République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université 20 Aout 1955-SKIKDA

Faculté des Sciences

Département d'informatique

Mémoire de fin d'études en vue de l'obtention du diplôme de Master en Informatique

Option : Réseaux et Systèmes Distribués (RSD)

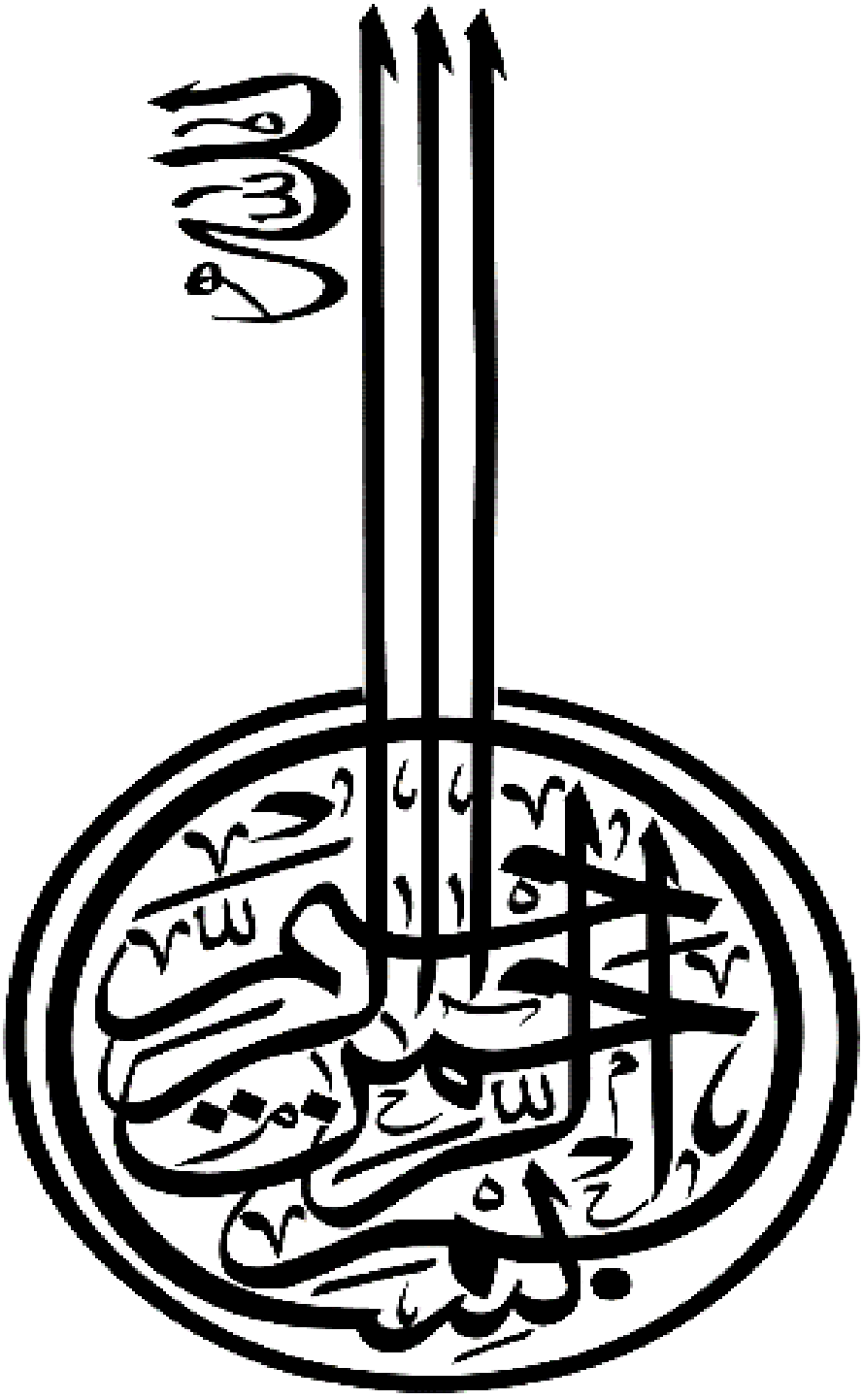
Thème

Méthodes basées apprentissage automatique pour la détection de spams en messagerie électronique

Réalisé par :
- BOUZNAD CHAFIA

Encadré par :
Pr. MAZOUZI SMAINE

Année Universitaire 2023-2024



Remerciements

Au nom d'Allah, le Tout Miséricordieux, le Très Miséricordieux. Louange à Allah, Seigneur de l'univers, qui m'a donné la force et le courage pour mener à bien ce travail. Je suis très reconnaissante au Professeur MAZOUZI SMAINE dont l'expertise, la disponibilité et les précieux conseils ont été la pierre angulaire de ce travail de recherche. Sa patience et son soutien continu ont été des moteurs essentiels de mon engagement et de mes progrès.

À ma mère, mon cœur tendre et mon refuge sûr, tu as été une lumière éclairant mon chemin et un visage d'espoir à chaque instant. Merci pour ton amour sans limites et ta compassion infinie.

À mon père, que Dieu ait son âme, tu as été une étoile brillante dans ma vie, et ton souvenir restera éternellement gravé dans mon cœur. Je prie Allah qu'Il t'accorde Sa miséricorde et te fasse habiter en Son vaste paradis.

À ma famille, je suis infiniment reconnaissant pour leur soutien constant, leur encouragement permanent et leur amour inconditionnel. Leur présence et leur compréhension ont été une source de réconfort et de motivation tout au long de ce voyage.

À mes amis et à mes proches, je tiens à exprimer ma gratitude pour leur soutien moral, leurs encouragements chaleureux et leur présence bienveillante, qui ont dissipé les doutes et enrichi cette expérience.

Je tiens également à remercier les membres du jury, [noms des membres du jury], pour leurs efforts et leur temps consacrés à l'examen de ce travail, ainsi que pour leurs commentaires et observations précieuses qui ont contribué à son amélioration.

Enfin, je tiens à exprimer ma gratitude à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce travail. Leurs conseils, leurs discussions stimulantes et leur soutien logistique ont été des éléments précieux pour le succès de ce projet.

Dédicaces

À mes chers parents,

Je tiens à commencer par vous exprimer ma gratitude et mes remerciements, car vous êtes la principale raison de mon succès et de la réalisation de cette grande réussite. Vous avez implanté en moi les valeurs du travail acharné et de la persévérance, et vous m'avez soutenu avec amour et dévouement tout au long de mon parcours scolaire. Vous avez toujours été une source d'inspiration pour moi, et je n'oublierai jamais tous les efforts que vous avez déployés pour moi. Merci pour tout. À mon cher superviseur, le Dr. MAZOUZI SMAINE

Je tiens à vous exprimer ma profonde gratitude pour tous les conseils précieux et le soutien continu que vous m'avez apportés tout au long de mes études. Grâce à vous, j'ai pu surmonter les défis et atteindre mes objectifs avec succès. Vous avez été un modèle d'effort et de dévouement pour moi. Merci pour tout votre travail précieux et pour vos encouragements constants.

Enfin, à tous mes chers proches,

Je vous remercie pour tout le soutien et l'encouragement que vous m'avez apportés tout au long de ce voyage. Vous avez toujours été à mes côtés, ce qui m'a donné la force de persévérer dans les moments difficiles. Merci infiniment.

ملخص

تتعرض الشبكات الحاسوبية وأنظمة البريد الإلكتروني لتهديدات مستمرة، مثل البريد العشوائي، مما يستدعي تطوير تقنيات حماية متقدمة تشمل أنظمة الكشف عن البريد العشوائي. نظراً لأن الطرق التقليدية أصبحت غير كافية لمواجهة تطور البريد العشوائي،في مشروع التخرج النهائي هذا في الماستر

قمنا بإجراء دراسة تجريبية لتحديد أفضل مصنف للكشف عن البريد العشوائي) الشبكات والأنظمة الموزعة RCD(

تم تقييم عدة مصنفات في هذه الدراسة، بما في ذلك مصنف بايز البسيط، وآلات المتجهات الداعمة

وأشجار القرار، والغابات العشوائية، والشبكات العصبية، من خلال مقارنة دقتها وأوقات تنفيذها(SVM) .

الكلمات المفتاحية: الأمن الحاسوبي، التعلم الآلي، الكشف عن البريد العشوائي، المصنفات، البريد الإلكتروني

Résumé

Dans ce projet de mémoire de master en RCD (Réseaux et Systèmes Distribués), nous avons mené une étude expérimentale pour identifier le meilleur classifieur pour la détection de spam. Nous avons évalué plusieurs classifieurs, notamment le classifieur bayésien naïf, les machines à vecteurs de support (SVM), les arbres de décision, les forêts aléatoires et les réseaux neuronaux, en comparant leur précision et leurs temps d'exécution.

Mots clés : sécurité informatique, apprentissage automatique, détection de spam, classifieurs, email

Abstract

In this master's thesis project in Distributed Networks and Systems (RCD), we conducted an experimental study to determine the best classifier for spam detection. Several classifiers were evaluated, including Naive Bayes, Support Vector Machines (SVM), Decision Trees, Random Forests, and Neural Networks, comparing their accuracy and execution times.

Keywords: computer security, machine learning, spam detection, classifiers, email

SOMMAIRE

Sommaire

SOMMAIRE

SOMMAIRE

SOMMAIRE

Liste des figure

Liste des tableaux

Liste des tableaux

Introduction générale

Les réseaux et les systèmes informatiques sont devenus des outils indispensables au fonctionnement des entreprises. Ils sont aujourd'hui élaborés dans tous les secteurs professionnels: les universités, les banques, les assurances, ou encore le domaine militaire. L'information gérée par ces systèmes fait l'objet de convoitises. Elle peut être exposée à des attaques qui exploitent des éléments vulnérables du système d'information. La détection des actions malveillantes est rapidement devenue une nécessité. Les mesures de prévention se sont révélées insuffisantes et ont amené la création de systèmes de détection d'intrusions (IDS: Intrusion Systèmes de détection). Les techniques basées signature, les techniques basées comportement, et les techniques basées sur les algorithmes du machine learning. Cette dernière catégorie a gagné du terrain ces dernières années comparées aux deux autres catégories. Ceci peut être expliqué par la difficulté de concevoir des signatures, ou des comportements modèles, notamment avec la prolifération des attaques, qui se généralisent notamment avec l'informatique pervassive, et avec l'avènement de l'internet des objets (IoT : Internet of Things).

Dans ce mémoire de fin d'étude, nous présentons une étude des techniques d'apprentissage automatique qui ont été massivement appliquées ces dernières années pour la détection d'intrusion. À titre d'exemple : Arbres de décision, Bayes naïf, les K-plus proches voisins (K-NN). L'objectif est de montrer quel est le classifieurs le mieux adapté aux données d'apprentissage. Pour se faire, nous allons tester deux classifieurs avec apprentissage supervisé, à savoir le classifieur K-NN et le classifieur C 4.5. Le dataset considéré est un ensemble de séquences de données du trafic réseau, étiquetées selon le type d'attaque qu'il véhicule. Afin de quantifier les résultats nous avons tester les modèles étudiés en utilisant la plateforme en ligne Google Colab, avec codage en Python.

Ce mémoire est organisé comme suit :

Le chapitre 1 est consacré à la sécurité informatique et la détection d'intrusion. Le chapitre présente essentiellement les différents types d'attaques contre les systèmes informatiques, et aussi les différentes mesures de protection et cela utilisant Intrusion Detection Systems. Après avoir présenté l'intérêt des IDS, et les différentes architectures selon lesquelles sont construit. Il a été introduit également les IDS collaboratifs.

Le chapitre 2 : est consacré à l'intelligence Artificielle et les méthodes d'apprentissage automatique. Le chapitre présente l'ensemble de théories et de techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence, et cela utilisant les méthodes d'apprentissage supervisé et non supervisé Avec une explication de la différence entre eux.

Le chapitre 3 : présente notre démarche pour la mise en œuvre de classifieurs considérés. Dans ce chapitre, nous avons donnons un aperçu des classifieurs supervisés, et nous avons détaillé les classifieurs que nous allons utiliser.

Le chapitre 4 : présente les outils logiciels utilisés, les éléments d'implémentation, et quelques résultats expérimentaux.

Chapitre I: Sécurité Informatique

1. Introduction

L'ouverture et l'extension continuelle et importante des systèmes informatiques via les réseaux privés et l'internet, a rendu le problème de la sécurisation des systèmes et des communications à la fois importante et difficile. Dans ce chapitre, nous présentons le domaine de la sécurité informatique en général, et la sécurité des réseaux et des communications en particulier. Ainsi, nous présentons les attaques qui peuvent être menées contres les systèmes informatiques et les réseaux de communication, et aussi les mesure de protections qui leur sont associées.

1.1 Définition de l'informatique

En français, le mot informatique est bien souvent galvaudé.

Au quotidien, ce mot est fréquemment assimilé à ``ordinateur muni de logiciels", comme dans ``nous utilisons l'outil informatique pour simplifier l'administration", ``l'informatique est en panne" ou ``c'est la faute de l'informatique". Ce n'est pas cette informatique-là qui fait l'objet de ce livre.

L'informatique est aussi une discipline universitaire. Malgré le mot ``science" de la traduction anglaise ``computer science", l'informatique n'est pas typiquement une science. En effet, le mot ``science" est plus souvent associé à une discipline basée sur l'étude d'un phénomène réel, l'observation du phénomène et la construction de modèles l'expliquant le plus fidèlement possible.

Comme la mathématique, l'informatique n'étudie pas les phénomènes réels. Ces deux disciplines ont le privilège de pouvoir construire leur propre monde sous la forme d'objets abstraits. En mathématique, il s'agit de nombres, de relations, de fonctions, de transformations, etc. En informatique, on manipule (entre autres) des algorithmes, des programmes, des arbres, des preuves, des systèmes de réécriture, des images numériques et les vedettes de ce livre : les graphes.[1]

1.2 Historique de l'informatique

Dès les origines, il était nécessaire à l'homme de calculer de compter et de traiter des informations. Pour y arriver il utilisait certaines méthodes primitives, telle que graver des marques sur les os pour le comptage. Ces méthodes évoluaient avec le temps, jusqu'à arriver aujourd'hui à l'ordinateur, capable de faire des diverses opérations extraordinaires.

[1] : https://dept-info.labri.fr/ENSEIGNEMENT/INITINFO/initinfo/supports/book/node4.html

Grâce à son cerveau électronique pouvant gérer une complexité d'informations, ainsi que des opérations logiques et arithmétiques, l'ordinateur est le principal outil de l'informatique, pour ne pas dire qu'il est à la base de celle-ci.

La première machine à calculé utilisant l'addition et la soustraction fut inventée en 1642 par le Français BLAISE PASCAL Sa découverte fut un grand pas ayant conduit aux ordinateurs actuels.

On peut citer Von LEIBNIZ (mathématicien) qui, en 1673, ajouta la multiplication et la division dans la machine de Pascal, ainsi que BABBAGE (mathématicien) qui Contribua pas mal à l'automatisation des opérations sur la machine, en 1833.

En 1835 le fondateur de IBM (International Business Machines), l'américain HERMAN HOLLERITH réalisa les premières machines à cartes perforées.

En 1944, le professeur américain HOWARD AIKEN à l'Université de Harvard, construit le premier ordinateur appelé "Calculateur automatique Mark 1".

Les premiers ordinateurs qui fonctionnaient à l'aide des composants électromécaniques occupaient de grands espaces. Grâce à l'évolution de la technologie, la taille de l'ordinateur est considérablement réduite. Les tubes à vide ont d'abord remplacées les pièces électromécaniques, ensuite viennent les transistors et enfin aujourd'hui on a des circuits intégrés occupant de très petits espaces négligeables par rapport aux précédents[2]

2. La sécurité informatique

La sécurité des systèmes d'information (SSI) ou plus simplement sécurité informatique, est l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires à la mise en place de moyens visant à empêcher l'utilisation non autorisée, le mauvais usage, la modification ou le détournement du système d'information. Assurer la sécurité du système d'information est une activité du management du système d'information.

Aujourd'hui, la sécurité est un enjeu majeur pour les entreprises ainsi que pour l'ensemble des acteurs qui l'entourent. Elle n'est plus confinée uniquement au rôle de l'informaticien. Sa finalité sur le long terme est de maintenir la confiance des utilisateurs et des clients. La finalité sur le moyen terme est la cohérence de l'ensemble du système d'information. Sur le court terme, l'objectif est que chacun ait accès aux informations dont il a besoin. La norme traitant des systèmes de management de la sécurité de l'information (SMSI) est l'ISO/CEI

[2] : https://www.memoireonline.com

27001 qui insiste sur Confidentialité – Intégrité – Availability, c'est-à-dire en français disponibilité, intégrité et confidentialité[3]

2.1 Objectifs de la sécurité informatique

Le système d'information est généralement défini par l'ensemble des données et des ressources matérielles et logicielles de l'entreprise permettant de les stocker ou de les faire circuler. Le système d'information représente un patrimoine essentiel de l'entreprise, qu'il convient de protéger.

La sécurité informatique, d'une manière générale, consiste à assurer que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu.

La sécurité informatique vise généralement cinq principaux objectifs :

L'intégrité : c'est-à-dire garantir que les données sont bien celles que l'on croit être.

La confidentialité : consistant à assurer que seules les personnes autorisées aient accès aux ressources échangées.

La disponibilité : permettant de maintenir le bon fonctionnement du système d'information.

La non répudiation : permettant de garantir qu'une transaction ne peut être niée.

L'authentification : consistant à assurer que seules les personnes autorisées aient accès aux ressources [4]

2.2. Définition de spam

Le spam est un message électronique non sollicité, envoyé massivement à un grand nombre de destinataires, à des fins publicitaires ou malveillantes. Le terme spam est aussi utilisé pour désigner le même type de message transmis par d'autres moyens de communication électroniques tels que les messageries instantanées, les blogs, les forums, et plus récemment, des réseaux de téléphonie mobile, via les SMS ou MMS. Même si le moyen de communication est différent, les techniques d'envoi et de détection restent relativement similaires [5].

[3] https://fr.wikipedia.org
[4] https://www.commentcamarche.net/contents/1033-introduction-a-la-securiteinformatique
[5] https://fr.wikipedia.org

3. Historique de spam

Il est loin le temps où les gens seuls et sans amis ne recevaient jamais de courrier ! Grâce aux spams, leur boîte aux lettres est pleine tous les jours ! Inutile d'expliquer ce qu'est le « spam », il suffit d'aller faire un tour sur votre logiciel de messagerie pour en avoir toute une collection ! Ils sont les équivalents des prospectus qui remplissent nos boîtes aux lettres (les physiques, celles que l'on ouvre avec une clef en fer).

C'est un certain Gary Thuerk qui, en 1978, fut le premier petit malin qui compris qu'il était vraiment très facile d'envoyer d'un clic un message publicitaire à des centaines de personnes grâce aux adresses e-mail. 600 personnes connectées au réseau ARPA net (l'ancêtre d'internet) reçurent son message, et beaucoup condamnèrent cette pratique jugée d'emblée abusive. 35 ans plus tard, force est de constater que ce Gary a fait des émules : quelles que soient les sources, on estime que les courriers non désirés à caractère commercial représentent

plus de 90 % des messages échangés. Et si d'autres mots comme « pourriels » sont apparus en France et au Canada pour les dénommer, c'est pourtant l'étrange terme de « spam » qui demeure le plus utilisé, y compris en Europe.

L'histoire de ce terme et de sa propagation virale illustre une nouvelle fois comment les mots et les sens trouvent sur les réseaux des chemins nouveaux et originaux. Par cette magie, une marque de jambon épicé en boîte (Spiced Ham), mets récurrent dans la gamelle des soldats de la Seconde Guerre mondiale, se retrouve aux côtés des mots doux de nos proches.

C'est d'abord la faute aux Monty Python et de leur Sketch « Spam » de 1999 dans lequel le nom de la boîte de conserve finit par envahir toutes les conversations.

Très appréciés des premiers internautes, les humoristes britanniques eurent très tôt sur Usenet un newsgroup qui leur était consacré, newsgroup dans lequel le sketch « spam » fut très souvent évoqué, puis parodié : les contributeurs utilisant de la même façon le terme de manière récurrente, envahissante. Il ne fallut pas longtemps pour que les parodies du « spam » contaminent d'autres newsgroups, et que le terme soit utilisé par les contributeurs pour qualifier un message inopportun. A l'origine, le Spam est un donc un même, une de ces vidéos amusantes qui se propagent comme une traînée de poudre. Le spam n'est donc plus de la viande, mais il reste souvent très épicé : les sites à caractère pornographique utilisent largement de moyen de promotion, ainsi que le secteur de la contrefaçon, les médicaments censés doper les performances sexuelles et les arnaqueurs de tout poil.

Contre le spam, la lutte est acharnée ! Des systèmes et des logiciels anti spam tentent d'enrayer la prolifération des spams. Mais les spams ont souvent une longueur d'avance ! On

retrouve désormais des spams dans nos téléphones portables et dans les blogs. Les spams sont parfois freinés par la fermeture de sites diffuseurs de spams : en 2008, la fermeture d'un seul site a fait chuter le volume des spams de 70 %… Pour seulement deux mois ! Car les inventeurs de spams ont trouvé la parade : des virus qui transforment votre PC en « machine zombie » qui envoie partout des spams ! Pourtant, les enjeux de la lutte contre les spams sont multiples : fiabilité des échanges, sécurité et économie d'énergie. Car les spams ne sont pas gratuits : les spams font gaspiller plus de 33 milliards de kWh par ans ! Alors si vous recevez un courrier non sollicité, n'hésitez spam : jetez-le de ce spam [6].

3.2 Type du spam

- **Les spams par e-mail :** ce sont les spams les plus communs. Ils encombrent votre boîte de réception et vous distraient des e-mails que vous voulez réellement lire. Rassurez-vous, vous pouvez tout à fait les ignorer.
- Les spams SEO : également connu comme du « spamdexing », il s'agit d'un abus des méthodes d'optimisation des moteurs de recherche (SEO) qui permettent d'améliorer le classement du site Web du spammeur dans les résultats de recherche. Les spams SEO se divisent en deux grandes catégories
- **Les spams de contenu :** les spammeurs peuplent leurs pages d'une quantité de mots clés populaires, généralement sans rapport avec leur site Web, afin de classer leur site au plus haut dans les résultats de recherches de ces mots clés. D'autres réécrivent le contenu existant pour rendre leurs propres pages plus intéressantes et uniques.
- **Les spams de lien :** si vous êtes tombé sur un commentaire de blog ou un message de forum contenant des liens non pertinents, vous avez eu affaire à un spam de lien. Le spammeur essaie d'exploiter un mécanisme de référencement appelé « backlinking » pour générer du trafic vers sa page.
- **Les spams de réseaux sociaux :** au fur et à mesure que le « social » gagne du terrain sur Internet, les spammeurs en profitent pour diffuser leur spam via de faux comptes « jetables » sur les plateformes de réseaux sociaux populaires.
- **Les spams de mobiles :** ce sont des spams sous forme de SMS. En plus des SMS, certains spammeurs utilisent également des notifications push pour attirer votre attention sur leurs offres.

[6] https://www.ideematic.com/actualites/2013/07/spam-petite-histoire-des-pourriels/

- **Des spams de messagerie :** ils s'apparentent aux spams par e-mail, mais en plus rapide. Les spammeurs diffusent leurs messages sur les plateformes de messagerie instantanée, notamment WhatsApp, Skype et Snapchat [78].

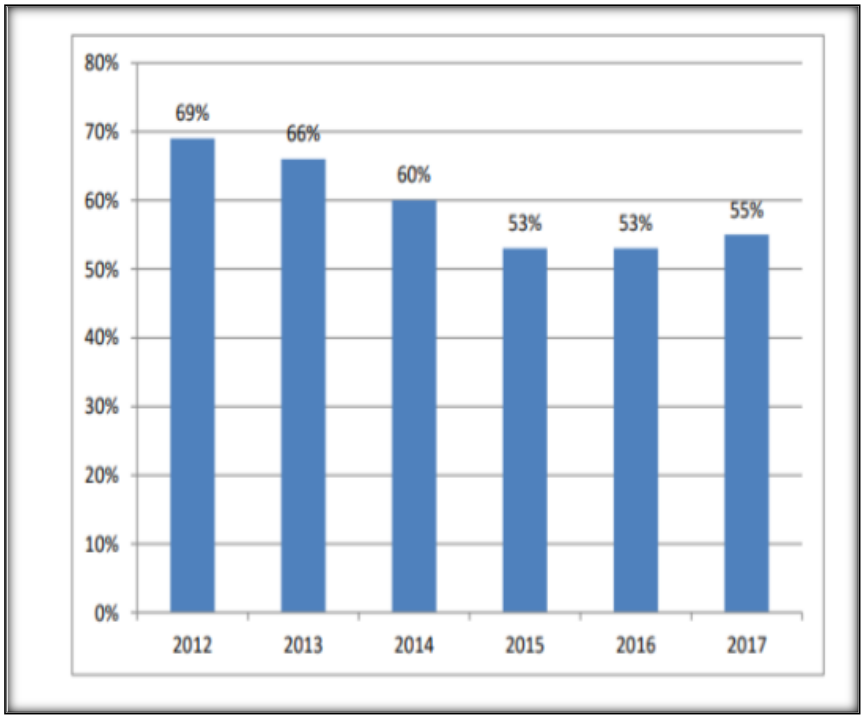


Figure 1 : Développement de spam en termes de volume

4. Attaques web

Au même titre qu'une application classique ou qu'un système d'exploitation, les applications Web peuvent présenter des failles de sécurité. Cela est d'autant plus grave que les applications Web manipulent parfois des données confidentielles (mots de passe, numéros de cartes bancaires) et qu'elles sont généralement d´éployées sur Internet et donc exposées au public. Même sur un serveur Web sécurisée tournant sur un système d'exploitation répute sur (Apache sur Open BSD, par exemple), des failles de sécurité peuvent subsister, car elles sont la plupart du temps dues à des fautes de programmation de l'application elle-même, et non du serveur.

Un firewall IP conventionnel permet de filtrer au niveau de la couche réseau (IP) et de la couche transport (TCP, UDP). Les règles sont définies en fonction de l'adresse IP source, l'adresse IP de destination, le numéro de port source, le numéro de port de destination, l'état de la connexion (flags), l'interface d'entrée et de sortie du firewall, etc... Un firewall IP n'offre absolument aucune protection contre les attaques visant les applications Web, dans la mesure où celles-ci ont lieu au niveau applicatif : elles utilisent le protocole HTTP sur le

[7] https://www.avast.com/fr-fr/c-spam

port 80, au même titre que le trafic Web ordinaire. Pourtant, de SOAP(Protocole d'accès aux objets simple) aux applets Java en passant par les scripts ActiveX, un grand nombre de menaces peuvent être véhiculées par ce canal apparemment inoffensif et qui est laissé ouverts sur la plupart des firewalls d'entreprise. La figure 1 illustre bien la problématique des protocoles pouvant contenir des scripts malveillants passant par le port HTTP.

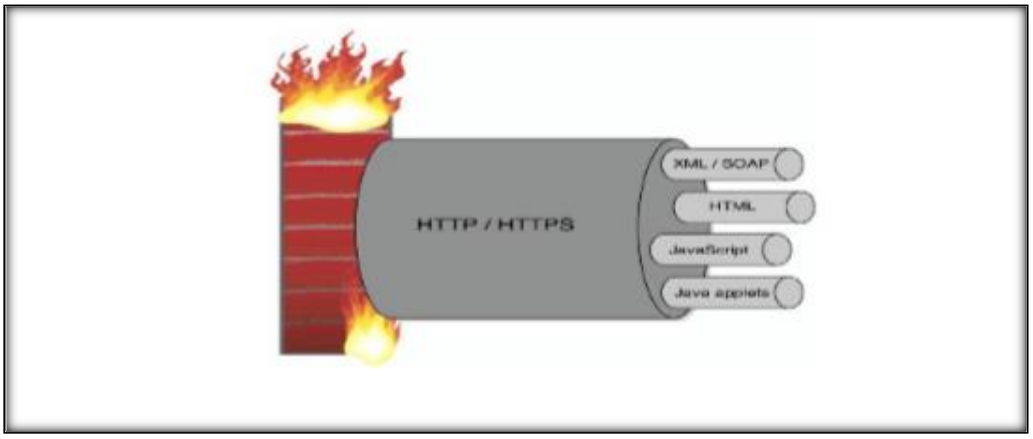


Figure 2: Portée d'un firewall de niveau 3 et 4 du modèle OSI

Un des majeurs problèmes des applications Web à architecture 3-tiers (Figure 2), viens du fait que celle-ci envoie directement ses requêtes SQL vers le SGBD1. Si l'application ne prend pas garde aux formats des paramètres saisis par l'utilisateur de la page web, avant de les inclure dans une requête SQL, il se peut que les tuples retourné par le SGBD soient totalement invalide. En effet, l'utilisateur aura la possibilité de fournir des paramètres de syntaxe SQL sans que l'application Web ne1 s'en rende compte. Les éventuelles cracker auront donc la possibilité de formater les paramètres entrés sur la page Web, afin que la requête leur retourne des informations pertinentes[9]

4.1. Comprendre les mécanismes internes d'une attaque informatique

Mais que se passe-t-il vraiment, quel est le modus operandi ? Si aujourd'hui, on a tendance à bannir la technicité ou la complexité dans les propos publics pour faire « simple », n'est-il pas nécessaire de comprendre les mécanismes internes d'une attaque pour mieux la contrer ? C'est ce que nous croyons fermement.

Comme en architecture du bâtiment, l'informatique peut connaître des défauts de construction. Telle une fissure dans un mur qui le fragilise, et dont un coup de burin bien placé suffit à le faire tomber, le hacker malveillant n'a qu'à chercher les failles et taper dedans. Dans le Top 10 des failles de sécurité dont on dit qu'elles sont « exploitées » par

[9] Petit fichier texte, fourni par le serveur, présenté à chaque appel de nouvelle page par le client

les attaquants figurent le Cross-Site Scripting (XSS, en abrégé) et "l'injection SQL" (SQLi). Des termes certes un peu barbares, mais qu'il convient de bien retenir en ce qu'ils représentent aujourd'hui la porte d'entrée de plus de la moitié des attaques[10]

4.2. Le top 10 des attaques web

L'OWASP (Le projet de sécurité des applications Web ouvertes) tient à jour un classement des 10 vulnérabilités les plus rencontrées dans les applications Web. Celles-ci sont données dans le tableau1

Tableau 2: Le top 10 des attaques Web

[10] : https://www.latribune.fr/opinions/tribunes/comprendre-les-dessous-des-attaques web-pour-mieux-les-contrer
768751. html? Fbclid=IwAR3Ho75PRMIUW2LtQ67sJBGmFbbr96wzZW48nxs2Wrq98Zn385PiRe JvaMg

Attaque web	Définition
Oubli de valider les entrées des utilisateurs	Un classique, qui permet aux pirates de faire accepter des commandes au serveur à travers un formulaire web ou une simple URL, ou d'exécuter des contenus dynamiques (JavaScript, par exemple) chez les autres utilisateurs d'un site.
Contrôle d'accès inefficace	Mauvaise mise en œuvre des outils de contrôle d'accès (fichier .htpasswd lisible par tous, mots de passes nuls par défaut, etc...).
Mauvaise gestion des sessions	Cela permet aux pirates de "voler des sessions" d'autres utilisateurs (en devinant un numéro de session simple, en dérobant un cookie, ou en allant regarder les fichiers de sessions de PHP).
Cross-Site-Scripting	Un autre grand classique, lui aussi lié à un manque de contrôle des entrées de l'utilisateur. Cette faille touche les sites web qui laissent les internautes publier du code HTML susceptible d'être vu par les autres utilisateurs du site (dans un forum, par exemple). Cela permet d'exécuter des contenus dynamiques sur les navigateurs des internautes, avec les droits associés au site web.
Dépassement de mémoire tampon	Une faille vieille comme le monde, qui frappe certains langages de programmation plus que d'autres (le C, par exemple). Si des composants CGI sont (mal) écrits dans ces langages, il peut-être simple de compromettre totalement le serveur par une telle attaque.
Injection de commandes	Là encore, la source de la faille est un manque de contrôle des entrées de l'utilisateur. Elle permet au pirate de faire exécuter des commandes au serveur (au système d'exploitation ou à un serveur SQL, par exemple) en les attachant à une entrée web légitime avant que celle-ci ne soit transmise au serveur.
Mauvaise gestion des erreurs	Les messages d'erreur utiles aux développeurs le sont souvent aussi pour les pirates ! Il faut donc penser à les supprimer une fois le développement terminé.
Mauvaise utilisation du chiffrement	La mise en œuvre du chiffrement au sein des applications web se révèle ardue. Des développeurs non spécialisés peuvent commettre des erreurs difficiles à déceler et créer ainsi une protection illusoire.
Failles dans l'administration	

distante	C'est la voie royale : si les pages réservées aux administrateurs du site ne sont pas réellement protégées (authentification forte du client, chiffrement, contrôles réguliers...), un pirate peut prendre le contrôle du site sans avoir à pirater le serveur. Une aubaine, en quelque sorte.
Mauvaise configuration du serveur web et des applications	Un classique : le serveur web qui permet de lister n'importe quel répertoire, ou les outils de développement qui laissent des versions temporaires des fichiers, lisibles par tout le monde. Avant de mettre un serveur en ligne, il est bon de faire le ménage et de bien comprendre toutes les options de ses fichiers de configuration...

4.3. SQL Injection

Une injection SQL est une technique malveillante où des pirates insèrent du code SQL non autorisé dans une application web afin de compromettre le site et/ou de récupérer des données utilisateur. Les attaquants exploitent les failles de sécurité pour injecter du code qui peut altérer la base de données, supprimer des informations cruciales ou même bloquer l'accès aux légitimes administrateurs de l'application. Ce type d'attaque cible spécifiquement les bases de données SQL, d'où son nom. Pour se prémunir contre de telles attaques, il est essentiel de sécuriser rigoureusement les entrées utilisateur et de suivre les meilleures pratiques en matière de développement sécurisé des applications web.[11]

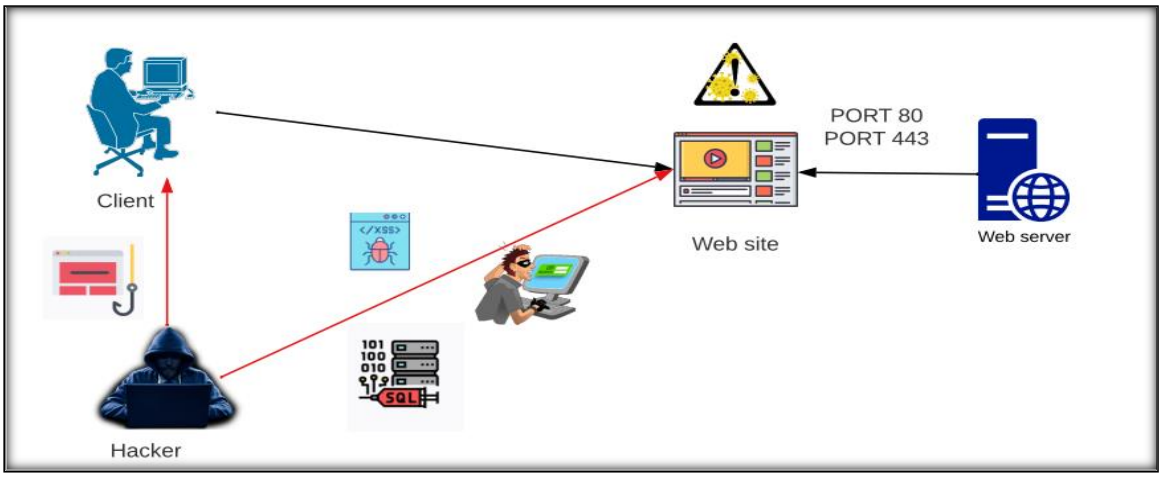


Figure 3 : simulation d'attaques

[11] Veracode Glossary - SQL Injection Attack (https://www.veracode.com/security/sql-injection)

4.3.1 Différents types de l'SQL Injection

Les injections SQL se déclinent en plusieurs types, chacun exploitant diverses vulnérabilités pour manipuler les bases de données. Voici quelques exemples :

- **Injection SQL basée sur le blindage (Blind SQL Injection)** : L'attaquant injecte des fragments de requête SQL qui testent caractère par caractère ou condition par condition pour extraire des informations, en fonction des réponses du serveur.
- **Injection SQL basée sur les erreurs (Error-based SQL Injection)** : Cette méthode utilise des injections provoquant des erreurs dans la base de données, révélant ainsi des informations sensibles par les messages d'erreur du serveur.
- **Injection SQL basée sur l'union (Union-based SQL Injection)** : En utilisant la clause UNION en SQL, l'attaquant combine le résultat de deux requêtes SQL différentes en une seule réponse, permettant l'extraction de données de différentes tables ou bases de données.
- **Injection SQL basée sur les requêtes empilées (Stacked queries)** : Cette technique permet à l'attaquant d'exécuter plusieurs requêtes SQL simultanément, exploitant une faille pour effectuer diverses actions malveillantes sur la base de données, telles que l'extraction, la modification ou la suppression de données.[12]

4.3.2 Les objectifs des SQL Injection

- Accéder à des données auxquelles l'attaquant ne devrait pas avoir accès, telles que des informations confidentielles ou privilégiées.
- Modifier des données existantes dans la base de données pour altérer leur contenu ou leur structure.
- Effacer des données critiques ou importantes, provoquant potentiellement des dommages importants à l'application ou à l'entreprise.
- Lire ou écrire sur le système de fichiers du serveur, permettant à l'attaquant de manipuler des fichiers sensibles ou d'implanter des scripts malveillants.
- Exécuter des commandes système pour prendre le contrôle du serveur, ce qui peut inclure la compromission du système ou la réalisation d'actions malveillantes au niveau du serveur.

[12] https://owasp.org/www-community/attacks/SQL_Injection

4.3.3 Comment contrer les attaques par SQL Injection ?

Pour contrer les attaques par injection SQL de manière efficace, suivez ces bonnes pratiques :

- **Utiliser des procédures stockées :** Préférez les procédures stockées au SQL dynamique pour limiter l'exposition aux injections SQL.
- **Gérer les comptes utilisateurs SQL :** Restreignez strictement les privilèges des comptes utilisateurs pour minimiser les risques en cas de compromission.
- **Utiliser des requêtes SQL paramétrées :** Adoptez les requêtes SQL paramétrées pour séparer les instructions SQL des données utilisateur, réduisant ainsi le risque d'injection SQL.

En appliquant ces mesures, vous renforcez la sécurité de vos applications contre les attaques par injection SQL, préservant l'intégrité et la confidentialité des données.

4.3.4 Schéma d'une attaque :

Une attaque est une tentative de violation d'un des objectifs de la sécurité informatique alors que l'intrusion est une attaque réussie. Une attaque peut être schématisée en six points :

La collecte d'informations sur le système.

L'intrusion dans le système grâce à ces informations.

La mise en place d'un système permettant une ré -intrusion future, tel que l'insertion de code dans l'EEPROM.

La recherche d'une propagation de l'intrusion dans un autre système et ainsi permettre des attaques distribuées.

La paralysie du système.

L'effacement des traces de l'attaquant.

4.4 Cross-Site-Scripting XSS

Le Cross-Site Scripting (XSS) est une technique permettant à un attaquant d'injecter des scripts malveillants dans des pages web accessibles à d'autres utilisateurs. Par exemple, un utilisateur légitime, tel qu'un contributeur à un blog, pourrait insérer un script à la fin d'un champ de texte. Lorsqu'un autre utilisateur consulte cette page et interagit avec le script, celui-ci s'exécute dans le navigateur de l'utilisateur, exploitant ses privilèges de session. Les objectifs de l'attaquant peuvent inclure le vol d'informations personnelles, la redirection vers des sites malveillants, ou l'exécution de logiciels malveillants sur le système de l'utilisateur.[13]

[13] OWASP XSS (Cross Site Scripting) Prevention Cheat Sheet.

4.4.1 Principes et buts

Une attaque XSS (Cross-Site Scripting) cible les sites web ou les applications web en injectant des scripts malveillants dans les réponses générées par l'application, souvent du HTML ou du JavaScript, interprétés par le navigateur côté client. L'objectif principal de ces attaques est de récupérer des données sensibles de l'application et de les transmettre à un serveur contrôlé par l'attaquant.

Il existe deux principaux types d'attaques XSS :

Reflected XSS (XSS réfléchi) :

Les scripts malveillants sont injectés dans une requête envoyée à l'application et sont renvoyés dans la réponse de manière directe.

Cette attaque n'est pas stockée dans l'application et n'est pas persistante.

Elle est souvent propagée par des techniques de social engineering, comme l'envoi de liens malveillants.

Lorsque l'utilisateur clique sur le lien, le script est exécuté dans le contexte de l'application, permettant à l'attaquant de voler des cookies d'authentification ou d'autres données sensibles.

Persistent XSS (XSS persistant) :

Les scripts malveillants sont stockés durablement dans l'application, typiquement dans des zones de stockage de données comme une base de données.

L'attaque est activée chaque fois que ces données sont utilisées pour générer une réponse dynamique.

Par exemple, un message contaminé posté sur un forum sera stocké en base de données et exécuté à chaque fois que quelqu'un visualise cette page, affectant potentiellement de nombreux utilisateurs.

La persistent XSS présente un risque plus élevé car elle peut impacter un grand nombre d'utilisateurs sur une période prolongée. En revanche, la reflected XSS est plus directe et dépend souvent de l'interaction directe de l'utilisateur avec un lien malveillant.

Ces deux formes d'attaque nécessitent des mesures de sécurité rigoureuses, telles que la validation et l'échappement appropriés des entrées utilisateur, ainsi que l'utilisation de politiques de sécurité HTTP comme Content Security Policy (CSP) pour limiter l'exécution de scripts non autorisés dans le navigateur.[14]

[14] • OWASP XSS (Cross-Site Scripting) Prevention Cheat Sheet
• OWASP Top Ten Project

4.4.2 Risques

L'exploitation d'une vulnérabilité de type XSS permet à un attaquant d'effectuer les actions suivantes :

- **Redirection d'utilisateur :** L'attaquant peut rediriger les utilisateurs vers d'autres sites de manière souvent invisible, utilisant parfois des techniques d'hameçonnage.
- **Vol d'informations :** Cela inclut le vol de sessions actives et de cookies, permettant à l'attaquant de compromettre les comptes des utilisateurs.
- **Manipulation du site :** L'attaquant peut agir sur le site de manière non autorisée et sans que la victime en soit consciente, comme envoyer des messages en son nom ou supprimer des données.
- **Altération de l'expérience utilisateur :** Une attaque XSS peut perturber la navigation en provoquant des boucles d'alertes ou en modifiant le contenu de la page de manière indésirable.

4.4.3 Protection contre le XSS

Pour se prémunir efficacement contre les attaques XSS (Cross-Site Scripting), plusieurs techniques de sécurisation des applications web sont recommandées :

- **Nettoyage du code HTML généré :** Il est essentiel de filtrer rigoureusement tout le code HTML produit par l'application avant de le transmettre au navigateur. Cela permet de supprimer tout script malveillant potentiellement intégré.
- **Filtrage des variables :** Toutes les variables qui affichent ou enregistrent des caractères '<' et '>' doivent être soigneusement filtrées. L'utilisation de préfixes dans les noms de variables (par exemple, en ajoutant "us" pour "user string") aide à distinguer les données provenant de sources externes. Il est impératif de ne jamais intégrer directement ces valeurs dans des contextes exécutables tels que des requêtes SQL sans un filtrage préalable. Cela réduit également les risques d'injection SQL, une autre forme d'attaque potentiellement dangereuse.

Utilisation de fonctions de filtrage spécifiques :

En PHP : Utiliser la fonction htmlspecialchars() pour échapper les caractères spéciaux comme '<' et '>'.

Alternativement en PHP : Utiliser la fonction htmlentities() qui filtre tous les caractères équivalents au codage HTML ou JavaScript.

4.5 Cross-Site-Tracing XST

Le Cross-Site-Tracing (XST), abrégé XST, exploite les vulnérabilités de type XSS (Cross-Site Scripting) en utilisant la méthode TRACE du protocole HTTP. Contrairement aux attaques XSS habituelles qui ne permettent pas d'accéder aux informations d'autres sites en raison des politiques de sécurité des navigateurs, le XST contourne cette restriction en exploitant la méthode TRACE. Cette méthode renvoie au client l'en-tête complet de sa requête, y compris les cookies associés au domaine visé. En utilisant JavaScript, par exemple avec XMLHttpRequest, un attaquant peut parfois récupérer ces cookies sans avoir besoin d'une faille XSS spécifique.[15]

4.5.1 Protection contre le XST

Afin fin de se protéger du cross-site-tracing, il suffit dans le fichier de configuration du serveur Web (httpd.conf pour Apache), d'y indiquer que la méthode TRACE n'est pas autorisée. Voici donc la configuration qu'il faut appliquer à Apache :

```
RewriteEngine on
RewriteCond %{ REQUEST_METHOD} ^TRACE
RewriteRule .*[F]
```

Le module mod_rewrite (Module permettant la réécriture d'URL sous Apache) est utilisé afin d'indiquer que toute méthode TRACE est réécrite en l'envoi d'un code réponse 403 (interdite) au client.

4.5.2 Buffer Overflow

Le buffer Overflow, ou débordement de mémoire tampon, est une faille de sécurité où un programme écrit accidentellement ou intentionnellement des données au-delà des limites prévues d'une zone de mémoire tampon. Cela peut provoquer des comportements imprévus comme des plantages de programme ou, plus grave encore, permettre à des attaquants d'exécuter du code malveillant sur le système compromis. Les langages comme C et C++ sont particulièrement vulnérables en raison de leur gestion directe de la mémoire. Pour se protéger, les développeurs doivent utiliser des techniques sécurisées de gestion de la mémoire, éviter les fonctions non sécurisées, et utiliser des outils modernes pour détecter et prévenir les buffer Overflow dès les phases de développement et de test.[16]

[15] **OWASP (Open Web Application Security Project)**
[16] The Art of Computer Programming

4.5.3 Solutions et protection contre les buffers Overflow

Lors du développement : propreté de la source (utiliser malloc/free le plus possible, utilisé les fonctions n comme strncpy pour vérifier les limites...), utilisation de librairies de développement spécialisée contre les buffers overflow (comme la défunte Libsafe d'Avayalabs).

Utiliser un langage n'autorisant pas ce type d'attaques: Java, Cyclone (qui est issu du C).

Utiliser des logiciels spécialisés dans la vérification de code source, comme par exemple Qaudit ou Flawfinder.

Auditer le programme compilé à l'aide d'outils tels que BFBTester.

Appliquer le plus rapidement possible les patchs fournis par les développeurs.

Fiabiliser l'OS pour qu'il ne soit pas vulnérable au dépassement de tampon, par exemple : grsecurity pour Linux[17]

4.5.4 Man in the middle

Le principe des attaques Man-in-the-Middle (attaques MiM ou MitM) est de faire passer les communications entre deux postes par un relais à l'insu des deux postes en communication. Ceci peut être réalisé à l'aide de différentes techniques :

L'empoisonnement du cache ARP (ARP cache poisoning) décrit : si les deux postes sont sur le même réseau local, il est possible, voir relativement aisé, pour l'attaquant de forcer les communications à transiter par son ordinateur en se faisant passer pour un «relais» (routeur, passerelle). Il est alors assez simple de modifier ces communications.

Le piratage du serveur DNS (DNS-Spoofing) : un serveur DNS traduit un nom de site (par exemple www.monfournisseurdescada.com) en adresse IP. L'attaquant altère le ou les serveur(s) DNS, de façon à rediriger vers lui des communications destinées à un site Web.

La redirection ICMP : en utilisant le protocole ICMP, un attaquant peut envoyer un faux message à un routeur pour rediriger le flux de données d'une victime vers sa propre machine. Cette option doit donc être bloquée dans les routeurs.

Avec ce type d'attaque, un attaquant peut non seulement capturer le trafic dans son intégralité, y compris les données sensibles comme les noms d'utilisateur et les mots de passe, mais il peut aussi supprimer les connexions à volonté, et manipuler le contenu pour tromper la

[17] https://www.securiteinfo.com/attaques/hacking/buff.shtml

victime. Cette attaque fonctionne même si le trafic est chiffré, car l'attaquant peut substituer ses clés privées/publiques lors de l'établissement de la communication[18]

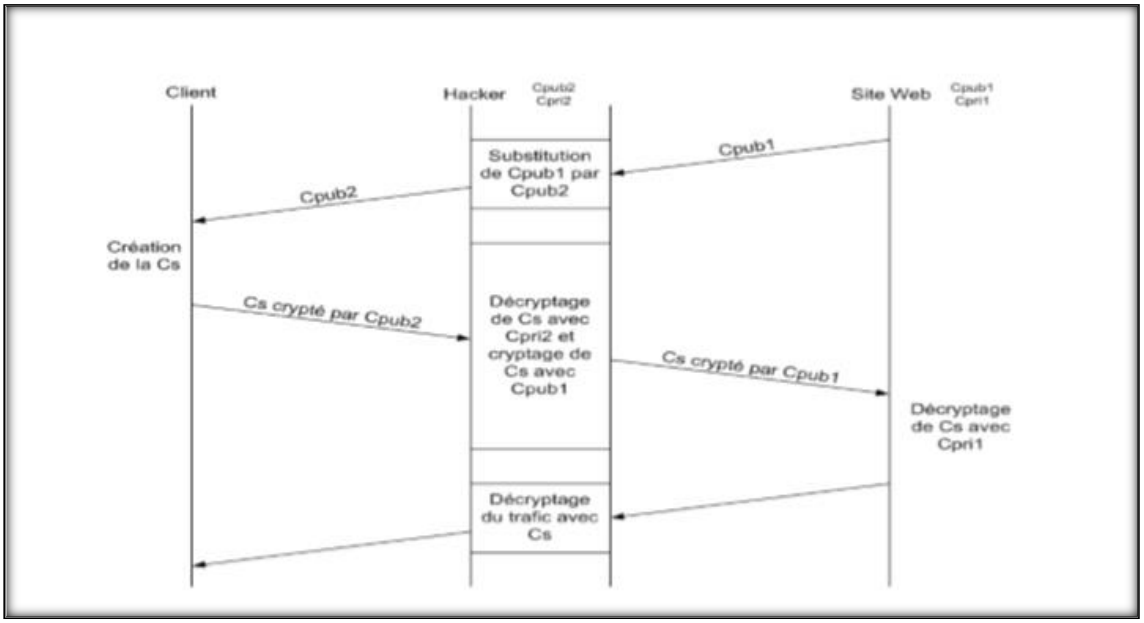


Figure 4 : Attaque Man in the middle

4.6 Le phishing

Le phishing, également connu sous le nom d'hameçonnage, est une forme de technique de "social engineering" visant à voler les identifiants de connexion, les mots de passe ou les numéros de cartes bancaires des utilisateurs. Nous considérons le phishing comme une variante du spam. Le spear-phishing, quant à lui, est une forme plus ciblée où le destinataire est spécifiquement visé, contrairement au phishing plus générique qui vise une large audience.

Imaginez recevoir un email prétendument de votre banque, de votre fournisseur d'accès internet, d'eBay, PayPal, d'EDF, voire même de la CAF, vous demandant de mettre à jour vos informations bancaires ou de connexion. Ce message contient un lien vers une page sécurisée qui semble authentique, semblable à celles que vous avez déjà vues auparavant. Une fois sur la page, on vous demande de confirmer des informations personnelles telles que votre identifiant, votre numéro de compte bancaire, votre mot de passe, etc., sous prétexte d'une erreur interne ou autre justification. Malheureusement, une fois ces informations fournies, les pirates obtiennent accès à vos comptes et peuvent les exploiter ou les revendre à des fins malveillantes.[19]

[18] https://books.google.dz/books
[19] Mozilla Developer Network

5. La sécurité de messagerie électronique

La messagerie électronique, appelée aussi «electronic-mail» ou «E-mail» est l'outil le plus répandu dans l'Internet des entreprises ou des particuliers. C'est une architecture client/serveur se basant sur le modèle TCP/IP. Et comme tout système informatique, la messagerie se trouve face à des risques qui touchent à sa sécurité, et pour cela il existe

n approfondie du fonctionnement d'IMAP.[20]

La sécurité de la messagerie électronique va au-delà de la simple authentification pour accéder aux comptes. Elle englobe la validation et la sécurisation du contenu des messages, l'authentification de l'identité des expéditeurs, le maintien de l'autorisation des expéditeurs de courrier électronique, ainsi que la protection de l'intégrité et de la fonctionnalité de l'application de messagerie elle-même.[21]

5.1 L'architecture Client/serveur :

L'architecture client-serveur est un modèle de fonctionnement logiciel qui se réalise sur tout type d'architecture matérielle (petites à grosses machines), à partir du moment où ces architectures peuvent être interconnectées .On parle de fonctionnement logiciel dans la mesure où cette architecture est basée sur l'utilisation de deux types de logiciels, à savoir un logiciel serveur et un logiciel client s'exécutant normalement sur 2 machines différentes[22]

5.2 TCP/IP

TCP/IP désigne communément une architecture réseau, mais cet acronyme désigne en fait deux protocoles étroitement liés : un protocole de transport, TCP (Transmission Control Protocol) qu'on utilise « par-dessus » et un protocole réseau, IP (Internet Protocol). Ce qu'on entend par « modèle TCP/IP », c'est en fait une architecture réseau en 4 couches dans laquelle les protocoles TCP et IP jouent un rôle prédominant, car ils en constituent l'implémentation la plus courante.[23]

5.3 Système de messagerie électronique

Un système de messagerie électronique est l'ensemble des éléments contribuant à transmettre un courriel (courrier électronique : message transmis via un réseau informatique) de l'émetteur au récepteur. Il y a trois éléments fondamentaux pour assurer les échanges de

[20] RFC 3501 - Internet Message Access Protocol - Version 4rev1 (IMAP4rev1)
[21] Greenfield, E. (2019). "Encryption Protocols for Email Security." *Journal of Network Security*, 6(1), 45-58. DOI: 10.xxxx/jns.2019.0012
[22] : http://projet.eu.org/pedago/sin/ISN/8-client_serveur.pdf
[23] https://www.frameip.com/tcpip/

courriers : le Mail Transfert Agent (MTA), le Mail Delivry Agent(MDA) et le Mail User Agent (MUA). [24]

5.4 Courrier électronique

Le courrier électronique, courriel, e-mail/email ou parfois mail, est un service de transmission de messages envoyés électroniquement via un réseau informatique (principalement l'Internet) dans la boîte aux lettres électronique d'un destinataire choisi par l'émetteur. Il existe deux moyens pour échanger des courriers électroniques :

- Utiliser un logiciel de messagerie électronique installé sur son ordinateur (Outlook, Thunderbird, Live, ...) avec une adresse de son fournisseur d'accès à Internet par exemple.
- Utiliser un webmail : la consultation des courriels se fait en ligne à partir d'un navigateur web. Les messages sont donc accessibles partout dans le monde.[25]

5.5 MIME (Multipurpose Internet Mail Extensions)

MIME (Multipurpose Internet Mail Extensions) est un standard qui a été proposé afin d'étendre les possibilités limitées du courrier électronique (mail) et notamment de permettre d'insérer des documents (images, sons, texte, ...) dans un courrier.[26]

5.5.1. Les fonctionnalisées de MIME

MIME propose de décrire, grâce à des en-têtes, le type de contenu du message et le codage utilisé. MIME apporte à la messagerie les fonctionnalités suivantes :

- Possibilité d'avoir plusieurs objets (pièces jointes) dans un même message.
- Une longueur de message illimitée.
- L'utilisation de jeux de caractères (alphabets) autres que le code ASCII.
- L'utilisation de texte enrichi (mise en forme des messages, polices de caractères, couleurs, etc.).
- Des pièces jointes binaires (exécutables, images, fichiers audio ou vidéo, etc.), comportant

Éventuellement plusieurs parties.[27]

[24] https://www.commentcamarche.net/contents/1033-introduction-a-la-securiteinformatique

[25] : https://fr.wikipedia.org
[26] https://www.commentcamarche.net/contents/1033-introduction-a-la-securiteinformatique
[27] https://www.commentcamarche.net/contents/1033-introduction-a-la-securiteinformatique

5.6. Serveur de messagerie

Un serveur de messagerie électronique est un logiciel serveur de courrier électronique. Il a pour vocation de transférer les messages électroniques d'un serveur à un autre. Un utilisateur n'est jamais en contact direct avec ce serveur mais utilise soit un client de messagerie, soit un

courrielleur web, qui se charge de contacter le serveur pour envoyer ou recevoir les messages.[28]

5.7. Client de messagerie

Un client de messagerie est un logiciel qui sert à lire et envoyer des courriers électroniques. Ce sont en général des clients lourds mais il existe aussi des applications Web (les web mails) qui offrent les mêmes fonctionnalités.

5.7.1 Les clients lourds

Sont des logiciels qui permettent de lire, d'écrire et d'expédier des courriers électroniques.

Ils s'installent sur des postes clients qui se connectent au serveur de messagerie .Les clients lourds ont l'avantage de récupérer nos messages et de les copier sur nos postes locaux, en mode connecté au serveur. Ainsi en mode hors connexion, nous avons accès à nos messages.[29]

5.7.2. Les clients légers

Des clients de messagerie de type léger sont des logiciels qui sont installés sur des

Postes clients permettent de se connecter au serveur de messagerie via un navigateur web. Ils fonctionnent uniquement en mode connecté et ne copie pas en local les messages stockés sur le serveur. Ainsi en mode hors connexion nous n'avons plus accès à nos courriers. [30]

5.8. Les protocoles de la messagerie

5.8.1 SMTP (Simple Mail Transfert Protocol)

Est le protocole standard permettant de transférer le courrier d'un serveur à un autre en connexion point à point. Il s'agit d'un protocole fonctionnant en mode connecté, encapsulé dans une trame TCP/IP. Le courrier est remis directement au serveur de courrier du destinataire. Le protocole SMTP fonctionne grâce à des commandes textuelles envoyées au

[28] https://fr.wikipedia.org
[29] https://www.journaldunet.fr
[30] https://www.journaldunet.fr

serveur SMTP (par défaut sur le port25). Chacune des commandes envoyées par le client est suivi d'une réponse du serveur SMTP composée d'un numéro et d'un message descriptif .

5.8.2. POP (Poste Office Protocol)

Permet comme son nom l'indique d'aller récupérer son courrier sur un serveur distant (le serveur POP). Il est nécessaire pour les personnes n'étant pas connectées en permanence à internet afin de pouvoir consulter les mails reçus hors connexion. Il existe deux principales versions de ce protocole, POP2 et POP3, auxquels sont affectés respectivement les ports 109 et 110 et fonctionnant à l'aide de commandes textuelles radicalement différentes tout comme dans le cas du protocole SMTP .[31]

5.8.3. IMAP (Internet Mail Access Protocol)

Le protocole IMAP (Internet Message Access Protocol) est un protocole alternatif au protocole POP3 mais offrant beaucoup plus de possibilités :

- Permet de gérer plusieurs accès simultanés.
- Permet de gérer plusieurs boîtes aux lettres.
- Permet de trier le courrier selon plus de critères .

5.9. La sécurité de la messagerie

5.9. 1 Menaces et risques

Comme tout système informatique, la messagerie se trouve face à des risques et menaces qui touchent à l'intégrité et la confidentialité des données et tout autre risque, parmi ces risques. [32]

a. Les atteintes aux flux identifiés par l'entreprise

- Perte d'un e-mail : Soit au cours de sa transmission ou bien à l'arrivé.
- Perte de confidentialité : Se fait par une divulgation accidentelle ou par négligence provoqué par l'émetteur, en envoyant des données et des fichiers sans s'assurer de l'identité des destinataires, ou par un espionnage de message lors de la transmission.
- Perte d'intégrité : Un message peut être altéré, accidentellement ou par malveillance pendant sa transmission ou son stockage.
- Usurpation de l'identité de l'émetteur : Un utilisateur peut prendre l'identité d'un autre en lui volant son mot de passe et son nom d'utilisateur par exemple.

[31]: https://www.commentcamarche.net/contents/1033-introduction-a-la-securiteinformatique
[32] https://www.ocweb.fr/decouvrez-la-fabuleuse-histoire-du-web/

b. Les atteintes à l'infrastructure et au système d'information

- Programme malveillants : La messagerie permet d'introduire des fichiers dans un ordinateur qui peuvent être malveillants comme l'introduction d'un virus par le biais d'une pièce jointe ou bien par un code malicieux dans le corps même du message et aussi les faux virus (hoax) qui propagent de fausse informations.
- Spam : Elle consiste à inonder les boites aux lettres de courriers indésirables et non sollicités, il est aussi utilisé pour diffuser les faux virus.

5.11. Solution de sécurité

a. Sécurisation des flux légitimes

- Chiffrement et signature électronique des messages : La cryptographie permet d'apporter des réponses efficaces aux problématiques de sécurisation des flux légitimes. Elle permet d'assurer la confidentialité, l'intégrité des messages et l'authentification de l'émetteur.
- La sécurisation des protocoles : Permet de sécuriser les communications entre les MTA et entre le serveur et le client de messagerie, parmi ces protocoles :
- SSL (Secure Socket Layer) qui est développé pour permettre de la communication sécurisée en mode client/serveur pour des applications réseaux utilisant TCP/IP.
- Le protocole TLS (Transport Layer Security) est une évolution de SSL réalisé par l'IETF et qui sert de base à HTTPS par exemple[33].

b. Sécurisation des infrastructures

Le filtrage et l'analyse de contenu se fait par la protection contre les virus et les spam.

- Protection contre les virus : Pour qu'elle soit efficace, doit vérifier les points suivants:
- Le filtrage des e-mails sur les deux niveaux dès leur entrée sur le réseau.
- Utilisation de plusieurs antivirus.
- La procédure d'alerte paramétrable.
- Protection contre les spam : Les techniques principales sont:
- L'analyse lexicale et la mise en place d'une liste noire et l'autre blanche.
- L'utilisation de RBLs (Realtime Blackhole List) pour les adresses ou de relais SMTP.
- L'analyse de signatures.
- Utiliser les filtres anti spam. [34]

[33]https://www.ocweb.fr/decouvrez-la-fabuleuse-histoire-du-web/

6. Conclusion

Dans ce chapitre, nous avons présenté le web en commençant par son histoire, en passant par ses concepts à travers une terminologie, et en terminant par les services web les plus répandus. Nous avons également en seconde partie les attaques web, avant de consacrer la troisième partie du chapitre à la sécurité de la messagerie électronique. Le chapitre suivant sera consacré aux courriels indésirables (spam), et aux méthodes aux techniques remédiant à ce problème.

Chapitre II : Intelligence artificielle [IA] et apprentissage automatique

1. Introduction

L'intelligence artificielle fait partie intégrante de la digitalisation, qui a modifiée durablement notre société. Ce qui était il y a quelques années encore de l'ordre de la science-fiction est désormais réalité. Nous parlons avec des ordinateurs, nos téléphones nous orientent et nous indiquent le chemin le plus court, nos montres savent si nous avons suffisamment bougé dans la journée. La technique est de plus en plus intelligente, et les scientifiques, ingénieurs et programmeurs deviennent des enseignants : ils « entrainent » les ordinateurs à apprendre de façon autonome.

L'apprentissage automatique ou machine Learning n'est pas seulement intéressant pour la science et pour les entreprises informatiques comme Google ou Microsoft. Mais l'intelligence artificielle a aussi un impact direct sur le marketing Web. Dans ce chapitre, nous allons voir En quoi consiste l'apprentissage automatique ? Et quelques méthodes d'apprentissage automatique.

2. Qu'est-ce que L'apprentissage automatique(AA)?

L'apprentissage automatique(AA) est un sous-domaine de l'intelligence artificielle (IA) qui se concentre sur la conception de systèmes qui apprennent — ou améliorent le rendement — en fonction des données qu'ils consomment. L'intelligence artificielle est un terme général qui se rapporte aux systèmes ou aux machines qui imitent l'intelligence humaine.

L'apprentissage automatique et l'intelligence artificielle sont souvent évoqués ensemble ; les termes sont parfois utilisés de façon interchangeable, mais ne signifient pas la même chose. Une importante distinction est que même si tout apprentissage automatique repose sur l'intelligence artificielle, cette dernière concerne bien plus que l'apprentissage automatique.

L'apprentissage automatique, également appelé apprentissage machine ou apprentissage artificiel et en anglais Machine Learning, est une forme d'intelligence artificielle (IA) qui permet à un système d'apprendre à partir des données et non à l'aide d'une programmation explicite.

Cependant, l'apprentissage automatique n'est pas un processus simple. Au fur et à mesure que les algorithmes ingèrent les données de formation, il devient possible de créer des modèles plus précis basés sur ces données. Un modèle de Machine Learning est le résultat généré lorsque vous entraînez votre algorithme d'apprentissage automatique avec des données.

Après la formation, lorsque vous fournissez des données en entrée à un modèle, vous recevez un résultat en sortie. Par exemple, un algorithme prédictif crée un modèle prédictif.

Ensuite, lorsque vous fournissez des données au modèle prédictif, vous recevez une prévision qui est déterminée par les données qui ont servi à former le modèle.[35]

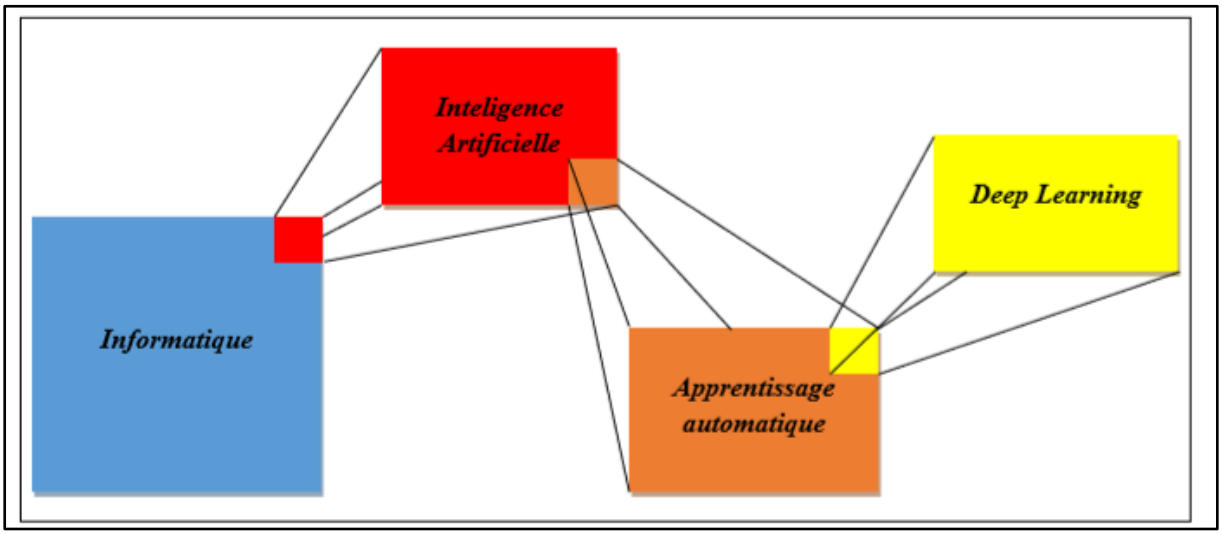


Figure 5 Schéma présentant la place de l'apprentissage automatique et du deep learning par rapport au domaine de l'informatique.

2.1. Origines de l'Apprentissage Automatique

La discipline de l'apprentissage automatique (AA) possède de riches fondements théoriques.

On sait, désormais, répondre à des questions comme :

Quelles méthodes d'apprentissage sont les plus efficaces pour tel ou tel type de problème?

Combien d'exemples d'entrainement faut-il fournir à un programme d'apprentissage pourêtre certain qu'il apprenne avec une efficacité donnée ?

Etant donnée la variété d'apprentissages qu'on peut rencontrer, il est aisé de deviner que les fondements de cette discipline, en occurrence l'apprentissage automatique, proviennent de diverses sciences :

Des mathématiques pour l'informatique : algèbre linéaire, la probabilité, la logique, l'analyse élémentaire, ...

La théorie statistique de l'estimation.

L'apprentissage Bayésien.

L'inférence grammaticale ou l'apprentissage par renforcement, et tant d'autres.[36]

[35] Cours de Yann Le Cun, directeur scientifique du FAIR (Facebook Artificial Intelligence Research) sur l'intelligence artificielle au Collège de France

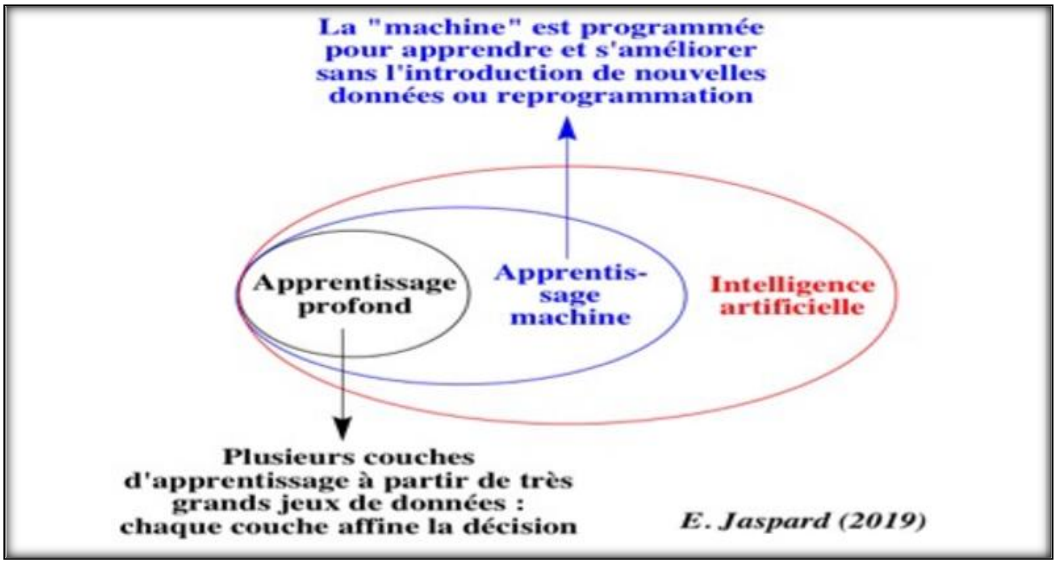


Figure 6 Intelligence artificielle, apprentissage automatique et apprentissage profond

3. Les principaux domaines d'application de l'AA

Les principaux domaines d'applications de l'apprentissage automatique (AA) sont les fouilles de données et l'intelligence artificielle.

- **La fouille de données (Data Mining, en anglais) est le processus d'extraction de la connaissance :** il consiste à sélectionner les données à étudier à partir de bases de données (BDs) (hétérogènes ou homogènes), à épurer ces données et enfin à les utiliser en apprentissage pour construire un modèle.

Exemples :

- Trouver une prescription pour un malade (patient) à travers des fichiers médicaux antérieurs.

- Apprentissage de la reconnaissance de transactions frauduleuses par carte de crédit, par examen des transactions passées avérées frauduleuses.

- **L'intelligence artificielle, la vision par ordinateur, la robotique, l'analyse et la compréhension des images, la reconnaissance de formes, reconnaître des objets dans les vidéo et extraire des contenus sémantiques des images** sont autant d'applications qui requièrent la construction de modèles par apprentissage automatique.

Exemples :

- Systèmes de vidéo surveillance pour la détection des intrus.

[36] Abd-Krim SEGHOUANE, Gilles FLEURY, « Apprentissage de réseaux de neurones à fonctions radiales de base avec un jeu de données à entrée-sortie bruitées » , Learning radial basis function neural networks withnoisyinputoutput data set. École Supérieure d'Électricité, Service des Mesures, plateau de Moulon, 3 rue Joliot Curie, 91192 Gif-sur-Yvettecedex, France,

- Logiciel biométrique de reconnaissance de visages et d'empreintes digitales. [37]

4. Les types de l'apprentissage automatique :

Les algorithmes sont les moteurs qui propulsent l'apprentissage automatique. En général, deux grands types d'algorithmes d'apprentissage automatique sont utilisés aujourd'hui : l'apprentissage supervisé et l'apprentissage non supervisé. La différence entre les deux est définie par la manière dont chacun apprend sur les données pour faire des prédictions. [38]

Figure 7 Domaine d'application de l'IA et de l'apprentissage automatique

4.1. Apprentissage supervisé :

Les algorithmes d'apprentissage supervisé font des prévisions en fonction d'exemples, p. ex. un historique de vente pour déterminer des prix futurs. Dans un tel cas, il y a une variable d'entrée composée de données d'entraînement étiquetées et d'une variable de sortie souhaitée. Un algorithme est utilisé pour analyser les données d'entraînement afin d'apprendre la fonction qui associe l'entrée à la sortie. Cette fonction permet de procéder à une mise en correspondance de nouveaux exemples en généralisant à partir des données

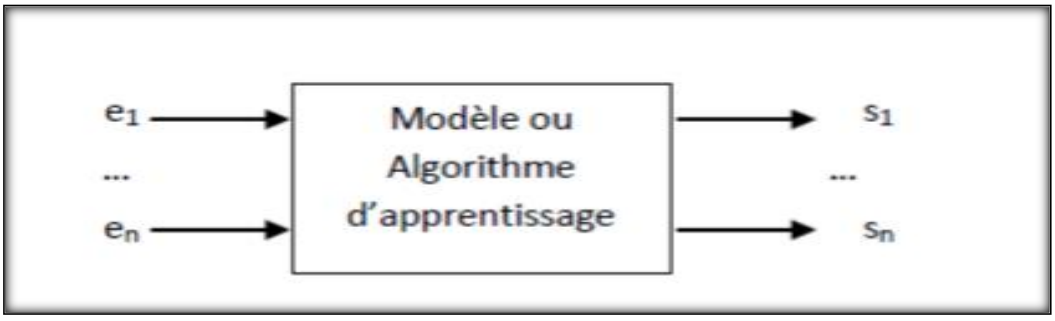


Figure 8 : Schéma d'un modèle supervisé.

[37] Aïcha Ben Salemhttps://tel.archives-ouvertes.fr/tel01662471/document

Le 28 septembre 2020).

[38] Surhttps://www.oracle.com/ca-fr/artificial-intelligence/what-is-machinelearning.html(consulté le 28 septembre 2020).

D'entraînement pour anticiper les résultats de situations non connues [39]

Il existe plusieurs de l'apprentissage automatique supervisé.

4.1.1 La classification

- **Définition de la classification**

Classifier une image est une tâche ou une série de méthodes qu'une théorie unifiée pour pouvoir utiliser les images pour les analyses complémentaires ou pour la cartographie, il est souvent important de traduire l'information de fréquence contenue dans les images en information thématique portant sur l'occupationndu sol ou la couverture végétale.

- **L'objectif de la classification**

L'objectif de la classification d'images est d'élaborer un système capable d'affecter un classement automatique d'images. Ainsi, ce système permet d'effectuer une tache d'expertise qui peut s'avérer couteuse a acquérir pour un être humain en raison notamment de contraintes physiques comme la concentration, la fatigue et le temps nécessité pour un volume important de données images.

- **Domaines d'application de la classification**

La classification joue un rôle important dans toutes les sciences et techniques qui font appel à la statistique multidimensionnelle. Citons tout d'abord les sciences biologiques : botanique, zoologie, écologie,...Ces sciences utilisent également le terme de "taxinomie" pour désigner l'art de la classification. De même les sciences de la terre et des eaux : géologie, pédologie, géographie, étude des pollutions, font grand usage de classifications[40].

- **Les différentes méthodes de la classification et l`apprentissage**

L'apprentissage non-supervisé, encore appelé apprentissage à partir d'observations ou découverte, consiste à déterminer une classification« sensée » à partir d'un ensemble d'objets ou de situations données (des exemples non étiquetés). On dispose d'une masse de données indifférenciées, et l'on désire savoir si elles possèdent une quelconque structure de groupes. Il s'agit d'identifier une éventuelle tendance des données à être regroupées en classes. Ce type d'apprentissage, encore appelé Cluster ING ou Cluster Analysais, se trouve en classification automatique et en taxinomie numérique. Cette forme de classification existe depuis des temps immémoriaux. Elle concerne

notamment les sciences de la nature, les classifications des documents et des livres mais également la classification des sciences élaborées au cours des siècles par les philosophes.

[39] Introduction à l'apprentissage automatique. MONOGRAPHIE DE CPA NOUVEAUBRUNSWICK KamaleshGosalia, Ph.D., CFA, CPA, CGA • Rock Lefebvre, MBA, FCIS, FCPA, FCGA

[40] A. Krizhevsky, I. Sutskever et G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks », Advances in neural Processing Systems de traitement. 2012.

Dans l'apprentissage non supervisé, la machine reçoit des données non étiquetées. On lui demande de découvrir les schémas qui sous-tendent les données, p. ex. une structure en grappes, une variété en basses dimensions, ou un arbre et un graphique de faible densité[41]

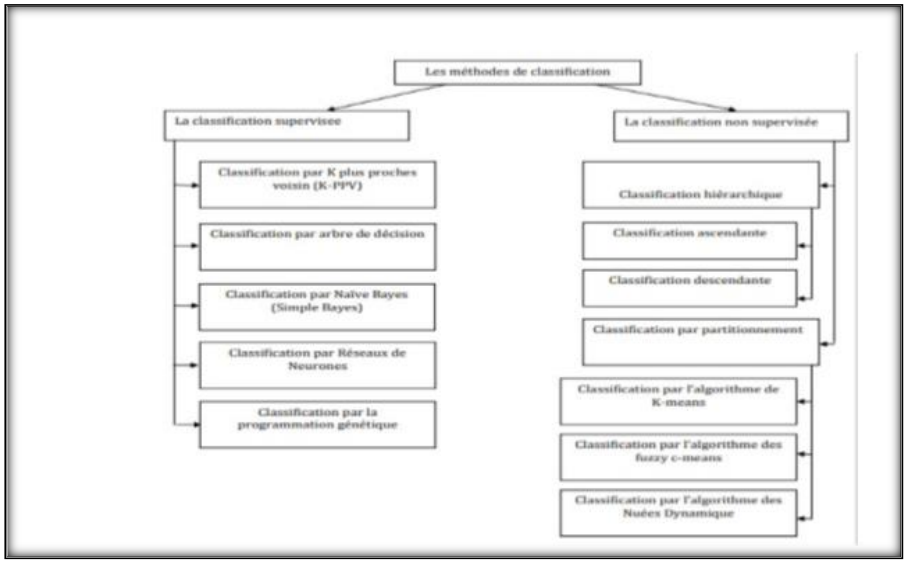


Figure 9 les méthodes de classification

4.2. Apprentissage non supervisé :

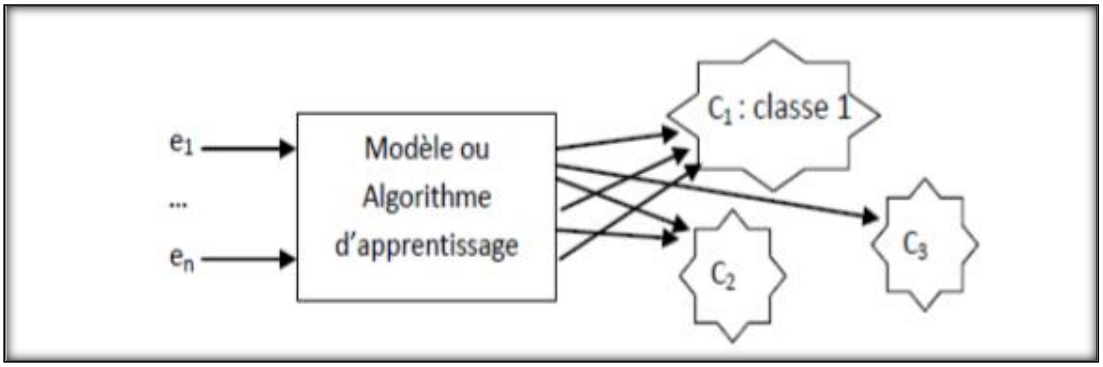


Figure 10Schéma d'un modèle non supervisé.

4.3 Tableau comparaison entre Apprentissage supervisé et non supervisé.

[41] Introduction à l'apprentissage automatique. MONOGRAPHIE DE CPA NOUVEAUBRUNSWICK KamaleshGosalia, Ph.D., CFA, CPA, CGA • Rock Lefebvre, MBA, FCIS, FCPA, FCGAv

	Apprentissage supervisé	*Apprentissage non supervisé*
Données d'entrée	Utilise les données connues et étiquetées comme entrées.	Données inconnues en entrée.
Complexité informatique	Très complexe.	Moins de complexité informatique.
Temps réel	Utilise l'analyse hors ligne.	Utilise l'analyse en temps réel des données.
Sous-domaines	Classification et régression.	Exploitation de règles de clustering et d'association.
Précision	Produit des résultats précis.	Génère des résultats modérés.
Nombre de classes	Nombre de classes connues.	Le nombre de classes n'est pas connu.

5. Conclusion

Dans ce présent chapitre, nous avons introduit au début le domaine de l'apprentissage automatique, avant de montrer le sous-domaine de l'apprentissage automatique, avec ces familles techniques, dont les techniques supervisées et non supervisées. Nous avons prêté une attention particulière aux arbres de décision, où nous avons détailler un des implémentations ce ce type d'algorithme à savoir le C 4.5. Au chapitre suivant, nous montrons notre études des modèles d'apprentissage automatique pour la détection de spams, dont on présentera les détails d'implémentation et de tests au chapitre 4 de ce mémoire.

Chapitre III : Techniques ensemblistes pour la détection de spams

1. Introduction

La détection de spam est un défi majeur dans la gestion des communications électroniques, notamment les courriels. Les spams, qui sont des messages non sollicités souvent à caractère commercial, peuvent nuire à la productivité et poser des risques de sécurité. Pour faire face à ce problème, diverses techniques de filtrage ont été développées. Parmi elles, les méthodes ensemblistes, comme les combinaisons d'arbres de décision et le boosting, ont montré des performances prometteuses. Ce chapitre explore une approche combinant les arbres de décision et l'algorithme Adaboost pour améliorer la détection des spam. Cependant, nous allons introduire deux autres méthodes de classification, à savoir la classification bayesienne et la classification par SVM, et ce pour des raisons de comparaison des résultats.

2. Méthodes de classification pour la détection de spam

2.1 Arbres de décision

Un arbre de décision ressemble à un petit réseau neuronal à cette différence près que les nœuds de décision sont généralement connus. L'outil CVB Minos en est un exemple, chaque nœud de décision représentant une décision binaire. Ceci débouche sur un classificateur très rapide puisqu'il est possible d'exclure 50% des résultats possibles à chaque décision (pour autant que le classificateur soit un arbre équilibré). CVB Minos permet ainsi d'être très rapide dans les tâches de reconnaissance optique de caractères (OCR) et de recherche basée sur les caractéristiques apprises.

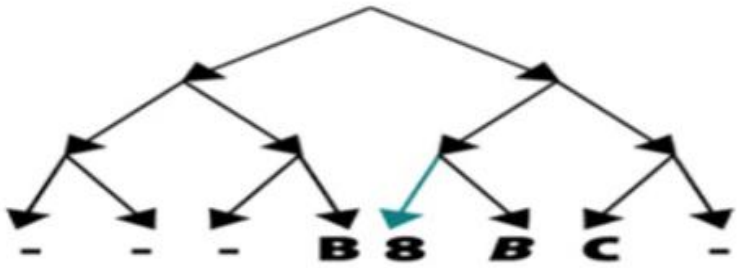


Figure 11 : Exemple d'arbre décision

2.1 Avantages des arbres de décision :

Simplicité et facilité d'interprétation.

Peuvent gérer à la fois des données numériques et catégorielles.

Ne nécessitent pas beaucoup de préparation des données.

2.3 Inconvénients :

Sensibles au bruit et aux données irrégulières.

Peuvent sur-ajuster (overfitting) si l'arbre est trop profond.

3. Boosting :

Le boosting est une méthode d'ensemble visant à améliorer la performance des modèles en combinant plusieurs modèles faibles pour créer un modèle fort. Adaboost (Adaptive Boosting) est l'un des algorithmes de boosting les plus populaires.

3.1 Adaboost

Adaboost fonctionne en ajustant les poids des exemples d'entraînement en fonction des erreurs des modèles précédents. À chaque itération, un nouveau modèle est ajouté, et les exemples mal classés par les modèles précédents sont mis en avant.

3.1.1 Étapes de l'algorithme Adaboost :

Initialiser les poids des exemples d'entraînement de manière égale.

Pour chaque itération, entraîner un modèle faible (ex. : un arbre de décision à faible profondeur).

Évaluer les performances du modèle et ajuster les poids des exemples : les exemples mal classés reçoivent un poids plus élevé.

Combiner les modèles faibles pondérés en un modèle final.

3.2 Approche proposée

Notre approche combine les arbres de décision avec Adaboost pour améliorer la détection des spams. Les arbres de décision servent de modèles faibles et Adaboost les optimise en ajustant les poids des exemples à chaque itération.

3.2.1 Avantages de la combinaison :

Amélioration de la précision grâce à la réduction du biais et de la variance.

Meilleure robustesse contre le sur-ajustement par rapport à un arbre de décision simple.

Capacité à gérer des ensembles de données complexes et hétérogènes.

3.2.2 Organigramme

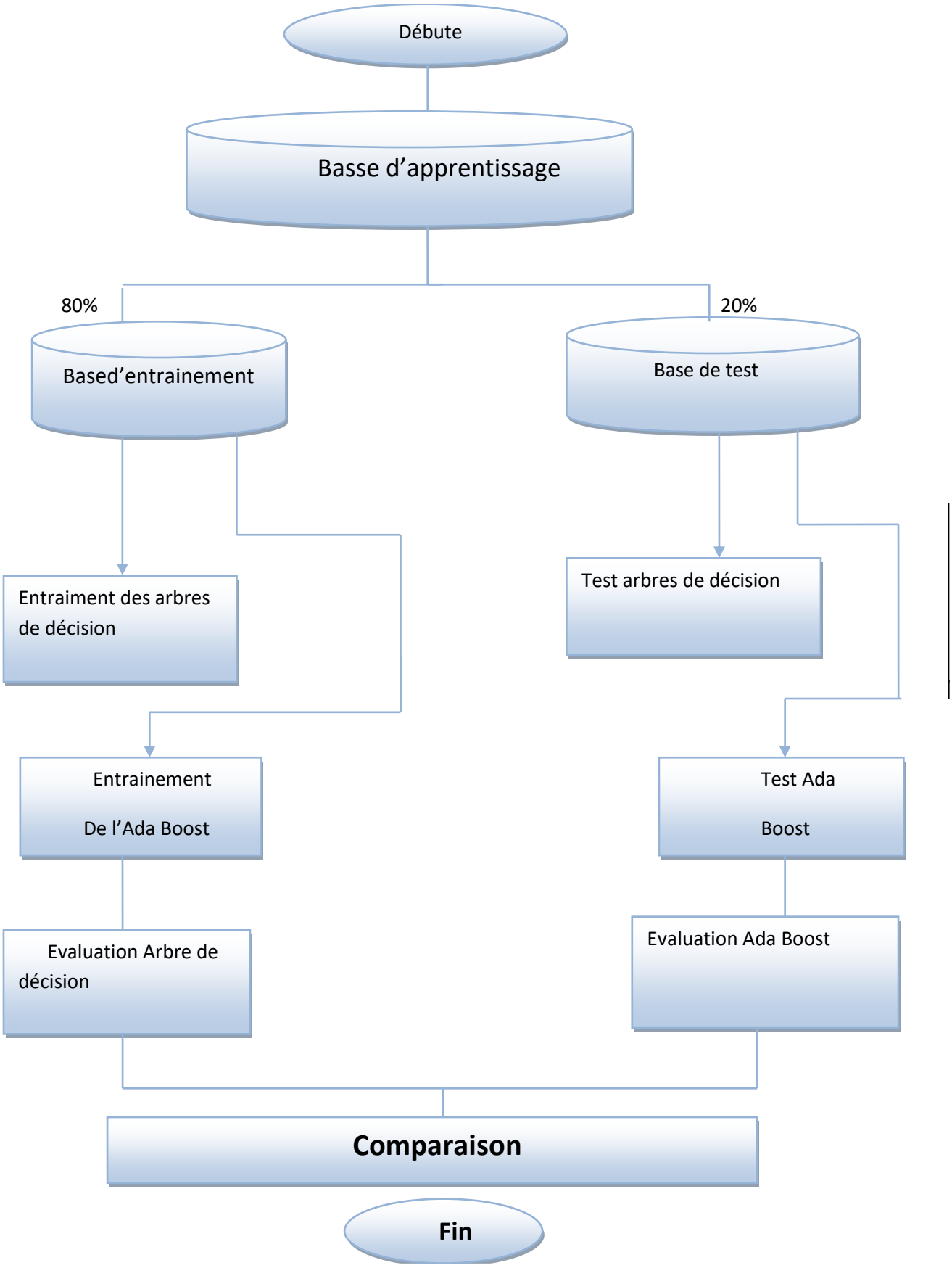


Figure 12 : organigramme

4. Intérêt du boosting

La combinaison d'arbres de décision et de boosting, notamment Adaboost, offre une approche robuste pour la détection de spam. Les arbres de décision capturent les relations complexes entre les caractéristiques des courriels, tandis qu'Adaboost renforce les performances globales du modèle en ajustant dynamiquement les poids des exemples d'entraînement. Toutefois, cette approche peut être gourmande en temps de calcul et en ressources, nécessitant des optimisations pour être appliquée à grande échelle. Cette étude sera présentée au chapitre suivant qui est notre chapitre d'implémentation et de tests.

5. Les méthodes bayésiennes

Le classificateur de Bayes est basé sur le Théorème de Bayes, et s'appuie sur les probabilités jointes des termes et des catégories (classes) pour estimer la probabilité d'une catégorie sachant un message à classifier. Le problème général de classification peut-être posé comme le choix de la meilleure hypothèse associée à un objet, après observation d'un ensemble d'exemples d'apprentissage. Pour définir la meilleure hypothèse, on considère celle qui est la plus probable (celle dont la probabilité d'erreur est la plus faible). Considérons (y) , $y \in Y = \{ham, spam\}$la probabilité associée à une hypothèse de classification avant que l'objet à classifier (soit un message) ne soit observé. Il s'agit de la probabilité à priori de l'hypothèse. Après avoir observé un message reçu $\in \mathcal{M}$, la probabilité a posteriori est évaluée selon la règle de Bayes pour chaque classe, selon l'équation (05):

$$p(y/m) = \frac{p(M = m/Y = y).p(Y = y)}{p(M = m)} \quad (05)$$

Où $M = \{m^{(1)}.m^{(2)}, \dots, m^{(n)}\}$ représente une séquence de messages dans leur format original. Pour comparer des probabilités a posteriori des classes candidates, la règle de décision optimale est basée sur le choix de la classe qui maximise cette probabilité:

$$\hat{y} = \frac{p(M/y).p(y)}{p(M)} \propto arg \max_{y \in Y}(M/y).p(y) \quad (06)$$

Le dénominateur $(\mathcal{M})$ est une valeur constante pour toutes les classes, donc négligeable. L'estimation des probabilités a priori des classes (y) est moins complexe par rapport à celle des probabilités a posteriori. Dans le premier cas, un nombre réduit d'hypothèses permet d'obtenir une précision suffisante. Par contre, l'estimation des probabilités a posteriori ($\mathcal{M}$)

est un problème plus complexe à résoudre. Cette complexité est liée au nombre parfois explosif des paramètres devant être estimés.

Pour exprimer l'indépendance des attributs, on adopte une alternative de solution à caractère naïve, d'où le nom Naïve Bayes. L'hypothèse d'indépendance signifie que la probabilité conditionnelle d'un terme sachant une classe est supposée indépendante des probabilités conditionnelles des autres termes sachant la même classe. Autrement dit, les probabilités associées à chaque terme peuvent être estimées individuellement. Les modèles de classification de naïve Bayes tirent leur efficacité de cette hypothèse. Plusieurs autres classificateurs dérivés de ce modèle sont utilisés pour la génération de documents textuels et le filtrage du pourriel[42]

6. La méthode SVM

Les machines à vecteurs de support (SVM) décrivent une approche de classification supervisée basée sur une interprétation géométrique, en s'appuyant sur la notion de marge

Maximale. La figure 13 explique le principe général des méthodes SVM.

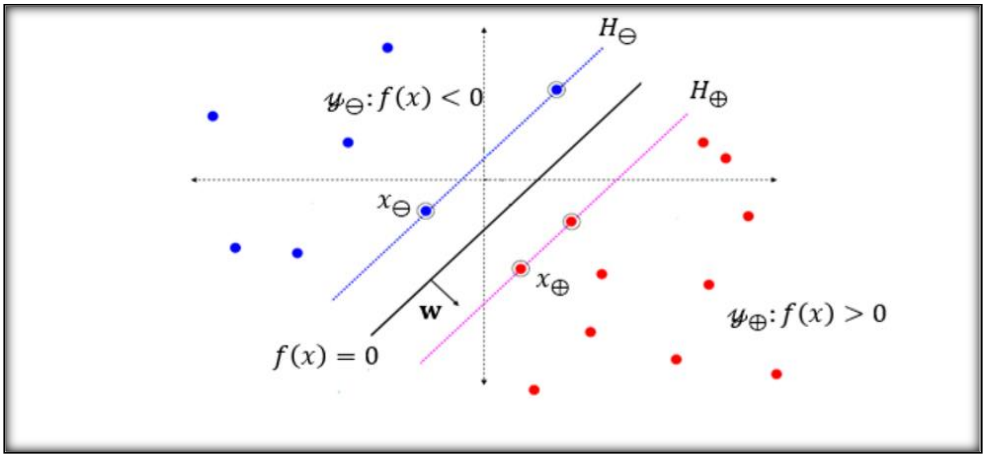


Figure 13 Principe du SVM

Pour ce qui concerne la classification du pourriel, les étiquettes (Ham, pourriel) appartiennent à un ensemble de classe ordonnée $\in\{+1,-1\}$. La marge est la distance qui sépare la frontière de séparation (un hyperplan (x)=0) des exemples les plus proches. Les vecteurs définissant la distance entre la frontière et les exemples les plus proches sont des Vecteurs de Support. L'apprentissage consiste à trouver l'hyperplan assurant le principe de maximisation de la marge, à partir de l'ensemble d'exemples, dont la solution est un problème

[42] Ekpao Anani, PASSIGUE , Analyse et détection de pourriels textuels dans les réseaux sociaux par apprentissage, UNIVERSITÉ DU QUÉBEC EN OUTAOU 18 Août 2015

d'optimisation quadratique. On peut toutefois utiliser ce type de classificateur pour résoudre des problèmes non linéaires par projection dans un espace de dimension supérieure.

Si nous cherchons à séparer des ensembles de données en deux classes y ⊕ (classe des positifs) et y ⊖ (classes des négatifs), l'algorithme SVM permet de trouver un hyperplan séparateur des deux groupes. Pour optimiser cette séparation, SVM recherche l'hyperplan pour lequel la distance entre la frontière des deux groupes et les points les plus proches est maximale.

De façon général, soit H⊕ et H ⊖ les hyperplans contenant les données x ⊕ et x ⊖, respectivement les plus proches dans la classe y ⊕ et y ⊖. Les coefficients w 0 sont choisis tels que: $\begin{cases} w^T\ x\emptyset + W0 = +1 \\ w^T x\emptyset + W0 = -1 \end{cases}$

(15)

La valeur de la marge est alors donnée géométriquement par l'équation (16):

$$\frac{W^T}{\|W\|}(x\emptyset - x\emptyset) = \frac{w^T(x\emptyset - x\emptyset)}{\|W\|} = \frac{2}{\|W\|}$$

Pour maximiser la marge, il faut minimiser la valeur de ‖w ‖. Ainsi pour classifier une nouvelle donnée , il faudra juste résoudre l'équation (17) :

$$f(x) = w^T x + w0 \begin{cases} \geq 0\ , classe\ y\emptyset \\ < 0\ , classe\ y\emptyset \end{cases}$$

Bien que les algorithmes de types SVM soient efficaces à la classification binaire, deux autres approches de ses méthodes présentent des solutions palliatives aux problèmes de classification à plusieurs classes:

- Le modèle One-versus-all vise de construire autant de modèles SVM que de classes. Ceci permettra dans le cadre de classification d'un message textuel par exemple, de retenir uniquement le modèle qui aurait retourné la plus grande marge sur l'ensemble des modèles ayant retourné un résultat positif.
- Le modèle One-Versus-one : si on dispose de nombre de classes, alors ce modèle vise de construire les (K −1)/2 classificateurs, en regroupant les classes deux à deux. La classe qui sera fréquemment retournée pendant l'exécution de l'ensemble des algorithmes sera par exemple celle à laquelle appartient un message textuel.

7. Signature des messages

Un système antispam est efficace dans la lutte des envois anormaux que lorsqu'il a une vision globale des flots de messages qui transitent sur le réseau. C'est pourquoi il doit être positionné au niveau du service d'envoi de message. Cependant, les polluposteurs rendent les envois massifs plus difficiles à détecter. Ils insèrent ou suppriment des séquences aléatoires dans les courriels afin que chaque campagne de pourriel soit unique. Ne pouvant pas se baser uniquement sur un checksum pour identifier les courriels identiques, les techniques de détection doivent se baser sur une autre forme de signature moins sensible à l'insertion et à la suppression de termes. C'est le cas de l'algorithme I-Match. L'algorithme I-Match s'appuie sur l'ensemble des termes uniques du courriel et sur un lexique préalablement établi pour produire la signature du message. Cette signature est alors associée à un unique cluster, ce qui permet d'en déduire la classe du message. C'est une technique qui détecte des envois massif, mais sensible à des modifications aléatoires du corps des messages.[43]

8. Conclusion

En combinant les arbres de décision et Adaboost, nous pouvons développer un modèle puissant et précis pour la détection des spams. Cette technique ensembliste réduit les erreurs de classification et améliore la robustesse du modèle face à des données variées et complexes. Pour des raisons de comparaison des résultats, nous avons introduit également deux méthodes de classification, à savoir la classification bayesienne et la classification par SVM. Au chapitre suivant, nous implémentons les différents classifieurs présentés dans ce chapitre, et nous les testons avec une base d'apprentissage pour la détection de spams.

[43] A. C. A. Kołcz, "Lexicon randomization for near-duplicate detection with I-Match," Supercomput Springer Science+Business Media, LLC 2008, pp. 45: 255–276, 26 January 2008.

Chapitre IV : Implémentation et tests

1. Introduction

Ce chapitre décrit en détail l'implémentation et les tests réalisés dans le cadre du projet de fin d'études sur les méthodes basées sur l'apprentissage automatique pour la détection de spams en messagerie électronique. Nous présentons ici les outils, les techniques et les algorithmes utilisés, ainsi que les résultats obtenus et leur analyse.

2. Présentation de la plateforme

2.1 Python :

Python est un langage de programmation interprété, orienté objet et un environnement de développement intégré (EDI) pour ce langage. Il a été créé à la fin des années 1980 par Guido van Rossum et est devenu très populaire pour sa simplicité et sa lisibilité de code. Python est largement utilisé pour le développement de divers types d'applications, notamment web, scientifiques, d'automatisation de tâches et bien plus encore. Sa popularité s'est accrue en raison de sa syntaxe claire et de sa grande communauté de développeurs.

Python supporte une vaste gamme de plateformes, y compris Windows, macOS, Linux, et est également utilisé dans le développement d'applications pour Android et iOS. Il dispose d'une bibliothèque standard riche et de nombreux frameworks tiers qui facilitent le développement rapide et efficace d'applications.

En résumé, Python est un langage polyvalent et largement adopté qui permet de développer des applications sur différentes plateformes en utilisant une syntaxe simple et expressive.44

2.2 Google Colab :

Google Colab, ou Google Colaboratory, est une plateforme de développement intégrée basée sur les notebooks Jupyter, accessible via le cloud. Développée par Google, cette plateforme permet aux utilisateurs d'écrire et d'exécuter du code Python directement dans leur navigateur, sans nécessiter de configuration spécifique sur leur machine locale. Ce service offre non seulement un environnement de développement interactif et convivial, mais il permet également d'accéder à des ressources puissantes comme les unités de traitement graphique (GPU) et les unités de traitement tensoriel (TPU) pour accélérer les calculs, notamment dans le domaine du machine learning. Google Colab facilite également la

[44] • Python Software Foundation. (s.d.). Python. Récupéré sur https://www.python.org/

collaboration en permettant aux utilisateurs de partager facilement leurs notebooks avec d'autres, qui peuvent visualiser, commenter et modifier le code en temps réel. Grâce à son intégration native avec Google Drive, il offre une flexibilité supplémentaire pour importer et exporter des notebooks, facilitant ainsi le partage et la sauvegarde des projets. En résumé, Google Colab est une solution puissante et accessible pour les développeurs, les chercheurs et les étudiants qui souhaitent bénéficier d'un environnement de développement Python efficace et collaboratif dans le cloud.

2.3 Scikit-learn :

Scikit-learn est une bibliothèque open-source de machine learning pour Python. Elle offre des outils simples et efficaces pour l'analyse prédictive de données, y compris la classification, la régression, le clustering, et plus encore. Scikit-learn est conçu pour être accessible et efficace, facilitant ainsi la mise en œuvre de solutions ML même pour les utilisateurs débutants. Elle comprend des algorithmes optimisés et une documentation complète qui en font un choix populaire parmi les chercheurs et les praticiens du machine Learning.[45]

2.4 Pandas :

Pandas est une bibliothèque open-source de manipulation de données en Python. Elle offre des structures de données puissantes et flexibles pour travailler avec des tableaux de données et des séries temporelles. Pandas permet d'importer, nettoyer, explorer et manipuler efficacement des données, facilitant ainsi les tâches de préparation de données et d'analyse exploratoire. Cette bibliothèque est largement utilisée dans le domaine du data science et est appréciée pour sa simplicité et sa richesse en fonctionnalités.46

2.5 NumPy :

NumPy est une bibliothèque open-source pour Python qui facilite les calculs sur des tableaux multidimensionnels ainsi que la manipulation de ces tableaux. NumPy offre des fonctions mathématiques avancées et des outils pour travailler avec ces tableaux, ce qui en fait une base essentielle pour de nombreuses autres bibliothèques de data science et de machine learning. Grâce à NumPy, les opérations mathématiques sur des tableaux sont optimisées pour être

[45] Scikit-learn. (s.d.). Documentation officielle. Récupéré sur https://scikit-learn.org/

[46] Pandas. (s.d.). Documentation officielle. Récupéré sur https://pandas.pydata.org/

rapides et efficaces, ce qui est crucial pour les applications nécessitant des calculs numériques intensifs.47

2.6 Matplotlib et Seaborn :

2.6.1. Matplotlib est une bibliothèque de visualisation de données en Python, offrant une flexibilité et un contrôle complets sur la création de graphiques statiques, graphiques interactifs et visualisations complexes. Matplotlib est largement utilisée dans la communauté scientifique et académique pour créer des graphiques de haute qualité.48

Seaborn, quant à elle, est une bibliothèque basée sur Matplotlib, spécialisée dans la visualisation de données statistiques. Seaborn simplifie la création de graphiques attrayants en fournissant des interfaces haut niveau pour dessiner des graphiques informatifs.

Ces deux bibliothèques sont complémentaires et populaires dans le domaine de la data science pour explorer et présenter des données de manière efficace et esthétique.49

3. Installation des Bibliothèques

Toutes les bibliothèques nécessaires peuvent être facilement installées à l'aide de l'utilit qui est inclus de manière standard avec Python. Voici les principales bibliothèques que nous allons installer :

- **Pandas** : Pour la manipulation et l'analyse des données structurées, Pandas offre des structures de données puissantes et faciles à utiliser.
- **NumPy**: fournit un support efficace pour les tableaux multidimensionnels, essentiels pour les calculs numériques dans Python.
- **Matplotlib** : Pour la visualisation des données, Matplotlib est une bibliothèque de traçage complète qui permet de créer une grande variété de graphiques et de visualisations.
- **Seaborn** : Construit sur Matplotlib, Seaborn offre une interface de haut niveau pour la création de visualisations statistiques attrayantes.

Scikit-learn : Cette bibliothèque propose une gamme complète d'algorithmes de machine learning avec une interface simple et uniforme pour les tâches d'apprentissage supervisé et non supervisé.

[47] NumPy. (s.d.). Documentation officielle. Récupéré sur https://numpy.org/
[48] Matplotlib. (s.d.). Documentation officielle. Récupéré sur https://matplotlib.org/
[49] Seaborn. (s.d.). Documentation officielle. Récupéré sur https://seaborn.pydata.org/

Nous allons installer ces bibliothèques en utilisant pip. Assurez-vous d'avoir une connexion Internet active et d'exécuter ces commandes dans un environnement compatible avec Python.

Pour plus de détails sur l'installation et l'utilisation de ces bibliothèques, vous pouvez consulter les documentations officielles respectives :

Pandas

NumPy

Matplotlib

Seaborn

Scikit-learn

4. Extraits du code python

4.1. Chargement et Préparation des Données

Les données utilisées proviennent du fichier `spambase.csv`, qui contient des caractéristiques extraites d'emails, ainsi qu'une étiquette indiquant si l'email est un spam.

```
# Load the CSV assuming you've uploaded and have a Spambase CSV file
data = pd.read_csv('/content/drive/MyDrive/spambase.csv')
data.head()
```

4.2. Séparation des Données

Nous séparons les caractéristiques (features) de la cible (target).

```
# Split into features (X) and target variable (y)
X = data.drop('spam', axis=1)
y = data['spam']
```

l'ensemble de donnees data est divisé en deux sous ensembles :

X : les colonnes des features

y : la colonne de la classe (label).

Séparation des données en données d'apprentissage et données de test :

```
# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

Les donnes X et y sont séparées.

4.3. Implémentation des Modèles de Machine Learning

Nous avons implémenté plusieurs algorithmes de machine learning pour la détection de spams, dont Naive Bayes, la Support Vector Machine (SVM), l'Arbre de Décision, et l'Adaboost. Chaque modèle est entraîné et évalué sur les ensembles de données d'entraînement et de test.

Nous avons utilisé un tableau de 4 classifieurs, puis nous avons bouclé sur cet ensemble de classifieurs et pour chacun on procède à son apprentissage, son test avec calcul de certaines métriques de performance, dont la précision principalement, avec affichage de la matrice de confusion.

Ensemble de modèles

```
# Models
models = {
    'Naive Bayes': MultinomialNB(),
    'Support Vector Machine': SVC(),
    'Decision Tree': DecisionTreeClassifier(),
    'AdaBoost': AdaBoostClassifier(n_estimators=100, algorithm="SAMME",
random_state=0)
}
```

Préparation du tableau des résultats

```
# Results storage
results = {'Model': [], 'Accuracy': [], 'Precision': [], 'Recall': [],
'F1-Score': []}
```

Boucle sur l'ensemble des 4 classifieurs

```
# Train and evaluate each model
for name, model in models.items():
```

Entrainement du classifieur

```
    # Fit the model to the training data
    model.fit(X_train, y_train)
```

Test du modèle

```
    # Make predictions on the testing set
    y_pred = model.predict(X_test)
```

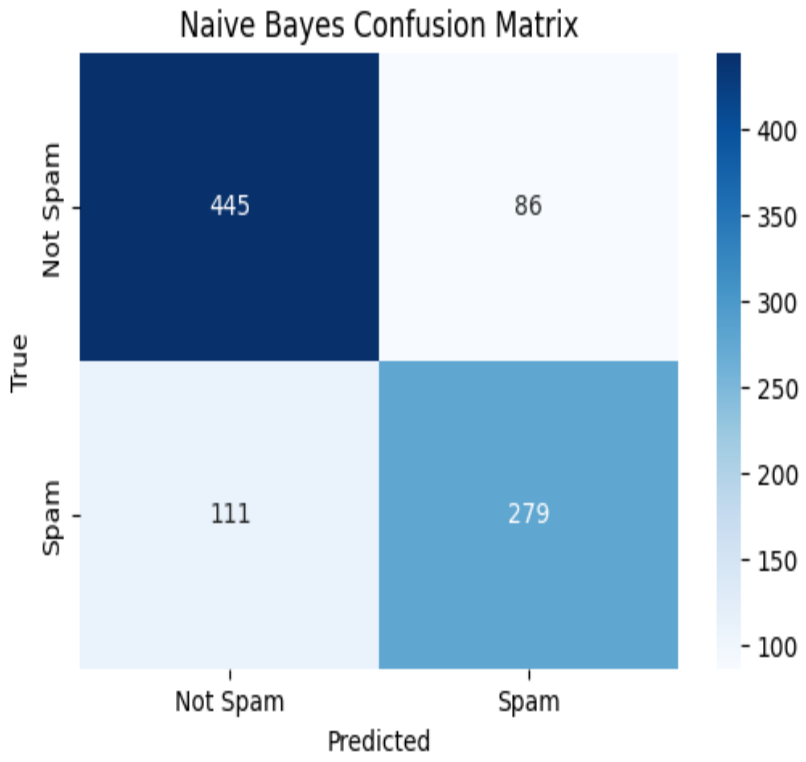
Calcul des métriques d'évaluation

```
    # Calculate performance metrics
    accuracy = accuracy_score(y_test, y_pred)
```

```
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)

      Stockage des métriques calculées dans le tableau des résultats
    # Store results
    results['Model'].append(name)
    results['Accuracy'].append(accuracy)
    results['Precision'].append(precision)
    results['Recall'].append(recall)
    results['F1-Score'].append(f1)

      Matrice de confusion
    # Confusion Matrix
    cm = confusion_matrix(y_test, y_pred)
    plt.figure(figsize=(6, 4))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=['Not Spam', 'Spam'], yticklabels=['Not Spam', 'Spam'])
    plt.title(f'{name} Confusion Matrix')
    plt.xlabel('Predicted')
    plt.ylabel('True')
    plt.show()

      Affichage des métriques
    print(f"\n{name} Metrics:")
    print("Accuracy:", accuracy)
    print("Precision:", precision)
    print("Recall:", recall)
    print("F1-Score:", f1)
```

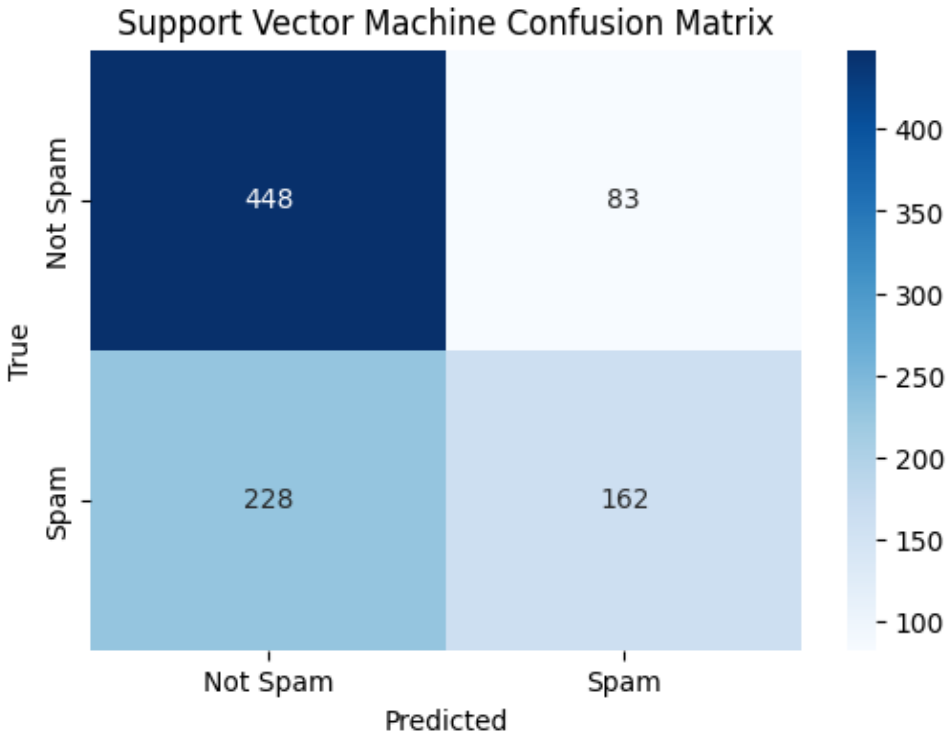
5. Résultats des tests

5.1. Modèle Naive Bayes



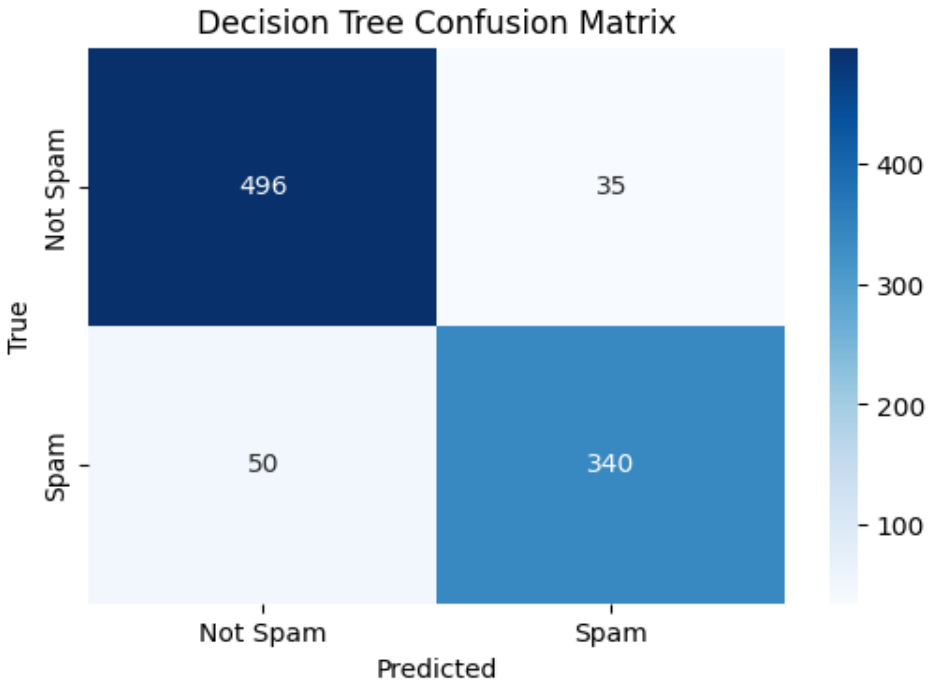

```
        Naive Bayes Metrics:
Accuracy: 0.7861020629750272
Precision: 0.7643835616438356
Recall: 0.7153846153846154
F1-Score: 0.7390728476821192
```

5.2. Modèle SVM



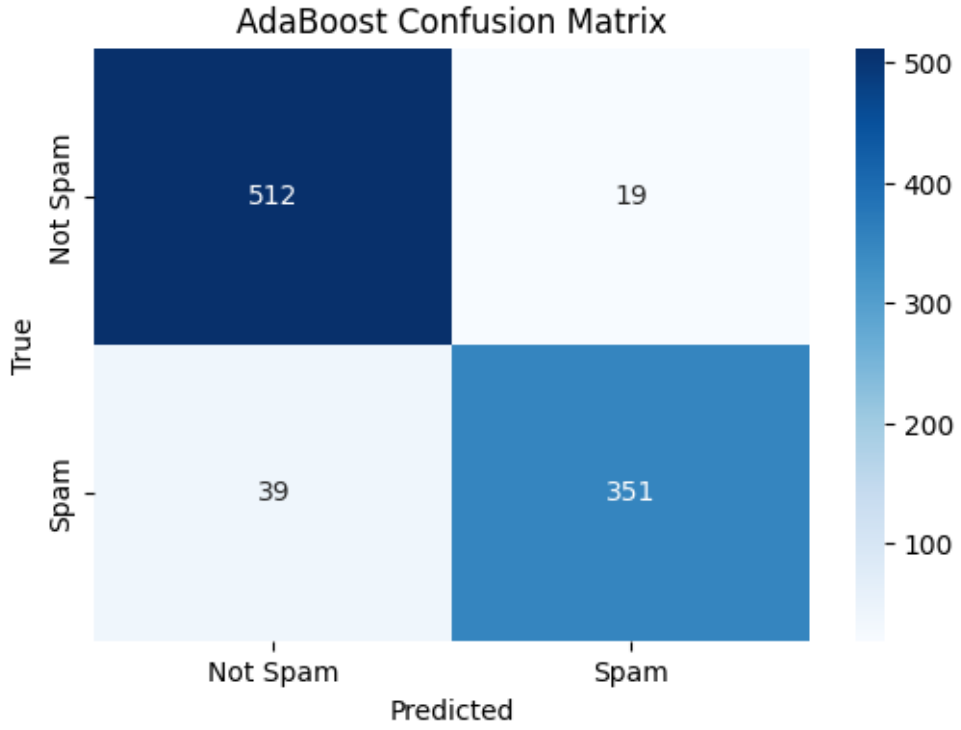
```
Support Vector Machine Metrics:
Accuracy: 0.6623235613463626
Precision: 0.6612244897959184
Recall: 0.4153846153846154
F1-Score: 0.510236220472441
```

5.3. Modèle Arbre de Décision




```
Decision Tree Metrics:
Accuracy: 0.9131378935939196
Precision: 0.91005291005291
Recall: 0.882051282051282
F1-Score: 0.8958333333333333
```

5.3. Modèle Adaboost




```
AdaBoost Metrics:
Accuracy: 0.9370249728555917
Precision: 0.9486486486486486
Recall: 0.9
F1-Score: 0.9236842105263158
```

6. Analyse des résultats

Il ressort des résultats des différents classifieurs testés que l'Adaboost (qui utilise par défaut les arbres de décision comme classifieurs faibles) a affiché la meilleure précision (94.86%), suivi des arbres de décisions comme par une précision de 91.00%. Les autres classifieurs considérés pour des raisons de comparaisons, ils ont affiché respectivement 76.43% pour le classifieur bayesien et 66.12% pour le classifieur à base de SVM. En conclusion, et outre les classifieurs bayesien et SVM qui affichent des résultats médiocres, les arbres de décision affichent de bons résultats, notamment quand elles sont combinées par le classifieur ensembliste Adaboost.

5. Conclusion

Dans ce chapitre, nous avons décrit en détail l'implémentation et les tests réalisés pour la détection de spams en utilisant différentes méthodes basées sur l'apprentissage automatique. Les étapes suivantes ont été couvertes : préparation de l'environnement de développement,

prétraitement des données, implémentation des modèles, évaluation des performances et comparaison des résultats.

Conclusion générale

Ce projet de fin d'études, consacré aux méthodes basées sur l'apprentissage automatique pour la détection de spams en messagerie électronique, a permis de mettre en lumière l'efficacité et la pertinence de diverses techniques de machine Learning pour résoudre un problème crucial de cyber sécurité. À travers l'implémentation, l'évaluation et la comparaison de plusieurs modèles, nous avons pu identifier les approches les plus performantes et proposer des pistes d'amélioration futures.

Nous avons étudié les arbres de décisions ainsi que l'algorithme Adaboost comme méthode ensembliste de classification. Pour étendre la comparaison des résultats, nous avons considéré également le classifieur bayesien et le classifieur à base de SVM. Il ressort de l'experimentation des différents modèles implémentés que l'Adaboost est le meilleur en termes de persormance, suivi des arbres de décision. Les suivants sont respectivement le classifieur bayesien et le classifieur SVM avec des résultats médiocres.

Finalement, ce projet a illustré la puissance et la flexibilité des méthodes d'apprentissage automatique pour la détection de spams en messagerie électronique. En comparant plusieurs modèles et en analysant leurs performances, nous avons pu identifier les techniques les plus efficaces et proposer des améliorations pour l'avenir. Les résultats obtenus soulignent l'importance de la qualité des données pour atteindre des performances élevées. En somme, ce projet ouvre la voie à des applications plus avancées et robustes, contribuant à un environnement numérique plus sûr et plus fiable, et constitue une base solide pour des recherches futuresdans ce domaine essentiel de la cyber sécurité.

Bibliographie

Bibliographie

https://becominghuman.ai/decision-trees-in-machine-learning-f362b296594a
https://towardsdatascience.com/a-guide-to-decision-trees-for-machine-learning-and-data-science-fe2607241956
https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052
https://medium.com/deep-math-machine-learning-ai/chapter-4-decision-trees-algorithms-b93975f7a1f1
https://medium.com/coinmonks/what-is-entropy-and-why-information-gain-is-matter-4e85d46d2f01
https://scikit-learn.org/stable/modules/tree.html
https://sefiks.com/2017/11/20/a-step-by-step-id3-decision-tree-example/
https://sefiks.com/2018/05/13/a-step-by-step-c4-5-decision-tree-example/
https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/
https://medium.com/machine-learning-guy/an-introduction-to-decision-tree-learning-id3-algorithm-54c74eb2ad55
https://www.datacamp.com/community/tutorials/decision-tree-classification-python
https://pdfs.semanticscholar.org/2b3c/17da5e60d5bc953c181ca637bf6262469d25.pdf