République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la Matière

Département d'Informatique

Mémoire de Fin d'études Master

Filière : Informatique

Option : Sciences et Technologie de l'Information et de la Communication (STIC)

Thème :

Détection des sites d'hameçonnage pour assurer la sécurité sur Internet

Encadré Par :

Dr. Hannousse Abdelhakim

Présenté par :

Chemlal Maroua

Octobre 2020

بسم الله الرحمن الرحيم

Remerciement

Avant tout, je louer le bon dieu qui m'a donné la capacité et la patience pour terminer ce travail de rechercher malgré tous les obstacles.

Je tiens à remercier mes parents et mes proches qui m'ont soutenu tout au long de mon cursus universitaire.

Avec les belles expressions de respect, j'adresse mes remerciements les plus sincères à mon encadreur ***Mr HANNOUSSE Abdelhakim*** *qui m'a mis sur la bonne voie par ses précieux conseils tout au long de ma rédaction. Je le remercie pour la qualité de son encadrement exceptionnel.*

Mes chaleureux remerciements vont également aux jurys qui ont accepté de lire et d'évaluer ce travail.

Je remercie infiniment tous mes enseignants du département de l'informatique en particulier le chef de département ***Mr. KOUAHLA Zinedine.***

Il est particulièrement agréable d'exprimer mes gratitudes et mes remerciements chaleureux à ***Mr. SOUILAH Abd El-Aali*** *pour ses remarques, ses conseils et son soutien moral dans les moments difficiles.*

Un grand remerciement à tous ceux qui m'ont aidées de près ou de loin à l'élaboration de ce travail.

Dédicace

Avec joie, fierté et respect je dédie ce modeste travail aux plus chères personnes dans ma vie :

À Mes chers parents pour tout ce qu'ils ont fait pour moi, pour leurs conseils, leur amour, et leur soutient durant ma période d'études.

À celle qui a été la source de mon existence et qui m'était toujours la lumière de ma vie. Merci ***ma mère****.*

À mon père qui m'a donné toujours le courage, la volonté et la gratitude. Merci ***mon père****.*

À mes chers ***frères*** *et* ***sœurs****, pour leur aide et leur soutien plus que précieux. À mes chers* ***neveux*** *et* ***nièces****. À mes* ***grands-parents****.*

À toutes les ***personnes*** *que* ***j'aime*** *et* ***m'aiment****.*

*Qu'****Allah*** *vous bénisse.*

Une grande dédicace pour toutes ***mes amies****.*

Aucun mot sur cette page en serait exprimé ce que je vous dois, ni combien je vous aime.

TABLE DES MATIERES

LISTE DES FIGURES

LISTE DES TABLEAUX

RESUME

Vue le développement énorme d'utilisation d'Internet, l'hameçonnage est devenu l'un des principaux problèmes de la sécurité sur Internet. Les criminels informatiques utilisent la technique d'hameçonnage pour obtenir illégalement des renseignements personnels des utilisateurs dans le but de perpétrer une usurpation d'identité. Vue la croissance des sites d'hameçonnage et les techniques utilisées pour ce but, il est devenu nécessaire de développer des outils automatiques pour la détection de ces sites.

Notre étude vise à développer un modèle efficace pour la détection des sites web d'hameçonnages. Notre analyse montre l'efficacité des techniques d'apprentissage automatique, notamment l'algorithme de forêt d'arbres décisionnels pour la classification des URLs. Dans ce mémoire, nous avons développé une extension (plugin) au Google Chrome qui sert à être un middleware entre les utilisateurs et les sites malveillants visités. L'extension utilise un modèle d'apprentissage automatique combiné des deux modèles de base. Le premier modèle est entrainé sur des caractéristiques liées à la structure et syntaxe d'URL. Le deuxième modèle est entrainé sur des caractéristiques liées au contenu HTML des URLs. Notre modèle combiné atteindre 96.36% de fiabilité pour la distinction des sites d'hameçonnages des sites légitimes.

Mot Clés : hameçonnage informatique, apprentissage automatique, combinaison des modèles d'apprentissage.

ABSTRACT

With the huge growth in Internet use, phishing has become one of the main problems in Internet security. Computer criminals use phishing techniques to illegally obtain personal information from users for malicious purposes. Due to the growth of phishing tactics and techniques used for phishing detection, it has become a necessity to develop automatic tools for the detection of the legitimacy of these websites.

Our study aims to develop an efficient model for the detection of phishing websites. Our analysis shows the effectiveness of machine learning techniques, specifically the Random Forest algorithm, for URL classification. In this thesis, we developed a Google Chrome plugin that serves as a middleware between users and the malicious visited websites. The extension uses a combined machine learning model of two base models. The first model is trained on features concerned with the structure and syntax of URLs. The second model is trained on features related to the HTML content of URL pages. Our combined model achieves 96.36% of accuracy in distinguishing phishing from legitimate websites.

Keywords: phishing, machine learning, combination of models.

الملخص

مع زيادة استخدام الإنترنت بشكل كبير، أصبح يُعد التصيد الاحتيالي أحد الأشكال الشائعة لجرائم الإنترنت وأحد المشكلات الرئيسية في أمان الإنترنت. يستخدم قراصنة الكمبيوتر تقنيات التصيد للحصول على معلومات شخصية بشكل غير قانوني من المستخدمين بغرض سرقة الهوية. نظرًا لتزايد مواقع التصيد والتقنيات المستخدمة لهذا الغرض ، فقد أصبح من الضروري تطوير أدوات آلية للكشف عن هذه المواقع.

تهدف دراستنا إلى تطوير نموذج فعال لاكتشاف مواقع التصيد الاحتيالي. يُظهر تحليلنا فعالية تقنيات التعلم الآلي، بما في ذلك خوارزمية Random Forest. في هذه المذكرة، قمنا بتطوير امتداد (مكون إضافي) لـ Google Chrome والذي يعمل كبرنامج وسيط بين المستخدمين والمواقع الضارة التي يتم زيارتها. يستخدم الملحق نموذجًا مدمجًا للتعلم الآلي لنموذجين أساسيين. الأول تم تدريبه على الخصائص المتعلقة بهيكل وصياغة URL. تم تدريب الثانية على الخصائص المتعلقة بمحتوى HTML لعناوين URL. يحقق نموذجنا المركب موثوقية تصل إلى 96.36٪ في تمييز مواقع التصيد الاحتيالي عن المواقع الشرعية.

الكلمات الدالة : الخداع الالكتروني ، التعليم الألي ، تركيب النماذج.

INTRODUCTION GENERALE

INTRODUCTION GENERALE

Petit à petit l'usage de l'Internet se renforçant, ce développe le concept d'identité numérique mais l'Internet est un véritable paradoxe en matière de droits car il se révèle bénéfique pour l'exercice des droits de l'homme et bafoue à la fois certains droits de la personne privée.

Depuis la dernière décennie, les internautes ont perdu des milliards de dollars par an à cause de l'hameçonnage, ce dernier est en train de devenir un des plus grands crimes du réseau. Parce qu'il est une tentative d'usurpation d'identité qui sert à obtenir des renseignements personnels par les courriers électroniques, par sites Web falsifiés ou par d'autres moyens électroniques.

Notre recherche s'inscrit dans le domaine de l'intelligence artificielle en se basant sur les techniques d'apprentissage automatique qui sont plus prometteuses pour cette étude. A ce propos, nous choisissons de développer un modèle d'apprentissage automatique dans le but de détecter les sites web d'hameçonnages. La problématique dont nous nous interrogeons tourne autour : comment nous choisissons un modèle d'apprentissage plus efficace pour la détection des sites Web d'hameçonnage ?

L'analyse de la littérature montre une diversification des algorithmes utilisés et des caractéristiques considérées pour l'apprentissage automatique. Dans ce contexte, nous collectons un ensemble des caractéristiques des différentes natures et on les tests sur différents algorithmes d'apprentissages et différentes combinaisons des algorithmes. Le meilleur modèle est opté pour être utilisé pour le développement d'une extension de Google Chrome dans la détection des sites web d'hameçonnages visité par l'utilisateur. L'extension sera un intermédiaire avec l'utilisateur qui lui permettra de vérifier la légitimité du site dans le cas d'un doute. Notre expérimentation montre l'efficacité de l'algorithme foret d'arbres décisionnels sur les deux classes des caractéristiques : ceux liées à la structure et lexique de l'URL et ceux extraites des

contenus HTML des pages. L'approche que nous avons proposée pour la détection de l'hameçonnage est la combinaison des deux modèles de foret d'arbres décisionnels chacun entrainé sur une classe particulière des caractéristiques. Le modèle obtenu permet une bonne prédiction de 96.36%.

Ce mémoire se compose de 5 chapitres, le premier chapitre est consacré à la présentation et la définition des concepts de base de l'hameçonnage informatique. Le deuxième chapitre décrit les différentes techniques utilisées pour la détection des sites d'hameçonnage. Le troisième chapitre discute des travaux connexes. Le quatrième chapitre décrit la démarche suivie pour la détermination d'un modèle efficace. Le dernier chapitre présente les résultats obtenus, le choix fait pour le modèle, et l'implémentation de l'extension de Google Chrome basée sur le modèle choisi. Nous clôturons notre mémoire par une conclusion et quelques perspectives.

CHAPITRE I

HAMEÇONNAGE INFORMATIQUE

CHAPITRE I.

HAMEÇONNAGE INFORMATIQUE

Pour un travail facile, rapide et efficace sur Internet, les sites Web sont parmi les outils les plus utilisés et sont le premier choix pour les commerçants et les industries quotidiennes. Malgré leur vaste utilisation par ces derniers, les failles de sécurité des sites web ont encore apportées divers dangers et des charges financières à leurs sociétés. Ainsi, la sécurité web devient l'une des questions fondamentales de l'Internet d'aujourd'hui.

Le thème sur lequel porte ce travail est l'hameçonnage, qui est très présent au cours de ces dernières années, le nombre d'attaques de l'hameçonnage a augmenté de façon spectaculaire. Le but de ces attaques est d'exploiter les informations sensibles de l'utilisateur (telles que les numéros de cartes de crédit ou de comptes bancaires) afin de leur voler de l'argent ; l'attaquant trompe sa victime avec des techniques d'ingénierie sociale (telles que SMS, voix, e-mail, le courrier électronique, site Web et logiciels malveillants).

De ce fait, les chercheurs explorent les solutions qui peuvent remédier efficacement à l'hameçonnage, la fourniture de ces solutions devient de plus en plus importante.

Dans ce chapitre, nous allons explorer les différents concepts de base du monde de l'hameçonnage ainsi que leurs définitions.

1.1. La communication

Le dictionnaire LA ROUSSE nous fournit la définition suivante de la communication : « *Action, fait de communiquer, de transmettre quelque chose : Communication de la chaleur à un corps. Action de communiquer avec quelqu'un, d'être en rapport avec autrui, en général par le langage ; échange verbal entre un locuteur et un*

interlocuteur dont il sollicite une réponse : le langage, le téléphone sont des moyens de communication. » [35].

D'après cette définition nous pouvons dire que la communication est une chaine de six éléments où le message est le noyau de cette chaine, en fait, pour bien communiquer, il est nécessaire de comprendre les éléments fondamentaux qui sous-tendent une communication efficace. Ces éléments sont représentés dans le schéma de Jakobson [1] que nous reprenons comme suit :

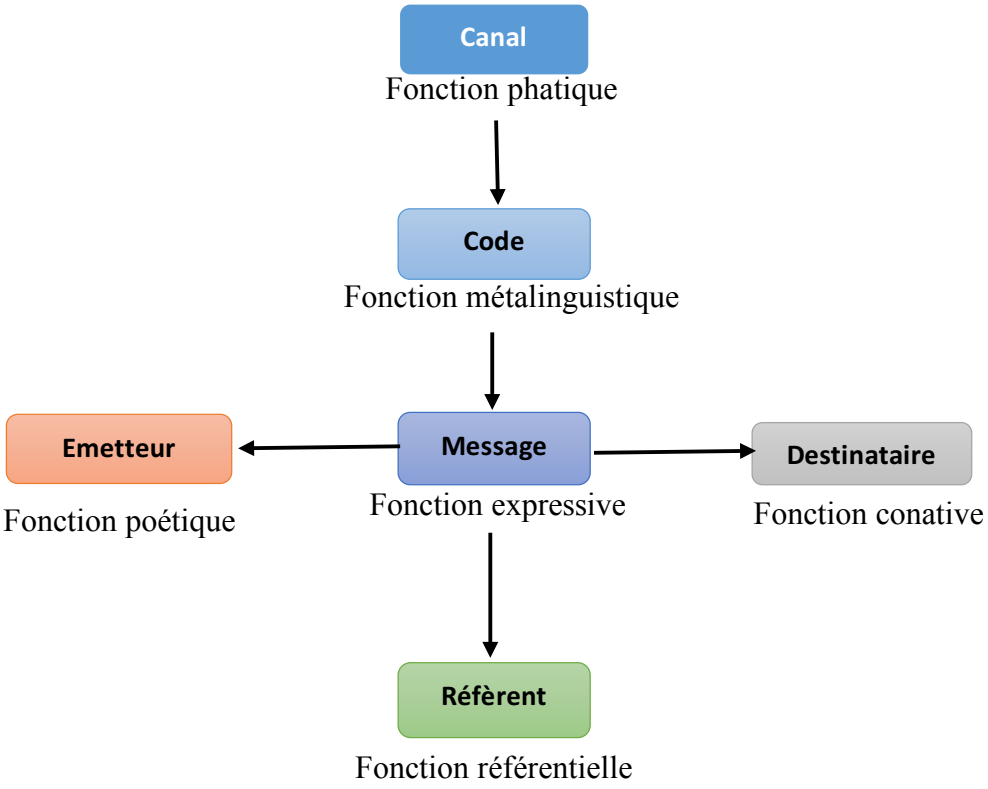


Figure 1.1 : Schéma de Jakobson [1]

Si nous voulons expliquer le schéma de Jakobson [2], nous pouvons dire que :

- **Le message :** c'est l'information transmise selon une certaine forme, ce qui est écrit, ce qui est dit.
- **L'émetteur :** c'est celui qui envoie le message, ce peut être une entreprise, une collectivité, une association, etc.

- **Le destinataire ou le récepteur :** c'est celui qui reçoit le message, qui le lit, qui l'entend. Ce peut être un client potentiel ou la cible.
- **Le contexte (référent) :** c'est le sujet du message, c'est ce dont on parle.
- **Le canal :** c'est le support du message entre l'émetteur et le récepteur. Le sens premier du terme *media*, pluriel du mot latin medium signifie "intermédiaire" (e.g. radio, télé, presse, affiche, web, etc.).
- **Le code :** si le message est codé par l'émetteur et décodé par le récepteur, il faut connaître le code pour comprendre le message, un mauvais code entraine certainement un mauvais décodage donc une mauvaise interprétation qui peut avoir comme synonyme « *déformation* ».

Cela nous amène à dire que la communication par Internet nous permet aussi des échanges d'informations sous des formats différents (textes, sons, vidéos, images) mais à travers d'outils spécifiques. Ces outils permettent la création de liens et favorisent la communication instantanée quel que soit l'heure et le lieu. L'Internet favorise aussi la communication simultanée entre plusieurs internautes.

1.2. La cybercriminalité

Pour attribuer plus de précision au terme cybercriminalité, nous pouvons dire qu'il désigne toutes les actions susceptibles de se commettre sur ou au moyen d'un système informatique généralement connecté à un réseau [2].

D'après cette définition, nous comprenons que la cybercriminalité est un crime dans lequel l'ordinateur est l'objet du crime (piratage, hameçonnage, spamming) il est utilisé comme un outil pour commettre une infraction (pornographie juvénile, crimes de haine, etc.). Les cybercriminels peuvent utiliser la technologie informatique pour accéder à des informations personnelles, des secrets commerciaux ou utiliser Internet à des fins d'exploitation ou malveillantes. Les criminels peuvent également utiliser des ordinateurs pour la communication et le stockage de documents ou de données. Ce sont « *les pirates informatiques* ». La cybercriminalité peut également être qualifiée de criminalité informatique. Aussi, il faut dire que les types courants de cybercriminalité incluent le vol d'informations bancaires en ligne, l'usurpation

d'identité, les délits d'éviction en ligne et l'accès non autorisé à un ordinateur. Des délits plus graves, comme le cyber terrorisme, sont également très préoccupants.

Si nous voulons diviser la cybercriminalité, nous pouvons dire qu'il y a deux catégories. La première qui cible les réseaux informatiques ou les appareils qui incluent les virus et les attaques par déni de service. Tandis que, la deuxième utilise les réseaux informatiques pour faire avancer d'autres activités criminelles telles que le cyber harcèlement, l'hameçonnage et la fraude ou le vol d'identité.

1.3. L'arnaque

Une arnaque désigne une escroquerie ou un vol, c'est-à-dire le fait d'obtenir quelque chose par une manœuvre frauduleuse. On considère que tout le monde connaît l'existence de l'arnaque à travers les collèges, et les médias. Ce phénomène est dangereux parce qu'il peut menacer tous les groupes de la société sans exception, à partir de l'utilisation de l'internet et les réseaux sociaux.

Avec Internet, le nombre de victimes augmente d'une façon virtuelle, à vive allure et à un coût très faible. Donc l'arnaque est un phénomène qui reste toujours populaire, où les attaquants expérimentés dans le domaine informatique profitent de la naïveté de leurs victimes.

1.4. L'hameçonnage

Avant de connaitre les types d'hameçonnage et ses secteurs, il est souhaitable de faire d'abord une petite définition d'hameçonnage. En fait, nous pouvons dire que le terme d'hameçonnage est le mélange de deux mots anglais : *phreaking* qui désigne le piratage de lignes téléphoniques et *phishing* qui se traduit par *pêche* en français.

Selon Webopedia [36], le sens de l'hameçonnage est « *Le fait d'envoyer un e-mail à un utilisateur prétendant faussement être une Enterprise légitime établie dans une tentative d'escroquerie à l'utilisateur en livrant des informations privées qui seront utilisées pour le vol d'identité.* » [36].

De plus, il s'agit d'une technologie frauduleuse que les pirates utilisent très largement sur le Web pour voler des employés ou des informations précieuses liées à l'identité de l'utilisateur. Une fois que les attaquants obtiennent des données personnelles (mots de passe, date de naissance et numéros de compte bancaire), ils utilisent les informations à leur avantage en créant de fausses identités ou en contrôlant les comptes en ligne.

Les attaques d'hameçonnage sont lancées de différentes manières, et la méthode la plus utilisée consiste à envoyer des e-mails aux utilisateurs et à les persuader de cliquer sur un faux lien dans l'e-mail ; les utilisateurs seront alors renvoyés vers le site suspect qui est similaire à un site de confiance par exemple le site d'une banque en ligne. Étant donné que le site est sous le contrôle de l'attaquant, l'attaquant obtient toutes les informations que l'utilisateur divulgue sur le site comme les noms d'utilisateurs et les mots de passe. Un utilisateur alerté peut découvrir la différence entre le faux site Web créé par l'attaquant et le site légitime. Cependant, un grand nombre d'utilisateurs souffre de ces attaques. Cela nous amène à dire que cette fraude exploite un défaut humain et non un défaut informatique, c'est ce que nous appelons la technologie de l'ingénierie sociale.

1.4.1 Les étapes d'attaque d'hameçonnage

Nous devons étudier précisément les processus et les étapes d'attaques d'hameçonnage afin de trouver une solution globale pour eux. En général, ces attaques sont constituées de 4 étapes, les attaquants vont suivre ces étapes après la conception d'un faux site très similaire aux sites de confiance des utilisateurs :

- **Distribution des liens malveillants :** Dans cette étape, les pirates envoient un lien malveillant à l'utilisateur par e-mail ou messages instantanés. Les utilisateurs peuvent être considérés comme des cibles de fraude intentionnelle, si la victime est connue de l'attaquant, mais la fraude et l'arnaque peuvent aussi être randomisées.
- **Visiter les sites de l'hameçonnage :** Dans cette étape, la victime va cliquer sur un lien malveillant qui a été reçu par un canal de communication. Par conséquent, la victime sera redirigée vers le site d'hameçonnage avec une apparence similaire au site réel que l'utilisateur connait.

- **Révéler des informations sensibles :** À cause de la grande ressemblance entre le faux site Web et le site réel, l'utilisateur sera convaincu de dévoiler ses renseignements personnels et financiers, comme il le fait sur le site réel.
- **Transfert des informations divulguées à l'attaquant :** Ensuite, les informations seront envoyées aux attaquants et l'ID utilisateur sera pris en otage par ces derniers.

1.4.2 Les secteurs les plus ciblés par l'hameçonnage

Au deuxième trimestre de 2020, OpSec Security [2], membre de l'APWG (Anti-Phishing Working Group), a constaté que les sites logiciels en tant que service (SaaS) et de messagerie Web demeuraient les principales cibles d'hameçonnage, avec plus de 35% de toutes les attaques : « *Au deuxième trimestre, nous avons détecté une concentration légèrement plus élevée d'attaques sur les principaux secteurs ciblés, avec des augmentations spécifiques par rapport au premier trimestre des cibles SAAS / Webmail et médias sociaux* », a déclaré Stefanie Wood Ellis, responsable des produits et du marketing anti-fraude chez OpSec Online. « *Les attaques visant le secteur des médias sociaux ont augmenté d'environ 20% au deuxième trimestre par rapport au premier trimestre, principalement en raison d'attaques ciblées contre Facebook et WhatsApp*. » OpSec Online (ancien membre fondateur de l'APWG MarkMonitor) [2].

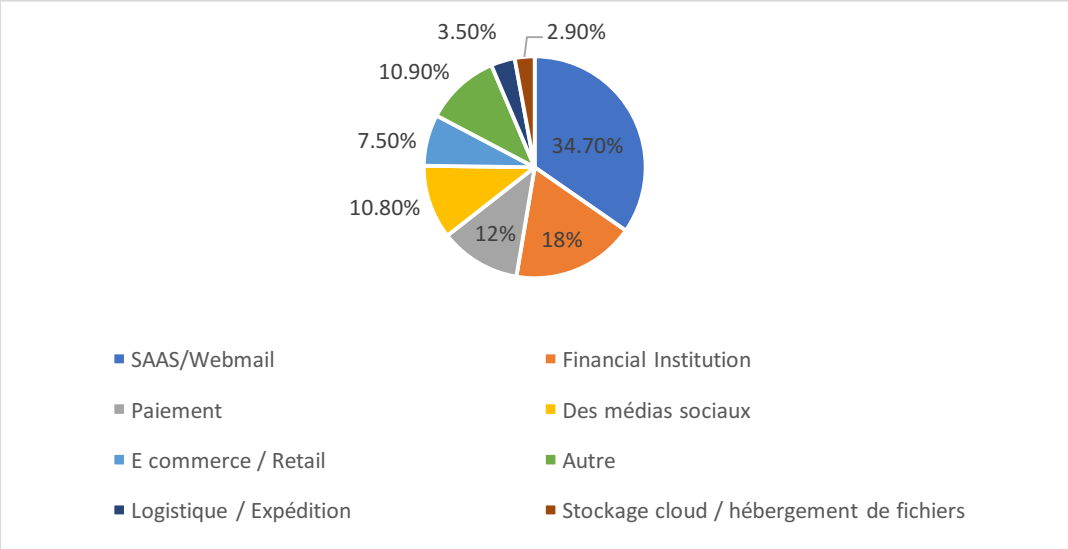


Figure 1.2. Les secteurs industriels les plus ciblés par l'hameçonnage [2]

Parmi les secteurs les plus vulnérable an risque d'hameçonnage il y a [2]:

a) Banque et finance :

L'exemple le plus remarquable et le secteur le plus menacé par les hackers c'est bien les établissements bancaires et financiers, ce sont de véritables mines d'informations sensibles. Leurs systèmes informatiques contiennent les numéros de cartes bancaires et les coordonnées de leurs clients, ainsi que toutes sortes de données financières et confidentielles.

b) L'e-commerce

Les commerçants sont exposés aux différents types de cyber attaques, lors de la proposition de leurs produits, l'exposition des données de leurs clients et leurs fonctions en ligne, dont les conséquences sont parfois terribles.

c) Médias sociaux

La plupart des e-mails d'hameçonnage ciblant les réseaux sociaux sont envoyés à des adresses personnelles plutôt qu'à des adresses professionnelles. Cependant, les lignes d'objet de ces e-mails sont conçues pour tromper ("*Security Alert !*"), et les pirates informatiques n'hésitent pas à localiser les adresses des entreprises. Les victimes d'hameçonnage réagiront émotionnellement sans prendre du recul et cliqueront sur des liens sans se rendre compte de la tromperie.

d) La santé

Les organismes de santé sont régulièrement visés par des cybers attaques. En effet, trop souvent, les systèmes informatiques utilisés dans les hôpitaux et chez les médecins sont mal protégés ou obsolètes. À l'heure de la généralisation du dossier médical partagé, il est essentiel d'adopter les bons réflexes, avec chiffrement des données, anti-virus, et pare-feu.

e) L'enseignement

Les systèmes informatiques des universités et écoles ont de nombreuses failles que les hackers utilisent pour subtiliser des données personnelles et des informations sur des

recherches sensibles. Ils se servent également parfois de l'infrastructure réseau des établissements éducatifs pour lancer d'autres attaques.

Nous pouvons constater ainsi qu'aucun secteur n'échappe totalement à la malveillance des pirates informatiques, c'est pourquoi il est indispensable de nous tourner vers un professionnel de la sécurité informatique si nous souhaitons nous protéger de façon efficace.

1.4.3 La prévalence des attaques d'hameçonnage

En fait, le nombre d'attaques d'hameçonnage est en augmentation rapide. En raison de leur valeur, comme mentionné précédemment, les institutions financières sont les cibles les plus vulnérables du piratage et de la fraude, et récemment, ce type d'attaque s'est diversifié et a également commencé à cibler les sites de réseaux sociaux, en utilisant des techniques plus sophistiquées que les attaquants adoptent quotidiennement. Le tableau 1.1 présente les faits saillants des statistiques obtenues par une étude récente sur les attaques d'hameçonnage selon des études récentes.

	Avril	**Mai**	**Juin**
Nombre de sites Web d'hameçonnage uniques détectés.	48951	52007	46036
Nombre de rapports d'e-mail d'hameçonnage uniques (campagnes) reçus par APWG des consommateurs.	43282	39908	44497
Nombre de marques ciblées par les campagnes d'hameçonnage.	364	352	363

Tableau 1.1. Statistiques sur l'hameçonnage pour le 2ème trimestre 2020 [2]

Les membres contributeurs de l'APWG signalent les URL d'hameçonnage dans l'APWG et étudient la nature et les techniques en constante évolution de la cybercriminalité. L'APWG suit le nombre de sites Web d'hameçonnage uniques, une

mesure principale de l'hameçonnage à travers le monde. Ceci est déterminé par les URLs de base unique des sites d'hameçonnage. Un seul site d'hameçonnage peut être annoncé comme des milliers d'URLs personnalisées, toutes menant essentiellement à la même destination d'attaque. Le nombre total de sites d'hameçonnage détectés au deuxième trimestre de 2020 était de 146 994. C'est une baisse de 11% par rapport aux 165772 du premier trimestre 2020 [2].

1.4.4 Les types d'attaques d'hameçonnage

Nous discutons dans cette sous-section les attaques les plus fréquemment utilisées actuellement. Leurs principales différences résident dans l'approche utilisée et dans le public ciblé. Les pirates trompent les utilisateurs, en utilisant l'ingénierie sociale pour apprendre le comportement en ligne et les préférences des victimes potentielles. Nous allons énumérer les types d'attaques d'hameçonnage les plus courants.

a) Usurpation de courrier

C'est l'un des types d'hameçonnage les plus faciles à utiliser pour obtenir des données d'utilisateurs à leur insu. Cela peut être fait de différentes manières : envoi d'un courrier électronique via un nom d'utilisateur connu ou l'envoi d'un courrier électronique imitant vos supérieurs et demandant des données importantes, ou bien usurper l'identité d'une organisation et demander aux employés de partager des données internes.

b) Cible de masse – usurpation de marque

C'est une attaque d'hameçonnage où des courriels sont envoyés à un groupe de personnes partageant un intérêt commun, en fonction de leurs préférences de marque, de leurs données démographiques et de leurs choix. Dans les attaques d'hameçonnage en masse, les e-mails envoyés aux victimes potentielles sont des clones d'e-mails transactionnels tels que des reçus, des rappels de paiement ou des cartes-cadeaux.

c) URL d'hameçonnage

Dans ces attaques, les fraudeurs utilisent l'URL de la page d'hameçonnage pour infecter la cible. Cela a un taux d'ouverture plus élevé parce que les gens sont assez

«*sociaux*» pour cliquer sur les liens envoyés par des inconnus, aussi ils sont prêts à accepter les demandes et les messages d'amis-liens MD (Message Direct) ou notifications par courrier électronique, ils sont même prêts à partager leurs emails et leurs cordonnées.

d) Lien caché

Un moyen de tromper quelqu'un avec de l'hameçonnage consiste à utiliser un lien caché. Nous avons tous reçu des courriers électroniques contenant la phrase d'action « CLIQUEZ ICI », « TÉLÉCHARGEZ MAINTENANT » ou « S'ABONNER » [3].

e) URL minuscule

Une autre façon de masquer les liens d'hameçonnage consiste à utiliser des outils de raccourcissement de liens comme *TinyURL* pour raccourcir l'URL et la rendre authentique [3].

f) URL mal orthographie

Dans ce type d'attaques, les pirates informatiques achètent des noms de domaine similaires aux sites Web populaires. Ensuite, ils piratent les utilisateurs en créant un site Web identique, où ils demandent aux cibles de se connecter en soumettant des informations personnelles [3].

g) Attaque d'homographe

Les attaques homographiques impliquent l'utilisation de mots d'apparence similaire - caractères ou groupes qui peuvent être facilement mal interprétés [3].

h) Attaque de sous-domaine

Ces types d'attaques s'adressent à des personnes non connaisseuses où les arnaqueurs exploitent le manque de compréhension de la différence entre un domaine et un sous-domaine pour lancer des attaques d'hameçonnage. Concernant cette attaque elle commence par le fait d'envoyer un e-mail à un utilisateur appartenant à une organisation donnée, per exemple www.organisationname.com, il peut aussi s'agir d'un identifiant de messagerie d'un collègue, pseudonyme@gmail.com, l'e-mail par la

suite, lui demande de cliquer sur le lien indiqué : www.organisationname.support.com et de se connecter pour accéder aux données afin de générer un rapport urgent, l'utilisateur clique sur le lien et termine par compromettre ses informations d'identification. Nous pouvons dire dans ce cas que cette technique est difficile à repérer car en effet, n'importe quel domaine connu peut être utilisé en tant que sous-domaine et le pire, la plupart des gens peuvent ne pas être conscients de la différence entre un domaine et un sous-domaine [3].

i) Messages contextuels : hameçonnage en session

Les messages contextuels sont le moyen le plus simple de mettre en œuvre une campagne d'hameçonnage. Avec les messages contextuels, les attaquants obtiennent une fenêtre qui leur permet de voler des informations de connexion en les redirigeant vers un faux site Web. Cette méthode d'hameçonnage est également appelée « *hameçonnage de session* » [3].

j) Usurpation de site Web

L'usurpation de site Web est similaire à l'usurpation d'adresse électronique, mais l'attaquant doit travailler plus dur, il publie un site Web en copiant la conception, le contenu et l'interface utilisateur d'un site Web légitime, il utilise également des outils de réduction d'URL pour créer une URL similaire pour le faux site [3].

k) Scripting

Étant donné que la plupart des pages Web sont écrites avec *JavaScript*, il devient plus facile pour les pirates de lancer une attaque de script. Le script ou le script intersites (XSS) utilise des scripts malveillants qui sont publiés sur l'ordinateur ou le téléphone de la victime par e-mail. Les pirates informatiques infectent un script de site Web légitime que vous visitez régulièrement et déterminé par l'ingénierie sociale, ce script va vous rediriger vers une page d'hameçonnage avec. Lorsque la page sera chargée, le script va s'exécuter et l'attaque se produit à l'insu de la victime [3].

l) Image d'hameçonnage

Les pirates utilisent des images et d'autres multimédias pour transmettre des fichiers de traitement par lots et des virus. Il existe deux façons d'inclure une image d'hameçonnage frauduleuse dans un e-mail. La première consiste à lier l'image d'une victime directement en tant qu'attaque par e-mail massive et la seconde consiste à utiliser une image cryptée (*.jpeg*) ou d'autres fichiers multimédias tels que des fichiers de chanson (*.mp3*) ou vidéo (*.mp4*) ou Fichiers GIF (*.gif*). L'attaquant inclurait un fichier de commandes (*.bat*) ou un virus dans une image et l'enverrait en pièce jointe à la victime. L'ordinateur ou le téléphone de la victime est infecté lorsque l'utilisateur télécharge cette voix et que le fichier de commandes et le virus sont téléchargés [3].

1.5. Les victimes de l'hameçonnage

Les attaques d'hameçonnage s'appuyant sur l'ingénierie sociale se servent de la crédulité des personnes pour accéder à des données confidentielles ou les forcer à réaliser des opérations spécifiques pour obtenir un versement d'argent. Il est possible d'affirmer que n'importe quel citoyen recourant à tout type de services en ligne sur Internet est susceptible d'être une victime potentielle d'attaque d'hameçonnage.

A plus large échelle, de la PME à la multinationale, indépendamment de son secteur d'activité, en passant par les organisations et gouvernements, tous sont désormais considérés comme des cibles possibles. Ainsi, toute entité susceptible d'enrichir les escrocs par ses fonds ou ses informations est une victime potentielle [3].

Pour les attaques destinées à infiltrer ces grandes structures, le pirate doit procéder à des attaques extrêmement ciblées pour gagner la confiance de ses interlocuteurs.

1.6. Caractéristiques du site d'hameçonnage

Après avoir étudié le concept de l'hameçonnage, nous allons citer quelques caractéristiques de ce type de sites :

L'utilisation des logos des sociétés célèbres, les similitudes visuelles entre le site réel de la compagnie et une partie du faux site, l'utilisation de codes JavaScript pour cacher

l'adresse IP dans l'URL, prendre le service de réduction d'URL comme URL minuscule, l'utilisation d'un numéro de port différent plutôt que ceux par défaut, tels que 8080 au lieu de 80, l'emploi du certificat SSL pour indiquer la sécurité, en utilisant par la suite le protocole HTTPS afin d'attirer l'attention des utilisateurs, il y a aussi l'exigence de l'installation de certains logiciels pour accéder au site .

On trouve aussi la barre d'adresse fausse qui montre l'URL de la véritable compagnie pendant que vous êtes sur le site hameçonnage, l'utilisation d'un menu Pop-Up et réduction immédiate sur le site réel de l'entreprise après l'achèvement du vol d'informations, dans un tel cas, l'attaquant profite du laps de temps, puisque l'utilisateur peut ne pas remarquer immédiatement les changements sur le site réel. Enfin, l'utilisation de texte caché qui peut être trouvé dans le code source de la page Web afin d'éviter des solutions de filtrage à base de texte. Ces caractéristiques sont les plus courantes mais des fonctionnalités supplémentaires peuvent être révélées par de nouvelles attaques d'hameçonnage [4].

1.7. Anatomie de l'hameçonnage URL

Avant de comprendre comment les attaquants procèdent lorsqu'ils créent un domaine d'hameçonnage, nous allons présenter la structure de l'URL. La figure 1.3 montre qu'une URL se compose de plusieurs parties distinctes.

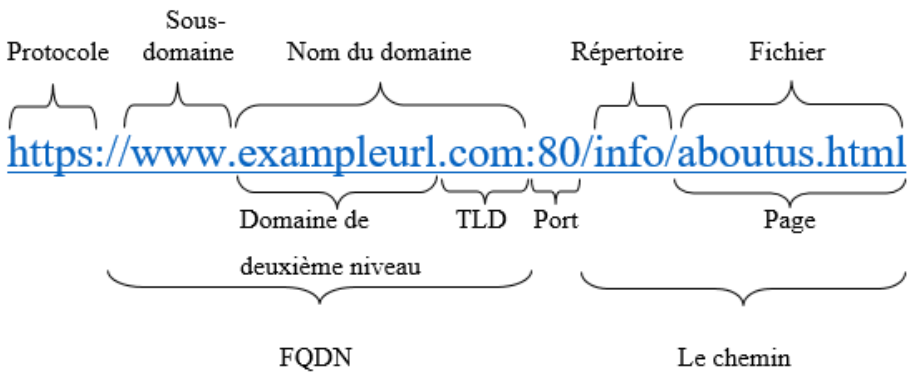


Figure 1.3. La structure de l'URL [3]

L'URL commence par le protocole utilisé pour accéder à la page. Dans la deuxième partie, il s'agit du nom de domaine complet qui identifie le serveur qui héberge la page Web, cette partie se compose d'un nom de domaine enregistré ce que nous appelons «*domaine de deuxième niveau*» et d'un suffixe que nous appelons «*domaine de premier niveau (TLD pour Top Level Domain)* », en fait, cette partie est limitée car elle doit être enregistrée auprès d'un registraire de nom de domaine.

La troisième partie, comporte le nom d'hôte, il se compose d'un nom de sous-domaine et d'un nom de domaine. D'ailleurs, l'hameçonnage a un contrôle total sur les parties du sous-domaine et peut lui attribuer n'importe quelle valeur. L'URL peut également avoir un chemin d'accès et des composants de fichier qui, eux aussi, peuvent être modifiés par l'hameçonnage à volonté.

Le nom et le chemin du sous-domaine sont entièrement contrôlables par l'hameçonnage. Nous utilisons le terme *FreeURL* pour faire référence à ces parties de l'URL.

En ce qui concerne le travail de l'attaquant, il se fait par le fait d'enregistrer n'importe quel nom de domaine qui n'a pas été enregistré auparavant, l'attaquant peut modifier le *FreeURL* à tout moment pour créer une nouvelle URL. La raison pour laquelle les défenseurs de la sécurité ont du mal à détecter les domaines d'hameçonnage est due à la partie unique du domaine du site Web. Lorsqu'un domaine est détecté comme frauduleux, il est facile d'empêcher ce domaine avant qu'un utilisateur n'y accède.

Certaines sociétés de renseignement sur les menaces détectent et publient des pages Web ou des adresses IP frauduleuses sous forme de listes noires, ce qui permet d'empêcher plus facilement les actifs nuisibles. L'attaquant dans ce cas doit choisir intelligemment les noms de domaine car l'objectif doit être de convaincre les utilisateurs, puis de paramétrer FreeURL pour rendre la détection difficile [3].

Dans le cas de l'hameçonnage, l'attaquant utilise principalement des méthodes importantes pour augmenter la vulnérabilité des victimes telles que l'utilisation de caractères aléatoires, l'utilisation combinée de mots, le cybersquattage, le typosquattage, etc.

Le *Cybersquattage* (également connu sous le nom de squat de domaine), c'est l'enregistrement, le trafic ou l'utilisation d'un nom de domaine avec l'intention

malveillante de profiter de la bonne volonté d'une marque appartenant à quelqu'un d'autre.

Par exemple, le nom de notre entreprise est *abcompany* et nous nous inscrivons en tant que abcompany.com. Les attaquants peuvent alors enregistrer abcompany.net, abcompany.org, abcompany.biz et ils peuvent l'utiliser à des fins frauduleuses.

Le *typosquattage*, également appelé détournement d'URL, est une forme de cybersquattage qui repose sur des erreurs typographiques commises par les internautes lors de la saisie d'une adresse de site Web ou sur la base d'erreurs typographiques difficiles à remarquer lors d'une lecture rapide. Un exemple célèbre de typosquatting est goggle.com, un site Web extrêmement dangereux. Un autre exemple similaire est *yutube*.com, qui est similaire à *goggle*.com, sauf qu'il cible les utilisateurs de Youtube. De même, www.airfrance.com a été typosquatté comme www.arifrance.com, détournant les utilisateurs vers un site Web proposant des voyages à rabais. Quelques autres exemples sont : paywpal.com, microroft.com, applle.com, et appie.com.

1.8. Conclusion

Nous avons présenté les notions de bases indispensables à notre travail de recherche, à savoir la communication, la cybercriminalité et l'arnaque et nous avons terminé cette partie par présenter le phénomène d'hameçonnage comme un vrai danger sur internet qui a créé un environnement de doute et de peur parmi les acheteurs en ligne. En fait, le nombre de sites d'hameçonnage est en augmentation continue, de nombreuses personnes ont été victimes, et de nombreuses organisations ont vu leurs noms se détériorer.

Diverses approches ont été élaborées pour résoudre le problème de la détection d'URL malveillants. Ces techniques sont principalement classées comme basés sur l'heuristique, l'apprentissage automatique, les listes noires d'hameçonnage et les listes blanches et à base de similarité visuelle. Nous exposerons ces méthodes en détails dans le chapitre suivant.

CHAPITRE II

LES TECHNIQUES DE DETECTION D'HAMEÇONNAGE

CHAPITRE II.

LES TECHNIQUES DE DETECTION D'HAMEÇONNAGE

L'hameçonnage est un concept lié au « *Social engineering* » qui consiste à présenter, sous forme de simulation, des contenus de Web factices dans le but de dérober aux internautes leurs informations personnelles : mot de passe de site bancaire, photographies, etc. D'un point de vue de droit, cette technique malveillante vise à pénétrer dans la vie privée des autres pour récupérer leurs coordonnées d'une manière illégale. Par conséquent, la loi incrimine cette pratique et punit sévèrement les «*Hackers*» qui s'y adonnent. Dans ce sens, il semble nécessaire que l'on puisse mettre à la lumière du jour les différentes approches qui visent à sensibiliser les utilisateurs d'Internet aux dangers de ce fait.

La première approche cible en premier lieu la formation des utilisateurs afin qu'ils détectent facilement la nature de tout contenu frauduleux et malveillant ayant pour dessein l'usurpation de leurs identités. Ensuite, la deuxième approche a un aspect plus pratique dans la mesure où l'on pourrait concevoir des logiciels classificateurs automatisés protégeant les internautes.

Le présent chapitre examinera, d'une part, au doigt et à l'œil les différentes approches de classification dans le but d'exposer leurs avantages et leurs inconvénients. D'autre part, nous étudierons au fil de ce qui reste les techniques de classification utilisées récemment dans les travaux de plusieurs chercheurs s'intéressant à l'hameçonnage.

2.1. Les techniques de classification

De nos jours, les desseins d'hameçonnage s'avèrent plus sophistiqués vu qu'ils développent des stratégies inédites et cela complique la tâche des internautes qui se prennent facilement en piège. Actuellement, les spécialistes comptent des milliers, voire des millions de sites de l'hameçonnage sur la toile électronique identiques à ceux d'origine. Les utilisateurs se retrouvent souvent dans des situations peu ou prou

compliquées car devant une telle nouveauté de fraude ingénieuse, ils sont incapables de détecter le contenu malveillant. Par conséquent, ils divulguent leurs informations personnelles, leurs coordonnées bancaires ce qui facilite la tâche aux pirates à récupérer très rapidement ces bases de données via une technologie de suivi telle que le Cheval de Troie. Certes, les compagnies d'antivirus ont conçu une protection anti-hameçonnage contre de telles menaces mais les « *Hackers* » eux génèrent plusieurs pages d'une vitesse vertigineuse, moult pages d'hameçonnage ne sont pas bloquées.

Dans la même optique, plusieurs approches ont présenté des stratégies pour résoudre ce problème de détection des pages malveillantes. Les techniques proposées présentent une panoplie de paramètres basés sur les heuristiques, l'apprentissage automatique, les listes noires d'hameçonnage, les listes blanches et les bases de similarité visuelle [5]. Dans ce qui suit, nous tenterons d'étudier chacune de ces techniques que l'on retrouve dans les méthodes de classification, à savoir :

- Les listes noires,
- Les heuristiques,
- La similitude visuelle,
- L'apprentissage automatique

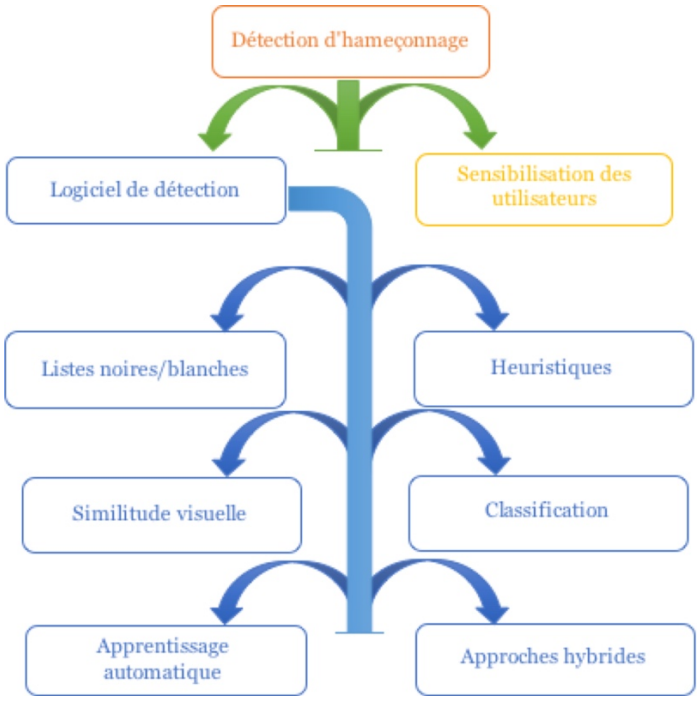


Figure 2.1. Méthodes de détection d'hameçonnage [26].

2.2. Les listes noires

Comme leur qualificatif l'indique, ces listes se caractérisent par un côté malveillant vu qu'ils régénèrent des URLs liées à des sites malveillants. Chaque fois qu'un navigateur accède à une page, il interpelle la liste noire pour vérifier si l'URL visitée figure dans ladite liste [5].

2.3. Les heuristiques

Un heuristique renvoie à l'application des règles apprises par l'expérience aux décisions subjectives. Il se distingue par un système bien conçu qui peut continuellement appliquer de mieux en mieux ses règles, en modifiant légèrement les paramètres au fur et à mesure de leur obtention de l'information. Par ailleurs, les technologies heuristiques jouent un rôle crucial dans les stratégies de détection des virus, voire les programmes malveillants. Dans cette même optique, les chercheurs essayent de comprendre le fonctionnement des sites Web d'hameçonnage, ils peuvent manœuvrer même à détecter les attaques basées sur les diverses fonctionnalités d'hameçonnage : les noms de domaines, les fautes orthographiques, la source des images, les liens, etc. [7].

2.4. La similitude visuelle

Les approches qui se servent de cette technique visent à comparer l'apparence visuelle du site Web suspect à celle du site Web légitime tout en utilisant plusieurs paramètres. D'abord, on enregistre des captures d'écran du site Web suspect dans la base de données. Si ladite capture d'écran saisie est similaire à celle de la base de données, elle sera considérée comme une page d'hameçonnage. En revanche, s'il existe plusieurs sites Web identiques, le premier site web saisi sera considéré comme légitime [5].

2.5. Apprentissage automatique

Lors de la seconde moitié du XX^{ème} siècle, les prémices définitionnelles de l'apprentissage automatique étaient attribuées à l'un des pionniers de l'intelligence artificielle, il s'agit d'Arthur Samuel [10]. Les diverses approches considèrent «*l'apprentissage de la machine tel un domaine d'étude qui permet à l'appareil*

d'apprendre sans être programmée explicitement. » [10]. Ce concept révolutionnaire, *Machine Learning* en anglais, est un sous-ensemble de l'intelligence artificielle, qui développe une capacité d'apprendre de ses expériences passées. Il entame un processus d'observation en vue de repérer les données afin de lister les différents modèles permettant au système de prendre des décisions opérationnelles appropriées lors des prochaines entrées de données que les développeurs pourront fournir. La tâche primordiale serait de programmer les ordinateurs à apprendre automatiquement sans aucune intervention ou assistance humaine, voire à ajuster les résultats en guise d'interprétations [8].

L'apprentissage automatique est un processus efficace pour l'analyse des méga-données, de même pour le traitement de l'ensemble de données complexes. En outre, il peut aussi repérer les nouveaux comportements au sein d'une organisation dans le but de proposer des suggestions similaires à d'autres, à titre d'exemple, quelles sources de données utiliser pour l'analyse ou quel contenu analytique serait pertinent comme réponse aux questions posées. Les deux principales composantes de la conception des modèles d'apprentissage et classification sont à l'origine les techniques issues de la statistique multivariée ou bien des techniques d'apprentissage automatique. Il faudra mentionner que l'apprentissage automatique est un outil très utilisé pour lutter contre les attaques d'hameçonnage. Dans ce cadre, nul ne peut nier que les algorithmes d'apprentissage automatique ont été l'une des remarquables techniques de détection des sites Web d'hameçonnage. Cette technique utilisée dans divers domaines d'application contient une vaste liste de classes de problèmes, de façon qu'il permet une diversité dans la méthode d'apprentissage automatique. De surcroît, ces approches essayent d'analyser les données d'une URL et de ses pages web qui y correspondent, en extrayant de meilleures représentations de l'URL, et en régénérant un modèle de prédiction sur les données d'apprentissage des URLs malveillantes et non malveillantes [3]. La figure 2.2 montre une classification des techniques d'apprentissage automatiques existantes.

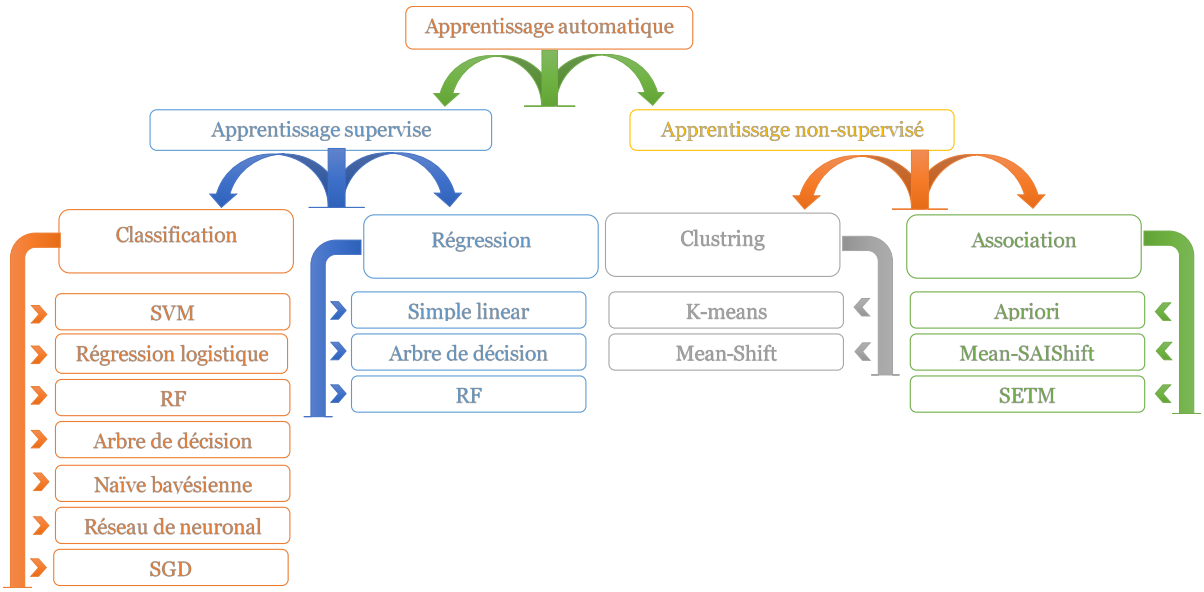


Figure 2.2. Classification des techniques d'apprentissage automatique [3].

2.5.1. Apprentissage supervisé

L'apprentissage automatique supervisé conçoit un modèle qui génère des prédictions basées sur des preuves certaines. Autrement dit, un algorithme d'apprentissage supervisé met en jeu un ensemble connu de données d'entrée et de réponses connues aux (données de sortie) et forme un modèle afin de générer des prédictions raisonnables pour la bonne valeur cible aux nouvelles données. L'usage adéquat du modèle d'apprentissage supervisé se fait en corrélation avec les données connues pour la sortie que l'utilisateur essaie de prédire. Ce modèle d'apprentissage artificiel connaît plusieurs techniques, nous en citons une [3]:

1. L'apprentissage supervisé consiste en des variables d'entrée (x) et une variable de sortie (Y). Nous utilisons alors un algorithme pour apprendre la fonction de mapping de l'entrée (x) à la sortie (Y) : $Y = f(X)$
2. Le but est d'appréhender si bien la fonction de mapping que, lorsque nous avons de nouvelles données d'entrée (x), nous pouvons prédire les variables de sortie (Y) pour ces données.

En guise de processus fonctionnel, il est montré que l'apprentissage supervisé se base sur les techniques de classification et de régression pour développer des modèles prédictifs.

a) La classification

La classification supervisée se base surtout sur le fait qu'il existe une classification préalable de quelques données en entrée, autrement dit, on dispose d'un ensemble de données déjà classées que l'on appelle « Ensemble d'apprentissage » et que l'on utilise comme base, pour classer le reste des données. En plus, on tente dans ce genre de classification de trouver le plus d'informations à partir des ensembles d'apprentissage, pour permettre un meilleur listage des données restantes. Les techniques de classification prédisent des réponses discrètes. Les modèles de classification répertorient les données d'entrée en catégories. Les applications typiques incluent l'imagerie médicale et la reconnaissance vocale. L'usage de la classification se fait par l'utilisateur qui choisit d'étiqueter, catégoriser ou séparer en groupes ou classes spécifiques ses propres données [38].

b) La régression

Les techniques de régression fonctionnent de façon à prédire les réponses continues, l'exemple le plus significatif sont les changements de température ou les fluctuations de la demande d'énergie, ainsi que les applications typiques qui comprennent la prévision de la charge électrique. Dans ces cas pratiques, l'utilisateur veille à manœuvrer les techniques de régression pour travailler avec une panoplie de données dont la nature de la réponse obtenue est un nombre réel [38].

2.5.2. Apprentissage non-supervisé

L'apprentissage non supervisé consiste à découvrir des similarités entre les observations dans un recueil d'exemples, dans l'intention de regrouper celles-ci en sous-catégories, appelés clusters ou classes. La technique employée vise à traduire des algorithmes dans le but de rapprocher les exemples qui se ressemblent et écarter ceux qui ont le moins de caractéristiques communes. D'ailleurs, le qualificatif « *non supervisé* » semble bien représentatif dans la mesure où le modèle n'est pas guidé ou

orienté par un agent. Son fonctionnement est de trouver des modèles cachés et même des structures intrinsèques dans les données. On l'utilise pour repérer des inférences à partir d'ensemble d'informations composés de données d'entrée sans réponses étiquetées. Donc, les données fournies pour ce genre de modèle ne retiennent aucune étiquette ou classe cible. Les comportements (classes, catégories) des données d'apprentissage ne sont pas repérés et cela reste encore une problématique à soulever [38].

a) Clustering

Le clustering est la technique d'apprentissage non supervisée la plus utilisée. Il fonctionne pour l'analyse exploratoire des données afin de trouver des modèles ou des regroupements discrets dans les données. Dans la même sens, les applications de l'analyse en grappes contiennent l'analyse de séquences de gènes, les études de marché et la reconnaissance d'objets [38]. Cette technique semble être définie comme l'affectation d'une panoplie d'observations en sous-catégories (appelés clusters) de sorte que les observations dans le même cluster soient identiques dans un certain sens. Le clustering est une méthode d'apprentissage non supervisée et une technique courante d'analyse des données statistiques utilisée dans plusieurs domaines [3].

b) L'association

Les normes et les règles d'association permettent d'établir des analogies entre des éléments de données dans de grandes bases de données. Cette technique non supervisée vise à identifier des relations fictives entre les variables dans les grandes bases de données. Cette règle montre la fréquence à laquelle un jeu d'élément se produit dans une transaction. A titre d'exemple typique, c'est l'analyse basée sur le marché. Par ailleurs, l'on reconnaît plusieurs algorithmes de recherche de règles d'association qui facilitent la trouvaille et l'extraction des informations pertinentes stockées dans de gigantesques bases de données. Notons que les règles d'association fonctionnent dans plusieurs applications. Citons alors le domaine du commerce électronique où l'on utilise ce type de méthodes pour sélectionner les produits à mettre en promotion, afin d'organiser les catalogues des articles, ainsi que pour bâtir des catalogues personnalisés pour chaque client. Autres exemples peuvent toucher aussi le

domaine médical dans lequel les règles d'association sont utilisées pour préciser le traitement efficace qui correspond à un ensemble de symptômes.

Le tableau 2.1 illustre de façon détaillée les avantages et les inconvénients des deux techniques en question :

Technique	Technique d'apprentissage automatique supervisé	Technique d'apprentissage automatique non-supervisée
Avantages	• Les variables d'entrée et de sortie seront données. • Les données étiquetées aident à la formation des données d'entrée des algorithmes. • La simplicité de la méthode d 'apprentissage supervisé. • L'utilisation des données de formation pour l'apprentissage du lien entre les entrées et les sorties. • La méthode très précise et fiable (Exactitude des résultats). • Les réponses sont capable d'être connues grâce à la connaissance des classes utilisées.	• Dans le modèle d'apprentissage non supervisé, seules les données d'entrée seront fournies et n'utilisent pas de données de sortie. • La facilite de l'obtention des données d'entrée des algorithmes sans étiquettes. • Il y a une Exactitude des résultats où La méthode se déroule en temps réel.
Inconvénients	• L'apprentissage supervisé peut subir des difficultés dans la classification des méga-données. • La méthode d'apprentissage supervisée est plus complexe que les méthodes non supervisées, elle se déroule hors ligne et prend beaucoup de temps pour le calcul.	• Il n'a y pas de précision d'information sur le tri des données et la sortie. • Le nombre de classes n'est pas connu

Tableau 2.1. Comparaison des approches supervisées et non-supervisées [39].

D'après ce tableau représentatif, on remarque bien que les deux méthodes d'apprentissage ont une place capitale dans l'exploration des données. L'usage de l'une ou de l'autre méthode dépend étroitement des besoins et des problèmes à résoudre.

2.6. Méthodes d'approche supervisée

Les algorithmes de régression et de classification sont des algorithmes d'apprentissage supervisé. Chacun est utilisé pour des tâches de prédiction dans l'apprentissage automatique et fonctionnent via les ensembles de données étiquetées. Cependant, les deux semblent distincts même quand il s'agit de la façon dont ils sont utilisés dans divers problèmes d'apprentissage automatique. La distinction capitale entre les algorithmes de régression et de classification réside dans le fait que les algorithmes de régression sont utilisés dans les prédictions des valeurs continues telles que le prix, le salaire, l'âge, etc. Alors que les algorithmes de classification fonctionnent d'une façon particulière afin de prédire ou classer les valeurs discrètes à l'exemple du masculin ou féminin, vrai ou faux, spam ou non spam, etc. L'identification du type de sortie attendue d'une application se passe grâce au choix d'un l'algorithme d'apprentissage automatique [10].

Quelques algorithmes s'utilisent à la fois pour classifier et régresser en fonction de certaines petites modifications, à l'image des arbres de décision et des réseaux de neurones artificiels. D'autres algorithmes ne peuvent pas être facilement utilisés pour les deux types de problèmes, tels que la régression linéaire pour la modélisation prédictive par régression et la régression logistique pour la modélisation prédictive par classification [39].

2.6.1. Les algorithmes d'apprentissage automatique par classification

Lorsque l'on évoque l'apprentissage automatique, on parle également de la classification qui est une approche d'apprentissage supervisée dans laquelle le programme informatique poursuit un apprentissage à travers des données saisies, ensuite utilise ce processus pour classer une nouvelle observation. Cet ensemble de données peut simplement être bi-classe ou il peut aussi être multi-classe [39].

L'approche de classification comprend plusieurs types d'algorithmes, nous en citons quelques-unes :

- Classification linéaire : régression logistique, Classification naïve bayésienne.
- Machine à vecteurs de support (SVM).
- Arbre de décision.
- Foret d'arbres décisionnels (Random Forest).
- Réseau neuronal.
- Voisin le plus proche.

Dans ce qui suit, nous présentons chaque méthode succinctement tout en procédant par une approche comparative.

a) Classification naïve bayésienne

Les méthodes Naïves Bayes sont listées parmi les modèles probabilistes les plus connus. Elles se basent principalement sur le théorème de Bayes. Les algorithmes Naïves Bayes sont utilisés dans la catégorisation et la classification de documents. Cette technique permet d'estimer la probabilité de chaque classe parmi les exemples trouvés, et donner une étiquette à la classe la plus probable [3].

b) Régression logistique

Cette technique est utilisée lorsque la variable dépendante est binaire. Il s'agit d'une méthode de référence pour les problèmes de classification binaire dans le domaine statistique. Les modèles de régression logistique produisent la probabilité qu'une variable ait une valeur donnée. A titre d'exemple, au lieu de prédire s'il est risqué ou non d'accorder un crédit à un client, cette méthode essaye d'estimer la probabilité pour que la décision soit favorable ou non. La régression logistique est utilisée seulement si la variable à prédire est de type binaire (ne peut prendre que deux valeurs), elle cible donc la classification [11].

c) Arbre de Décision

L'arbre de décision est une technique de base de l'analyse de données. Il a été utilisé dans plusieurs domaines comme l'apprentissage automatique, la reconnaissance de

forme, le diagnostic médical, etc. L'arbre de décision connaît moult algorithmes qui génèrent une structure arborescente où chaque nœud interne indique un test d'un attribut. Chaque branche illustre le résultat du test et chaque nœud feuille contient une étiquette de classe. L'arbre de décision vise à partitionner récursivement l'espace d'attributs au point où tous les cas soient complétement partitionnés en sous-ensembles qui ne se croisent pas [3].

d) Forêt d'arbre décisionnels

Parmi les méthodes d'apprentissage il y a la méthode de la forêt aléatoire qui utilise plusieurs algorithmes pour obtenir une meilleure classification. Les données sont entrées au sommet dans lequel l'algorithme commence toujours par un « *arbre de décision* » (c-à-d un graphique de type arborescence ou un modèle de décisions). Pour la segmentation des données en petits ensembles par des variables spécifiques, il faut qu'ils aient parcouru à travers l'arborescence [3].

e) Voisin le plus proche

Les KNN ou « *K-nearest neighbours* » en anglais, est une approche de classification supervisée. Elle utilise une métrique de distance comme la distance Euclidienne, Hamming, etc. pour la classification de nouveaux points de données, et tout cela par l'examination des K points de données de l'apprentissage qui se rapprochent le plus de celui-ci dans l'entrée ou dans l'espace des fonctions. La conséquence indésirable de cette approche est qu'elle est incapable d'analyser des données volumineuses pour un ensemble d'apprentissage qui nécessite un large espace de stockage [9].

f) SVM

SVM, ou « *Support Vector Machine* » en anglais, est un algorithme d'apprentissage automatique supervisé utilisé en particulier pour les problèmes de classification. Cet algorithme place toutes les données sous forme de points dans une zone de *n* dimensions et les catégorise avec l'hyperboloïde qui distingue les classes. Il est très efficace pour traiter les données en masse et ceci au moyen de vecteurs de support pour la prédiction [3].

g) Réseaux de neurones

Les réseaux de neurones artificiels sont l'un des principaux outils utilisés dans l'apprentissage automatique. Ce sont des structures inspirées des circuits nerveux du système nerveux humain et se composent d'un groupe d'unités informatiques interconnectées. Chaque neurone envoie et reçoit des impulsions d'autres neurones, et ces changements sont utilisées pour prédire la classe des données en entrée [3].

h) SGD

SGD signifie « *descente de gradient stochastique* ». Le gradient estime chaque échantillon à la fois et le modèle est mis à jour en cours de route avec un programme de force décroissante [31].

Le tableau 2.2 représente les avantages et les inconvénients des différentes méthodes d'apprentissage supervisé :

	Les avantages	Les inconvénient
Naïves Bayes	• L'implémentation se fait d'une manière simple, facile et rapide par rapport aux autres méthodes sophistiqués.	• Ces performances sont limitées quand il s'agit d'une grande quantité de données d'apprentissage. • Il est considéré comme une mauvaise estimation.
Régression logistique	• Facile et rapide à comprendre, à entrainer et prévoir • Efficace pour la résolution des petits problèmes de données de classification	• Il y a un petit manque de précision. • Susceptible d'être utilisé seulement pour les données linéaires. • Ne peut pas être adaptée pour des données complexes • Le modèle finit parfois par sur-adapter
Arbre de décision	• Facilité d'utilisation et d'apprentissage grâce à la capacité à travailler sur des données symboliques.	• Très sensible au changement des données (bruit). • Les interactions entre les variables sont difficiles à détecter. • La construction de l'arbre va-t-être recommencée lors du changement des données. • Instabilité

Foret d'arbre décisionnelle	• Reconnaissance très rapide • Efficace sur inputs de grande dimension • Elle a la capacité de réduire le temp de calcul et d'obtenir diverses variétés de modèles.	• L'opération d'Apprentissage est lente • Algorithme complexe
SVM	• Leur capacité à manipuler de grandes quantités de données. • Ces méthodes utilisent un petit nombre d'hyper paramètres.	• La classification des corpus nécessite l'utilisation des fonctions mathématiques complexes. • Cet algorithme demande beaucoup de temps pendant les phases de test.
KNN	• Ce type d'algorithme est facile à utiliser. • Pour les classes réparties irrégulièrement et pour les données volumineuses, cet algorithme est très efficace.	• Il demande beaucoup de temps lors de la classification. • La grande capacité de stockage qu'elle nécessite pour le traitement des corpus.
Réseau de neurones	• L'opération de traitement des grands corpus se fait d'une façon très efficace et rapide. • Cet algorithme peut être combiné avec d'autres méthodes de classification. • La probabilité d'erreur est très faible en comparaison avec d'autres méthodes de classification. • Les réseaux de neurones ne nécessitent pas l'utilisation de modèles mathématiques très complexes pour leur fonctionnement.	• Le manque de possibilité de l'interprétation des résultats obtenus par la classification des réseaux de neurones ainsi que la convergence de ces résultats est incertaine. • L'algorithme est considéré comme une boîte noire. • En cas d'erreurs, il est impossible de déterminer la cause cette erreur.

Tableau 2.2. Avantages et inconvénients des méthodes d'apprentissage supervisé [31].

2.6.2. Les algorithmes d'apprentissage automatique par régression

La régression c'est une technique d'apprentissage automatique supervisée. Les modèles de régression sont utilisés pour prédire une valeur continue. L'un des exemples courants de la régression est : prédire les prix d'une maison compte tenu des

caractéristiques de la maison comme la taille, le prix, etc. [11]. Les algorithmes les plus courants de la régression sont :

- Régression linéaire simple.
- Régression de l'arbre de décision.
- Régression de Foret d'arbres décisionnels.

a) Régression linéaire simple

Il s'agit d'une technique statistique conçue pour prédire la valeur des variables continues. Sa finalité principale est de déterminer le meilleur modèle qui associe des variables de sortie quantitatives à plusieurs variables de prédicat d'entrée. C'est ce qu'on appelle l'adaptation du modèle aux données. Les modèles linéaires sont les plus couramment utilisés. C'est ce qu'on appelle la régression linéaire. L'équation de régression est une relation entre la variable à prédire et plusieurs autres variables prédictives [11].

b) Régression de l'arbre de décision

Les arbres de décision ne sont pas seulement utilisés pour la classification, ils peuvent également être utilisés pour la régression. En cas d'arbres de décision de régression, à chaque niveau, nous devons identifier l'attribut de division. L'algorithme peut être utilisé pour identifier le nœud de partition en réduisant l'écart type (quand le gain d'informations de classification est utilisé) [11].

c) Régression de la Foret d'arbres décisionnels

La forêt aléatoire est une méthode dans laquelle nous considérons la prédiction de plusieurs arbres de régression décisionnelle. On identifie *n,* où *n* est le nombre de régresseurs d'arbre de décision à créer. La valeur moyenne de chaque branche est attribuée à chaque nœud feuille dans l'arbre de décision [11].

2.7. Méthodes d'approche non-supervisé

Nous ne pouvons pas appliquer des méthodes d'apprentissage automatique sans les lier directement à des problèmes de régression ou de classification, contrairement à

l'apprentissage automatique non supervisé, car les valeurs de données des sorties ne sont pas connues et l'algorithme ne peut donc pas être entraîné. Cependant, l'apprentissage non supervisé peut être utilisé pour découvrir l'infrastructure de données. Cette approche contient deux techniques, la première est le clustring et la seconde est l'association, nous allons les détailler dans les sous sections ci-dessous [3].

2.7.1. Les algorithmes d'apprentissage automatique par regroupement

Le clustering est une technique d'apprentissage automatique non supervisé qui implique le regroupement de points de données (classer chaque point de données dans un groupe spécifique). Théoriquement, les points de données d'un même groupe devraient avoir des attributs et/ou caractéristiques similaires, tandis que les points de données de différents groupes devraient avoir des attributs et/ou des fonctions très différents. Il s'agit de la technique d'analyse statistique des données la plus couramment utilisée dans de nombreux domaines [40].

La méthode la plus courante du clustering est l'analyse de cluster, qui est utilisée pour l'analyse d'exploration de données afin de découvrir des modèles ou des clusters cachés dans les données. Le regroupement est modélisé à l'aide d'une mesure de similarité définie par une mesure telle que la distance euclidienne ou la distance de probabilité. Les algorithmes couramment utilisés sont [40]:

- Le clustering K-means .
- Mean-Shift Clustering.

a) K-means

Le clustering K-means est un algorithme permettant de classer ou de grouper des objets en fonction des attributs/fonctions des numéros de groupe K où K est un entier positif. Le regroupement est effectué en minimisant la somme des carrés des distances entre les données et le centre de gravité du cluster correspondant. Ainsi, le but de la classification K-mean est de classer les données [18].

b) Mean-Shift Clustering

Fait partie de la catégorie des algorithmes de clustering. Il consiste à attribuer les points de données aux clusters de manière itérative en décalant les points vers le mode (le mode est la densité de points de données la plus élevée de la région, dans le contexte du Meanshift). En tant que tel, il est également connu sous le nom d'algorithme de recherche de mode. L'algorithme de décalage moyen a des applications dans le domaine du traitement d'image et de la vision par ordinateur [18].

2.7.2. Les algorithmes d'apprentissage automatique par règles d'association

L'exploration de règles d'association est un processus qui vise à trouver des modèles fréquents, des associations ou des structures causales à partir d'ensembles de données trouvés dans divers types de bases de données (telles que des bases de données relationnelles), des bases de données de transactions et d'autres formes de référentiels de données. Étant donné un ensemble de transactions, l'exploration de règles d'association vise à trouver des règles qui nous permettent de prédire l'occurrence d'un élément particulier en fonction de l'occurrence d'autres éléments dans la transaction [18]. Les algorithmes les plus courants qui utilise des règles d'association sont AIS, SETM et Apriori et leurs variantes, certaines d'entre eux seront discutées ci-dessous.

a) Apriori

C'est est un algorithme pour l'exploration fréquente d'ensembles d'éléments et l'apprentissage de règles d'association sur des bases de données relationnelles. Il procède en identifiant les éléments individuels fréquents dans la base de données et en les étendant à des ensembles d'éléments de plus en plus grands tant que ces ensembles d'éléments apparaissent suffisamment souvent dans la base de données. Les ensembles d'éléments fréquents déterminés par Apriori peuvent être utilisés pour déterminer des règles d'association qui mettent en évidence les tendances générales de la base de données : cela a des applications dans des domaines tels que l'analyse du panier de marché [18].

b) AIS

Lors de l'analyse des données, des groupes d'objets sont créés et calculés. L'algorithme AIS dans les données de transaction spécifie de grands groupes d'objets qui contiennent une transaction, et les grands groupes d'objets sont développés avec d'autres éléments dans les données de transaction afin de créer de nouveaux groupes d'objets candidats [18].

Pour conclure, le tableau 2.3 représente la comparaison des différentes techniques utilisées pour la détection des sites web d'hameçonnage :

	Heuristique	**L'apprentissage automatique**	**Liste noire**
Implémentation cote client	Non	Non	Oui
Détection en temps réel	Oui	Oui	Non
Exige l'entrainement	Oui besoin de formation fréquente sur les nouvelles fonctionnalités avec le temps	Oui besoin de formation fréquente sur les nouvelles fonctionnalités avec le temps	Non
Exige des MAJ	Oui	Oui	Oui besoin de MAJ fréquentes de la liste noire et blanche
Calcul : cout de communication	Haut /Moyen	Haut /Moyen	Faible/élevé
Complexité de stockage	Haute. Besoin de stocker et de mettre à jour les datasets	Haute. Besoin de stocker et de mettre à jour les datasets	Moyen. La liste noire est stockée sur le serveur central
Avantage	Atténuer les attaques d'hameçonnage de zéro heure.	• Atténuer les attaques d'hameçonnage de zéro heure. • Construit ses propres modéles de classification.	• Requérant peu de ressources sur la machine hôte. • Efficacité lorsque le taux de FP minimaux sont requis. • Attaques d'hameçonnage de zéro heure.

Désavantage	Taux de PF plus élevé que la liste noire. Coût de calcul élevé.	• Long. • Chère. • Grand nombre de règles.	• Peut entrainer des requêtes excessives avec des serveurs fortement chargés. • Elle est statique, donc la détection de l'hameçonnage dépend juste de la liste déjà prédéfinie, si le URL du site n'existe pas dans la base il ne sera pas détecté.

Tableau 2.3. Comparaison des techniques de détection d'hameçonnage. [5]

L'étude de la littérature et la comparaison des différents techniques sont montrées que les techniques d'apprentissage automatique sont les plus prometteuses, mais elles ont aussi quelques désavantages comme le coût de calcul le plus élevé.

2.8. Conclusion

Dans ce chapitre, nous avons présenté les techniques de détection d'hameçonnage. Différents types de solutions ont été proposés. On constate que les solutions pour la détection de l'hameçonnage fonctionnent mieux que les solutions de prévention de l'hameçonnage et la formation des utilisateurs, car ils ne nécessitent pas de modifications des plates-formes d'authentification et ne dépendent pas de la capacité de l'utilisateur à détecter l'hameçonnage. Les solutions de détection de l'hameçonnage sont moins coûteuses que les solutions de prévention de l'hameçonnage en termes de gestion des mots de passe et des périphériques supplémentaires nécessaires. Cependant, il existe encore de nombreux défis dans la détection de l'hameçonnage.

CHAPITRE III

TRAVAUX CONNEXES SUR LA DETECTION DE L'HAMEÇONNAGE

CHAPITRE III.

TRAVAUX CONNEXES SUR LA DETECTION DE L'HAMEÇONNAGE

Des recherches sur les méthodes de détection de l'hameçonnage ont été effectuées. Nous présentons dans ce chapitre les méthodes basées sur les caractéristiques de l'URL, celles basées sur les caractéristiques de l'HTML, ainsi que celles basées sur l'URL et l'HTML à la fois. Un grand nombre d'articles utilisent une grande variété de caractéristiques. Malheureusement, il y a de nombreuses caractéristiques moins discriminatives et qui peuvent ralentir les performances du système.

3.1. Les approches basées sur les caractéristiques de l'URL

Les approches basées sur l'URL analysent diverses caractéristiques en fonction de l'URL cible, telles que la longueur de l'URL, la présence de caractères spéciaux dans l'URL, les caractéristiques de nom d'hôte telles que la présence de l'adresse IP, et d'autres informations pouvant être extraites de l'URL à l'aide de serveurs tiers, de moteurs de recherche ou de serveurs DNS. L'intuition de ces approches est solide, l'URL peut vraiment fournir des informations pertinentes qui peuvent être utilisées comme bonne indicateurs des attaques d'hameçonnage.

Verma et Das [25] se concentrent sur la création d'un système d'analyse et de classification d'URL pour détecter principalement les attaques d'hameçonnage. Ils considèrent l'analyse d'URL comme un excellent moyen de maintenir la distance entre l'attaquant et la victime, plutôt que de visiter le site Web et de gagner des emplois ils ont utilisé une classe unique de caractéristiques qui englobe d'autres caractéristiques lexicales : des N-grammes basés sur des caractères à partir d'URL et évaluons une gamme d'algorithmes d'apprentissage en ligne. ils ont obtenu des précisions très élevées avec la régularisation adaptative des vecteurs pondérés par poids et par

confiance. Le temps nécessaire pour évaluer le modèle sur l'ensemble de 90 000 URL a pris 2 secondes pour l'ensemble des uni-grammes, 6 secondes pour l'ensemble des bi-grammes et 30 secondes pour l'ensemble de tous les trigrammes. L'augmentation du temps semble raisonnable, car même pour l'ensemble de tous les trigrammes, ils ont pu traiter 3000 URL par seconde. C'est plus que suffisamment rapide pour être intégré à un système en temps réel, tel qu'un navigateur Web.

Sahingoz et al. [26] ont proposé un système anti-hameçonnage en temps réel, qui utilise sept algorithmes de classification différents et des caractéristiques basées sur le traitement du langage naturel (NLP). Le système présente les propriétés distinctives suivantes : indépendance linguistique, utilisation d'un grand dataset pour les sites d'hameçonnage et ceux légitimes, exécution en temps réel, détection de nouveaux sites Web, indépendance vis-à-vis des services tiers et utilisation de caractéristiques riches en caractéristiques. Pour mesurer les performances du système, un nouvel ensemble de données (dataset) est construit et les résultats expérimentaux y sont testés. D'après les résultats expérimentaux et comparatifs des algorithmes de classification mis en œuvre, l'algorithme forêt d'arbres décisionnels, avec uniquement des caractéristiques basées sur NLP donne les meilleures performances avec un taux de précision de 97,98% pour la détection des URLs d'hameçonnage.

Jagadeesan et coll. [20] utilisent uniquement des caractéristiques extraites directement de l'l'URL du site Web pour déterminer s'il s'agit d'un site Web d'hameçonnage. Par conséquent, on n'est pas obligé de visiter le site Web pour déterminer s'il s'agit d'un hameçonnage. Cela peut également empêcher les utilisateurs de visiter des sites Web d'hameçonnage et de s'exposer au code malveillant. Pour cela, ils ont utilisé l'algorithme de forêt d'arbres décisionnels car il a l'avantage de ne pas surcharger les données.

Dans [8], l'objectif était de créer une extension pour Chrome qui agira comme un middleware entre les utilisateurs et les sites Web malveillants et réduira le risque que les utilisateurs découvrent accidentellement ces sites. Ils ont remarqué que tous les ingrédients nocifs ne peuvent pas être collectés de manière exhaustive, car cela demande un développement de longue durée. Pour résoudre ce problème, ils ont utilisé des outils d'apprentissage automatique et ont classé chaque nouveau contenu qu'ils ont

rencontré dans des catégories spécifiques afin de pouvoir prendre les mesures correspondantes.

Jeeva et Rajsingh [22] se sont concentrés sur la recherche des caractéristiques nécessaires pour distinguer les sites Web d'hameçonnage des sites légitimes. Ils ont effectué une "*extraction de règles d'association*", ce qui peut aider à distinguer les deux. Ils ont utilisé l'algorithme *Apriori* pour définir des règles qui peuvent être utilisées pour identifier les sites Web d'hameçonnage. Cela a révélé de nombreuses caractéristiques de classification utiles, telles que le nombre de barres obliques, de mots-clés dans la partie du chemin d'URL, etc.

Lui et al. [23] ont proposé une méthode qui se concentre principalement sur les caractéristiques de fréquence des traits. Dans cette méthode, ils ont combiné l'analyse statistique des URL avec la technologie d'apprentissage automatique pour obtenir des résultats de classification des URL malveillantes. Ils ont également comparé six algorithmes d'apprentissage automatique pour vérifier l'efficacité de l'algorithme proposé. La précision de l'algorithme est de 99,7% et le taux de faux positifs est inférieur à 0,4%.

Prakash et al. [21] ont utilisé un algorithme qui divise l'URL en plusieurs parties et compare chaque partie avec les entrées de la liste noire. L'inconvénient de cette solution est que la liste noire ou la liste blanche nécessite un mécanisme de mise à jour continue.

Stiawan et Zaini [33] ont proposé l'intuition que le nom de domaine des sites Web d'hameçonnage est le signe révélateur d'hameçonnage et détient la clé d'une détection réussite d'hameçonnage. Ils se sont concentré sur cet aspect des sites Web d'hameçonnage et des caractéristiques de conception qui explorent la relation entre le nom de domaine et les éléments clés du site Web. Leurs travaux diffèrent de l'état de l'art existant car leurs ensembles de caractéristiques garantissent qu'il y a un biais minimal ou nul par rapport à un ensemble de données. Leurs modèles d'apprentissage sont entraînés avec seulement sept caractéristiques et ont atteint un taux de vrais positifs de 98% et une précision de classification de 97%, sur un échantillon de données. Par rapport aux travaux de pointe, le traitement et la classification par

instance de données sont 4 fois plus rapides pour les sites Web légitimes et 10 fois plus rapides pour les sites Web d'hameçonnage. Ils ont démontré les lacunes de l'utilisation des caractéristiques basées sur les URL, car elles sont susceptibles d'être biaisées en faveur de la collecte et de l'utilisation des ensembles de données. Ils ont montré la robustesse de leurs algorithmes d'apprentissage en testant leurs classificateurs sur des URL d'hameçonnage inconnues et ont obtenu une précision de 99,7% par rapport au meilleur résultat connu précédemment avec un taux de détection de 95%.

Chou et al. [34] ont développés Spoofguard, un plug-in de navigateur qui peut identifier les sites d'hameçonnage en vérifiant une série d'heuristiques. Cette méthode utilise à la fois une évaluation sans état (qui détermine les pages Web suspectes qui extraient certaines fonctions d'une page Web) et une évaluation avec état basée sur l'historique des visites précédentes de l'utilisateur. Cette méthode souffre du taux de fausses alarmes car elle enregistre les données passées de l'utilisateur.

3.2. Les approches basées sur les caractéristiques de l'HTML

Les approches basées sur l'HTML analysent diverses caractéristiques en fonction de contenu de la page web.

Jain et Gupta [29] ont proposé une nouvelle méthode anti-hameçonnage qui extrait uniquement les caractéristiques de la page web client. La méthode proposée est rapide et fiable car elle ne s'appuie pas sur des tiers, mais extrait uniquement les caractéristiques du code source de l'URL. La méthode proposée intègre diverses nouvelles caractéristiques spécifiques aux hyperliens pour détecter les attaques d'hameçonnage. Les caractéristiques spécifiques des hyperliens sont divisées en douze catégories différentes et ont été utilisées pour former des algorithmes d'apprentissage automatique. Les auteurs utilisent un ensemble de données de sites Web d'hameçonnage et de sites Web légitimes pour évaluer les performances de leur méthode sur différents algorithmes de classification. Les résultats montrent qu'en termes de détection des sites d'hameçonnage, Ils ont obtenu une précision de 98,4% avec le classificateur de régression logistique.

Jain et Gupta [16] ont proposé à nouveau 12 propriétés caractérisant les pages HTML. Six fonctions sont de leur proposition, les six autres étaient proposées par Whittaker et al. [28]. Voici le schéma de leur approche:

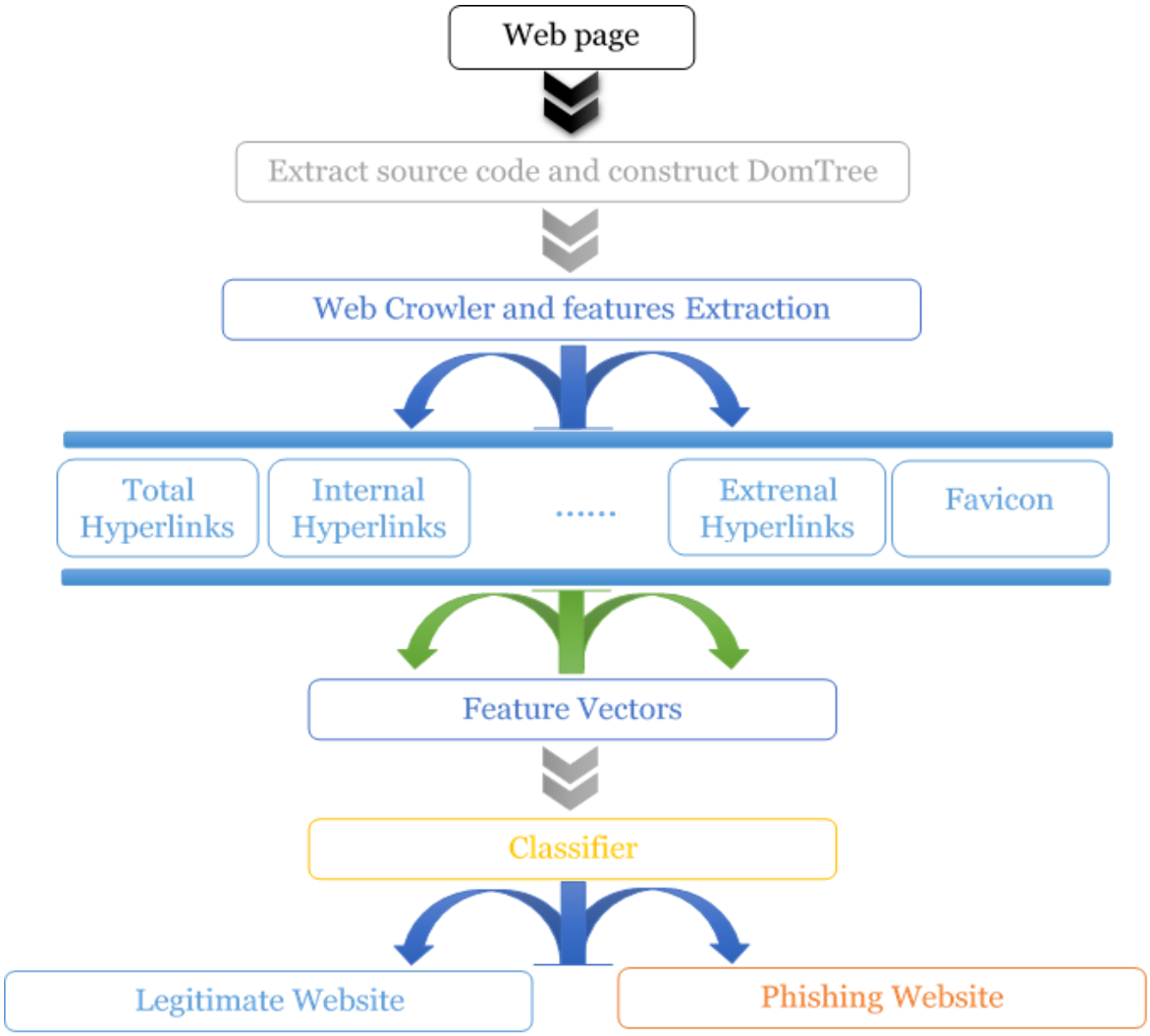


Figure 3.1. Processus de l'approche Jain et Gupta [16]

Les caractéristiques proposées par [16] identifient la relation entre le contenu de la page Web et l'URL de la page Web. Leurs caractéristiques sont basées sur des hyperliens de la page Web. Un site Web peut être transformé en une arborescence Document Object Model (DOM), utilisé pour extraire les caractéristiques des hyperliens, comme illustré à la Figure suivante :

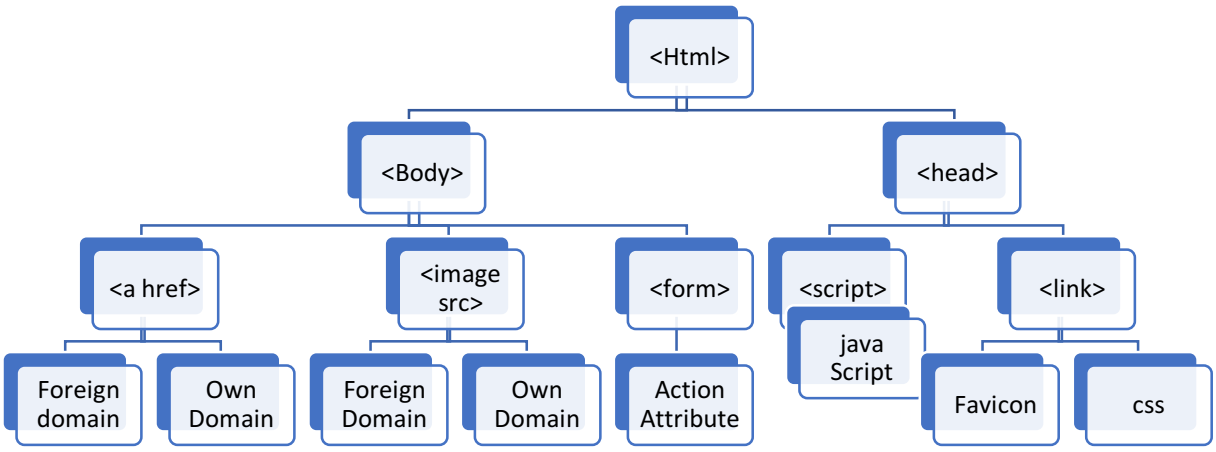


Figure 3.2. Arborescence Document Object Model (DOM) [32]

Shirazi [18] a observé deux préoccupations concernant les approches d'apprentissage automatique existantes. L'étude s'est concentrée sur les caractéristiques dérivées de l'utilisation du nom de domaine dans l'hameçonnage et les sites Web légitimes et a rapporté une précision de 97 à 98% sur les ensembles de données choisis. Pour prouver les performances de l'ensemble du modèle, ils l'ont évalué avec des échantillons des pages d'hameçonnage provenant d'une source complètement différente et ont obtenu un taux de détection de 99%.

Medvet et al. [13] ont utilisés une nouvelle technique introduite pour comparer visuellement des pages d'hameçonnage suspectes avec des pages légitimes. Leur objectif était de déterminer si ces deux pages sont étranges. Ils ont identifié et examiné les fonctions des trois pages, qui ont joué un rôle clé dans la transformation des pages contrefaites en des pages légitimes. Ces caractéristiques incluent le texte et son style, les images intégrées dans la page et l'apparence visuelle globale de la page présentée par le navigateur. Afin de tester la faisabilité de leur méthode, ils ont utilisé un ensemble de données contenant 41 pages d'hameçonnage réelles et les cibles légales correspondantes pour l'évaluation expérimentale. Leurs résultats expérimentaux sont satisfaisants en termes de faux positifs et de faux négatifs.

3.3. Les approches hybrides

Dans J. Patel [22], différentes techniques de détection d'hameçonnage sont discutées, et une technique basée sur des règles heuristiques est implémentée pour détecter les URL d'hameçonnage et obtenir différentes fonctions à partir d'une URL donnée. Le groupe de caractéristiques comprend des caractéristiques liées à la barre d'adresse, des caractéristiques HTML-JavaScript et caractéristiques de domaine. Ils ont implémenté différentes règles heuristiques et ont pris des décisions basées sur les résultats des règles heuristiques. De plus, différents poids sont attribués à chaque règle heuristique pour détecter correctement les URL.

L'une des méthodes heuristiques les plus connues est l'outil d'analyse de réseau et anti-hameçonnage Carnegie Mellon (CANTINA) proposé par Han et Zhang [10]. CANTINA est basé sur le contenu et est utilisé pour détecter les sites Web d'hameçonnage basés sur l'algorithme de récupération d'informations TF-IDF et un algorithme de lien hypertexte robuste. En utilisant une somme pondérée de 8 caractéristiques (4 liées au contenu, 3 lexicales et 1 basé sur un service tiers Whois), ils ont montré que CANTINA peut détecter correctement environ 95% des sites d'hameçonnage. Le but de cette méthode est d'éviter de télécharger la page Web réelle, réduisant ainsi le risque potentiel de rechercher un contenu malveillant sur le système de l'utilisateur.

Rao et al. [15] ont proposé une méthode de classification qui utilise une méthode d'extraction de caractéristiques basée sur l'heuristique. Ils ont divisé les caractéristiques extraites en trois catégories, telles que les caractéristiques d'URL, les caractéristiques tierces et les caractéristiques basées sur les hyperliens. La technique proposée donne une précision de 99,55%. L'inconvénient est que, comme ce modèle utilise des fonctions tierces, la classification du site Web dépend de la vitesse des services fournis par le tiers.

Rao et Ali [14] ont implémenté une application de bureau appelée *PhishShield*, qui se concentre sur l'URL de la page d'hameçonnage et le contenu du site Web. *PhishShield*, prend l'URL comme entrée et affiche l'état de l'URL en tant que site Web d'hameçonnage ou légitime. Les heuristiques utilisées pour détecter l'hameçonnage

incluent les liens vides dans le corps du code HTML, le contenu protégé par des droits d'auteur, la protection du titre et l'identification du site Internet. Il peut détecter les attaques d'hameçonnage qui ne peuvent pas être détectées par la liste noire et est plus rapide que les techniques d'évaluation visuelles utilisées pour détecter l'hameçonnage. Le taux de précision de PhishShield est de 96,57%, couvrant divers sites Web d'hameçonnage, ce qui contribue à réduire le taux de faux positifs.

Bhattacharya [12] fournit une application de bureau appelée *PhishSaver*, qui se concentre sur l'URL de la page Web d'hameçonnage et le contenu du site Web. L'objectif est de détecter les sites d'hameçonnage à l'aide d'une application de bureau appelée PhishSaver. PhishSaver utilise une combinaison de listes noires et de nombreuses heuristiques pour détecter plusieurs attaques d'hameçonnage. Pour la liste noire, il a utilisé le service Google API (il s'agit de la liste noire de navigation sécurisée de Google), qui est constamment mis à jour et maintenu par Google. Il est également possible d'exécuter PhishSaver en tant que démon, ce qui signifie qu'il peut détecter les attaques d'hameçonnage en temps réel pendant que les utilisateurs naviguent sur Internet. PhishSaver prend l'URL comme entrée et affiche l'état de l'URL en tant que site Web d'hameçonnage ou légitime. Les heuristiques utilisées pour détecter l'hameçonnage sont des liens de pied de page de valeur nulle, des liens vides dans le corps du code HTML, du contenu protégé par le droit d'auteur, du contenu de l'en-tête et du réseau d'identification du site. PhishSaver peut détecter les attaques d'hameçonnage qui peuvent ne pas être mises sur liste noire et est plus rapide que les techniques d'évaluation visuelle utilisées pour détecter l'hameçonnage. Il est à observer que, par rapport à l'API de navigation sécurisée fournie par Google pour effectuer des recherches en ligne, PhishSaver a un taux de précision plus élevé. Il s'agit de la principale méthode de détection utilisant des listes noires. PhishSaver couvre plus d'attaques, d'hameçonnage et entraîne moins d'erreurs.

Belhait [3] a proposé une approche similaire à la nôtre. L'auteur a examiné la combinaison OR des deux modèles entrainés chacun sur des caractéristiques des classes différentes (URL et HTML). Le modèle adopté été basé sur le classificateur de forêt d'arbres décisionnels qui fournis une fiabilité d'environ 90%. Notre travail peut être considéré comme complémentaire au travail présenté dans [3]. Nous

expérimentons plus des combinaisons des modèles et plus des caractéristiques des deux catégories URL et HTML.

3.4. Conclusion

Nous avons présenté dans ce chapitre, un aperçu sur les travaux qui ont été faits pour classifier les pages web au moyen des machines d'apprentissage et des autres approches comme l'utilisation des listes noires, des heuristiques et des similarités visuelles. Notre état de l'art a été établi selon un point de vue, celui de l'extraction des caractéristiques les plus efficaces pour notre travail. Nous avons choisi celle de l'apprentissage automatique basée sur les caractéristiques combinées de l'URL, et HTML ce qui fera l'objet de notre étude dans le prochain chapitre.

CHAPITRE IV

METHODOLOGIE D'ANALYSE ET APPROCHE PROPOSEE

CHAPITRE IV.

METHODOLOGIE D'ANALYSE ET APPROCHE PROPOSEE

Les sites Web d'hameçonnage restent une menace de sécurité persistante. Jusqu'à présent, les approches d'apprentissage automatique semblent avoir le meilleur potentiel de défense. Cependant, les approches d'apprentissage automatique existantes pour la détection de l'hameçonnage soulèvent deux préoccupations principales. Le premier est le grand nombre de caractéristiques d'entraînement utilisées et le manque d'arguments de validation pour ces choix de caractéristiques. La deuxième préoccupation est le type d'ensembles de données utilisés dans la littérature qui sont biaisés par inadvertance en ce qui concerne les caractéristiques basées sur l'URL ou le contenu du site Web ou aussi par les parties tierces.

Dans ce mémoire, nous avons établi un modèle hybride qui utilise des caractéristiques des types différentes et qui peut être utilisé pour ajouter ou supprimer des entités a l'ensemble de données. De plus, l'utilisateur peut refaire l'étape d'extraction pour obtenir la valeur mise à jour de la caractéristique actuellement définie. Dans ce travail, nous avons utilisé les caractéristiques définis par Jain et Gupta [16] et Shirazi [33] pour le contenu des pages et les caractéristiques définis par Verma et Das [25], et Sahingoz et al. [26], Suman et al. [12], Srinivasa et Pai [14], et Kulkarni et Brown [7] pour la structure des URLs.

Ce chapitre présente les étapes qui nous ont permis l'adoption de notre approche. Nous commençons dans la première partie par une description de la méthodologie d'analyse qui nous a amenée à sélectionner le modèle, la liste des caractéristiques, ainsi que la construction de l'ensemble de données.

4.1. Méthodologie d'analyse

Dans un premier temps, une analyse bibliographique sur les caractéristiques utilisées pour identifier les techniques d'hameçonnage et de machines learning nous permet d'examiner les différents types de méthodes d'attaque implémentées pour la détection.

Dans notre recherche, nous avons remarqué que la méthode la plus couramment utilisée est l'apprentissage automatique et que l'identification de l'hameçonnage repose sur l'hameçonnage de la structure des URL. Cette dernière s'est affaiblie avec le développement des cybercriminels.

D'autres approches étaient créées en se basant sur le contenu de la page en plus de quelques caractéristiques de l'URL. De là vient la recherche de Jain et Gupta [16], mais celle-ci a un inconvénient : si les pages sont infectées seulement dans leurs URLs, elles ne seront pas détectées car le travail de Jain et Gupta [16], n'a pas pris en considération les entités qui servent à détecter l'hameçonnage dans la partie URL. De là vient notre idée de combiner l'approche de Jain et Gupta [16] qui est composée de 12 entités HTML et la compléter en lui rajoutant 8 entités HTML et 41 entités URL. Pour faciliter l'utilisation de notre approche en temps réel, nous avons créé une extension qui sera comme un intergiciel à l'utilisateur. Le schéma de la figure 4.1 détaille les étapes de notre approche proposée.

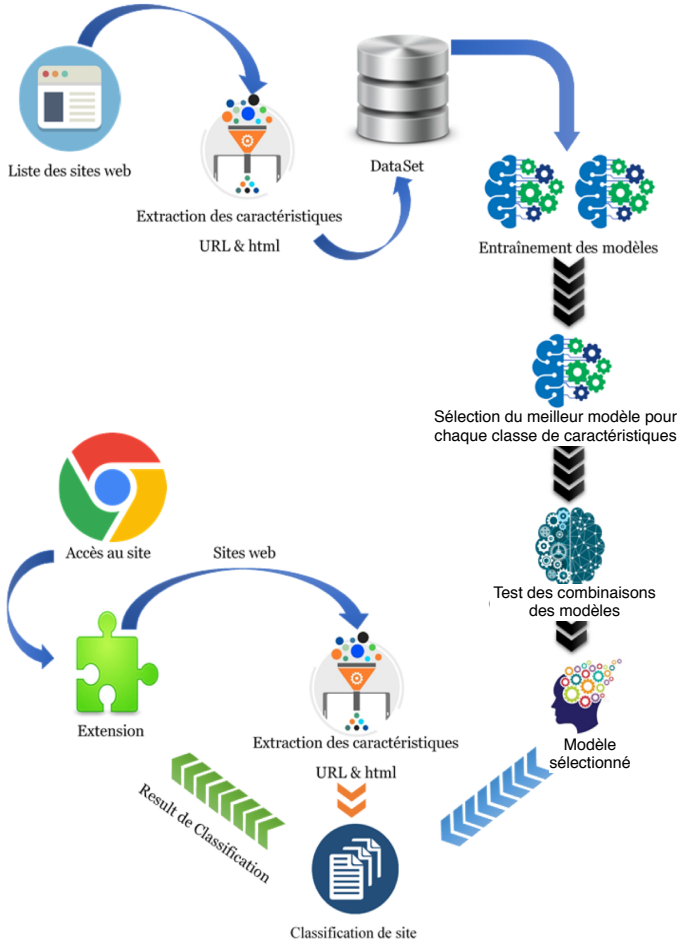


Figure 4.1. Le processus de démarche.

4.2. Construction du dataset

Dans cette section, nous décrivons le processus adopté pour la construction de l'ensemble de données (dataset) utilisé dans notre modèle.

4.2.1. Collection des URLs

Tout d'abord, il y a eu besoin de collecter une liste des URLs. Pour construire une liste acceptable et équilibrée, deux classes d'URL sont nécessaires : légitime et hameçonnage. Les URLs d'hameçonnage ont principalement été fournies par PhishTank[1] on a aussi utilisé le site Openphish[2]. Depuis ces deux sources on a été capable de collecter les adresses de nombreux sites Web malveillants. En même temps, il était nécessaire de collecter des sites Web légitimes. Pour collecter ces pages, nous avons assemblé une liste depuis Alexa[3]. En plus, nous avons filtré la liste créée par Sahingoz et al. [26] et qui a été originalement obtenu à l'aide de l'API de recherche Yandex[4]. Le filtrage consiste à supprimer les pages qui sont devenues inaccessibles en raison de la courte durée de vie des sites d'hameçonnage. Finalement, la liste a été examinée en supprimant les URLs répétés.

Nous avons donc obtenu une liste qui comporte 11558 URL. La liste contient au total 5779 URL légitimes et 5779 URL d'hameçonnage. Le tableau 4.1 affiche le nombre d'instances et sources des URLs d'hameçonnage et légitimes.

Source	Nombres des sites	Type des sites
Yandex	1518	Légitime
Alexa	4261	Légitime
Openphish	1717	Hameçonnage
Phishtank	4062	Hameçonnage

Tableau 4.1. Les statistiques de dataset.

[1] Phishtank site web : https://www.phishtank.com/
[2] Openphish site web : https://openphish.com/
[3] Alexa site web : https://www.alexa.com/
[4] Moteur de recherche Yandex : https://yandex.com/

4.2.2. Extraction des caractéristiques

La liste des URLs collectés a été utilisée pour l'extraction des différentes caractéristiques pour les 11 558 sites Web légitimes et d'hameçonnage. L'étape d'extraction a pris environ 6 jours pour extraire et créer les caractéristiques HTML, notamment avec les sites légitimes. Comme mentionné, ces sites ont un taux de liens plus élevé que ceux de l'hameçonnage, qui peut atteindre 1000 liens sur un seul site. Pour cela certaines caractéristiques prennent beaucoup de temps lors de leurs extractions.

Nous avons développé des scripts Python pour l'extraction des caractéristiques URL et HTML. Le choix de Python a été adopté à cause de ses bibliothèques riches qui nous ont servis à l'extraction et la manipulation des données collectées. L'ensemble de caractéristiques adopté dans notre approche peut être classé en deux catégories: caractéristiques basées sur HTML et caractéristiques basées sur l'URL.

4.2.2.1. Extraction des caractéristiques basées sur l'HTML

Dans notre approche, on a utilisé 20 caractéristiques, 12 (caractéristiques F1-F12) ont été proposées par Jain et Gupta [16] caractérisant la page HTML, les autres étaient proposées par Shirazi [33]. Pour l'extraction, nous avons construit le vecteur de caractéristiques suivant :

$$F = \{F1, F2, \ldots, F20\}$$

Certaines fonctions produiront des valeurs de 1 et 0, où 1 signifie hameçonnage et 0 signifie légitime. Nous discuterons de toutes ces fonctionnalités dans les sous-sections suivantes.

1) F1 et F2 : Total et pas de liens hypertexte [16]

Par rapport aux sites Web légitimes, les sites d'hameçonnage ont des pages très limitées, parfois seulement une ou deux. De plus, les sites d'hameçonnage ne fournissent parfois aucun lien hypertexte car l'attaquant utilise la technologie des liens

hypertextes cachés. Jain et Gupta [16] ont analysé le fait que si un site Web est original, ils peuvent extraire au moins un lien hypertexte du code source. Lors du calcul du total, ils n'ont pris en compte que les balises *href*, *Link* et *src*, mais dans notre proposition, nous avons considéré tous les hyperliens dans toutes les balises HTML.

$$F_1 = \sum de\ tous\ les\ liens\ dans\ la\ page$$

$$F2 = \begin{cases} 0 & si\ F1 > 0 \\ 1 & sinon \end{cases}$$

2) F3 et F4 : Liens hypertextes internes et externes [16]

Les hyperliens internes et externes signifient que les hyperliens contiennent respectivement des domaines de base identiques et différents. La plupart des sites d'hameçonnage copient et collent le code HTML du site Web officiel cible, ils peuvent donc contenir de nombreux hyperliens vers le site Web cible. Sur les sites Web légitimes, la plupart des hyperliens contiennent le même domaine de base, tandis que dans les sites Web de phishing, de nombreux hyperliens peuvent contenir le domaine du site Web légitime correspondant. Ici, nous considérons le rapport entre le nombre des liens internes (F3) et externes (F4) et le nombre total d'hyperliens.

$$F3 = \begin{cases} \frac{H_{internal}}{F_1} & si\ \ F_1 > 0 \\ 0 & sinon \end{cases}$$

$$F4 = \begin{cases} \frac{H_{External}}{F_1} & si\ \ F_1 > 0 \\ 0 & sinon \end{cases}$$

Où $H_{Internal}$, $H_{External}$ sont respectivement le nombre de hyperliens internes et externes contenu dans une page Web.

3) F5 : Hyperlien nulle [16]

Dans un lien hypertexte vide, l'attribut *href* de la balise d'ancrage ne contient pas l'URL. Lorsque les utilisateurs cliquent sur un lien vide, ils reviennent à la même page. Un site Web légitime contient de nombreuses pages Web. Par conséquent, afin de se

comporter comme un site Web légitime, l'attaquant n'accorde aucune valeur au lien hypertexte et le lien apparaît comme actif sur le site Web. Ici, nous comptons le rapport entre les hyperliens nuls et le total des hyperliens présents sur le site Web.

$$F5 = \begin{cases} \frac{H_{Null}}{F_1} & si \quad H_1 > 0 \\ 0 & sinon \end{cases}$$

Où H_{Null} est le nombre des hyperliens nuls contenu dans une page Web.

4) F6 : CSS externe [16]

Les feuilles de style en cascade (CSS) sont un langage utilisé pour décrire le format des documents et définir l'apparence des sites Web écrits en HTML, XHTML et XML. Les attaquants essaient toujours d'imiter des sites Web légitimes et de conserver la même conception de site Web d'hameçonnage que le site Web cible pour attirer des victimes potentielles. Formellement, CSS contient une liste de règles qui peuvent associer un ensemble de sélecteurs d'attributs et de valeurs à un ensemble de déclarations. Le CSS de tout site Web est contenu dans un fichier CSS externe ou dans le code HTML lui-même. Les fichiers CSS externes utilisent la balise <*Link*> pour s'associer à certains sites Web HTML. Pour cela, nous considérons le nombre des fichiers CSS externes utilisé dans la page.

5) F7 et F8 : Redirection interne et externe [16]

La redirection indique si la page Web est redirigée vers un autre emplacement. Lorsque le navigateur tente d'ouvrir l'URL redirigée, il ouvre une page Web avec une autre URL. Parfois, la redirection d'URL peut perturber les utilisateurs sur le site Web dans lequel ils naviguent. De plus, les redirections peuvent amener les utilisateurs à visiter de faux sites Web. Sur les sites d'hameçonnage, certains liens peuvent être redirigés vers les domaines légaux correspondants. Parfois, les faux sites Web peuvent également être redirigés vers des domaines légaux après avoir rempli le formulaire de connexion. Pour cela nous considérons le rapport entre les liens avec des redirections internes (F7) et externes (F8) vis-à-vis le nombre total des liens dans la page.

$$F7 = \begin{cases} \frac{H_{i-redirect}}{F_1} & si \quad F_1 > 0 \\ 0 & sinon \end{cases}$$

$$F8 = \begin{cases} \frac{H_{e-redirect}}{F_1} & si \quad F_1 > 0 \\ 0 & sinon \end{cases}$$

Où $H_{i\text{-}redirect}$ $H_{e\text{-}redirect}$ sont les nombres d'hyperliens présents sur la page avec redirection interne et externe, respectivement.

6) F9 et F10 : Erreur interne et externe [16]

Grâce à cette heuristique, les erreurs dans les hyperliens du site peuvent être vérifiés. Lorsqu'un utilisateur demande une URL et que le serveur ne peut pas déterminer l'URL demandée, une erreur «*404 introuvable* » se produit. L'attaquant peut également ajouter un lien inexistant. Lorsqu'un utilisateur tente d'accéder à un lien rompu ou brisé, une erreur «*404 introuvable* » est générée. Contrairement à Jain et Gupta [16], nous considérons tous les codes d'erreurs supérieurs ou égaux à 400.

$$F9 = \begin{cases} \frac{H_{i-error}}{H_{Internal}} & si \quad H_{internal} > 0 \\ 0 & sinon \end{cases}$$

$$F10 = \begin{cases} \frac{H_{e-error}}{H_{external}} & si \quad H_{external} > 0 \\ 0 & sinon \end{cases}$$

Ou $H_{i\text{-}error}$, H_{e_error}, $H_{Internal}$ et $H_{External}$ Sont les nombres d'erreurs internes, d'erreurs externes, le nombre total de liens hypertextes internes et de liens externes dans une page Web.

7) F11 : Lien vers le formulaire de connexion [16]

Les sites d'hameçonnage contiennent généralement un formulaire de connexion qui peut être utilisé pour voler les informations d'identification des internautes. Après avoir rempli le formulaire sur le faux site Web, les informations personnelles de l'utilisateur sont transférées à l'attaquant. Le formulaire de connexion d'un site d'hameçonnage s'affiche de la même manière que sur un site Web légitime.

Dans cette fonction, nous vérifions l'authenticité du formulaire de connexion. Sur les sites Web légitimes, la portée comprend généralement l'URL du site Web actuel. Cependant, l'attaquant peut utiliser d'autres domaines au lieu du domaine visité, des valeurs vides (hyperliens dans la section pied de page) ou des fichiers PHP [48] dans le champ "*form action*" du site d'hameçonnage. Ce fichier PHP contient des scripts pour enregistrer les données d'entrée (par exemple, ID utilisateur ou mot de passe) dans un fichier texte enregistré sur l'ordinateur de l'attaquant. Le fichier PHP est généralement nommé index.php, login.php, etc.

8) F12 : favicon externe [16]

Favicon est une icône d'image liée à un site Web spécifique. L'attaquant peut copier les favicons du site Web cible. Favicon est un fichier *.ico* lié à une URL, situé dans la balise de lien de l'arborescence DOM. Si l'icône de favicon affichée dans la barre d'adresse n'est pas du site Web actuel, cela est considéré comme une tentative d'hameçonnage. Cette fonction contient deux valeurs, 0 (légitime) et 1 (hameçonnage). Si l'icône de favicon appartient au même domaine, cette caractéristique aura 0, sinon elle vaut 1.

9) F13 : Envoi d'informations par courrier électronique [33]

La présence de l'action "*mailto :*" dans un formulaire d'envoi est considéré comme un signe d'hameçonnage.

10) F14 : Médias internes [33]

Le rapport des fichiers multimédias (sons, vidéos, images) internes et le nombre total des fichiers médias est considéré comme une propriétés des sites légitimes et d'hameçonnage.

11) F15 : URL de l'ancre [33]

Pour cette caractéristique, nous étudions si la balise <*a*> et la page Web ont des noms de domaine différents. La page est donc considérée comme hameçonnage si l'ancre n'est pas liée à la page Web, par exemple :

1. <a href = "#">
2. <a href="#Content">
3. <a href="skip">
4. <a href="JavaScript : : void(0)">
5. <a href="lien interne">

12) F16 : Liens dans les balises <Meta>, <Script> et <Link> [33]

Cette caractéristique observe le domaine dans les balises de l'en-tête telles que <SCRIPT>, Balises <META> et <LINK>. Le rapport entre les liens de ces balises qui pointent vers un domaine, différent de celui du site, et le nombre total de liens dans ces balises est considéré pour distinguer les pages légitimes des pages d'hameçonnage.

13) F17 : Gestionnaire de formulaire serveur (SFH) [33]

Cette fonctionnalité examine l'action du formulaire d'envoi sur la page. Si l'action est « *Aucune* », « *vide* » ou « *à propos de : vide* », nous marquons le site comme hameçonnage. Les sites légitimes pointeront vers une URL.

14) F18 : i_frame [33]

HTML utilise la balise <*Iframe*> pour afficher une autre page à l'intérieur de la page actuelle. Cette caractéristique vérifie s'il y a une balise <*Iframe*> dans la page et sa bordure est définie sur transparent. Si ces deux éléments sont présents, nous marquons la page web comme hameçonnage.

15) F19 : Correspondance du titre de la page et du nom de domaine [33]

De nombreux sites Web légitimes réutilisent les noms de domaine dans les titres de page. Nous avons constaté que de nombreux sites Web d'hameçonnage utilise cette caractéristique pour faire croire aux utilisateurs qu'ils visitent des sites web légitimes. Cependant, il est évident que les sites Web d'hameçonnage n'utiliseront pas le nom de domaine d'hameçonnage dans la page de titre car il sera clairement visible pour les utilisateurs.

16) F20 : Nom de domaine avec logo copyright [33]

De nombreux sites web légitimes utilisent des logos protégés par le droit d'auteur pour indiquer la propriété de la marque et le nom de leur organisation. Habituellement, le nom de domaine est situé avant ou après le logo de copyright de ces sites. Afin de générer cette fonction, nous avons considéré jusqu'à 50 caractères avant et après le logo du copyright, supprimé les espaces et vérifié si le nom de domaine existe dans la chaîne résultante.

4.2.2.2. Extraction des caractéristiques Basé sur l'URL

Les caractéristiques à base d'URL sont basées sur certains aspects de l'URL du site Web. L'attaquant tente a caché l'URL actuel de l'utilisateur d'une manière ou d'une autre. Par exemple, les URLs avec des adresses IP, des signes «at » (@), des doubles barres obliques et des préfixes ou suffixes sont toutes des méthodes permettant de masquer les URL. D'autres méthodes notables sont la longueur de l'URL, c'est-à-dire si le site Web a des sous-domaines, utilise des services raccourcis ou utilise des ports non standard. Pour cette raison nous avons utilisé des caractéristiques proposées par Verma et Das [25], Suman et al. [12], Srinivasa et Pai [14], Sahingoz et al. [26] et Kulkarni et Brown [7]. Nous discuterons de toutes les caractéristiques de l'URL dans les sous-sections suivantes.

1) F21 : Utilisation de l'adresse IP [14, 25]

Les adresses IP sont utilisées dans le nom d'hôte (*hostname)* d'URL pour masquer l'identité des sites. Par conséquent, la présence des adresses IP dans le nom d'hôte est considérée comme un indicateur d'hameçonnage.

2) F22-23 : longueur d'URL [7, 25]

Les attaquants utilisent des URLs long pour cacher le vrai domaine et sous-domaines de l'URL. Pour cela, la longueur d'URL (F22) de nom d'hôte (F23) est considéré pour distinguer le type d'URL.

3) F24 : Utilisation des services de réduction d'URL [7, 25]

Les services de réduction d'URL sont utilisés pour indiquer des URLs courtes qui servent à la redirection vers d'autres URLs longues et complexes. Ce service peut aussi être utilisé par les attaqueurs pour masquer le nom des vrais hôtes. Donc, l'utilisation des services de réduction d'URL est considérée comme une tentative d'hameçonnage.

4) F25-F40 : Nombre des caractères spéciaux (collectés de [7, 12, 25, 14])

Les caractères spéciaux sont utilisés pour tromper les utilisateurs novices. Ici, nous considérons le nombre des caractères spéciaux utilisés dans l'URL, nous considérons notamment : {'.', '-', '@', ' ?', ' &', '|', '=', '_', '%', '/', '* ', ' :', ',', ' ;', '$','%20'}

5) F41 : Extension de chemin [7]

Certaines extensions sont utilisées dans le chemin de l'URL pour déclencher des scripts d'hameçonnage. Ici, nous considérons les extensions : *.exe, .txt* et *.js*.

6) F42 : Utilisation du protocole « HTTPS » dans l'URL [7, 17]

La plupart des sites d'hameçonnage ne fournissent aucune fonctionnalité de sécurité contrairement aux sites légitimes. Pour cela les URLs utilisant le protocole HTTPS sont considérés légitimes.

7) F43-F45 : Pourcentage des chiffres dans l'URL [12, 26]

Les sites d'hameçonnage utilisent beaucoup de chiffres dans leurs URLs pour cacher les vrais domaines des pages. Nous considérons alors le pourcentage des chiffres dans le domaine d'URL (F43), sous-domaine (F44), et chemin d'URL (F45).

8) F46 : Mots clés d'hameçonnage [7]

Les attaquants utilisent des mots sensibles pour gagner la confiance des utilisateurs dans les pages Web visitées. La présence fréquentes de ces mots est considérée comme un indice d'hameçonnage. Ici nous considérons la liste des mots suivants : {*'wp', 'login', 'includes', 'admin', 'content', 'site', 'images', 'js', 'alibaba', 'css', 'myaccount', 'dropbox', 'themes', 'plugins', 'signin', 'view'*}

9) F47 : URL anormal [Observé]

Les sites d'hameçonnage utilisent le modèle suivant "*w [w]? [0-9] **" au lieu de "*www*" pour tromper les utilisateurs. Ainsi, les URL avec des sous-domaines correspondant à ce modèle sont considérés comme hameçonnage.

10) F48-49 : Nombre de redirections [7, 12]

La redirection d'URL est une technique utilisée pour ouvrir des pages avec des URLs différentes de celles initialement sélectionnées par les utilisateurs. Ceci est utile pour empêcher l'accès aux liens rompus lorsque des pages Web sont déplacées. Les URLs peuvent être redirigées vers des pages avec le même domaine (c'est-à-dire une redirection interne) ou vers des pages de différents domaines (c'est-à-dire des redirections externes). Cependant, la redirection peut également être utilisée à des fins hostiles. Le nombre de redirections internes (F48) et externes (F49) est considéré pour distinguer les sites d'hameçonnage.

11) F50-F54 : Caractéristiques des mots [26]

Les techniques de traitement du langage naturel sont également utilisées dans la détection des sites d'hameçonnage. Nous considérons le nombre de mots (F50), la répétition de caractères (F51), la longueur du mot le plus court (F52), la longueur du mot le plus long (F53) et la longueur moyenne des mots dans l'URL (F54).

12) F55 : Punycode [12]

Le punycode est une syntaxe de codage utilisé dans les noms de domaines pour remplacer certains caractères ASCII par des caractères Unicode. Les URLs auront alors l'air d'être légitimes là où elles se réfèrent à différents sites Web. Les URLs avec punycode sont considérées comme hameçonnage.

$$\text{punny}_{\text{code}} = \begin{Bmatrix} si\ l'URL\ commence\ par\ (\text{"}http{:}//xn--\text{"}) \Longrightarrow hameçonnage \\ sinon \Longrightarrow légitime \end{Bmatrix}$$

13) F56 : Domaines de marques [7]

Les URLs d'hameçonnage utilisent des noms de domaine de marque dans différentes parties d'URL. La présence de noms de marque dans la partie domaine est considérée comme un indicateur de légitimité alors que leur présence dans des sous-domaines ou des chemins est considérée comme un indicateur d'hameçonnage.

14) F57-F58 : Nombre des termes courants [7]

Les termes courants dans les URLs tels que '*www*', '*.com*' ne sont utilisés qu'une seule fois dans les URLs légitimes alors qu'ils sont utilisés plusieurs fois dans des URLs d'hameçonnage. Nous considérons le nombre d'occurrence de '*www*' (F57) et de '*.com*' (F58) dans les URLs comme des indices d'hameçonnage.

15) F59 : Longueur d'enregistrement de domaine [12]

La période d'enregistrement du nom de domaine peut remonter à plusieurs années. Le montant du renouvellement du nom de domaine est payé d'avance au registraire. Un propriétaire de domaine peut payer au moins pour 1 an et jusqu'à 10 ans. Pour chaque domaine acheté, il doit payer le montant du renouvellement annuel. Les webmasters affirment que les moteurs de recherche privilégient davantage les noms de domaine enregistrés depuis longtemps. Les domaines achetés pour le spam Web ne sont généralement pas enregistrés pendant plus d'un an.

16) F60 : Age du domaine [12]

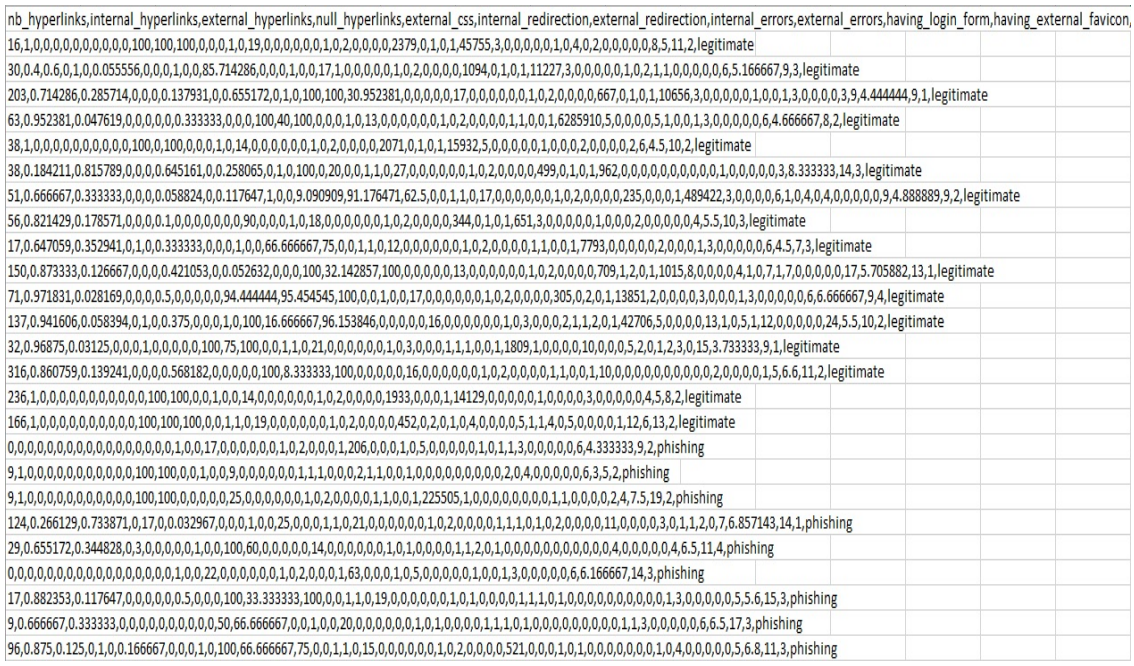
Généralement, en raison des plaintes des utilisateurs et des autorités, la durée de vie d'un site d'hameçonnage est très courte avant qu'il soit fermé par son hébergeur, on peut donc passer un minimum de temps à dire qu'un site est sûr ou pas.

17) F61 : Trafic du site [12]

Contrairement aux sites légitimes, les sites d'hameçonnage sont moins fréquentés. Nous considérons donc le nombres des visiteurs des sites comme un indice d'hameçonnage.

4.2.3. Les vecteurs de caractéristiques

Après l'extraction des caractéristiques, des vecteurs des caractéristiques ont été générés pour chaque URL. L'ensemble de ces vecteurs frome notre dataset qui a été enregistré dans un fichier .CSV comme le montre la Figure 4.2.



```
nb_hyperlinks,internal_hyperlinks,external_hyperlinks,null_hyperlinks,external_css,internal_redirection,external_redirection,internal_errors,external_errors,having_login_form,having_external_favicon,
16,1,0,0,0,0,0,0,0,0,100,100,100,0,0,0,1,0,19,0,0,0,0,0,0,1,0,2,0,0,0,0,2379,0,1,0,1,45755,3,0,0,0,0,0,1,0,4,0,2,0,0,0,0,0,8,5,11,2,legitimate
30,0.4,0.6,0,1,0,0.055556,0,0,0,1,0,0,85.714286,0,0,0,1,0,0,17,1,0,0,0,0,0,1,0,2,0,0,0,0,1094,0,1,0,1,11227,3,0,0,0,0,0,1,0,2,1,1,0,0,0,0,0,6,5.166667,9,3,legitimate
203,0.714286,0.285714,0,0,0,0.137931,0,0.655172,0,1,0,100,100,30.952381,0,0,0,0,0,17,0,0,0,0,0,0,1,0,2,0,0,0,0,667,0,1,0,1,10656,3,0,0,0,0,0,1,0,0,1,3,0,0,0,0,3,9,4.444444,9,1,legitimate
63,0.952381,0.047619,0,0,0,0,0,0.333333,0,0,0,100,40,100,0,0,0,1,0,13,0,0,0,0,0,0,1,0,2,0,0,0,0,1,1,0,0,1,6285910,5,0,0,0,0,5,1,0,0,1,3,0,0,0,0,0,6,4.666667,8,2,legitimate
38,1,0,0,0,0,0,0,0,0,0,100,0,100,0,0,0,1,0,14,0,0,0,0,0,0,1,0,2,0,0,0,0,2071,0,1,0,1,15932,5,0,0,0,0,0,1,0,0,0,2,0,0,0,0,0,2,6,4.5,10,2,legitimate
38,0,0.184211,0.815789,0,0,0,0.645161,0,0.258065,0,1,0,100,0,20,0,0,1,1,0,27,0,0,0,0,0,0,1,0,2,0,0,0,0,499,0,1,0,1,962,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,3,8.333333,14,3,legitimate
51,0.666667,0.333333,0,0,0,0.058824,0,0.117647,1,0,0,9.090909,91.176471,62.5,0,0,1,1,0,17,0,0,0,0,0,0,1,0,2,0,0,0,0,235,0,0,0,1,489422,3,0,0,0,0,6,1,0,4,0,4,0,0,0,0,0,9,4.888889,9,2,legitimate
56,0,0.821429,0.178571,0,0,0,0.1,0,0,0,0,0,0,0,0,90,0,0,0,0,1,0,18,0,0,0,0,0,0,1,0,2,0,0,0,0,344,0,1,0,1,651,3,0,0,0,0,0,1,0,0,0,2,0,0,0,0,0,4,5.5,10,3,legitimate
17,0,0.647059,0.352941,0,1,0,0.333333,0,0,0,1,0,0,66.666667,75,0,0,1,1,0,12,0,0,0,0,0,0,1,0,2,0,0,0,0,1,1,0,0,1,7793,0,0,0,0,0,2,0,0,0,1,3,0,0,0,0,0,6,4.5,7,3,legitimate
150,0,0.873333,0.126667,0,0,0,0.421053,0,0.052632,0,0,0,100,32.142857,100,0,0,0,0,0,13,0,0,0,0,0,0,1,0,2,0,0,0,0,709,1,2,0,1,1015,8,0,0,0,0,4,1,0,7,1,7,0,0,0,0,0,17,5.705882,13,1,legitimate
71,0,0.971831,0.028169,0,0,0,0.5,0,0,0,0,0,0,94.444444,95.454545,100,0,0,1,0,0,17,0,0,0,0,0,0,1,0,2,0,0,0,0,305,0,2,0,1,13851,2,0,0,0,0,3,0,0,0,1,3,0,0,0,0,0,6,6.666667,9,4,legitimate
137,0,0.941606,0.058394,0,1,0,0.375,0,0,0,1,0,0,100,16.666667,96.153846,0,0,0,0,0,16,0,0,0,0,0,0,1,0,3,0,0,0,2,1,1,2,0,1,42706,5,0,0,0,0,13,1,0,5,1,12,0,0,0,0,0,24,5.5,10,2,legitimate
32,0,0.96875,0,0.03125,0,0,0,1,0,0,0,0,0,0,100,75,100,0,0,1,1,0,21,0,0,0,0,0,0,0,1,0,3,0,0,0,1,1,1,0,0,1,1809,1,0,0,0,0,10,0,0,0,5,2,0,1,2,3,0,15,3.733333,9,1,legitimate
316,0,0.860759,0.139241,0,0,0,0.568182,0,0,0,0,0,0,100,8.333333,100,0,0,0,0,0,16,0,0,0,0,0,0,1,0,2,0,0,0,0,1,1,0,0,1,10,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,1,5,6.6,11,2,legitimate
236,1,0,0,0,0,0,0,0,0,0,0,0,0,100,100,0,0,1,0,0,14,0,0,0,0,0,0,1,0,2,0,0,0,0,1933,0,0,0,1,14129,0,0,0,0,0,1,0,0,0,0,3,0,0,0,0,0,4,5,8,2,legitimate
166,1,0,0,0,0,0,0,0,0,0,0,0,0,100,100,100,0,0,1,1,0,19,0,0,0,0,0,0,1,0,2,0,0,0,0,452,0,2,0,1,0,4,0,0,0,0,5,1,1,4,0,5,0,0,0,0,1,12,6,13,2,legitimate
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,17,0,0,0,0,0,0,1,0,2,0,0,0,1,206,0,0,0,1,0,5,0,0,0,0,0,1,0,1,1,3,0,0,0,0,0,6,4.333333,9,2,phishing
9,1,0,0,0,0,0,0,0,0,0,0,0,0,0,100,100,0,0,1,0,0,9,0,0,0,0,0,0,1,1,1,0,0,0,2,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,2,0,4,0,0,0,0,0,6,3,5,2,phishing
9,1,0,0,0,0,0,0,0,0,0,0,0,0,0,100,100,0,0,0,0,0,25,0,0,0,0,0,0,1,0,2,0,0,0,0,1,1,0,0,1,225505,1,0,0,0,0,0,0,0,1,1,0,0,0,0,2,4,7.5,19,2,phishing
124,0,0.266129,0.733871,0,17,0,0.032967,0,0,0,1,0,0,25,0,0,0,1,1,0,21,0,0,0,0,0,0,1,0,2,0,0,0,0,1,1,1,0,1,0,2,0,0,0,0,11,0,0,0,0,3,0,1,1,2,0,7,6.857143,14,1,phishing
29,0,0.655172,0.344828,0,3,0,0,0,0,0,1,0,0,100,60,0,0,0,0,0,14,0,0,0,0,0,0,1,0,1,0,0,0,0,1,1,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,4,0,0,0,0,0,4,6.5,11,4,phishing
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,22,0,0,0,0,0,0,1,0,2,0,0,0,1,63,0,0,0,1,0,5,0,0,0,0,0,1,0,0,1,3,0,0,0,0,0,6,6.166667,14,3,phishing
17,0,0.882353,0.117647,0,0,0,0,0,0,0.5,0,0,0,100,33.333333,100,0,0,1,1,0,19,0,0,0,0,0,0,1,0,1,0,0,0,0,1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,1,3,0,0,0,0,0,5,5.6,15,3,phishing
9,0.666667,0.333333,0,0,0,0,0,0,0,0,0,0,50,66.666667,0,0,1,0,0,20,0,0,0,0,0,0,1,0,1,0,0,0,0,1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,1,1,3,0,0,0,0,0,6,6.5,17,3,phishing
96,0,0.875,0.125,0,1,0,0.166667,0,0,0,1,0,100,66.666667,75,0,0,1,1,0,15,0,0,0,0,0,0,1,0,2,0,0,0,0,521,0,0,0,1,0,1,0,0,0,0,0,0,0,1,0,4,0,0,0,0,0,5,6.8,11,3,phishing
```

Figure 4.2. Aperçu du fichier dataset

4.3. Les algorithmes d'apprentissages utilisés

La technique d'apprentissage automatique que nous choisissons d'utiliser est une technique supervisée, car elle est la plus adaptée à notre travail. Comme nous l'avons mentionné au chapitre II, contrairement aux méthodes non supervisées, les techniques supervisées utilisent des prédictions sur des ensembles de données ou tirent des conclusions à partir des données. Dans notre projet, nous devons prédire si le site est d'hameçonnage. Les techniques de supervision utilisent deux méthodes pour prédire les résultats :

1. La classification
2. La régression

La classification classe les données en catégories. Elle est utilisée pour prédire les variables dépendantes binaires via un ensemble de variables indépendantes, tandis que la régression est utilisée pour prédire des variables continues telles que les changements de température. Étant donné que notre méthode n'a que deux valeurs, légitime ou hameçonnage, la classification est donc la technique avec laquelle nous avons travaillé.

La technique de classification contient plusieurs algorithmes chacun a son propre traitement pour classifier les données, nous avons considéré les algorithmes suivants :

1. Foret d'arbres décisionnels (RF).
2. Arbre de décision (DT)
3. Machine à vecteurs de support (SVM)
4. Voisin le plus proche (KNN)
5. Gradient stochastique (SGD)
6. Naïve Bayésiennes (Naïve Bayes)
7. Réseau neurones (NNs)
8. Logistic Regression (LR)

4.4. Evaluation et mesures de performances adoptées

Dans notre étude, nous adoptons la cross-validation [32] comme méthode d'évaluation des modèles puisqu'elle est connue comme la méthode d'estimation la plus fiabile. La cross-validation consiste simplement à diviser le dataset en k échantillons, enlever une de l'apprentissage puis à l'utiliser pour la phase de test et le processus réitéré. Nous avons aussi choisi les mesures de performances les plus utilisées dans le domaine pour l'évaluation de la prédictivité des classifieurs. Toutes ces mesures sont basées sur la matrice de confusion décrites dans le tableau 4.2:

		Prédiction	
		Hameçonnage	**Légitime**
Nature de site	**Hameçonnage**	Vrai positif (VP)	Faux négatif (FN)
	Légitime	Faux positif (FP)	Vrai négatif (VN)

Tableau 4.2. Matrice de confusion.

D'après la matrice de confusion, un vrai positif (VP) indique que le site d'hameçonnage a été bien classifié, contrairement à faux négatif (FN) qui indique qu'il a été classé comme légitime. Un vrai négatif (VN) montre que le site légitime est classé correctement alors qu'un faux positif (FP) indique le contraire.

Voici la liste des mesures adoptés pour la comparaison des classifieurs :

1. **Fiabilité :** La métrique la plus couramment utilisée pour juger un modèle. C'est un rapport entre l'observation correctement prédite et le total des observations. Elle est la mesure de performance la plus intuitive surtout pour l'ensemble de données équilibrées comme celui de notre cas.

$$Fiabilité = \frac{VP + VN}{VP + FP + FN + VN}$$

2. **Précision :** C'est la proportion des pages d'hameçonnage était effectivement identifiées correctement par le classifieur.

$$Précision = \frac{VP}{VP + FP}$$

3. **Rappel :** C'est la proportion des pages d'hameçonnage réelles qui ont été effectivement identifiées par le classifieur.

$$Rappel = \frac{VP}{VP + FN}$$

4. **F1-score :** C'est la moyenne harmonique entre la précision et le rappel. Par conséquent, ce score prend en compte à la fois les faux positifs et les faux négatifs. F1-Score est généralement plus utile que la Fiabilité, surtout si la répartition des classes est inégale

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

4.5. La combinaison des modèles

Afin d'améliorer les résultats, nous pouvons faire la combinaison entre un ensemble de modèles et on obtient alors un nouveau modèle. La combinaison d'un ensemble de modèles peut s'avérer plus performant que chacun d'entre eux pris individuellement. Tout en étant très différents les uns des autres ces modèles se complètent.

Dans le présent travail, nous avons pensé à combiner les meilleurs modèles obtenus pour chaque classe de caractéristiques. Le but étant d'obtenir un modèle plus performant, nous avons donc testé trois types de combinaisons :

4.5.1. Combinaison « And »

C'est la concaténation des modèles avec la fonction « ET logique ». Si on combine deux modèles, le résultat obtenu est comme indiqué dans le tableau 4.3.

Modèle 1	Modèle 2	Modèle final
1	1	1
1	0	0
0	1	0
0	0	0

Tableau 4.3. Combinaison "And".

La valeur 0 dans le tableau indique la légitimité du site alors que la valeur 1 indique que le site est un site d'hameçonnage. Le site ne peut donc pas être hameçonnage sauf si les résultats de classification du 1er modèle et du 2ème est hameçonnage, les autres cas classifieront la page en légitime.

4.5.2. Combinaison « OR »

C'est la combinaison des modèles avec la fonction « OR logique ». Si on combine deux modèles, le résultat obtenu est comme indiqué dans le tableau 4.4.

Modèle 1	Modèle 2	Modèle final
1	1	1
1	0	1
0	1	1
0	0	0

Tableau 4.4. Combinaison "OR".

Selon cette combinaison, le site ne peut être légitime que si les résultats de classification du 1er modèle et du 2ème est légitime, les autres cas classifieront la page en hameçonnage.

4.5.3. Combinaison « Stack »

C'est une méthode d'ensemble dans laquelle les modèles sont combinés à l'aide d'un autre algorithme d'apprentissage automatique. L'idée de base est de former un nouveau modèle entrainé sur les prédictions des modèles de bases.

4.6. Conclusion

Ce chapitre était consacré à la description des différentes étapes de notre étude. Nous avons présenté les étapes de création du dataset et l'ensemble des caractéristiques URL et HTML adoptées. Nous avons présenté aussi les différents classifieurs utilisés ainsi que les mesures de performances adoptées pour l'évaluation et la comparaison. Dans le chapitre suivant, nous allons exposer les résultats de nos expérimentations ainsi que les étapes de l'implémentation de l'extension Google Chrome.

CHAPITRE V

EXPERIMENTATION & IMPLEMENTATION

CHAPITRE V.

EXPERIMENTATION & IMPLEMENTATION

Ce chapitre présente les étapes qui nous ont permis la construction de notre modèle hybride pour la détection des sites l'hameçonnage. Nous commençons par la présentation des outils de programmations utilisées. Ensuite, nous présentons les différents résultats des tests effectués. Nous terminons par la modélisation d'une extension au Google Chrome pour la détection passive des risques d'hameçonnage.

5.1. Outils d'implémentation

Nous avons utilisé différents outils pour l'implémentation et le test de notre modèle proposé et le développement de l'extension de Google Chrome. Les expérimentations et l'implémentation ont été élaborées sous le système d'exploitation Windows 10, sur un laptop Packard Bell avec un processeur Intel (R) de vitesse 1.60 GHz et une RAM de 2,00 Go. L'ensemble des outils utilisés pour l'implémentation sont :

- Nous avons opté pour WEKA [32] pour l'évaluation de la performance des classifieurs examinés. Weka est un logiciel libre en Java qui met en œuvre de nombreux algorithmes d'exploration de données et d'apprentissage automatique. Il contient une riche collection de techniques de modélisation, de regroupement, de classification, de régression et de prétraitement des données.
- Le langage Python (version 3.8) a été utilisé pour développer les scripts pour l'extraction des caractéristiques et aussi pour l'implémentation du modèle final utilisé pour l'extension.
- Pycharm[1] a été utilisé pour coder et exécuter les scripts Python.

[1] L'environnement de développement Pycharm 2020 : https://www.jetbrains.com/fr-fr/pycharm/download/

- La bibliothèque *Beautifulsoup4*[2] de Python a été utilisée pour l'analyse des codes HTML des pages web, ce qui nous a facilité la tâche d'extraction des caractéristiques à base d'HTML.
- L'ensemble des outils *Joblib*[3] est destiné à exécuter des scripts python en parallèle. Dans notre implémentation, *Joblib* est utilisé pour sauvegarder les modèles entrainés afin de pouvoir les charger au cas de besoin. Spécifiquement, *Joblib* a été utilisée pour la combinaison des modèles.
- L'API *Whois*[4] a été utilisée pour l'extraction des caractéristiques URLs basées sur le service externe Whois.
- Les bibliothèques *NumPy*[5], *Panda*[6], et *CSV*[7] ont été utilisées pour la manipulation et la création de l'ensemble de données.
- Les langages Javascript, PHP, et HTML sont utilisés pour coder notre extension.
- La distribution XAMPP[8] est utilisée pour mettre en place un serveur Web local avec les PHP, Apache, et MySQL afin de tester notre extension.

5.2. La sélection des modèles

Pour former le modèle de détection, huit algorithmes d'apprentissage automatique ont été utilisées et leurs performances ont été mesurées avec Weka : forêt d'arbres décisionnels (RF), arbre de décision (DT), machine à vecteurs de support (SVM), plus proche voisin (KNN), gradient stochastique (SGD), naïve bayésienne (Naïve Bays), réseau de neurones artificiels (NNs) et régression logistique (LR). Pour tous ces algorithmes, nous avons adopté les paramètres par défauts proposées par Weka. Dans les paragraphes qui suivent, nous allons présenter les résultats obtenus après avoir appliqués ces algorithmes individuellement sur les différentes classes de caractéristiques URL et HTML et sur la totalité des caractéristiques (URL + HTML).

[2] Bibliothèque BeautifulSoup4 de Python : https://pypi.org/project/beautifulsoup4/
[3] Python Joblib : https://joblib.readthedocs.io/
[4] API Whois pour Python: https://pypi.org/project/whois/
[5] Bibliothèque Numpy : https://numpy.org/
[6] Bibliothèque Panda : https://pandas.pydata.org/
[7] Bibliothèque CSV : https://docs.python.org/3/library/csv.html
[8] Distribution XAMPP : https://www.apachefriends.org/fr/index.html

5.2.1. Foret d'arbres décisionnels (RF).

Class des caractéristiques	Matrice de Confusion				Précision (%)	Rappel (%)	Fiabilité (%)	F1-Score (%)
	VP (%)	FN (%)	VN (%)	FP (%)				
HTML+ URL	67,98	02,94	27,54	01,52	95,5	95,50	95,53	95,50
HTML	64,44	08,42	22,06	05,07	0,86,3	86,50	86,50	86,30
URL	67,48	03,13	27,34	02,02	0,94,8	94,80	94,83	94,80

Tableau 5.1. Performance de l'algorithme Foret d'arbres décisionnels.

Le tableau montre que RF est plus fiable lorsqu'il on utilisent la totalité des caractéristiques extraites où sa fiabilité dépasse 95%. En plus, RF est moins fiable avec des classes des caractéristiques séparées.

5.2.2. Arbre de décision (DT)

Class des caractéristiques	Matrice de Confusion				Précision (%)	Rappel (%)	Fiabilité (%)	F1-Score (%)
	VP (%)	FN (%)	VN (%)	FP (%)				
HTML + URL	66,36	03,60	26,88	03,15	93,20	93,20	93,24	93,20
HTML	62,85	11,30	19,18	06,65	081,60	82,00	82,04	81,60
URL	66,52	03,69	26,79	02,98	93,30	93,30	93,32	93,30

Tableau 5.2. Performance de l'algorithme arbre de décision.

L'algorithme DT est plus fiable lorsque nous utilisons les caractéristiques de la classe URL uniquement et moins fiable avec les caractéristiques de la classe HTML.

5.2.3. Machine à vecteurs de support (SVM)

Class des caractéristiques	Matrice de Confusion				Précision (%)	Rappel (%)	Fiabilité (%)	F1-Score (%)
	VP (%)	FN (%)	VN (%)	FP (%)				
HTML+URL	64,35	07,56	22,92	05,16	87,10	87,30	87,27	87,10
HTML	58,68	17,70	12,78	10,82	69,90	71,50	71,47	70,30
URL	64,51	06,10	24,38	04,99	88,80	88,90	88,90	88,80

Tableau 5.3. Performance de l'algorithme Machine à vecteurs de support.

L'algorithme SVM est moins fiable que RF et DT. Il donne sa meilleure valeur de fiabilité 88.80% avec la classe des caractéristiques URL.

5.2.4. Voisin le plus proche (KNN)

Class des caractéristiques	Matrice de Confusion				Précision (%)	Rappel (%)	Fiabilité (%)	F1-Score (%)
	VP (%)	FN (%)	VN (%)	FP (%)				
HTML+ URL	62,97	06,65	23,82	06,97	86,80	86,80	86,80	86,80
HTML	62,78	10,59	19,89	06,72	82,30	82,70	82,68	82,30
URL	64,15	05,35	25,01	05,35	89,20	89,20	89,17	89,20

Tableau 5.4. Performance de l'algorithme de voisin le plus proche.

L'algorithme KNN est plus performant que SVM ; il a donné sa meilleure performance (89.20% de fiabilité) avec la classe des caractéristiques URL.

5.2.5. Gradient stochastique (SGD)

Class des caractéristiques	Matrice de Confusion				Précision (%)	Rappel (%)	Fiabilité (%)	F1-Score (%)
	VP (%)	FN (%)	VN (%)	FP (%)				
HTML + URL	64,73	07,03	23,45	04,77	88,00	88,20	88,19	88,10
HTML	58,61	17,58	12,90	10,89	70,00	71,50	71,52	70,40
URL	64,26	07,69	22,79	05,24	86,90	87,10	87,06	86,90

Tableau 5.5. Performance de l'algorithme gradient stochastique.

L'algorithme SGD est plus fiable lorsque nous utilisons une combinaison des caractéristiques URL et HTML avec un taux de fiabilité de 88.10%.

5.2.6. Naïve bayésienne (Naïve Bayes)

Class des caractéristiques	Matrice de Confusion				Précision (%)	Rappel (%)	Fiabilité (%)	F1-Score (%)
	VP (%)	FN (%)	VN (%)	FP (%)				
HTML + URL	42,20	01,85	28,63	27,30	82,20	70,80	70,85	71,90
HTML	65,64	21,98	08,50	03,86	73,00	74,10	74,14	70,20
URL	35,68	01,38	29,09	33,8	81,00	64,80	64,78	65,50

Tableau 5.6. Performance de l'algorithme Naïve Bayésienne.

L'algorithme Naïve Bayes est le moins performant, sa meilleure performance (71.90%) a été marquée lors de l'utilisation de la combinaison des classes des caractéristiques HTML et URL.

5.2.7. Réseau de neurones artificiels (NNs)

Class des caractéristiques	Matrice de Confusion				Précision (%)	Rappel (%)	Fiabilité (%)	F1-Score (%)
	VP (%)	FN (%)	VN (%)	FP (%)				
HTML + URL	64,48	5,28	25,20	05,02	89,70	89,70	89,69	89,70
HTML	61,52	15,15	15,32	07,98	75,80	76,90	76,85	75,90
URL	64,51	06,10	24,38	04,99	88,80	88,90	88,90	88,80

Tableau 5.7. Performance de l'algorithme réseau de neurones artificiels.

L'algorithme NNs marque une fiabilité maximale de 89.70% lorsque l'on utilise la totalité des caractéristiques.

5.2.8. Régression logistique (LR)

Class des caractéristiques	Matrice de Confusion				Précision (%)	Rappel (%)	Fiabilité (%)	F1-Score (%)
	VP (%)	FN (%)	VN (%)	FP (%)				
HTML + URL	64,93	05,63	24,85	04,57	89,70	89,80	89,79	89,70
HTML	64,40	17,96	12,52	05,46	76,00	76,90	76,93	74,80
URL	63,90	07,11	23,36	05,60	87,10	87,30	87,28	87,20

Tableau 5.8. Performance de l'algorithme régression logistique.

L'algorithme LR est plus fiable lorsque l'on utilise la totalité des caractéristiques HTML et URL avec un taux de fiabilité de 89.70%.

Après avoir essayé plusieurs classifieurs, nous avons trouvé que l'algorithme de forêt d'arbres décisionnels (RF) est le mieux adapté, que ce soit pour les classes séparés de caractéristiques ou bien avec la combinaison de toutes les caractéristiques. Ces

résultats nous ont amenés à utiliser ce classifieur pour les tests des différentes combinaisons de modèles.

5.3. La combinaison des modèles

Dans cette partie d'étude nous expérimentons trois types de combinaisons des meilleurs classifieurs de chaque classe de caractéristiques. Comme RF été le meilleur classifieurs des classes de caractéristiques HTML et URL, nous testons la combinaison des deux modèles RF ; chacun étant entrainé sur une classe de caractéristiques différente.

Type de combinaison	Matrice de confusion				Précision (%)	Rappel (%)	Fiabilité (%)
	VP (%)	VN (%)	FP (%)	FN (%)			
URL '**AND**' HTML	90.79	95.59	4.41	9.21	95.37	90.79	92.27
URL '**OR**' HTML	99.56	67.26	32.74	0.44	75.25	99.56	89.70
URL '**Stack**' HTML	99.16	89.77	10.23	0.84	90.65	99.16	96.36

Tableau 5.9. Performance des différentes combinaisons de RF

Comme le montre le tableau 5.9., la combinaison *Stack* des modèles nous a donné une fiabilité relativement élevée 96,36% dans la détection des sites Web d'hameçonnage, qui a atteint un taux de véritable positif de plus de 99,16% et un taux de faux négatif de 0,84%. De plus, la fiabilité, la précision et le rappel sont respectivement de 96,36%, 90,65% et 99,16%, de très bons scores comparés aux autres combinaisons.

5.4. Choix du meilleur modèle

Le but de cette étude est d'éliminer, voire réduire le risque d'accéder aux sites d'hameçonnage (c-à-d, réduire les faux négatifs). Pour cela et après l'analyse des résultats obtenues, nous avons opté pour la combinaison « *Stack* » des deux modèles basés sur l'algorithme de Foret d'arbres décisionnels (RF), chacun entrainé sur une

classe différente de caractéristiques. Ce dernier donne une valeur très élevée de vrais positifs (99.16%) avec un taux raisonnable de faux négatifs (0,84%).

Pour intégrer ce modèle dans une extension dans le navigateur Google Chrome, on était obligés de le sauvegarder sous une forme permettant de le recharger plus tard. Pour cela, nous avons utilisé le langage Python pour l'enregistrement du modèle au format PKL. Ce dernier nous permet de sauvegarder le modèle dans un fichier et de le charger plus tard pour la prédiction des nouvelles instances d'URL.

Un fichier PKL est un fichier créé par *pickle*, un module Python qui permet aux objets d'être sérialisés en fichiers sur le disque et de les désérialiser dans le programme au moment de l'exécution. Il contient un flux d'octets qui représente les objets [37].

5.5. Processus de développement de l'extension

On a décidé de créer une extension au lieu d'un logiciel, car les extensions sont faciles à utiliser et à mettre à jour. Les extensions permettent aux navigateurs la détection instantanée des sites d'hameçonnage visités par les utilisateurs. Pour cela, nous avons suivi le processus suivant pour le développement de cette extension :

1. Préparer un logo pour l'extension
2. Construire un fichier JSON nommé *'manifest.json'* auquel on a ajouté le nom de l'extension, le logo, la description de l'extension et la version.
3. Créer un autre fichier HTML nommé *'popup.html'* qui contient le format HTML du menu qui sera affiché pour indiquer le type d'URL.
4. Etablir un fichier Javascript nommé *'popup.js'*, qui permet de récupérer l'URL depuis la barre d'adresse du navigateur et afficher le résultat de la détection renvoyé par le modèle.
5. Enfin un dernier fichier PHP nommé *'Serveur.php'* joue le rôle d'un intermédiaire entre le modèle et le code Javascript.

Le schéma suivant explique le fonctionnement de l'extension :

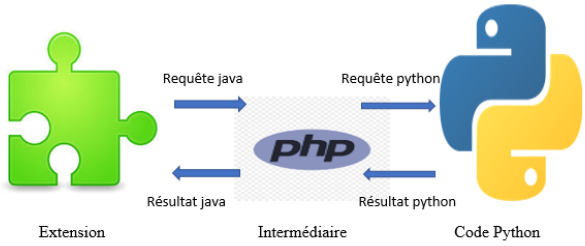


Figure 5.1. Fonctionnement de l'extension.

Pour installer l'extension dans Google Chrome, on suit les étapes suivantes :

1. Ouvrir le navigateur Google Chrome
2. Cliquer sur personnaliser et contrôler
3. Cliquer sur plus d'outils
4. Cliquer sur extension, une nouvelle fenêtre va s'ouvrir, on active le mode développeur, ensuite on clique sur charger l'extension non empaquetée et enfin on choisit l'emplacement de l'extension.

Voici un aperçu du navigateur Google Chrome après l'installation de notre extension :

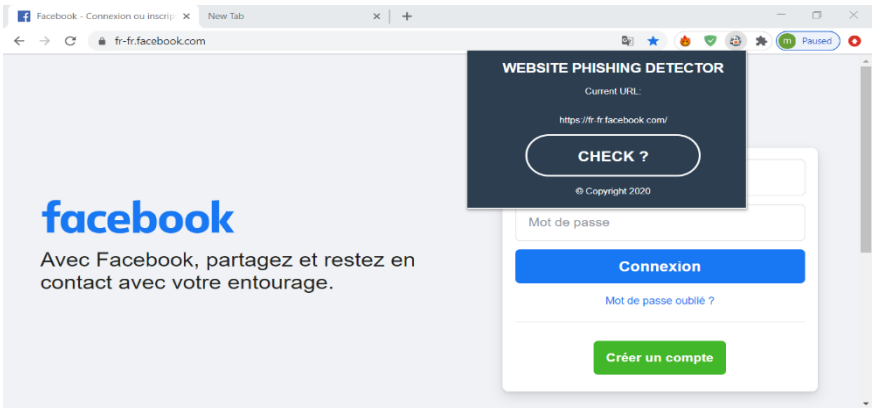


Figure 5.2. Résultat de l'installation de l'extension.

5.5.1. Diagramme de séquence de l'extension

Pour présenter le fonctionnement de l'extension, on a choisi le diagramme de séquence car il décrit simplement les interactions entre les objets dans un ordre séquentiel, c'est-

à-dire l'ordre dans lequel ces interactions ont lieu. Cela montre clairement le mode de fonctionnement de notre extension et ces différentes interactions.

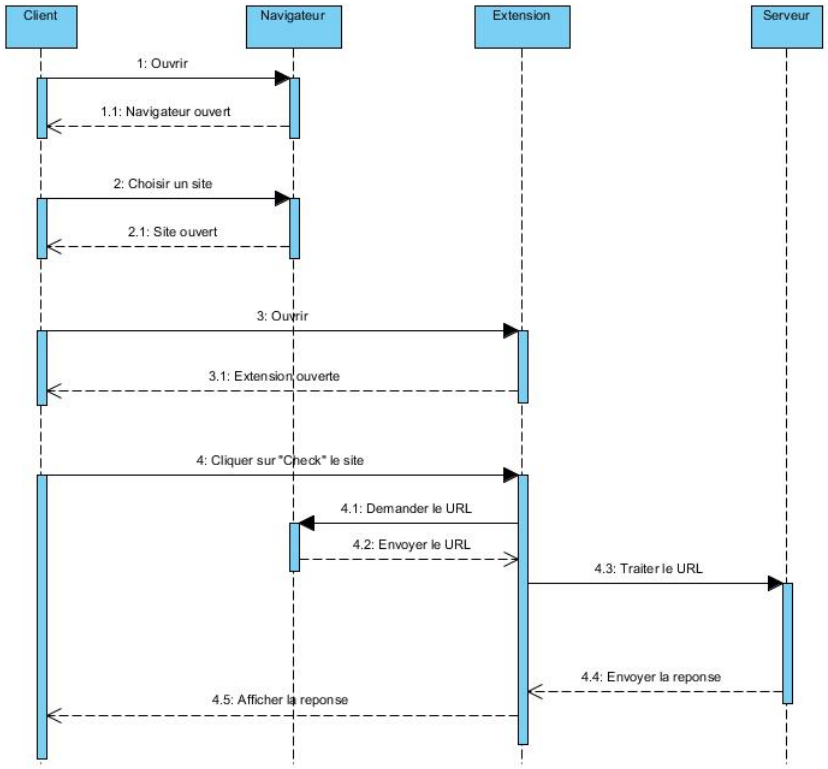


Figure 5.3. Diagramme de séquence - Mode de fonctionnement de l'extension développée.

Depuis le diagramme de séquence :

1. Le client d'abord va ouvrir le navigateur.
2. Le client choisi le site qu'il veut visiter dans le navigateur.
3. Le navigateur affiche le site voulu.
4. Le client sélectionne l'extension pour détecter la légitimité de la page ouverte.
5. Une fenêtre contextuelle de l'extension sera affichée.
6. Le client clique sur vérifier le site (bouton '*CHECK*').
7. L'extension récupère l'URL de la page ouverte depuis la barre d'adresse du navigateur.
8. L'extension envoie l'URL au modèle situé dans le server pour traiter l'URL (construire le vecteur caractéristique et tester l'URL).

9. Le serveur obtient la réponse de notre modèle et l'envoie à l'extension.
10. L'extension affiche le résultat obtenu au client.

5.5.2. Interface de l'extension

L'image de la figure 5.4 présente le comportement de l'extension dans le cas d'une page web légitime qui s'affiche en vert.

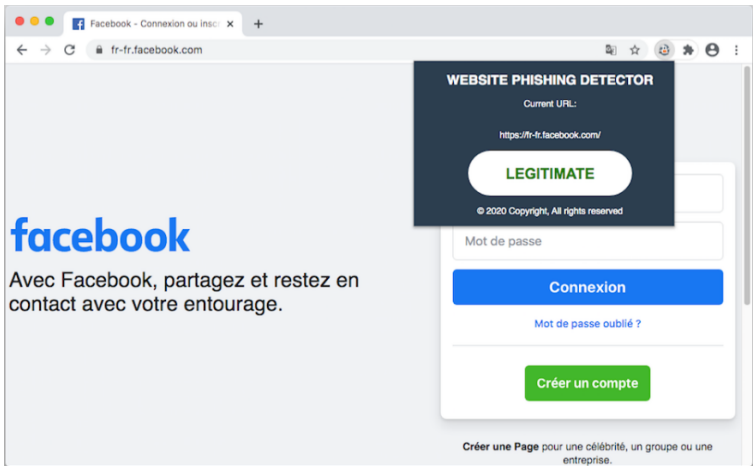


Figure 5.4. Comportement de l'extension dans le cas d'un site Web légitime.

La Figure 5.5 présente le comportement de l'extension dans le cas d'une page web d'hameçonnage quoi s'affiche en rouge. Cette page contient des fonctionnalités qui ont été classées malveillantes aussi avec l'avertisseur intégré de Google Chrome.

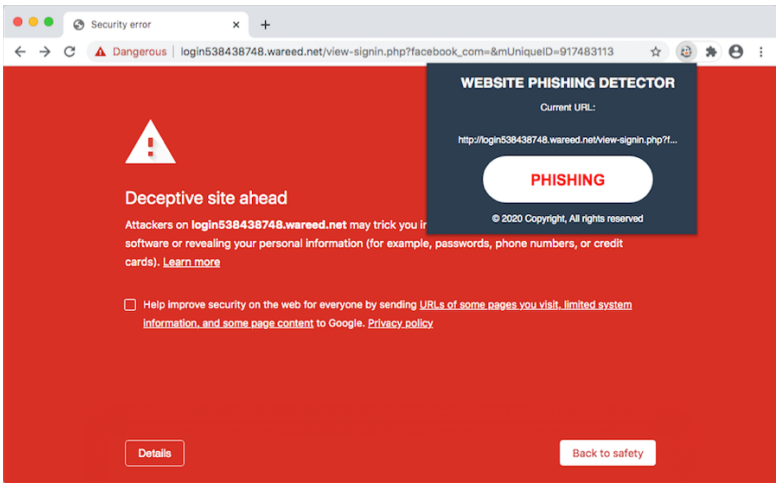


Figure 5.5. Comportement de l'extension pour d'un site Web d'hameçonnage.

La Figure 5.6 présente le comportement de l'extension dans le cas d'une page inaccessible.

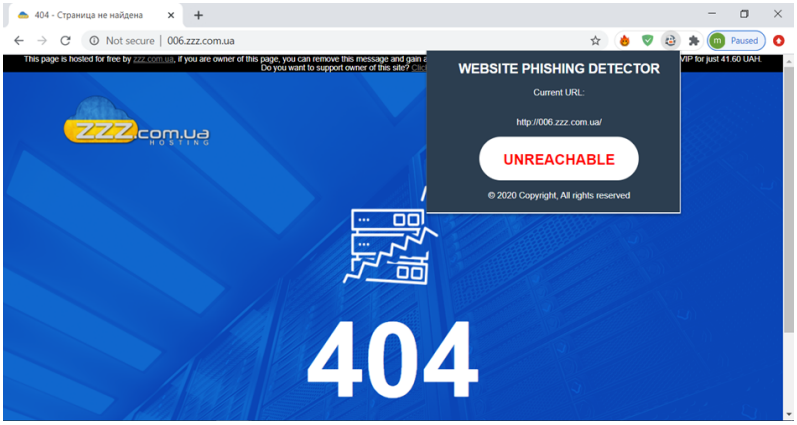


Figure 5.6. Comportement de l'extension au cas d'un site Web inaccessible.

5.6. Conclusion

Dans ce chapitre, nous avons présenté les différentes étapes qu'on a suivi pour la sélection du meilleur modèle de classification ainsi que la construction de notre extension. L'étape la plus importante était l'extraction des caractéristiques, où 61 caractéristiques étaient appliquées dans notre modèle, pour construire les vecteurs de caractéristiques. Pour prédire nos résultats, on a utilisé une combinaison de deux modèles de type foret d'arbres décisionnels, chacun étant entrainé sur une classe différente des caractéristiques. Cette sélection est due aux taux de fiabilité assez élevée obtenu par ce modèle. Malgré les bons résultats obtenus par notre système qui est estimé de 96,36% de fiabilité, la contrainte du temps était très importante, des améliorations doivent donc être faites dans les fonctionnalités HTML pour une prédiction plus rapide.

CONCLUSION GENERALE

CONCLUSION GENERALE

Notre étude a montré que les techniques d'apprentissage automatiques sont vraiment capables d'alléger le problème d'hameçonnage informatique. Spécifiquement, les expérimentations élaborées dans ce mémoire montrent l'efficacité de l'algorithme forêt d'arbres décisionnels par rapport aux autres algorithmes d'apprentissage. En plus, l'utilisation des diverses caractéristiques d'URL et HTML en même temps est plus efficace que l'utilisation des caractéristiques d'une seule classe, ce qui montre l'efficacité des approches hybrides dans la détection des sites d'hameçonnage.

Dans ce mémoire, nous avons opté pour l'utilisation de caractéristiques diverses et la combinaison des modèles. Les résultats montrent que la combinaison *stack* des deux classifieurs de type forêt d'arbres décisionnels, chacun entrainé sur une classe différente de caractéristiques donne une meilleure fiabilité de 96.36%. Mettant nos résultats en pratique, nous avons développé une extension dans Google Chrome qui permet à l'utilisateur de vérifier la légitimité des pages visitées.

Nos expérimentations montrent aussi que le choix des modèles efficaces et des caractéristiques discriminatives reste un problème majeur pour ces approches. L'efficacité des modèles est étroitement liée à l'ensemble de données utilisé (dataset) pour entrainer et tester les modèles ; on constate l'absence des benchmarks pour l'évaluation des modèles dans le domaine d'hameçonnage. En plus, chaque jour on signale des milliers de nouveaux sites d'hameçonnage avec une évolution rapide de leurs techniques. Cela explique le fait que les mécanismes et les modèles existants ne sont pas suffisants pour détecter les nouvelles attaques d'hameçonnage ; ce qui nécessite des études exhaustives et régulières de ces techniques et des adaptations périodiques des modèles d'apprentissage. Les approches hybrides ont des limites puisqu'elles ne peuvent pas être utilisées pour la détection instantanée des sites d'hameçonnage vu le temps nécessaire pour l'élaboration des vecteurs de caractéristiques de chaque page visitée.

Par conséquent, nous souhaitons à l'avenir explorer la robustesse des techniques de *Deep Learning* pour la détection de l'hameçonnage. Aussi nous planifions de réduire le temps de prédiction de notre modèle pour avoir des avertissements instantanés lors de la visite des sites suspects.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

[1] Roman Jakobson, *Essais de linguistique générale*, Editions de Minuit, 1981

[2] APWG, *Phishing Activity Trends Report*: *Unifying the Global Response To cybercrime*,
2nd Quarter, Anti-Phishing Working Group, 2020.

[3] Belhait Sara, *Détection des sites de phishing avec la forêt d'arbre décisionnels: HTML et URL.*, Master thesis, Badji Mokhtar University of Annaba, 2019.

[4] Wu Min, *Fighting phishing at the user interface*, PhD thesis, Massachusetts Institute of Technology, 2006.

[5] Gaurav Varshney, Manoj Misra, and Pradeep K. Atrey. *A survey and classification of web phiching detection schemes*, Security and communication Networks, 9(18) pp. 6266-6284, 2016.

[6] Lutz M. and Biernat E. *Data science: fondamentaux et études de cas: Machine learning avec Python et R*. Editions Eyrolles. 2015.

[7] Arun Kulkarni and Leonard L. Brown, *Phishing Websites Detection using Machine Learning*, International Journal of Advanced Computer Science and Applications, 10(7), pp.8-13, 2019.

[8] Deshmukh A., Raju K. D., Ravula R., and Kumar D. M. U. *Cyber Security Engineering for Malware Analysis-Machine Learning for Spam Detection Case Study*, Journal of engineering sciences, 10(10), pp.301-306, 2019

[9] Zine-Laabidine O., *Plateforme de développement pour l'Internet des objets (IdO) avec un apprentissage automatique*, Master thesis, Université 8 Mai 1945, Guelma, 2019.

[10] Han M. and Zhang H., *Business intelligence architecture based on internet of things*, Journal of Theoretical and Applied Information Technology, 50(1), pp. 90-95, 2013.

[11] Ying Pan and Xuhua Ding. *Anomaly Based Web Phishing Page Detection*, in proceedings of Annual Computer Security Applications Conference, pp. 381-392, 2006.

[12] Bhattacharya Suman, Chetan Kumar, and Praveen Kumar. *Detecting Phishing Websites a Heuristic Approach*. International Journal of Latest Engineering Research and Applications, 2(3), pp. 120-129, 2017.

[13] Medvet Eric, Engin Kirda, and Christopher Kruegel. *Visual-similarity-based phishing detection*. In proceedings of the 4th international conference on Security and privacy in communication netowrks, ACM, pp. 1-6, 2008.

[14] Rao Routhu Srinivasa, and Alwyn Roshan Paie. *Detection of phishing websites using an efficient feature-based machine learning framework*, Neural Computer Science, pp. 147-156, 2015.

[15] Rao Routhu Srinivasa, and Syed Taqi Ali. *Phishshield: a desktop application to detect phishing webpages through heuristic approach.* In Procedia Computing and Applications, vol.55, pp. 147-156, 2018.

[16] Jain Ankit Kumar, and Brij B. Gupta. *A machine learning based approach for phishing detection using hyperlinks information*, Journal of Ambient Intelligence and Humanized Computing, vol. 10, pp. 2015-2028, 2019.

[17] Stiawan Deris, and Zaini N., *Phishing detection system using machine learning classifiers*. Indonesian Journal of Electrical Engineering and Computer Science, v17(3), pp. 1165-1171, 2020.

[18] Hossein Shirazi. *Unbiased phishing detection using domain name-based features*. PhD thesis, Colorado State University, 2008.

[19] Zhang J., Porras P. A., and Ullrich J., *Highly Predictive Blacklisting*. In USENIX Security Symposium, 33(6), pp. 107-122, 2008.

[20] Jagadeesan S., Chaturvedi A., and Kumar S., *URL Phishing Analysis using Random Forest*. Int. J. Pure Appl. Math, 118(20), pp. 4159-4163, 2018.

[21] Prakash P., Kumar M., Kompella R. R., and Gupta M. *Phishnet: predictive blacklisting to detect phishing attacks*. In Proceedings of IEEE INFOCOM, IEEE, pp. 1-5, 2010.

[22] Patel J., *Design and Implementation of Heuristic based Phishing detection technique*. Master thesis, University of Victoria, British Columbia, Canada, 2018.

[23] Liu C., Wang L., Lang B., and Zhou Y., *Finding effective classifier for malicious URL detection*. In Proceedings of the 2nd International Conference on Management Engineering, Software Engineering and Service Sciences, ACM, pp. 240-244, 2018.

[24] Jeeva S. Carolin, and Elijah Blessing Rajsingh, *Intelligent phishing URL detection using association rule mining*. Human-centric Computing and Information Sciences, 6(10), pp. 1-19, 2016.

[25] Verma R., and Das A., *What's in a URL: Fast feature extraction and malicious URL detection*. In Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics, ACM, pp. 55-63., 2017.

[26] Sahingoz O. K., Buber E., Demir O., and Diri B., *Machine learning based phishing detection from URLs*. Expert Systems with Applications, 17(1), pp. 345-357, 2019.

[27] Lei Chen, Yan Jia, Timos Sellis, and Guanfeng Lie, *Web Technologies and Applications*, In the 16th Asia-Pacific Web Conference, APWeb, Changsha, China, Springer, 2014.

[28] Colin Whittaker, Brain Ryner, and Marria Nazif., *Large-Scale Automatic classification of Phishing pages*. In NNDS'10, pp. 1-14, 2010.

[29] Ankit kumar Jain and Gupta B; B.. *A novel approach to protect against phishing attacks at client side using auto-updated white-list*. EURASIP journal on information security, 9(1), pp. 1-11, 2016.

[30] Ankit kumar Jain and Gupta B. B., *Towards detection of phishing websites on client-side using machine learning based approach*, Journal of telecommunication systems, 68(4), pp. 687-700, 2018

[31] Gillot P., Zemmari A., Benois-Pineau J., and Nesterov Y., *Algorithmes de Descente de Gradient Stochastique avec le filtrage des paramètres pour l'entraînement des réseaux à convolution profonds*, In Congré AFIAP et CFPT, Marne-la-Vallée, France, pp. 1-7, 2018.

[32] Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., and Witten I. H., *The WEKA data mining software: an update*. *ACM SIGKDD explorations newsletter*, *11*(1), pp. 10-18, 2009.

[33] Shirazi H., Bezawada B., Ray I., *Know thy domain name: unbiased phishing detection using domain name based features*, In Proceedings of the 23rd ACM on Symposium on Access Control Models and Technologies, ACM, pp. 69-75, 2018.

[34] Chou N., Ledesma R., Teraguchi Y., Boneh D., and Mitchell J. C., *Client-side defence against web-based identity theft*. In *NDSS*. The Internet Society, 2004.

WEBOGRAPHIE

[35] Dictionnaire français, *Dictionnaire Larousse français monolingues et bilingue*, https://www.larousse.fr/dictionnaires/francais [consulté Mars 2020]

[36] Webopedia, *What is phishing?* https://www.webopedia.com/TERM/P/phishing.html [consulté Janvier 2020]

[37] Python.org, Pickle Python object serialization, https://docs.python.org/3/library/pickle.html [consulté Aout 2020]

[38] Matlab, Simulink, 3 choses à savoir, Machine learning, https://fr.mathworks.com/discovery/machine-learning.html, [consulté Avril 2020].

[39] Guru99, Sepervised vs unsupervised learning :algorithms, example, difference, https://www.guru99.com/supervised-vs-unsupervised-learning.html [consulté Mars 2020].

[40] George Seif, The 5 clustering algorithmes data scientists need to know, https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68 [consulté Avril 2020].