



MEMOIRE

Présenté à

**L'Institut Supérieur des Sciences Appliquées et de
Technologie de Gafsa**

(Département Informatique et télécommunication)

En vue de l'obtention Diplôme en Expert Cyber Sécurité

MASTERE

Dans la discipline Expert Cyber Sécurité

Par

Moez KHLIFI

SYSTEME INTELLIGENT DE DETECTION D'HAMEÇONNAGE BASE SUR L'APPRENTISSAGE AUTOMATIQUE

Soutenu devant le jury composé de :

Mme	Fatma HRIZI
M.	Said TAIEB
Mme	Hayfa CHARFI
M.	Ahmed KHLIFI

<i>Président</i>
<i>Rapporteur</i>
<i>Encadreur</i>
<i>Co-Encadreur</i>

A.U : 2022 – 2023

Dédicaces

A mon père Abdallah qui par sa présence, son sérieux, et ses précieux conseils m'a permis d'arriver là où je suis. Papa qui est toujours là derrière moi, je ne saurais te remercier assez pour tout ce que tu fais pour nous, tu es le meilleur des exemples, Merci.

A ma mère Hawa, qui ne cesse de nous pousser d'avantage, qui s'est sacrifiée pour nous et a toujours su nous épauler et nous soutenir dans nos moments difficiles, j'espère être à la hauteur de tes attentes maman.

A mon frère Mondher et ma sœur Malek, qui m'ont beaucoup aidés et soutenu chacun à sa façon surtout en cette année chargée d'évènements.

À tous les membres de la famille Khlifi.

Et tous mes collègues, mes amis.

Moez

Remerciements

Je tiens premièrement à prosterner remerciant Allah le tout puissant de m'avoir donné le courage et la patience pour terminer ce travail, ce mémoire n'aurait jamais été réalisé sans sa bénédiction.

Je tiens également à remercier chaleureusement mes encadrants, Mme Hayfa Charfi et M. Ahmed Khlifi, pour leur soutien constant et leurs précieux conseils tout au long de l'élaboration de ce projet. Leur encadrement attentif a grandement contribué à sa réussite.

Je tiens à exprimer l'honneur que me font les membres du jury pour avoir accepté de me prêter leur attention et évaluer ce projet.

Je remercie tous les enseignants pour leur soutien le long de ces années d'études à l'institut supérieur des sciences appliquées et de technologies de Gafsa. (ISSAT) pour la réalisation de ce travail.

Finalement, je remercie toute personne ayant contribué de près ou de loin à l'accomplissement de ce travail et qui m'a été bénéfique durant mon parcours pour sa réalisation.

Table des matières

Introduction générale.....	1
Chapitre 1 : Etat De L’art.....	2
Introduction	3
1. Ce qu’est l’hameçonnage	3
1.1 Définition.....	3
1.2 les nuances entre hameçonnage, vol d'identité et fraude liée à l'identité.....	3
2. La prévalence des attaques de phishing.....	4
3. Les victimes de l’hameçonnage.....	6
3.1 Les utilisateurs individuels : Un large éventail de cibles	6
3.2 Les organisations de toutes tailles : Des cibles lucratives	6
4. L’étape d’attaque de phishing	7
5. Les types d'attaques d'hameçonnage	7
6. La nécessité de la détection d'hameçonnage	9
6.1 L'ampleur croissante des attaques d'hameçonnage.....	9
6.2 La protection des données et de la vie privée	10
7. Attaques d'hameçonnage ciblant les grandes sociétés.....	11
7.1 Dangers des attaques d'hameçonnage sur les entreprises : Risques et impacts	11
7.2 Exemples d'attaques d'hameçonnage contre les entreprises	12
8. L'intelligence artificielle dans le cybersécurité	13
8.1 Définition.....	13
8.2 L'impact de l'intelligence artificielle sur la cybersécurité	13
8.3 Le Machine Learning et le Deep Learning dans la cybersécurité	14
Conclusion.....	17
Chapitre 2 : Les Techniques De Détection D’hameçonnage.....	18
Introduction	19
1. Les solutions existantes de système de détection d'hameçonnage	19
1.1 Filtres anti-spam	19
1.2 PhishBlock.....	22
2. Modèle de détection de phishing proposé	25
2.1 Contexte et problématique.....	25
2.2 Architecture de système proposée	25
2.3 L’apprentissage	26
3. Solution proposée	27

4. Algorithme de classification.....	28
4.1 L'Algorithme de modèle Naïve Bayes	28
4.2 Algorithme de modèle KNN	29
4.3 L'Algorithme de modèle Machine à Vecteur de Support (SVM)	30
Conclusion.....	31
Chapitre 3 : Étude Expérimentale.....	32
Introduction	33
1. Préparation de base d'apprentissage.....	33
2. Apprentissage des modèles de Machine Learning	34
2.1 Le Dataset.....	34
2.1.1. Définition.....	34
2.1.2. Principales sources pour accéder à des datasets	34
2.1.3. Caractéristiques du dataset utilisé et son utilisation	35
2.2 Les outils utilisés	36
3. Expérimentations.....	37
3.1 Importation des bibliothèques et chargement des données.....	37
3.2 Les informations détaillées sur le dataset	39
4.3 Préparation et division des données : Détection d'hameçonnage	40
4.4 Phase d'apprentissage et validation.....	40
4.4.1 La régression logistique.....	40
4.4.2 Les K plus proches voisins (KNN).....	41
4.4.3 Le Machine à Vecteurs de Support (SVM)	43
4.4.4 Le Naïve Bayésien (NB)	44
4.4.5 Application de modèle.....	45
4. Comparaison des modèles	46
4.1 Création d'un DataFrame	46
4.2. Comparaison des résultats	47
4.3 Évaluation des performances des modèles de détection d'hameçonnage	47
Conclusion.....	48
Conclusion générale	49
Références bibliographiques	50

Liste des figures

Figure 1: Le statistique des attaques informatiques en 2023.....	4
Figure 2: Mise en évidence d'un rapport du Groupe de travail de l'Anti Phishing.....	5
Figure 3: Statistiques sur l'hameçonnage pour le 1 ^{er} trimestre 2023.....	5
Figure 4: Anatomie d'un email de phishing	8
Figure 5: Le processus d'attaque par Spear Phishing	8
Figure 6 : Stratégies de phishing	10
Figure 7: Technique de détection de phishing basée sur les fonctionnalités.....	11
Figure 8: Les 12 technologies de l'intelligence artificielle	14
Figure 9 : Le processus de fonctionnement du Machine Learning.....	15
Figure 10: Le processus de fonctionnement de Deep Learning	16
Figure 11: Représentation des dispositifs de filtrage anti spam des messages email.....	19
Figure 12: Le fonctionnement des systèmes de filtrage des e-mails	22
Figure 13: Le fonctionnement de PhishBlock.....	23
Figure 14: Le cycle du phishing de PhishBlock.....	24
Figure 15: Modèle Naïve Bayes.....	29
Figure 16: Fonctionnement d'un KNN.....	30
Figure 17 : Utilisation de l'algorithme SVM.....	31
Figure 18: Logo de Python.....	36
Figure 19: Logo de Google Colab.....	37
Figure 20: Importation des bibliothèques et les utilitaires	38
Figure 21: Chargement des données de dataset.....	38
Figure 22: Les dimensions (la forme) du dataset chargé et la liste des noms de colonnes	39
Figure 23: Le nombre d'entrées et le type de données de chaque colonne	39
Figure 24: Répartition des données en ensembles d'apprentissage	40
Figure 25: Apprentissage de modèle avec de régression logistique.....	41
Figure 26: Evaluer les performances du modèle de régression logistique	41
Figure 27: Apprentissage de modèle avec KNN	42
Figure 28: Evaluer les performances du KNN	42
Figure 29: le résultat de précision d'entraînement et de test de KNN	43
Figure 30: Entraînement d'un modèle SVM.....	43
Figure 31: Evaluer les performances du modèle SVM.....	44
Figure 32: Utilisation du classifieur Naïve Bayésien pour l'apprentissage de modèle.....	44
Figure 33: Analyse des performances du modèle Naïve Bayésien.....	45
Figure 34: Serveur de développement Flask en cours d'exécution.....	45
Figure 35 : Interface graphique de site web de détection d'hameçonnage.....	46
Figure 36: Création un DataFrame en utilisant la bibliothèque pandas	46
Figure 37 : Comparaison des modèles.....	47

Liste d'abréviations

SVM : Support Vector Machine (Machine à Vecteurs de Support)

KNN: K-Nearest Neighbors (K Plus Proches Voisins)

LR : Logistic Regression (Régression Logistique)

NB : Naive Bayes (Classifieur Bayésien Naïve)

ML : Machine Learning (Apprentissage automatique)

DL : Deep Learning (Apprentissage profond)

URL : Uniform Resource Locator (Localisateur Uniforme de Ressources)

Introduction générale

L'avènement des technologies de l'information a considérablement transformé notre société, ouvrant de nouvelles opportunités tout en créant de nouveaux défis en matière de sécurité. Parmi ces défis, la préservation de la confidentialité des données et la protection contre les cybermenaces sont devenues des préoccupations majeures. L'une des formes d'attaque cybercriminelle les plus répandues et les plus préjudiciables est l'hameçonnage, également connu sous le nom de phishing.

Dans ce contexte, notre projet de recherche vise à contribuer à la sécurité en ligne en proposant une approche novatrice pour la détection d'hameçonnage. Notre objectif est de développer des solutions efficaces pour contrer cette menace croissante en utilisant des techniques d'apprentissage automatique.

Dans le cadre de ce mémoire, nous examinerons les dangers spécifiques liés à l'hameçonnage, en mettant particulièrement l'accent sur les conséquences pour les grandes entreprises. Les organisations de grande envergure sont souvent ciblées en raison de leurs ressources et de leur importance, ce qui fait de la détection d'hameçonnage une priorité cruciale pour leur sécurité. Nous présenterons des exemples concrets d'attaques d'hameçonnage ayant eu un impact significatif sur ces entreprises.

La structure de ce mémoire sera divisée en trois chapitres. Le premier chapitre fournira un aperçu de l'état de l'art dans le domaine de la détection d'hameçonnage, en examinant les approches existantes, les cas réels d'attaques dans les grandes entreprises.

Le deuxième chapitre présentera notre méthodologie détaillée, et les solutions existantes et en expliquant les différentes étapes de notre projet de recherche, de la collecte des données à la mise en place des modèles de détection.

Enfin, le troisième chapitre évaluera les performances de notre approche en présentant les résultats obtenus et en discutant des limites de notre étude.

Chapitre 1 : Etat De L'art

Introduction

Avant de rentrer dans le cœur du sujet et de parler des attaques de phishing et des éventuels moyens de protections, il convient de comprendre l'enjeu aujourd'hui autour de la cybersécurité, principalement pour les entreprises. Cette question de cybersécurité un problème récent, mais devenu aujourd'hui quasiment non négligeable, et essentiel au bon développement de l'activité d'une entreprise. Nous allons donc, dans cette partie introductive, essayer de comprendre l'émergence de ce nouvel enjeu dans le contexte actuel.

Nous verrons également les caractéristiques de ce problème nouveau.

1. Ce qu'est l'hameçonnage

Dans cette section, nous définirons ces concepts, soulignerons les subtilités qui les différencient et mettrons en évidence les distinctions clés par rapport à la protection des informations personnelles.

1.1 Définition

L'hameçonnage est une forme de cyberattaque dans laquelle les fraudeurs se font passer pour des entités légitimes, telles que des entreprises, des institutions financières ou des organismes gouvernementaux, afin de tromper les utilisateurs et de leur soutirer des informations sensibles. Les attaquants envoient généralement des e-mails, des messages instantanés ou des appels téléphoniques frauduleux pour inciter les victimes à divulguer des informations personnelles, financières ou d'identification [1].

1.2 les nuances entre hameçonnage, vol d'identité et fraude liée à l'identité

Bien que les termes hameçonnage, vol d'identité et usurpation d'identité soient souvent utilisés de manière interchangeable, il existe certaines différences. L'hameçonnage est l'un des nombreux moyens par lesquels les voleurs d'identité peuvent « voler » des informations par tromperie, en incitant des consommateurs sans méfiance à fournir par inadvertance des informations d'identité ou financières sous de faux prétextes, ou en les incitant à donner à des criminels un accès non autorisé à leurs ordinateurs et à leurs données personnelles. Les États-Unis et certains autres pays utilisent le terme « vol d'identité » ; le Royaume-Uni utilise souvent le terme « fraude d'identité » pour désigner la pratique courante consistant à obtenir et à utiliser frauduleusement les données d'identité d'une autre personne. La fraude liée à l'identité peut aussi désigner l'utilisation criminelle

Subséquente de données d'identification d'autres personnes pour obtenir des biens ou des services, ou l'utilisation de données d'identification fictives (non nécessairement associées à une personne réellement en vie) pour commettre un crime [2].

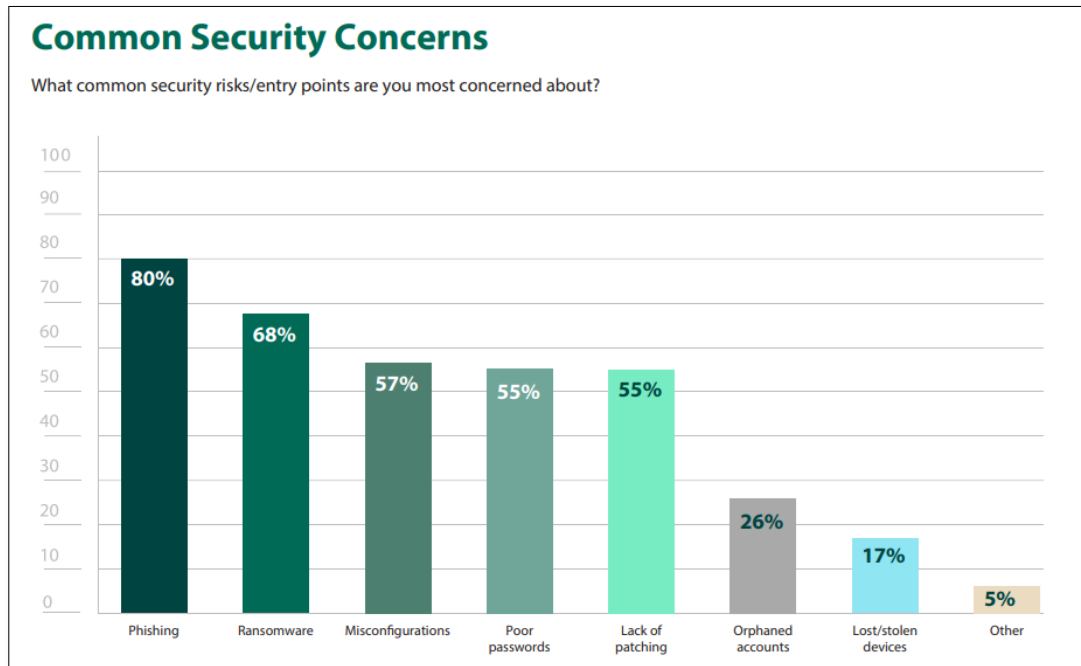


Figure 1: Le statistique des attaques informatiques en 2023 [3]

2. La prévalence des attaques de phishing

Le nombre d'attaques de phishing est en augmentation rapide. En raison de leur valeur, les institutions financières sont les cibles favorites des pirates de phishing.

Récemment, cette forme d'attaque s'est diversifiée et a commencé à cibler les sites de réseautage social ainsi. En outre, les technologies adoptées par les attaquants deviennent plus sophistiquées chaque jour.

La figure 2 présente les faits saillants des statistiques obtenues par une étude récente sur les attaques de phishing dans tout le monde [4].

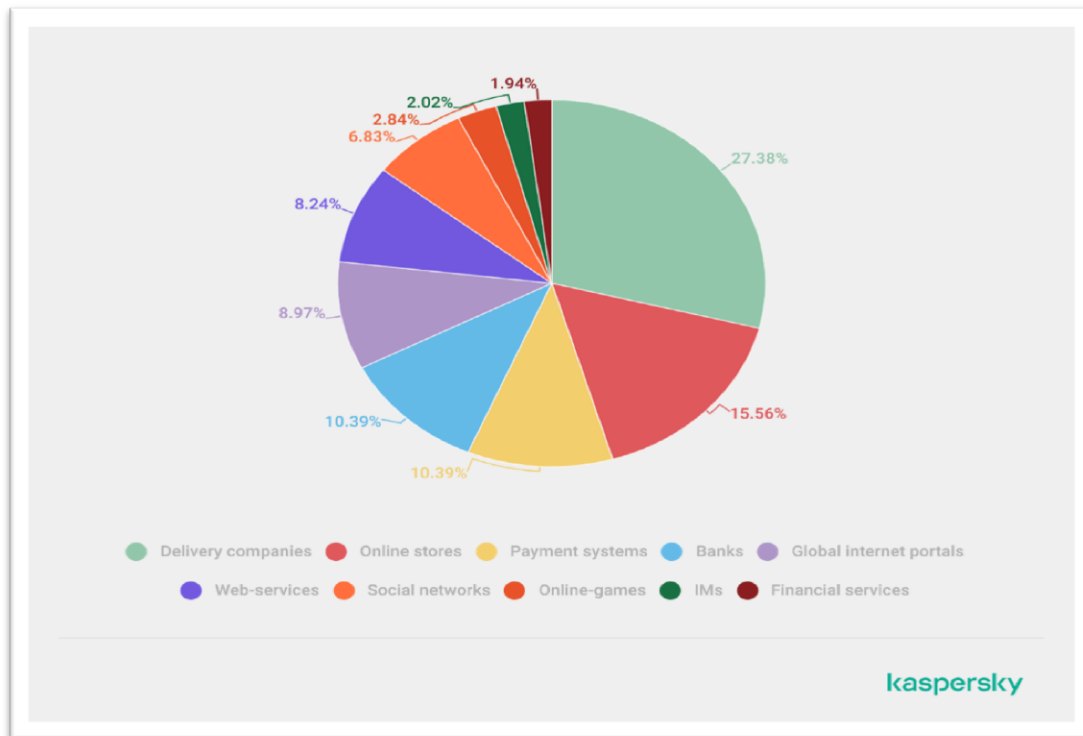


Figure 2: Mise en évidence d'un rapport du Groupe de travail de l'Anti Phishing

Selon le rapport sur les tendances de la sécurité de l'information de Verizon de 2022, le phishing est la première cause des incidents de sécurité des entreprises. En effet, les attaques de phishing représentent plus de 36% des incidents de sécurité signalés. En outre, selon le rapport de la société de sécurité informatique Proofpoint de 2022, près de 88% des organisations ont subi des attaques de phishing au cours de l'année écoulée.

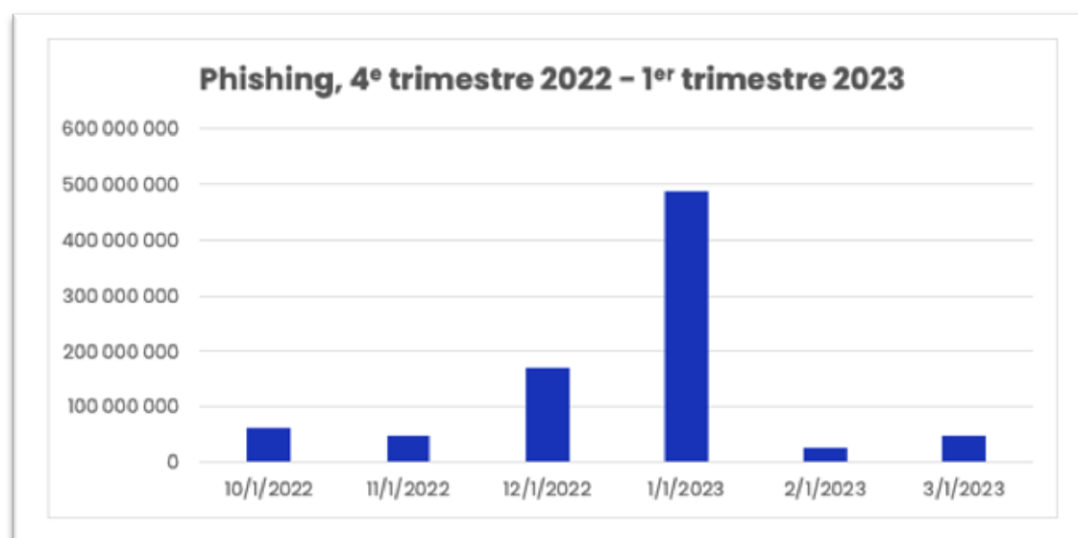


Figure 3: Statistiques sur l'hameçonnage pour le 1er trimestre 2023 [5]

Ce qui concerne les utilisateurs individuels, une étude de Google de 2019 a révélé que les attaques de phishing étaient en constante augmentation. Selon cette étude, près de 1,5% des e-mails reçus par les utilisateurs de Gmail sont des tentatives de phishing. Bien que cela puisse sembler faible, cela représente tout de même un nombre important d'attaques compte tenu du nombre d'utilisateurs de Gmail dans le monde.

3. Les victimes de l'hameçonnage

Dans cette section, notre attention sera particulièrement portée sur les victimes d'hameçonnage. Tout d'abord, nous procéderons à la définition précise de ces concepts afin de bien les cerner. Ensuite, nous soulignerons les subtilités qui les différencient les uns des autres, car il existe différentes formes d'hameçonnage.

3.1 Les utilisateurs individuels : Un large éventail de cibles

Les attaques d'hameçonnage visent tout utilisateur individuel qui utilise des services en ligne sur Internet. Les cybercriminels exploitent la crédulité et la confiance des personnes pour accéder à leurs données confidentielles ou les inciter à effectuer des opérations spécifiques, telles que des versements d'argent. Ainsi, chaque citoyen qui utilise des services en ligne, qu'il s'agisse de la banque en ligne, du shopping en ligne ou des réseaux sociaux, est susceptible d'être une victime potentielle d'une attaque d'hameçonnage. [6]

3.2 Les organisations de toutes tailles : Des cibles lucratives

Les attaques d'hameçonnage ne se limitent pas aux individus, elles visent également les organisations, des PME aux multinationales, quel que soit leur secteur d'activité. Les cybercriminels considèrent toutes les entités capables de leur fournir des fonds ou des informations précieuses comme des cibles potentielles. Ainsi, les entreprises, les institutions gouvernementales et les organisations de divers secteurs sont confrontées au risque d'hameçonnage.

Les attaques d'hameçonnage ne connaissent pas de frontières et touchent aussi bien les utilisateurs individuels que les organisations de toutes tailles.

4. L'étape d'attaque de phishing

Nous devons étudier exactement les processus et les étapes des attaques de phishing afin de trouver des solutions complètes contre elles. En général, une attaque de phishing se compose de 4 étapes. Une fois qu'un attaquant a créé un faux site Web qui ressemble étroitement au site Web de confiance d'un utilisateur, l'attaquant prend les mesures suivantes :

- **Distribution des liens malveillants :** Dans cette étape, l'attaquant tente d'envoyer l'un des liens malveillants à l'utilisateur par e-mail ou par message instantané. Si l'attaquant connaît la victime, l'utilisateur peut être la cible d'un hameçonnage délibéré ou aléatoire.
- **Visiter un site de phishing:** Dans cette étape, la victime cliquera sur un lien malveillant reçu via un canal de communication. Ainsi, les victimes seront redirigées vers des sites de phishing qui ressemblent à de vrais sites pour la sensibilisation des utilisateurs.
- **Divulgarion d'informations sensibles:** Dans cette étape, étant donné que le faux site Web est complètement similaire au vrai site Web, les utilisateurs seront incités à divulguer leurs informations personnelles et financières comme ils le feraient sur le vrai site Web.
- **Transfert des informations divulguées à l'attaquant :** Une fois que l'utilisateur a divulgué les informations sur le faux site Web, les informations seront envoyées à l'attaquant et l'identité de l'utilisateur sera détournée par l'attaquant.

Les quatre étapes ci-dessus doivent être complétées pour que l'attaque de phishing réussisse. Techniquement, le blocage d'une étape peut entraîner l'échec de toute l'attaque de phishing. Pour cette raison, plusieurs méthodes préventives ont été proposées pour chacune des étapes ci-dessus, dont certaines seront discutées plus loin [7].

5. Les types d'attaques d'hameçonnage

Il existe plusieurs types d'attaques d'hameçonnage. Voici les plus courants :

➤ **Hameçonnage par e-mail**

Cette attaque se produit lorsque l'attaquant envoie un e-mail frauduleux qui ressemble à un e-mail légitime provenant d'une entreprise ou d'une organisation connue. Le but est de

tromper la victime pour qu'elle divulgue des informations confidentielles telles que des noms d'utilisateur et des mots de passe.

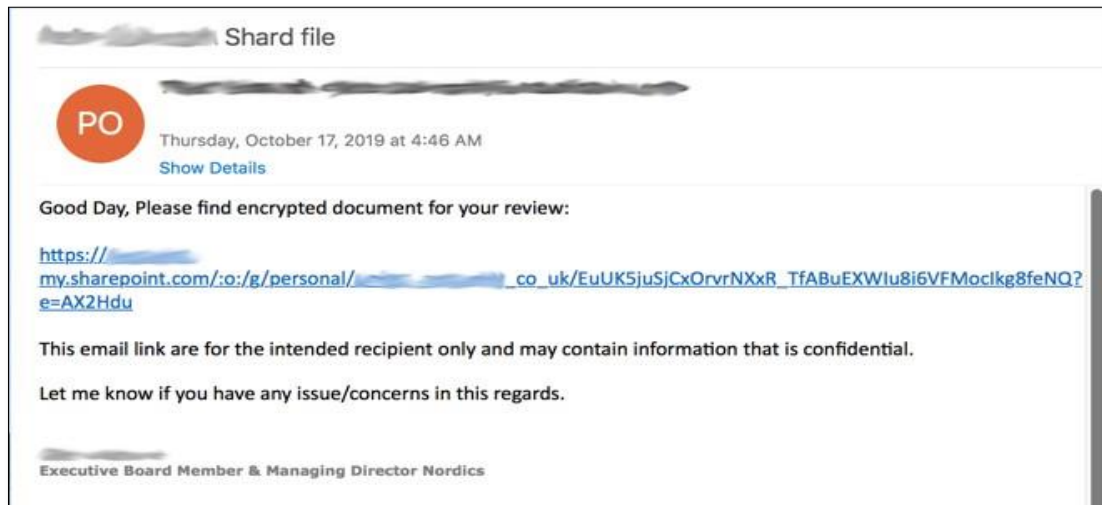


Figure 4: Anatomie d'un email de phishing [8]

➤ Hameçonnage de spear phishing

Cette attaque est similaire aux e-mails de phishing, mais plus ciblée. Les attaquants se concentrent sur des individus ou des organisations spécifiques et utilisent des informations spécifiques pour personnaliser les attaques. Les victimes du harponnage sont souvent des personnes qui disposent d'informations ou d'un accès précieux.

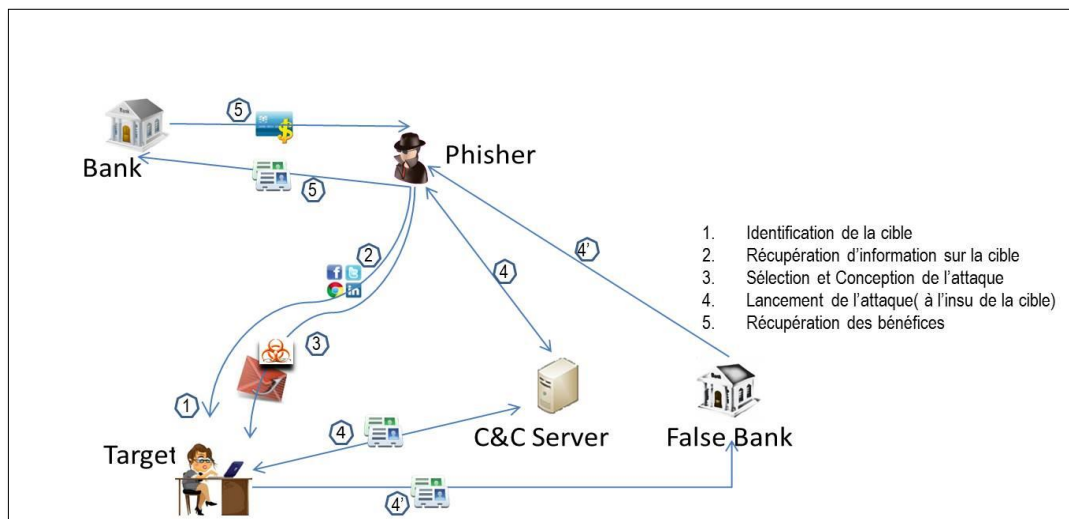


Figure 5: Le processus d'attaque par Spear Phishing [9]

➤ Hameçonnage par SMS

Dans ce type d'attaque, l'attaquant envoie un SMS frauduleux à la victime. Les messages texte peuvent contenir des liens malveillants qui, lorsqu'ils sont cliqués, redirigent les victimes vers des sites de phishing.

➤ **Hameçonnage par téléphone**

Dans ce type d'attaque, l'attaquant appelle la victime et se fait passer pour un employé d'une entreprise légitime. L'attaquant peut essayer de convaincre la victime de divulguer des informations confidentielles, telles que des numéros de carte de crédit ou des mots de passe.

➤ **Hameçonnage de smishing**

Cette attaque est similaire à l'hameçonnage par SMS, mais elle se produit sur des applications de messagerie, telles que WhatsApp ou Facebook Messenger.

➤ **Hameçonnage par ingénierie sociale**

Cette attaque consiste à manipuler la victime pour révéler des informations confidentielles. Les attaquants peuvent utiliser des techniques de manipulation psychologique pour faire croire aux victimes qu'elles doivent révéler des informations confidentielles pour des raisons légitimes.

6. La nécessité de la détection d'hameçonnage

Dans cette section, nous aborderons l'importance cruciale de la détection d'hameçonnage. Avec la sophistication croissante des attaques d'hameçonnage, il devient primordial de mettre en place des mécanismes de détection efficaces pour identifier et contrer ces tentatives malveillantes.

6.1 L'ampleur croissante des attaques d'hameçonnage

La détection d'hameçonnage est essentielle pour protéger les utilisateurs en ligne contre les tentatives de fraude et d'escroquerie. L'hameçonnage est une technique courante utilisée par les cybercriminels pour tromper les utilisateurs et les inciter à divulguer des informations personnelles, telles que des informations de compte bancaire ou de carte de crédit, des noms d'utilisateur et des mots de passe.

Les attaques d'hameçonnage ont connu une augmentation significative en termes de fréquence et de sophistication. Les cybercriminels ont développé des techniques plus avancées et adaptatives pour tromper les utilisateurs et contourner les mesures de sécurité traditionnelles [10].

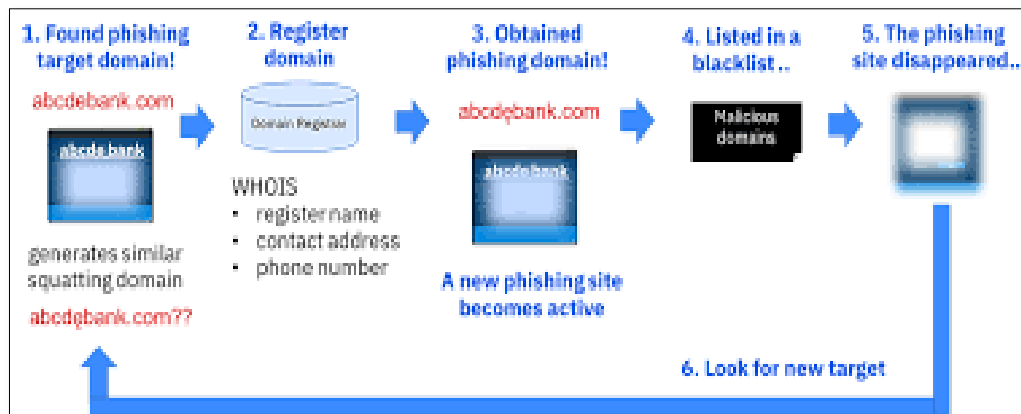


Figure 6 : Stratégies de phishing [11]

6.2 La protection des données et de la vie privée

L'hameçonnage vise souvent à voler des informations sensibles, telles que des identifiants de connexion, des données financières ou des informations personnelles. Lorsque ces données tombent entre de mauvaises mains, cela peut avoir des conséquences graves, allant de l'usurpation d'identité au vol d'argent. La détection d'hameçonnage joue un rôle crucial dans la préservation de la confidentialité des données et la protection de la vie privée des utilisateurs en identifiant les tentatives de phishing et en les bloquant avant qu'elles ne causent des dommages.

L'objectif de la détection d'hameçonnage est de protéger les utilisateurs en ligne contre les attaques d'hameçonnage en identifiant et en bloquant les e-mails, les messages instantanés, les sites web et autres formes de communication suspectes ou non sollicitées. La détection d'hameçonnage permet de détecter les tentatives de phishing et de les bloquer avant qu'elles n'aient la possibilité de causer des dommages, tels que la divulgation d'informations personnelles et financières ou la contamination par des virus informatiques.

En fin de compte, l'objectif de la détection d'hameçonnage est de garantir la sécurité en ligne pour les utilisateurs, les entreprises et les organisations [12].

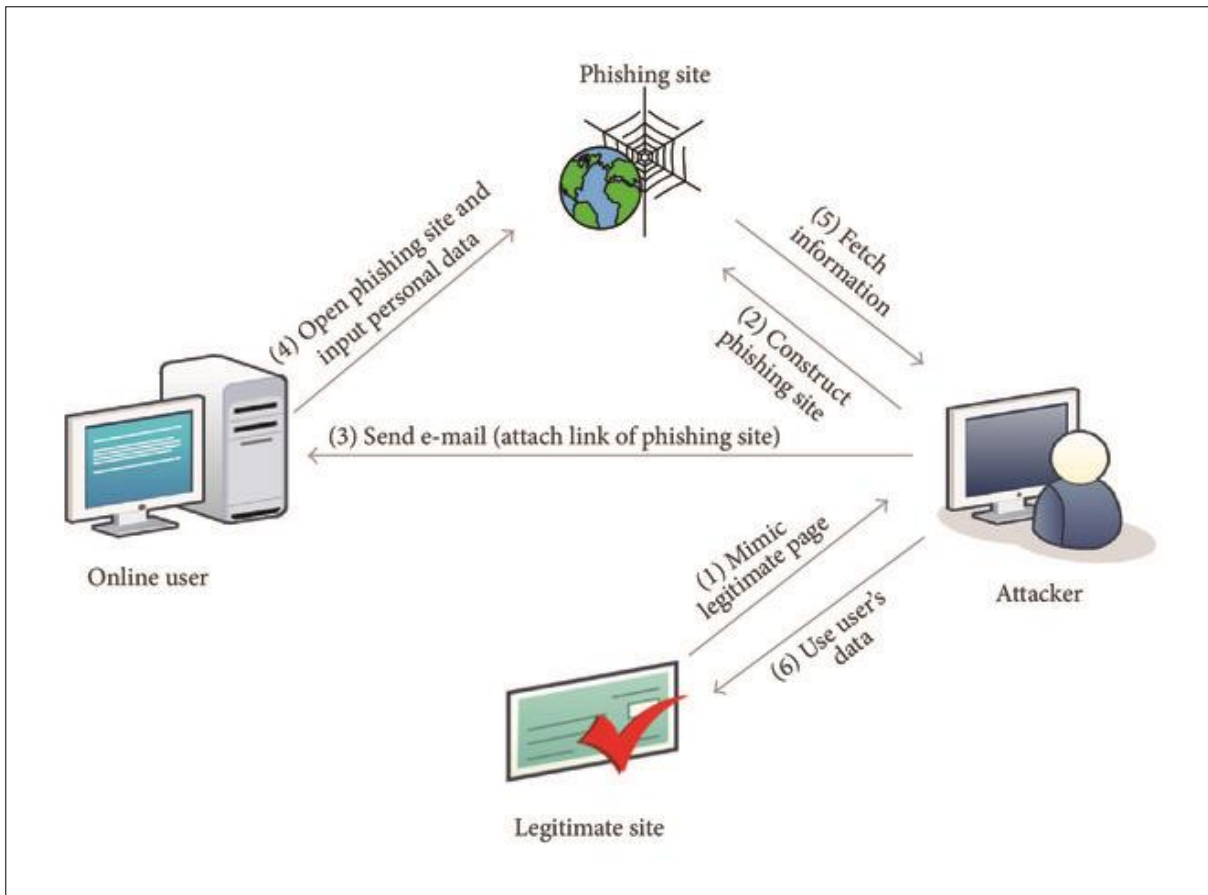


Figure 7: Technique de détection de phishing basée sur les fonctionnalités [13]

En résumé, la détection d'hameçonnage est essentielle pour protéger les utilisateurs contre les cyberattaques et les tentatives de fraude en ligne. En étant vigilants et en utilisant les bonnes pratiques de sécurité, les utilisateurs peuvent contribuer à réduire les risques d'être victimes d'une attaque d'hameçonnage.

7. Attaques d'hameçonnage ciblant les grandes sociétés

Les attaques d'hameçonnage ciblant les grandes entreprises représentent un défi majeur en matière de cybersécurité. Dans cette section, nous explorerons les risques et les impacts associés à ces attaques, en plus de fournir des exemples détaillés d'attaques notables.

7.1 Dangers des attaques d'hameçonnage sur les entreprises : Risques et impacts

Les attaques d'hameçonnage représentent un danger significatif pour les entreprises, entraînant divers risques et impacts. Parmi ceux-ci :

- **Vol d'informations sensibles** : Les attaquants visent à obtenir des informations confidentielles telles que des identifiants de connexion, des données bancaires ou des secrets commerciaux. Le vol de telles informations peut avoir des conséquences financières graves et causer une perte de confiance des clients et partenaires.
- **Fraude financière** : Les attaques d'hameçonnage permettent aux cybercriminels d'accéder aux systèmes de paiement de l'entreprise, leur permettant d'effectuer des transactions frauduleuses, de détourner des fonds ou de compromettre les processus financiers internes. Cela peut entraîner des pertes financières importantes et des problèmes de conformité.
- **Domages à la réputation** : Les attaques d'hameçonnage peuvent nuire à la réputation d'une entreprise. La perte de confiance des clients et les impacts médiatiques négatifs peuvent avoir des conséquences durables sur l'image et la crédibilité de l'entreprise.

Il est essentiel que les entreprises prennent des mesures proactives pour se protéger contre les attaques d'hameçonnage. Cela implique de sensibiliser les employés aux techniques de phishing, de mettre en place des mesures de sécurité solides telles que des filtres anti-spam et des logiciels de détection d'hameçonnage.

7.2 Exemples d'attaques d'hameçonnage contre les entreprises

Exemples notables d'attaques d'hameçonnage contre les grandes entreprises :

- **Attaque contre Sony Pictures Entertainment (2014)** : Une attaque sophistiquée d'hameçonnage perpétrée par un groupe lié à la Corée du Nord a visé Sony Pictures Entertainment. Les employés ont été ciblés par des e-mails de phishing incitant à ouvrir des pièces jointes malveillantes, entraînant le vol de données sensibles et des répercussions importantes sur la réputation de l'entreprise.
- **Attaque contre JPMorgan Chase (2014)** : Des cybercriminels russes ont mené une attaque massive d'hameçonnage contre JPMorgan Chase et d'autres institutions financières. Les employés ont été incités à divulguer leurs informations d'identification, ce qui a permis aux attaquants d'accéder à des données sensibles et de compromettre la vie privée de millions de clients.
- **Attaque contre Target (2013)** : Des pirates informatiques ont ciblé Target, une importante chaîne de magasins américaine, en exploitant une attaque d'hameçonnage ciblée. Ils ont compromis les informations de connexion d'un

fournisseur tiers pour accéder au système de paiement de l'entreprise, entraînant le vol massif de données de clients et d'importants préjudices financiers.

Ces exemples illustrent la gravité des attaques d'hameçonnage auxquelles les grandes entreprises sont confrontées. Il est impératif pour ces entreprises de mettre en place des mesures de sécurité solides [14].

8. L'intelligence artificielle dans le cybersécurité

8.1 Définition

L'intelligence artificielle (IA) dans la cybersécurité fait référence à l'utilisation de techniques et d'algorithmes d'IA pour renforcer la sécurité des systèmes informatiques et protéger les données contre les attaques malveillantes. Elle vise à automatiser des tâches complexes liées à la détection, à la prévention et à la réponse aux cyberattaques, en exploitant les capacités de l'IA telles que l'apprentissage automatique, le traitement du langage naturel, la vision par ordinateur et les réseaux neuronaux [15].

8.2 L'impact de l'intelligence artificielle sur la cybersécurité

Dans ce titre, on souligne que l'intelligence artificielle marque une nouvelle ère de protection numérique en renforçant les capacités de détection, de prévention et de réponse aux cybermenaces. On peut développer les points suivants :

- L'intelligence artificielle permet une surveillance continue et proactive des systèmes informatiques, en analysant les comportements et les schémas d'activité pour détecter les signes précurseurs de cyberattaques.
- Grâce à ses algorithmes d'apprentissage automatique, l'intelligence artificielle peut identifier les menaces connues et nouvelles, en apprenant des modèles de logiciels malveillants et en adaptant ses stratégies de défense en conséquence.
- L'utilisation de l'intelligence artificielle dans l'analyse des vulnérabilités aide à identifier les faiblesses des systèmes et des applications, permettant ainsi de les corriger avant qu'elles ne soient exploitées par des attaquants.

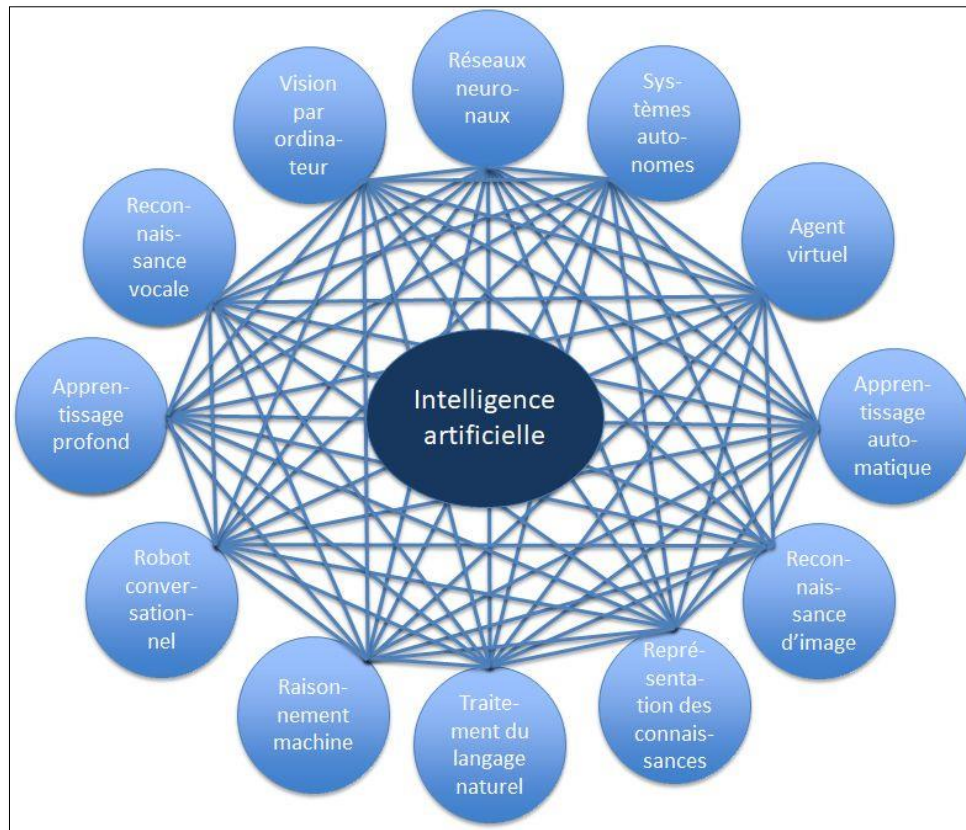


Figure 8: Les 12 technologies de l'intelligence artificielle [16]

8.3 Le Machine Learning et le Deep Learning dans la cybersécurité

8.3.1 Le Machine Learning

La machine Learning (apprentissage automatique) dans la cybersécurité fait référence à l'utilisation de techniques et d'algorithmes qui permettent aux systèmes informatiques d'apprendre et de s'améliorer à partir des données, afin de détecter et de prévenir les menaces et les attaques malveillantes. Il s'agit d'une branche de l'intelligence artificielle qui vise à permettre aux ordinateurs d'apprendre à partir de l'expérience et à prendre des décisions.

Le machine learning dans la cybersécurité permet de :

- ❖ **Détection les attaques** : En apprenant à partir des données normales, les modèles de machine learning peuvent identifier les comportements anormaux et les activités suspectes, qui pourraient indiquer des tentatives d'intrusion ou des attaques.
- ❖ **Détection des logiciels malveillants** : Les modèles de machine learning peuvent être formés pour reconnaître les signatures et les caractéristiques des logiciels malveillants, facilitant ainsi la détection précoce et précise des menaces.

- ❖ **Prévention des attaques** : En analysant les données en temps réel, les systèmes basés sur la machine learning peuvent détecter les attaques en cours et prendre des mesures pour les contrer, telles que le blocage des adresses IP suspectes ou la désactivation des connexions.

Le ML dans la cybersécurité permet aux systèmes de sécurité de tirer parti des données et des modèles pour améliorer la détection, la prévention et la réponse aux menaces et aux attaques malveillantes [17].

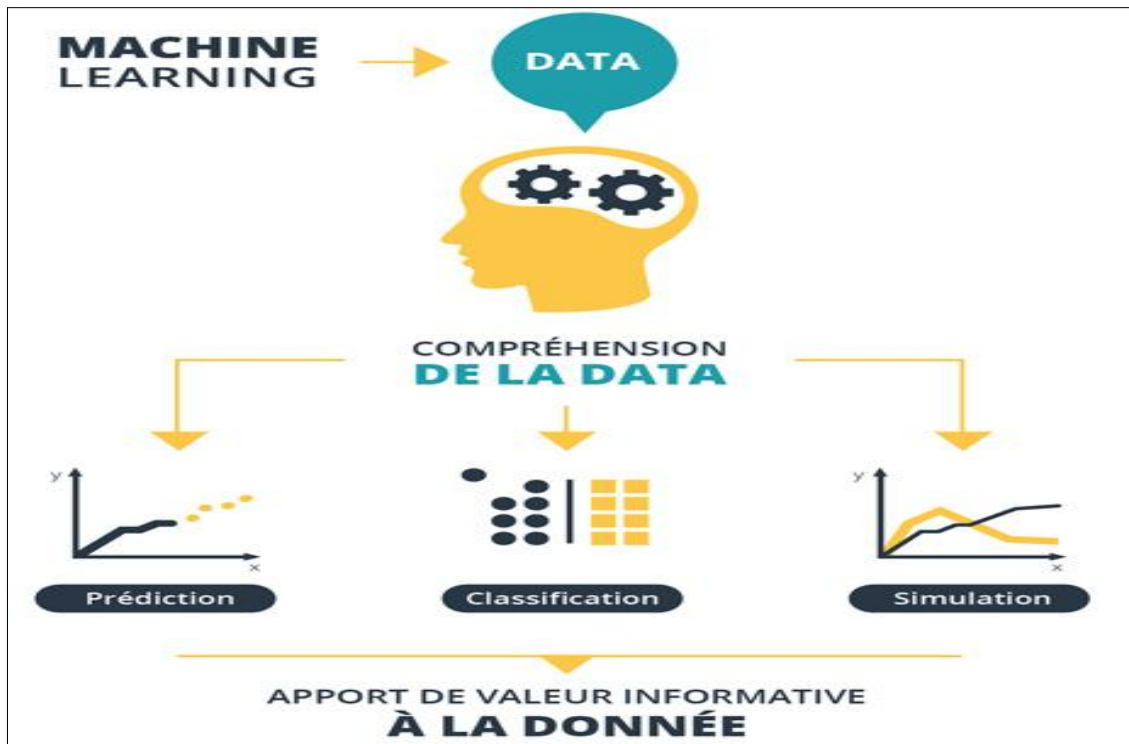


Figure 9 : Le processus de fonctionnement du Machine Learning

8.3.2 Le Deep Learning

Le DL, ou apprentissage profond, dans la cybersécurité fait référence à l'utilisation de réseaux de neurones artificiels profonds pour résoudre des problèmes de sécurité informatique. Il s'agit d'une sous-branche du machine learning qui utilise des architectures de réseaux de neurones profonds pour apprendre à partir de grandes quantités de données et extraire des informations pertinentes pour la détection au cyberattaques.

Le deep learning dans la cybersécurité offre plusieurs avantages :

- ❖ **Évolution constante** : Les réseaux de neurones profonds peuvent s'adapter et évoluer en fonction des nouvelles menaces et des stratégies d'attaque en constante évolution. Ils sont capables d'apprendre de nouvelles caractéristiques et de nouvelles techniques d'attaque, ce qui leur permet de rester efficaces face aux cybermenaces émergentes.
- ❖ **Traitement efficace des données non structurées** : Le deep learning excelle dans le traitement des données non structurées, telles que les journaux d'événements, les images, les vidéos, les enregistrements audio, etc. Cela permet de détecter des menaces potentielles provenant de sources diverses et de prendre des décisions basées sur des informations variées.
- ❖ **Réduction des faux positifs** : Les réseaux de neurones profonds peuvent apprendre à différencier les comportements légitimes des activités malveillantes, ce qui permet de réduire les faux positifs [18].

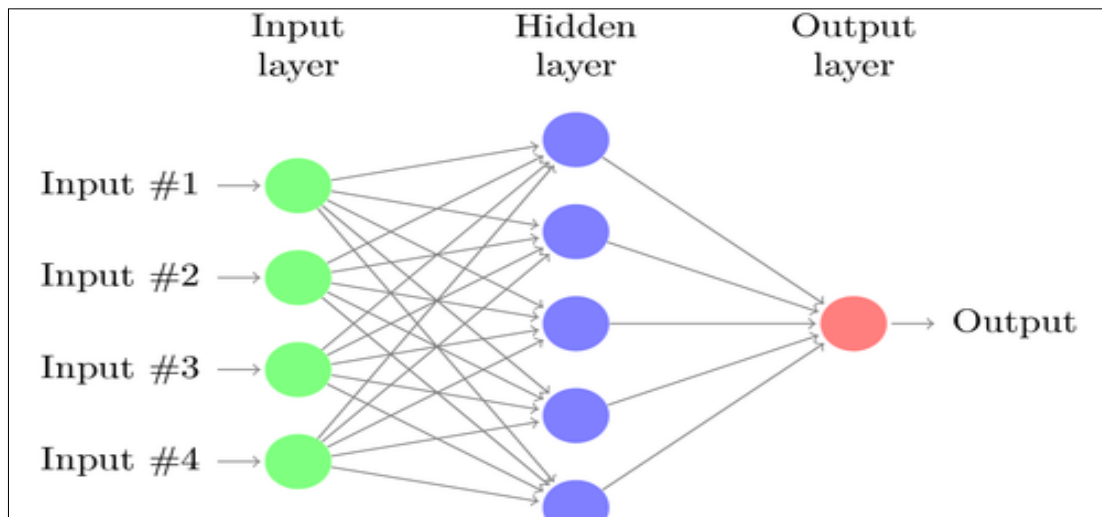


Figure 10: Le processus de fonctionnement de Deep Learning

Conclusion

En conclusion, il est clair que la détection d'hameçonnage est un enjeu important pour la sécurité informatique, et que des approches sophistiquées et innovantes sont nécessaires pour prévenir les attaques d'hameçonnage. Les systèmes de détection d'hameçonnage doivent être en mesure d'analyser en temps réel un grand volume de données, tout en minimisant les faux positifs et les faux négatifs. Les futures recherches devraient se concentrer sur le développement de techniques de détection plus avancées et plus efficaces, en utilisant des méthodes telles que le Machine Learning, le traitement du langage naturel, ou l'analyse du comportement utilisateur.



Chapitre 2 : Les Techniques De Détection D'hameçonnage



Introduction

Les systèmes de détection d'hameçonnage sont des logiciels de sécurité qui utilisent des algorithmes pour détecter et prévenir les attaques d'hameçonnage. Ils identifient les signes de l'hameçonnage tels que des URL suspectes, des fautes d'orthographe, des images trompeuses et des messages incitant à divulguer des informations personnelles. Ces systèmes visent à empêcher les utilisateurs d'interagir avec des sites web ou des e-mails malveillants.

Leur objectif est de protéger les utilisateurs contre les tentatives de phishing

1. Les solutions existantes de système de détection d'hameçonnage

1.1 Filtres anti-spam

Un filtre anti-spam est un outil utilisé pour lutter contre le spam, ou "spam", dans les e-mails. Les filtres anti-spam améliorent l'expérience utilisateur et réduisent le risque de fraude et d'attaques de phishing en utilisant des algorithmes pour détecter et bloquer les messages indésirables. Dans ce rapport, nous allons explorer les différents types de filtres anti-spam, leur fonctionnement et leur efficacité.

Ce service est une plateforme externalisée qui assure le traitement et la gestion de tous vos e-mails, en les filtrant, les classant et en les acheminant vers vos serveurs appropriés, tout en détectant et rejetant automatiquement les e-mails indésirables ou contenant des virus [19].

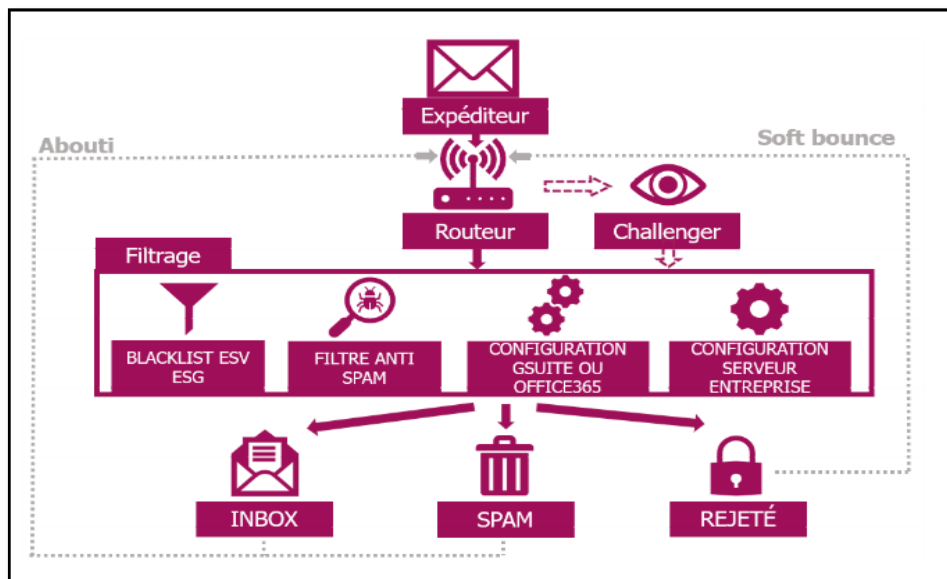


Figure 11: Représentation des dispositifs de filtrage anti spam des messages email [20]

1.1.1 Types de filtres anti-spam

Il existe une multitude de méthodes de filtrage anti-spam qui peuvent être appliquées pour empêcher la réception de messages non sollicités dans les boîtes de réception électroniques. Les types les plus courants de filtres anti-spam sont les suivants :

- **Filtres basés sur les règles** : Les filtres basés sur les règles fonctionnent en utilisant des règles prédéfinies pour déterminer si un message est du spam ou non. Ces règles peuvent inclure des critères tels que le contenu du message, les en-têtes du message et l'adresse IP de l'expéditeur.
- **Filtres basés sur la liste noire** : Les filtres basés sur la liste noire fonctionnent en comparant l'adresse IP de l'expéditeur avec une liste de serveurs de messagerie connus pour envoyer du spam. Si l'adresse IP correspond à une entrée de la liste noire, le message est bloqué.
- **Filtres basés sur l'apprentissage automatique** : Les filtres basés sur l'apprentissage automatique fonctionnent en utilisant des algorithmes d'apprentissage pour analyser les caractéristiques des messages pour déterminer s'ils sont du spam ou non.

1.1.2 Les avantages

Les filtres anti-spam offrent plusieurs avantages pour les utilisateurs de courrier électronique, notamment :

- **Réduction des courriers indésirables** : Les filtres anti-spam permettent de filtrer les courriers électroniques indésirables, réduisant ainsi le nombre de messages non sollicités dans la boîte de réception.
- **Amélioration de la sécurité** : Les filtres anti-spam peuvent détecter les messages frauduleux, tels que les tentatives de phishing, les virus et les logiciels malveillants.
- **Protection de la réputation de l'entreprise** : En bloquant les messages indésirables, les filtres anti-spam peuvent protéger la réputation de l'entreprise en empêchant les messages de spam d'être envoyés aux clients ou aux partenaires commerciaux.

1.1.3 Les inconvénients

Bien que les filtres anti-spam offrent de nombreux avantages, ils peuvent également présenter certains inconvénients, notamment :

- **Risque de faux positifs** : Les filtres anti-spam peuvent parfois considérer des messages légitimes comme du spam, ce qui peut entraîner la suppression de messages importants.
- **Complexité de la configuration** : Les filtres anti-spam peuvent nécessiter une configuration complexe pour fonctionner correctement, ce qui peut nécessiter des compétences techniques.
- **Dépendance à l'égard des mises à jour** : Les filtres anti-spam doivent être régulièrement mis à jour pour détecter les nouveaux types de spam, ce qui peut nécessiter des ressources supplémentaires pour l'administration du système.
- **Réduction de la confidentialité** : Les filtres anti-spam peuvent analyser les messages entrants, ce qui peut réduire la confidentialité des utilisateurs.

1.1.4 Le fonctionnement de système

Le fonctionnement d'un système de filtrage anti-spam peut varier en fonction de la méthode utilisée, mais voici une explication générale :

Un système de filtrage anti-spam analyse chaque e-mail entrant en utilisant des règles pré-établies, des algorithmes de détection de spam, et/ou des techniques de machine Learning pour déterminer s'il est probablement un spam ou non. Les règles peuvent inclure des critères tels que l'adresse de l'expéditeur, le contenu de l'e-mail, la présence de liens suspects, la taille du fichier attaché, etc.

Si l'e-mail est considéré comme spam, il peut être marqué comme tel ou placé dans un dossier spam. Si l'e-mail est considéré comme légitime, il peut être remis à la boîte de réception de l'utilisateur.

Certains systèmes de filtrage anti-spam permettent aux utilisateurs de personnaliser les règles et de signaler les e-mails indésirables ou les faux positifs. Le système peut également apprendre des actions de l'utilisateur et ajuster ses règles pour mieux filtrer les e-mails à l'avenir [21].



Figure 12: Le fonctionnement des systèmes de filtrage des e-mails [22]

En résumé, le système de filtrage anti-spam utilise des règles, des algorithmes et des techniques de machine Learning pour analyser les e-mails entrants et identifier les spams. Il permet aux utilisateurs de recevoir uniquement les e-mails légitimes dans leur boîte de réception et de réduire le temps et les efforts nécessaires pour trier et supprimer manuellement les e-mails indésirables.

1.2 PhishBlock

PhishBlock est un projet open source de détection d'URL de phishing. Il utilise des techniques d'apprentissage automatique pour analyser les URL et détecter les signes d'hameçonnage. Les données sont collectées à partir de sources publiques et les algorithmes apprennent à reconnaître les schémas et les caractéristiques communs des URL de phishing en analysant un ensemble de données de formation.

Le projet est hébergé sur GitHub et est disponible gratuitement pour tous les utilisateurs. Les développeurs peuvent également contribuer au projet en proposant des améliorations ou des corrections de bogues.

PhishBlock est un exemple de la manière dont l'apprentissage automatique peut être utilisé pour détecter les attaques d'hameçonnage de manière efficace et rapide. En utilisant des algorithmes d'apprentissage automatique, les systèmes peuvent apprendre à reconnaître les schémas et les caractéristiques communs des URL de phishing [23].

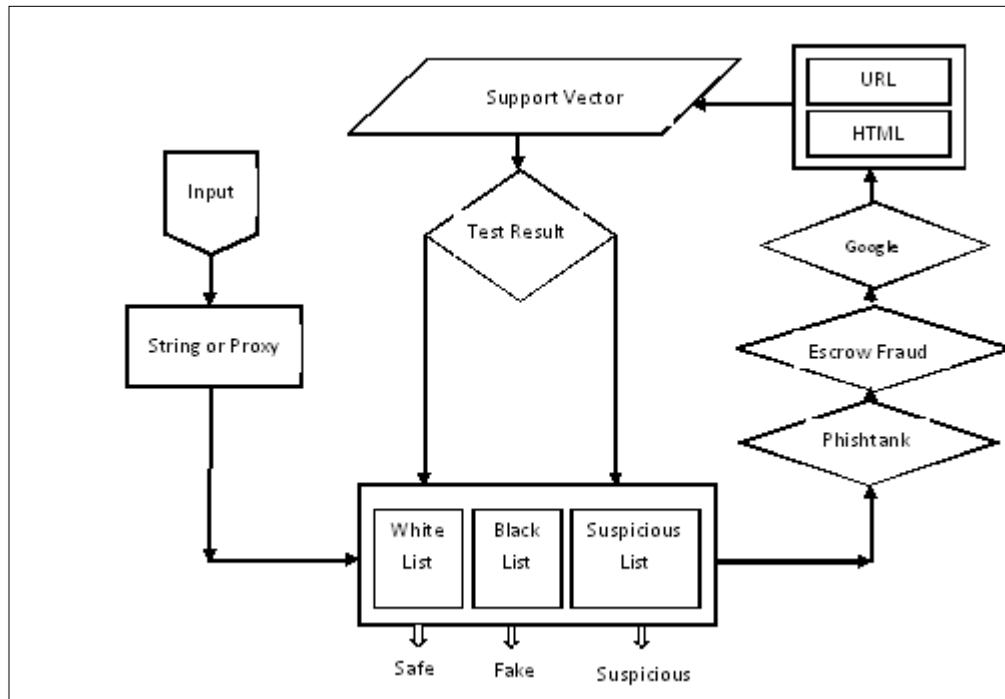


Figure 13: Le fonctionnement de PhishBlock [24]

1.2.1. Les étapes principales de son fonctionnement

- **Collecte des données** : PhishBlock collecte des données à partir de sources publiques, telles que des sites web de signalement de phishing et des bases de données de malware, afin de créer un ensemble de données de formation.
- **Analyse des données** : Les données collectées sont analysées pour identifier les caractéristiques communes des URL de phishing, telles que des mots clés, des fautes d'orthographe, des noms de domaine suspects, etc.
- **Entraînement de l'algorithme** : Un algorithme d'apprentissage automatique est entraîné à partir de l'ensemble de données de formation pour reconnaître les schémas et les caractéristiques communs des URL de phishing.
- **Détection des URL de phishing** : Lorsqu'un utilisateur visite un site web ou clique sur un lien, l'URL est analysée par l'algorithme de PhishBlock pour

déterminer si elle présente des signes d'hameçonnage. Si l'URL est considérée comme suspecte, l'utilisateur est averti et peut prendre les mesures nécessaires pour éviter une attaque de phishing.

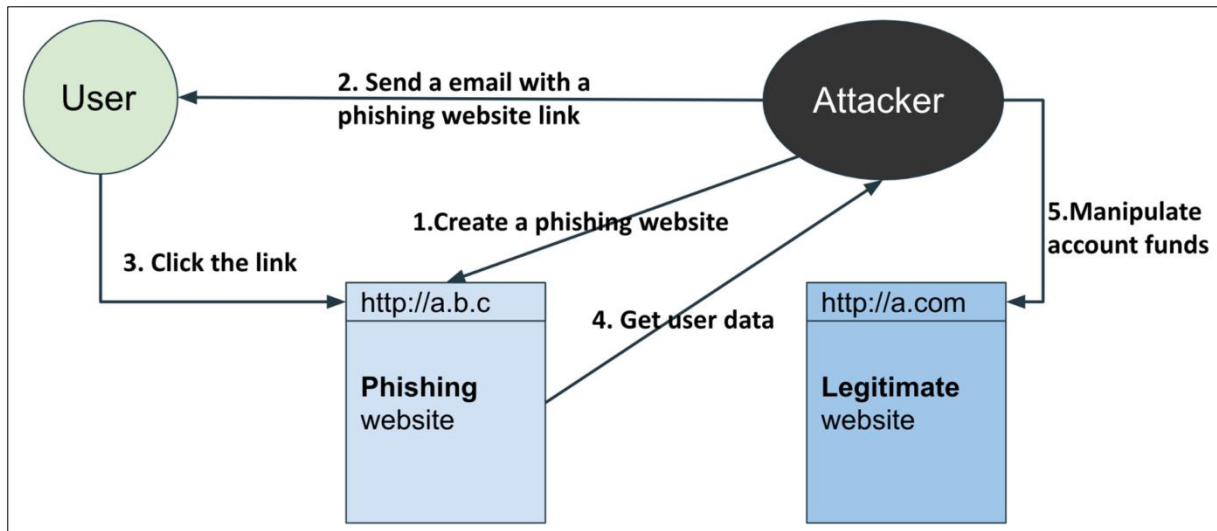


Figure 14: Le cycle du phishing de PhishBlock [25]

PhishBlock est un exemple de système de détection d'hameçonnage basé sur l'apprentissage automatique qui peut être utilisé pour protéger les utilisateurs contre les tentatives d'hameçonnage en ligne. En utilisant des algorithmes d'apprentissage automatique.

1.2.2. Les avantages

PhishBlock offre plusieurs avantages par rapport aux méthodes traditionnelles de détection d'hameçonnage basées sur des règles prédéfinies :

- **Précision accrue** : L'utilisation de techniques d'apprentissage automatique permet à PhishBlock de reconnaître les schémas et les caractéristiques communs des URL de phishing avec une grande précision, ce qui permet de réduire le nombre de faux positifs et de faux négatifs.
- **Adaptabilité** : Contrairement aux méthodes traditionnelles de détection basées sur des règles prédéfinies, PhishBlock peut s'adapter à de nouveaux schémas d'hameçonnage et à des tactiques de phishing émergentes, ce qui permet une détection plus efficace des tentatives d'hameçonnage.
- **Rapidité** : Grâce à son algorithme d'apprentissage automatique, PhishBlock peut détecter les URL de phishing en temps réel, ce qui permet de réduire le temps de réponse aux attaques d'hameçonnage.

1.2.3. Les inconvénients

- **Coûts** : PhishBlock est une solution commerciale qui nécessite un abonnement payant, ce qui peut représenter un coût supplémentaire pour les organisations.
- **Besoin d'une connexion Internet** : Pour que PhishBlock fonctionne correctement, une connexion Internet est nécessaire, ce qui peut poser des problèmes dans les environnements où la connectivité Internet est limitée ou instable.
- **Risque de faux positifs** : Comme pour toute solution de détection d'hameçonnage, il existe un risque de faux positifs. Cela peut entraîner des désagréments pour les utilisateurs légitimes qui sont bloqués d'accéder à des sites web légitimes.

2. Modèle de détection de phishing proposé

2.1 Contexte et problématique

Les filtres antispam et PhishBlock, bien qu'ils soient largement utilisés dans le domaine de la sécurité informatique, présentent certains inconvénients. Tout d'abord, ils peuvent souvent générer un nombre élevé de faux positifs, c'est-à-dire des e-mails légitimes identifiés à tort comme du spam ou des tentatives d'hameçonnage. Cela peut entraîner une perte de temps et une frustration pour les utilisateurs qui doivent vérifier manuellement leurs messages pour récupérer les courriers électroniques importants.

L'intelligence artificielle basée sur l'apprentissage automatique peut être employée pour détecter les tentatives de phishing en utilisant des algorithmes capables d'apprendre à partir d'un vaste ensemble de données. Dans cette section, nous présentons un modèle de détection de phishing basé sur l'apprentissage automatique avec l'algorithme SVM. Nous expliquons comment les données ont été préparées, comment le modèle SVM a été entraîné et comment les performances ont été évaluées. Nous discutons également des avantages et des limites de ce modèle, ainsi que des perspectives pour améliorer sa précision et sa fiabilité.

2.2 Architecture de système proposée

Le système de détection d'hameçonnage est conçu pour identifier les URL malveillantes ou suspectes, en utilisant un modèle de machine learning basé sur SVM (Support Vector Machine). Voici une architecture générale pour ce système :

- ❖ **Interface utilisateur (UI)** : C'est la partie visible de l'application où les utilisateurs peuvent entrer une URL à vérifier. L'interface utilisateur peut être

développée en HTML, CSS et JavaScript, et elle permet aux utilisateurs de soumettre une URL pour l'analyse.

- ❖ **Serveur Flask** : Le serveur Flask est responsable de la gestion des requêtes HTTP, du routage des URL et du traitement des données. Il reçoit les requêtes des utilisateurs via l'interface utilisateur, traite les données et renvoie les résultats correspondants. Le serveur Flask utilise des routes pour définir les points d'entrée de l'application et les fonctions associées pour le traitement des requêtes.
- ❖ **Modèle SVM** : Le modèle SVM est utilisé pour prédire si l'URL soumise est malveillante ou non. Le modèle SVM est entraîné sur un ensemble de données contenant des exemples d'URLs malveillantes et non malveillantes. Une fois que les caractéristiques de l'URL soumise ont été extraites, elles sont fournies en entrée au modèle SVM qui retourne une prédiction (comme, 1 pour une URL sûre et -1 pour une URL malveillante).
- ❖ **Résultats** : Les résultats de la prédiction du modèle SVM sont renvoyés au serveur Flask, qui les affiche à l'utilisateur sur l'interface utilisateur. Les résultats peuvent indiquer si l'URL est sûre ou malveillante, et éventuellement fournir une probabilité ou un score de confiance associé à la prédiction.

2.3 L'apprentissage

L'apprentissage est une étape très importante du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié itérativement jusqu'à l'obtention du comportement désiré, et ce par l'ajustement des poids (connexion ou synapse) des neurones à une source d'information bien définie (Hebb 1949; Grossberg 1982; Rumelhart et al. 1986).

L'apprentissage implique également l'identification de motifs parmi les données utilisées pour entraîner un réseau, mais son objectif principal est la résolution de problèmes par la prédiction.

Il existe différents types de règles d'apprentissage qui peuvent être regroupées en deux catégories : l'apprentissage supervisé et l'apprentissage non supervisé [26].

1.3.1. La classification

Une diversité de méthodes traditionnelles ont été développées, pouvant être regroupées en deux principales catégories : les méthodes de classification supervisée, qui nécessitent un ensemble de données étiquetées pour l'apprentissage, et les méthodes de classification non

supervisée, qui permettent d'identifier des motifs et des structures sans l'utilisation d'étiquettes préexistantes.

❖ **Méthodes supervisées**

La classification supervisée vise à établir des règles pour classer des objets en fonction de variables qualitatives ou quantitatives qui les caractérisent. Pour ce faire, nous utilisons initialement un échantillon d'apprentissage dont la classification est déjà connue. Cet échantillon est utilisé pour apprendre les règles de classification.

❖ **Méthodes non supervisées**

La méthode en question fonctionne de manière opposée, c'est-à-dire qu'elle ne nécessite pas d'apprentissage ni de tâche préalable d'étiquetage manuel. Elle consiste à représenter un nuage de points d'un espace quelconque sous forme d'un ensemble de groupes appelés "clusters". Cette méthode est généralement liée au domaine de l'analyse des données, telle que l'Analyse en Composantes Principales (ACP). Un "cluster" est une collection d'objets qui sont similaires entre eux et qui diffèrent des objets appartenant à d'autres groupes.

1.3.2. La détection

La détection fait référence à la capacité d'un modèle d'apprentissage à identifier si une URL donnée est un cas de phishing ou non.

La détection d'hameçonnage implique généralement les étapes suivantes :

- ❖ **Collecte des données** : Un ensemble de données d'exemples d'URL étiquetées comme étant soit des URL de phishing, soit des URL légitimes est collecté. Ces données serviront à entraîner et à évaluer le modèle de détection.
- ❖ **Prétraitement des données** : Les données collectées peuvent nécessiter un prétraitement pour les rendre exploitables. Cela peut inclure le nettoyage des données, la normalisation des caractéristiques.
- ❖ **Entraînement du modèle de détection** : Les exemples d'URL prétraités et leurs étiquettes sont utilisés pour entraîner un modèle d'apprentissage automatique.

3. Solution proposée

L'objectif de ce projet est de proposer une nouvelle approche basée sur le modèle SVM (Support Vector Machine) pour la classification et la détection des attaques de phishing. En exploitant les capacités puissantes du SVM, nous visons à améliorer la précision et l'efficacité de la détection des sites web de phishing. En utilisant des caractéristiques spécifiques extraites des URL ou des contenus des pages web, notre modèle SVM sera en mesure de distinguer

avec précision les sites web légitimes des sites de phishing. Grâce à cette approche basée sur le SVM, nous cherchons à fournir une solution solide pour renforcer la sécurité en ligne et protéger les utilisateurs contre les tentatives d'hameçonnage.

Un autre avantage du modèle SVM est qu'il peut traiter directement les données brutes sans nécessiter une extraction de caractéristiques préalable. Contrairement à certains autres algorithmes classiques, le SVM peut apprendre automatiquement les caractéristiques discriminantes les plus importantes à partir des données d'entraînement. Cette capacité d'apprentissage intégrée permet au SVM de découvrir des schémas complexes et d'effectuer une classification précise des attaques de phishing.

4. Algorithme de classification

La classification est une méthode d'exploration de données permettant d'affecter des occurrences de données à l'une des rares catégories. Il existe de nombreux algorithmes de classification développés pour se surpasser. Ils fonctionnent tous selon des méthodes mathématiques telles que le SVM, la programmation linéaire, le KNN (K plus proches voisins), la régression logistique et les méthodes bayésiennes.

Ces méthodes analysent les données disponibles de plusieurs manières pour en faire la prévision. Voici ceux que nous allons utiliser dans notre projet : le SVM, le KNN, la régression logistique et le Naive Bayes.

4.1 L'Algorithme de modèle Naïve Bayes

Le modèle bayésien naïve est un modèle probabiliste simplifié basé sur l'hypothèse forte d'indépendance conditionnelle entre les attributs. Le classificateur bayésien naïf fonctionne en supposant que la probabilité d'un attribut n'influence pas la probabilité des autres attributs. Ainsi, étant donné un ensemble d'attributs, le classificateur bayésien naïf effectue 2^n hypothèses indépendantes.

Malgré cette simplification, les résultats du classificateur bayésien naïf sont souvent précis. Cela peut être dû à la nature des données ou à la capacité du modèle à capturer des tendances générales.

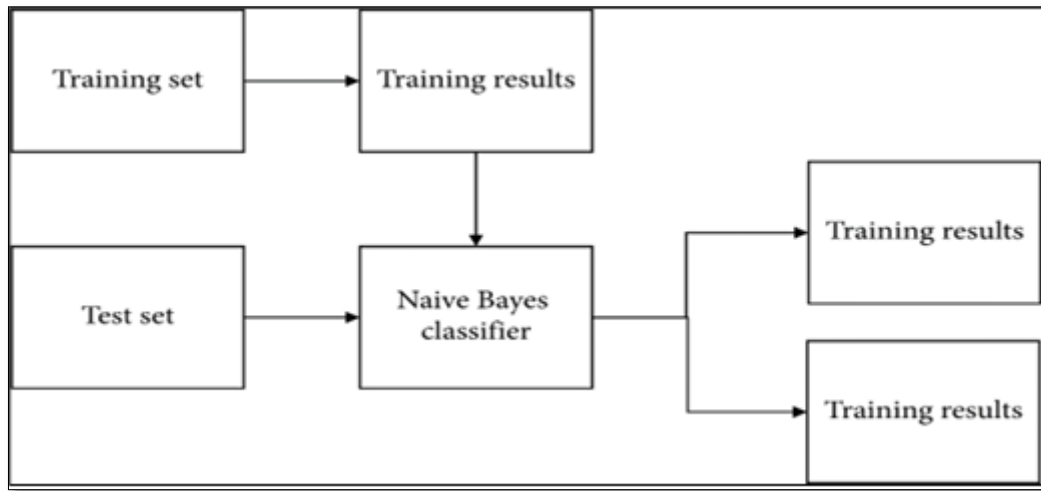


Figure 15: Modèle Naïve Bayes

Il est important de noter que le modèle bayésien naïf peut ne pas capturer toutes les subtilités et variations de l'hameçonnage, mais il peut constituer une première ligne de défense efficace en détectant de nombreux cas d'hameçonnage. Il est souvent utilisé en combinaison avec d'autres techniques de détection pour améliorer la précision globale du système.

4.2 Algorithme de modelé KNN

L'algorithme K-plus proches voisins (KNN) est un algorithme de classification et de régression utilisé en apprentissage automatique (machine learning). Il est basé sur l'idée que les instances similaires tendent à être proches les unes des autres dans l'espace des caractéristiques.

Dans le cas de la régression, KNN peut également estimer une valeur numérique en utilisant une moyenne ou une médiane des valeurs des K voisins les plus proches. L'algorithme KNN est simple à comprendre et à mettre en œuvre, mais il peut être sensible au bruit et au choix du paramètre K. Il est souvent utilisé en combinaison avec d'autres algorithmes et techniques de prétraitement des données pour améliorer ses performances.

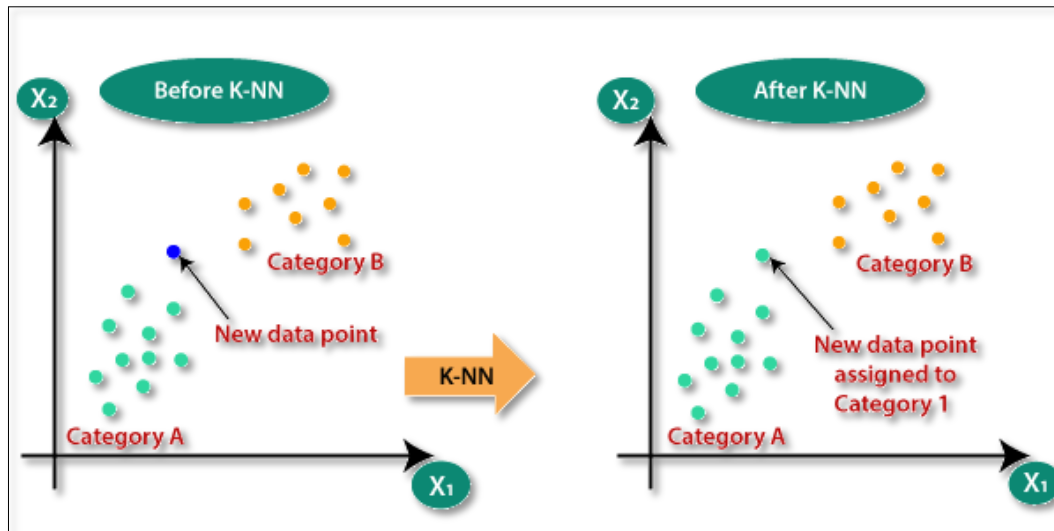


Figure 16: Fonctionnement d'un KNN

L'algorithme KNN (K-plus proches voisins) compare la similarité entre une nouvelle instance et les instances d'entraînement pour prédire sa classe. En recherchant les K voisins les plus proches dans l'espace des caractéristiques, il mesure la distance entre la nouvelle instance et ces voisins à l'aide d'une mesure comme la distance euclidienne. Ensuite, il attribue à la nouvelle instance la classe majoritaire parmi ces K voisins.

4.3 L'Algorithme de modèle Machine à Vecteur de Support (SVM)

SVM (Support Vector Machine) est un classificateur linéaire appartenant à une classe d'algorithmes qui utilisent une séparation linéaire des données pour trouver les frontières entre les classes.

Les machines à vecteurs de support sont un puissant algorithme d'apprentissage automatique utilisé pour la classification et la régression. L'objectif des SVM est de trouver la meilleure séparation entre les différentes classes d'un ensemble de données en construisant un hyperplan optimal dans un espace de caractéristiques de plus grande dimension. Cet hyperplan permet de maximiser la marge entre les échantillons des différentes classes, ce qui conduit à une meilleure capacité de généralisation [26].

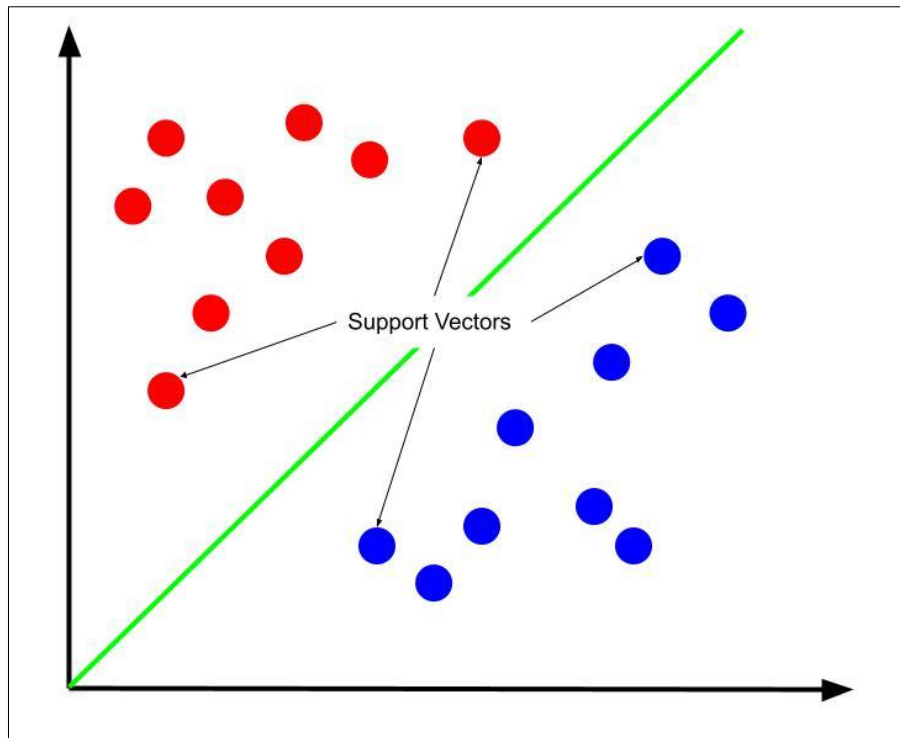


Figure 17 : Utilisation de l'algorithme SVM

L'objectif principal des SVM est de maximiser la marge, c'est-à-dire la distance entre l'hyperplan et les échantillons les plus proches de chaque classe. Cela permet de garantir une bonne capacité de généralisation aux nouvelles données. Les SVM peuvent également utiliser des noyaux pour effectuer des transformations non linéaires des données, leur permettant ainsi de gérer des problèmes de classification non linéaires.

Conclusion

Dans ce chapitre, nous avons décrit notre principe général et montré comment l'utilisation des techniques d'apprentissage peut grandement simplifier le processus de détection d'hameçonnage. L'utilisation de l'apprentissage automatique permet également d'inclure toutes les fonctionnalités aberrantes présentes dans un système de détection de phishing.

Chapitre 3 : Étude Expérimentale

Introduction

Dans ce chapitre, on donne les détails de réalisation notre système de détection d'hameçonnage, qui est la phase la plus essentielle pour profiter mieux qu'on puisse de l'analyse et les méthodes proposées dans ce projet mené en amont et de mettre réalisable, en donnant les résultats obtenus.

La réalisation de ce projet se fait par la préparation de base d'apprentissage, l'application nos modèles LR, SVM et KNN et Naïve Bayes de l'apprentissage profondeur puis test et évaluation.

1. Préparation de base d'apprentissage

Pour l'application nous avons utilisé l'approche d'analyse d'URLs pour dire qu'un site est sans risque sans avoir à examiner son contenu, cette approche cherche à éviter l'attaque de phishing .

L'URL tout seul peut contenir un lot important d'information qui nous donne la possibilité de juger d'une façon proactive son contenu.

Nous avons réalisé un système de classification qui peut analyser une base d'URLs étiquetés selon un nombre d'heuristiques de détection d'hameçonnage qui peuvent détecter les techniques utilisées pour tromper les utilisateurs.

Il est essentiel de partitionner les données en ensembles distincts pour l'apprentissage, la validation et les tests.

- **Partitionnement aléatoire** : Les données sont mélangées de manière aléatoire, puis réparties en ensembles d'apprentissage, de validation et de test selon un ratio prédéfini. Par exemple, on peut réserver 70% des données pour l'apprentissage, 15% pour la validation et 15% pour les tests.
- **Partitionnement croisé (cross-validation)** : Cette méthode permet d'estimer la performance du modèle de manière plus robuste en utilisant plusieurs partitions différentes des données. La validation croisée k-fold est une approche courante, où les données sont divisées en k sous-ensembles égaux.
- **Partitionnement stratifié** : Lorsque les classes cibles sont déséquilibrées, il est important de préserver la répartition des classes lors de la partition. Le partitionnement stratifié garantit que chaque ensemble contient une proportion équitable d'exemples de chaque classe. Cela évite un biais potentiel lors de l'évaluation du modèle [27]

2. Apprentissage des modèles de Machine Learning

Dans le cadre de la réalisation de ce mémoire, nous avons utilisé un ensemble de données **dataset** spécifique pour soutenir notre recherche et nos analyses.

2.1 Le Dataset

2.1.1. Définition

Un dataset, également appelé ensemble de données, est une collection structurée de données regroupées pour un usage commun. Il s'agit d'une représentation organisée d'informations qui peut être utilisée dans l'analyse, la recherche, l'apprentissage automatique (machine learning) et d'autres domaines d'étude.

Un dataset peut prendre différentes formes, selon le domaine d'application et le type de données concernées. Il peut être constitué de fichiers textuels, de tableaux de données, de bases de données relationnelles, de documents multimédias ou d'autres formats numériques.

Les datasets sont généralement préparés en respectant une structure et une organisation spécifiques. Ils peuvent inclure des variables, des attributs ou des champs qui décrivent les différentes caractéristiques des données [28].

2.1.2. Principales sources pour accéder à des datasets

Voici quelques-unes des sources les plus populaires pour trouver des datasets :

- **Kaggle** : Kaggle est une plateforme en ligne populaire qui héberge une vaste collection de datasets provenant de divers domaines. Il propose également des compétitions de data science où vous pouvez mettre vos compétences à l'épreuve en utilisant les datasets disponibles.
- **UCI Machine Learning Repository** : L'UCI Machine Learning Repository est une archive en ligne contenant une grande variété de datasets utilisés dans le domaine de l'apprentissage automatique. Vous pouvez trouver des datasets classés par thème et utilisés dans des études de recherche.
- **Google Dataset Search** : Google Dataset Search est un moteur de recherche spécifiquement conçu pour trouver des datasets en ligne. Il explore les sites web et les répertoires où les datasets sont hébergés et permet de trouver rapidement des sources fiables de données.
- **GitHub** : GitHub est une plateforme de développement collaboratif populaire qui héberge également de nombreux datasets publics.

2.1.3. Caractéristiques du dataset utilisé et son utilisation

La composition du dataset utilisé dans cette étude sur le système de détection d'hameçonnage est d'une importance cruciale pour garantir la représentativité et la diversité des données.

Cette composition diversifiée permet d'entraîner le modèle de détection à reconnaître les différentes formes d'hameçonnage et à améliorer sa capacité à généraliser et à détecter les nouvelles variantes d'attaques.

Le dataset utilisé dans cette étude sur le système de détection d'hameçonnage est composé de 31 variables (ou attributs) qui capturent différentes caractéristiques des URLs potentiellement malveillantes. Voici une brève description de chaque variable :

- Index : L'indice de la variable dans le dataset.
- UsingIP : Indique si l'URL utilise une adresse IP au lieu d'un nom de domaine.
- LongURL : La longueur de l'URL.
- ShortURL : Indique si l'URL est une URL courte générée par un service de raccourcissement d'URL.
- Symbol@ : Indique la présence du symbole "@" dans l'URL.
- Redirecting// : Indique si l'URL contient une redirection.
- PrefixSuffix- : Indique la présence de préfixes ou suffixes inhabituels dans l'URL.
- SubDomains : Le nombre de sous-domaines dans l'URL.
- HTTPS : Indique si l'URL utilise le protocole HTTPS.
- DomainRegLen : La longueur de l'enregistrement de domaine.
- Favicon : Indique la présence d'une icône de site web (favicon) dans l'URL.
- NonStdPort : Indique si l'URL utilise un port non standard.
- HTTPSDomainURL : Indique si l'URL contient le terme "https" dans le domaine ou l'URL.
- RequestURL : Indique si l'URL contient une demande spécifique.
- LinksInScriptTags : Indique si l'URL contient des liens dans les balises de script.
- ServerFormHandler : Indique si l'URL utilise un gestionnaire de formulaire côté serveur.
- InfoEmail : Indique si l'URL contient une adresse e-mail d'information.
- AbnormalURL : Indique si l'URL est anormale ou suspecte.
- WebsiteForwarding : Indique si le site web effectue une redirection.
- StatusBarCust : Indique si la barre d'état personnalisée est utilisée dans l'URL.

- **DisableRightClick** : Indique si le clic droit est désactivé sur le site web
- **UsingPopupWindow** : Indique si des fenêtres contextuelles sont utilisées sur le site web.
- **IframeRedirection** : Indique si la redirection est effectuée via une balise iframe
- **AgeofDomain** : L'âge du domaine.
- **DNSRecording** : Indique si l'enregistrement DNS est présent pour le domaine.
- **WebsiteTraffic** : Le trafic estimé du site web.
- **PageRank** : Le PageRank du site web.
- **GoogleIndex** : Indique si le site web est indexé par Google.
- **LinksPointingToPage** : Le nombre de liens pointant vers la page.
- **Class** : La classe d'hameçonnage (0 pour non-hameçonnage, 1 pour hameçonnage).

2.2 Les outils utilisés

Au cours de la réalisation de ce mémoire nous avons utilisé des éléments suivants :

- **Python** est un langage de programmation interprété multiplateforme. Il favorise la programmation impérative structurée et orientée objet. C'est un langage gratuit de haut niveau. Il nécessite relativement peu de connaissances sur le fonctionnement des ordinateurs à utiliser. Il est relativement facile à apprendre. Il est relativement simple à prendre en main. Enfin, il est largement utilisé en bio-informatique et plus utilisé dans le domaine du Machine Learning, du Big Data et de la Data Science, en analyse de données.



Figure 18: Logo de Python

- **Google Colab** ou **Colaboratory** est un service cloud fourni par Google (gratuit), basé sur **Jupyter Notebook**, pour la formation et la recherche en apprentissage automatique. Cette plateforme vous permet d'entraîner des modèles de machine learning directement dans le cloud. Par conséquent, il n'est pas nécessaire d'installer quoi que ce soit sur un ordinateur autre que le navigateur. C'est une machine virtuelle puissante avec un meilleur accès aux GPU gratuits et variés pour effectuer un apprentissage en profondeur, et pour exécuter PyTorch, OpenCV, Tensorflow ou Keras.



Figure 19: Logo de Google Colab


- **HTML** : langage de balisage utilisé pour structurer le contenu de la page web.
- **CSS** : langage de style utilisé pour la mise en forme et la présentation visuelle de la page web.
- **JavaScript** : langage de programmation utilisé pour ajouter des fonctionnalités interactives à la page web.

3. Expérimentations

Dans cette partie, nous essayons de détailler le déroulement de notre modèle de détection de phishing.

3.1 Importation des bibliothèques et chargement des données

Au démarrage de notre projet de système de détection d'hameçonnage dans Google Colab, la première étape consiste à importer les bibliothèques nécessaires qui seront utilisées tout au long du code. Les bibliothèques jouent un rôle crucial car elles fournissent des fonctionnalités et des outils prêts à l'emploi qui nous permettra d'implémenter efficacement notre solution de détection d'hameçonnage.



```

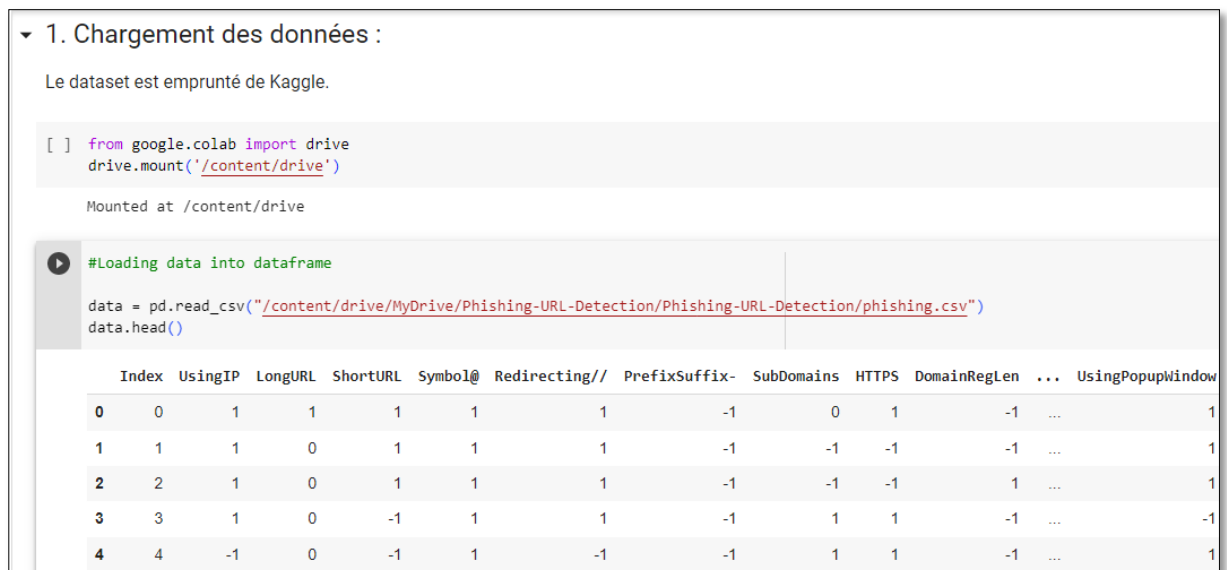
✓ 0s #importation des bibliothèques

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from sklearn import metrics
import warnings
warnings.filterwarnings('ignore')

```

Figure 20: Importation des bibliothèques et les utilitaires

Le chargement des données est une étape essentielle pour notre projet de système de détection d'hameçonnage. Cette étape consiste à importer les données au dataset pertinentes dans notre environnement de développement, afin de les préparer pour l'analyse et la modélisation ultérieures.



▼ 1. Chargement des données :

Le dataset est emprunté de Kaggle.

```

[ ] from google.colab import drive
    drive.mount('/content/drive')

Mounted at /content/drive

```

```

#Loading data into dataframe

data = pd.read_csv("/content/drive/MyDrive/Phishing-URL-Detection/Phishing-URL-Detection/phishing.csv")
data.head()

```

	Index	UsingIP	LongURL	ShortURL	Symbol@	Redirecting//	PrefixSuffix-	SubDomains	HTTPS	DomainRegLen	...	UsingPopupwindow
0	0	1	1	1	1	1	-1	0	1	-1	...	1
1	1	1	0	1	1	1	-1	-1	-1	-1	...	1
2	2	1	0	1	1	1	-1	-1	-1	1	...	1
3	3	1	0	-1	1	1	-1	1	1	-1	...	-1
4	4	-1	0	-1	1	-1	-1	1	1	-1	...	1

Figure 21: Chargement des données de dataset

3.2 Les informations détaillées sur le dataset

Notamment les informations sur les colonnes, les types de données, le nombre de valeurs non nulles et d'autres statistiques utiles. Cela permet d'avoir un aperçu plus complet des données présentes dans le dataset.

```
[ ] #Shape of dataframe

data.shape

(11054, 32)

[ ] #Listing the features of the dataset

data.columns

Index(['Index', 'UsingIP', 'LongURL', 'ShortURL', 'Symbol@', 'Redirecting//',
      'PrefixSuffix-', 'SubDomains', 'HTTPS', 'DomainRegLen', 'Favicon',
      'NonStdPort', 'HTTPSDomainURL', 'RequestURL', 'AnchorURL',
      'LinksInScriptTags', 'ServerFormHandler', 'InfoEmail', 'AbnormalURL',
      'WebsiteForwarding', 'StatusBarCust', 'DisableRightClick',
      'UsingPopupWindow', 'IframeRedirection', 'AgeofDomain', 'DNSRecording',
      'WebsiteTraffic', 'PageRank', 'GoogleIndex', 'LinksPointingToPage',
      'StatsReport', 'class'],
      dtype='object')
```

Figure 22: Les dimensions (la forme) du dataset chargé et la liste des noms de colonnes

Et aussi le nombre total d'entrées non nulles dans chaque colonne, le type de données de chaque colonne, la consommation de mémoire du dataset, etc.

```
[ ] #Information about the dataset

data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11054 entries, 0 to 11053
Data columns (total 32 columns):
#   Column                                Non-Null Count  Dtype
---  ---                                -
0   Index                                11054 non-null  int64
1   UsingIP                             11054 non-null  int64
2   LongURL                             11054 non-null  int64
3   ShortURL                            11054 non-null  int64
4   Symbol@                             11054 non-null  int64
5   Redirecting//                       11054 non-null  int64
6   PrefixSuffix-                       11054 non-null  int64
7   SubDomains                          11054 non-null  int64
8   HTTPS                              11054 non-null  int64
9   DomainRegLen                       11054 non-null  int64
10  Favicon                             11054 non-null  int64
11  NonStdPort                          11054 non-null  int64
12  HTTPSDomainURL                     11054 non-null  int64
13  RequestURL                         11054 non-null  int64
14  AnchorURL                          11054 non-null  int64
```

Figure 23: Le nombre d'entrées et le type de données de chaque colonne

4.3 Préparation et division des données : Détection d'hameçonnage

La préparation et la division des données sont des étapes essentielles dans le domaine de la détection. Elles permettent de mettre en place un processus robuste pour analyser et traiter les données, ainsi que de créer des ensembles distincts pour l'entraînement et l'évaluation des modèles. Ces étapes garantissent une meilleure qualité des résultats et une évaluation précise de l'efficacité des techniques de détection d'hameçonnage utilisées.

▼ 4. Fractionner les données:

Les données sont divisées en ensembles d'entraînement et de test, avec une répartition de 80-20.

```
[ ] # Splitting the dataset into dependant and independant fetature

X = data.drop(["class"],axis =1)
y = data["class"]

[ ] # Splitting the dataset into train and test sets: 80-20 split

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)
X_train.shape, y_train.shape, X_test.shape, y_test.shape

((8843, 30), (8843,), (2211, 30), (2211,))
```

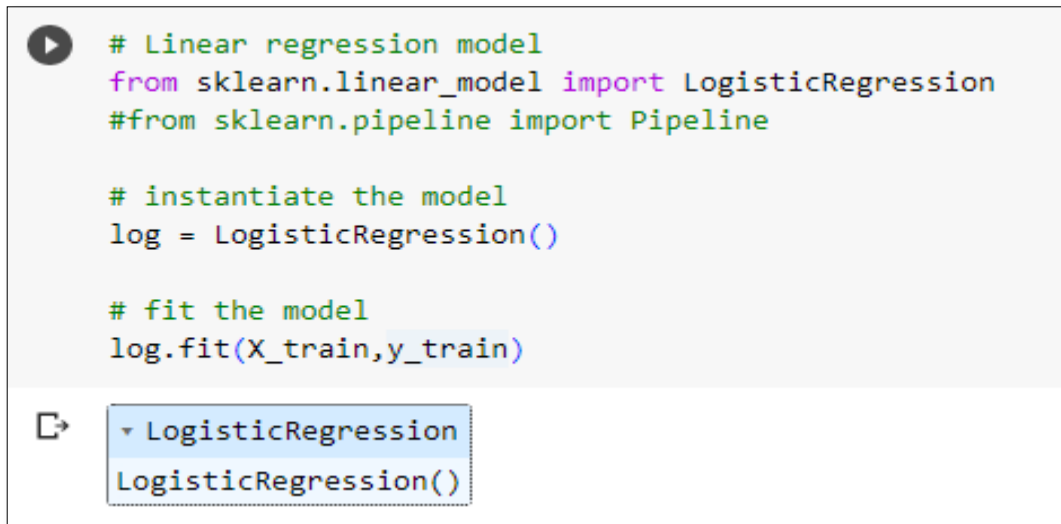
Figure 24: Répartition des données en ensembles d'apprentissage

4.4 Phase d'apprentissage et validation

Dans cette partie, on lance l'apprentissage de notre modèle en lien avec la base d'apprentissage (régression logistique, KNN, SVM et Naïve Bayésien).

4.4.1 La régression logistique

Le code suivant correspond à la mise en œuvre d'un modèle de régression logistique pour la classification binaire.



```
# Linear regression model
from sklearn.linear_model import LogisticRegression
#from sklearn.pipeline import Pipeline

# instantiate the model
log = LogisticRegression()

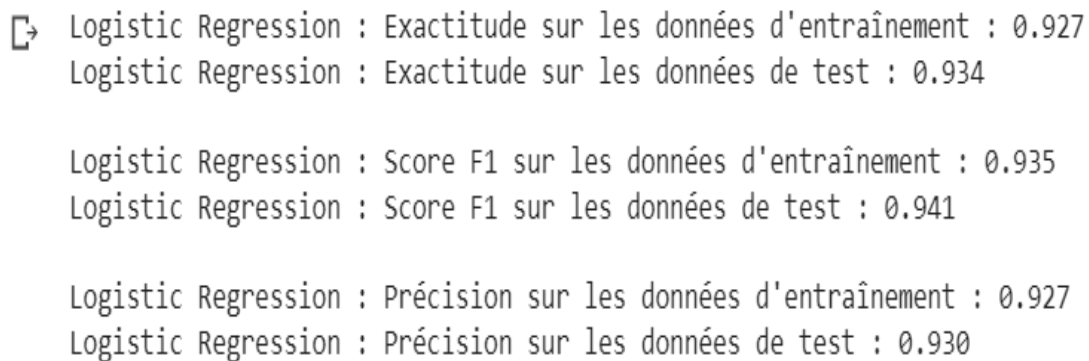
# fit the model
log.fit(X_train,y_train)
```

LogisticRegression

LogisticRegression()

Figure 25: Apprentissage de modèle avec de régression logistique

Après avoir exécuté ces étapes, votre modèle de régression logistique est prêt à être utilisé pour faire des prédictions sur de nouvelles données.



```
Logistic Regression : Exactitude sur les données d'entraînement : 0.927
Logistic Regression : Exactitude sur les données de test : 0.934

Logistic Regression : Score F1 sur les données d'entraînement : 0.935
Logistic Regression : Score F1 sur les données de test : 0.941

Logistic Regression : Précision sur les données d'entraînement : 0.927
Logistic Regression : Précision sur les données de test : 0.930
```

Figure 26: Evaluer les performances du modèle de régression logistique

Ce code permet d'évaluer et d'afficher différentes mesures de performance du modèle de régression logistique, telles que l'exactitude, le score F1 et la précision, tant sur les données d'entraînement que sur les données de test.

4.4.2 Les K plus proches voisins (KNN)

Le code correspond à la mise en œuvre d'un modèle de classification des K plus proches voisins (K-Nearest Neighbors).

```
[ ] # K-Nearest Neighbors Classifier model
    from sklearn.neighbors import KNeighborsClassifier

    # instantiate the model
    knn = KNeighborsClassifier(n_neighbors=1)

    # fit the model
    knn.fit(X_train,y_train)
```

▼ KNeighborsClassifier
KNeighborsClassifier(n_neighbors=1)

Figure 27: Apprentissage de modèle avec KNN

Une fois ces étapes exécutées, votre modèle de classification basé sur les K plus proches voisins est entraîné et prêt à être utilisé pour effectuer des prédictions sur de nouvelles données.

```
➔ K-Nearest Neighbors : Accuracy on training Data: 0.989
  K-Nearest Neighbors : Accuracy on test Data: 0.956

  K-Nearest Neighbors : f1_score on training Data: 0.990
  K-Nearest Neighbors : f1_score on test Data: 0.961

  K-Nearest Neighbors : precision on training Data: 0.989
  K-Nearest Neighbors : precision on test Data: 0.960
```

Figure 28: Evaluer les performances du KNN

Ce code permet d'évaluer et d'afficher différentes mesures de performance du modèle KNN.

Dans la figure suivante, les scores de précision d'entraînement et de test sont présentés pour chaque valeur de `n_neighbors`. Cette visualisation vous permet d'observer comment le modèle se comporte avec différentes configurations.

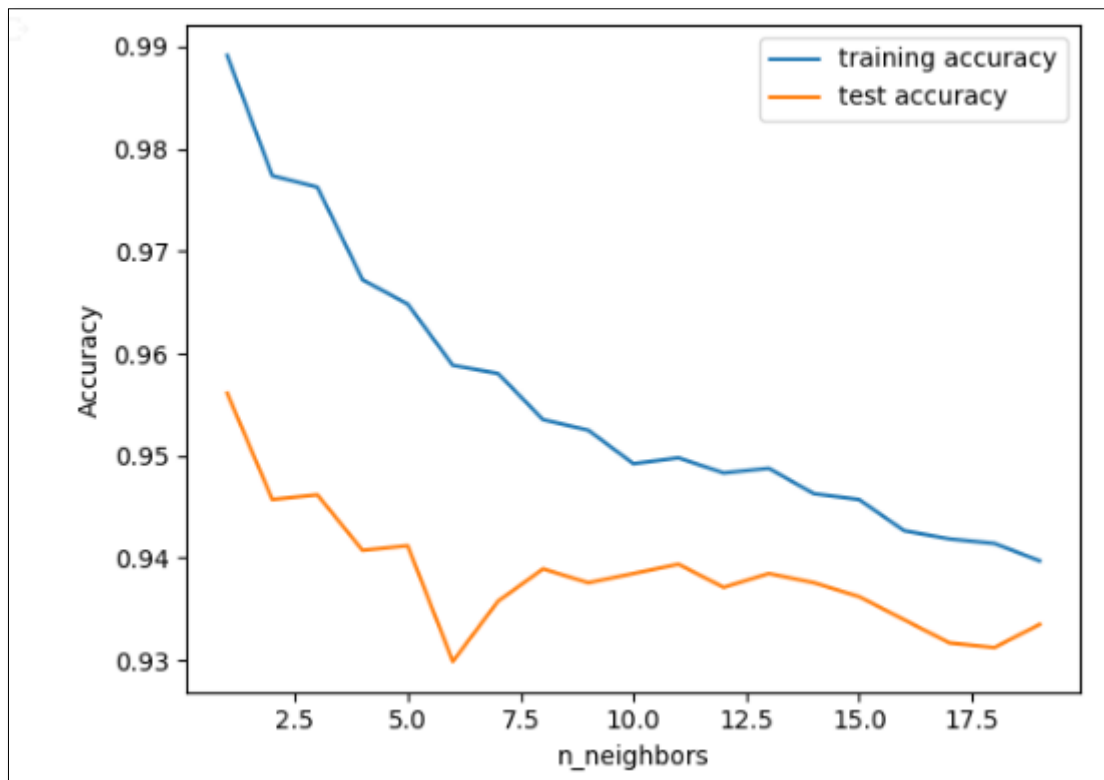


Figure 29: le résultat de précision d'entraînement et de test de KNN

4.4.3 Le Machine à Vecteurs de Support (SVM)

Dans ce code, nous utilisons le modèle de Classifieur à Vecteurs de Support (SVC) pour effectuer une classification. Pour ce faire, nous importons la classe SVC du module sklearn.svm et la classe GridSearchCV du module sklearn.model_selection

```
# Support Vector Classifier model
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV

# defining parameter range
param_grid = {'gamma': [0.1], 'kernel': ['rbf', 'linear']}

svc = GridSearchCV(SVC(), param_grid)

# fitting the model for grid search
svc.fit(X_train, y_train)
```

GridSearchCV
 estimator: SVC
 SVC

Figure 30: Entraînement d'un modèle SVM

Après l'exécution de ce code, le modèle SVM contiendra les meilleurs paramètres trouvés par la recherche en grille et sera prêt à être utilisé pour effectuer des prédictions sur de nouvelles données.

```
Support Vector Machine : Accuracy on training Data: 0.969
Support Vector Machine : Accuracy on test Data: 0.964

Support Vector Machine : f1_score on training Data: 0.973
Support Vector Machine : f1_score on test Data: 0.968

Support Vector Machine : precision on training Data: 0.965
Support Vector Machine : precision on test Data: 0.957
```

Figure 31: Evaluer les performances du modèle SVM

4.4.4 Le Naïve Bayésien (NB)

Le code suivant illustre notre apprentissage avec Naïve Bayésien (NB)

```
# Naive Bayes Classifier Model
from sklearn.naive_bayes import GaussianNB
from sklearn.pipeline import Pipeline

# instantiate the model
nb= GaussianNB()

# fit the model
nb.fit(X_train,y_train)
```

```
▼ GaussianNB
GaussianNB()
```

Figure 32: Utilisation du classifieur Naïve Bayésien pour l'apprentissage de modèle

La figure suivante montre l'utilisation des fonctions de la bibliothèque sklearn.metrics pour calculer les performances du modèle Naïve Bayésien.

```
Naive Bayes Classifier : Accuracy on training Data: 0.605
Naive Bayes Classifier : Accuracy on test Data: 0.605

Naive Bayes Classifier : f1_score on training Data: 0.451
Naive Bayes Classifier : f1_score on test Data: 0.454

Naive Bayes Classifier : precision on training Data: 0.997
Naive Bayes Classifier : precision on test Data: 0.995
```

Figure 33: Analyse des performances du modèle Naïve Bayésien

Les résultats des différentes métriques sont affichés pour évaluer les performances du modèle Naïve Bayésien sur les données d'entraînement et de test.

4.4.5 Application de modèle

L'objectif principal de notre application est de fournir aux utilisateurs un outil simple et efficace pour détecter les URL d'hameçonnage sur un site web donné. Pour cela, l'utilisateur peut soumettre l'URL du site à scanner à notre application.

```
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment
  Use a production WSGI server instead.
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 962-634-770
```

Figure 34: Serveur de développement Flask en cours d'exécution

- **Flask** : Flask est un framework web léger et flexible pour Python. Il est conçu pour faciliter la création d'applications web rapidement et avec un minimum de complexité. Flask ne nécessite pas de dépendances externes et offre une grande simplicité grâce à sa conception minimaliste.

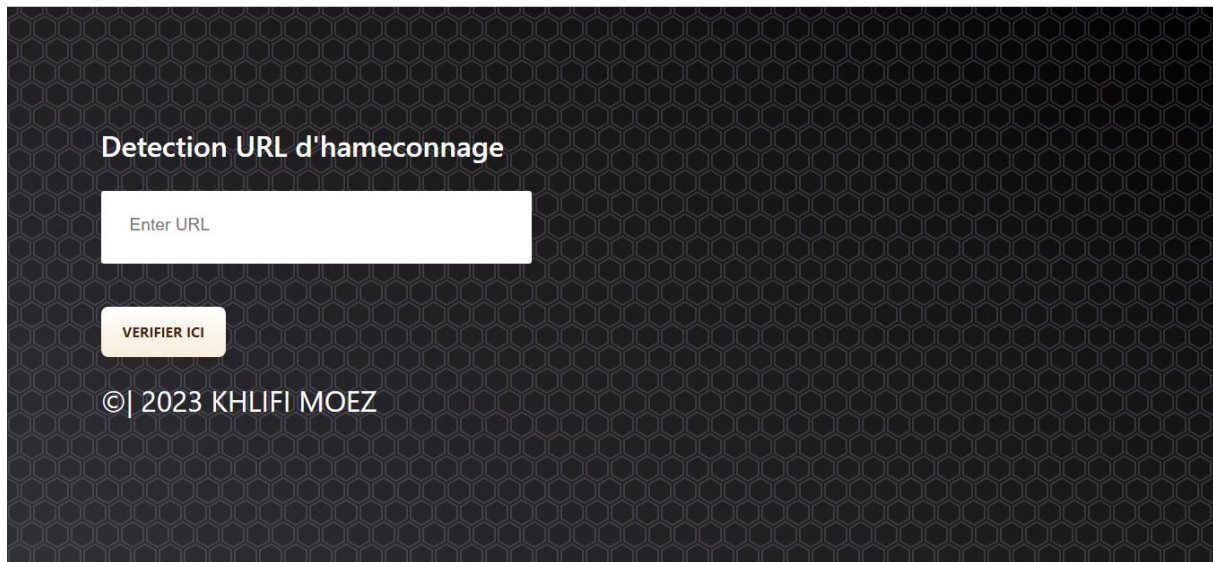


Figure 35 : Interface graphique de site web de détection d'hameçonnage

4. Comparaison des modèles

4.1 Création d'un DataFrame

Pour comparer les performances des modèles, on va créer un DataFrame. Les colonnes de ce dataframe sont les listes créées pour stocker les résultats du modèle.

```
[ ] #creating dataframe
    result = pd.DataFrame({ 'ML Model' : ML_Model,
                           'Accuracy' : accuracy,
                           'f1_score' : f1_score,
                           'Precision': precision,
                           })

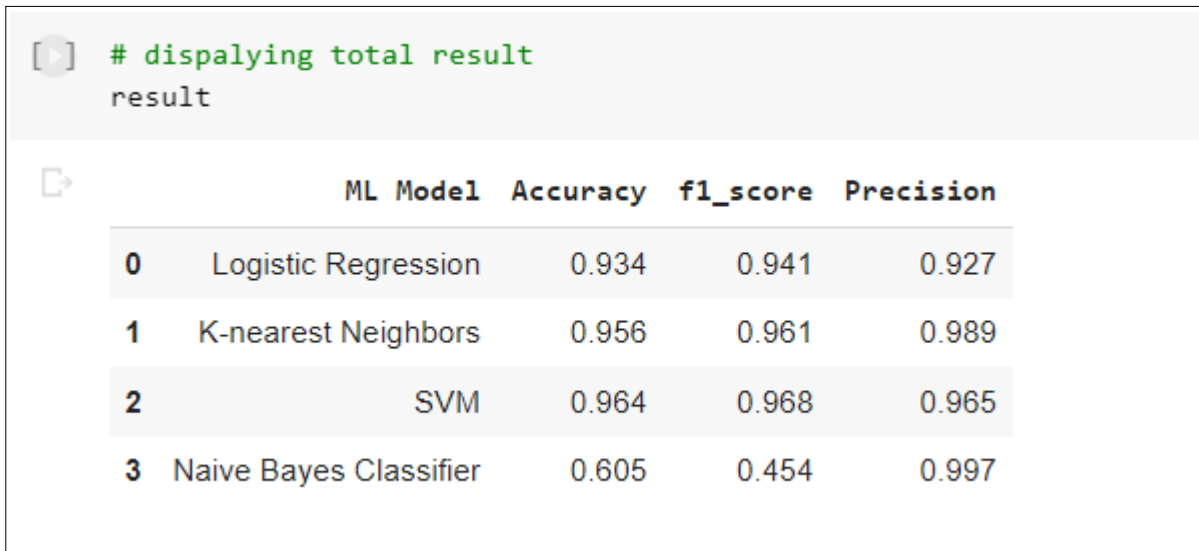
[ ] import pandas as pd

# Création du DataFrame
result = pd.DataFrame.from_records([
    ...{'ML Model': ML_Model[i], 'Accuracy': accuracy[i], 'f1_score': f1_score[i], 'Precision': precision[i]}
    for i in range(len(ML_Model))
])
```

Figure 36: Création un DataFrame en utilisant la bibliothèque pandas

4.2. Comparaison des résultats

Dans notre projet, nous avons développé quatre modèles de classification pour détecter l'hameçonnage, à savoir la régression logistique (LR), la machine à vecteurs de support (SVM), les k plus proches voisins (KNN) et le classifieur bayésien naïve (NB). Nous avons évalué les performances de chaque modèle et comparé les résultats :



The screenshot shows a Jupyter Notebook cell with a green comment icon and the text "# displaying total result". Below it is a table with 5 columns: an index, the ML Model name, Accuracy, f1_score, and Precision. The table contains four rows of data for different models.

	ML Model	Accuracy	f1_score	Precision
0	Logistic Regression	0.934	0.941	0.927
1	K-nearest Neighbors	0.956	0.961	0.989
2	SVM	0.964	0.968	0.965
3	Naive Bayes Classifier	0.605	0.454	0.997

Figure 37 : Comparaison des modèles

4.3 Évaluation des performances des modèles de détection d'hameçonnage

Le meilleur choix parmi les modèles de classification dépend de la métrique d'évaluation que vous considérez comme la plus importante dans votre projet. Dans le tableau que vous avez fourni, vous avez les mesures d'exactitude (Accuracy), de score F1 (F1_score) et de précision (Precision) pour chaque modèle.

D'après les résultats obtenus lors de test des trois méthodes de classification, on peut dire le SVM prend moins du temps pour apprendre mais demande beaucoup du temps pour les tests, à l'inverse de l'KNN qui nécessite un grand temps d'apprentissage et moins du temps pour les tests. et le modèle NB prend moins du temps pour apprendre et pour teste mais sons prestations est plus faible que les autres et aussi LR.

Alors le modèle **SVM** avec une exactitude de 0.964 serait le meilleur choix parmi les quatre modèles.

Conclusion

Dans ce chapitre, nous avons abordé les choix technologiques et l'environnement logiciel utilisés pour développer notre solution. Nous avons présenté la structure de notre solution et les tests effectués, ainsi que notre ensemble de données d'apprentissage et les modèles utilisés. Nous avons ensuite évalué les performances de notre système.

Ce résumé met en évidence les étapes de développement, les choix de modèles et les résultats obtenus lors des tests et de l'évaluation.

Conclusion générale

Après avoir mené à bien ce projet de détection d'hameçonnage, nous pouvons conclure que notre approche basée sur l'apprentissage automatique a été fructueuse. Nous avons réussi à développer et à évaluer plusieurs modèles de classification, à partir desquels nous avons obtenu des performances encourageantes.

En suite en utilisant des algorithmes tels que la régression logistique, la machine à vecteurs de support (SVM), les k plus proches voisins (KNN) et le classifieur bayésien naïf (NB), nous avons pu détecter efficacement les URL d'hameçonnage. Chaque modèle a montré des résultats significatifs dans la détection d'hameçonnage, avec des scores d'exactitude, de score F1 et de précision respectables.

Cependant, il convient de noter que certains modèles ont présenté de meilleures performances que d'autres dans certaines métriques d'évaluation. Par exemple, le modèle SVM a obtenu la meilleure exactitude globale, tandis que le modèle KNN a montré une excellente précision. Ces différences soulignent l'importance de choisir le modèle le mieux adapté aux exigences spécifiques de l'application.

Malgré ces résultats positifs, notre système présente également certaines limites. L'un des principaux défis auxquels nous avons été confrontés est la disponibilité d'un ensemble de données de grande taille et représentatif pour l'apprentissage automatique.

En conclusion, ce projet nous a permis de mettre en œuvre et d'évaluer avec succès des modèles de détection d'hameçonnage basés sur l'apprentissage automatique. Nous sommes satisfaits des résultats obtenus, tout en reconnaissant les opportunités d'amélioration et les perspectives prometteuses pour poursuivre nos travaux dans ce domaine crucial de la sécurité en ligne.

Références bibliographiques

- [1] : paulo, b. p. les facteurs clés de succès dans la lutte contre les cyberattaques par phishing en entreprise
- [2] : Mattatia, F. (2014). L'usurpation d'identité sur internet dans tous ses états. *Revue de science criminelle et de droit pénal comparé*, 2(2), 331-337.
- [3] : 154 Cyber Security Statistics: 2023 Trends & Data | Terranova Security. (2023, April 24). <https://terrانovasecurity.com/cyber-security-statistics/>
- [4] : SEMMOUD, A., & BENMAMMAR, B. (2020). La sécurité intelligente des réseaux informatiques. *Gestion et contrôle intelligents des réseaux: Sécurité intelligente, optimisation multicritères, Cloud Computing, Internet of Vehicles, radio intelligente*, 1.
- [5]: Stansfield, T. (2023, 13 avril). Q1 2023 Phishing and Malware Report : Phishing Increases 102% QoQ. Vade | AI-Powered, Collaborative Email Security. www.vadesecure.com/en/blog/q1-2023-phishing-and-malware-report-phishing-increases-102-qoq
- [6]: StPutz, J. L. (2019). *Cybercriminalité: criminalité informatique en droit luxembourgeois*. Éditions Larcier.
- [7]: Naru, F., & Laffan, K. *Nudge management*. *l'Économie*, 106.
- [8]: Anatomie d'un email de phishing. (n.d.). <https://www.vadesecure.com/fr/blog/anatomie-dun-email-de-phishing>
- [9]: Le Spear phishing : une variante plus efficace du phishing. (s. d.). CONIX. www.conix.fr/le-spear-phishing-une-variante-plus-efficace-du-phishing/
- [10]: paulo, b. p. les facteurs clés de succès dans la lutte contre les cyberattaques par phishing en entreprise.
- [11]: You are being redirected... (s. d.-b). You are being redirected... www.agilly.net/comment-avoir-une-longueur-davance-sur-les-sites-web-de-phishing/
- [12]: Degrave, E. (2013). L'e-gouvernement et la protection de la vie privée. *Chroniques de Droit public*, (3), 234-241.
- [13]: Jain, A., & Gupta, B. B. (2017). Phishing Detection : Analysis of Visual Similarity Based Approaches. *Security and Communication Networks*, 2017, 1-20. www.doi.org/10.1155/2017/5421046
- [14]: Pugnetti, C., & Casián, C. (2021). Les PME suisses face aux cyberrisques: une enquête sure les attitudes des employés e les failles comportementales.
- [15]: Bonfanti, M. E., & Kohler, K. (2020). Intelligence artificielle et cybersécurité. *Politique de sécurité: analyses du CSS*, 265.
- [16]: Lanzini, L. (2018, 18 novembre). Etat des lieux des technologies de l'intelligence artificielle dans la profession comptable. *Compta Online*. www.compta-online.com/etat-des-lieux-des-technologies-de-intelligence-artificielle-dans-la-profession-comptable-ao3605

- [17]: Trung, N. D., Huy, D. T. N., & Le, T. H. (2021). IoTs, machine learning (ML), AI and digital transformation affects various industries-principles and cybersecurity risks solutions. *Management*, 18, 10-14704.
- [18]: Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H. & Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. *Ieee access*, 6, 35365-35381.
- [19]: da Cruz, J. M. M. (2009). Méthodologie d'évaluation des filtres anti-spam. *Journées Réseaux*, Nante du, 1.
- [20]: Boitmobile. (s. d.). Filtrage anti spam - Définitions Marketing » L'encyclopédie illustrée du marketing. copyright Définitions Marketing - Boitmobile. www.definitions-marketing.com/definition/filtrage-anti-spam/
- [21]: Aublet-Cuvelier, L., & da Cruz, J. M. M. (2011). Les défis et les opportunités techniques du fonctionnement d'un service antispam mutualisé.
- [22]: Boitmobile. (s. d.-b). Filtrage anti spam - Définitions Marketing » L'encyclopédie illustrée du marketing. copyright Définitions Marketing - Boitmobile. www.definitions-marketing.com/definition/filtrage-anti-spam/
- [23]: Fahmy, H. M., & Ghoneim, S. A. (2011, March). PhishBlock: A hybrid anti-phishing tool. In *2011 International Conference on Communications, Computing and Control Applications (CCCA)* (pp. 1-5). IEEE.
- [24]: https://www.researchgate.net/figure/Phishblock-Design-Fahmy-and-Ghoneim-2011_fig5_267156776
- [25]: Tang, L., & Mahmoud, Q. H. (2021). A Survey of Machine Learning-Based Solutions for Phishing Website Detection. *Machine learning and knowledge extraction*, 3(3), 672-694. [www.doi.org/10.3390/make3030034](https://doi.org/10.3390/make3030034)
- [26]: Amini, M. R. (2001). *Apprentissage Automatique et Recherche de l'Information: application à l'Extraction d'Information de surface et au Résumé de texte* (Doctoral dissertation, Paris 6).
- [27]: Paquette, G., Crevier, F., & Aubin, C. (1997). Méthode d'ingénierie d'un système d'apprentissage (MISA). *Revue informations in cognito*, 8, 37-52.
- [28]: Qu'est-ce qu'un dataset ? Comment le manipuler ? (s. d.). *Formation Data Science* | DataScientest.com. <https://datascientest.com/dataset-definition>

Résumé

Ce projet propose un système de détection d'hameçonnage basé sur l'apprentissage automatique en utilisant les algorithmes SVM, LR, KNN et NB.

Les données d'entraînement comprennent des exemples de sites web légitimes ainsi que des tentatives d'hameçonnage. Les caractéristiques pertinentes telles que le contenu du message et les liens sont extraites et utilisées pour entraîner les modèles. Les performances sont évaluées avec des mesures telles que la précision, le rappel et le score F1.

Ce projet vise à renforcer la détection d'hameçonnage pour une meilleure sécurité en ligne.

Mots clés : détection d'hameçonnage, apprentissage automatique, SVM, LR, KNN, NB.

Abstract

This project proposes a phishing detection system based on machine learning using SVM, LR, KNN, and NB algorithms.

The training data consists of examples of legitimate websites, as well as phishing attempts. Relevant features such as message content and links are extracted and used to train the models. Performance is evaluated using measures such as accuracy, recall, and F1 score.

The goal of this project is to enhance phishing detection for improved online security.

Keywords: phishing detection, machine learning, SVM, LR, KNN, NB.

ملخص

يقدم هذا المشروع نظامًا لاكتشاف التصيد الاحتيالي باستخدام تقنيات التعلم الآلي مثل SVM، LR، KNN، و NB. تتضمن بيانات التدريب أمثلة لمواقع الويب شرعية بالإضافة إلى محاولات الصيد الاحتيالي. يتم استخراج الميزات ذات الصلة مثل محتوى الرسالة والروابط واستخدامها لتدريب النماذج. يتم تقييم الأداء باستخدام مقاييس مثل الدقة والاستدعاء ونقاط F1.

يهدف هذا المشروع إلى تعزيز اكتشاف التصيد الاحتيالي لتحقيق أمان أفضل عبر الإنترنت.

الكلمات الرئيسية: اكتشاف التصيد الاحتيالي، التعلم الآلي، SVM، LR، KNN، NB.

