



FACULTÉ POLYDISCIPLINAIRE DE SAFI

## FACULTÉ POLYDISCIPLINAIRE DE SAFI

### Rapport du projet de fin d'études

Pour l'obtention de la licence en Informatique

Sciences Mathématiques et Informatiques

---

# Un système basé sur les techniques d'apprentissage automatique pour la détection des attaques de phishing par e-mail

---

RÉALISÉ PAR :

*MARZOUKI Laila*

*ELHADDADI Ghizlane*

ENCADRÉ PAR :

**PR.MOURDI YOUSSEF**

2023/2024



## Remerciements

Nous souhaitons exprimer notre profonde gratitude à toutes les personnes qui ont contribué à la réalisation de ce travail.

Tout d'abord, nous tenons à remercier chaleureusement notre encadrant **Monsieur Youssef Mourdi**, pour sa supervision attentive, ses conseils avisés, et son soutien constant tout au long de ce projet. Son expertise et ses encouragements ont été essentiels pour mener à bien cette recherche.

Nous adressons également nos sincères remerciements à **Monsieur Said Salloum**, qui a généreusement partagé avec nous son ensemble de données, rendant ainsi possible une partie essentielle de cette étude. Son apport a été crucial pour la progression de notre projet, et nous sommes reconnaissants pour sa disponibilité et son soutien. Nous tenons à exprimer notre gratitude envers nos collègues et amis **Salma Sahmad, Zineb Ezzahi, et Farah Hanti** pour leurs encouragements, leurs feedbacks constructifs, et leur soutien moral indéfectible. Leurs conseils et suggestions ont grandement enrichi notre réflexion et nous ont aidés à surmonter les défis rencontrés au cours de cette étude. Leur collaboration et leurs discussions stimulantes ont été une source inestimable de motivation et d'inspiration.

Enfin, nous adressons nos plus sincères remerciements à nos familles pour leur amour, leur patience, et leur soutien inconditionnel tout au long de cette période exigeante. Leur compréhension et leur soutien moral ont été des piliers essentiels qui nous ont permis de nous concentrer pleinement sur notre travail.

Nous vous remercions tous sincèrement pour votre précieuse contribution et votre soutien tout au long de ce projet.

## Résumé

Le phishing demeure une menace persistante et sophistiquée dans le domaine de la sécurité informatique, exploitant l'ingénierie sociale et des techniques de tromperie pour compromettre la sécurité des utilisateurs. Cette étude propose une approche novatrice pour la détection précoce et efficace des e-mails de phishing en utilisant des techniques avancées de traitement du langage naturel (NLP) et d'apprentissage automatique. Notre méthodologie repose sur une analyse exhaustive des caractéristiques lexicales, syntaxiques et sémantiques des e-mails suspects, combinée à l'extraction automatique de caractéristiques et à la modélisation prédictive. Nous avons exploré plusieurs techniques adaptées à la complexité des données analysées, incluant les SVM, la régression logistique, les k plus proches voisins (KNN), les forêts aléatoires, le gradient boosting, les arbres de décision, Naïve Bayes, ainsi que les réseaux de neurones multicouches (MLP), les réseaux de neurones convolutifs (CNN) et les réseaux de neurones récurrents (RNN).

Nous avons développé et évalué ces modèles de détection de phishing en utilisant des ensembles de données diversifiés et représentatifs, intégrant des techniques de prétraitement des données robustes pour améliorer la généralisation et la résilience aux variantes de phishing. Nos résultats expérimentaux démontrent une précision significativement améliorée par rapport aux approches traditionnelles, avec une sensibilité accrue à détecter les tentatives de phishing même lorsque celles-ci utilisent des tactiques d'obfuscation avancées.

En conclusion, notre approche représente une avancée importante dans la lutte contre le phishing en renforçant la sécurité des systèmes et en protégeant les utilisateurs contre les fraudes en ligne. En exploitant les SVM pour leur capacité à gérer efficacement les caractéristiques complexes des e-mails de phishing, ainsi que d'autres techniques de pointe comme les réseaux de neurones, notre méthodologie offre une solution robuste et adaptable pour une détection proactive et fiable. Ces résultats suggèrent la possibilité d'intégrer nos modèles dans des systèmes de sécurité informatique pour une défense accrue contre les attaques cybernétiques sophistiquées.

**Mots-clés :** Phishing par e-mail, apprentissage automatique, SVM, réseaux de neurones, traitement du langage naturel, cybersécurité

## Abstract

Phishing remains a persistent and sophisticated threat in the field of computer security, exploiting social engineering and deception techniques to compromise user security. This study proposes an innovative approach for early and effective detection of phishing emails using advanced natural language processing (NLP) and machine learning techniques. Our methodology relies on a comprehensive analysis of lexical, syntactic, and semantic features of suspicious emails, combined with automatic feature extraction and predictive modeling. We explored multiple techniques tailored to the complexity of the analyzed data, including SVM, logistic regression, k-nearest neighbors (KNN), random forests, gradient boosting, decision trees, Naive Bayes, as well as multi-layer perceptrons (MLP), convolutional neural networks (CNN), and recurrent neural networks (RNN).

We developed and evaluated these phishing detection models using diverse and representative datasets, integrating robust data preprocessing techniques to enhance generalization and resilience against phishing variants. Our experimental results demonstrate significantly improved accuracy compared to traditional approaches, with increased sensitivity in detecting phishing attempts even when employing advanced obfuscation tactics.

In conclusion, our approach represents a significant advancement in combating phishing by enhancing system security and protecting users against online fraud. By leveraging SVM for effectively managing complex phishing email features, along with other cutting-edge techniques like neural networks, our methodology provides a robust and adaptable solution for proactive and reliable detection. These findings suggest the possibility of integrating our models into computer security systems for enhanced defense against sophisticated cyber attacks.

**Keywords :** Email phishing, machine learning, SVM, neural networks, natural language processing, cybersecurity

# Table des figures

1	Augmentation du Nombre d'Articles Scientifiques sur la Détection du Phishing par e-mail (2010-2023) . . . . .	1
2	Industries les plus ciblées, 1er trimestre 2024 . . . . .	3
1.1	Réseaux neuronaux . . . . .	8
1.2	Naïve Bayes . . . . .	8
1.3	Régression logistique . . . . .	9
1.4	Machines à vecteurs de support . . . . .	9
1.5	Knn . . . . .	10
1.6	Forêt d'arbres décisionnels . . . . .	10
1.7	Clustering . . . . .	11
1.8	Processus de l'Apprentissage Automatique . . . . .	13
1.9	Réseaux neuronaux convolutifs . . . . .	15
1.10	Réseaux neuronaux récurrents . . . . .	16
1.11	Réseaux adversariaux génératifs [2] . . . . .	17
1.12	Exemple de courbe ROC avec l'AUC . . . . .	20
1.13	Comparaison des secteurs les plus ciblés par trimestre . . . . .	22
3.1	Processus de Détection des e-mails de Phishing . . . . .	33
3.2	Représentation des classes dans l'ensemble de données équilibrée final . . . . .	35
3.3	Représentation des classes dans l'ensemble de données déséquilibrée final 1 . . . . .	35
3.4	Représentation des classes dans l'ensemble de données déséquilibrée final 2 . . . . .	36
4.1	Comparaison des performances des modèles de classification avec TF-IDf sans sélection de caractéristiques sur données équilibrées . . . . .	46
4.2	Comparaison des performances des modèles de classification avec TF-IDF et sélection de caractéristiques sur données équilibrées . . . . .	47
4.3	Performances des différents modèles avec Word2vec sur données équilibrées . . . . .	48
4.4	Comparaison graphique des performances des modèles de classification avec TF-IDF sans sélection de caractéristiques sur données déséquilibrées1 . . . . .	49
4.5	Comparaison graphique des performances des modèles de classification avec TF-IDF et sélection de caractéristiques sur données déséquilibrées1 . . . . .	50
4.6	Illustration graphique des performances des modèles avec Word2vec sur données déséquilibrées1 . . . . .	51
4.7	Illustration graphique des résultats des modèles de classification avec TF-IDF sans sélection de caractéristiques sur données déséquilibrées2 . . . . .	52
4.8	Illustration graphique des résultats des modèles de classification avec TF-IDF et selection de caractéristiques sur données déséquilibrées2 . . . . .	53
4.9	Comparaison des résultats des modèles avec Word2vec sur données déséquilibrées2 . . . . .	54
4.10	Matrice de confusion du modèle MLP sans sélection de caractéristiques sur données équilibrées . . . . .	56

## Table des figures

---

4.11 Matrice de confusion du modèle CNN sans sélection de caractéristiques sur données équilibrées . . . . .	56
4.12 Matrice de confusion du modèle RNN sans sélection de caractéristiques sur données équilibrées . . . . .	56
4.13 Matrice de confusion du modèle MLP avec selection de caractéristiques sur données équilibrées . . . . .	57
4.14 Matrice de confusion du modèle CNN avec selection de caractéristiques sur données équilibrées . . . . .	57
4.15 Matrice de confusion du modèle RNN avec selection de caractéristiques sur données équilibrées . . . . .	57
4.16 Matrice de confusion du modèle MLP sans sélection des caractéristiques sur données déséquilibrées1 . . . . .	58
4.17 Matrice de confusion du modèle CNN sans sélection des caractéristiques sur données déséquilibrées1 . . . . .	58
4.18 Matrice de confusion du modèle RNN sans sélection des caractéristiques sur données déséquilibrées1 . . . . .	58
4.19 Matrice de confusion du modèle MLP avec sélection des caractéristiques sur données déséquilibrées1 . . . . .	58
4.20 Matrice de confusion du modèle CNN avec sélection des caractéristiques sur données déséquilibrées1 . . . . .	58
4.21 Matrice de confusion du modèle RNN avec sélection des caractéristiques sur données déséquilibrées1 . . . . .	58
4.22 Matrice de confusion du modèle MLP sans sélection des caractéristiques sur données déséquilibrées2 . . . . .	59
4.23 Matrice de confusion du modèle CNN sans sélection des caractéristiques sur données déséquilibrées2 . . . . .	59
4.24 Matrice de confusion du modèle RNN sans sélection des caractéristiques sur données déséquilibrées2 . . . . .	59
4.25 Matrice de confusion du modèle MLP avec sélection des caractéristiques sur données déséquilibrées2 . . . . .	60
4.26 Matrice de confusion du modèle CNN avec sélection des caractéristiques sur données déséquilibrées2 . . . . .	60
4.27 Matrice de confusion du modèle RNN avec sélection des caractéristiques sur données déséquilibrées2 . . . . .	60
4.28 Distribution des Scores des En-têtes d’E-mails . . . . .	62
4.29 Distribution des Scores des Liens d’E-mails . . . . .	63
4.30 Screenshot de l’application de visualisation des résultats . . . . .	66

# Liste des tableaux

1.1	Les types d'intelligence artificielle . . . . .	7
1.2	Matrice de confusion . . . . .	19
1.3	Caractéristiques des victimes de phishing par e-mail . . . . .	23
2.1	Résumé des études sur la détection de phishing (Partie 1) . . . . .	30
2.2	Résumé des études sur la détection de phishing (Partie 2) . . . . .	31
3.1	Détails de données utilisées . . . . .	34
3.2	Répartition des classes dans l'ensemble de données équilibrée final . . . . .	35
3.3	Répartition des classes dans l'ensemble de données déséquilibrée final 1 . . . . .	35
3.4	Répartition des classes dans l'ensemble de données déséquilibrée final 2 . . . . .	36
3.5	Critères pour évaluer les URL . . . . .	43
4.1	Performance des modèles de classification avec TF-IDf sans sélection de caractéristiques sur données équilibrées . . . . .	46
4.2	Performance des modèles de classification avec TF-IDF et sélection de caractéristiques sur données équilibrées . . . . .	47
4.3	Performance des modèles de classification avec Word2Vec sur données équilibrées	48
4.4	Performance des modèles de classification avec TF-IDF sans sélection de caractéristiques sur données déséquilibrées1 . . . . .	49
4.5	Performance des modèles de classification avec TF-IDF et sélection de caractéristiques sur données déséquilibrées1 . . . . .	50
4.6	Performance des modèles de classification avec Word2Vec sur données déséquilibrées1 . . . . .	51
4.7	Performance des modèles de classification avec TF-IDF sans sélection de caractéristiques sur données déséquilibrées2 . . . . .	52
4.8	Performance des modèles de classification avec TF-IDF et sélection de caractéristiques sur données déséquilibrées2 . . . . .	53
4.9	Performance des modèles de classification avec Word2Vec sur données déséquilibrées2 . . . . .	54
4.10	Comparaison des Performances sur Différents Jeux de Données . . . . .	55
4.11	Comparaison des Méthodes d'Extraction de Caractéristiques . . . . .	55
4.12	Comparaison avec et sans Sélection de Caractéristiques . . . . .	55
4.13	Performance des modèles de réseaux de neurones sans sélection de caractéristiques sur données équilibrées . . . . .	56
4.14	Performance des modèles de reseaux de neurones avec selection de caractéristiques sur données équilibrées . . . . .	56
4.15	Performance des modèles de réseaux de neurones sans sélection des caractéristiques sur données déséquilibrées1 . . . . .	57

## Liste des tableaux

---

4.16 Performance des modèles de réseaux de neurones avec sélection des caractéristiques sur données déséquilibrées1 . . . . .	58
4.17 Performance des modèles de réseaux de neurones sans sélection des caractéristiques sur données déséquilibrées2 . . . . .	59
4.18 Performance des modèles de réseaux de neurones avec sélection des caractéristiques sur données déséquilibrées2 . . . . .	59
4.19 Résultats de l'analyse sémantique du corps . . . . .	61
4.20 Résultats de l'analyse des en-têtes . . . . .	61
4.21 Résultats de l'analyse des liens . . . . .	62
4.22 Résultats de combinaison des scores . . . . .	63

# Table des matières

<b>Résumé</b>	i
<b>Abstract</b>	ii
<b>Table des figures</b>	iii
<b>Liste des tableaux</b>	v
<b>Table des matières</b>	vii
<b>Introduction générale</b>	1
<b>1 Contexte général et concepts de base</b>	6
1.1 Introduction . . . . .	6
1.2 Intelligence Artificielle (IA) . . . . .	6
1.3 Apprentissage Automatique (AA) . . . . .	7
1.3.1 Apprentissage supervisé . . . . .	7
1.3.2 Apprentissage non supervisé . . . . .	11
1.3.3 Apprentissage par renforcement . . . . .	12
1.3.4 Apprentissage semi-supervisé . . . . .	12
1.3.5 Apprentissage par transfert . . . . .	12
1.3.6 Processus de l'Apprentissage Automatique . . . . .	13
1.4 Apprentissage Profond (AP) . . . . .	15
1.4.1 Définition et Concept de Base . . . . .	15
1.4.2 Types de Réseaux de Neurones . . . . .	15
1.5 Traitement Automatique du Langage Naturel (TALN) . . . . .	17
1.5.1 Définition et Objectifs . . . . .	17
1.5.2 Techniques et Méthodes . . . . .	17
1.5.3 Techniques de Représentation . . . . .	18
1.6 Évaluation des performances . . . . .	18
1.6.1 Métriques d'évaluation . . . . .	18
1.6.2 Techniques de validation . . . . .	20
1.7 Phishing par e-mail . . . . .	22
1.7.1 Définition et Principes de Base . . . . .	22
1.7.2 Victimes Ciblées . . . . .	23
1.7.3 Techniques utilisées dans le Phishing . . . . .	23
1.7.4 Méthodes de Détection . . . . .	24
1.8 Conclusion . . . . .	24
<b>2 Revue de littérature</b>	25

2.1	Introduction . . . . .	25
2.2	Revue des techniques de détection de phishing par e-mail . . . . .	25
2.2.1	Techniques basées sur le traitement automatique du langage naturel (NLP) . . . . .	26
2.2.2	Études sur l'apprentissage automatique et les approches de classification . . . . .	26
2.2.3	Approches de deep learning et nouvelles techniques . . . . .	28
2.2.4	Comparaison des approches et synthèse . . . . .	30
2.3	Conclusion . . . . .	31
<b>3</b>	<b>Méthodologie</b>	<b>32</b>
3.1	Introduction . . . . .	32
3.2	Méthodologie générale . . . . .	32
3.3	Analyse du corps . . . . .	33
3.3.1	jeu des données . . . . .	33
3.3.2	Techniques de nettoyage et de normalisation des données . . . . .	34
3.3.3	Extraction et selection des caractéristiques . . . . .	36
3.3.4	Architecture des Algorithmes Utilisés . . . . .	38
3.3.5	Métriques d'évaluation . . . . .	40
3.3.6	Intégration du Score Sémantique aux Résultats du Modèle . . . . .	40
3.4	Analyse des entêtes . . . . .	41
3.5	Analyse des liens . . . . .	42
3.6	Combinaison des scores . . . . .	44
3.7	Conclusion . . . . .	44
<b>4</b>	<b>Résultats et discussion</b>	<b>45</b>
4.1	Introduction . . . . .	45
4.2	Présentation des résultats de l'analyse du corps . . . . .	45
4.2.1	Résultats des Modèles de Machine Learning . . . . .	45
4.2.2	Résultats des Modèles de Deep Learning . . . . .	56
4.2.3	Résultats de l'analyse sémantique du corps . . . . .	60
4.3	Résultats de l'Analyse des En-têtes et des Liens . . . . .	61
4.3.1	Analyse des En-têtes . . . . .	61
4.3.2	Analyse des Liens . . . . .	62
4.3.3	Combinaison des scores . . . . .	63
4.4	Discussion . . . . .	64
4.5	Conclusion . . . . .	64
<b>Conclusion</b>		<b>65</b>
<b>Bibliographie</b>		<b>68</b>



# Introduction générale

## Contexte de recherche

La recherche sur le phishing par e-mail occupe une place cruciale dans le domaine de la cybersécurité, étant donné la persistance et la sophistication croissante de cette menace. Les travaux de recherche visent à comprendre les mécanismes sous-jacents du phishing par e-mail, à développer des techniques de détection efficaces et à proposer des stratégies de prévention robustes. Selon une étude récente publiée dans le Journal of Cybersecurity Research [10] , les chercheurs ont constaté une augmentation significative du nombre d'articles scientifiques consacrés au phishing par e-mail au cours des dernières années, reflétant l'ampleur de cette menace et l'importance de trouver des solutions innovantes pour la contrer. Les approches de recherche actuelles comprennent l'utilisation de techniques avancées telles que le machine learning et le deep learning pour détecter les caractéristiques subtiles des e-mails de phishing, ainsi que l'analyse des tendances et des motifs des attaques pour anticiper les futures menaces. En outre, les chercheurs explorent également des stratégies de sensibilisation et de formation pour éduquer les utilisateurs sur les dangers du phishing et les meilleures pratiques pour s'en protéger. Dans ce contexte, la recherche continue de jouer un rôle essentiel dans la lutte contre le phishing par e-mail, en contribuant à renforcer la sécurité des individus et des organisations dans un environnement numérique de plus en plus complexe et interconnecté.

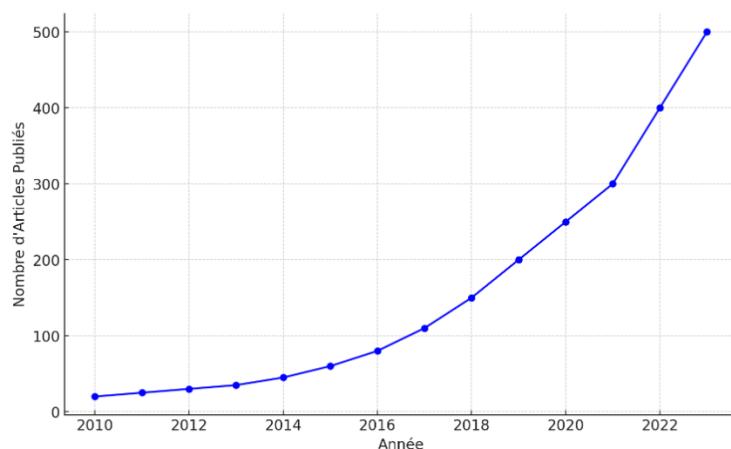


FIGURE 1 – Augmentation du Nombre d'Articles Scientifiques sur la Détection du Phishing par e-mail (2010-2023)

# Problématique

## Définition précise du problème de phishing par e-mail.

Le phishing est une forme de cybercriminalité en constante augmentation, caractérisée par l'utilisation d'e-mails frauduleux, de sites web malveillants ou d'autres moyens électroniques pour tromper les utilisateurs et les inciter à divulguer des informations sensibles telles que des mots de passe, des numéros de carte de crédit ou des informations bancaires. Ce problème croissant représente une menace sérieuse pour la sécurité en ligne des individus et des organisations. Les attaquants utilisent diverses techniques de phishing pour voler les informations sensibles des utilisateurs. Ces techniques comprennent l'envoi d'e-mails ou de messages instantanés frauduleux imitant des institutions légitimes, la création de sites web falsifiés imitant des sites de confiance, ou encore l'utilisation de programmes malveillants pour collecter secrètement des données sur les activités en ligne des utilisateurs.

Les chercheurs en cybersécurité déploient plusieurs techniques pour détecter et contrer le phishing, telles que l'analyse des caractéristiques des e-mails suspects, la surveillance des activités sur les sites web potentiellement dangereux, ou encore l'utilisation de systèmes de détection d'intrusion pour identifier les comportements suspects sur les réseaux informatiques. Cependant, malgré les progrès dans ce domaine, le phishing reste un défi persistant en raison de sa nature évolutive et de l'utilisation de méthodes de plus en plus sophistiquées par les cybercriminels.

Les problèmes émergents dans ce domaine comprennent la difficulté à distinguer les e-mails de phishing des communications légitimes, l'adaptation rapide des attaquants aux nouvelles techniques de détection, et la nécessité de protéger la vie privée des utilisateurs tout en luttant contre le phishing. Les efforts de recherche continus visent à développer des solutions plus efficaces pour contrer cette menace persistante dans le paysage numérique actuel.

## Pourquoi c'est une menace majeure et pourquoi il est crucial de la détecter efficacement ?

La menace du phishing est considérée comme majeure pour plusieurs raisons, et il est crucial de la détecter efficacement pour plusieurs motifs :

- **Impact financier** : Selon le rapport sur les tendances de l'activité de phishing de l'Anti-Phishing Working Group (APWG), les pertes financières dues au phishing ont atteint plus de 1,8 milliard de dollars rien qu'au premier trimestre de 2024.
- **Nombre d'attaques** : L'APWG a signalé une augmentation significative du nombre d'attaques de phishing, avec plus de 2,5 millions d'e-mails de phishing uniques signalés au cours de l'année 2023.
- **Variété des cibles** : Les statistiques montrent que les attaques de phishing ne se limitent pas à un seul secteur, mais touchent une grande variété d'industries. Par exemple, les réseaux sociaux étaient le secteur le plus fréquemment attaqué au premier trimestre de 2024, représentant 37,6% de toutes les attaques de phishing, suivis par les services financiers et les services de paiement en ligne.

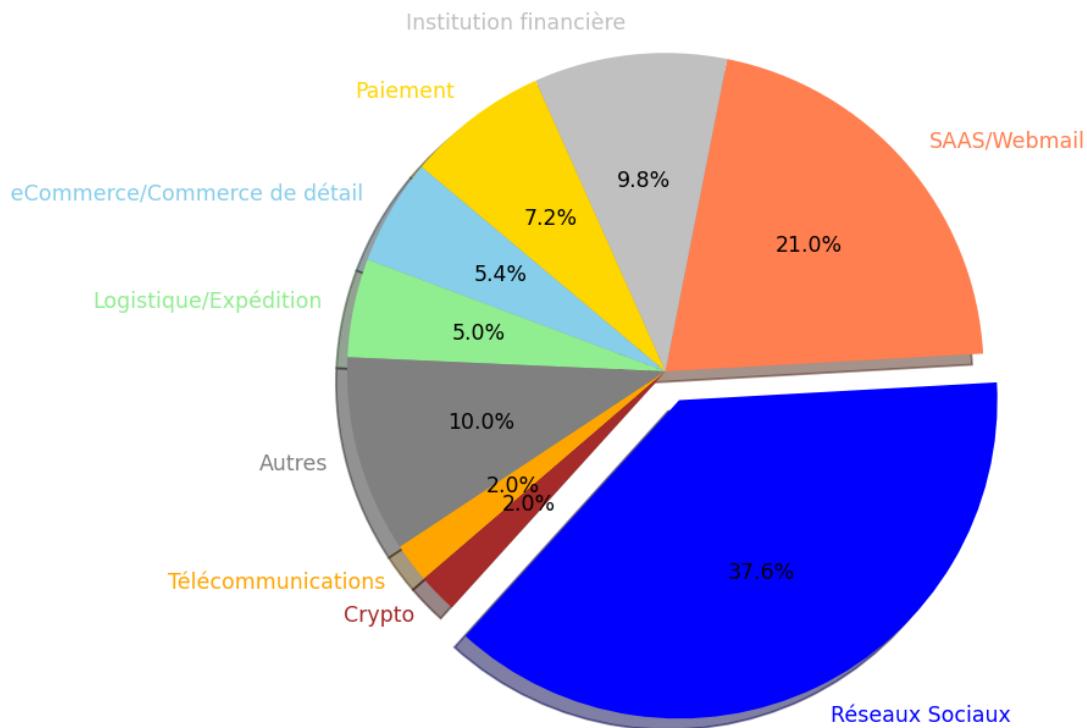


FIGURE 2 – Industries les plus ciblées, 1er trimestre 2024

- **Évolution des techniques** : Les attaquants de phishing utilisent des techniques de plus en plus sophistiquées pour contourner les défenses de sécurité. Par exemple, OpSec Security a observé une augmentation de 30% des volumes de détection de vishing (fraude par téléphone) et de smishing (fraude par SMS) au premier trimestre de 2024 par rapport au trimestre précédent [11].

Ces statistiques mettent en évidence l'ampleur de la menace que représente le phishing et soulignent l'importance cruciale de détecter efficacement ces attaques pour protéger les individus et les entreprises contre les pertes financières, les violations de données et les dommages à la réputation. Une détection précoce et précise permet de prendre des mesures de sécurité appropriées pour contrer ces menaces et limiter leurs impacts.

## Objectifs

### Objectif général :

Le projet vise à développer et à tester des méthodes de détection efficaces du phishing par e-mail en utilisant des techniques de machine learning et de deep learning. L'objectif principal est de créer un système performant capable de détecter automatiquement les e-mails de phishing avec une précision élevée, afin de renforcer la sécurité des utilisateurs et des entreprises contre cette menace croissante dans le domaine de la cybersécurité.

### Objectifs spécifiques :

- Rassembler un ensemble de données représentatif d'e-mails légitimes et de e-mails de

phishing, puis les prétraiter pour les rendre adaptés à l'entraînement des modèles de machine learning et de deep learning.

- Concevoir et mettre en œuvre des modèles de machine learning, tels que des classificateurs basés sur des algorithmes d'apprentissage supervisé (par exemple, SVM, Random Forest) pour identifier les caractéristiques distinctives des e-mails de phishing.
- Explorer les architectures de réseaux de neurones profonds, telles que les réseaux de neurones convolutifs (CNN) et les réseaux de neurones récurrents (RNN), pour extraire des motifs complexes et subtiles des e-mails de phishing.
- Entraîner les modèles sur les données prétraitées et les évaluer en utilisant des mesures de performance telles que la précision, le rappel et la courbe ROC pour évaluer leur efficacité dans la détection du phishing.
- Optimiser les paramètres des modèles pour améliorer leurs performances et valider leur généralisabilité en les testant sur des ensembles de données indépendants pour garantir leur fiabilité et leur robustesse dans des environnements réels.

En combinant les approches de machine learning et de deep learning, le projet vise à fournir une solution avancée et adaptable pour détecter le phishing par e-mail de manière efficace et proactive, contribuant ainsi à renforcer la sécurité des utilisateurs et des organisations contre cette menace persistante.

## Organisation du rapport

Ce rapport est structuré en plusieurs chapitres afin de fournir une analyse complète et détaillée de la détection de phishing par e-mail à l'aide de techniques de machine learning et de deep learning.

L'introduction générale pose le cadre de la recherche en présentant le contexte, la problématique, les objectifs du projet et l'organisation du rapport. Cette section établit le fondement nécessaire pour comprendre les enjeux de la détection de phishing et les méthodes utilisées pour y répondre.

Le premier chapitre, intitulé "Contexte général et concepts de base", présente une introduction aux principaux sujets et théories fondamentales qui sous-tendent l'étude. Il explore le cadre général du domaine de recherche, définissant les termes clés et les concepts essentiels. Ce chapitre pose les bases en fournissant un aperçu des éléments contextuels et théoriques nécessaires à la compréhension des chapitres suivants, en établissant une compréhension claire et commune des notions de base pour les lecteurs.

Le deuxième chapitre est consacré à l'état de l'art et fournit une revue des travaux de recherche existants sur le phishing et les techniques de cybersécurité, ainsi qu'une analyse critique des approches actuelles de détection utilisant le machine learning et le deep learning. Cette section offre un aperçu des avancées récentes dans le domaine et identifie les lacunes ou les opportunités pour la recherche future.

Le troisième chapitre décrit en détail la méthodologie utilisée dans l'étude, incluant les techniques de nettoyage des données, l'implémentation des algorithmes, et les métriques d'évaluation

## **Introduction générale**

---

des performances. Cette section détaille les étapes techniques de la création et de l'évaluation des modèles de détection de phishing.

Le quatrième chapitre présente les résultats et leur discussion, comparant les performances des différents modèles et analysant les résultats obtenus. Enfin, le dernier chapitre offre une conclusion qui récapitule les principales découvertes du projet, discute des implications et des limites de l'étude, et propose des directions pour des recherches futures.

# Chapitre 1

## Contexte général et concepts de base

### 1.1 Introduction

Dans le paysage complexe et dynamique actuel, où la technologie façonne chaque aspect de notre vie quotidienne, la compréhension des concepts fondamentaux et du contexte général devient essentielle. Ce chapitre vise à explorer ces fondations, en mettant en lumière les principaux termes et concepts qui sous-tendent notre sujet d'étude. De l'intelligence artificielle à la cybersécurité, en passant par d'autres avancées technologiques, cette exploration est cruciale pour appréhender les défis et les opportunités qu'elles présentent. En examinant ces bases, nous jetons les bases nécessaires pour aborder ensuite les développements récents et les implications futures dans ce domaine dynamique.

### 1.2 Intelligence Artificielle (IA)

L'intelligence artificielle (IA)[6], définie par *John McCarthy* comme la science de créer des machines intelligentes, englobe diverses techniques et théories pour simuler l'intelligence humaine. Son histoire, marquée par des avancées et des revers, a évolué depuis les années 1950 avec les travaux d'*Alan Turing* jusqu'à l'ère actuelle du Deep Learning.

L'IA inclut l'apprentissage automatique (machine learning) et l'apprentissage profond (deep learning), utilisant des méthodes comme les réseaux de neurones et le traitement du langage naturel pour accomplir des tâches cognitives complexes.

Les types d'intelligence artificielle peuvent être catégorisés en trois grandes catégories : l'IA faible, l'IA forte et la super intelligence artificielle (ASI) [6], présentés dans le tableau suivant :

Type d'IA	Description	Capacités	Exemples
IA faible (ANI)	Se concentre sur des tâches spécifiques et limitées	Excellent dans des domaines précis, sans capacité de généralisation	Reconnaissance faciale dans les systèmes de sécurité
IA forte (AGI)	Vise à reproduire l'intelligence humaine globale	Capable de résoudre divers problèmes cognitifs	AlphaZero de DeepMind maîtrisant des jeux complexes comme les échecs et le go
Super intelligence artificielle (ASI)	Hypothétique intelligence surpassant celle des humains dans tous les domaines	Raisonnement et résolution de problèmes supérieurs, apprentissage automatique, adaptation, compréhension du langage naturel, création et innovation	Concept hypothétique estimé possible dans les prochaines décennies

TABLE 1.1 – Les types d'intelligence artificielle

L'intelligence artificielle a de nombreuses applications pratiques dans divers domaines :

- Santé : diagnostic médical, analyse d'images, développement de médicaments.
- Finance : détection de fraudes, trading algorithmique, service à la clientèle automatisé.
- Transport : véhicules autonomes, gestion du trafic, optimisation des itinéraires.
- Éducation : tutoriels personnalisés, analyse des performances des étudiants.
- Service à la clientèle : chatbots, assistants virtuels.
- Marketing : publicité ciblée, recommandation de produits.
- Manufacture : maintenance prédictive, robotique industrielle.
- Divertissement : recommandation de contenu, création assistée par IA.
- Sécurité : reconnaissance faciale, cybersécurité.
- Agriculture : agriculture de précision, gestion des ressources.

## 1.3 Apprentissage Automatique (AA)

Le *Machine Learning* [3], également connu sous le nom d'apprentissage automatique, est une méthode où les ordinateurs apprennent à reconnaître des modèles, à anticiper des résultats et à déduire des conclusions à partir de données antérieures sans nécessiter de codage direct. C'est une discipline de l'intelligence artificielle qui utilise des algorithmes pour analyser divers types de données comme les nombres, le texte et les images. En permettant aux machines d'interpréter les données rapidement et précisément, le machine learning facilite la compréhension et l'application des connaissances dans des contextes variés.

Les types d'apprentissage en machine learning peuvent être regroupés en plusieurs catégories principales :

### 1.3.1 Apprentissage supervisé

Le modèle est entraîné sur un ensemble de données où chaque exemple est associé à une sortie connue. L'objectif est de prédire ces sorties pour de nouvelles données en se basant sur les exemples d'entraînement disponibles[8]. Ci-dessous, nous allons définir chaque méthode d'apprentissage, en soulignant les algorithmes couramment utilisés et les approches pour les appliquer efficacement.

## Réseaux neuronaux

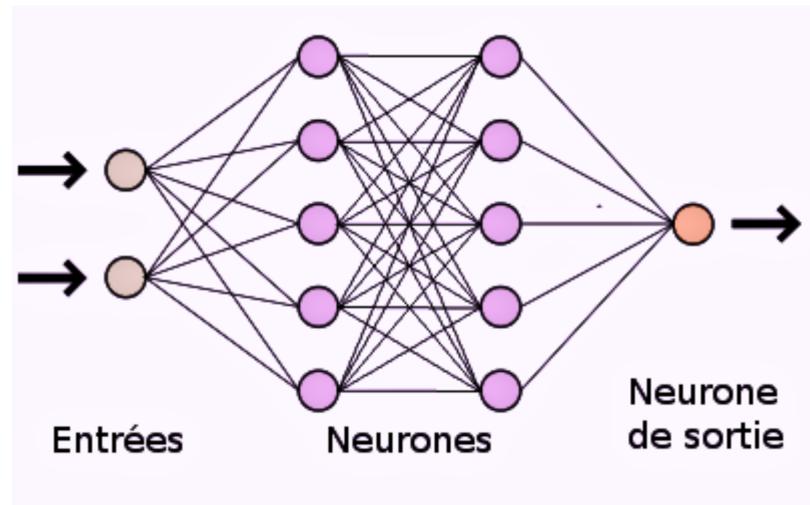


FIGURE 1.1 – Réseaux neuronaux

Des modèles inspirés du cerveau humain qui utilisent des couches de nœuds interconnectés pour apprendre à partir des données en ajustant les poids entre les nœuds.

## Naïve Bayes

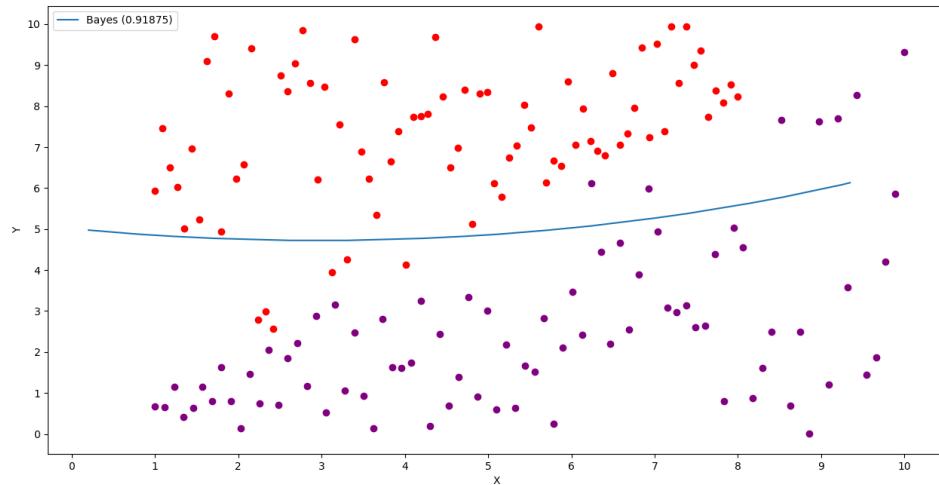


FIGURE 1.2 – Naïve Bayes

Un modèle de classification basé sur le théorème de Bayes, supposant que les caractéristiques sont indépendantes les unes des autres pour estimer les probabilités de classification.

$$P(C_k | x) = \frac{P(x | C_k) \cdot P(C_k)}{P(x)}$$

avec  $P(C_k | x)$  : Probabilité conditionnelle de la classe  $C_k$  sachant les caractéristiques  $x$ ,  $P(x | C_k)$  : Probabilité conditionnelle des caractéristiques  $x$  sachant la classe  $C_k$ ,  $P(C_k)$  : Probabilité a priori de la classe  $C_k$  et  $P(x)$  : Probabilité marginale des caractéristiques  $x$ .

## Régression linéaire

Identifie la relation linéaire entre une variable dépendante et une ou plusieurs variables indépendantes pour faire des prédictions continues. La formule de la régression linéaire simple est donnée par :

$$Y = \beta_0 + \beta_1 X_1 + \epsilon$$

où  $Y$  est la variable dépendante,  $X_1$  est la variable indépendante,  $\beta_0$  est l'intercept,  $\beta_1$  est la pente et  $\epsilon$  est l'erreur.

## Régression logistique

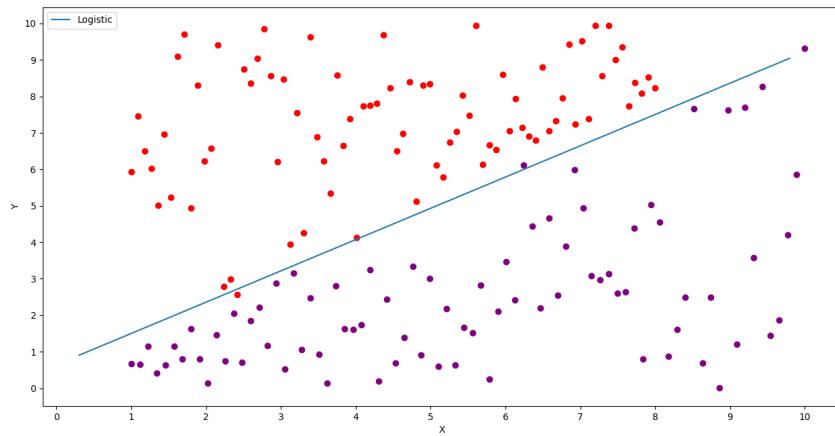


FIGURE 1.3 – Régression logistique

Utilisée pour prédire une variable binaire en trouvant la relation logistique entre les variables d'entrée. La fonction logistique utilisée dans la régression logistique est :

$$p(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1)}}$$

où  $p(X)$  est la probabilité d'appartenir à la classe positive,  $X_1$  est la variable indépendante,  $\beta_0$  est l'intercept, et  $\beta_1$  est la pente.

## Machines à vecteurs de support (SVM)

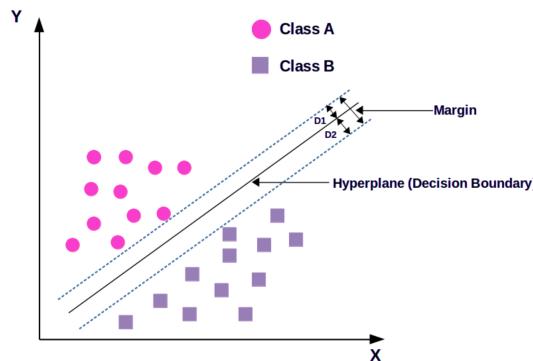


FIGURE 1.4 – Machines à vecteurs de support

Un modèle qui crée un hyperplan pour séparer les classes de données avec une marge maximale. L'hyperplan de séparation pour SVM linéaire est :

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p = 0$$

où  $X_1, X_2, \dots, X_p$  sont les variables d'entrée,  $\beta_0, \beta_1, \beta_2, \dots, \beta_p$  sont les coefficients du modèle.

### K plus proches voisins (kNN)

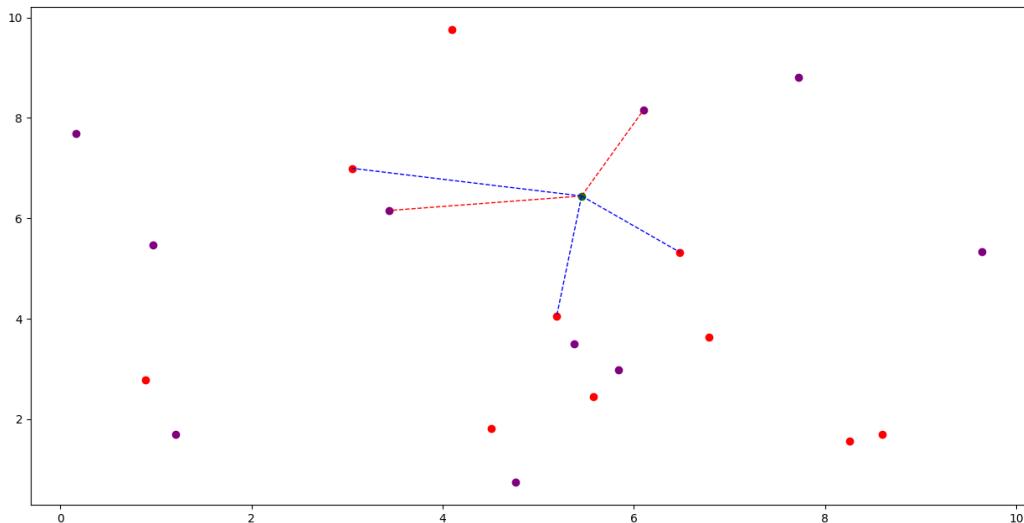


FIGURE 1.5 – Knn

Classifie les données en fonction de la proximité avec d'autres données, en supposant que des données similaires sont proches les unes des autres. La formule de la distance euclidienne utilisée dans kNN est :

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

où  $x$  et  $x'$  sont deux points dans l'espace des caractéristiques, et  $x_i, x'_i$  sont les valeurs de la  $i$ -ème caractéristique.

### Forêt d'arbres décisionnels

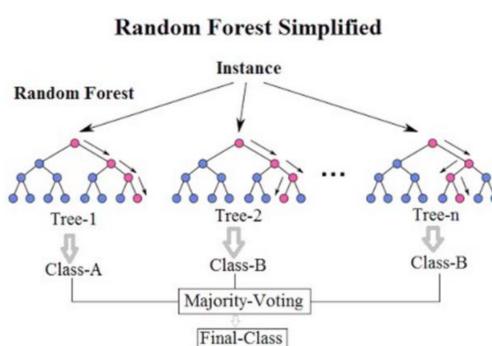


FIGURE 1.6 – Forêt d'arbres décisionnels

Un ensemble d'arbres de décision combinés pour améliorer la précision des prédictions en réduisant la variance. La prédiction pour une forêt d'arbres décisionnels est donnée par la moyenne ou le vote des prédictions individuelles des arbres. Bien sûr ! Voici la définition de la forêt d'arbres décisionnels (Random Forest) avec sa formulation mathématique :

La prédiction d'une forêt aléatoire pour une observation  $x$  est obtenue par la moyenne ou le vote des prédictions de chaque arbre individuel  $T_i$  dans la forêt :

$$\hat{Y}_{\text{RF}}(x) = \frac{1}{N} \sum_{i=1}^N T_i(x)$$

où  $N$  est le nombre d'arbres dans la forêt. Chaque arbre  $T_i$  est construit de manière à maximiser la différence de performance entre les classes ou les groupes d'observations.

### 1.3.2 Apprentissage non supervisé

Le modèle est entraîné sur des données non étiquetées et cherche à découvrir des structures ou des motifs intrinsèques dans les données. Il est utilisé pour regrouper les données en clusters ou pour réduire la dimensionnalité des données. Il existe plusieurs types d'algorithme utilisés en apprentissage non supervisé[1] :

#### Méthode des K-moyennes

La méthode des K-moyennes partitionne un ensemble de données en  $K$  clusters en minimisant la variance intra-cluster. L'algorithme cherche à minimiser la somme des carrés des distances euclidiennes des points au centre de leur cluster ( $\mu_i$ ) :

$$\min_S \sum_{i=1}^K \sum_{x \in S_i} \|x - \mu_i\|^2$$

#### Algorithme apriori

L'algorithme apriori est utilisé pour l'extraction de règles d'association dans des ensembles de données transactionnelles, identifiant des associations fréquentes entre les items.

#### Clustering hiérarchique

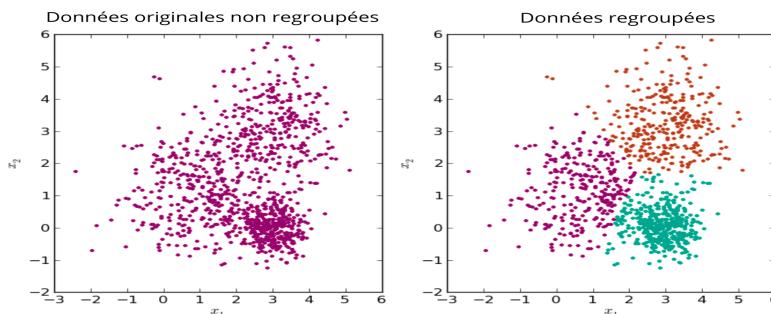


FIGURE 1.7 – Clustering

Le clustering hiérarchique construit une hiérarchie de clusters en fusionnant successivement les clusters les plus proches ou en divisant les clusters existants jusqu'à l'obtention d'une structure de clustering complète.

### Décomposition en valeurs singulières (SVD)

La décomposition en valeurs singulières factorise une matrice  $X$  de taille  $m \times n$  en trois matrices de plus petite taille :  $U$ ,  $\Sigma$ , et  $V^T$ . Elle est souvent utilisée pour réduire la dimensionnalité des données en conservant les structures sous-jacentes importantes :

$$X = U\Sigma V^T$$

### Analyse en composantes principales (ACP)

L'analyse en composantes principales est une technique statistique qui transforme des variables corrélées en un ensemble de variables non corrélées appelées composantes principales, réduisant ainsi la dimensionnalité des données tout en préservant leur variance maximale.

### 1.3.3 Apprentissage par renforcement

Le modèle apprend par interaction avec un environnement dynamique. Il reçoit des récompenses ou des punitions pour ses actions et ajuste ses stratégies pour maximiser les récompenses sur le long terme. Voici les algorithmes couramment utilisés pour ce type d'apprentissage :

**Q-Learning** : Un algorithme de reinforcement learning qui apprend une fonction d'action-valeur optimale en explorant l'environnement basé sur des récompenses.

**SARSA (State-Action-Reward-State-Action)** : Un autre algorithme de reinforcement learning similaire à Q-Learning mais qui prend en compte les actions réelles prises par l'agent.

### 1.3.4 Apprentissage semi-supervisé

Ce type d'apprentissage combine des données étiquetées et non étiquetées pour l'entraînement. Il peut améliorer les performances du modèle en utilisant à la fois des données annotées et non annotées.

**Méthodes de propagation de label** : Utilisation de graphes pour propager les étiquettes des données étiquetées aux données non étiquetées en fonction de leur similarité.

**Réseaux neuronaux semi-supervisés** : Des réseaux neuronaux utilisant à la fois des données étiquetées et non étiquetées pour améliorer les performances de la tâche.

### 1.3.5 Apprentissage par transfert

Ici, les connaissances acquises lors de l'entraînement sur une tâche sont transférées et utilisées pour améliorer les performances sur une tâche similaire mais différente.

**Réseaux neuronaux pré-entraînés (GPT, BERT)** : Des modèles de langage pré-entraînés sur de grandes quantités de données textuelles et réutilisés pour des tâches spécifiques.

**Fine-tuning de modèles pré-entraînés** : Ajustement des poids d'un modèle pré-entraîné sur des données spécifiques à la tâche.

**Méthodes de transfert de connaissances basées sur les caractéristiques** : Utilisation de caractéristiques apprises sur une tâche pour aider à résoudre une autre tâche similaire.

**Méthodes de transfert de connaissances basées sur les modèles** : Utilisation d'un modèle entier pré-entraîné comme point de départ pour une tâche connexe.

### 1.3.6 Processus de l'Apprentissage Automatique

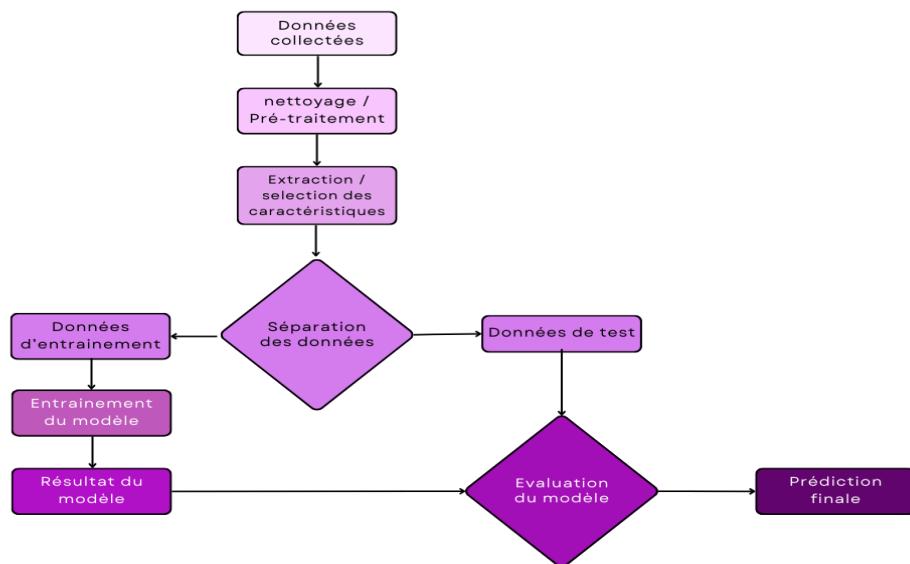


FIGURE 1.8 – Processus de l'Apprentissage Automatique

### Collecte et pré-traitement des Données

La collecte et le pré-traitement des données impliquent la récupération des données pertinentes pour résoudre un problème spécifique, suivie de leur nettoyage, transformation et préparation pour les rendre utilisables par les algorithmes d'apprentissage automatique.

Cette étape est cruciale car la qualité des données en entrée affecte directement la performance du modèle. La collecte inclut souvent l'agrégation de données à partir de différentes sources, tandis que le pré-traitement implique le nettoyage des données pour éliminer les valeurs aberrantes, remplir les valeurs manquantes, normaliser les données si nécessaire, et transformer les données brutes en un format adapté à l'entraînement des modèles.

### Extraction des Caractéristiques

L'extraction des caractéristiques consiste à identifier et à extraire les aspects significatifs des données qui peuvent être utilisés comme entrées pour l'apprentissage automatique. Voici quelques techniques couramment utilisées :

Technique	Description
Analyse en composantes principales (PCA)	Réduction de la dimensionnalité en préservant au maximum la variance des données.
Transformations non linéaires (Kernelization)	Projection des données dans un espace de caractéristiques non linéaire.

## Sélection des Caractéristiques

La sélection des caractéristiques vise à choisir les caractéristiques les plus pertinentes parmi toutes celles extraites, afin d'améliorer les performances du modèle et de réduire la complexité. Voici plusieurs approches couramment utilisées :

Technique	Description
Analyse discriminante linéaire (LDA)	Trouve une combinaison linéaire des caractéristiques qui sépare au mieux les classes.
Sélection de caractéristiques basée sur les modèles	Utilisation de modèles pour évaluer l'importance des caractéristiques.
Sélection de caractéristiques basée sur les filtres	Utilisation de mesures statistiques pour évaluer l'importance des caractéristiques.
Sélection de caractéristiques récursives (RFE)	Élimination itérative des caractéristiques moins importantes jusqu'à une sélection optimale.

Ces techniques permettent de maximiser l'efficacité des modèles d'apprentissage automatique en se concentrant sur les caractéristiques les plus informatives et en réduisant la dimensionnalité lorsque cela est nécessaire.

## Séparation des Données

La séparation des données est le processus de division des données disponibles en ensembles distincts pour l'entraînement, la validation et le test des modèles.

Cette étape vise à évaluer la capacité du modèle à généraliser sur de nouvelles données qu'il n'a pas encore vues. Les données sont typiquement divisées en trois ensembles : l'ensemble d'entraînement, utilisé pour ajuster les paramètres du modèle ; l'ensemble de validation, utilisé pour ajuster les hyperparamètres et évaluer les performances du modèle pendant l'entraînement ; et l'ensemble de test, réservé pour évaluer la performance finale du modèle après l'entraînement.

## Entraînement du modèle

L'entraînement du modèle consiste à ajuster les paramètres du modèle en utilisant les données d'entraînement afin qu'il puisse faire des prédictions ou des classifications sur de nouvelles données.

Durant cette étape, le modèle utilise les données d'entraînement pour apprendre à partir

des exemples fournis. En fonction du type de modèle (supervisé, non supervisé, ou autre), des algorithmes spécifiques sont utilisés pour minimiser l'erreur ou optimiser une fonction objectif définie. L'objectif est de trouver le meilleur ensemble de paramètres qui généralisera bien sur de nouvelles données.

## Évaluation du modèle

L'évaluation du modèle consiste à mesurer les performances du modèle entraîné en utilisant des critères pertinents, tels que l'exactitude, la précision, le rappel, ou d'autres mesures spécifiques au problème.

Cette dernière étape est cruciale pour déterminer si le modèle répond aux exigences de performance souhaitées. Elle utilise généralement l'ensemble de validation ou l'ensemble de test pour évaluer comment le modèle se comporte sur des données qu'il n'a pas utilisées pendant l'entraînement. Les résultats de l'évaluation aident à ajuster le modèle si nécessaire, à comparer différentes approches ou à décider de son déploiement.

## 1.4 Apprentissage Profond (AP)

### 1.4.1 Définition et Concept de Base

L'apprentissage profond, une branche de l'intelligence artificielle, utilise des réseaux de neurones pour analyser de grandes quantités de données. Inspiré par le cerveau humain, ces réseaux transforment les données à travers plusieurs couches pour découvrir des motifs et faire des prédictions précises. Les neurones artificiels, comme les blocs de construction, calculent et ajustent les informations à chaque étape, permettant aux machines d'apprendre et de comprendre comme les humains, mais avec des calculs mathématiques plutôt que des signaux biologiques[4].

### 1.4.2 Types de Réseaux de Neurones

Les réseaux de neurones profonds comprennent diverses architectures [5], telles que :

#### Réseaux neuronaux convolutifs (CNN)

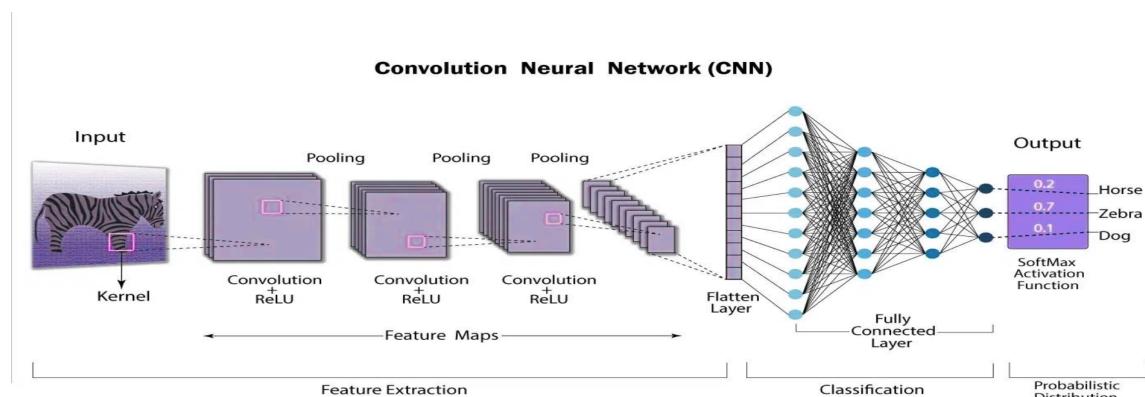


FIGURE 1.9 – Réseaux neuronaux convolutifs

Les CNN sont particulièrement efficaces pour l'analyse d'images, mais peuvent également être appliqués à des séquences de texte comme les e-mails. Ils sont utilisés pour extraire des caractéristiques locales et spatiales à différentes échelles dans les données textuelles, ce qui est utile pour détecter des motifs dans les parties du texte telles que les en-têtes, les signatures et les contenus des e-mails.[27]

### Réseaux neuronaux récurrents (RNN)

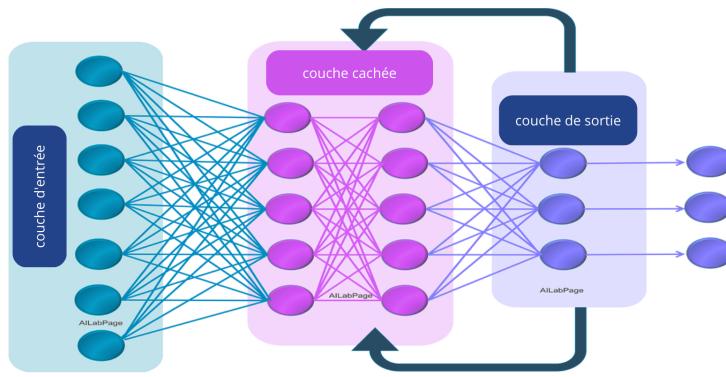


FIGURE 1.10 – Réseaux neuronaux récurrents

possèdent des connexions cycliques qui permettent de mémoriser des informations antérieures, adaptés notamment pour des tâches comme le sous-titrage d'images, le traitement du langage naturel et la traduction automatique.

### Réseaux de fonction de base radiale (RBFN)

sont des réseaux neuronaux feedforward utilisant des fonctions de base radiales comme activations, souvent employés pour la classification, la prédiction de séries temporelles et la régression linéaire.

### Réseaux de mémoire à long et court terme (LSTM)

une forme évoluée des RNN, sont capables de mémoriser des dépendances sur de longues périodes, utiles pour prédire des séries chronologiques et d'autres applications comme la composition musicale et la reconnaissance vocale.

### Réseaux adversariaux génératifs (GAN)

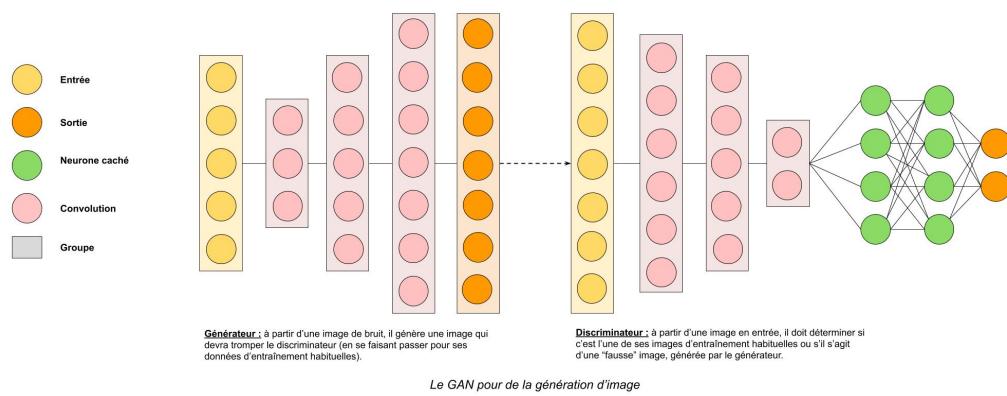


FIGURE 1.11 – Réseaux adversariaux génératifs [2]

génèrent de nouvelles données similaires à celles d'un ensemble d'apprentissage en utilisant un générateur et un discriminateur pour créer et évaluer ces données, souvent utilisés dans l'amélioration des textures 2D pour les jeux vidéo.

### Machines de Boltzmann restreintes (RBM)

sont des réseaux neuronaux stochastiques développés par Geoffrey Hinton, utilisés pour apprendre à partir d'une distribution de probabilité sur un ensemble d'entrées, composés de deux couches : unités visibles et unités cachées.

## 1.5 Traitement Automatique du Langage Naturel (TALN)

### 1.5.1 Définition et Objectifs

*Le traitement du langage naturel* (NLP) est une discipline qui porte essentiellement sur la compréhension, la manipulation et la génération du langage naturel par les machines[18]. Son objectif principal est de permettre aux machines de comprendre, d'interpréter et de générer du langage humain de manière efficace et précise.

### 1.5.2 Techniques et Méthodes

Pour atteindre ses objectifs, le TALN utilise diverses techniques et méthodes :

- **Tokenisation et Segmentation** : Diviser le texte en unités linguistiques telles que mots ou phrases.
- **Analyse Syntaxique** : Identifier la structure grammaticale des phrases pour comprendre les relations entre les mots.
- **Reconnaissance d'Entités Nommées (NER)** : Identifier et classer des entités telles que noms de personnes, lieux, dates, etc.
- **Analyse sémantique** : Comprendre le sens des phrases et des documents en examinant les relations entre les mots et les concepts.

- **Traitements du Langage Naturel Génératif (NLG)** : Générer du texte de manière automatique, souvent utilisé dans les systèmes de dialogue et de résumé automatique.
- **Modèles de Langage** : Utilisation de modèles statistiques et d'apprentissage profond pour capturer et générer du langage naturel.

### 1.5.3 Techniques de Représentation

Pour la représentation et l'analyse du texte, plusieurs techniques sont couramment utilisées :

- **Bag-of-Words (BoW)** : La méthode BoW crée des vecteurs en comptant la fréquence de chaque mot dans un document, sans tenir compte de l'ordre des mots.
- **TF-IDF (Term Frequency-Inverse Document Frequency)** : La méthode TF-IDF évalue l'importance d'un mot dans un document par rapport à un corpus, pondérant les mots en fonction de leur fréquence dans le document et leur rareté dans le corpus.
- **Word2Vec** : Utilise des réseaux de neurones pour apprendre des vecteurs de mots qui capturent des relations sémantiques. Les modèles les plus courants sont Continuous Bag of Words (CBOW) et Skip-Gram.
- **GloVe (Global Vectors for Word Representation)** : Apprend des représentations de mots en utilisant des statistiques globales des co-occurrences de mots dans un corpus.

## 1.6 Évaluation des performances

L'évaluation des performances des modèles de détection de phishing est cruciale pour mesurer leur efficacité et leur robustesse dans la classification des e-mails comme légitimes ou malveillants.

### 1.6.1 Métriques d'évaluation

Les métriques d'évaluation jouent un rôle crucial dans l'analyse de la performance des modèles de détection de phishing. Elles fournissent des mesures quantitatives et qualitatives pour évaluer à la fois l'exactitude et la capacité du modèle à identifier correctement les e-mails légitimes et malveillants. Voici quelques métriques couramment utilisées :

- **Matrices de confusion** : Tableaux qui montrent la performance d'un modèle de classification en comparant les prédictions avec les vraies valeurs. Elles fournissent des informations détaillées sur les erreurs de classification, telles que les faux positifs, les faux négatifs, les vrais positifs et les vrais négatifs. Voici la structure d'une matrice de confusion :

	Prédit Positif	Prédit Négatif
Réel Positif	TP	FN
Réel Négatif	FP	TN

TABLE 1.2 – Matrice de confusion

Avec :

**TP (True Positives)** : Les exemples positifs correctement classés comme positifs.

**TN (True Negatives)** : Les exemples négatifs correctement classés comme négatifs.

**FP (False Positives)** : Les exemples négatifs incorrectement classés comme positifs.

**FN (False Negatives)** : Les exemples positifs incorrectement classés comme négatifs.

- **Précision** : La précision mesure la proportion d'e-mails identifiés comme malveillants qui le sont réellement parmi tous les e-mails identifiés comme malveillants par le modèle. Elle est définie par la formule suivante :

$$\text{Précision} = \frac{TP}{TP + FP}$$

Une précision élevée indique que le modèle produit peu de faux positifs parmi les prédictions positives, garantissant ainsi que la plupart des e-mails identifiés comme malveillants le sont réellement.

- **Exactitude (Accuracy)** : La proportion des prédictions correctes (vrais positifs et vrais négatifs) parmi l'ensemble des prédictions. Elle est définie par la formule suivante :

$$\text{Exactitude} = \frac{TP + TN}{TP + TN + FP + FN}$$

Une exactitude élevée indique que le modèle classe correctement une grande proportion des e-mails, qu'ils soient légitimes ou malveillants.

- **Rappel** : Le rappel mesure la sensibilité du modèle à détecter tous les e-mails malveillants présents dans l'ensemble de données. Il est calculé comme le rapport des vrais positifs sur l'ensemble des instances réellement malveillantes. Elle est définie par :

$$\text{Rappel} = \frac{TP}{TP + FN}$$

Un rappel élevé signifie que le modèle identifie efficacement la majorité des e-mails malveillants présents.

- **F-score** : Moyenne harmonique de la précision et du rappel, utilisée pour équilibrer les

deux métriques. Elle est définie par :

$$F1 = 2 \cdot \frac{\text{Précision} \cdot \text{Rappel}}{\text{Précision} + \text{Rappel}}$$

Un F-score élevé indique un bon équilibre entre la précision et le rappel, ce qui signifie que le modèle est efficace à la fois pour identifier correctement les e-mails malveillants (précision) et pour détecter la majorité des e-mails malveillants présents (rappel).

- **Courbe ROC et AUC (Area Under the Curve)** : La courbe ROC (Receiver Operating Characteristic) trace le taux de vrais positifs (TPR) contre le taux de faux positifs (FPR) à différents seuils de classification.

Le TPR est le rappel, et le FPR est défini par :

$$\text{FPR} = \frac{FP}{FP + TN}$$

L'aire sous la courbe (AUC) représente la probabilité que le modèle classe correctement une paire aléatoire d'instances positives et négatives. Une AUC de 1 indique un modèle parfait, tandis qu'une AUC de 0,5 indique un modèle sans discrimination (équivalent à un tirage au sort).

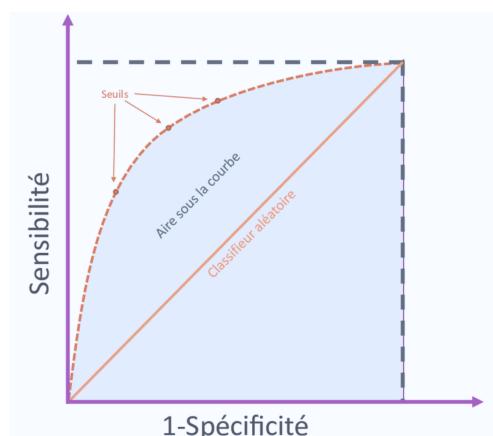


FIGURE 1.12 – Exemple de courbe ROC avec l'AUC

Ces métriques permettent de quantifier la performance du modèle à différentes étapes du processus de classification et sont essentielles pour ajuster les algorithmes et optimiser les paramètres.

## 1.6.2 Techniques de validation

Les techniques de validation garantissent la robustesse et la généralisabilité des modèles de détection de phishing par e-mail, assurant leur capacité à fonctionner efficacement dans des environnements réels.

### 1. Séparation des ensembles d'entraînement, de validation et de test :

La séparation des jeux de données en ensembles d'entraînement, de validation et de test

est une autre technique couramment utilisée pour évaluer la performance des modèles. Cette approche implique les étapes suivantes :

- **Ensemble d'entraînement** : Cet ensemble est utilisé pour ajuster les paramètres du modèle en exposant celui-ci à des exemples étiquetés. Une plus grande partie des données est généralement allouée à l'ensemble d'entraînement afin de permettre au modèle d'apprendre les motifs et les relations présents dans les données.
- **Ensemble de validation** : Une fois que le modèle est entraîné sur l'ensemble d'entraînement, l'ensemble de validation est utilisé pour ajuster les hyperparamètres et pour évaluer la performance du modèle pendant le processus d'entraînement. Cela permet de choisir les meilleures configurations de modèle et d'optimiser ses performances sans introduire de biais de sélection de modèle.
- **Ensemble de test** : Après avoir finalisé le processus d'entraînement et de validation, l'ensemble de test est utilisé pour évaluer objectivement la performance finale du modèle. Cet ensemble est crucial car il fournit une estimation impartiale de la capacité du modèle à généraliser à de nouvelles données non vues auparavant, simulant ainsi son comportement dans des conditions réelles.

Cette division en trois ensembles distincts permet non seulement d'évaluer la performance du modèle de manière rigoureuse, mais aussi de détecter tout signe de surapprentissage (overfitting), où le modèle pourrait trop s'adapter aux détails spécifiques des données d'entraînement au détriment de sa capacité à généraliser. En répartissant judicieusement les données entre ces ensembles, les praticiens de l'apprentissage automatique peuvent garantir que leurs modèles de détection de phishing sont robustes et fiables dans des scénarios réels.

### 2. Validation croisée :

La validation croisée est une technique essentielle pour évaluer la performance d'un modèle de manière plus fiable et pour minimiser le risque de surapprentissage. Parmi les variantes de cette méthode, les suivantes sont particulièrement notables :

- **Validation croisée k-fold** : Les données sont divisées en  $k$  sous-ensembles de taille égale. Le modèle est entraîné sur  $k-1$  sous-ensembles et testé sur le sous-ensemble restant. Ce processus est répété  $k$  fois, chaque sous-ensemble étant utilisé exactement une fois comme ensemble de test.
- **Validation croisée leave-one-out (LOO)** : Chaque observation est utilisée comme ensemble de test une fois, tandis que toutes les autres observations servent d'ensemble d'entraînement. Cela garantit une évaluation exhaustive mais peut être coûteux en termes de temps de calcul pour de grands ensembles de données.
- **Stratified k-fold validation croisée** : Une amélioration de la méthode k-fold où chaque fold contient approximativement la même proportion de classes que l'ensemble de données initial. Cela est particulièrement utile pour les jeux de données déséquilibrés, car cela assure que chaque fold est représentatif de la distribution des classes dans l'ensemble complet, ce qui conduit à une évaluation plus fiable du modèle.

## 1.7 Phishing par e-mail

### 1.7.1 Définition et Principes de Base

#### Définition du Phishing par e-mail

Le phishing par e-mail est une forme d'attaque cybernétique où les attaquants envoient des messages électroniques frauduleux qui semblent provenir d'entités légitimes, telles que des banques, des entreprises ou des services en ligne. L'objectif est d'inciter les destinataires à divulguer des informations personnelles ou financières sensibles, telles que des mots de passe, des numéros de carte de crédit, ou à télécharger des pièces jointes malveillantes.

#### Principes de Base du Phishing par e-mail

Le phishing exploite souvent des techniques d'ingénierie sociale pour induire les victimes à agir rapidement et de manière irréfléchie. Les e-mails de phishing peuvent contenir des éléments persuasifs tels que des appâts attrayants (promotions, cadeaux) ou des menaces (fermeture de compte, sécurité compromise) pour inciter les utilisateurs à divulguer des informations confidentielles ou à cliquer sur des liens malveillants. Les attaques peuvent viser à voler des identifiants, à compromettre des comptes en ligne, à détourner des fonds ou à installer des logiciels malveillants sur les systèmes des victimes.

#### Statistiques sur le Phishing par e-mail

Selon le rapport annuel sur la cybercriminalité de l'année précédente, les attaques de phishing par e-mail ont représenté plus de 80% des incidents de sécurité signalés. Environ 1 personne sur 4 ciblée par le phishing parvient à se faire duper, entraînant des pertes financières importantes pour les individus et les organisations. Les secteurs les plus visés incluent les services financiers, le commerce électronique et les services en ligne.

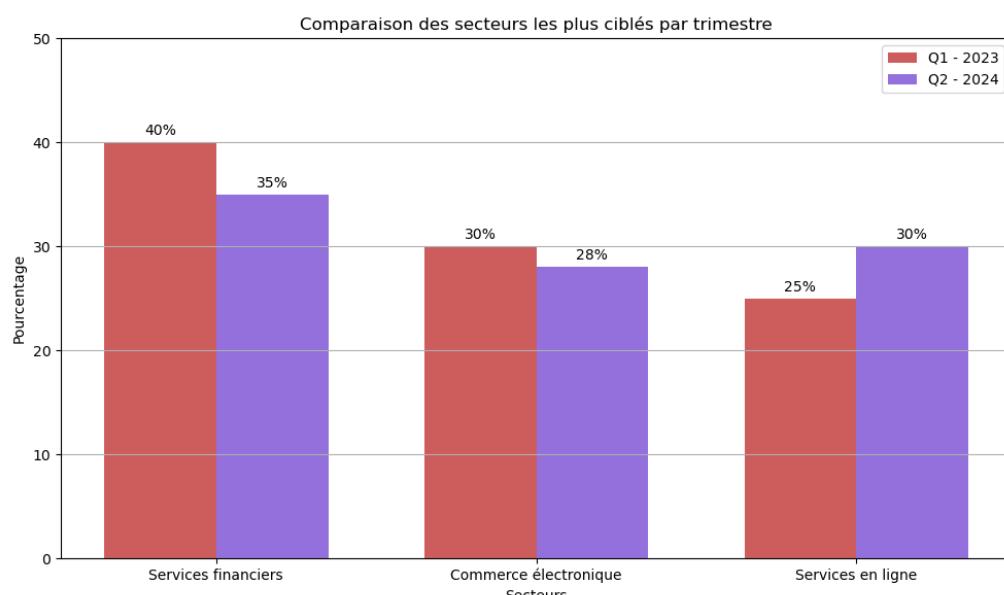


FIGURE 1.13 – Comparaison des secteurs les plus ciblés par trimestre

### 1.7.2 Victimes Ciblées

Les victimes de phishing par e-mail peuvent varier largement, mais certaines caractéristiques démographiques et comportementales peuvent augmenter la susceptibilité aux attaques :

- **Âge** : Les jeunes adultes, en particulier ceux âgés de 18 à 25 ans, sont souvent plus vulnérables aux attaques de phishing en raison de leur usage intensif des technologies numériques [34, 28].
- **Genre** : Des études montrent que les femmes peuvent être plus susceptibles de tomber dans le piège du phishing que les hommes [34, 26, 29].
- **Éducation Anti-phishing** : Les individus sans formation spécifique en matière de cybersécurité sont plus exposés au phishing [28].
- **Éducation Générale** : Les personnes sans formation en informatique peuvent être plus facilement trompées par des attaques de phishing.
- **Comportement en ligne** : Les utilisateurs qui effectuent fréquemment des achats en ligne ou des opérations bancaires en ligne sont des cibles privilégiées [34, 26].

### Caractéristiques des Victimes de Phishing

Facteur	Susceptibilité Élevée	Susceptibilité Faible
Âge	18-25 ans	Plus de 25 ans
Genre	Femmes	Hommes
Formation Anti-phishing	Pas de formation	Formation anti-phishing
Éducation Générale	Non-informaticiens	Informaticiens
Comportement en ligne	Achats et opérations bancaires en ligne	Utilisation simple du web et des emails

TABLE 1.3 – Caractéristiques des victimes de phishing par e-mail

### 1.7.3 Techniques utilisées dans le Phishing

Le phishing par e-mail utilise plusieurs techniques pour tromper les utilisateurs, telles que :

- **Phishing de spearphishing** : ciblage spécifique des individus ou des organisations avec des informations personnelles précises.
- **Hameçonnage** : utilisation de faux sites web pour collecter des informations d'identification.
- **Pharming** : redirection du trafic web vers des sites web frauduleux sans le consentement des utilisateurs.
- **Malware** : utilisation de logiciels malveillants pour voler des informations sensibles.

### **1.7.4 Méthodes de Détection**

Les méthodes de détection du phishing par e-mail incluent :

- Analyse des URL et des liens inclus dans les e-mails.
- Analyse du contenu et de la syntaxe des e-mails pour détecter des signes de fraude.
- Utilisation de filtres anti-spam et anti-phishing pour bloquer les e-mails suspects.
- Formation des utilisateurs pour reconnaître les techniques de phishing et éviter les pièges.

## **1.8 Conclusion**

Dans cette étude, nous avons exploré divers aspects de l'intelligence artificielle, du machine learning, du deep learning et du traitement du langage naturel. Ces technologies révolutionnent nos capacités à analyser des données, à automatiser des tâches complexes et à améliorer la sécurité numérique. L'accent mis sur l'évaluation des performances des modèles et la lutte contre le phishing par email met en lumière des défis persistants mais surmontables grâce à l'innovation continue dans ces domaines.

En conclusion, cette étude offre une base solide pour orienter les prochaines étapes de la recherche et du développement technologique visant à renforcer la résilience contre les menaces numériques, tout en soulignant l'importance de l'adaptabilité et de l'innovation continue dans la lutte contre le phishing et les attaques cybernétiques en général.

# Chapitre 2

## Revue de littérature

### 2.1 Introduction

L'objectif de cette revue de littérature est d'explorer et d'analyser les recherches existantes sur la détection de phishing par e-mail en utilisant des techniques avancées telles que le machine learning (ML), le traitement du langage naturel (NLP) et le deep learning (DL). En comprenant les méthodes actuelles, les tendances et les défis, nous visons à identifier les lacunes et à orienter les futurs travaux de recherche. Cette analyse est cruciale pour développer des solutions plus efficaces contre le phishing par e-mail, en s'appuyant sur un cadre théorique solide et les meilleures pratiques issues des études précédentes. La revue est structurée en plusieurs sections : d'abord, nous présentons les approches ML, en comparant les différents algorithmes et leurs performances dans la détection de phishing ; ensuite, nous abordons les techniques de NLP utilisées pour analyser le contenu des e-mails et identifier les signaux de phishing ; nous discutons également des modèles de DL, en mettant en lumière leur capacité à détecter des schémas complexes ; enfin, nous effectuons une comparaison des différentes méthodes pour fournir une vue d'ensemble complète et détaillée de l'état de la recherche dans ce domaine. Cette structure permet de couvrir de manière exhaustive les avancées technologiques et de fournir des recommandations pour les futures recherches.

### 2.2 Revue des techniques de détection de phishing par e-mail

Les algorithmes d'apprentissage automatique sont cruciaux pour détecter les e-mails de phishing. En utilisant l'apprentissage supervisé, les modèles sont entraînés sur des données étiquetées pour identifier les e-mails frauduleux. Les techniques de traitement automatique du langage naturel (NLP), d'apprentissage automatique (ML) et d'apprentissage profond (DL) analysent le contenu des e-mails, les entêtes et les liens pour distinguer les tentatives de phishing des e-mails légitimes. Ainsi, une combinaison de ces approches permet une détection proactive et efficace des attaques de phishing.

### 2.2.1 Techniques basées sur le traitement automatique du langage naturel (NLP)

L'étude [21] propose un modèle de classification des e-mails de phishing basé sur la découverte des connaissances (KD) et l'exploration de données. Ce modèle vise à construire un classificateur intelligent capable de distinguer les messages électroniques légitimes des spams en extrayant des caractéristiques utiles d'un ensemble de données d'entraînement. Les chercheurs utilisent des techniques de traitement linguistique et des ontologies pour améliorer la similarité entre les e-mails partageant des termes sémantiques similaires. Le principe de fréquence des termes dans les documents est également appliqué pour pondérer les termes de hameçonnage dans chaque e-mail. Le modèle proposé réduit le nombre de fonctionnalités utilisées dans le processus de classification à seulement 16, ce qui améliore les performances et l'efficacité de la classification tout en réduisant le bruit et en augmentant la précision.

Le processus de découverte des connaissances commence par la compréhension du problème et des données, suivi d'une phase de prétraitement des données. L'architecture du modèle inclut la collecte de données à partir d'un ensemble d'e-mails réels, comprenant à la fois des e-mails de phishing et légitimes. Les e-mails sont transformés dans un format compatible avec le package de courrier Java, et un ensemble de 16 fonctionnalités est extrait des en-têtes et des corps des e-mails.

La pondération des termes de hameçonnage est effectuée en identifiant les termes les plus fréquents dans les e-mails de phishing et en leur attribuant un poids basé sur leur fréquence. Cette étape, combinée à des techniques de parsing, de tokenization, de stemming et de suppression des mots vides, améliore la précision de la classification.

Pour évaluer le modèle, cinq techniques de classification bien connues sont utilisées : J48, Bayes naïf, SVM, Perceptron multi-couche (MLP) et Forêt aléatoire. Chaque algorithme est brièvement décrit, mettant en évidence ses spécificités. Par exemple, J48 utilise une implémentation Java de l'algorithme C4.5 pour construire un arbre de décision, tandis que SVM cherche à trouver le meilleur hyperplan séparant les classes dans un espace multidimensionnel.

Le modèle proposé démontre une capacité significative à détecter les e-mails de phishing, avec des performances améliorées par rapport aux approches similaires. La comparaison des résultats de précision de différentes approches de classification montre que le modèle, combinant J48, Bayes Net, SVM, Forêt aléatoire et Perceptron multi-couche, atteint une précision de 0.991, surpassant légèrement les autres méthodes.

### 2.2.2 Études sur l'apprentissage automatique et les approches de classification

Dans leur étude[15] sur le hameçonnage par e-mail, Cleber K. Olivoa, Altair O. Santina et Luiz S. Oliveira ont abordé le problème comme une question de reconnaissance de motifs et d'apprentissage automatique. Ils ont examiné en détail les caractéristiques clés du phishing, telles que les hyperliens trompeurs et les e-mails encodés en HTML. Pour entraîner leur modèle de détection, les chercheurs ont utilisé un ensemble de données comprenant 450 messages de phishing étiquetés et 450 e-mails légitimes.

L'algorithme d'apprentissage automatique choisi pour cette tâche est le Support Vector Machine (SVM), connu pour sa capacité à gérer les problèmes de classification à deux classes. Le SVM a été entraîné à détecter les vecteurs de support à partir de la base de données d'entraînement, puis utilisé pour classer les échantillons non étiquetés. Les chercheurs ont exploré plusieurs types de noyaux pour le SVM, dont le noyau gaussien s'est révélé le plus performant.

L'évaluation des performances du modèle de détection s'est appuyée sur les courbes ROC (Receiver Operating Characteristic) et l'AUC (Area Under the Curve). Ces métriques ont permis de mesurer la sensibilité et la spécificité du classificateur, ainsi que sa capacité à discriminer entre les e-mails de phishing et légitimes. Les résultats ont montré une détection efficace du phishing, soulignant ainsi l'importance de l'approche de reconnaissance de motifs et d'apprentissage automatique dans la lutte contre les attaques par e-mail.

Dans cette étude [24], Harikrishnan NB, Vinayakumar R et Soman KP du Center for Computational Engineering and Networking (CEN) à l'Amrita School of Engineering, Coimbatore ont visé à classifier les e-mails en légitimes ou spam à l'aide de techniques de traitement de texte et d'apprentissage automatique. Deux sous-tâches distinctes sont définies : la classification des e-mails avec en-têtes et sans en-têtes. Les ensembles de données utilisés pour l'entraînement et le test sont largement déséquilibrés, ce qui représente un défi supplémentaire pour les modèles de classification.

Pour la représentation des données, l'étude utilise la méthode TF-IDF (Term Frequency-Inverse Document Frequency), qui évalue l'importance des mots dans un corpus. Afin de réduire la dimensionnalité des données et de retenir les informations les plus pertinentes, les techniques SVD (Singular Value Decomposition) et NMF (Non-negative Matrix Factorization) sont appliquées. Ces représentations sont ensuite utilisées pour entraîner divers algorithmes d'apprentissage automatique classiques tels que l'arbre de décision, la forêt aléatoire, AdaBoost, KNN et SVM.

Les résultats montrent que, pour les données d'entraînement, les arbres de décision et les forêts aléatoires présentent les meilleures performances en termes d'exactitude. Cependant, ces modèles montrent des signes de surapprentissage sur les données de test, principalement en raison de la nature déséquilibrée des ensembles de données. Les représentations TF-IDF combinées à SVD ou NMF offrent des performances similaires, démontrant l'efficacité de la réduction de dimensionnalité dans ce contexte.

En conclusion, l'étude démontre l'efficacité des techniques classiques de machine learning pour la détection des e-mails de phishing en utilisant des représentations TF-IDF enrichies par des méthodes de réduction de dimensionnalité. Néanmoins, l'amélioration de la robustesse du modèle pourrait être atteinte en intégrant des sources de données externes et en explorant des approches basées sur l'apprentissage profond, ce qui pourrait potentiellement offrir de meilleures performances dans la gestion des données déséquilibrées et la complexité des motifs de phishing.

Dans cette étude [13], Bergholz, Paaß, Reichartz, Strobel et Chang ont développé de nouvelles caractéristiques basées sur des modèles pour détecter les escroqueries de phishing. Ils ont utilisé la compression dynamique de chaînes de Markov et un modèle de sujet de Dirichlet latent spécifique à la cible pour améliorer la détection. Leurs classificateurs, notamment SVM avec un noyau RBF, ont surpassé les méthodes précédentes. Cette étude met en lumière l'efficacité

des techniques de classification statistique pour distinguer les e-mails légitimes des e-mails de phishing.

Masoumeh Zareapoor et Seeja K. R [38] ont exploré des méthodes pour identifier efficacement les e-mails de hameçonnage en réduisant la complexité des données. Ils ont comparé deux approches principales : les méthodes d'extraction de caractéristiques, telles que l'Analyse en Composantes Principales (PCA) et l'Analyse Sémantique Latente (LSA), avec des techniques de sélection de caractéristiques comme le Chi-Carré et le Gain d'Information (IG). L'objectif était d'améliorer la classification des e-mails en distinguant efficacement les e-mails légitimes des e-mails de hameçonnage. Les résultats ont montré que les méthodes d'extraction de caractéristiques ont mieux fonctionné avec un nombre limité de caractéristiques, simplifiant ainsi les données tout en préservant la performance de la classification. En revanche, les techniques de sélection de caractéristiques ont nécessité un ensemble plus large de caractéristiques pour obtenir des résultats comparables.

Les méthodes d'extraction de caractéristiques, telles que PCA et LSA, ont démontré leur efficacité en réduisant la complexité des données tout en préservant ou améliorant la performance de la classification. PCA, en transformant les caractéristiques originales en un ensemble de composantes principales non corrélées, permet une représentation plus concise des données. De même, LSA, en capturant les relations sémantiques entre les termes dans un corpus de documents, offre une perspective précieuse sur le contenu des e-mails. En revanche, les techniques de sélection de caractéristiques classiques comme le Chi-Carré et le Gain d'Information ont nécessité un ensemble plus vaste de caractéristiques pour obtenir des résultats comparables, ce qui peut augmenter la complexité et la dimensionnalité des modèles.

Ces résultats suggèrent que l'utilisation de méthodes d'extraction de caractéristiques peut simplifier les modèles de classification tout en maintenant leur efficacité, ce qui peut être particulièrement bénéfique dans des environnements où la rapidité et la précision sont essentielles. En revanche, les techniques de sélection de caractéristiques peuvent encore être utiles mais peuvent nécessiter une analyse plus approfondie pour identifier et conserver un ensemble approprié de caractéristiques pertinentes. En conclusion, cette étude met en lumière l'importance de choisir des méthodes appropriées pour la prétraitement des données dans la détection d'e-mails de hameçonnage.

### 2.2.3 Approches de deep learning et nouvelles techniques

Paul Boyle et Lynsay A. Shepherd ont développé MailTrout [35], une extension de navigateur utilisant l'apprentissage automatique pour détecter les e-mails de hameçonnage. Le modèle d'apprentissage automatique (ML) de MailTrout a été développé en utilisant Python 3, la bibliothèque Keras pour l'apprentissage profond, et la bibliothèque open-source TensorFlow. L'algorithme choisi est un réseau de neurones récurrents à mémoire à long terme bidirectionnelle (BLSTM), qui est particulièrement adapté pour le traitement du langage naturel (NLP) en raison de sa capacité à apprendre les dépendances à long terme dans les séquences de texte. Le modèle a été conçu pour classer les e-mails en cinq catégories : légitime (HAM) et quatre types de phishing : usurpation d'identité (IMP), compromission de courrier d'affaires (BEC), extorsion (EXT), et arnaques de gains inattendus (UNX).

Pour l'entraînement, le modèle a utilisé un ensemble de données composé de 11 227 e-mails, comprenant des e-mails légitimes et des exemples de phishing obtenus à partir de diverses sources, y compris des forums en ligne et des outils OCR pour extraire le texte des images d'e-mails de phishing. Les e-mails ont été convertis en séquences de mots tokenisées et normalisées à une longueur fixe de 500 mots. Les données ont été divisées en ensembles d'entraînement et de validation selon le principe de Pareto (80%/20%).

L'extension de navigateur MailTrout a été développée pour Google Chrome. Elle permet aux utilisateurs de sélectionner du texte dans un e-mail et de le soumettre au modèle ML pour une classification en temps réel. Le résultat est affiché dans une fenêtre contextuelle, indiquant si l'e-mail est légitime ou un type spécifique de phishing, accompagné d'une probabilité. L'interface utilisateur utilise un schéma de couleurs contrastées pour faciliter la compréhension des instructions et des résultats, même pour les personnes atteintes de déficiences visuelles.

Des tests utilisateurs ont été menés avec 44 participants pour évaluer la convivialité de l'extension. Les résultats ont montré une grande satisfaction des utilisateurs, avec une note de 87,5 sur 100 sur l'échelle de convivialité du système (SUS). Les participants ont trouvé l'extension facile à utiliser et efficace pour identifier les e-mails de phishing. Cependant, certains utilisateurs ont signalé des problèmes avec la précision des classifications lorsqu'une partie du texte était omise. Le modèle a montré une grande précision générale, mais des catégories comme la compromission de courrier d'affaires (BEC) ont présenté un taux plus élevé de faux négatifs pendant les tests utilisateurs.

En résumé, MailTrout s'avère être un outil prometteur pour aider les utilisateurs à identifier les e-mails de phishing, combinant une technologie de pointe en apprentissage automatique avec une interface utilisateur accessible et intuitive.

Cette étude [25] de *Hiransha M, Nidhin A Unnithan, Vinayakumar R et Soman KP* explore l'utilisation de modèles de deep learning pour la classification des e-mails en légitimes et en phishing, en exploitant les embeddings Keras et les couches de convolution (CNN). Les expérimentations ont été menées dans un environnement TensorFlow GPU-acceléré, avec des prétraitements incluant la tokenisation et la normalisation en minuscules des e-mails. Un dictionnaire a été établi pour attribuer des identifiants uniques à chaque mot, permettant la représentation de chaque e-mail par un vecteur unique.

Deux tâches distinctes ont été abordées : la première incluant les en-têtes d'e-mail et la seconde se concentrant uniquement sur le corps des messages. Les jeux de données utilisés comprenaient respectivement 4,583 et 5,700 e-mails pour l'entraînement, avec des proportions de légitimes et de phishing spécifiées. Pour l'évaluation, 4,195 et 4,300 e-mails ont été utilisés pour les tests.

L'architecture proposée combine des embeddings Keras avec des couches CNN, suivies d'une classification. Les résultats ont démontré une amélioration progressive de la performance avec l'augmentation du nombre d'époques de CNN. Par exemple, dans la tâche 1 (sans en-têtes d'e-mail), le modèle CNN avec 1000 époques a atteint une précision de 95.18%, tandis que dans la tâche 2 (avec en-têtes), le modèle CNN avec 500 époques a obtenu une précision de 97.04%.

En conclusion, cette étude souligne l'efficacité des modèles CNN avec embeddings Keras pour la classification précise des e-mails en légitimes et en phishing, avec des implications significatives pour la sécurité et la gestion des communications électroniques dans divers contextes organisationnels. Les résultats indiquent que l'inclusion des en-têtes d'e-mail peut significativement améliorer les performances des modèles de classification.

### 2.2.4 Comparaison des approches et synthèse

Cette sous-section présente une comparaison des différentes approches utilisées pour la détection des e-mails de phishing, résumées dans le tableau suivant. Nous analyserons les méthodes employées, les jeux de données utilisés, les techniques de classification et les performances obtenues. L'objectif est de synthétiser les résultats des différentes études pour identifier les approches les plus efficaces et les opportunités d'amélioration dans ce domaine.

Étude	Article	Données utilisées	Prétraitement et extraction de fonctionnalités	Techniques de classification	Performance
[21]	AN INTELLIGENT CLASSIFICATION MODEL FOR PHISHING e-mail DETECTION	5940 e-mails légitimes du Spam Assassin, 4598 e-mails de phishing du corpus Nazario Phishing.	Extraction de 16 caractéristiques des en-têtes et corps des e-mails, Utilisation de l'Information Gain (IG) pour sélectionner les caractéristiques les plus discriminantes et Traitement sémantique avec l'ontologie WordNet.	J48 (C4.5), Naïve Bayes, SVM (Support Vector Machine), Multi-Layer Perceptron (MLP) et Random Forest.	Random Forest a dominé grâce à ses ensembles d'arbres pour traiter efficacement les caractéristiques non linéaires et corrélées
[15]	Obtaining the threat model for e-mail phishing	450 e-mails légitimes et 450 e-mails de phishing sélectionnés du serveur SMTP de l'Université du Paraná.	Utilisation de l'algorithme Hill Climbing pour sélectionner les meilleures caractéristiques parmi jusqu'à 211 combinaisons possibles; la limite choisie est de huit caractéristiques.	Support Vector Machine (SVM)	Les classificateurs avec 6, 9 et 11 caractéristiques ont montré des performances optimales avec une AUC proche de 1.
[24]	A Machine Learning Approach Towards Phishing e-mail Detection	Pour l'entraînement : 4082 e-mails légitimes avec en-tête, 501 e-mails de phishing avec en-tête, 5088 e-mails légitimes sans en-tête et 612 e-mails de phishing sans en-tête.	TF-IDF pour transformer les e-mails en format numérique et SVD/NMF pour la sélection des caractéristiques et la réduction de la dimensionnalité.	Arbre de Décision, Random Forest, Ada-Boost, KNN et SVM	Meilleure précision d'entraînement avec Decision Tree et Random Forest, mais surapprentissage détecté sur les données de test en raison du déséquilibre élevé des données.
[13]	Improved Phishing Detection using Model-Based Features	Base07 : 6951 e-mails légitimes et 857 e-mails de phishing.	Adaptive Dynamic Markov Chains (DMC) et Latent Class-Topic Models (CLTOM).	SVM avec un noyau RBF	Les méthodes de classification ont dépassé les performances des méthodes standards sur les corpus de benchmark publics.
[38]	Feature Extraction or Feature Selection for Text Classification : A Case Study on Phishing e-mail Detection	1000 e-mails de Phishing et 1700 e-mails légitimes.	Extraction des caractéristiques : Principal Components Analysis (PCA) et Latent Semantic Analysis (LSA). Sélection des caractéristiques : Chi-Square et Information Gain.	J48 decision tree	Les méthodes d'extraction de caractéristiques sont efficaces avec peu de caractéristiques mais moins performantes avec un nombre élevé.

TABLE 2.1 – Résumé des études sur la détection de phishing (Partie 1)

Étude	Article	Données utilisées	Prétraitement et extraction de fonctionnalités	Techniques de classification	Performance
[35]	MailTrout : A Machine Learning Browser Extension for Detecting Phishing e-mails	L'ensemble de données sur les e-mails de fraude publié par Verma (2018).		Artificial neural networks (ANNs), Long Short-Term Memory networks (LSTMs) et Bidirectional Long Short-Term Memory networks (BLSTMs).	Le modèle surpasse d'autres méthodes ML de détection de phishing, avec des performances améliorées par rapport à SpamAssassin et au meilleur classificateur RNN connu (Halgaš et al., 2019) [30].
[25]	Deep Learning Based Phishing E-mail Detection	Pour l'entraînement : Task 1 - 4583 e-mails entre 4082 légitimes et 501 de phishing ; Task 2 - 5700 e-mails entre 5088 légitimes et 612 de phishing. Pour le test : Task 1 - 4195 e-mails ; Task 2 - 4300 e-mails.		Embeddings Keras combiné avec les couches CNN	Amélioration graduelle de la performance avec l'augmentation du nombre d'époques de CNN.

TABLE 2.2 – Résumé des études sur la détection de phishing (Partie 2)

## 2.3 Conclusion

Les études antérieures sur la détection de phishing par e-mail révèlent plusieurs limitations, notamment l'utilisation de petits ensembles de données qui limitent la généralisation des modèles. Les approches telles que SVM avec un noyau gaussien sont coûteuses en calculs et sujettes au sur-apprentissage. De plus, l'entraînement sur des ensembles de données déséquilibrés biaise les résultats en faveur de la classe majoritaire. MailTrout, malgré son modèle BLSTM puissant, est sensible aux données d'entrée incomplètes et nécessite des ressources importantes. Les techniques de sélection de caractéristiques, comme le Chi-Carré et le Gain d'Information, ajoutent de la complexité sans améliorer nécessairement la performance.

Notre approche novatrice combine le traitement du langage naturel (NLP) avec une analyse holistique des en-têtes, du corps du texte et des liens d'e-mails pour améliorer la détection de phishing. Entraîné sur un ensemble de données diversifié et étendu, notre modèle utilise le deep learning et l'apprentissage semi-supervisé, validé par des évaluations comparatives rigoureuses. Malgré des avancées significatives, des améliorations continues sont nécessaires pour renforcer la robustesse de notre solution contre les techniques de phishing sophistiquées.

# Chapitre 3

## Méthodologie

### 3.1 Introduction

*Le phishing est un crime qui combine à la fois de l'ingénierie sociale et des subterfuges techniques pour voler les données d'identité personnelle des consommateurs ainsi que leurs informations d'accès financières. Les schémas d'ingénierie sociale ciblent les victimes imprudentes en les trompant pour qu'elles croient qu'elles traitent avec une entité de confiance et légitime, par exemple en utilisant des adresses e-mail et des messages e-mail trompeurs. Ces tactiques sont conçues pour diriger les consommateurs vers des sites web contrefaits qui les incitent à divulguer des données financières telles que des noms d'utilisateur et des mots de passe. Les stratagèmes de subterfuge technique installent des logiciels malveillants sur les ordinateurs pour voler directement les informations d'accès, souvent en utilisant des systèmes qui interceptent les noms d'utilisateur et les mots de passe des comptes des consommateurs ou les dirigent vers des sites web contrefaits[11].*

#### Utilisation du NLP pour la détection de phishing

Dans le cadre de la détection des e-mails de phishing, le NLP analyse et comprend les nuances du langage humain pour identifier les tentatives de tromperie sophistiquées. Contrairement à la communication humaine directe, les e-mails de phishing utilisent des techniques de manipulation pour induire en erreur, souvent en créant un sentiment d'urgence ou en offrant des incitations pour inciter les utilisateurs à révéler des informations sensibles ou à cliquer sur des liens malveillants. Les algorithmes avancés de NLP, tels que la classification de texte, l'analyse

sémantique et la détection des entités nommées, permettent de repérer les modèles linguistiques typiques des tentatives de phishing. Intégrés avec des méthodes de machine learning et d'apprentissage profond, ces systèmes deviennent plus efficaces pour filtrer les e-mails frauduleux tout en minimisant les faux positifs, renforçant ainsi la sécurité des utilisateurs contre les attaques de phishing.

### 3.2 Méthodologie générale

Dans notre approche pour détecter les e-mails de phishing, nous combinons plusieurs techniques avancées de machine learning et de traitement du langage naturel (NLP).

Tout d'abord, une étape de traduction sera effectuée pour les e-mails rédigés dans des langues

différentes de l'anglais. Cette traduction est réalisée à l'aide de l'API `mtranslate`, qui détecte automatiquement la langue source et la traduit en anglais. Ensuite, l'e-mail sera décomposé en trois composantes principales : l'en-tête, le corps et les liens. Chaque composante est analysée séparément en utilisant des méthodes spécifiques adaptées à leurs caractéristiques respectives.

Pour l'analyse de l'en-tête, nous vérifions les adresses IP contre des listes noires et utilisons des techniques de machine learning pour détecter des anomalies dans les métadonnées et les domaines expéditeurs. Le corps de l'e-mail est soumis à des techniques de NLP telles que la classification de texte pour identifier des motifs linguistiques typiques des tentatives de phishing, l'analyse sémantique pour décoder les intentions suspectes, et la détection des entités nommées pour repérer les informations sensibles.

Les liens contenus dans les e-mails sont extraits et analysés pour détecter des URL malveillantes. Cette étape inclut l'analyse heuristique pour identifier des tentatives de dissimulation ou de redirection.

Les scores de risque générés pour chaque composante (en-tête, corps et liens) sont ensuite combinés pour produire une prédiction finale sur la nature de l'e-mail. Cette approche holistique permet une détection plus précise et efficace des tentatives de phishing, tout en minimisant les faux positifs et négatifs, renforçant ainsi la sécurité des utilisateurs contre ces menaces de plus en plus sophistiquées.

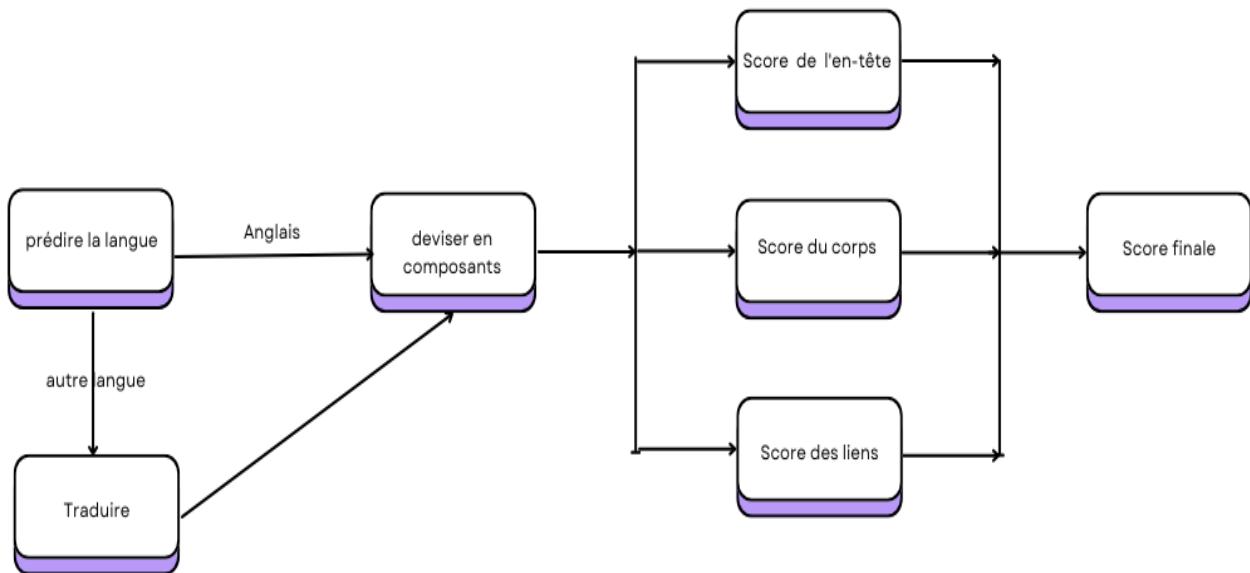


FIGURE 3.1 – Processus de Détection des e-mails de Phishing

### 3.3 Analyse du corps

#### 3.3.1 jeu des données

Dans notre étude, nous avons combiné deux ensembles de données d'e-mails publics, à savoir le corpus Enron [16], composé de 500 000 e-mails légitimes provenant de 158 employés d'Enron, et le corpus de Phishing de Jose Nazario[31], contenant des e-mails de phishing recueillis sur une

période de 2004 à 2020. Face au déséquilibre naturel entre les classes légitimes et de phishing, nous avons choisi trois échantillons distincts. Le premier échantillon comprenait 20 000 e-mails légitimes provenant du corpus d'Enron, le second échantillon comprenait 2 279 e-mails légitimes du même corpus , et le troisième comprenait 1000 e-mail légitime . Ces échantillons ont été comparés avec l'intégralité des e-mails de phishing issus du dataset de Jose Nazario. Cette sélection nous a permis de comparer le comportement des algorithmes sur des ensembles de données équilibrés et non équilibrés afin d'analyser l'impact de l'équilibre des classes sur la performance des algorithmes de détection. Le tableau ci-dessous illustre la répartition des classes avant le nettoyage et la combinaison, mettant en évidence l'importance de la stratification dans notre analyse comparative.

ensemble de données	Nombre total	ensemble équilibrée	ensemble dés-équilibrée 1	ensemble dés-équilibrée 2
le corpus Enron	500 000	2279	20000	1000
le corpus Nazario	5313	5313	5313	5313

TABLE 3.1 – Détails de données utilisées

### 3.3.2 Techniques de nettoyage et de normalisation des données

Pour préparer nos données d'e-mails en vue de l'entraînement des modèles de détection de phishing, nous avons suivi un processus rigoureux de nettoyage et de normalisation. Tout d'abord, nous avons focalisé notre analyse sur le corps des e-mails, en excluant les entêtes potentielles présentes dans les données brutes. Ensuite, nous avons procédé comme suit :

**1. Suppression des e-mails avec corps nul :** Nous avons éliminé les e-mails dont le corps était manquant afin de garantir l'inclusion uniquement des e-mails complets dans notre ensemble de données. Cela a été réalisé avec des méthodes adaptées à chaque ensemble de données (phishing et légitime).

**2. Suppression des e-mails vides :** Nous avons supprimé tous les e-mails qui ne contenaient pas de texte significatif, assurant ainsi la qualité des données et évitant l'introduction de bruit dans notre analyse.

**3. Élimination des doublons :** Nous avons vérifié la présence de doublons parmi les e-mails restants et les avons supprimés pour éviter toute redondance dans notre ensemble de données final.

**4. Filtrage spécifique des données de phishing :** Nous avons identifié et retiré les e-mails contenant des éléments non pertinents tels que des balises ou des marqueurs indiquant des métadonnées internes plutôt que des messages réels.

**5. Combinaison des données :** Une fois ces étapes de nettoyage achevées pour chaque classe (phishing et légitime) pour les deux échantillons de données , nous avons combiné les deux ensembles de données en trois ensembles finaux équilibré , déséquilibrée avec dominance de classe de phishing et déséquilibrée avec dominance de classe légitime.les tableaux 3.2 , 3.3 et 3.4 et les figures 3.2 , 3.3 et 3.4 illustrent la base sur laquelle nos différents modèles de détection de phishing sont entraînées.

Classe	Nombre d'e-mails
Phishing (classe 1)	2279
Légitime (classe 0)	2279
<b>Total</b>	<b>4558</b>

TABLE 3.2 – Répartition des classes dans l'ensemble de données équilibrée final

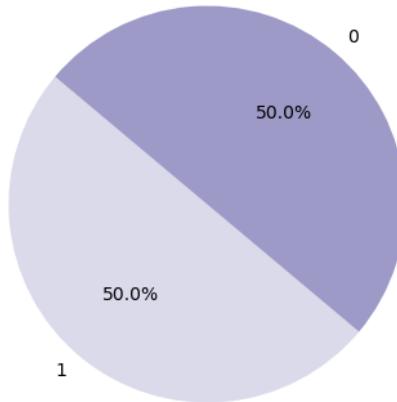


FIGURE 3.2 – Représentation des classes dans l'ensemble de données équilibrée final

Classe	Nombre d'e-mails
Phishing (classe 1)	2279
Légitime (classe 0)	20000
<b>Total</b>	<b>22279</b>

TABLE 3.3 – Répartition des classes dans l'ensemble de données déséquilibrée final 1

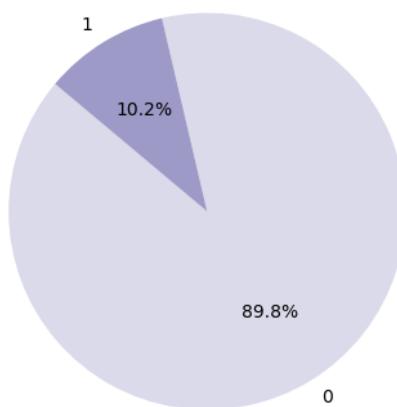


FIGURE 3.3 – Représentation des classes dans l'ensemble de données déséquilibrée final 1

Classe	Nombre d'e-mails
Phishing (classe 1)	2279
Légitime (classe 0)	1000
<b>Total</b>	<b>3279</b>

TABLE 3.4 – Répartition des classes dans l'ensemble de données déséquilibrée final 2

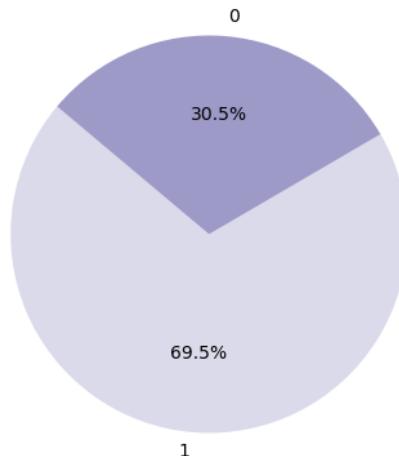


FIGURE 3.4 – Représentation des classes dans l'ensemble de données déséquilibrée final 2

**6. Normalisation avancée du texte :** Une étape avancée de prétraitement a consisté à normaliser le texte en le convertissant en minuscules, en supprimant les ponctuations, les liens, les adresses e-mails présentes dans le corps, les caractères spéciaux et les balises HTML, ainsi qu'à éliminer les mots vides (stopwords).

**7. Tokenization et lemmatization :** Nous avons utilisé la tokenization pour diviser le texte en mots individuels et la lemmatization pour ramener ces mots à leur forme de base en fonction de leur contexte dans la phrase, en utilisant une fonction pour obtenir le type de mot correct (part-of-speech).

Ces étapes de prétraitement avancées garantissent que notre modèle de détection de phishing est formé sur des données nettoyées et normalisées de manière rigoureuse, essentielles pour l'extraction efficace des caractéristiques et l'évaluation précise de sa performance dans des conditions réelles d'utilisation.

### 3.3.3 Extraction et selection des caractéristiques

Pour l'étape d'extraction et de sélection des caractéristiques dans le cadre du développement d'un modèle de détection de phishing, nous avons utilisé : la vectorisation TF-IDF en comparaison avec Word2vec et la sélection de caractéristiques basée sur le test du Chi-deux (Chi2). Voici comment ces étapes ont été implémentées :

### 1. Extraction des caractéristiques

#### TF-IDF

La méthode TF-IDF (Term Frequency-Inverse Document Frequency) est utilisée pour quantifier l'importance d'un terme dans un corpus de documents. Voici comment nous avons intégré cette technique dans notre pipeline :

- **TF-IDF Vectorizer** : Nous avons implémenté une fonction qui prend en entrée une colonne de texte d'entraînement. Cette fonction prépare le texte en le concaténant si nécessaire et utilise TfidfVectorizer de scikit-learn pour calculer les scores TF-IDF pour chaque terme.
- **Sortie** : La fonction retourne un dictionnaire contenant le vectoriseur utilisé et les caractéristiques TF-IDF transformées pour l'ensemble d'entraînement avec 1000 features.

#### Word2Vec

La méthode Word2Vec est utilisée pour représenter les mots sous forme de vecteurs dans un espace de caractéristiques de taille fixe. Voici comment nous avons intégré cette technique dans notre pipeline :

- **Word2Vec Model** : Nous avons implémenté une fonction qui prend en entrée une colonne de texte d'entraînement, ainsi qu'une colonne de texte de test optionnelle. Cette fonction utilise le modèle Word2Vec de gensim pour apprendre des représentations vectorielles des mots dans le texte d'entraînement.
- **Préparation du Texte** : Le modèle Word2Vec est initialisé avec les paramètres suivants : taille du vecteur de 100, min\_count de 5 (le nombre minimal d'occurrences d'un mot pour qu'il soit inclus dans le vocabulaire du modèle), taille maximale du vocabulaire non spécifiée, et utilisation de 1 thread pour l'entraînement. Le modèle est ensuite entraîné sur les phrases de la colonne de texte d'entraînement.
- **Filtrage du Vocabulaire** : Après l'entraînement, nous filtrons les mots dans les colonnes de texte pour ne garder que ceux présents dans le vocabulaire du modèle Word2Vec. Nous calculons ensuite la moyenne des vecteurs Word2Vec pour chaque document afin de créer une représentation vectorielle fixe pour chaque document.
- **Sortie** : La fonction retourne un dictionnaire contenant le modèle Word2Vec entraîné et les caractéristiques transformées pour l'ensemble d'entraînement. Si une colonne de texte de test est fournie, les caractéristiques transformées pour cet ensemble sont également incluses. Chaque document est représenté par un vecteur de 100 caractéristiques.

### 2. Sélection de caractéristiques avec Chi2

Une fois les caractéristiques extraites, nous avons utilisé la méthode Chi2 pour sélectionner les caractéristiques les plus informatives en fonction de leur relation avec les classes (phishing ou légitime). Voici comment cela a été mis en œuvre :

- **Sélection de caractéristiques** : Nous avons développé une fonction qui prend en entrée les caractéristiques extraites et les étiquettes de classe correspondantes . Cette fonction utilise SelectPercentile avec le test du Chi-deux pour sélectionner un pourcentage spécifié de caractéristiques les plus discriminantes.
- **Sortie** : La fonction retourne un dictionnaire contenant le sélecteur de caractéristiques et les caractéristiques sélectionnées pour l'ensemble d'entraînement .

### 3.3.4 Architecture des Algorithmes Utilisés

Cette section décrit l'implémentation des différents algorithmes de machine learning et deep learning utilisés pour détecter les emails de phishing. Nous avons utilisé plusieurs algorithmes classiques ainsi que des modèles plus avancés basés sur des réseaux de neurones.

#### Régression Logistique

Pour la régression logistique, nous avons utilisé la classe `LogisticRegression` de la bibliothèque `scikit-learn`. Le modèle a été entraîné avec les paramètres suivants :

- `max_iter = 1000`
- `penalty = 'l2'`
- `C = 1e10`

Une standardisation des données a été effectuée à l'aide de `StandardScaler` lorsque cela était nécessaire. Le modèle a été ajusté sur les données d'entraînement et sauvegardé pour les prédictions futures.

#### Machine à Vecteurs de Support (SVM)

Pour le SVM, nous avons utilisé la classe `SVC` avec les paramètres suivants :

- `kernel = 'rbf'`
- `C = 1.0`
- `gamma = 'scale'`

Le modèle a été entraîné sur les données d'entraînement en utilisant les hyperparamètres spécifiés et a montré de bonnes performances pour la détection de phishing.

#### Arbre de Décision

L'algorithme d'arbre de décision a été implémenté en utilisant `DecisionTreeClassifier` avec une profondeur maximale définie à `5` (`max_depth = 5`).

Cette restriction de profondeur aide à éviter le sur-apprentissage et à maintenir un modèle généraliste.

#### K Plus Proches Voisins (KNN)

Le modèle KNN a été configuré avec le nombre de voisins (`n_neighbors`) fixé à `5`.

L'algorithme a été entraîné pour classer les emails en fonction de leurs voisins les plus proches dans l'espace des caractéristiques.

### Forêt Aléatoire

Pour la forêt aléatoire, nous avons utilisé `RandomForestClassifier` avec les paramètres suivants :

- `max_depth = 5`
- `n_estimators = 20`

Ce modèle agrège les prédictions de plusieurs arbres de décision pour améliorer la robustesse et la précision.

### Gradient Boosting

Le modèle de Gradient Boosting a été configuré avec `GradientBoostingClassifier`, en utilisant une perte logarithmique (`loss = 'log_loss'`) et une profondeur maximale des arbres (`max_depth = 3`). La vitesse d'apprentissage a été fixée à `0.1` pour équilibrer la convergence et la précision.

### Naïve Bayes

Pour le modèle Naïve Bayes, nous avons utilisé `MultinomialNB` avec un paramètre `alpha` fixé à `1.0`.

La transformation Min-Max a été appliquée aux données d'entraînement pour assurer que toutes les caractéristiques sont non négatives.

### Réseau de Neurones artificiels (ANN)

Le modèle de réseau de neurones artificiels (ANN) a été implémenté en utilisant la bibliothèque `Keras`. L'architecture du modèle était la suivante :

- Une couche d'entrée avec 512 neurones et une activation `ReLU`
- Deux couches cachées de 256 et 128 neurones avec des activations `ReLU`, accompagnées de couches de `Dropout` fixées à 0.5 pour éviter le sur-apprentissage
- Une couche de sortie avec une activation `sigmoid` pour la classification binaire.

Le modèle a été entraîné en utilisant l'optimiseur `Adam` avec une learning rate de 0.001 et une fonction de perte `binary_crossentropy`. Cela a permis d'obtenir une précision et une généralisation satisfaisantes pour la classification des e-mails en phishing ou non.

Cette architecture a été choisie pour sa capacité à capturer des modèles complexes dans les données extraites après prétraitement, visant ainsi à identifier efficacement les tentatives de phishing parmi les e-mails.

### Réseau de Neurones Convolutif (CNN)

Le modèle CNN a été construit avec les éléments suivants :

- Deux couches convolutionnelles avec 128 et 64 filtres respectivement, utilisant des noyaux de taille 5 et des activations `ReLU`
- Des couches de pooling pour la réduction de la dimensionnalité

- Une couche de Flatten suivie par une couche dense avec 64 neurones et une activation ReLU
- Une couche de sortie avec une activation sigmoid

Ce modèle a été optimisé avec Adam et a utilisé la fonction de perte binary\_crossentropy.

### Réseau de Neurones Récurrent (RNN)

Le modèle RNN a été implémenté avec :

- Deux couches LSTM, la première avec 128 unités et la seconde avec 64 unités
- Une couche dense avec 64 neurones et une activation ReLU
- Une couche de sortie avec une activation sigmoid

L'optimiseur utilisé était Adam et la fonction de perte était binary\_crossentropy.

#### 3.3.5 Métriques d'évaluation

Une fois les modèles de détection de phishing entraînés et évalués, nous avons utilisé plusieurs métriques d'évaluation pour mesurer leur performance. Ces métriques nous ont permis de quantifier l'efficacité de chaque modèle dans la détection des e-mails de phishing.

Les principaux critères d'évaluation incluent la précision, le rappel, le F1-score et l'exactitude. La précision indique la précision des prédictions positives du modèle, tandis que le rappel mesure sa capacité à identifier correctement les exemples positifs. Le F1-score combine ces deux mesures en une seule valeur harmonique, fournissant une vue globale de la performance du modèle. Enfin, l'exactitude donne une mesure générale de la proportion d'e-mails correctement classés par le modèle.

L'évaluation du vote majoritaire revêt une importance significative dans le contexte de la détection de phishing, car elle permet de renforcer la fiabilité des prédictions en combinant les résultats de plusieurs modèles de manière judicieuse.

Ces résultats détaillés sont présentés et discutés dans le chapitre suivant, offrant ainsi une analyse approfondie de la capacité de chaque modèle à détecter efficacement les e-mails de phishing dans des conditions réelles d'utilisation.

#### 3.3.6 Intégration du Score Sémantique aux Résultats du Modèle

Après avoir entraîné notre modèle de détection de phishing sur les caractéristiques extraites des e-mails, nous avons introduit une étape supplémentaire pour enrichir nos prédictions. Cette étape consiste à calculer un score sémantique pour chaque e-mail dans notre ensemble de test, après que le modèle a généré ses prédictions.

Nous avons défini des ensembles de mots clés et de verbes liés aux e-mails de phishing. Le score est calculé en fonction de la présence de ces mots et de la structure des phrases, intégrant les éléments suivants :

- Verbes d'action (ex : "click", "follow")
- Adverbes de position (ex : "here", "there")
- Mots urgents (ex : "now", "immediately")
- Descripteurs de direction (ex : "above", "below")

Le score est calculé selon une formule qui prend en compte la présence de liens, de mots urgents, le nombre de mots dans l'e-mail et de la structure du texte.

Une fois les prédictions du modèle obtenues pour l'ensemble de test, nous avons combiné ces prédictions avec les scores sémantiques calculés. Cette combinaison nous permet de renforcer nos résultats en tenant compte à la fois des caractéristiques extraites par le modèle et des informations sémantiques supplémentaires.

Cette approche hybride nous offre une perspective plus riche sur la détection des e-mails de phishing, en intégrant des éléments d'analyse syntaxique et sémantique aux capacités prédictives du modèle.

### 3.4 Analyse des entêtes

L'analyse des en-têtes de courriel est cruciale pour évaluer la légitimité et la sécurité d'un courriel. Cette section décrit en détail la méthode utilisée pour analyser les en-têtes de courriel et en extraire des informations pertinentes.

L'analyse des en-têtes se déroule en plusieurs étapes méthodiques, chacune visant à extraire ou vérifier des éléments spécifiques des en-têtes de courriel. Voici les étapes détaillées :

#### 1. Extraction des Champs de Base :

- **Expéditeur (From)** : nous avons développées une méthode qui utilise une expression régulière pour extraire l'adresse e-mail de l'expéditeur. Par exemple, pour une ligne d'en-tête "From : John Doe john.doe@example.com", l'adresse "john.doe@example.com" est extraite.
- **Destinataire (To)** : De manière similaire, une autre fonction extrait l'adresse e-mail du destinataire à partir de l'en-tête.
- **Sujet (Subject)** : extraction de sujet du courriel en recherchant la ligne correspondante dans l'en-tête.
- **Date (Date)** : une dernière fonction extrait la date d'envoi du courriel à partir de l'en-tête.

#### 2. Validation DNS :

- La méthode de validation DNS examine les lignes "Received" dans l'en-tête pour identifier les hôtes impliqués dans la transmission du courriel. Elle tente ensuite de résoudre ces noms de domaine en adresses IP valides à l'aide de requêtes DNS. Si les hôtes sont validés, ils sont ajoutés à une liste de validation.

### 3. Vérification des Signatures DKIM et SPF :

- **DKIM** : une fonction vérifie la présence d'une signature DKIM dans l'en-tête pour s'assurer que le courriel n'a pas été altéré pendant la transmission.
- **SPF** : une autre fonction vérifie les en-têtes "Received-SPF" pour confirmer que le courriel a passé la vérification SPF, indiquant qu'il provient d'une source autorisée par le domaine de l'expéditeur.

### 4. Détection des Mots-clés de la Liste Noire :

- Une méthode recherche des mots-clés spécifiques dans l'en-tête pour détecter des contenus potentiellement malveillants ou suspects. Les mots-clés comprennent des termes tels que "Account", "Debit", "Recently", etc. La présence de ces mots-clés indique un risque potentiel.

### 5. Analyse Supplémentaire :

- Cette étape effectue une analyse supplémentaire en recherchant des champs spécifiques comme "X-Mailman-Version", "X-Spam-Flag" et "X-Virus-Scanned". Ces informations supplémentaires peuvent fournir des indices supplémentaires sur la légitimité du courriel.

### 6. Calcul du Score de l'En-tête :

- Finalement, une dernière fonction compile les résultats des vérifications précédentes pour calculer un score global. Le score est calculé en ajoutant des points pour les éléments négatifs (présence de DKIM, SPF, DNS non validé, présence de mots-clés de la liste noire, indicateurs de spam). Le score final permet d'évaluer la probabilité de phishing d'un courriel.

#### Exemple de Calcul du Score :

- DKIM non Vérifié : +0.2
- SPF non Vérifié : +0.2
- DNS Validé : +0.2
- Mots-clés de la Liste Noire : +0.3 (par mot-clé trouvé)
- Analyse Supplémentaire : Chaque mot-clé trouvé dans les résultats de l'analyse supplémentaire augmente le score de 0.1

## 3.5 Analyse des liens

L'analyse des caractéristiques d'une URL est essentielle pour évaluer sa sécurité et sa fiabilité. Voici en détail le processus d'analyse des différentes composantes et caractéristiques clés d'une URL :

### 1. Composants de l'URL :

L'URL est décomposée en trois parties principales :

- **Domaine** : C'est l'adresse principale de l'entité en ligne, souvent précédée de https :// ou http ://.
- **Chemin** : Il indique l'emplacement spécifique des ressources sur le serveur.
- **Requête** : Elle comprend des paramètres optionnels qui peuvent être utilisés pour transmettre des informations au serveur.

### 2. Extraction du TLD (Top-Level Domain) :

Le TLD est extrait pour identifier la catégorie du domaine, comme .com, .org, .net, etc. Cela permet de classer le type d'organisation ou de service auquel le domaine appartient.

### 3. Scores des Caractéristiques :

Chaque aspect de l'URL est évalué et reçoit un score basé sur les critères suivants :

Critère	Description
Longueur du domaine	Les domaines excessivement longs ou courts peuvent être indicatifs de sites non standard.
Longueur du chemin	Un chemin trop long peut être complexe ou indicatif de comportements anormaux.
Présence de paramètres dans la requête	Plus il y a de paramètres, plus l'URL peut être complexe ou potentiellement malveillante.
Nombre de sous-domaines	Un grand nombre de sous-domaines peut indiquer une structure de site inhabituelle ou une tentative de confusion.
Présence de caractères inhabituels dans le domaine	Des caractères non standard dans le domaine peuvent indiquer une tentative de phishing ou une URL suspecte.
Utilisation d'un raccourcisseur d'URL connu	Certains services de raccourcissement d'URL sont associés à des pratiques malveillantes ou à du spam.
Conformité du TLD avec une liste de confiance	Certains TLD comme .com, .org, .net sont généralement considérés comme plus fiables que d'autres.
Utilisation du protocole HTTPS	Les connexions HTTPS sont sécurisées et protègent les données contre l'interception.
Présence de redirection dans l'URL	Les redirections peuvent être utilisées pour tromper les utilisateurs vers des sites malveillants.

TABLE 3.5 – Critères pour évaluer les URL

### 4. Score Final :

Les scores individuels attribués à chaque caractéristique sont agrégés pour calculer un score global. Ce score final reflète le niveau de sécurité et de fiabilité de l'URL analysée. Un score élevé indique une URL potentiellement dangereuse ou suspecte, tandis qu'un score bas peut indiquer une URL plus sûre et légitime.

## 3.6 Combinaison des scores

La combinaison des scores pour évaluer la probabilité qu'un email soit un phishing est réalisée en suivant un processus intégré. Tout d'abord, l'email est potentiellement traduit vers l'anglais pour une analyse uniforme. Ensuite, l'analyse se divise en deux axes principaux : l'en-tête et le corps de l'email. Les liens contenus dans le corps sont extraits et évalués pour détecter tout signe de phishing potentiel, tandis que l'en-tête est scruté pour identifier des indicateurs de fraude ou de contenu suspect. Parallèlement, le corps de l'email est examiné à la recherche de motifs de texte associés à des activités malveillantes. En consolidant les scores issus de ces analyses distinctes, un score final est calculé. Ce score final représente de manière synthétique la probabilité que l'email soit une tentative de phishing, offrant ainsi une évaluation globale de sa sécurité et de sa légitimité présumée pour l'utilisateur.

## 3.7 Conclusion

Dans cette étude approfondie, plusieurs algorithmes ont été explorés et intégrés pour renforcer la détection des e-mails de phishing. Parmi ceux-ci, les techniques de traitement du langage naturel (NLP) ont joué un rôle crucial. L'utilisation de la vectorisation TF-IDF pour représenter les caractéristiques textuelles et la sélection des termes significatifs à l'aide du test du Chi-deux ont permis de filtrer et de hiérarchiser les informations les plus pertinentes dans les e-mails suspectés. Ces méthodes ont été complétées par l'application de modèles de classification avancés tels que les machines à vecteurs de support (SVM) et les réseaux de neurones convolutifs (CNN), qui ont démontré une capacité accrue à distinguer les patterns subtils des e-mails de phishing par rapport aux communications légitimes.

En parallèle, l'analyse sémantique a été un pilier essentiel de cette approche intégrée. En examinant les structures de phrases, les mots-clés et les schémas de formulation typiques des tentatives de phishing, il a été possible d'identifier les intentions malveillantes derrière ces communications.

En outre, l'analyse des en-têtes et des liens a été une autre composante critique de notre méthodologie. En examinant les métadonnées des e-mails, telles que les adresses IP d'origine, les en-têtes SMTP et les URLs inclus dans les messages, nous avons pu détecter des anomalies ou des indicateurs de compromission potentielle. L'application de techniques de clustering et de vérification des signatures numériques pour les liens inclus dans les e-mails a permis de prévenir efficacement les utilisateurs contre les redirections malveillantes ou les sites web frauduleux. En combinant ces différentes approches, nous avons créé un système robuste et adaptable capable de détecter et de neutraliser efficacement les menaces de phishing, offrant ainsi une protection avancée contre cette forme croissante de cybercriminalité.

# Chapitre 4

## Résultats et discussion

### 4.1 Introduction

Ce chapitre présente en détail les résultats obtenus lors de l'application des méthodologies décrites précédemment pour la détection des e-mails de phishing. Nous avons évalué plusieurs modèles de machine learning et de deep learning en utilisant des métriques standard telles que l'exactitude, la précision, le rappel, le F1-score et l'AUC (aire sous la courbe ROC). Ces évaluations nous ont permis de mesurer l'efficacité de chaque modèle dans la distinction entre les e-mails légitimes et les e-mails malveillants.

Nous avons également exploré l'impact de l'intégration du score sémantique ainsi que de l'analyse des en-têtes et des liens des e-mails sur les performances des modèles. Cette approche vise à renforcer la capacité des modèles à détecter les stratagèmes de phishing en exploitant à la fois le contenu des messages et les caractéristiques structurelles des e-mails suspects.

### 4.2 Présentation des résultats de l'analyse du corps

#### 4.2.1 Résultats des Modèles de Machine Learning

Dans cette section, nous présentons les résultats des différents modèles de machine learning en utilisant deux types de données : équilibrées et déséquilibrées. La comparaison se fait également en fonction des méthodes d'extraction des caractéristiques, à savoir TFIDF et Word2Vec, et en considérant l'utilisation ou non de la sélection de caractéristiques avec le test du chi-carré (CHI-Square).

## 1. Données Équilibrées

### Extraction des caractéristiques avec TF-IDF sans sélection des caractéristiques

Modèle	Exactitude	Précision	Rappel	F1	AUC	VN	FP	FN	VP
Régression logistique	0.946272	0.926247	0.966063	0.945736	0.968119	436	34	15	427
SVM	0.972588	0.990588	0.952489	0.971165	0.988876	466	4	21	421
Arbre de décision	0.913377	0.923077	0.895928	0.909300	0.933992	437	33	46	396
Forets aleatoires	0.900219	0.953488	0.834842	0.890229	0.977720	452	18	73	369
KNN	0.890351	0.983051	0.787330	0.874372	0.939930	464	6	94	348
Gradient Boosting	0.955044	0.967366	0.938914	0.952928	0.990844	456	14	27	415
Naive Bayes	0.949561	0.945946	0.950226	0.948081	0.990180	446	24	22	420

TABLE 4.1 – Performance des modèles de classification avec TF-IDf sans sélection de caractéristiques sur données équilibrées

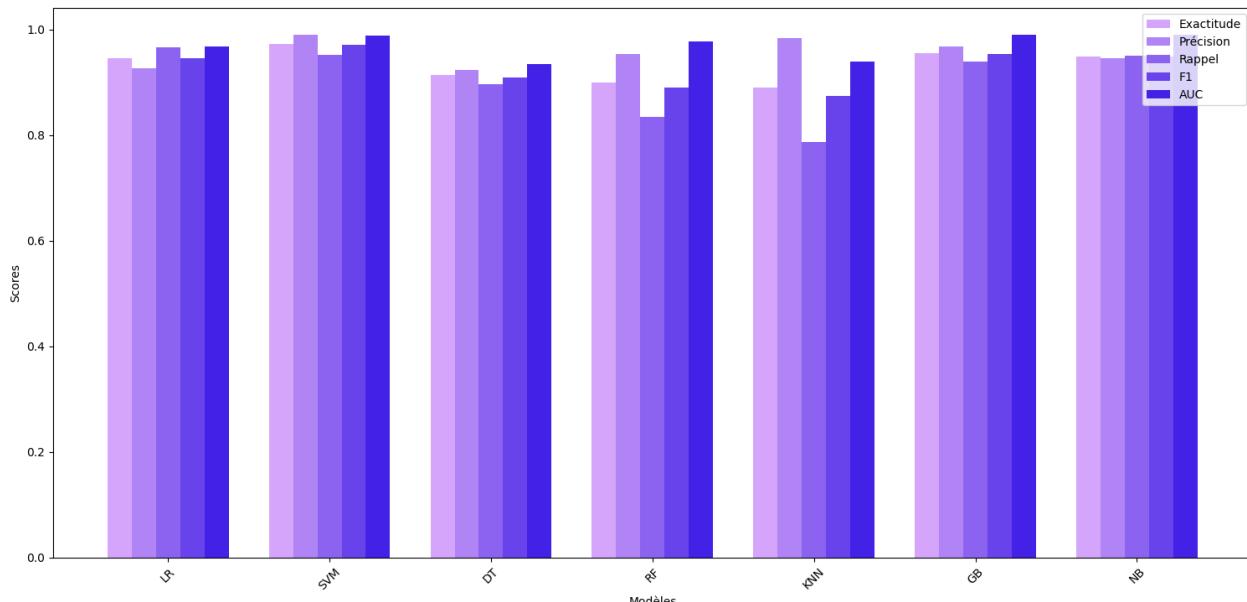


FIGURE 4.1 – Comparaison des performances des modèles de classification avec TF-IDf sans sélection de caractéristiques sur données équilibrées

Le tableau et le graphique en barres pour les données équilibrées extractées avec TFIDF et sans sélection de caractéristiques montrent que le SVM se distingue par la meilleure précision (99,06%) et la plus haute AUC (98,88%), surpassant les autres modèles. La régression logistique et le Gradient Boosting offrent des performances équilibrées, tandis que le KNN présente un rappel relativement faible malgré une haute précision.

### Extraction des caractéristiques avec TFIDF avec sélection des caractéristiques

Modèle	Exactitude	Précision	Rappel	F1	AUC	VN	FP	FN	VP
Régression logistique	0.950658	0.938190	0.961538	0.949721	0.969409	442	28	17	425
SVM	0.962719	0.978873	0.943439	0.960829	0.989232	461	9	25	417
Arbre de décision	0.914474	0.925234	0.895928	0.910345	0.934567	438	32	46	396
Forets aleatoires	0.916667	0.962121	0.861991	0.909308	0.979794	455	15	61	381
KNN	0.925439	0.926941	0.918552	0.922727	0.963447	438	32	36	406
Gradient Boosting	0.955044	0.963048	0.943439	0.953143	0.990671	454	16	25	417
Naive Bayes	0.944079	0.931567	0.954751	0.943017	0.990065	439	31	20	422

TABLE 4.2 – Performance des modèles de classification avec TF-IDF et sélection de caractéristiques sur données équilibrées

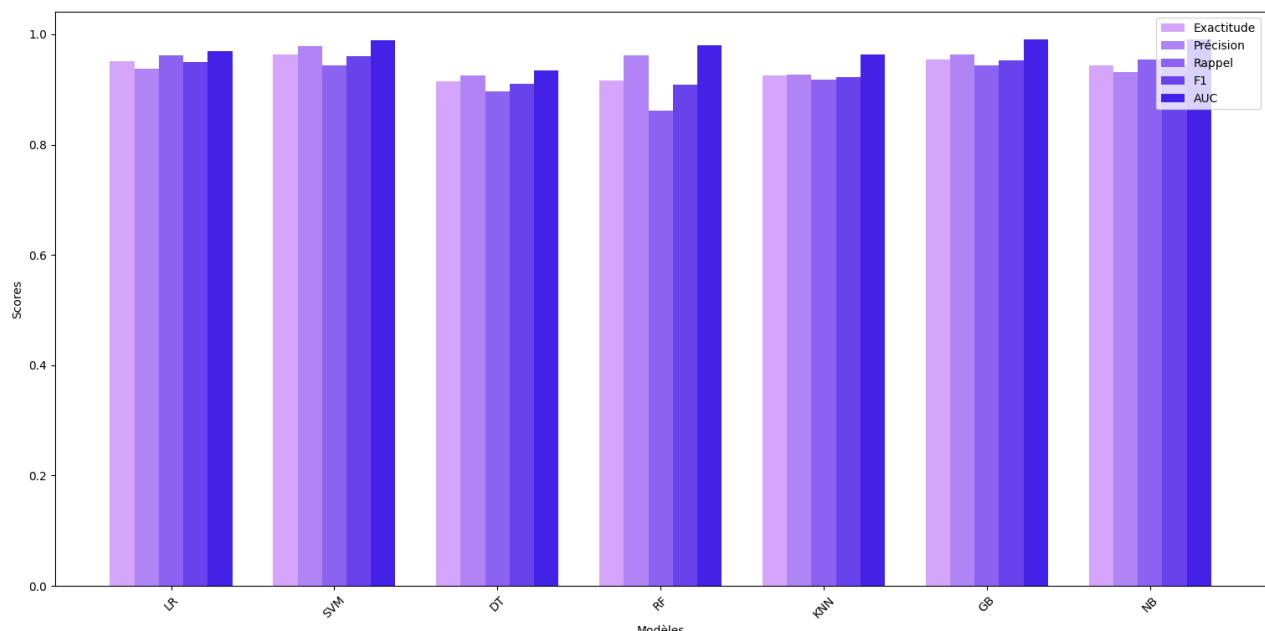


FIGURE 4.2 – Comparaison des performances des modèles de classification avec TF-IDF et sélection de caractéristiques sur données équilibrées

Le tableau et le graphique en barres pour les données avec sélection de caractéristiques montrent que le SVM maintient des performances remarquables avec une précision élevée de 97,89% et une AUC de 98,92%. Comparativement aux résultats sans sélection, bien que légèrement inférieurs, le SVM confirme sa robustesse et sa capacité à maintenir une précision élevée, soulignant son efficacité même avec des caractéristiques sélectionnées.

### Extraction des caractéristiques avec Word2vec

Modèle	Exactitude	Précision	Rappel	F1	AUC	VN	FP	FN	VP
Régression Logistique	0.959430	0.953020	0.963801	0.958380	0.991764	449	21	16	426
SVM	0.969298	0.974771	0.961538	0.968109	0.992828	459	11	17	425
KNN	0.970395	0.977011	0.961538	0.969213	0.984103	460	10	17	425
Arbre de Décision	0.949561	0.952055	0.943439	0.947727	0.961993	449	21	25	417
Forêt Aléatoire	0.955044	0.958810	0.947964	0.953356	0.990560	452	18	23	419
Gradient Boosting Tree	0.966009	0.963883	0.966063	0.964972	0.994118	454	16	15	427
Naive Bayes	0.937500	0.926829	0.945701	0.936170	0.972451	437	33	24	418

TABLE 4.3 – Performance des modèles de classification avec Word2Vec sur données équilibrées

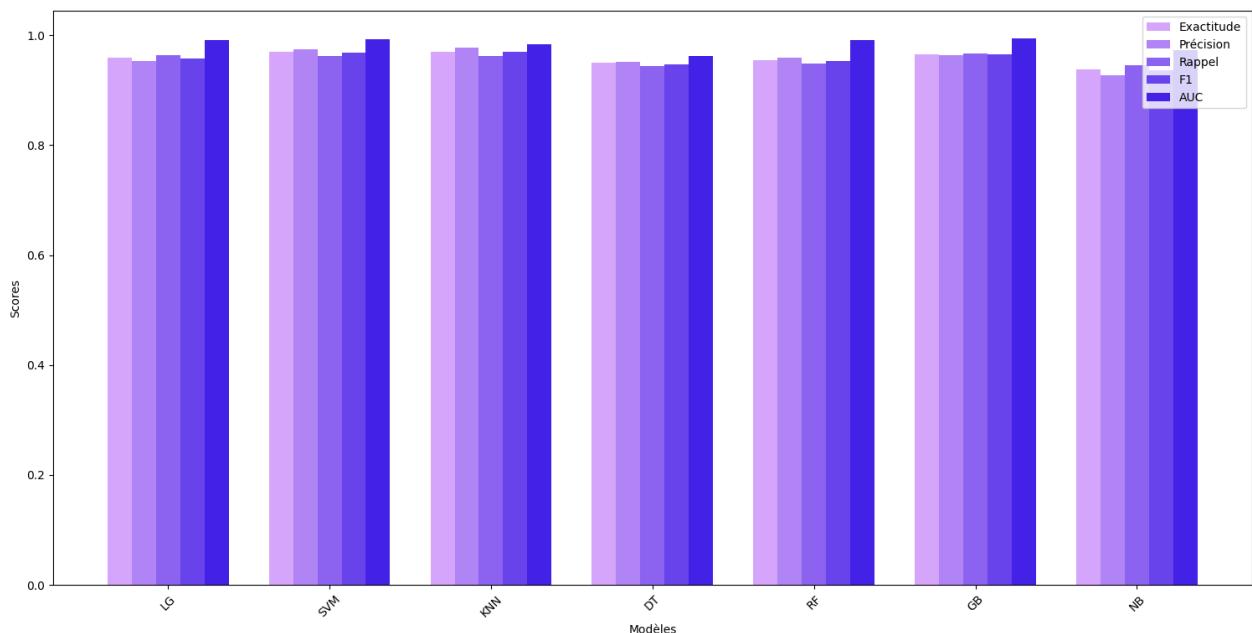


FIGURE 4.3 – Performances des différents modèles avec Word2vec sur données équilibrées

Le graphique en barres révèle pour les données extractées avec Word2vec que le SVM (précision : 97,48%, AUC : 99,28%), le KNN (précision : 97,70%, AUC : 98,41%) et le Gradient Boosting Tree (précision : 96,39%, AUC : 99,41%) se distinguent avec les meilleures performances en précision et AUC parmi les modèles évalués. Le Naive Bayes, bien qu'avec des scores inférieurs (précision : 92,68%, AUC : 97,25%), montre une stabilité dans ses résultats.

## 2. Données déséquilibrées avec dominance de données de phishing

### Extraction des caractéristiques avec TFIDF sans sélection des caractéristiques

Modèle	Exactitude	Précision	Rappel	F1	AUC	VN	FP	FN	VP
Régression Logistique	0.976885	0.859048	0.939583	0.897512	0.988498	3902	74	29	451
SVM	0.991023	0.988889	0.927083	0.956989	0.997374	3971	5	35	445
Arbre de Décision	0.962522	0.919571	0.714583	0.804220	0.887670	3946	30	137	343
Forêt Aléatoire	0.941203	1.000000	0.454167	0.624642	0.963896	3976	0	262	218
KNN	0.979129	0.957447	0.843750	0.897010	0.961627	3958	18	75	405
Gradient Boosting	0.981822	0.978417	0.850000	0.909699	0.987362	3967	9	72	408
Naive Bayes	0.979354	0.936937	0.866667	0.900433	0.993527	3948	28	64	416

TABLE 4.4 – Performance des modèles de classification avec TF-IDF sans sélection de caractéristiques sur données déséquilibrées1

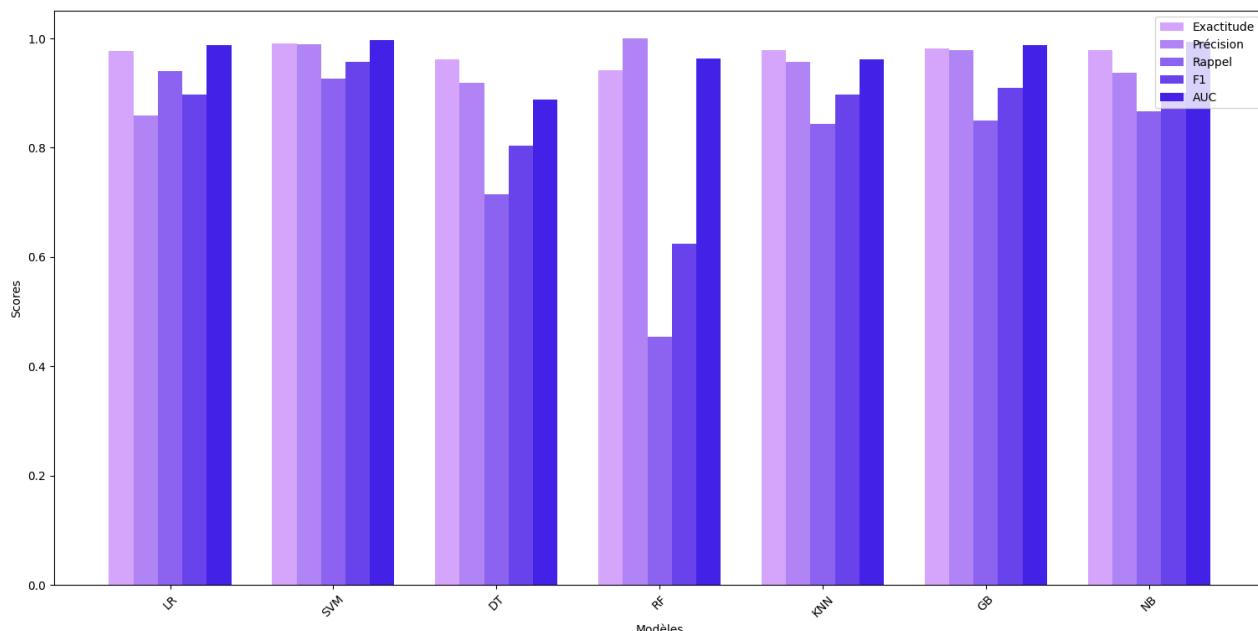


FIGURE 4.4 – Comparaison graphique des performances des modèles de classification avec TF-IDF sans sélection de caractéristiques sur données déséquilibrées1

Les performances des modèles de classification sans sélection de caractéristiques sur les données déséquilibrées avec dominance de données de phishing révèlent que le SVM se distingue avec une exactitude de 99.10%, une précision de 98.89%, et une AUC de 99.74%, surpassant les autres modèles dans la capacité à gérer les données déséquilibrées dominées par des emails de phishing.

### Extraction des caractéristiques avec TFIDF avec sélection des caractéristiques

Modèle	Exactitude	Précision	Rappel	F1	AUC	VN	FP	FN	VP
Régression Logistique	0.980476	0.886051	0.939583	0.912032	0.976562	3918	58	29	451
SVM	0.989004	0.979955	0.916667	0.947255	0.994255	3967	9	40	440
Arbre de Décision	0.962522	0.919571	0.714583	0.804220	0.887670	3946	30	137	343
Forêt Aléatoire	0.945916	1.000000	0.497917	0.664812	0.964808	3976	0	241	239
KNN	0.979129	0.923414	0.879167	0.900747	0.972874	3941	35	58	422
Gradient Boosting	0.981822	0.978417	0.850000	0.909699	0.987140	3967	9	72	408
Naive Bayes	0.979578	0.947126	0.858333	0.900546	0.993969	3953	23	68	412

TABLE 4.5 – Performance des modèles de classification avec TF-IDF et sélection de caractéristiques sur données déséquilibrées1

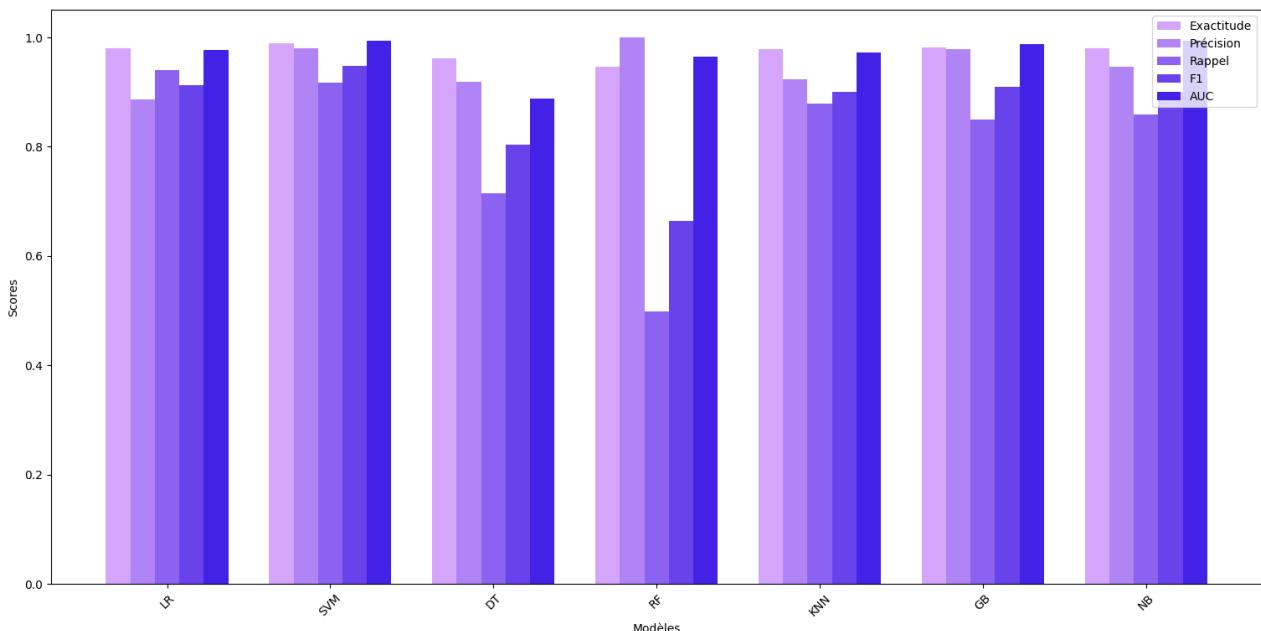


FIGURE 4.5 – Comparaison graphique des performances des modèles de classification avec TF-IDF et sélection de caractéristiques sur données déséquilibrées1

Les résultats montrent que le SVM se démarque avec une exactitude de 98.90%, une précision de 97.99%, et une AUC de 99.43%, ce qui en fait le meilleur modèle dans la capacité à discriminer les données déséquilibrées dominées par des emails de phishing. Les autres modèles présentent également de bonnes performances, notamment la Régression Logistique et le Gradient Boosting, bien qu'ils ne surpassent pas le SVM dans toutes les métriques clés.

### Extraction des caractéristiques avec Word2vec

Modèle	Exactitude	Précision	Rappel	F1	AUC	VN	FP	FN	VP
Régression Logistique	0.987657	0.953092	0.931250	0.942044	0.992808	3954	22	33	447
SVM	0.990350	0.986637	0.922917	0.953714	0.996977	3970	6	37	443
KNN	0.990126	0.965812	0.941667	0.953586	0.988748	3960	16	28	452
Arbre de Décision	0.980251	0.933628	0.879167	0.905579	0.968146	3946	30	58	422
Forêt Aléatoire	0.982271	0.969555	0.862500	0.912900	0.990005	3963	13	66	414
Gradient Boosting Tree	0.988555	0.975610	0.916667	0.945220	0.996536	3965	11	40	440
Naive Bayes	0.892280	0.000000	0.000000	0.000000	0.983141	3976	0	480	0

TABLE 4.6 – Performance des modèles de classification avec Word2Vec sur données déséquilibrées1

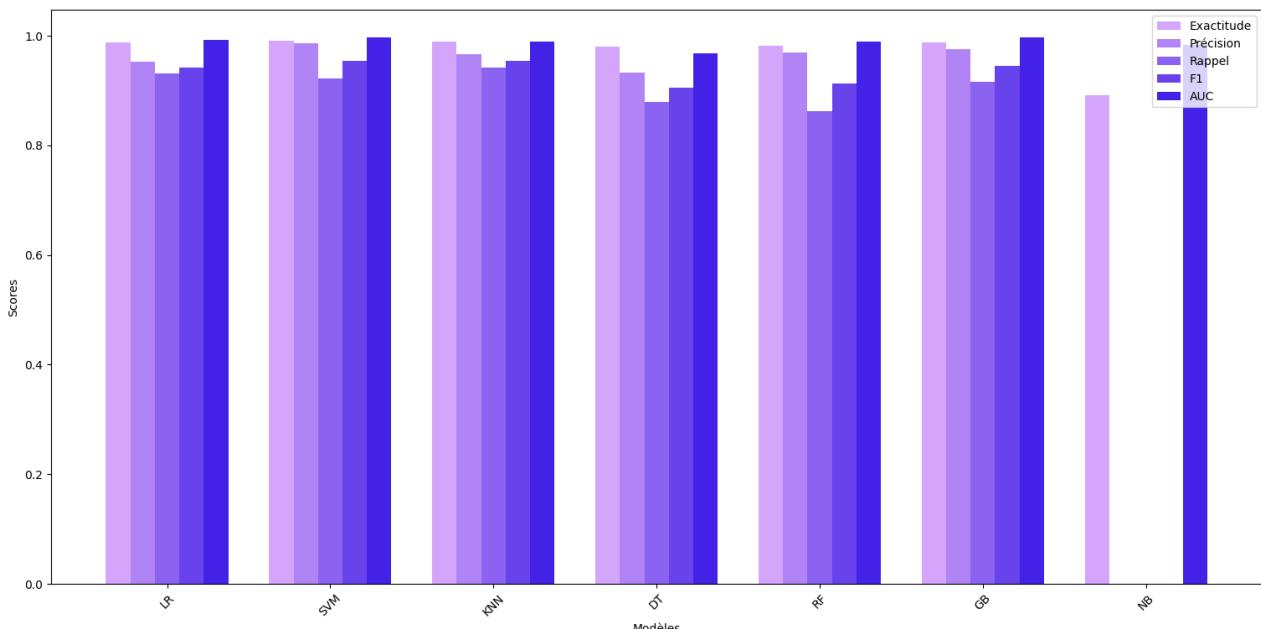


FIGURE 4.6 – Illustration graphique des performances des modèles avec Word2vec sur données déséquilibrées1

Les résultats montrent que le SVM et le KNN se distinguent avec les meilleures performances globales, présentant des scores élevés en exactitude, précision, rappel, F1-score et AUC. La régression logistique et le Gradient Boosting Tree suivent de près, avec des performances robustes dans toutes les métriques. En revanche, Naive Bayes montre des performances significativement inférieures dans toutes les métriques évaluées, en particulier avec une précision de 0 due à la nature déséquilibré des données.

### 3. Données déséquilibrées avec dominance de données légitimes

#### Extraction des caractéristiques avec TFIDF sans sélection des caractéristiques

Modèle	Exactitude	Précision	Rappel	F1	AUC	VN	FP	FN	VP
Régression Logistique	0.955793	0.965885	0.972103	0.968984	0.984160	174	16	13	453
SVM	0.971037	0.989059	0.969957	0.979415	0.991518	185	5	14	452
KNN	0.748476	0.740800	0.993562	0.848763	0.900136	28	16	23	463
Arbre de Décision	0.908537	0.953125	0.916309	0.934354	0.950175	169	21	39	427
Forêt Aléatoire	0.879573	0.855046	1.000000	0.921860	0.979444	111	79	0	466
Gradient Boosting Tree	0.960366	0.974138	0.969957	0.972043	0.991298	178	12	14	452
Naive Bayes	0.957317	0.967949	0.972103	0.970021	0.992388	175	15	13	453

TABLE 4.7 – Performance des modèles de classification avec TF-IDF sans sélection de caractéristiques sur données déséquilibrées2

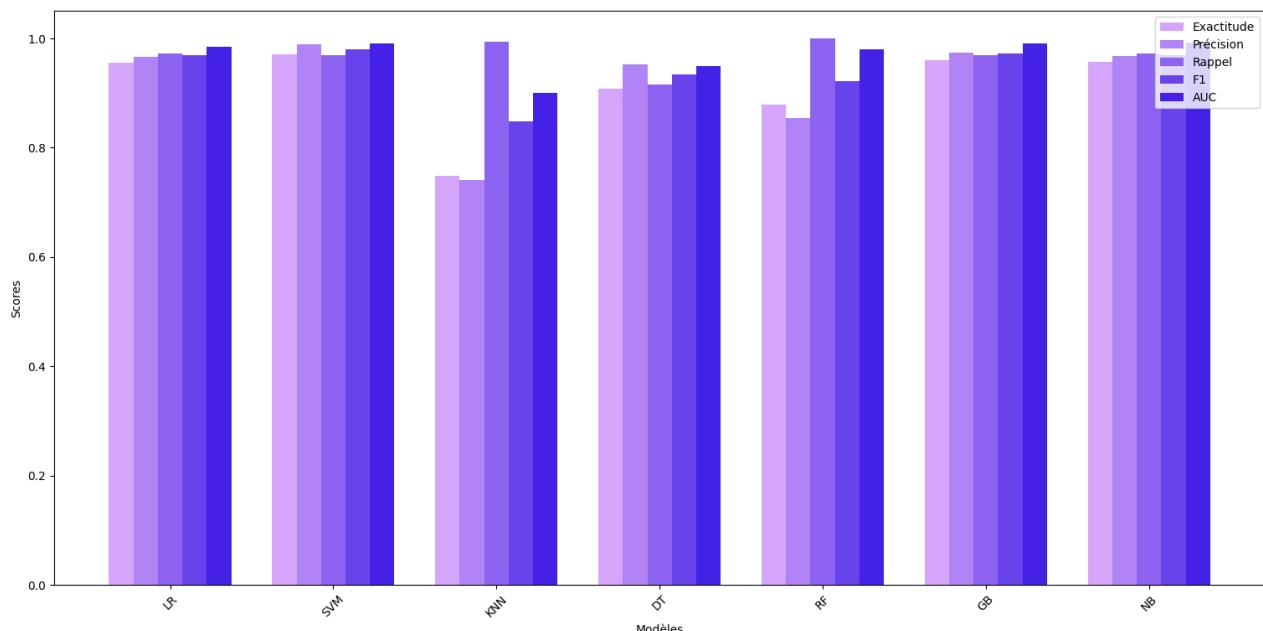


FIGURE 4.7 – Illustration graphique des résultats des modèles de classification avec TF-IDF sans sélection de caractéristiques sur données déséquilibrées2

Les modèles évalués sur les données extraites avec TFIDF sans sélection de caractéristiques ont montré des performances variées. Le SVM a obtenu la meilleure exactitude avec 97.10%, suivi de près par la régression logistique avec 95.58%. Cependant, le KNN a montré un rappel très élevé de 99.36%, mais cela s'accompagne d'une précision relativement faible de 74.08%, ce qui indique une propension à classifier de manière incorrecte certaines instances légitimes comme des phishing.

### Extraction des caractéristiques avec TFIDF avec sélection des caractéristiques

Modèle	Exactitude	Précision	Rappel	F1	AUC	VN	FP	FN	VP
Régression Logistique	0.946646	0.957537	0.967811	0.962647	0.970256	170	20	15	451
SVM	0.961890	0.976242	0.969957	0.973089	0.991603	179	11	14	452
KNN	0.847561	0.827957	0.991416	0.902344	0.950672	94	96	4	462
Arbre de Décision	0.891768	0.960373	0.884120	0.920670	0.951779	173	17	54	412
Forêt Aléatoire	0.891768	0.87054	0.995708	0.928929	0.981731	121	69	2	464
Gradient Boosting Tree	0.952744	0.971800	0.961373	0.966559	0.990338	177	13	18	448
Naive Bayes	0.961890	0.964211	0.982833	0.973433	0.992393	173	17	8	458

TABLE 4.8 – Performance des modèles de classification avec TF-IDF et sélection de caractéristiques sur données déséquilibrées2

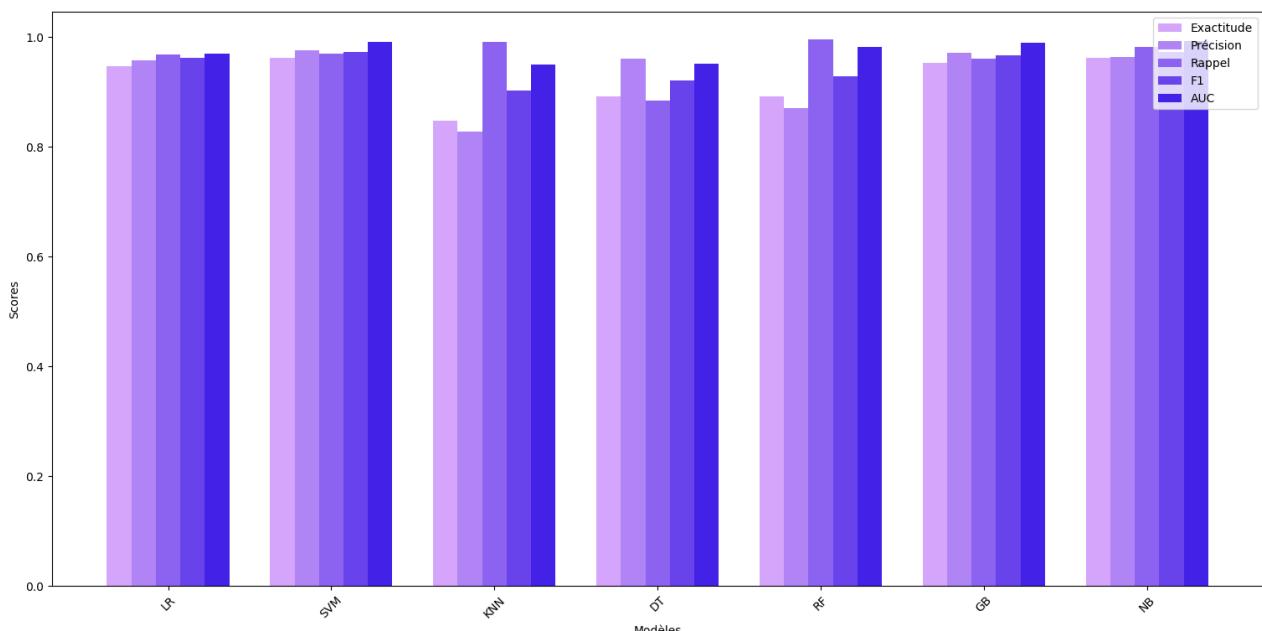


FIGURE 4.8 – Illustration graphique des résultats des modèles de classification avec TF-IDF et selection de caractéristiques sur données déséquilibrées2

Avec TFIDF et une sélection de caractéristiques, les performances des modèles se sont améliorées dans l'ensemble. Le SVM a continué à dominer avec une exactitude de 96.19% et une précision de 97.62%. Le KNN a montré une amélioration significative en précision (82.80%) tout en maintenant un rappel élevé de 99.14%, ce qui suggère une meilleure capacité à classifier correctement les instances positives tout en minimisant les faux positifs.

### Extraction des caractéristiques avec Word2vec

Modèle	Exactitude	Précision	Rappel	F1	AUC	VN	FP	FN	VP
Régression Logistique	0.964939	0.974304	0.976395	0.975348	0.991597	178	12	11	455
SVM	0.974085	0.986985	0.976395	0.981661	0.992015	184	6	11	455
KNN	0.966463	0.984716	0.967811	0.976190	0.987091	183	7	15	451
Arbre de Décision	0.954268	0.969828	0.965665	0.967742	0.957505	176	14	16	450
Forêt Aléatoire	0.967988	0.974414	0.980687	0.977540	0.994793	178	12	9	457
Gradient Boosting Tree	0.969512	0.974468	0.982833	0.978632	0.991484	178	12	8	458
Naive Bayes	0.885671	0.868173	0.989270	0.924774	0.983691	120	70	5	461

TABLE 4.9 – Performance des modèles de classification avec Word2Vec sur données déséquilibrées2

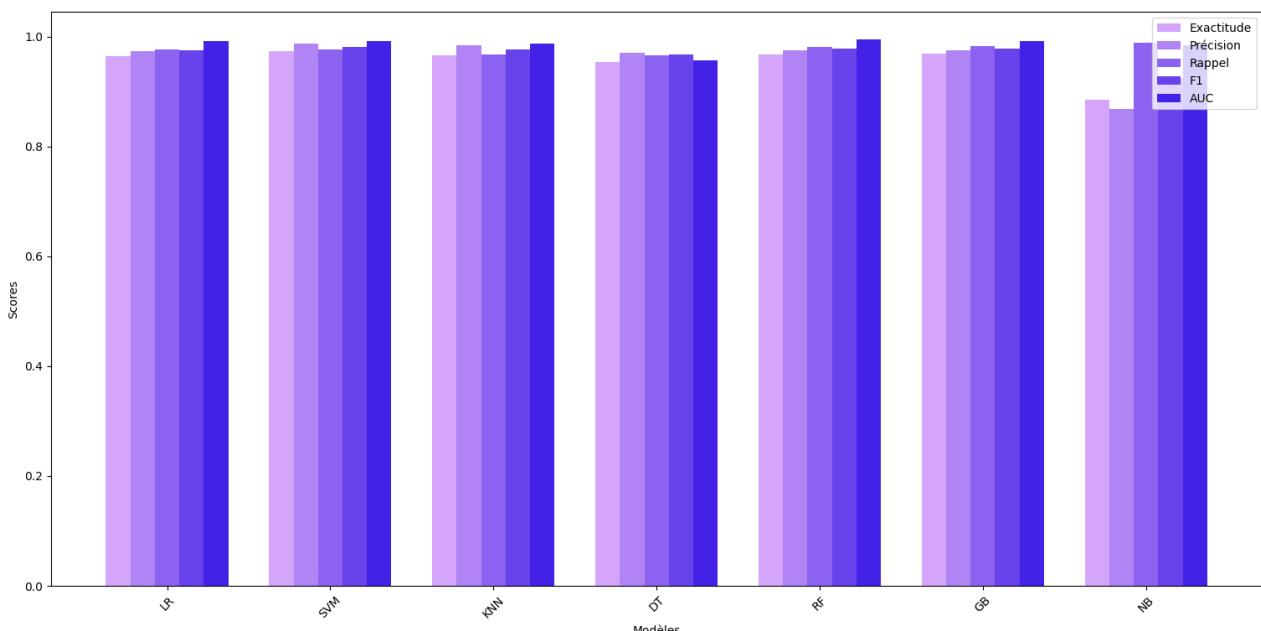


FIGURE 4.9 – Comparaison des résultats des modèles avec Word2vec sur données déséquilibrées2

L'utilisation de Word2vec pour extraire les caractéristiques a produit des résultats prometteurs. Le SVM a atteint une exactitude de 97.41% et une précision impressionnante de 98.70%, avec un rappel et un F1-score élevés (97.64% et 98.17%, respectivement). Cela indique une bonne capacité à généraliser et à bien performer sur les données de phishing, démontrant ainsi l'efficacité de Word2vec dans la représentation sémantique des textes pour la classification.

## 4. Analyse des résultats des Modèles de Machine Learning

### 4.1 Comparaison des Jeux de Données

Ensemble de Données	Résultats	Modèles Performants
Données Équilibrées	Meilleure performance globale avec des F1 scores élevés (entre 0.890 et 0.969). Tous les modèles prédisent bien les classes positives et négatives.	SVM, Naive Bayes, Gradient Boosting
Données Déséquilibrées (Dominance Phishing)	Performance générale inférieure, surtout pour le rappel et le F1 score du phishing. SVM et Gradient Boosting sont plus résistants aux déséquilibres avec des scores corrects. La précision est souvent plus élevée que le rappel à cause du déséquilibre des classes.	SVM, Gradient Boosting
Données Déséquilibrées (Dominance Légitime)	Performances similaires aux données équilibrées pour la classe légitime. Diminution du rappel et du F1 score pour le phishing, indiquant des difficultés à prédire les cas de phishing. KNN et Naive Bayes sont moins performants.	SVM, Gradient Boosting, Naive Bayes (moins performant)

TABLE 4.10 – Comparaison des Performances sur Différents Jeux de Données

### 4.2 Comparaison entre TFIDF et Word2vec

Méthode d'Extraction de Caractéristiques	Performances Générales	Meilleurs Modèles
TFIDF	Bonnes sur données équilibrées, moins bonnes sur données déséquilibrées	SVM, Gradient Boosting
Word2vec	Amélioration notable, surtout sur données déséquilibrées (dominance phishing)	Régression Logistique, SVM, KNN

TABLE 4.11 – Comparaison des Méthodes d'Extraction de Caractéristiques

### 4.3 Comparaison avec sélection et sans sélection de caractéristiques

Sélection de Caractéristiques	Performances Générales
Sans Sélection	Bonnes sur données équilibrées, mais surajustement possible. Moins efficace sur données déséquilibrées (Random Forest sensible aux déséquilibres).
Avec Sélection	Amélioration notable, surtout sur données déséquilibrées. Régression Logistique, SVM et Gradient Boosting en bénéficiant le plus.

TABLE 4.12 – Comparaison avec et sans Sélection de Caractéristiques

En résumé, l'utilisation de Word2vec pour l'extraction de caractéristiques semble bénéficier aux performances des modèles, surtout sur des jeux de données déséquilibrés ou lorsque la tâche nécessite une meilleure représentation sémantique des caractéristiques. De plus, la sélection de caractéristiques améliore souvent la robustesse et les performances des modèles, en particulier sur des jeux de données complexes ou déséquilibrés.

## 4.2.2 Résultats des Modèles de Deep Learning

Dans cette section, nous mettons en évidence la comparaison de trois techniques de deep learning, à savoir le Perceptron Multicouche (MLP), le Réseau de Neurones Convolutif (CNN) et le Réseau de Neurones Récurrent (RNN). Ces modèles ont été entraînés sur des données équilibrées et déséquilibrées, en utilisant l'extraction de caractéristiques par TF-IDF, et en comparant les résultats avec et sans utilisation de la sélection de caractéristiques par chi-square.

### 1. Données Équilibrées

Le Tableau 4.13 montre les résultats obtenus en utilisant TF-IDF pour l'extraction des caractéristiques sans sélection de caractéristiques par chi-square. Le Tableau 4.14 présente les résultats en utilisant TF-IDF avec la sélection de caractéristiques par chi-square.

Les matrices de confusion correspondantes pour les modèles MLP, CNN et RNN sont illustrées aux Figures 4.10 et 4.13 pour MLP, aux Figures 4.11 et 4.14 pour CNN, et aux Figures 4.13 et 4.15 pour RNN.

Modèle	Accuracy	Precision	Recall	F1	AUC	TN	FP	FN	TP
MLP	0.968	0.966	0.968	0.967	0.992	455	15	14	428
CNN	0.957	0.961	0.950	0.955	0.987	453	17	22	420
RNN	0.808	0.802	0.800	0.801	0.892	383	87	88	354

TABLE 4.13 – Performance des modèles de réseaux de neurones sans sélection de caractéristiques sur données équilibrées

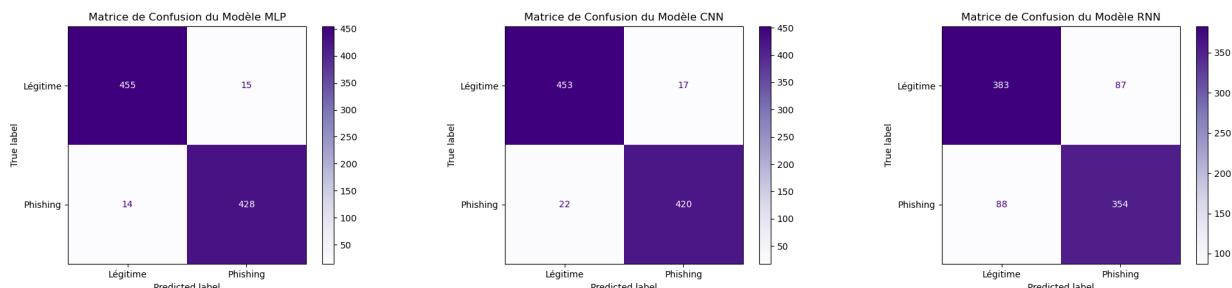


FIGURE 4.10 – Matrice de confusion du modèle MLP sans sélection de caractéristiques sur données équilibrées

FIGURE 4.11 – Matrice de confusion du modèle CNN sans sélection de caractéristiques sur données équilibrées

FIGURE 4.12 – Matrice de confusion du modèle RNN sans sélection de caractéristiques sur données équilibrées

Modèle	Accuracy	Precision	Recall	F1	AUC	TN	FP	FN	TP
MLP	0.966	0.964	0.966	0.965	0.993	454	16	15	427
CNN	0.953	0.959	0.943	0.951	0.987	452	18	25	417
RNN	0.815	0.834	0.771	0.801	0.902	402	68	101	341

TABLE 4.14 – Performance des modèles de réseaux de neurones avec sélection de caractéristiques sur données équilibrées

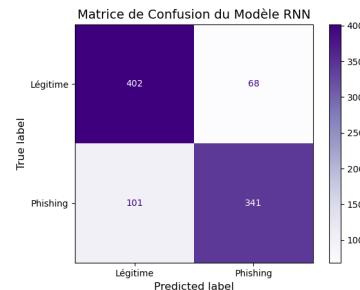
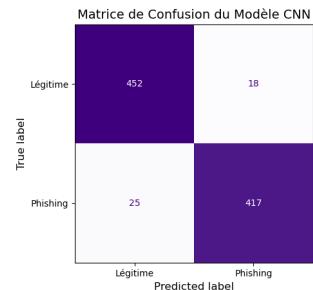
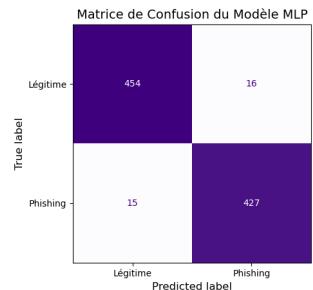


FIGURE 4.13 – Matrice de confusion du modèle MLP avec sélection de caractéristiques sur données équilibrées

FIGURE 4.14 – Matrice de confusion du modèle CNN avec sélection de caractéristiques sur données équilibrées

FIGURE 4.15 – Matrice de confusion du modèle RNN avec sélection de caractéristiques sur données équilibrées

Pour les données équilibrées, le modèle MLP montre les meilleures performances globales avec une exactitude de 0.968 et un F1-score de 0.967 sans sélection de caractéristiques. La sélection de caractéristiques par chi-square réduit légèrement les performances du MLP, mais il reste le modèle le plus performant. Les modèles CNN et RNN affichent des performances inférieures, avec des baisses significatives dans leurs métriques de performance.

## 2. Données déséquilibrées avec dominance de données de phishing

Le Tableau 4.15 montre les résultats obtenus en utilisant TF-IDF pour l'extraction des caractéristiques sans sélection de caractéristiques par chi-square. Le Tableau 4.16 présente les résultats en utilisant TF-IDF avec la sélection de caractéristiques par chi-square. Les matrices de confusion correspondantes pour les modèles MLP, CNN et RNN sont illustrées aux Figures 4.16 et 4.19 pour MLP, aux Figures 4.17 et 4.20 pour CNN, et aux Figures 4.18 et 4.21 pour RNN.

Modèle	Accuracy	Precision	Recall	F1	AUC	TN	FP	FN	TP
MLP	0.991	0.989	0.929	0.958	0.998	3971	5	34	446
CNN	0.985	0.931	0.933	0.932	0.993	3943	33	32	448
RNN	0.892	0.000	0.000	0.000	0.446	3976	0	480	0

TABLE 4.15 – Performance des modèles de réseaux de neurones sans sélection des caractéristiques sur données déséquilibrées1

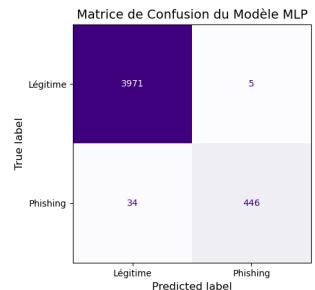


FIGURE 4.16 – Matrice de confusion du modèle MLP sans sélection des caractéristiques sur données déséquilibrées1

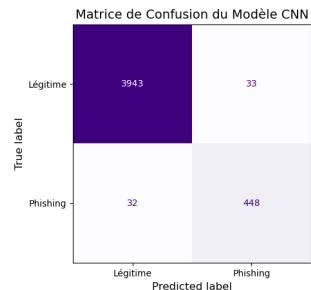


FIGURE 4.17 – Matrice de confusion du modèle CNN sans sélection des caractéristiques sur données déséquilibrées1

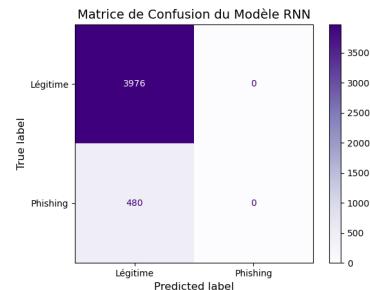


FIGURE 4.18 – Matrice de confusion du modèle RNN sans sélection des caractéristiques sur données déséquilibrées1

Modèle	Accuracy	Precision	Recall	F1	AUC	TN	FP	FN	TP
MLP	0.990	0.988	0.920	0.953	0.996	3971	5	38	442
CNN	0.986	0.960	0.916	0.938	0.993	3958	18	40	440
RNN	0.892	0.000	0.000	0.000	0.690	3976	0	480	0

TABLE 4.16 – Performance des modèles de réseaux de neurones avec sélection des caractéristiques sur données déséquilibrées1

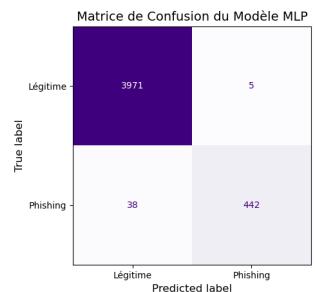


FIGURE 4.19 – Matrice de confusion du modèle MLP avec sélection des caractéristiques sur données déséquilibrées1

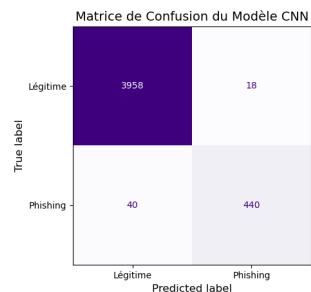


FIGURE 4.20 – Matrice de confusion du modèle CNN avec sélection des caractéristiques sur données déséquilibrées1

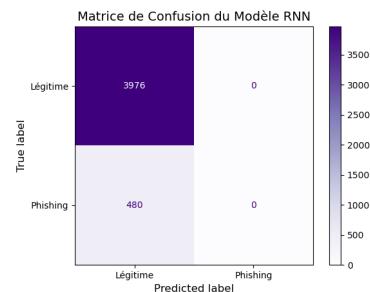


FIGURE 4.21 – Matrice de confusion du modèle RNN avec sélection des caractéristiques sur données déséquilibrées1

Dans le scénario de données déséquilibrées dominées par des e-mails de phishing, le MLP reste supérieur avec une exactitude de 0.991 et un F1-score de 0.958 sans sélection de caractéristiques. L'application de la sélection de caractéristiques par chi-square réduit légèrement les performances du MLP mais il maintient une bonne performance. Le CNN améliore sa précision mais au détriment du rappel, tandis que le RNN n'affiche pas de performances notables, restant inférieur aux autres modèles.

### 3. Données déséquilibrées avec dominance de données légitimes

Le Tableau 4.17 montre les résultats obtenus en utilisant TF-IDF pour l'extraction des caractéristiques sans sélection de caractéristiques par chi-square. Le Tableau 4.18 présente les

résultats en utilisant TF-IDF avec la sélection de caractéristiques par chi-square.

Les matrices de confusion correspondantes pour les modèles MLP, CNN et RNN sont illustrées aux Figures 4.22 et 4.25 pour MLP, aux Figures 4.23 et 4.26 pour CNN, et aux Figures 4.24 et 4.27 pour RNN.

Modèle	Accuracy	Precision	Recall	F1	AUC	TN	FP	FN	TP
MLP	0.966	0.991	0.961	0.976	0.996	186	4	18	448
CNN	0.946	0.975	0.948	0.961	0.985	179	11	24	442
RNN	0.710	0.710	1.000	0.830	0.589	0	190	0	466

TABLE 4.17 – Performance des modèles de réseaux de neurones sans sélection des caractéristiques sur données déséquilibrées2

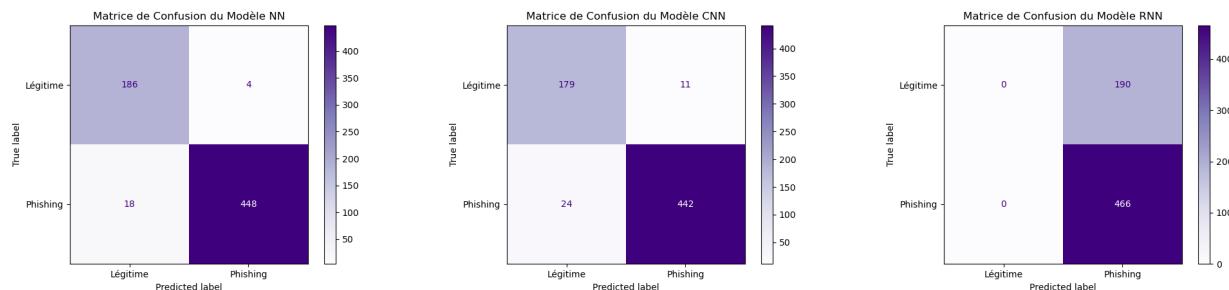


FIGURE 4.22 – Matrice de confusion du modèle MLP sans sélection des caractéristiques sur données déséquilibrées2

FIGURE 4.23 – Matrice de confusion du modèle CNN sans sélection des caractéristiques sur données déséquilibrées2

FIGURE 4.24 – Matrice de confusion du modèle RNN sans sélection des caractéristiques sur données déséquilibrées2

Modèle	Accuracy	Precision	Recall	F1	AUC	TN	FP	FN	TP
MLP	0.971	0.982	0.976	0.979	0.994	182	8	11	455
CNN	0.955	0.961	0.976	0.969	0.990	172	18	11	455
RNN	0.818	0.830	0.935	0.879	0.890	101	89	30	436

TABLE 4.18 – Performance des modèles de réseaux de neurones avec sélection des caractéristiques sur données déséquilibrées2

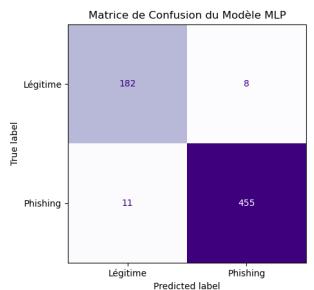


FIGURE 4.25 – Matrice de confusion du modèle MLP avec sélection des caractéristiques sur données déséquilibrées2

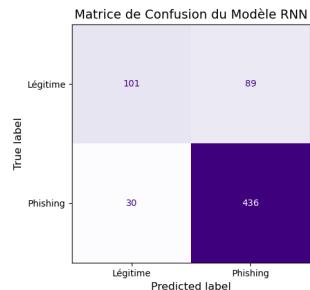


FIGURE 4.26 – Matrice de confusion du modèle CNN avec sélection des caractéristiques sur données déséquilibrées2

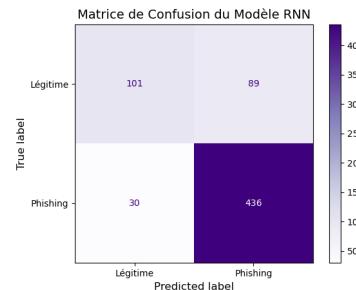


FIGURE 4.27 – Matrice de confusion du modèle RNN avec sélection des caractéristiques sur données déséquilibrées2

Pour les données déséquilibrées dominées par des e-mails légitimes, le MLP continue de dominer avec une exactitude de 0.966 et un F1-score de 0.976 sans sélection de caractéristiques. La sélection de caractéristiques améliore encore les performances du MLP, atteignant une exactitude de 0.971 et un F1-score de 0.979. Le CNN montre également des améliorations notables, mais le RNN reste en dessous des performances des MLP et CNN, même après sélection de caractéristiques.

En conclusion, le modèle MLP s'est révélé être le plus performant dans la plupart des scénarios, notamment pour les données équilibrées et les données déséquilibrées dominées par des e-mails légitimes. La sélection de caractéristiques par chi-square a des effets variables sur les performances des modèles selon le type de données utilisé. Le CNN, bien que performant dans certains cas, reste généralement inférieur au MLP. Le RNN montre des performances inférieures dans tous les scénarios testés. Ces observations indiquent que le MLP est le modèle le plus robuste pour la détection de phishing par e-mail dans les contextes de données étudiés.

### 4.2.3 Résultats de l'analyse sémantique du corps

Dans cette partie, nous avons découvert l'importance de l'analyse sémantique, qui sera combinée par la suite avec les résultats du modèle choisi pour générer le score du corps. Nous avons opté pour SVM avec TF-IDF et sélection de caractéristiques basée sur le chi-square en raison de leurs performances très élevées. Voici les résultats de l'analyse sémantique sur notre ensemble de données.

Index	Classe	Score de l'analyse
0	1	0.750
1	1	0.325
2	1	0.750
3	1	0.625
4	1	0.750
4553	0	0.325
4554	0	0.450
4555	0	0.125
4556	0	0.325
4557	0	0.325

TABLE 4.19 – Résultats de l'analyse sémantique du corps

## 4.3 Résultats de l'Analyse des En-têtes et des Liens

L'analyse des en-têtes et des liens des e-mails a constitué une étape cruciale dans la détection des caractéristiques suspectes. Les résultats obtenus lors de ces analyses sont présentés ci-dessous :

### 4.3.1 Analyse des En-têtes

Les scores élevés des en-têtes (voir Tableau 4.20) indiquent une forte probabilité que les e-mails soient de type phishing, ce qui est encourageant pour la détection des tentatives de phishing. Cependant, l'efficacité de cette méthode est moindre pour les e-mails légitimes, nécessitant ainsi une amélioration pour une détection plus précise de cette catégorie d'e-mails.

Index	Classe	Score d'en-tête
0	1	0.9
1	1	0.9
2	1	0.9
3	1	0.9
4	1	0.9
4553	0	0.6
4554	0	0.6
4555	0	0.6
4556	0	0.6
4557	0	0.6

TABLE 4.20 – Résultats de l'analyse des en-têtes

Nous procédons maintenant à examiner la distribution des scores attribués aux en-têtes dans notre échantillon d'e-mails. Le diagramme présenté à la Figure ci dessous illustre cette distribution.

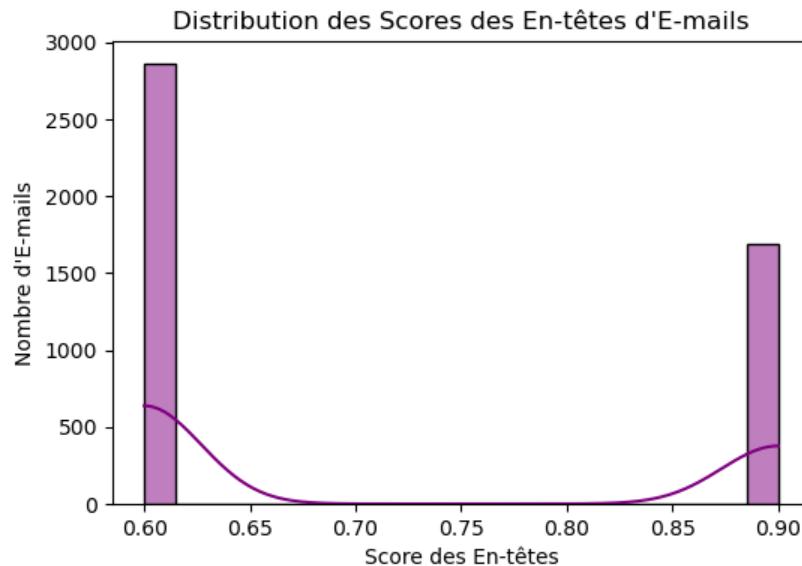


FIGURE 4.28 – Distribution des Scores des En-têtes d'E-mails

En analysant la figure, il est clair que l'analyse des scores pour les e-mails légitimes est moins précise, avec une distribution moins concentrée et des scores plus variables. En revanche, une partie significative des e-mails de phishing est détectée avec des scores élevés, montrant une capacité robuste du modèle pour cette catégorie spécifique d'e-mails malveillants.

### 4.3.2 Analyse des Liens

Les scores élevés des liens (voir Tableau 4.21) indiquent grande probabilité de phishing. Cela suggère que les caractéristiques analysées, telles que la longueur du domaine, l'utilisation de raccourisseurs d'URL, et la présence de caractères inhabituels, sont des indicateurs efficaces pour identifier les tentatives de phishing.

Index	Classe	Link Score
0	1	0.977778
1	1	0.866667
2	1	0.977778
3	1	0.977778
4	1	0.977778
4553	0	0
4554	0	0
4555	0	0
4556	0	0
4557	0	0

TABLE 4.21 – Résultats de l'analyse des liens

Nous procédons maintenant à examiner la distribution des scores attribués aux liens dans notre échantillon d'e-mails. Le diagramme présenté à la Figure 4.29 illustre cette distribution.

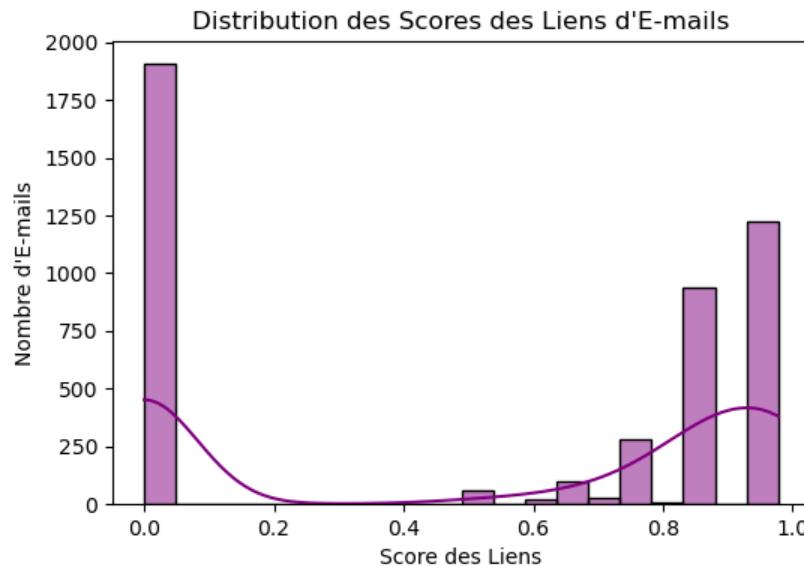


FIGURE 4.29 – Distribution des Scores des Liens d’E-mails

Ce diagramme illustre clairement que la majorité des scores des liens sont concentrés autour de 0.0, indiquant que la plupart des e-mails sont perçus comme légitimes. Néanmoins, une proportion significative de scores proches de 1.0 montre que certains e-mails sont identifiés comme potentiellement malveillants.

En conclusion, l’intégration de l’analyse des liens a significativement renforcé la capacité du modèle à identifier les e-mails malveillants, surpassant l’analyse des en-têtes en termes de précision. Toutefois, des possibilités d’amélioration subsistent afin d’assurer une détection encore plus robuste et précise.

#### 4.3.3 Combinaison des scores

Pour améliorer la détection des e-mails de phishing, nous avons combiné les scores issues de l’analyse des en-têtes, du corps de l’e-mail, et des liens contenus dans l’e-mail. Cette approche permet de bénéficier des forces de chaque type d’analyse pour obtenir une détection plus robuste.

Vraie Classe	Score de phishing
1	0.933333
1	1.000000
1	0.877778
1	0.729093
1	0.729093
0	0.212845
0	0.416557
0	0.314293
0	0.257512
0	0.000000

TABLE 4.22 – Résultats de combinaison des scores

En analysant la distribution des scores combinés, il est clair que cette combinaison améliore la précision de la détection des e-mails de phishing, en particulier pour les e-mails légitimes. La

distribution des scores combinés montre une meilleure séparation entre les e-mails de phishing et les e-mails légitimes.

En conclusion, l'intégration de l'analyse combinée des caractéristiques a considérablement renforcé la capacité du modèle à identifier les e-mails malveillants, surpassant les analyses individuelles en termes de précision et de robustesse.

### 4.4 Discussion

Les résultats obtenus révèlent les nuances et les performances variées des différentes approches de modélisation et techniques d'analyse utilisées dans la détection de phishing par e-mail. Chaque méthode présente ses propres forces et faiblesses, ce qui souligne l'importance de choisir la bonne approche en fonction des caractéristiques spécifiques des données et des objectifs de détection.

Les modèles traditionnels de machine learning, tels que SVM utilisant TF-IDF, se distinguent par leur robustesse et leur capacité à gérer les équilibres de données. Ces modèles sont efficaces dans la détection de schémas de phishing classiques où les e-mails malveillants sont moins fréquents mais potentiellement plus dangereux. Cependant, leur efficacité peut être considérablement améliorée avec une sélection de caractéristiques appropriée, comme illustré dans nos expériences.

À l'inverse, les modèles de deep learning, comme les réseaux de neurones multicouches (MLP), se révèlent particulièrement performants dans des contextes où les données sont plus équilibrées ou dominées par des e-mails légitimes. Ces modèles exploitent la capacité des réseaux de neurones à capturer des interactions complexes entre les caractéristiques des e-mails, ce qui est crucial pour la détection de variations subtiles de phishing. Cependant, ils nécessitent souvent des volumes de données plus importants et des ressources computationnelles considérables pour être pleinement efficaces.

L'analyse spécifique des en-têtes et des liens dans les e-mails s'avère prometteuse mais nécessite encore des améliorations, notamment en termes de précision pour la classification des e-mails légitimes. L'intégration et l'optimisation combinées de différentes caractéristiques semblent être une voie prometteuse pour une détection plus robuste et précise du phishing par e-mail. Cette approche hybride permet de tirer parti des avantages complémentaires des méthodes traditionnelles de machine learning et des capacités d'apprentissage profond des réseaux neuronaux.

### 4.5 Conclusion

Ce chapitre a présenté les résultats et la discussion des différentes méthodes mises en œuvre pour détecter les e-mails de phishing. Pour une détection efficace des e-mails malveillants, une approche hybride combinant des modèles de machine learning traditionnels ou des techniques de deep learning avec une analyse sémantique approfondie, ainsi qu'une analyse intégrée des caractéristiques comme les en-têtes et les liens, semble être la solution la plus prometteuse. Cela permettrait de maximiser la précision tout en assurant une détection robuste même en présence de déséquilibres dans les données.

# Conclusion et perspectives

## Résumé des contributions du projet

Le projet de détection de phishing par e-mail représente une contribution significative dans la lutte contre la cybercriminalité. Il s'est concentré sur le développement d'une solution robuste et fiable en utilisant des méthodes avancées d'analyse du langage naturel (NLP) et de machine learning.

Les principales contributions du projet incluent :

**1. Utilisation d'un ensemble de données diversifié** : Le projet a utilisé un ensemble de données public varié pour l'entraînement des modèles, permettant ainsi une représentation exhaustive des e-mails de phishing et légitimes.

**2. Prétraitement des données** : Une étape cruciale de nettoyage et de préparation des données a été réalisée pour assurer la qualité des données utilisées dans l'entraînement des modèles.

**3. Création d'échantillons équilibrés et non équilibrés** : Deux ensembles de données distincts ont été créés, avec une répartition équilibrée et non équilibrée entre les e-mails de phishing et les e-mails légitimes. Cela a permis de tester et de comparer les performances des modèles dans différents scénarios.

**4. Extraction et sélection de caractéristiques** : Les caractéristiques pertinentes des e-mails ont été extraites et sélectionnées en utilisant des techniques telles que TF-IDF et le test du chi-deux. Cette étape a été essentielle pour améliorer la précision et l'efficacité des modèles de détection.

**5. Entraînement des modèles de machine learning** : Plusieurs modèles, incluant SVM et autres réseaux de neurones, ont été entraînés et évalués sur les données préparées. Des techniques avancées telles que l'analyse sémantique du corps de l'e-mail ont été intégrées pour améliorer la précision des prédictions.

**6. Analyse approfondie des e-mails** : En plus de l'analyse du contenu des e-mails, une attention particulière a été portée à l'analyse des liens et des en-têtes, renforçant ainsi la capacité de la solution à détecter les signaux de phishing.

**7. Production d'une solution intégrée de détection** : La combinaison de tous ces éléments a conduit à une solution intégrée capable de détecter efficacement les e-mails de phishing, offrant une protection avancée contre les menaces numériques.

En résumé, ce projet a non seulement abouti à une technologie de détection avancée, mais il a également enrichi la compréhension des techniques de sécurité informatique dans le domaine de la détection de phishing par e-mail.

## Discussion sur l'application de visualisation des résultats

L'application développée pour visualiser les résultats de notre modèle de détection de phishing joue un rôle central dans ce projet. Conçue avec une interface conviviale, elle permet aux utilisateurs de soumettre des e-mails pour analyse et d'interpréter les résultats de classification en temps réel. Cette application sert plusieurs objectifs essentiels :

**Accessibilité et facilité d'utilisation :** En offrant une interface intuitive, l'application rend la détection de phishing accessible même à des utilisateurs non techniques. Ils peuvent facilement comprendre les prédictions du modèle sans nécessiter de connaissances spécialisées en apprentissage automatique ou en sécurité informatique.

**Transparence et confiance :** En permettant aux utilisateurs de voir directement comment leurs e-mails sont classés (phishing ou légitimes), l'application renforce la transparence du processus de détection. Cela contribue à accroître la confiance des utilisateurs dans la fiabilité et l'efficacité de notre solution.

**Utilisation en temps réel :** Grâce à son fonctionnement en temps réel, l'application permet une évaluation instantanée des e-mails soumis. Cela est particulièrement précieux dans des environnements où une réponse rapide aux menaces de phishing est cruciale pour prévenir les attaques.

**Amélioration continue :** En recueillant les retours des utilisateurs sur l'efficacité des classifications et sur l'expérience d'utilisation, l'application facilite également l'amélioration continue de notre modèle de détection. Les commentaires des utilisateurs peuvent être utilisés pour ajuster et optimiser les performances du modèle au fil du temps.

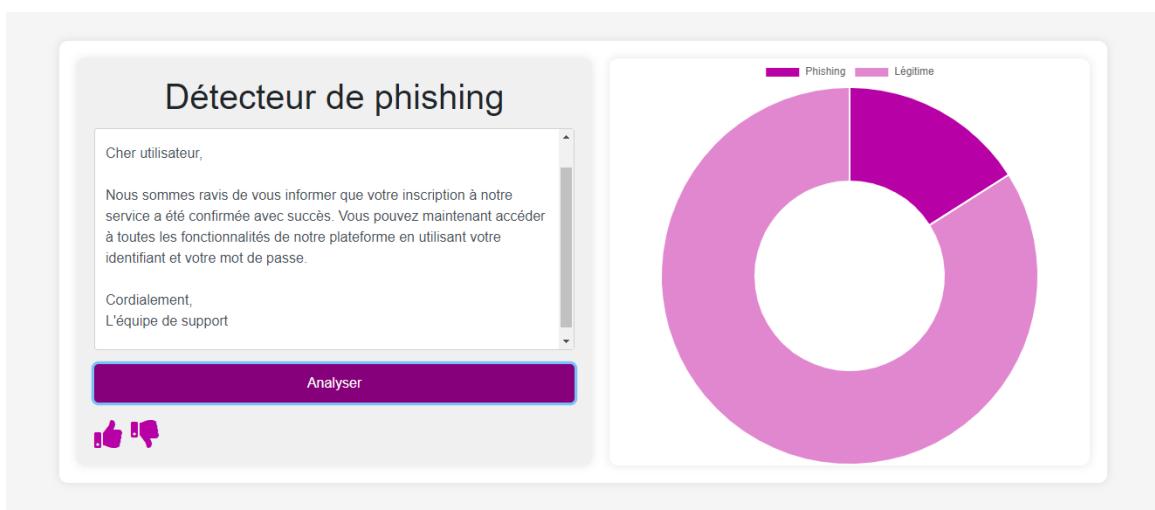


FIGURE 4.30 – Screenshot de l'application de visualisation des résultats

L'application de visualisation des résultats représente donc une composante essentielle de

notre projet, en complément des aspects techniques avancés de la détection de phishing par e-mail. Elle incarne notre engagement à fournir une solution pratique, efficace et accessible pour renforcer la sécurité des communications électroniques.

## Réponses aux objectifs initialement posés

Le projet a été conçu pour répondre de manière exhaustive aux questions et objectifs définis initialement. L'évaluation des résultats obtenus à partir de l'analyse des e-mails de phishing démontre une performance remarquable avec un faible taux d'erreurs de prédiction. Cette performance élevée confirme la pertinence et la fiabilité de notre solution.

Pour atteindre ces résultats, nous avons procédé à une analyse approfondie des e-mails de phishing à l'aide de techniques avancées d'apprentissage automatique et de traitement du langage naturel. Notre modèle a été spécifiquement entraîné pour détecter efficacement les tentatives de phishing tout en minimisant les faux positifs, assurant ainsi une protection robuste contre les attaques malveillantes.

L'excellente performance de notre solution renforce sa valeur stratégique dans la sécurisation des communications électroniques. En conséquence, nous sommes confiants dans le fait que notre approche offre une réponse efficace et proactive aux défis croissants posés par les cybermenaces.

## Les travaux futurs

Pour améliorer notre solution actuelle, plusieurs axes de développement sont envisagés. Premièrement, nous prévoyons d'élargir notre corpus de données en incluant une plus grande diversité de scénarios de phishing afin d'améliorer la robustesse et la généralisation de notre modèle. Deuxièmement, nous chercherons à affiner nos algorithmes d'apprentissage automatique pour réduire encore le taux d'erreurs de prédiction, en particulier en minimisant les faux positifs. Parallèlement, nous visons à développer une capacité de détection en temps réel des e-mails de phishing pour une réponse proactive et immédiate. En explorant ces avenues, nous ambitionnons de renforcer notre défense contre les menaces persistantes tout en maintenant une haute précision et une efficacité opérationnelle.

En outre, nous examinerons les possibilités d'intégrer des techniques de détection multimodales, combinant texte, image et autres modalités pour une détection plus exhaustive et précise des tentatives de phishing. Nous prévoyons également d'intégrer des mécanismes de sécurité avancés pour contrer les attaques sophistiquées et émergentes, assurant ainsi une protection continue et proactive de nos systèmes et données critiques. Enfin, nous mettrons en place un processus d'évaluation continue pour surveiller et optimiser la performance de notre solution dans des environnements dynamiques, garantissant ainsi sa pertinence et son efficacité à long terme.

# Bibliographie

- [1] Apprentissage non-supervisé : définition et algorithmes populaires. <https://www.journaldunet.fr/intelligence-artificielle/guide-de-l-intelligence-artificielle/1501309-apprentissage-non-supervise/>.
- [2] Architecture gan (image) (1). <https://penseeartificielle.fr/tout-pour-bien-debuter-en-deep-learning-5/architecture-gan-image-1/>.
- [3] L'apprentissage automatique, ou machine learning, qu'est-ce que c'est ? <https://www.redhat.com/fr/topics/ai/what-is-machine-learning>.
- [4] L'apprentissage profond, ou deep learning, qu'est-ce que c'est ? <https://www.redhat.com/fr/topics/ai/what-is-deep-learning>.
- [5] Les algorithmes de deep learning. <https://www.jedha.co/formation-ia/algorithmes-deep-learning>.
- [6] L'intelligence artificielle : c'est quoi ? définition, formes et enjeux. <https://www.lepont-learning.com/fr/intelligence-artificielle-ia-definition/>.
- [7] PhishTank - A collaborative clearing house for data and information about phishing on the Internet. <https://www.phishtank.com/>.
- [8] Qu'est-ce que l'apprentissage supervisé ?, url = <https://www.ibm.com/fr-fr/topics/supervised-learning>.
- [9] OpenPhish - Phishing Intelligence. <https://www.openphish.com/>, 2014.
- [10] *le Journal of Cybersecurity Research*, 2024.
- [11] Phishing activity trends report, anti-phishing working group (apwg). 2024.
- [12] Phishing activity trends report, anti-phishing working group (apwg). 2024.
- [13] Siehyun Strobel Jeong-Ho Chang Andr ´e Bergholz, Gerhard Paaf, Frank Reichartz. Improved phishing detection using model-based features. *The Fifth Conference on Email and Anti-Spam, 21-22 August 2008, Mountain View, California, USA*, 2008.
- [14] Scott Banister and Scott Weiss. Ironport corpus, 2000. Acquis par Cisco en 2007.
- [15] Luiz S. Oliveira Cleber K. Olivo, Altair O. Santin. Obtaining the threat model for e-mail phishing. *Applied Soft Computing*, 2009.
- [16] William W. Cohen. Enron email dataset, 2014. AEUB Natural Language Processing Group, T. E. S. Datasets, Athens, Greece.

- [17] G. V. Cormack and T. R. Lynam. Trec 2005 spam track overview. In *Proc. TREC*, 2005.
- [18] DataScientest. Introduction au nlp (natural language processing), 2023. Accessed : 2024-06-22.
- [19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*, 2018.
- [20] Elastic. What is machine learning ?, 2023. Accessed : 2024-06-22.
- [21] Adwan Yasin et Abdelmunem Abuhasan. An intelligent classification model for phishing email detection. *International Journal of Network Security Its Applications (IJNSA) Vol.8, No.4, July 2016*.
- [22] Meraj Farheen Ansari et Pawan Kumar Sharma et Bibhu Dash. Prevention of phishing attacks using ai-based cybersecurity awareness training. *International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN)*, 3(3), 2022.
- [23] Stanford NLP Group. Glove : Global vectors for word representation.
- [24] Soman KP Harikrishnan NB, Vinayakumar R. A machine learning approach towards phishing email detection. *CEN-Security@IWSPA 2018*, 2018.
- [25] Vinayakumar R Soman KP Hiransha M, Nidhin A Unnithan. Deep learning based phishing e-mail detection. *CEUR Workshop Proceedings (CEUR-WS.org)*, 2018.
- [26] T.N. Jagatic, N.A. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Communications of the ACM*, 50(10) :94–100, 2007.
- [27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 2012.
- [28] P. Kumaraguru, S. Sheng, A. Acquisti, L.F. Cranor, and J. Hong. Teaching johnny not to fall for phish. *ACM Transactions on Internet Technology*, 10(2) :1–31, 2010.
- [29] Ponnurangam Kumaraguru, Justin Cranshaw, Alessandro Acquisti, Lorrie Cranor, Jason Hong, Margaret A. Blair, and Tania Pham. School of phish : a real-world evaluation of anti-phishing training. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, page 1–12, 2009.
- [30] Ioannis Agrafiotis1 Lukáš Halgaš1 and Jason R.C. Nurse2. Catching the phish : Detecting phishing attacks using recurrent neural networks (rnns). 2019.
- [31] Jose Nazario. Nazario phishing corpus, 2004.
- [32] Stuart Russell and Peter Norvig. *Artificial Intelligence : A Modern Approach*. Pearson, 4th edition, 2021.
- [33] Sage. Définition du deep learning, 2023.
- [34] S. Sheng, M. Holbrook, P. Kumaraguru, L.F. Cranor, and J. Downs. Who falls for phish ? a demographic analysis of phishing susceptibility and effectiveness of interventions, 2010.

## Bibliographie

---

- [35] Paul Boyle Lynsay A. Shepherd. Mailtrout : A machine learning browser extension for detecting phishing emails. 2021.
- [36] The Cognitive Assistant that Learns and Organizes (CALO) Project. Calo project, 2014.
- [37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.
- [38] Masoumeh Zareapoor. Feature extraction or feature selection for text classification : A case study on phishing email detection. *I.J. Information Engineering and Electronic Business*, 2015, 2, 60-65, 2015.