inconvénients :
-Dans une situation où de nombreux termes ne sont pas importants pour la discrimination de classe, SVM peut être rencontré [37].
-SVM n'est pas utilisé pour calculer la probabilité d'appartenir à une classe .
-L'utilisation de fonctions de base ne vous permet pas de choisir des attributs importants pour la classification .

-**Décision Tree** :
est la plus populaire dans la classification et la prédiction. Chaque nœud interne fait référence à un test sur un attribut, chaque branche représente un résultat du test et chaque nœud est une feuille (nœud) avec un libellé de classe [40]. comme illustrer par la Figure 1.13 .

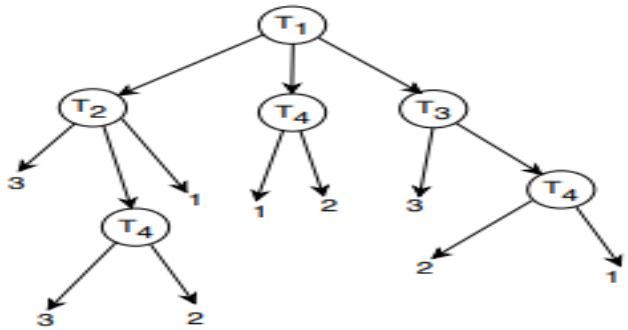

FIGURE 1.13 – Arbre de decision [40].

- Comment construire un arbre de décision ?
"Chaque arbre répond "oui" ou "non" (décision prise). Chaque nœud de l'arbre

de décision contient un test (IF ... THEN) et les feuilles contiennent Oui ou Non. Chaque test examine la valeur de l'attribut pour chaque exemple, qui est un ensemble d'attributs valeurs" [38].
"Pour construire un arbre de décision, il est nécessaire de trouver l'attribut à tester dans chaque nœud, processus répétitif. Pour déterminer l'attribut à tester à chaque étape, un calcul statistique est utilisé pour déterminer dans quelle mesure cet attribut est séparé des exemples de type oui / non. Nous créons ensuite un nœud contenant ce test et créons autant de valeurs possibles que possible pour ce test" [38].
Les arbres de décision peuvent varier selon les dimensions : les tests multivariés ou monochromatiques peuvent avoir deux résultats ou plus, les caractéristiques peuvent être concluantes ou numériques et nous pouvons avoir deux classes ou plus de deux chapitres. Si nous avons deux catégories et des entrées binaires, ils remplissent une fonction logique, appelée arbre de décision logique.
Les algorithmes les plus connus dans l'arbre de décision sont ID3 (Itérative Dichotomiser 3) développé en 1986 par Ross Quinlan et C4.5 une extension de ID3 par Ross Quinlan aussi [43].

-**Naive Bayes** :
est une Type de techniques d'apprentissage supervisée.Principalement ciblé sur l'industrie de la classification de texte, Ils sont utilisés à des fins de compilation et de classification. Sa structure dépend de la probabilité conditionnelle [31].

1.5.2 Apprentissage Non Supervisé

Contrairement à l'apprentissage supervisé, l'apprentissage non supervisé, également appelé agrégation ou apprentissage en groupe, consiste à classifier un échantillon identifié dans des groupes avec des données semi-similaires ou des groupes eux-mêmes [30][44]. La Figure 1.14 montre l'apprentissage non-supervisés

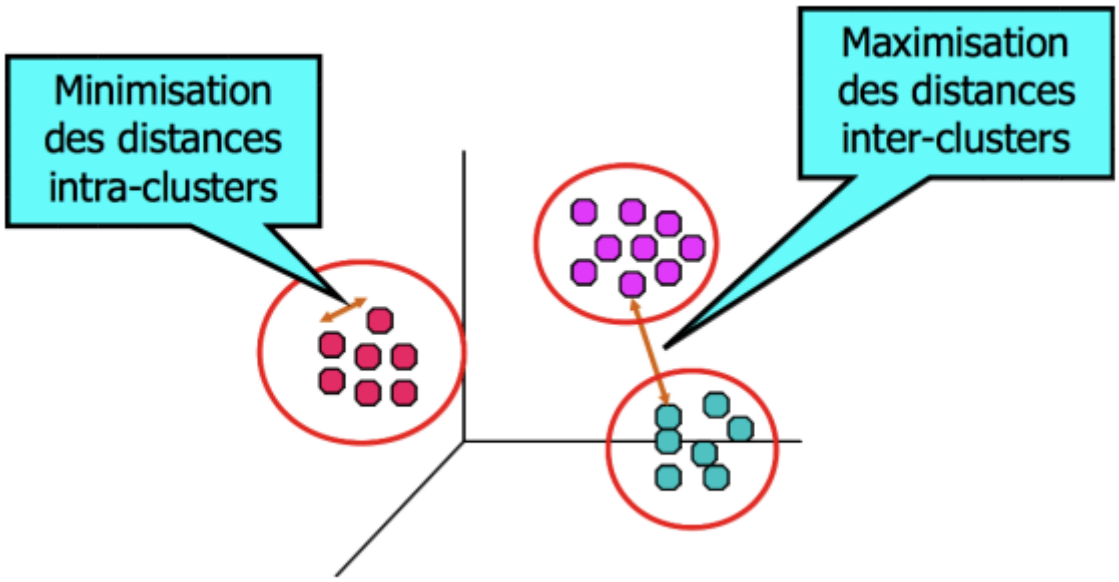

FIGURE 1.14 – Apprentissage non-supervisés [14].

- l'apprentissage non-supervisée sont classé deux problèmes :

Clustering

Il s'agit d'une exploration des données utilisées pour la segmenter en groupes significatifs (c'est-à-dire en grappes) sur la base de leurs modèles internes sans connaissance préalable des informations d'identification du groupe. Les informations d'identification sont définies par la similarité d'objets de données individuels et par des aspects de leur dissimilarité par rapport au reste (qui peuvent également être utilisés pour détecter des anomalies) [15].

Réduction de la dimensionnalité

"Ce n'est pas l'augmentation quotidienne, mais la diminution quotidienne. Éliminez ce qui n'est pas essentiel. »- Bruce Lee
La réduction s'agit d'essayer de réduire la complexité des données tout en conservant autant que possible la structure pertinente. Si vous prenez une image simple de 128 x 128 x 3 pixels (longueur x largeur x valeur RVB), cela correspond à 49 152 dimensions de données. Si vous êtes en mesure de réduire la dimensionnalité de l'espace dans lequel vivent ces images sans détruire trop de contenu significatif

dans les images, alors vous avez fait du bon travail en matière de réduction de dimensionnalité.

Quelques techniques d'apprentissages non supervisé

Il y a plusieurs algorithmes dans l'apprentissage non supervisée, nous mentionnons les plus connues :

-k-means clustering

est un type de méthodes d'apprentissage non supervisée qui créent automatiquement des groupes au début(Figure 1.15), dans lesquels les éléments ayant des propriétés similaires sont placés dans le même bloc et où les valeurs moyennes dans un cluster particulier sont au centre de ce bloc [47].

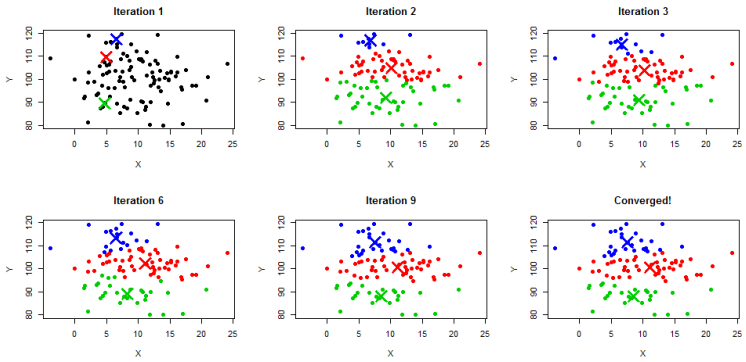

FIGURE 1.15 – k-means [12].

L'algorithme k-mean est très populaire car il est facile à comprendre et à mettre en œuvre.

Généralement applicable aux données de grandes tailles, et choisissez une bonne idée de distance,mais l'inconvénient majore de k-means est exigé de donnée le nombre des cluster(groupe) comme paramètre.

L'algorithme k-means est écrit comme suit (Algorithm 1) [20].

Algorithm 1 Algorithme K-means [20].

Entrée :
D : un ensemble de données contenant n objets
K : nombre de groupe
Sortie : K groupe
choisir arbitrairement k objets de D en tant que
(1) centres de cluster initial ;
(2) Répète
(3) (ré)affecter chaque objet au cluster auquel l'objet est le plus similaire,
basé sur la valeur moyenne des objets dans le cluster ;
(4) mettre à jour les moyens de cluster,c'est-à-dire calculer la valeur moyenne des objets pour chaque cluster ;
Jusqu'à pas de changement

-PCA (Principal Component Analysis)

Méthode de réduction des dimensions souvent utilisée pour réduire les dimensions de grands ensembles de données, en convertissant un grand ensemble de variables en variables plus petites et en préservant la plupart des informations du grand groupe.
L'analyse du composant principal est une méthode statistique utile et est utilisée dans des domaines tels que la compression d'images, la reconnaissance faciale, les neurosciences et l'infographie .

1.5.3 Apprentissage semi-supervisé

Apprentissage semi-supervisé utilise un ensemble de données distinctes et non nommées. Il s'agit donc d'une combinaison d'apprentissage supervisé utilisant uniquement des données privilégiées et d'apprentissage non supervisé utilisant uniquement des données non privilégiées. Il améliore donc grandement la qualité de l'apprentissage non supervisé [29]. La Figure 1.16 illustre l'apprentissage semi-supervisé.

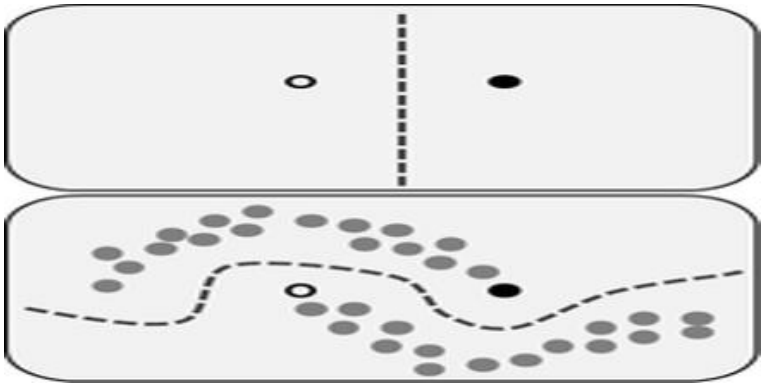

FIGURE 1.16 – Apprentissage semi-supervisés [46].

L'apprentissage semi-supervise contient deux classes importantes sont :

Classification semi-supervisée

L'ensemble d'apprentissage se compose de données étiquetées et non étiquetées, mais la tâche d'apprentissage est principalement supervisée. La classification est effectuée premièrement par un apprentissage avec des données étiquetées, et d'autre part, l'apprentissage est renforcé à l'aide des données non étiquetées. Le but est le même pour l'apprentissage supervisé mais en tirant profit des observations non étiqueté.

Clustering semi-supervisé

Contrairement à la classification semi-supervisée, où l'accent est mise sur le traitement des données manquantes ou insuffisantes dans les algorithmes supervisés, le clustering semi-supervisé est utilisé lorsque que la quantité de supervision est tellement faible ou partielle qu'il est impossible d'appliquer des techniques supervisées.
Les routes actuelles du bloc semi-supervisé sont divisées en deux méthodes générales, appelées méthodes de recherche et similitudes [4].

Quelques techniques d'apprentissage semi-supervisé

Les algorithmes semi-supervisés les plus largement utilisés sont :
- Auto entrainement
- co-apprentissage
- S3vms
- T-SVM
- Algorithme COP k-means [29] : c'est la même chose que la méthode K-means mais avec une modification, COP k-means prend en compte les limitations de la comparaison de paires d'objets.
L'algorithme COP k-means est d'écrit par algorithm 2.

Algorithm 2 Algorithme COP K-means

D : un ensemble de données
$Con_{=}$: un ensemble de contraintes must-link
$Con_{\neq}$:un ensemble de contraintes cannot-link
1 :Sélectionnez aléatoirement :k *point* :centre de clusters initaux
2 : Chaque point $\mathbf{D}_i \in D$ est assigné à son cluster le plus proche tout en assurant qu'ancune contrainte $Con_{=}$ et $Con_{\neq}$ n'est brisée.
3 : Mis à jour de Chaque centre de cluster pour étre le moyen de ses ponits constitutifs.
4 : Répéter(2) et (3) jusqu'à la convergence.

- Algorithme SKMS (Semi-supervised kernel Mean Shift clustering)
L'objectif de cette méthode est d'intégrer la supervision dans la méthode mean shift clustering [3] qui utilise uniquement des contraintes paires pour guider la procédure de clustering.
L'algorithme SKMS généralise l'opération de projection linéaire à une transformation linéaire de l'espace du noyau qui va permettre d'escalader la distance entre les points de contrainte.
À l'aide de cette transformation, les points de must-link sont rapprochés, tandis que les points de cannot-link peuvent être déplacés plus loin.
- Algorithme SKLR
regroupement semi supervisé par des contraintes de distances relatives.étant donné qu'elles sont soumises à un ensemble supplémentaire de limites pour la comparaison de distance relative entre les éléments de données, l'objectif étant d'identifier les informations secondaires qui ne sont pas directement exprimées dans les paramètres de vecteur [29]. L'algorithme SKLR se décompose selon les étapes suivantes :

Algorithm 3 Algorithme SKLR [29].

Entrée :
initiale $(n * n)$: matrice kernel K_0
C_{neq} et C_{eq} : ensemble de comparaisons relatives
γ : Faceur de distance constante
Sortie : matrice kernel K
1 : Trouver une représentation de base niveau :low-rank
2 : Calculez la matrice K de noyau de $(n * n)$ de low-rank K_0
3 : En utilisant une décomposition incompléte de Cholesky
4 : Trouvez $(n * r)$ colonne de la matrice orthogonale Q
5 : Appliquer la transformation $\hat{M} \leftarrow Q^T\ MQ$ Sur tout les matrices
6 : Initialiser la matrice du noyau $\hat{K} \leftarrow K_0$
7 : $*$ Répéter
8 : (1) Sélectionnez une contrainte insatisfaisante $C \in C_{neq} \cap C_{eq}$
9 : (2) Appliquer la projection de Bregman
10 : Jusqu'à ce que toute les contrainte soient satisfaites
11 :$*$ Retour $K \leftarrow Q\ K \succ Q$

1.5.4 Apprentissage par renforcement

L'apprentissage par renforcement est la formation de la machine à l'expérience et aux erreurs, c'est-à-dire qu'elle dépend principalement de l'expérience acquise afin de trouver la meilleure solution possible à un problème particulier.
L'apprentissage par renforcement est une approche courante en robotique,et aussi il est également adapté aux applications d'internet des objets [53].

Quelques techniques d'apprentissage par renforcement

L'algorithme d'apprentissage par renforcement le plus utilisés sont :
- Algorithme TD(0) (Temporal Difference)
Cet algorithme repose sur une comparaison entre la récompense que nous recevons réellement et la récompense que nous attendons d'estimations préétablies. Il repose sur deux processus convergents, le premier fournissant une estimation précise de la récompense immédiate reçue dans chaque état, le second se rapprochant de la fonction de valeur résultant de ces estimations [7].
- Algorithme Sarsa (état S, action a, récompense r, état s, action a)
L'algorithme SARSA est similaire à l'algorithme TD (0), à la différence qu'il fonctionne sur des paires (S,a) au lieu de valeurs case. Son équation de mise à jour

est similaire à l'équation TD (0) en remplaçant la fonction value par la fonction function.
Les informations nécessaires pour effectuer une telle mise à jour alors que l'agent effectue un transfert est le quintupler
(sn,an,rn,sn+1,an+1)[7].

- Algorithme Q-learning

L'algorithme Q-Learning lui-même simplifie l'algorithme SARSA. La seule différence entre SARSA et Q-Learning réside dans la définition du terme d'erreur : l'algorithme SARSA effectue les mises à jour selon les procédures réellement sélectionnées, tandis que l'algorithme Q-Learning effectue les mises à jour selon les procédures optimales même si ce n'est pas optimal. Cette simplicité l'a rendu plus connu et utilisé que d'autres algorithmes d'apprentissage [7].

1.6 Conclusion

Ce chapitre se concentre principalement sur une présentation des définitions et des concepts de l'apprentissage automatique. j'ai commencé par définir l'apprentissage automatique et ses types, puis j'ai mentionner les techniques de chaque classe.

Chapitre 2

TECHNIQUES DE CLASSIFICATION DU TRAFIC INTERNET UTILISANT L'APPRENTISSAGE AUTOMATIQUE

2.1 Introduction

Les utilisateurs, les données et les transactions croissent de façon exponentielle dans nos réseaux. Par conséquent, les utilisateurs cherchent à fournir un service de qualité et une sécurité. C'est exactement ce pour quoi la classification du trafic internet (CTI) est conçue.
Dans ce chapitre, nous commençons par définir CTI et ses modèles et expliquer la classification du trafic Internet ainsi que les caractéristiques, le type et l'importance de cette classification et des technologies qui y sont associées. Ce chapitre comprend des œuvres d'art sur CTI.

2.2 Définition du trafic internet

Le terme trafic internet est notamment représenté sur le mouvement du flux d'informations au niveau d'un réseau informatique (internet) [16].

2.3 Modèles de trafic internet

Il existe différents modèles, qui sont utilisés en fonction de la complexité et de la précision de l'étude [45].

2.3.1 Modèles de renouvellement

Dans les modèles de renouvellement, les arrivées sont indépendantes et réparties symétriquement, mais le trafic internet est souvent fortement corrélé. Par conséquent, l'hypothèse d'un trafic non lié peut conduire à des modèles irréalistes [45].
- **distributions de Poisson** : l'un des modèles de trafic les plus anciens et les plus utilisés. Ils font deux hypothèses de base le nombre de sources est illimité et les modèles de trafic sont aléatoires. cela simplifie les problèmes d'attente impliquant l'accès de Poisson, en le considérant comme un modèle sans mémoire. Une autre caractéristique est que la combinaison de plusieurs flux de poisson génère un nouveau courant de poisson avec une distribution exponentielle et son taux est

la somme de temps d'imbrication des composants pour le processus de Poisson [45].
- **Processus de bernoulli** : sont les processus analogiques temporels discrets des processus de Poisson [45].

2.3.2 Modèles de Markov

Soit X(k) ; k ∈ N un processus. L'ensemble des valeurs possibles de X(k) est appelé l'espace d'états (généralement un sous-ensemble de N). Dans le modèle de trafic Markovien, le changement d'état est interprété comme une nouvelle arrivée (d'un paquet). X(lc) ; k∈N est un processus de Markov si sa distribution conditionnelle de probabilité de l'état futur ne dépend que de l'état présent et non pas des états passés.
Ainsi, la loi conditionnelle de $X(n+1)$ sachant le passé $X(k); 0 \leq k \leq n$ ne dépend que de $X(n)$ seulement, c-à-d :
P [X(n + 1) = x|X(O), X(1),X(2), ... ,X(n)] = P [X(n + 1) = x|X(n)]
où x est un état quelconque du processus. Le processus de Poisson est l'un des processus de Markov les plus utilisés.
Ces processus sont très utilisés dans la théorie des systèmes des files d'attente. Un système de file d'attente se décrit par un processus d'arrivée de paquets, un mécanisme de service et une discipline d'attente.
Dans une file d'attente, on suppose que les durées des inter-arrivées sont indépendantes et de même loi. Par exemple, la loi des arrivées peut être à intervalles réguliers, notée D (déterministe) ou elle peut être Poissonienne, notée M (Markov), etc...
Les durées de service sont des variables positives indépendantes et de même loi. La loi de service peut être de durée constante, notée D (déterministe) ou de durée suivant une loi exponentielle, notée M (Markov), etc...
La discipline d'attente est généralement « premier arrivée, premier servi» (First In First Out - FIFO).
L'inconvénient des modèles Markoviens est qu'ils sont à mémoire courte : l'état du processus dépend seulement de l'état précédent. Ceci est inadéquat pour le trafic caractérisé par une forte corrélation avec le passé et une dépendance à long terme.
Par contre, l'avantage de l'utilisation des processus de Markov par rapport aux autres modèles de trafic est qu'ils permettent, en théorie, de résoudre analytique-

ment des systèmes de files d'attente [55].

2.3.3 Modèles stochastiques linéaires

Contrairement aux modèles de Markov dans lesquels l'état suivant dépend uniquement de l'état actuel, les modèles de régression automatique définissent la variable aléatoire suivante dans une séquence comme une fonction explicite des variables précédentes dans un laps de temps qui s'étend du présent au passé. ces modèles conviennent à la modélisation des dépendances à court terme [45].

2.3.4 Modèles de trafic auto-similaires

Les travaux de recherche ont montré le caractère auto-similaire et la dépendance à long terme du trafic. Ainsi, les modèles traditionnels, tels que le processus de Poisson, les modèles AR, MA ou ARIMA, ne peuvent pas capturer ces dépendances à long terme (Leland et al, 1994 ; Paxon et Floyd, 1995 ; Park et Willinger, 2000 ; Willinger, Paxson et Taqqu, 1998 ; Owezarski et Larrieu, 2004 ; Park, Kim et Crovella, 1997 ; Abry et Veitch, 1998).
Des modèles auto-similaires ont été proposés pour modéliser le trafic. Parmi ces modèles, on cite le mouvement Brownien fractionnaire (Fractional Brownian Motion - FBM) (Mandelbrot et Ness, 1968) et le modèle bruit fractionnaire Gaussien (Fractional Gaussian Noise - FGN). Ces modèles ne peuvent décrire que la dépendance à long terme.
D'autres modèles, tels que le modèle ARIMA fractionnaire (FARIMA), sont capables de décrire la dépendance à long et à court terme du trafic [55].
- **Le modèle « mouvement Brownien fractionnaire»**
Le processus Mouvement Brownien Fractionnaite (Fractional Brownian Motion - FBM) a été défini par (Mandelbrot et Ness, 1968).
Un processus $X(t); t \geq O$ est un mouvement brownien fractionnaire (FBM) s'il est un processus Gaussien d'espérance nulle et vérifiant :
$Cov(X(s), X(t)) = \frac{1}{2}\ (t^{2H} + s^{2H} - |t-s|^{2H})$
ou $Cov(X(s), X(t))$ est la covariance du processus entre les instants s et t et H est le paramètre de Hurst tel que $\frac{1}{2} \leq H \leq 1$.
La modélisation du trafic internet par un mouvement brownien fractionnaire

(FBM) a été proposée par (Norros, 1995 ; Norros etPruthi, 1996). Ainsi, le trafic internet est modélisé par un processus $Y(t); t \geq O$ défini par
$Y(t) = \varphi t + \sqrt{\sigma^2 m} X(t)$.
Le processus Y(t) possède trois paramètres φ, σ^2 et H : le paramètre φ représente le débit moyen du trafic, le paramètre σ^2 représente la variance du débit du trafic et X(t) est un processus FBM de paramètre H (le paramètre de Hurst) [55].

- **Le modèle « Bruit Fractionnaire Gaussien »**

Soit $X(t); t \geq O$ un mouvement Brownien fractionnaire. Le processus Z(t) tel que Z(t) = X(t + 1) - X(t) est appelé bruit fractionnaire Gaussien (Fractional Gaussian Noise - FGN) et il a le même paramètre de Hurst (noté H) que X(t) (Liu et al, 2006).
Ainsi, le trafic internet est modélisé par un processus $Y(t); t \geq O$ défiui par :
$Y(t) = \varphi L + \sigma Z(t)$.
Le processus Y(t) possède trois paramètres φ, σ et H : le paramètre φ représente le débit moyen du trafic, , le paramètre σ représente la variance du débit du trafic et Z(t) est un processus FGN de paramètre H (le paramètre de Hurst).
L'avantage des modèles FBM et FGN est la simplicité de leurs formules pour prévoir le débit du trafic, une fois que les différents paramètres sont estimés. Cependant, leur inconvénient est leur incapacité de capturer les dépendances à court terme du trafic (puisque H $> \frac{1}{2}$) (Crovella et Bestavros, 1997) [55].

- **Le processus « ARIMA Fractionnaire » (FARIMA)**

Le processus ARIMA fractionnaire (FARIMA) est la généralisation naturelle du modèle ARIMA(p, d, q) lorsque le degré de différenciation d peut avoir des valeurs réelles (non seulement des entiers naturels) (Xue et aL, 1999 ; Hosking, 1981a ; Krunz et Makowski, 1998).
Une série temporelle y(t) est un processus FARIMA(p, d, q) si
(1 - $\sum_{i=1}^{p} \phi_i L^i$)$(1 - L^i) y(t)$=$(1 + \sum_{i=1}^{q} \theta_i L^i) \in (t)$
Ou ϕ_i et θ_i sont les paramètres du modèle, $\in$ (t) est un bruit blanc de variance σ^2 et L est l'opérateur « retard» .
Dans le cas Ou p = 0 et q = 0, le processus FARIMA(O, d, 0) est appelé fractional differencing noise model (FDN) (Hosking, 1981a), qui est la forme la plus simple du processus FARIMA. Le paramètre d du processus FARIMA(O, d, 0) indique le niveau de dépendance à long terme tout comme le paramètre de Hurst H. Il a été démontré que, si d $\in [0, \frac{1}{2}]$, le processus FARIMA(O, d, 0) présente une dépendance à long terme avec un paramètre de Hurst H = d + $\frac{1}{2}$ (Hosking,

1981b). Le processus FARIMA(O, d, 0) est similaire au prucessus FGN qui peut seulement décrire une dépendance à long terme.
Le processus FARIMA(p, d, q) peut être vu comme la combinaison du processus ARMA avec le FARIMA(O, d, 0). Ainsi, il est capable de modéliser les processus présentant des dépendances à court et à long terme (Crovella et Bestavros, 1997). (Xue et al, 1999) présentent des directives pour générer du trafic selon le processus FARIMA et pour estimer ses paramètres afin de modéliser et de prévoir un trafic réel. L'inconvénient de ce modèle est la complexité des calculs pour estimer les différents paramètres (Xue et al, 1999) [55].

2.4 Classification du trafic internet

La classification du trafic est l'un des derniers systèmes de sécurité et de gestion de réseau qui nous aide à comprendre la nature du trafic internet [39]. Il consiste à examiner les paquets IP (internet protocole) pour en extraire certaines fonctionnalités qui nous aident à répondre à certaines questions sur le contenu transféré ou les intentions de l'utilisateur. Il traite des flux de paquets définis en tant que séquences de paquets spécifiées par les mêmes adresse IP (internet protocole) source, port source, adresse IP (internet protocole) de destination, port de destination et protocole de couche de transport. Toutefois, les packages peuvent être regroupés de quelque manière que ce soit en fonction des besoins de classification [28].

2.4.1 Type de La classification du trafic internet

La classification du trafic internet a la capacité de résoudre divers problèmes de gestion de réseau difficiles pour les fournisseurs de services internet (ISP) et les fournisseurs d'équipements. Nous devons donc connaître la base ou les critères de cette classification.
La plupart des auteurs ont généralement convenu de distinguer deux grandes catégories de trafic de communication : le trafic de type "stream" et le trafic "élastique" [22].
- le trafic de type "stream", tels que les services téléphoniques et vidéo, qui ont une durée réelle, où il faut contrôler le retard et la modification du transfert de

données et maintenir la sécurité temporelle, le degré de perte de paquets étant acceptable.
- Le trafic "élastique" s'effectue principalement dans la transmission de données, la sécurité sémantique doit être respectée, mais les restrictions de délai de transport sont moins puissantes. Ce type est actuellement utilisé sur les réseaux IP(.
Une analyse des caractéristiques du trafic internet au niveau de la représentation est effectuée selon trois entités de trafic : paquets,flots et sessions [22].
• **Les paquets** constituent l'entité ayant le trafic le plus élevé dans les réseaux de données, l'unité de base traitée par la couche réseau et le processus d'apparition de ceux-ci est très complexe. Le traitement des paquets est en microsecondes et en millisecondes, en fonction de la taille des ordres de taux de transfert de liaison.
•**Les flots** sont considérés comme les mieux adaptés à la réalisation d'études de trafic IP et sont compatibles avec une transmission continue de la chaîne de paquets. Les flux sont liés aux connexions audio / vidéo. Les échelles de temps d'écoulement vont de quelques secondes à quelques minutes, voire quelques heures.
•**Les sessions** sont le niveau le plus élevé et ne diffère pas du niveau de flux. Les sessions sont créées par la couche d'application et sont dimensionnées entre quelques minutes et quelques heures [22].

2.4.2 Importance de la classification du trafic internet

L'importance de la classification du trafic Internet réside dans l'étude de la qualité de service .
La plupart des plans de service (Diffserv [5] ou IntServ [6]) ont un degré de trafic IP dans leur conception : supposons d'abord que les routeurs peuvent reconnaître et distinguer les catégories de trafic, et le second suppose que ces périphériques sont capables de distinguer les catégories de trafic avec précision. La classification du trafic peut également améliorer la qualité du service via internet [52].
Il est actuellement considéré comme le composant central des produits émergents qui soutiennent la qualité de service (QOS) [19] et des architectures QoS automatisées [8].
La classification du trafic est également une solution importante pour répondre aux nouvelles exigences imposant aux réseaux de fournisseurs de services Internet de fournir des fonctionnalités LI pour une utilisation dans le monde de la téléphonie, telles que l'écoute électronique et les journaux d'appels. La classification du

trafic détermine également les modèles de trafic et les catégories d'applications utilisés à tout moment.

2.4.3 Étapes nécessaires pour la classification du trafic internet

Pour classifier le trafic sur Internet, nous devons suivre des étapes importantes, que nous pouvons résumer ci-dessous [2] :
• Première étape consiste à créer un inventaire de tout le contenu .
•Deuxième étape consiste a réduire la taille de l'échantillon en échantillonnant des données volumineuses.
• Troisième étape permet de préparer les données pour l'apprentissage machine.
• Dernière étape consiste a appliquer les algorithmes d'apprentissage pour générer le modèle de classification.

2.5 Domaines d'utilisation de la classification du trafic internet

la classification du trafic internet est utilisé dans de nombreux domaines en raison de sa grande importance, notamment :
- Sécurité : C'est la zone la plus utilisée, en particulier ces dernières décennies, afin de surveiller les données du réseau et de le sauver des attaques, et de connaître l'adresse utilisée si elle est sécurisée ou non, comme l'utilisation du Wi-Fi.
- Sites Web : Surveillez les utilisateurs de ces sites Web, s'il s'agit d'une bonne utilisation, ou si des modifications doivent être apportées.
- Le domaine de l'éducation : en observant ses utilisateurs, s'ils ont des problèmes dans les sites pédagogiques ou qu'ils progressent tel quel dans les sites pédagogiques et en extrayant les leçons.
Applications lentes : cela se fait en connaissant la vitesse de leur travail afin de comprendre la raison de leur lenteur ou ralentissement et de travailler pour les renouveler.

2.6 Application de l'apprentissage automatique dans la classification du trafic internet

Un certain nombre de concepts généraux d'apprentissage automatique prennent une signification spécifique lorsqu'ils sont appliqués à la classification du trafic. Il existe des termes liés aux flux dans les trois termes suivants :
• Flux ou transition unidirectionnel : ensemble de paquets partagés de la même manière : adresses IP source et cible, ports IP source et cible et numéro de protocole.
• Flux bidirectionnel : mouvement unidirectionnel s'exécutant dans des adresses compensées par un autre mouvement unidirectionnel entre les mêmes adresses IP et les ports source et cible.
• Flux complet : Un flux bidirectionnel capturé sur toute sa durée de vie, de l'établissement à la fin de la connexion de communication.

Une classe indique généralement le trafic IP causé par une application ou un groupe d'applications. Les instances sont généralement plusieurs paquets appartenant au même flux. Les caractéristiques sont généralement des attributs numériques calculés sur plusieurs paquets appartenant à des flux individuels. Les exemples incluent les longueurs moyennes des paquets, l'écart type des temps d'arrivée inter-paquets, les longueurs totales de flux (en octets et / ou paquets), la transformée de Fourier de l'heure inter-arrivée des paquets, etc. Comme indiqué précédemment, toutes les fonctionnalités ne sont pas également utiles, de sorte que l'apprentissage automatique choisit le plus petit ensemble de fonctionnalités qui conduisent à une différenciation efficace entre les membres d'une classe et d'autres trafics en dehors de la classe [39].
- Entraînement et test d'un classifieur de trafic de l'apprentissage supervisé :
Les Figures 2.1,2.2 et 2.3 illustrent les étapes nécessaires à la création d'un classifieur de trafic à l'aide d'un algorithme d'apprentissage automatique.
La Figure 2.1 illustre le processus global d'entraînement et de test qui mène au modèle de notation. Le moyen idéal de former un algorithme d'apprentissage automatique supervisé est de fournir des exemples pré-catégorisés de deux types de trafic qui correspondent à la classe de trafic que l'on aimerait identifier plus tard dans le réseau et au trafic représentatif des applications complètement différentes que l'on s'attendrait à voir à l'avenir [39].

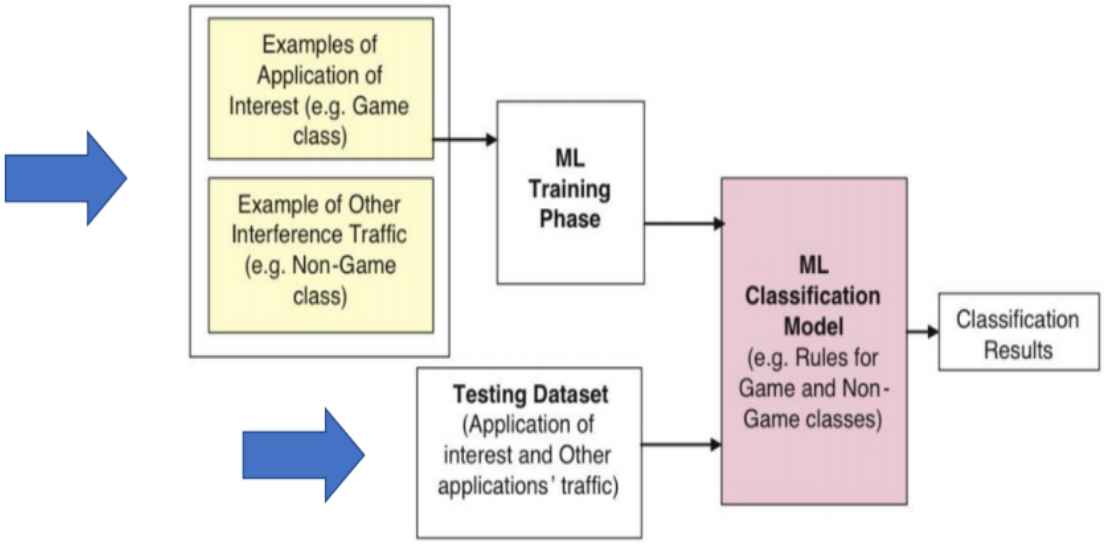

FIGURE 2.1 – Entraînement et test pour un classifieur de trafic d'apprentissage automatique supervisé à deux classes [39].

La figure 2.2 montre la séquence d'événements impliqués dans l'entraînement du classifieur de trafic d'apprentissage automatique supervisé. Premièrement, une combinaison de «suivi du trafic» est collectée qui contient à la fois les cas de la demande d'intérêt et les cas d'autres applications qui se chevauchent.
L'étape "Traitement des statistiques de flux" comprend le calcul des propriétés statistiques de ces flux (telles que l'heure d'arrivée moyenne des paquets, la longueur moyenne des paquets et / ou la durée des flux) en guise d'introduction à la génération de fonctionnalités [39].

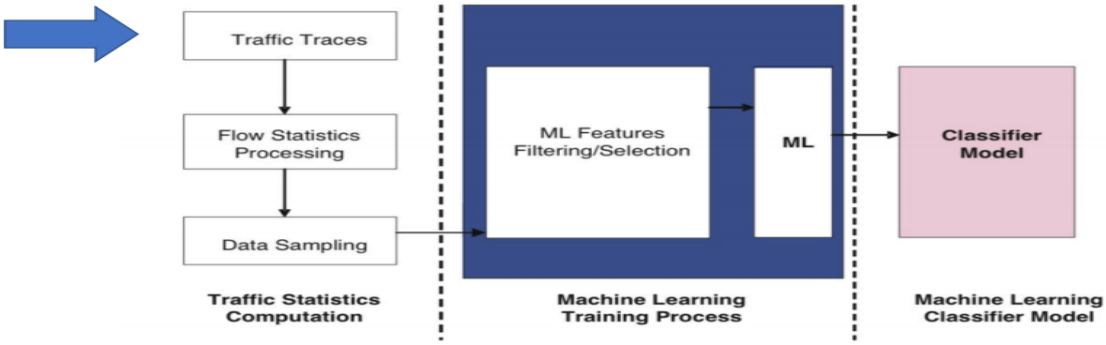

FIGURE 2.2 – Entraînement du classifieur de trafic d'apprentissge automatique supervisé [39].

L'approbation réciproque peut être utilisée pour créer des résultats d'évaluation précis pendant la phase de développement. Toutefois, si les données proviennent de paquets IP groupés au même moment et du même point de mesure du réseau, les résultats de la vérification risquent d'être exagérés [39].

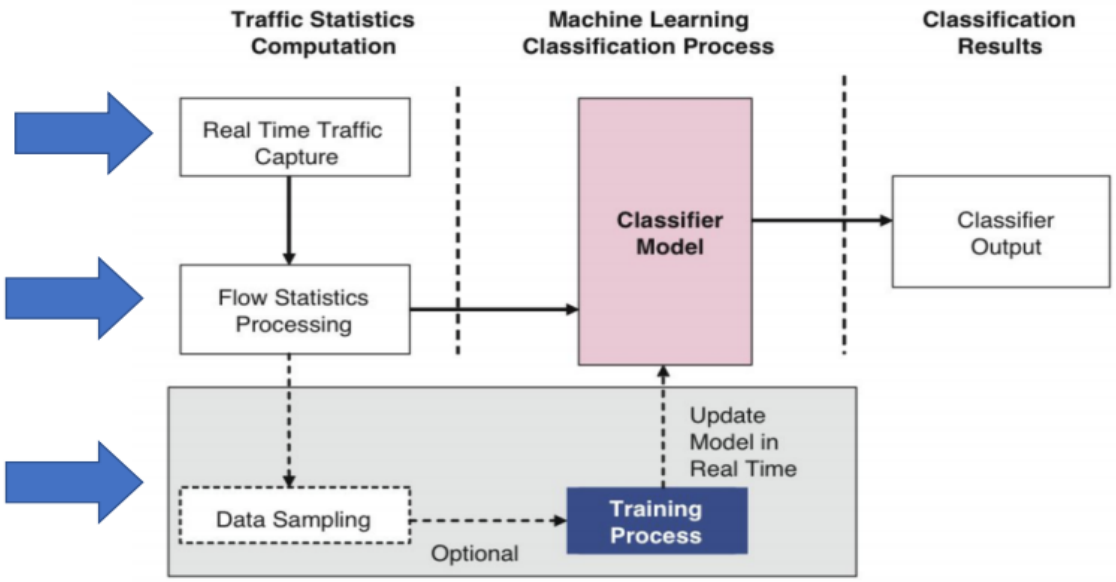

FIGURE 2.3 – Flux de données dans un classifieur de trafic d'un apprentissage supervisé opérationnel [39].

- Apprentissage supervisé ou non supervisé

Comme indiqué précédemment, la classification du trafic IP consiste généralement

à identifier le trafic appartenant à des applications connues dans des flux de paquets IP auparavant invisibles. Le principal défi consiste à déterminer la relation entre les classes de trafic IP et les applications à l'origine du trafic IP.
Les schémas de l'apprentissage supervisés nécessitent une phase de formation pour renforcer le lien entre les classes et les applications. La formation nécessite une pré-classification des flux dans les ensembles de données de formation. Pour cette raison, l'apprentissage automatique supervisé peut être intéressant pour identifier une application spécifique qui suscite l'intérêt. Cependant, un classifieur d'apprentissage automatique supervisé fonctionne mieux lorsqu'il est formé à des exemples de toutes les catégories qu'il s'attend à voir dans la pratique. Par conséquent, ses performances peuvent être dégradées ou faussées s'il n'est pas entraîné dans un mélange de trafic analogique ou si la liaison réseau sous surveillance commence à voir un trafic d'application auparavant inconnu [42].
Lors de l'évaluation de programmes de l'apprentissage supervisés dans un contexte opérationnel, il convient de considérer comment le classificateur recevra des exemples de formation supervisée adéquats, quand il sera nécessaire de se recycler et comment l'utilisateur détectera un nouveau type d'applications.
Il peut sembler que l'un des avantages des schémas de l'apprentissage non supervisés est la découverte automatique de classes grâce à la reconnaissance de modèles «naturels» dans l'ensemble de données.
Cependant, les clusters résultants doivent encore être étiquetés (par exemple, via une inspection directe par un expert humain) afin que les nouvelles instances puissent être correctement mappées aux applications [15].
Un autre problème pour les schémas d'apprentissage non supervisés est que les clusters ne correspondent pas nécessairement aux applications. L'idéal serait que le nombre de clusters formés soit égal au nombre de classes d'applications à identifier et que chaque application domine un groupe de clusters. Cependant, en pratique, le nombre de clusters est souvent supérieur au nombre de classes d'applications [54] . Une application peut s'étendre et dominer un certain nombre de clusters, ou inversement, une application peut également s'étendre mais ne dominer aucun des clusters. Le mappage d'un cluster à une application source peut devenir un défi de taille.
Relier un group à une application source peut devenir un défi de taille.
Lors de l'évaluation de schémas d'apprentissage non supervisés dans un contexte opérationnel, il vaut la peine de considérer comment les clusters seront étiquetés,

comment les étiquettes seront mises à jour lorsque de nouvelles applications sont détectées et le nombre optimal de clusters.

2.7 État de l'art

Après avoir introduit la classification du trafic et les bases de l'apprentissage automatisé, nous présentons maintenant certains travaux antérieurs qui portent sur l'utilisation de l'apprentissage automatique de la classification du trafic.

Réseau de neurone

- Kuldeep Singh et Agrawal ont continué à classer le trafic IP en utilisant RBF dans leurs études [50], ils ont conclu que RBF fonctionnait mieux que le réseau de neurones de prolifération postérieure. Le temps d'entraînement et la complexité des FBR sont très élevés. Par conséquent, cette technique n'est pas efficace pour classer le trafic IP sur Internet. Un meilleur classement peut être obtenu avec d'autres technologies machine learning .

- Agrawal et Sohi ont prouvé que les applications poste à poste constituaient une partie importante du trafic Internet actuel. Il revêt une grande importance dans les opérations de réseau, ainsi que dans l'ingénierie du trafic applicatif, la planification de la capacité, la fourniture de ressources, la différenciation des services, etc. Où une approche réseau neuronal du trafic P2P a été introduite à l'aide du réseau Perceptron Multilayer Network (MLP). Le résultat est que l'ensemble "de fonctionnalités" global est plus efficace car il peut améliorer la précision moyenne de 1,98% et le rappel moyen de 27,81% sur l'ensemble des fonctionnalités sélectionnées dans la méthode conventionnelle. Notez que l'appel est très volumineux, car une résolution élevée n'est importante que lorsque le classeur atteint une valeur de réponse élevée [1].

SVM

W.Gao, Y.Tian, T.Huang, Q.Yang in [17] proposent une classe hiérarchique de trois niveaux, le premier niveau dépend d'une classe SVM binaire qui sépare le

trafic en deux catégories (P2P et non P2P) et le second niveau sépare le trafic P2P en plusieurs catégories. Utilisation de SVDD et utilise le niveau III pour identifier des applications spécifiques. La publication d'un classeur hiérarchique peut aider à déterminer le trafic P2P en temps réel. Cependant, la solution proposée ne peut pas respecter cette dernière propriété car elle utilise le temps de flux calculé uniquement une fois le flux terminé.

Gowsalya et Amali in [18] montrent que SVM offre des performances très appréciées. On peut en conclure que la performance des algorithmes dépend, entre autres choses, de la nature et de l'importance des données.

Arbres de décision

Soysal et Schmidt ont également présenté une approche systématique pour étudier et évaluer les performances de la classification du trafic Internet pour trois algorithmes : les réseaux bayésiens (BN), les arbres de décision (DT) et les perceptrons multicouches (MLP). Les résultats de la présentation montrent que les arbres de décision ont une résolution et un taux de commandes plus élevés et que les arbres de décision nécessitent un temps d'installation plus long et sont progressivement vulnérables en raison de la perte de base ou de petites mesures de préparation des informations [51].

Naive bayes

La catégorisation du trafic à l'aide d'approches statistiques est l'une des premières solutions dans ce cadre, comme celle proposée par Moore et Zuev (2005) in [56] dans laquelle ils utilisaient la technique naïve de Bayes.
Alors que Singh et Ajrawal [56] obtenaient leur premier trafic Internet en temps réel en utilisant le logiciel Wire Shark, un outil de collecte. Le trafic Internet a été catégorisé à l'aide de cinq tâches ML. Les résultats ont montré que Bays's Net fournit une meilleure classification des données de trafic Internet en termes d'exactitude de classification, de temps de formation de classeur, de valeurs d'appel et de précision de classeur pour des applications Internet individuelles.

C4.5

Le C4.5 (classification 4.5 par Ross Quinlan) in [49] a également donné des résultats très précis avec un ensemble de données d'entité réduit, dont les performances ont été grandement améliorées. Ainsi, Bays'Net et C4.5 sont deux des méthodes Machine Learning les plus efficaces pour classer le trafic IP à 94%.

Autres recherches

Une partie du travail repose également sur des approches mixtes, dans lesquelles la classification est basée sur les ports et sur l'approche statistique. Afin de surmonter les limites de chaque approche de classification, Awad et ses collaborateurs (2014) ont proposé sur cette base un système de classification hybride appelé signature statistical and port classifier (SSPC).

TABLE 2.1 – Travaux connexes dans le trafic internet (travail personnel)

Travail	Année	Technique
Moore et Zuev	2005	Naive bayes
W.Gao, Y.Tian, T.Huang, Q.Yang	2010	SVM
Soysal et Schmidt	2010	Arbres de décision
Singh et Ajrawal	2011	Naive bayes
Kuldeep Singh et Agrawal	2011	Réseau de neurone
Agrawal et Sohi	2011	Réseau de neurone
Gowsalya et Amali	2014	SVM
Awad et ses collaborateurs	2014	SSPC

.
- SVM : Support vecteur machine.
- SSPC : Signature statistical and port classifier.

2.8 Mesures d'évaluation

La plupart des chercheurs utilisent la précision pour évaluer le trafic Internet, la précision est une mesure pour évaluer les modèles de classification. De manière informelle, la précision est la fraction des prédictions que notre modèle a obtenues.

Formellement, la précision a la définition suivante :

$$Prcision = \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

Pour la classification binaire, la précision peut également être calculée en termes de positifs et de négatifs comme suit :

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

Ces termes sont définis ci-dessous dans la matrice de confusion

TABLE 2.2 – Matrice de confusion pour la classification binaire

	Predicted YES	Predicted NO
Actual YES	True Positive	False Negative
Actual NO	False Positive	True Negative

Définitions des termes

- False négatifs (FN) : est le nombre d'exemples positifs mal classés.
- False positif (FP) : est le nombre d'exemples négatifs mal classés.
- True positif (TP) : est le nombre d'exemples positifs correctement classés.
- True négatif (TN) : est le nombre d'exemples négatifs correctement classés.

Toutes les mesures étant limitées à 0% (mauvais) et à 100% (optimal).

2.9 Conclusion

Dans ce chapitre, nous avons étudié la classification du trafic internet, où nous avons expliqué certaines des caractéristiques et l'importance de ce mouvement et les étapes nécessaires pour cela.
Nous avons également démontré des techniques de trafic internet en appliquant l'apprentissage automatique à la classification du trafic.
Enfin, nous avons mentionné quelques-uns des travaux qui ont été accomplis dans ce domaine.
le chapitre suivant, nous avons faire quelque études appliquée expérimentation sur l'algorithme SVM qui présent dans ce chapitre.

Chapitre 3

EXPÉRIMENTATIONS

3.1 Introduction

Dans ce chapitre, nous appliquons l'une des applications d'apprentissage automatique que nous avons étudiées dans le chapitre précédent, qui a été appliquée à un échantillon d'étudiants.
Mais avant de commencer à développer les détails de l'expérience, nous avons jugé utile de rappeler l'application, le programme et l'environnement dans lesquels cette étude a été réalisée.

3.2 Outils utilisés

3.2.1 Matériel

Pour réaliser notre expérimentation, nous avons configuré un cluster qui contient un seul noeud, ce noeud est un ordinateur portable ACER aspire, basée sur un processeur Intel (R) i3-3, équipé de 2 Go de RAM, sous Windows 7, 32 bits.

3.2.2 Logiciel

Dans cette section, nous montrons comment installer et configurer les logiciels utilisés pour notre système.

pycharm

Pycharm fait partie de la boîte à outils de programmation JetBrains [24]. Qui contient des environnements pour générer du code dans différents langages tels que PHP .Il est également considéré comme l'un des environnements de développement les plus complets en Python.
- Pycharm est disponible en trois éditions :
1-pycharm Edu est gratuit et à des fins éducatives.
2-pycharm Community est également gratuit et destiné au développement Python .
3-pycharm Professionnel est payé, a tout ce que l'édition communautaire et est également très bien adapté pour le développement et scientifique .

Langage R

- R est un logiciel de développement scientifique spécialisé dans le calcul et l'analyse statistique, il est un langage interprété et multi-plateforme (Linux, Mac, Windows). -il crée par Ross Ihaka et Robert Gentleman on 1993 au département de statistique de l'université d'Auckland, on 1995 dépôts des codes sources sous licence GNU/GPL, et on 2002 la fondation dépose ses statuts sous la présidence de Gentleman et Ihaka, mais on 2010 c'est le début de Rstudio [9].

3.2.3 Raison du choix pycharm et R$_s$*tudeo*

- J'ai choisi l'environnement pycharm parce qu'elle contient des packages prêts à l'emploi qui m'aide à résoudre mon problème (classification du trafic internet), cet environnement est très simple à utiliser. De plus, l'environnement pycharm est complet et générer du code dans différents langages [24].

- J'ai choisi logiciel Rstudio parce qu'il est simple et direct à utiliser surtout au fonction mathématique et plot et les graphes qui nous devons utiliser, en plus il contient des fonction qui facilite l'utilisation du data set.

3.3 Architecture du système :

Le but de la création de notre système est de classifier le trafic internet à partir d'un ensemble de données à l'aide d'un algorithme d'apprentissage automatique (SVM). Son architecture se compose d'un ensemble de processus successifs comme le montre .

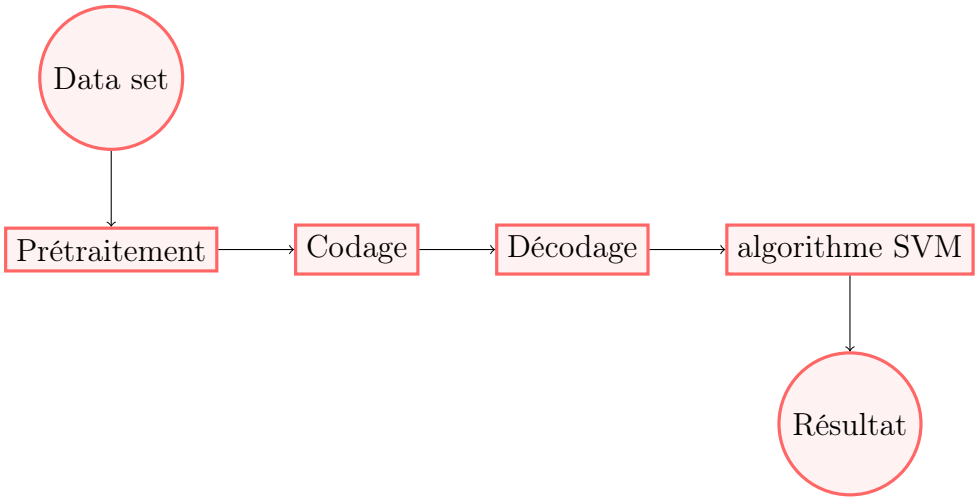

FIGURE 3.1 – Architecture générale de notre système.

- Prétraitement L'ensemble de données utilisé est un ensemble de fichiers au format Excel. Il est nécessaire de le décoder avec un outil capable de lire le type de ce fichier, nous utilisons donc Wireshark [13] pour cela. Nous savons également qu'il est difficile d'intégrer des paquets avec des caractéristiques différentes dans le même ensemble de données, car chaque protocole a ses propres caractéristiques. Pour faire tout cela, nous appliquons un filtre avec Wireshark, et enregistrons le résultat de ces packages dans un fichier pour effectuer le traitement et conserver tous les attributs des packages et la facilité de suivi dans la mise en œuvre du programme.

- Codage Pour chaque élément de l'ensemble de données, nous le remplaçons par un symbole entier unique qui devrait être un entier, tel que : 0, 1, 2, etc. En même temps, nous faisons un dictionnaire de l'objet pour trouver son symbole, nous utilisons une bibliothèque avec des algorithmes prêts à l'emploi.

Nous avons créé un programme pour ce faire, mais il y avait un problème. Il peut répète la même valeur dans différents attributs, il peut prendre la même valeur, et notre codeur de programme le considère comme unique. Nous avons proposé comme solution de reformater les données avant codage en concaténant chaque valeur à sa propriété. Ainsi, il sera clairement codé.

- Décodage Dans cette étape, nous décodons les éléments précédents à l'aide du dictionnaire créé lors de l'étape de codage.

3.4 Data set :

Sur la base de notre étude, plusieurs ensembles de données peuvent être utilisés dans le domaine du trafic Internet. Cependant, bon nombre de ces groupes sont anciens et peu fiables. Il manque une foule de fonctionnalités et de métadonnées.

Ce travail a été effectué sur des données (Data set) stockées dans fichier excel, qui a été pris à l'Université de Laghouat (Amar Telidji) après des Journées d'étude qui se sont tenus à leur niveau.
Cet ensemble de données est une collection de fichiers au format Excel. Comme ces données se composent de plusieurs colonnes, comme le nom d'utilisateur, qui se compose de 132 étudiants et a une adresse IP différente. En plus des informations entrantes et sortantes pour tous les étudiants, ainsi que le temps de téléchargement et le temps de navigation.
Voici un exemple sur lesquelles nous avons travaillé :

donnée.PNG
Type : Image PNG
Taille : 48,0 Ko
Dimension : 713 x 555 pixels

	Users	Hits	Incoming (MB)	Outgoing (MB)	Download Time (h:m:s)	Browsing Time (h:m:s)	% of Hits
		489	4 982,41	81.06	18.93509259	1.05885417	0.117
		85	87	5.30	1.19270833	1.23008102	0.019
3	192.168.3.222	59 319	179.91999999999999	23.26	4.42949074	4.36951389	0.018
4	192.168.2.141	47 976	1 976,06	26.51	3.12394676	1.25350694	0.015
5	192.168.2.166	47 514	993.66999999999996	19.02	1.82871528	0.47060185	0.015
6	192.168.1.53	41 947	88.719999999999999	24.25	0.50467593	0.90421296	0.013
7	192.168.1.149	39 509	292.06999999999999	11.36	0.73833333	0.91528935	0.012
8	192.168.7.20	39 190	398.41000000000003	9.61	0.86891204	0.22921296	0.012
9	192.168.25.120	39 094	795.47000000000003	10.65	0.57560185	0.31074074	0.012
10	192.168.1.25	38 681	142.69999999999999	12.90	0.46685185	1.04712963	0.012
11	192.168.1.78	37 810	91.689999999999998	11.09	1.32045139	1.75325231	0.012
12	192.168.2.189	36 604	1 042,49	63.02	6.52857639	0.76047454	0.011
13	192.168.1.123	36 576	145.11000000000001	2.47	0.23660880	0.20625000	0.011
14	192.168.1.32	33 134	295.61000000000001	10.72	0.63050926	0.84707176	0.010
15	192.168.25.76	32 444	1 088,96	368.17	12.86892361	2.30152778	0.010
16	192.168.6.36	31 838	256.75999999999999	3.43	0.62731481	0.29608796	0.010
17	192.168.1.67	31 623	5 808,16	9.87	4.71453704	1.07284722	0.010
18	192.168.3.181	31 408	380.51999999999998	22.35	0.68799769	0.73890046	0.010
19	192.168.1.107	30 811	394.83999999999997	41.94	0.49256944	0.63802083	0.009
20	192.168.2.97	27 446	359.33999999999997	9.35	4.46032407	3.21396991	0.008
21	192.168.1.129	27 117	218.47999999999999	8.48	0.54920139	0.72789352	0.008

FIGURE 3.2 – Ensemble de données utilisé .

- J'ai utilisé R Studio pour afficher ces données et les préparer avec certaines

fonctions qui nous aident à faciliter nos tâches.

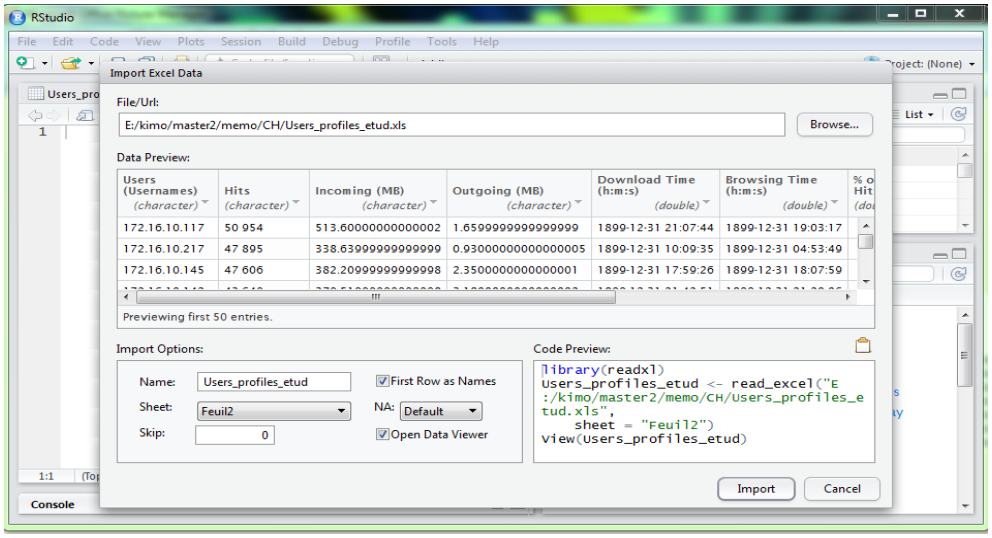

FIGURE 3.3 – Importe les données stockées.

- Cette image montre comment importer data set.

```
> View(Users_profiles_etud)
>
```

FIGURE 3.4 – Afficher data set.

- Cette image afficher data set.

```
> Hits <- Users_profiles_etud$`% of Hits`
> Hits
  [1] 0.018 0.017 0.017 0.016 0.015 0.015 0.015 0.015 0.015 0.014 0.014 0.014 0.014
 [14] 0.014 0.014 0.014 0.013 0.013 0.013 0.013 0.012 0.012 0.012 0.012 0.012 0.012
 [27] 0.012 0.012 0.012 0.011 0.011 0.011 0.011 0.011 0.011 0.011 0.011 0.011 0.010
 [40] 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010
 [53] 0.010 0.009 0.009 0.009 0.009 0.009 0.008 0.008 0.008 0.008 0.008 0.008 0.008
 [66] 0.008 0.008 0.008 0.008 0.008 0.007 0.007 0.007 0.007 0.007 0.007 0.007 0.007
 [79] 0.007 0.007 0.007 0.007 0.007 0.007 0.006 0.006 0.006 0.006 0.006 0.006 0.006
 [92] 0.005 0.005 0.005 0.005 0.005 0.004 0.003 0.003 0.003 0.003 0.002 0.002 0.002
[105] 0.002 0.002 0.002 0.002 0.002 0.002 0.001 0.001 0.001 0.001 0.001 0.001 0.001
[118] 0.001 0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
[131] 0.000 0.000 1.000
```

FIGURE 3.5 – Affiche la colonne Hits du dataset.

- Cette image afficher le colonne hits du data set.

3.5 Algorithme utilisé

Dans notre travail, nous avons utilisé l'algorithme SVM dont nous avons parlé dans le premier chapitre, car il fait partie des algorithmes les plus utilisés parmi les utilisateurs généraux, car il aide à résoudre le problème de classification correctement, rapidement et avec une grande précision.
- on a utiliser les packages du python comme suit :

1- init.py : c'est le début de progrmme

```
1  from .linear import LinearSVM
2  from .kernel import KernelSVM
3  import logging
4  logger_svm = logging.Logger('SVM')
5  logger_svm.setLevel(logging.ERROR)
6
```

FIGURE 3.6 – Partie init.

```
logger_svm = logging.Logger('SVM')
logger_svm.setLevel(logging.ERROR)
```

FIGURE 3.7 – Fonction logger pour la création d'objet.

- Cette fonction est conçue pour la création de l'objet logger.

2- kernel.py : c'est le noyau de programme

```
1   import logging
2   from .solver import fit_kernel
3   from .linear import LinearSVM
4   from util.generate_data import *
5   from util import kernel_dict
6   from functools import partial
7   logging.basicConfig(level=logging.DEBUG)
8
9
10  class KernelSVM(LinearSVM):...
57
```

FIGURE 3.8 – Partie kernel.

-On va défini la partie kernel comme suit :

```
logging.basicConfig(level=logging.DEBUG)
```

FIGURE 3.9 – Fonction logging.

-Cette fonction conçu comme une simple configuration ensuite a une appel a debug.

```
class KernelSVM(LinearSVM):
    def __init__(self):
        self.X = None
        self.y = None
        self.alphas = None
        self.b = None
        self.kernel = None
    def fit(self, X, y, kernel_type='rbf', k=1.0):
        self.kernel = partial(kernel_dict[kernel_type.lower()], k=k)
        # Get alphas
        C = 1.0  # Penalty
        alphas = fit_kernel(X, y, C, self.kernel)
        # normalize
        alphas = alphas / np.linalg.norm(alphas)
        # Get b
        b_vector = y - np.sum(self.kernel(X) * alphas * y[:, None], axis=0)
        b = b_vector.sum() / b_vector.size
        # Store values
        self.X = X
        self.y = y
        self.alphas = alphas
        self.b = b
    def predict(self, X):
        prod = np.sum(self.kernel(self.X, X) * self.alphas * self.y[:, None],
                      axis=0) + self.b
        y = np.sign(prod)
        return y
```

FIGURE 3.10 – Class kernel.

- Dans cette fonction nous avons implémanter la version noyau de support vector machine.au début on a définis le fichier init puis on a aussi ajuster le modèle selon aux données d'entraînement,puis obtenus alpha pour faire la normalisation et stocker la valeur.

3- linear.py

```
1   import logging
2   from .solver import *
3   from util.generate_data import *
4   logging.basicConfig(level=logging.DEBUG)
5
6
7   class LinearSVM(object):...
6
```

FIGURE 3.11 – Partie linear.

-on va défini comme suit :

```
class LinearSVM(object):
    def __init__(self):
        self.x = None
        self.y = None
        self.w = None
        self.b = None
    def fit(self, X, y, soft=True):
        if soft is True:
            C = 1.0  # Penalty
            alphas = fit_soft(X, y, C)
        else:
            alphas = fit(X, y)
        w = np.sum(alphas * y[:, None] * X, axis=0)
        b_vector = y - np.dot(X, w)
        b = b_vector.sum() / b_vector.size
        norm = np.linalg.norm(w)
        w, b = w / norm, b / norm
        self.w = w
        self.b = b
    def predict(self, X):
        y = np.sign(np.dot(self.w, X.T) + self.b * np.ones(X.shape[0]))
        return y
```

FIGURE 3.12 – Class linear.

- Dans cette fonction nous avons faire implémentation de la machine vectorielle support linéaire.

4- solver.py

```
import numpy as np
from cvxopt import matrix, solvers

solvers.options['show_progress'] = False

def fit(x, y):...

def fit_soft(x, y, C):...

def fit_kernel(x, y, C, kernel_fun):...
```

FIGURE 3.13 – Partie solver.

- Dans cette fonction nous avons ajuster les alphas pour SVM à double problème avec marge dure.puis nous obtenons le noyau.

5- exemple.py :c'est le fichier qui grace a lui,nous faisons importer dataset pour faire l'exécution.

```
from svm import LinearSVM, KernelSVM
from util.generate_data import generate_data, plot_data_separator, \
    train_test_split
from util.metric import accuracy
import logging
import numpy as np

X, y = generate_data(dataname='../Users_profiles_etud.excel')
X_train, y_train, X_test, y_test = train_test_split(X, y, prop=0.25)
...
clf = LinearSVM()
clf.fit(X=X_train, y=y_train, soft=True)
acc_train = accuracy(clf, X=X_train, y=y_train)
acc_test = accuracy(clf, X=X_test, y=y_test)
logging.info('Accuracy (on training) = {}'.format(acc_train))
logging.info('Accuracy (on test) = {}'.format(acc_test))
# plot_data_separator(X, y, clf.w, clf.b, '../svm.png')


clf = KernelSVM()
clf.fit(X=X_train, y=y_train, kernel_type='rbf', k=1.0)
acc_train = accuracy(clf, X=X_train, y=y_train)
logging.info('Accuracy (on training) kernel rbf = {}'.format(acc_train))
acc_test = accuracy(clf, X=X_test, y=y_test)
logging.info('Accuracy (on testing) kernel rbf = {}'.format(acc_test))
```

FIGURE 3.14 – Importer dataset au algorithme utilisé .

.

3.6 Résultats et discussion

Auparavant, nous avons parlé du protocole d'évaluation de notre système, et dans cette section, nous discuterons des résultats après avoir utilisé l'algorithme SVM (Machine Vector Support). Malheureusement, nous n'avons pas été en mesure d'achever les travaux et d'atteindre les résultats souhaités.

3.7 Conclusion

Ce chapitre décrit le côté pratique de notre recherche sous Classification du trafic internet à l'aide de l'apprentissage automatique, où nous avons utilisé à la fois pycharm et Rstudio. De plus, nous avons parlé de l'architecture proposée de notre système, de nos algorithmes et de notre ensemble de données et de la manière de l'utiliser. Nous avons conclu le chapitre par une expérience pour évaluer les résultats du travail du système, mais nous n'avons malheureusement pas pu atteindre les résultats prévus pour les atteindre.

CONCLUSION ET TRAVAUX FUTURS

Dans ce travail, nous avons examiné et catégorisé les concepts liés au trafic Internet, et notre étude s'est également concentrée sur l'application des techniques d'apprentissage automatique. Ce dernier concerne la mise en œuvre de l'algorithme SVM. Malheureusement, nous n'avons pas été en mesure d'achever nos travaux et d'atteindre les résultats escomptés.
Notre travail futur représente l'achèvement de ce avec quoi nous avons commencé et l'ajout d'améliorations supplémentaires au niveau du sujet car il est d'une grande importance, dans nos expériences, nous avons inclus très peu de données en raison du temps et du manque d'équipement. À l'avenir, nous explorerons l'utilisation d'énormes quantités de données, qui pourraient être au centre de nos travaux futurs.

Bibliographie

[1] S Agrawal and BS Sohi. Generalization and optimization of feature set for accurate identification of p2p traffic in the internet using neural network. *WSEAS Transactions on Communications*, 10(2) :55–65, 2011.

[2] Massih-Reza Amini. *Apprentissage automatique et recherche de l'information : application à l'extraction d'information de surface et au résumé de texte*. PhD thesis, Paris 6, 2001.

[3] Saket Anand, Sushil Mittal, Oncel Tuzel, and Peter Meer. Semi-supervised kernel mean shift clustering. *IEEE transactions on pattern analysis and machine intelligence*, 36(6) :1201–1215, 2013.

[4] Sugato Basu et al. *Semi-supervised clustering : Learning with limited user feedback*. Computer Science Department, University of Texas at Austin, 2003.

[5] Steven Blake, David Black, Mark Carlson, Elwyn Davies, Zheng Wang, and Walter Weiss. An architecture for differentiated services. Technical report, 1998.

[6] Robert Braden, David Clark, and Scott Shenker. Integrated services in the internet architecture : an overview. Technical report, 1994.

[7] Olivier Buffet, Frédéric Alexandre, François Charpillet, Alain Dutech, Frédéric Garcia, Claude Kirchner, Manuel Samuelides, and Olivier Sigaud. Apprentissage par renforcement pour la conception de systemes multi-agents. *Rapport d'avancement de these*, 2002.

[8] J But, N Williams, S Zander, L Stewart, and G Armitage. Angel-automated network games enhancement layer. 2006.

[9] Laurent Cauquil. Introduction au logiciel r. 2019.

[10] Clement Chatelain. Les support vector machine (svm). Technical report, Technical report, 2003.

[11] Antoine Cornuéjols. Une nouvelle méthode d'apprentissage : Les svm. séparateurs à vaste marge. *Bulletin de l'AFIA*, 51 :14–23, 2002.

[12] Vinod Kumar Dehariya, Shailendra Kumar Shrivastava, and RC Jain. Clustering of image data set using k-means and fuzzy k-means algorithms. In *2010 International Conference on Computational Intelligence and Communication Networks*. IEEE, 2010.

[13] L Dhanabal and SP Shantharajah. A study on nsl-kdd dataset for intrusion detection system based on classification algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(6) :446–452, 2015.

[14] Alican Dogan and Derya Birant. Machine learning and data mining in manufacturing. *Expert Systems with Applications*, 166 :114060, 2021.

[15] Jeffrey Erman, Martin Arlitt, and Anirban Mahanti. Traffic classification using clustering algorithms. In *Proceedings of the 2006 SIGCOMM workshop on Mining network data*, pages 281–286. ACM, 2006.

[16] Fatih Ertam and Engin Avcı. A new approach for internet traffic classification : Ga-wk-elm. *Measurement*, 95 :135–142, 2017.

[17] Wen Gao, Yonghong Tian, Tiejun Huang, and Qiang Yang. Vlogging : A survey of videoblogging technology on the web. 2010.

[18] RS Anu Gowsalya and SMJ Amali. Svm based network traffic classification using correlation information. *International Journal of Research in Electronics and Communication Technology (IJRECT 2014), ISSN*, pages 2348–9065, 2014.

[19] Eric Guichard. *Mesures de l'internet*, volume 6. Citeseer, 2004.

[20] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining : concepts and techniques*. Elsevier, 2011.

[21] Mohamadally Hasan and Fomani Boris. Svm : Machines à vecteurs de support ou séparateurs à vastes marges. *Rapport technique, Versailles St Quentin, France. Cité*, 64, 2006.

[22] Khadija Ramah Houerbi. *Mesures et Caracté risation du Trafic dans le Réseau National Universitaire (RNU)*. PhD thesis, Ecole nationale des sciences de l'informatique, université de Manouba, Tunis, 2009.

[23] Fatima Hussain, Rasheed Hussain, Syed Ali Hassan, and Ekram Hossain. Machine learning in iot security : Current solutions and future challenges. *arXiv preprint arXiv :1904.05735*, 2019.

[24] Quazi Nafiul Islam. *Mastering PyCharm*. Packt Publishing Ltd, 2015.

[25] Seong-Gyun Jeong. *Modélisation de structures curvilignes et ses applications en vision par ordinateur*. PhD thesis, 2015.

[26] Pierre-Marc Jodoin and Carl Lemaire. Ift 603–techniques d'apprentissage. *mars*, page 1, 2020.

[27] Yves Kodratoff, Ryszard Stanislaw Michalski, Jaime Guillermo Carbonell, and Tom Michael Mitchell. *Apprentissage symbolique : une approche de l'intelligence artificielle*. Cépaduès-éditions, 1993.

[28] Maciej Korczynski and A Duda. Classifying application flows and intrusion detection in internet traffic. *Ecole Doctorale Mathema-tiques, Sciences et Technologies de l'Information, Informatique (EDM-STII), Grenoble, France*, 2012.

[29] Salsabil Lakhdari and Amaria Saidi. *Étude des techniques d'apprentissage semi-supervisé par regroupement*. PhD thesis, 2017.

[30] Dan Li, Kerry D Wong, Yu-Hen Hu, and Akbar M Sayeed. Detection, classification, and tracking of targets. 2002.

[31] Daniel Lowd and Pedro Domingos. Naive bayes models for probability estimation. In *Proceedings of the 22nd international conference on Machine learning*, pages 529–536. ACM, 2005.

[32] P Mahé. Noyaux pour graphes et support vector machines pour le criblage virtuel de molécules. *Rapport de stage, DEA MVA*, 2003, 2002.

[33] Medjeded MERATI. *Les MODELES DEFORMABLES et leurs applications dans la segmentation d'images médicales.* PhD thesis, Tiaret, 2009.

[34] Ryszard S Michalski. Learning by being told and learning from examples : an experimental comparison of the two methods of knowledge acquisition in the context of development an expert system for soybean disease diagnosis. *International Journal of Policy Analysis and Information Systems*, 4(2) :125–161, 1980.

[35] Federica Minichiello. Machines, données et apprentissage : relations et enjeux. *Revue internationale d'éducation de Sèvres*, (74) :12–15, 2017.

[36] Tom Michael Mitchell. *The discipline of machine learning*, volume 9. Carnegie Mellon University, School of Computer Science, Machine Learning, 2006.

[37] Dan Munteanu. A quick survey of text categorization algorithms. *Annals of University\Dunarea de Jos" of Galati*, 1, 2007.

[38] Marref Nadia. *Apprentissage Incrémental &Machines à Vecteurs Supports.* PhD thesis, 2013.

[39] Thuy TT Nguyen and Grenville Armitage. A survey of techniques for internet traffic classification using machine learning. *IEEE communications surveys & tutorials*, 10(4) :56–76, 2008.

[40] Nils J Nilsson. Introduction to machine learning. an early draft of a proposed textbook (1998). *Software available at http ://robotics. stanford. edu/people/nilsson/mlbook. html.*

[41] Fernando Santos Osório. *INSS : un système hybride neuro-symbolique pour l'apprentissage automatique constructif.* PhD thesis, Institut National Polytechnique de Grenoble-INPG, 1998.

[42] Junghun Park, Hsiao-Rong Tyan, and C-C Jay Kuo. Internet traffic classification for scalable qos provision. In *2006 IEEE International Conference on Multimedia and Expo*, pages 1221–1224. IEEE, 2006.

[43] Ricco Rakotomalala. Arbres de décision. *Revue Modulad*, 33 :163–187, 2005.

[44] Amir mohammad Rooshenas, Hamid R Rabiee, Ali Movaghar, and M Yousof Naderi. Reducing the data transmission in wireless sensor networks using

the principal component analysis. In *2010 Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 133–138. IEEE, 2010.

[45] Gemma Maria Echevarria Sanchez, Timothy Van Renterghem, Pieter Thomas, and Dick Botteldooren. The effect of street canyon design on traffic noise exposure along roads. *Building and Environment*, 97 :96–110, 2016.

[46] Warren S Sarle. Algorithms for clustering data, 1990.

[47] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos : Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1) :3–30, 2011.

[48] HA Simon. Machine learning : An artificial intelligence approach, chapter why should machine learn, 1983.

[49] Kuldeep Singh and Sunil Agrawal. Comparative analysis of five machine learning algorithms for ip traffic classification. In *2011 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, pages 33–38. IEEE, 2011.

[50] Kuldeep Singh and Sunil Agrawal. Feature extraction based ip traffic classification using machine learning. In *Proceedings of the International Conference on Advances in Computing and Artificial Intelligence*, pages 208–212. ACM, 2011.

[51] Murat Soysal and Ece Guran Schmidt. Machine learning algorithms for accurate flow-based network traffic classification : Evaluation and comparison. *Performance Evaluation*, 67(6) :451–467, 2010.

[52] Lawrence Stewart, Grenville Armitage, Philip Branch, and Sebastian Zander. An architecture for automated network control of qos over consumer broadband links. In *TENCON 2005-2005 IEEE Region 10 Conference*, pages 1–6. IEEE, 2005.

[53] Sofia Zaidenberg. *Apprentissage par renforcement de modeles de contexte pour l'informatique ambiante*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2009.

[54] Sebastian Zander, Thuy Nguyen, and Grenville Armitage. Automated traffic classification and application identification using machine learning. In *The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05) l*, pages 250–257. IEEE, 2005.

[55] Mohamed Faten Zhani. Prévision du trafic internet : modèles et applications. 2011.

[56] Denis Zuev and Andrew W Moore. Traffic classification using a statistical approach. In *International workshop on passive and active network measurement*, pages 321–324. Springer, 2005.