الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Master

En Télécommunication
Spécialité : Réseaux & Télécommunications

Présenté par

Bousmaha Ilhem

Un modèle d'apprentissage en profondeur pour détecter L'hameçonnage des SMS (smishing)

Proposé par : **Boumahdi Fatima** & **Remmide Mohamed Abdelkarim**

Année Universitaire 2022-2023

Remerciement

Au premier lieu, je tiens à remercier Allah le tout puissant qui m'a donné la volonté, la patience, le courage et la santé pour terminer ce travail.

Je tiens à remercier tout particulièrement mon encadrante Madame Boumahdi Fatima et mon Co-promoteur Monsieur Remmide Mohamed Abdelkarim d'avoir accepté de m'encadrer dans la conception et l'élaboration de ce travail, pour leurs aides et leurs orientations précieuses, tout le long de ce projet.

Mes remerciements les plus vifs s'adressent aussi à Monsieur le président et les membres de jury d'avoir accepté d'examiner et d'évaluer mon travail.

Je remercie infiniment tous mes enseignants du département de l'électronique en particulier le chef de département Mr. Ait Saadi.

Je remercie aussi ma mère et ma famille qui m'ont donné leurs aides, patience, leurs soutiens et leurs sacrifices qui m'ont poussé à y aller jusqu' au bout de cette tâche.

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

Dédicace

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

À ma grand-mère **Dahbia** ALLAH YERHMHA, qui a toujours été un pilier de force dans ma vie, je suis remplie de gratitude pour tout ce qu'elle a apporté à ma vie. J'espère sincèrement qu'elle est fière de moi.

À celle qui a été la source de mon existence et qui m'était toujours la lumière de ma vie, ma mère **Nassima**. Qui s'est sacrifiée pour mon bonheur et ma réussite, quoi que je dise, je ne saurai point te remercier comme il se doit. Ce mémoire de PFE est le fruit de votre dévouement et de vos sacrifices. Merci d'avoir été ma plus grande inspiratrice.

À mon grand père **Abdelkader** , qui m'a donné toujours le courage, la volonté et la gratitude. Que Dieu le Tout Puissant le garde et le procure santé et bonheur.

À mon oncle **Omar** et sa femme **Khadidja**, aucune dédicace ne saurait exprimer mon respect, mon amour pour les efforts qu'ils ont fourni pour moi. J'espère que je pourrais leurs rendre un peu de ce qu'ils ont fait pour moi un jour.

À ma tante **Djaouida**, Aucun langage ne saurait exprimer mon respect et ma considération pour leur soutien et encouragements. Je dédie ce travail en reconnaissance de l'amour qu'elle m'offre quotidiennement et sa bonté exceptionnelle.

À mes tantes **Wahiba** et **Nesrine** , mes oncles **Abdelkrim**, **Mohamed**, **Mustapha** et **Salem** leur soutien fut une lumière dans tout mon parcours. Aucune dédicace ne saurait exprimer l'amour l'estime et le respect que j'ai toujours eu pour eux.

À mes cousines **Khawla**, **Soumia**, **Nouha** et **Asmaa**, mes cousins **Ayoub** et **Salim** qui ont toujours été présentes pour moi, et ils m'ont chaleureusement encouragé tout au long de mon parcours. Merci du fond du cœur pour votre amour, votre soutien et votre présence précieuse dans ma vie.

À mes amies **Sarah**, **Nihad**, **Youssra**, **Sofia** et **Nerdjes** qui m'ont toujours étés la pour moi, et à qui je souhaite plus de succès.

À Tous les membres de ma famille **BOUMAHDI** ....

ملخص

أصبحت الرسائل الاحتيالية شكل من أشكال هجوم الهندسة الاجتماعية الذي يتضمن رسائل نصية قصيرة احتيالية، مشكلة رئيسية في الأمن السيبراني في مجال الاتصالات المتنقلة. في هذه الدراسة، نقترح طريقة جديدة للكشف عن الابتزاز بناءً على التعلم الفيدرالي، وهي تقنية تعلم لا مركزية تحافظ على الخصوصية. باستخدام خوارزميات التعلم العميق.

تظهر التجارب أن طريقة التعلم الفيدرالية تحقق دقة تبلغ ٩٢.٣٨٪، مما يوضح فعالية التعلم الفيدرالي في حل صعوبات اكتشاف الرسائل الاحتيالية مع الحفاظ على سرية البيانات. تقدم الطريقة المقترحة في هذا العمل حلاً للهجمات الصاخبة وتمهد الطريق لعمليات البحث المستقبلية في مجال أمان الهاتف المحمول الذي يحافظ على الخصوصية.

الكلمات المفتاحية: التعلم الفيديرالي ، الرسائل الاحتيالية ، الامن الهاتفي ، الهندسة الاجتماعية ، الأمن السيبراني.

Abstract

Smishing, a form of social engineering attack involving fraudulent SMS messages, has become a major cybersecurity issue in mobile communications. In this study, we propose a new smishing detection method based on federated learning, a decentralized learning technique that preserves privacy. Using deep learning algorithms including LSTM, Bi-LSTM, CNN and MLP, we build a robust smishing detection model in a federated learning framework.

Experiments show that the federated learning method using CNN achieves an accuracy of 92.38 %, demonstrating the efficacy of federated learning in solving the challenges of smishing detection while preserving data confidentiality. The proposed method offers a solution to smishing attacks, and paves the way for future research into privacy-preserving mobile security.

Keywords: Federated learning, Smishing, Mobile security, social engineering, Cyber security

Résumé

Le smishing, une forme d'attaque d'ingénierie sociale impliquant des messages SMS frauduleux, est devenu un problème majeur de cybersécurité dans le domaine des communications mobiles. Dans cette étude, nous proposons une nouvelle méthode de détection du smishing basée sur l'apprentissage fédéré, une technique d'apprentissage décentralisée qui préserve la vie privée. En utilisant des algorithmes d'apprentissage profond, notamment LSTM, Bi-LSTM, CNN et MLP, nous construisons un modèle robuste de détection du smishing dans un cadre d'apprentissage fédéré.

Les expériences démontrent que la méthode d'apprentissage fédéré en utilisant CNN atteint une précision de 92,38 %, cela démontre l'efficacité de l'apprentissage fédéré pour résoudre les difficultés de la détection du smishing tout en préservant la confidentialité des données. La méthode proposée offre une solution aux attaques de smishing et ouvre la voie à de futures recherches dans le domaine de la sécurité mobile préservant la vie privée.

Mots Clées : Apprentissage fédéré, Smishing, Sécurité mobile, ingénierie sociale, Cybersécurité

Liste des abréviations

ANN : *Artificial neural network*

Bi-LSTM : *Bi-directional Long Short Term Memory*

CNN : *Convolutional Neural Network*

DL : *Deep Learning*

FL : *Federated Learning*

FP : *Faux Positifs*

FN : *Faux Négatifs*

GloVe : *Global Vectors for Word Representation*

IP : *Internet Protocol*

IA : *Intelligence Artificielle*

KNN : *K-Nearest Neighbors Algorithm*

LSTM : *Long Short Term Memory*

MLP : *Multilayer Perceptron*

ML : *Machine Learning*

MMS : *Multimedia Messaging Service*

RNN : *Recurrent Neural Networks*

ROC : *Receiver Operating Characteristic*

SMS : *Short Message Service*

SVM : *Support Vector Machines*

Smishing : *SMS Phishing*

SMC : *Calcul Multipartite Sécurisé*

TI : *Technologies de l'information*

VOIP : *Voice Over Internet Protocol*

VP : *Vrais Positifs*

VN : *Vrais Négatifs*

Table des matières

Table des figures

Liste des tableaux

Introduction Générale

Avec la survenance de la pandémie de COVID-19, le monde n'a pas seulement été témoin d'une crise sanitaire, mais aussi d'une augmentation considérable du nombre de cyberattaques. Alors que les gens utilisent de plus en plus les plateformes virtuelles pour communiquer et échanger des informations, les pirates exploitent les vulnérabilités sous diverses formes. Le smishing, une technique trompeuse qui combine les messages SMS et les techniques d'hameçonnage pour tromper et escroquer des personnes peu méfiantes, est l'une de ces menaces.

Problématique

Le smishing, désigne une forme d'attaque où un lien vers un site web est transmis à un utilisateur de smartphone via un message SMS. Ce phénomène, initialement nommé par McAfee [1], une société spécialisée dans la sécurité Internet, se caractérise par l'exploitation des canaux de communication mobiles pour tromper les utilisateurs. Le smishing tire parti de la popularité des smartphones et de leur utilisation généralisée pour inciter les individus à cliquer sur des liens malveillants, divulguant ainsi des informations sensibles ou facilitant des actions frauduleuses. Cette forme d'attaque représente une menace croissante dans le paysage actuel, nécessitant une attention particulière et des solutions efficaces pour protéger les utilisateurs contre les pièges du smishing.

Objéctif

Notre recherche vise à résoudre les difficultés associées à la détection du smishing en tirant parti de la puissance de l'apprentissage fédéré dans le domaine de l'intelligence artificielle. Alors que les chercheurs ont fait des progrès significatifs dans la détection du smishing en utilisant diverses techniques d'apprentissage automatique et d'apprentissage profond, la mise en œuvre de l'apprentissage fédéré dans ce domaine reste inexplorée.

En utilisant l'apprentissage fédéré, notre objectif est de contourner les limitations de l'acquisition centralisée des données et de protéger la confidentialité et la sécurité des

données des utilisateurs en permettant l'apprentissage de modèles locaux sur les appareils clients, sans nécessité de transmettre des données brutes à un serveur central. Dans le cadre de l'apprentissage fédéré, nous intégrerons des algorithmes d'apprentissage profond pour créer un modèle de détection du smishing robuste et précis.

Notre recherche vise à résoudre le problème persistant des attaques par smishing, qui évoluent constamment pour contourner les méthodes de détection traditionnelles. En examinant des techniques et des modèles avancés, nous espérons contribuer à la détection de l'hameçonnage par SMS et fournir une protection plus efficace contre ces activités frauduleuses.

Notre méthode est applicable dans une variété de contextes, y compris les appareils mobiles, les plateformes de messagerie et les organisations qui gèrent des données utilisateur sensibles.

Plan du mémoire

Le mémoire se compose de quatre chapitres qui sont suivis d'une conclusion générale. Chacun de ces chapitres aborde une partie du travail réalisé dans ce projet :

- **Le premier chapitre** aborde l'apprentissage automatique et l'apprentissage profond, ainsi que leurs différents algorithmes. Nous explorerons également l'apprentissage fédéré.

- **Le deuxième chapitre** traite de l'ingénierie sociale, du phishing et de ses différents types. Nous nous concentrerons particulièrement sur le smishing. Nous abordons également les travaux connexes portant sur la détection du smishing.

- **Le troisième chapitre** présente le processus de la solution proposée, en expliquant en détail les étapes impliquées. Nous mettrons en évidence les différentes étapes de la détection de smishing et les méthodes utilisées.

- **Le dernier chapitre** présente les résultats de notre solution, en évaluant l'efficacité des algorithmes développés.

Chapitre 1

Apprentissage automatique

1.1 Introduction

L'intelligence artificielle (IA), l'apprentissage machine (ML) et l'apprentissage profond (DL) sont des sous-ensembles les uns des autres, et ils sont liés les uns aux autres. Mais chacun peut être appliqué de différentes manières pour obtenir la solution souhaitée à un problème. Il est donc essentiel de comprendre les différences et les similitudes entre eux pour les utiliser de la bonne manière.

Dans ce chapitre, nous allons explorer trois concepts essentiels de l'IA : l' apprentissage automatique, l'apprentissage profond et l'apprentissage fédéré. Il est crucial de comprendre leur relation et leurs différences. Nous aborderons brièvement ces concepts pour en saisir l'importance et les implications qu'ils offrent.

1.2 L'intelligence artificielle

L'intelligence artificielle est un domaine d'étude très large dans lequel les machines présentent des capacités cognitives telles que le comportement appris, l'interaction active avec l'environnement, le raisonnement et la déduction, la vision par ordinateur, la reconnaissance vocale, la résolution de problèmes, la représentation des connaissances, la perception, etc [2].

Plus familièrement, l'intelligence artificielle fait référence à toute activité dans laquelle une machine imite le comportement intelligent normalement manifesté par les humains. L'intelligence artificielle s'inspire des éléments de l'informatique, des mathématiques et des statistiques [2].

1.3 Apprentissage automatique

L'apprentissage automatique est un sous-ensemble de l'intelligence artificielle, qui a permis de nouvelles avancées dans un certain nombre de secteurs . L'apprentissage automatique consiste à former une machine de manière à ce qu'elle puisse apprendre à partir des données qui lui ont été présentées ou de ses expériences passées. Grâce à ses nombreuses techniques,l'apprentissage automatique est appliqué à un certain nombre de disciplines différentes [3].

L'apprentissage automatique comprend différents algorithmes, dont la méthode des k plus proches voisins (KNN : k-nearest neighbors) , k-moyennes (k-Means), les arbres de décision, la forêt aléatoire (Random forest), les machines à vecteurs de support (SVM : Support Vector Machine). L'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement sont des catégories de l'apprentissage automatique. [3]

1.3.1 Types d'apprentissage automatique

Il existe trois types : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

1.3.1.1 Apprentissage supervisé

La tâche d'apprentissage automatique de l'apprentissage supervisé consiste à former une fonction qui traduit une entrée en une sortie à l'aide d'exemples de paires entrée-sortie. Il dérive une fonction à partir de données de formation étiquetées, constituées d'une collection de cas pratiques. Les algorithmes qui nécessitent une aide extérieure sont connus sous le nom d'algorithmes d'apprentissage automatique supervisé. L'ensemble de données d'entrée est séparé en ensembles de données de formation et de test. L'ensemble de données de formation contient une variable de sortie qui nécessite une prédiction ou une classification. Tous les algorithmes utilisent l'ensemble de données de formation pour découvrir divers modèles qu'ils utilisent ensuite dans l'ensemble de données de test pour faire des prédictions ou catégoriser les données [4].

L'apprentissage supervisé a été appliqué avec succès dans plusieurs domaines tels que : La recherche d'informations, l'analyse de marché , l'exploration de données, la vision par ordinateur, la reconnaissance vocale, la détection de spam, la bioinformatique, la chimioinformatique et l'exploration de données [5].

La figure 1.1 illustre le déroulement des algorithmes d'apprentissage automatique supervisé.

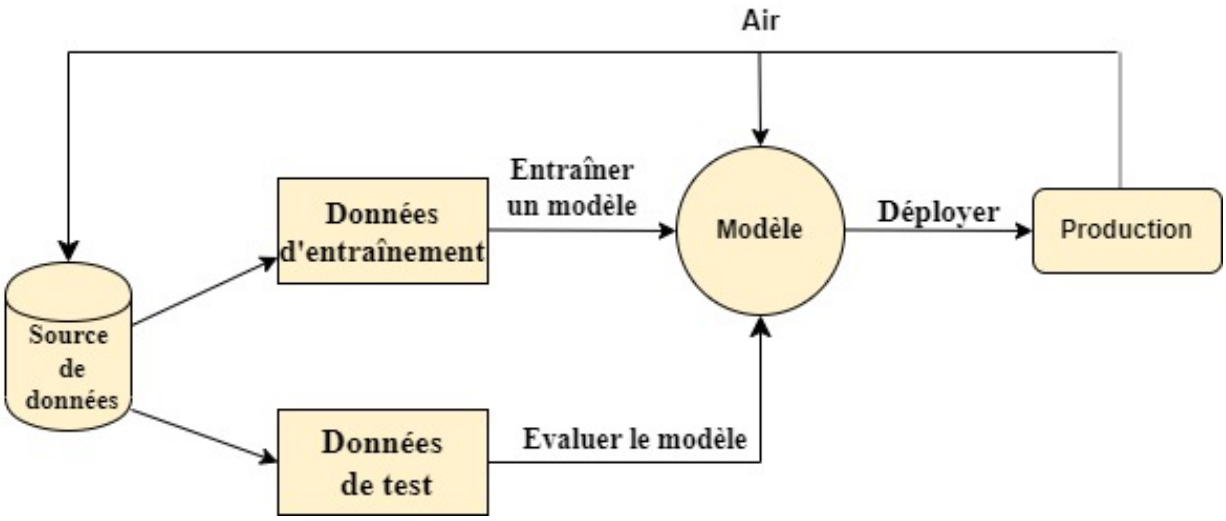


Figure 1.1: Diagramme d'apprentissage supervisé. [4]

1.3.1.2 Apprentissage non supervisé

Lorsqu'un système d'apprentissage est censé identifier des modèles sans aucune étiquette ou directive, on parle d'apprentissage non supervisé. Afin de trouver des informations structurelles intéressantes, telles que des groupes d'éléments qui partagent des propriétés communes (un processus connu sous le nom de clustering) ou des représentations de données qui sont projetées d'un espace à haute dimension vers un espace à plus faible dimension (un processus connu sous le nom de réduction de la dimensionnalité) [6].

1.3.1.3 Apprentissage par renforcement

L'apprentissage par renforcement est une branche de l'apprentissage automatique consacrée à la prise de décision. Il s'agit d'un agent qui apprend à fonctionner de manière à maximiser les récompenses dans un environnement donné. L'apprentissage par renforcement se concentre sur l'apprentissage par essais et erreurs, par opposition à l'apprentissage supervisé, où le modèle est formé à l'aide de données étiquetées [7].

L'agent apprend le meilleur comportement en recevant un retour d'information sous forme d'incitations ou de pénalités en fonction de ses comportements. Les systèmes automatisés qui doivent prendre des décisions séquentielles et qui peuvent apprendre de leurs erreurs afin d'obtenir les résultats souhaités peuvent bénéficier de l'apprentissage par essais et erreurs [7].

La figure 1.2 représente le fonctionnement d'algorithme d'apprentissage par renforcement :

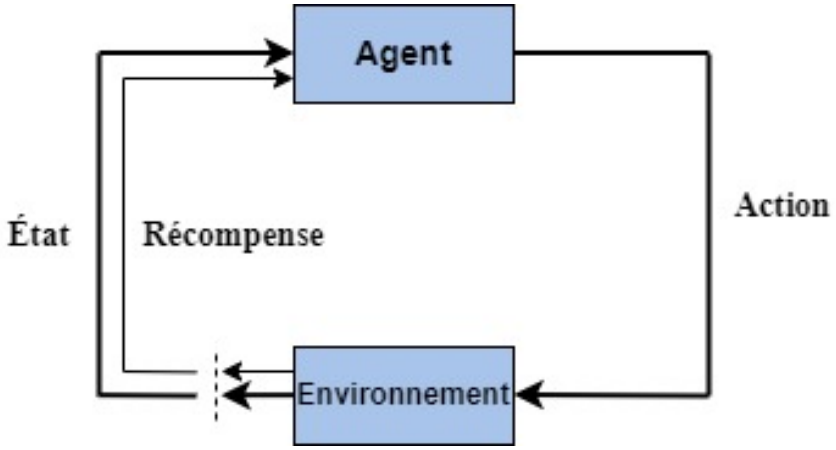


Figure 1.2: Représentation du fonctionnement d'algorithme d'apprentissage par renforcement. [8]

1.3.2 Algorithmes d'apprentissage automatique

Cette section présente divers algorithmes d'apprentissage automatique qui couvrent un certain nombre de méthodes et de méthodologies :

1.3.2.1 Decision Tree

Un arbre de décision est un graphique qui présente les décisions et leurs résultats sous la forme d'un arbre. Les arêtes du graphique indiquent les conditions ou les règles de prise de décision, tandis que les nœuds du graphique représentent un événement ou un choix. Chaque arbre comporte des nœuds et des branches. Chaque nœud représente un ensemble de caractéristiques qui doivent être catégorisées, tandis que chaque branche indique une valeur possible pour le nœud [4].

1.3.2.2 Random Forest

La forêt aléatoire est l'un des algorithmes les plus précis et les plus performants en matière de classification et de régression. Tout comme une forêt n'est rien d'autre qu'une collection d'arbres, la méthode de la forêt aléatoire utilisée dans l'apprentissage automatique est une collection de divers arbres de décision. Pour déterminer le résultat final, ces arbres de décision sont réunis. Les caractéristiques sont choisies au hasard, d'où le nom "aléatoire". La forêt aléatoire est un algorithme efficace, adaptable et évolutif. La technique de la forêt aléatoire est utilisée pour la détection du phishing , la prédiction des troubles liés à la consommation d'opioïdes et les risques ou fraudes liés aux cartes de crédit [3].

1.3.2.3 Support Vector Machine

La machine à vecteur de support (SVM) est une autre approche moderne et populaire de l'apprentissage automatique. Les machines à vecteurs de support dans l'apprentissage automatique sont des modèles d'apprentissage supervisés avec des algorithmes d'apprentissage connexes qui examinent les données utilisées pour les analyses de régression et de classification [4].

Les SVM peuvent effectivement effectuer une classification non linéaire en plus de la classification linéaire en cartographiant implicitement leurs entrées dans des espaces de caractéristiques à haute dimension. Cette technique est connue sous le nom d'astuce du noyau. Elle consiste essentiellement à tracer des limites entre les classes. Les marges sont tracées de manière à ce que la distance entre elles et les classes soit la plus courte possible, ce qui minimise l'erreur de classification [4].

1.3.2.4 AdaBoost

AdaBoost, souvent connu sous le nom de boosting adaptatif, est un algorithme de boosting. Cet algorithme applique la stratégie de correction de son prédécesseur. Il se concentre principalement sur les situations de formation dans lesquelles l'ajustement du modèle précédent était inadéquat. Par conséquent, les situations complexes reçoivent plus d'attention que les autres pour chaque nouveau prédicteur [9].

Une série d'apprenants faibles est adaptée à diverses données d'apprentissage pondérées. Il commence par prévoir le lot initial de données et attribue à chaque observation le même poids. Lorsque le premier apprenant fait une prédiction erronée, l'observation qui a été prévue par erreur se voit attribuer plus de poids. En raison de la nature itérative du processus, il continue d'ajouter des apprenants jusqu'à ce que la précision ou la quantité de modèles ne puisse plus être augmentée [9].

AdaBoost utilise principalement des timbres de décision. Toutefois, si un algorithme d'apprentissage automatique prend des poids à partir d'un ensemble de données d'apprentissage, nous pouvons utiliser cette méthode comme base d'apprentissage. Dans l'apprentissage automatique, nous pouvons utiliser la méthode AdaBoost pour les problèmes de classification et de régression [9].

1.4 Apprentissage profond

L'apprentissage profond est un sous-ensemble de l'apprentissage automatique, qui est consacré à la recherche et à la création de machines intelligentes. [10]

L'apprentissage profond est utilisé sur le lieu de travail pour effectuer des tâches pratiques dans de nombreuses disciplines différentes, notamment la vision par ordinateur (pour les images), le traitement du langage naturel (pour les textes) et la reconnaissance automatique de la parole (audio) [10].

En bref, les réseaux neuronaux artificiels, une classe d'algorithmes vaguement modelés sur le cerveau humain, sont le principal composant des réseaux neuronaux artificiels, un sous-ensemble de techniques de la boîte à outils de l'apprentissage automatique. [10]

1.5 Réseaux de neurones

Les réseaux neuronaux se sont avérés très efficaces dans diverses applications d'apprentissage automatique [11]. Sans aucun doute, l'efficacité de la formation dépend grandement de la manière dont les réseaux neuronaux sont construits. Il est communément admis que certaines topologies de réseau peuvent être apprises et généralisées efficacement.

1.5.1 Réseaux de neurones convolutifs CNN

Le réseau neuronal convolutionnel (Convolutional Neural Network : CNN) est un type de réseau neuronal à propagation avant qui utilise des structures convolutionnelles pour extraire des caractéristiques des données. Contrairement aux techniques conventionnelles, le CNN ne nécessite pas d'extraction manuelle des caractéristiques. Il a été inspiré par la perception visuelle et ses fonctions d'activation imitent le transfert de signaux électriques neuronaux dépassant un seuil [12].

Le CNN présente plusieurs avantages par rapport aux réseaux neuronaux généraux. Premièrement, il établit des liens locaux, où chaque neurone se connecte uniquement à quelques neurones de la couche précédente, ce qui réduit les paramètres et accélère la convergence. Deuxièmement, il partage les poids de nombreuses connexions, ce qui réduit le nombre de paramètres nécessaires. Troisièmement, il utilise des couches de mise en commun pour réduire la dimensionnalité des données, préservant les informations utiles tout en réduisant la quantité de données et le nombre de paramètres [12].

Grâce à ses trois caractéristiques séduisantes, le CNN est devenu l'un des algorithmes les plus représentatifs dans le domaine de l'apprentissage profond. Il permet une extraction automatique des caractéristiques, réduit les paramètres nécessaires et préserve les informations pertinentes tout en réduisant la quantité de données [12].

La figure 1.3 illustre la structure du réseau neuronal convolutif :

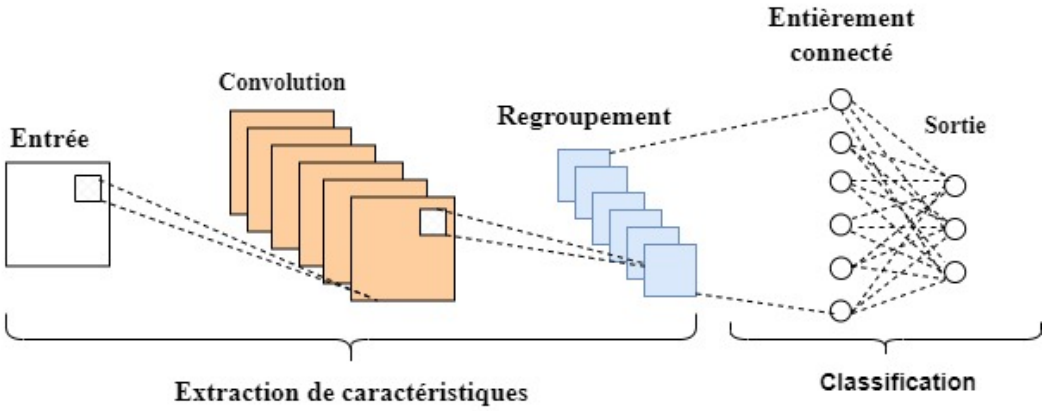


Figure 1.3: Structure du réseau de neurones convolutifs [13]

1.5.1.1 Couche de convolution

La couche convolutive est essentielle au fonctionnement des CNN. L'utilisation de noyaux est l'élément principal des paramètres de la couche. Bien que ces noyaux aient souvent de faibles dimensions spatiales, ils couvrent toute la profondeur de l'entrée. Chaque filtre est convolué à travers les dimensions spatiales de l'entrée par la couche convolutive lorsque les données l'atteignent, créant ainsi une carte d'activation en 2D [14].

Chaque noyau aura une carte d'activation associée, qui sera empilée le long de la dimension de profondeur pour créer l'ensemble du volume de sortie de la couche convolutive [14].

1.5.1.2 Couche de mise en commun

Les couches de mise en commun permettent de réduire progressivement la dimensionnalité de la représentation, ce qui réduit la complexité de calcul et le nombre de paramètres du modèle. La fonction "MAX" est utilisée par la couche de mise en commun pour mettre à l'échelle la dimensionnalité de chaque carte d'activation dans l'entrée. Ces couches prennent généralement la forme de couches de mise en commun maximale avec des noyaux de 22 dimensions appliqués avec un pas de 2 le long des dimensions spatiales de l'entrée. Cela permet de conserver le volume de profondeur à sa taille normale tout en réduisant la carte d'activation à 25 % de sa taille d'origine [14].

1.5.1.3 Couche entièrement connectée

Une couche entièrement connectée est une couche où chaque neurone est connecté à tous les neurones de la couche précédente. Chaque connexion est pondérée par un poids, et les opérations effectuées à cette couche sont généralement des multiplications matricielles

suivies de fonctions d'activation. Des couches entièrement connectées sont souvent utilisées dans la partie finale d'un réseau de neurones pour effectuer une classification ou une prédiction [14].

1.5.2 Réseau de neurones récurrentes RNN

Le réseau neuronal récurrent (RNN) est un modèle de séquence neuronale avancé qui excelle dans des tâches critiques telles que la modélisation du langage, la reconnaissance vocale et la traduction automatique [15]. Il est bien connu qu'une régularisation efficace est nécessaire au succès des applications de réseaux neuronaux. Malheureusement, la technique de régularisation la plus efficace pour les réseaux neuronaux feedforward, le dropout, ne fonctionne pas bien avec les RNN. Les RNN de grande taille ayant tendance à se surajuster, les applications pratiques des RNN utilisent souvent des modèles trop petits. Les RNN ne bénéficient que modestement des techniques de régularisation actuellement disponibles [15].

L'architecture générale du réseau neuronal récurrent (RNN) est représentée dans la figure 1.4 :

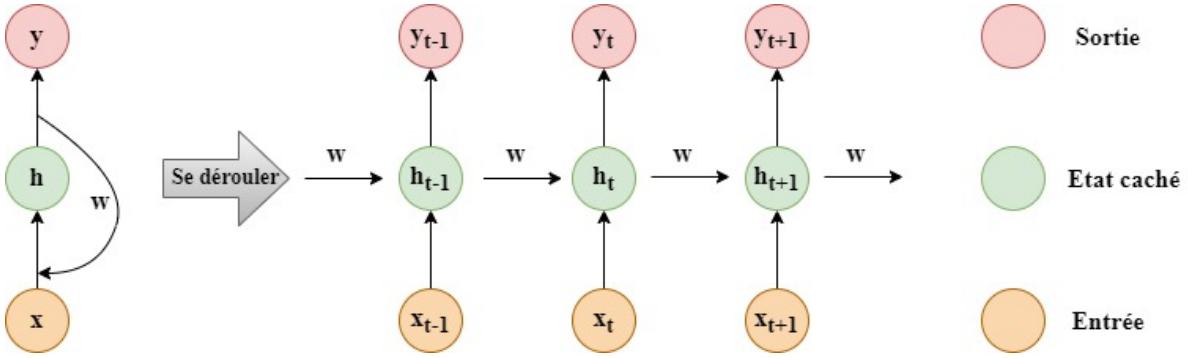


Figure 1.4: Architecture générale du réseau neuronal récurrent (RNN) [16]

1.5.2.1 Long Short-Term Memory(LSTM)

Le LSTM est un type particulier de réseau neuronal récurrent. Cette architecture est spécifiquement introduite pour résoudre le problème des gradients qui disparaissent et explosent. Ce type de réseau est également plus apte à maintenir des connexions à longue distance et à comprendre la connexion entre les valeurs au début et à la fin d'une séquence [17].

Les expressions sont introduites par le modèle LSTM, en particulier les portes (gates). Il existe en fait trois types distincts de portes :

- Forget gate : régule la quantité d'informations que recevra la cellule mémoire de la phase précédente.

- Update (input) gate : détermine si la cellule de mémoire doit être mise à jour ou non. De plus, elle régule la quantité de données qu'une éventuelle nouvelle cellule mémoire enverra à la cellule mémoire actuelle.

- Output gate : contrôle la valeur de l'état caché suivant.

La figure 1.5 représente l'architecture du bloc LSTM :

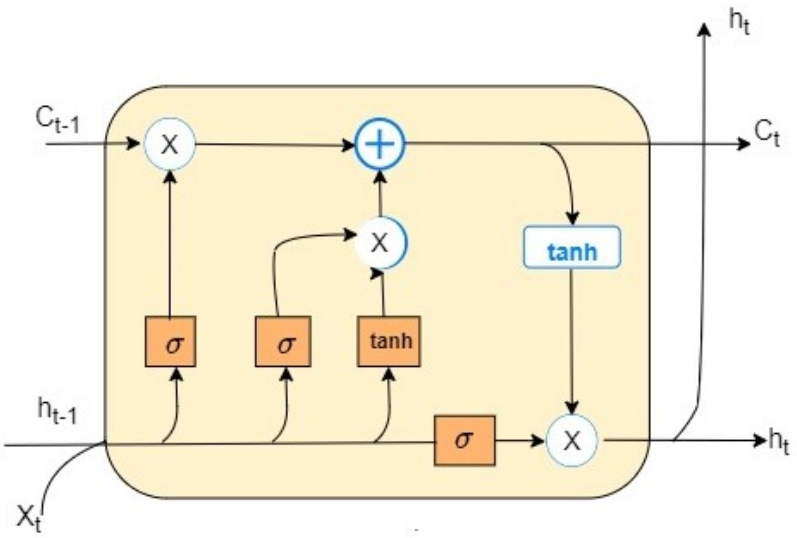


Figure 1.5: L'architecture du bloc LSTM [17]

1.5.2.2 Bi-LSTM

Un réseau neuronal récurrent largement utilisé pour le traitement du langage naturel est appelé LSTM bidirectionnel (BiLSTM). Il peut utiliser des données des deux côtés et, contrairement au LSTM ordinaire, l'entrée circule dans les deux sens. Dans les deux sens de la séquence, il constitue un outil puissant pour modéliser les relations séquentielles entre les mots et les phrases [17].

Le BiLSTM inverse la direction du flux d'informations en ajoutant une couche LSTM supplémentaire. Cela signifie simplement que dans la couche LSTM supplémentaire, la séquence d'entrée circule en sens inverse. Les sorties des deux couches LSTM sont ensuite combinées de diverses manières, notamment par la moyenne, la somme, la multiplication et la concaténation [17].

L'architecture de Bi-LSTM est présentée dans la figure 1.6 :

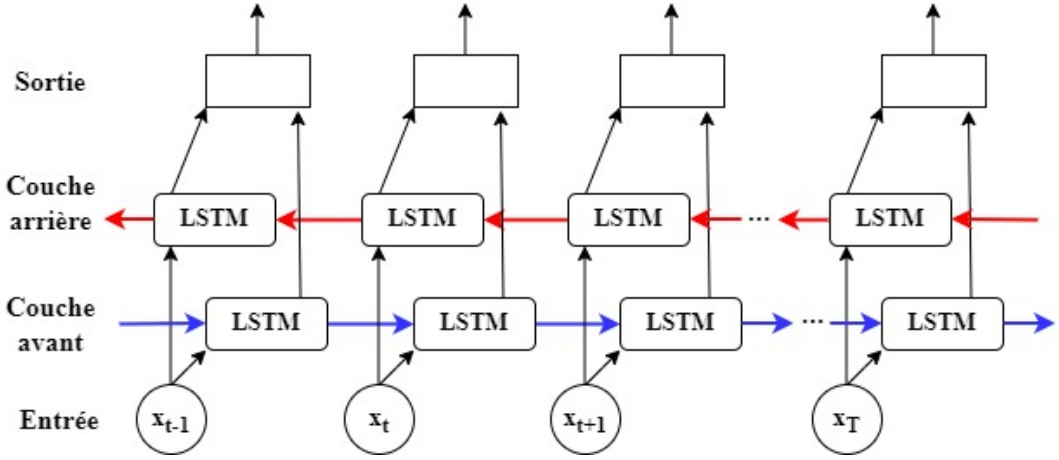


Figure 1.6: Architecture de Bi-LSTM [17]

1.5.3 Perceptron multicouche(MLP)

Le perceptron multicouche (MLP) est un complément aux réseaux neuronaux de type feed-forward. Comme le montre la figure 1.7, il comporte trois types de couches : une couche d'entrée, une couche de sortie et une couche cachée [18].

La couche d'entrée reçoit le signal d'entrée qui sera traité. La couche de sortie accomplit les tâches nécessaires, telles que la prédiction et la classification. Le véritable moteur de calcul du MLP consiste en un nombre arbitraire de couches cachées placées entre les couches d'entrée et de sortie. Dans un MLP, les données se déplacent de la couche d'entrée à la couche de sortie dans le sens direct, comme dans un réseau de type feed forward. L'approche d'apprentissage par propagation arrière est utilisée pour former les neurones du MLP [18].

Les MLP peuvent résoudre des problèmes qui ne sont pas linéairement séparables puisqu'ils sont conçus pour approximer n'importe quelle fonction continue. La classification, la reconnaissance, la prédiction et l'approximation des formes sont les principales applications des MLP [18].

La figure 1.7 illustre une représentation schématique d'un MLP (Multilayer Perceptron) :

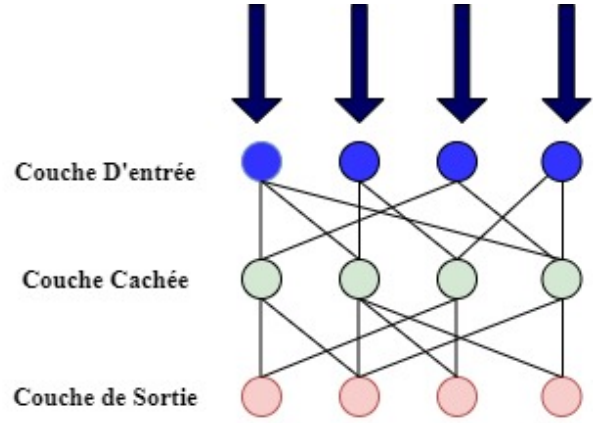


Figure 1.7: Représentation schématique d'un MLP avec une seule couche cachée. [18]

1.6 Apprentissage fédéré

L'apprentissage fédéré est une configuration dans laquelle plusieurs clients collaborent pour résoudre des problèmes d'apprentissage automatique, coordonnés par un agrégateur central. Cette approche permet la décentralisation des données d'apprentissage, préservant ainsi la confidentialité des données de chaque appareil. Les principes clés de l'apprentissage fédéré sont le calcul local et la transmission du modèle, réduisant les coûts systémiques et les problèmes de confidentialité [19].

Dans cette configuration, les données originales du client restent localement et ne sont pas échangées. Chaque appareil utilise ses propres données pour l'apprentissage local, télécharge le modèle vers le serveur central pour agrégation, puis reçoit le modèle mis à jour du serveur. Ainsi, des modèles statistiques peuvent être formés sur des appareils distants ou des centres de données isolés, tout en préservant la confidentialité des données locales [19].

L'apprentissage fédéré adopte une architecture d'apprentissage automatique distribuée, où le serveur central est responsable de l'agrégation des modèles, tandis que les clients mettent à jour leur modèle localement. Cela permet aux clients de conserver la propriété de leurs données, en ne téléchargeant que les modèles mis à jour sur le serveur central sans exposer leurs données privées [20].

Des améliorations récentes visent à surmonter les défis statistiques et à améliorer la sécurité de l'apprentissage fédéré. Des recherches sont également en cours pour rendre l'apprentissage fédéré plus personnalisable [20].

La figure 1.8 représente l'architecture générale de l'apprentissage fédéré :

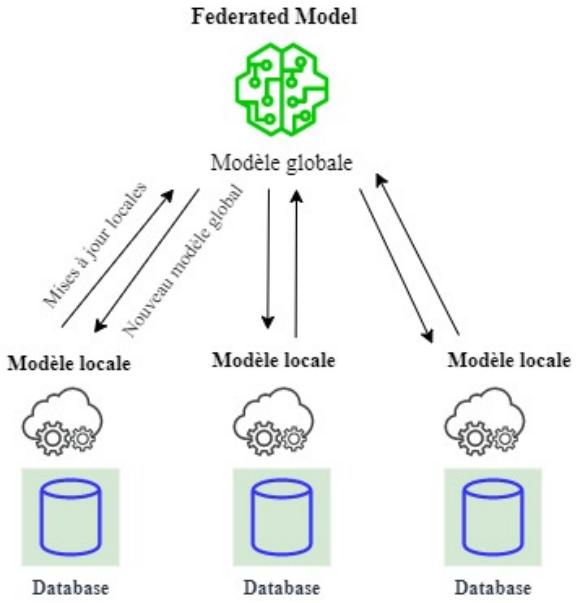


Figure 1.8: Architecture générale d'apprentissage fédéré [21]

1.6.1 Définition de l'apprentissage fédéré

Définissez N propriétaires de données $\{F_1, ...F_N\}$, qui souhaitent tous former un modèle d'apprentissage automatique en consolidant leurs données respectives $\{D_1, ...D_N\}$. Une méthode classique consiste à rassembler toutes les données et à utiliser D = $D_1 \cup ... \cup D_N$ pour entraîner un modèle M_{SUM} [20].

Un système d'apprentissage fédéré est un processus d'apprentissage dans lequel les propriétaires de données forment en collaboration un modèle M_{FED} , processus dans lequel tout propriétaire de données F_i n'expose pas ses données D_i aux autres. En outre, la précision de M_{FED}, désignée par V_{FED} , devrait être très proche de la performance de M_{SUM} , V_{SUM}. [20]

Formellement, est un nombre réel non négatif, si | V_{FED} - V_{SUM} |< (1), nous disons que l'algorithme d'apprentissage fédéré a une perte d'exactitude de δ. [20]

1.6.2 Applications de L'apprentissage fédéré

Les principaux fournisseurs de services ont mis en œuvre des techniques d'apprentissage fédéré, qui sont essentielles pour prendre en charge les applications sensibles à la confidentialité, dans lesquelles les données d'apprentissage sont dispersées à la périphérie. [22]

Parmi les exemples d'utilisations prospectives, citons l'adaptation des véhicules autonomes au comportement des piétons, la compréhension du sentiment, de la localisation

sémantique ou des activités des utilisateurs de téléphones mobiles, et la prévision d'événements sanitaires tels que le risque de crise cardiaque à partir de la technologie portable. Nous passons en revue ci-dessous quelques-unes des utilisations canoniques de l'apprentissage fédéré :

- Smartphones : en apprenant conjointement le comportement des utilisateurs sur un grand nombre de téléphones, les modèles statistiques peuvent alimenter des applications telles que la prédiction du mot suivant, la détection des visages et la reconnaissance vocale. Cependant, les utilisateurs peuvent être réticents à partager leurs données pour protéger leur vie privée ou pour économiser la bande passante et la batterie limitées de leur téléphone. L'apprentissage fédéré a le potentiel d'activer des capacités prédictives sur les smartphones sans dégrader l'expérience utilisateur ni révéler d'informations privées [22].

- Les organisations : Dans le contexte de l'apprentissage fédéré, les organisations ou les institutions peuvent également être considérées comme des "dispositifs". Les hôpitaux, par exemple, sont des entreprises qui abritent une grande quantité de données sur les patients pour des traitements prédictifs. Les hôpitaux doivent respecter des lois strictes en matière de confidentialité et peuvent être amenés à conserver des données locales en raison d'exigences éthiques, administratives ou réglementaires. Parce qu'il peut alléger la charge du réseau et faciliter l'apprentissage privé parmi de nombreux dispositifs/organisations, l'apprentissage fédéré est une solution possible pour ces applications [22].

- L'internet des objets : De nombreux capteurs peuvent être présents dans les réseaux contemporains de l'IdO, comme les technologies vestimentaires, les véhicules autonomes ou les maisons intelligentes, ce qui leur permet de recueillir des données entrantes, d'y répondre et de s'y adapter en temps réel. Par exemple, une flotte de véhicules autonomes pourrait avoir besoin de la version la plus récente d'un modèle de trafic, de construction ou de comportement des piétons pour fonctionner correctement. Cependant, comme les données sont privées et que chaque dispositif a un accès limité, la création de modèles agrégés dans ces circonstances peut être difficile. Tout en protégeant la vie privée des utilisateurs, les techniques d'apprentissage fédéré peuvent aider à développer des modèles qui s'adaptent rapidement aux changements dans ces systèmes [22].

1.6.3 Confidentialité de l'apprentissage fédéré

La confidentialité est l'une des propriétés essentielles de l'apprentissage fédéré [20]. Cela nécessite des modèles et des analyses de sécurité pour fournir des garanties de confidentialité significatives. Nous allons comparer brièvement les différentes techniques de la confidentialité de l'apprentissage fédéré.

1.6.3.1 Calcul multipartite sécurisé (SMC)

Les modèles de sécurité SMC impliquent naturellement plusieurs parties et fournissent une preuve de sécurité dans un cadre de simulation bien défini pour garantir une connaissance nulle complète, c'est-à-dire que chaque partie ne connaît rien d'autre que son entrée et sa sortie. La connaissance zéro est très souhaitable, mais cette propriété désirée nécessite généralement des protocoles de calcul compliqués et peut ne pas être atteinte efficacement. [20]

1.6.3.2 Confidentialité différentielle

Une autre ligne de travail utilise les techniques de confidentialité différentielle ou de k-anonymat pour la protection de la confidentialité des données. Les méthodes de confidentialité différentielle, de k-anonymat et de diversification consistent à ajouter du bruit aux données ou à utiliser des méthodes de généralisation pour masquer certains attributs sensibles jusqu'à ce que le tiers ne puisse pas distinguer l'individu, rendant ainsi les données impossibles à restaurer pour protéger la vie privée des utilisateurs. [20]

1.6.3.3 Cryptage homomorphe

Le cryptage homomorphique est également adopté pour protéger la confidentialité des données de l'utilisateur par l'échange de paramètres sous le mécanisme de cryptage pendant l'apprentissage automatique. Contrairement à la protection différentielle de la confidentialité, les données et le modèle lui-même ne sont pas transmis et ne peuvent pas être devinés par les données de l'autre partie. Par conséquent, il y a peu de possibilités de fuites au niveau des données brutes. [20]

1.6.4 L'avantage de l'apprentissage fédéré

Par rapport aux méthodes centralisées et traditionnelles d'apprentissage automatique, l'apprentissage fédéré n'en est encore qu'à ses débuts mais offre déjà des avantages substantiels. Les avantages de l'apprentissage fédéré comprennent :

- La sécurité : l'apprentissage fédéré assure la protection des données des utilisateurs, L'agrégation sécurisée est utilisée dans l'apprentissage fédéré pour protéger les mises à jour des clients des regards indiscrets. Ainsi, le serveur est incapable de déterminer la valeur ou l'origine de toute mise à jour de modèle fournie par l'utilisateur. Par conséquent, les attaques par attribution et inférence de données sont moins probables [23].

Comme les données personnelles restent locales, les entreprises telles que les institutions financières et les hôpitaux, qui sont soumises à des lois strictes en matière de

confidentialité, peuvent bénéficier de la sécurité qu'elles offrent. Les données sont moins vulnérables aux violations puisqu'il est plus facile de les regrouper sur un serveur central et externe [23].

- S'adapter à de nouvelles situations : L'un des principaux avantages de l'apprentissage fédéré est sa capacité à s'adapter à des circonstances nouvelles qui n'existaient pas lorsque le modèle a été développé. Dans de nombreuses circonstances, les modèles peuvent être utilisés dans de nouveaux contextes sans avoir besoin d'être complètement ré-entraînés. Ainsi, le modèle peut s'appuyer sur ses connaissances antérieures pour générer des prédictions basées sur ses expériences passées [24].

- Utilisation croisée des modèles : L'apprentissage fédéré présente l'avantage supplémentaire de permettre l'amélioration d'une application au profit d'autres. Par exemple, si un modèle acquiert la capacité de prédire les résultats avec plus de précision dans un domaine, il peut transférer ces connaissances à un autre domaine pour améliorer potentiellement les performances ou réduire les coûts par rapport à la formation manuelle des modèles ou à l'utilisation de techniques d'apprentissage automatique plus conventionnelles comme les techniques d'apprentissage supervisé ou les réseaux neuronaux artificiels (ANN) [24] .

- Les téléphones mobiles et autres appareils peuvent développer conjointement un modèle de prédiction grâce à l'apprentissage fédéré. Cette méthode permet d'éviter de télécharger et de stocker des données d'entraînement sur un serveur central en les conservant localement sur l'appareil [23].

- Diversité des données : Les entreprises peuvent être empêchées de combiner des ensembles de données provenant de nombreuses sources pour des raisons autres que la sécurité des données, comme l'indisponibilité du réseau dans les périphériques. L'apprentissage fédéré facilite l'accès à des données diverses, même lorsque des sources de données particulières ne peuvent interagir qu'à des périodes spécifiques [25].

1.7 Word embedding

Comme le texte ne peut pas être traité par des modèles d'apprentissage automatique, nous devons trouver un moyen de transformer ces données textuelles en données numériques. Ce travail peut être accompli à l'aide de méthodes comme Bag of words et le TF-IDF [26]. En outre, il existe deux autres méthodes que nous pouvons employer, comme one-hot encoding ou l'attribution de numéros spécifiques à chaque mot d'un lexique. Cette dernière méthode est plus efficace que one-hot encoding puisque nous avons un vecteur dense au lieu d'un vecteur épars. Cette stratégie fonctionne donc même lorsque notre

vocabulaire est étendu [26].

Les mots ayant des significations similaires peuvent être regroupés à proximité les uns des autres dans l'espace vectoriel à l'aide de l'intégration des mots. Par exemple, lorsqu'un terme comme "grenouille" est représenté, les voisins les plus proches d'une grenouille seraient "grenouilles", "crapauds" et "Litoria". Cela suggère que, même si les vecteurs de deux mots sont comparables, il est acceptable qu'un classificateur ne voie que le mot "grenouille" pendant la formation et ne voie pas du tout le mot "Litoria" pendant le test. Les incorporations de mots permettent également de découvrir des relations. La recherche d'un mot équivalent peut se faire en ajoutant les différences vectorielles entre deux mots à un autre vecteur de mot. Par exemple, "homme" -"femme" + "reine" "roi" [26].

1.7.1 Word2vec

Word2vec, développé par [27], est une méthode d'intégration de mots (word embedding), largement appliquée au traitement du langage naturel.

En regroupant les mots apparentés, les représentations distribuées des mots dans un espace vectoriel permettent aux algorithmes d'apprentissage d'obtenir de meilleures performances dans les tâches de traitement du langage naturel. Les représentations de mots ont été utilisées pour la première fois en 1986 par Rumelhart, Hinton et Williams [27]. Depuis lors, ce concept a été utilisé avec succès dans la modélisation statistique du langage. Des applications à la reconnaissance automatique de la parole, à la traduction automatique et à une variété de tâches de traitement du langage naturel font également partie de l'étude de suivi.

Le modèle Skip-gram, une technique efficace pour l'apprentissage de représentations vectorielles de haute qualité des mots, à partir de quantités importantes de données textuelles non structurées, a été développé par [27]. Le modèle Skip-gram ne nécessite pas de multiplications matricielles denses pendant la formation, contrairement à la majorité des autres architectures de réseaux neuronaux utilisées précédemment pour l'apprentissage des vecteurs de mots. De ce fait, l'apprentissage est très efficace : un seul ordinateur fonctionnant au maximum de son efficacité peut s'entraîner sur plus de 100 milliards de mots en une seule journée. [27]

La figure 1.9 représente L'architecture du modèle Skip-gram :

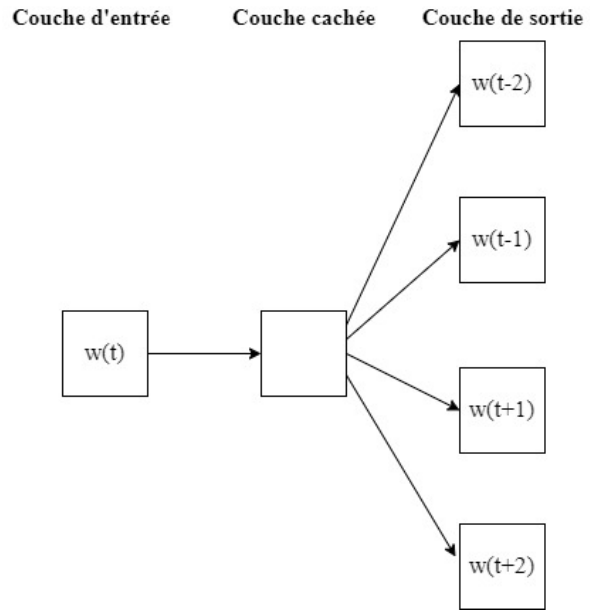


Figure 1.9: L'architecture du modèle Skip-gram. [27]

Les vecteurs formés codent clairement plusieurs régularités et modèles linguistiques, ce qui rend les représentations de mots calculées à l'aide de réseaux neuronaux particulièrement intrigantes.

1.7.2 GloVe

Une autre approche de la production de word embedding est appelée GloVe (Global Vectors for Word Representation). Elle est basée sur des algorithmes de factorisation de matrice mot-contexte. Vous construisez une grande matrice de données de cooccurrence et comptez le nombre de fois où chaque "mot" (les lignes) apparaît dans un "contexte" particulier (les colonnes) dans le corpus. La façon dont nous analysons généralement notre corpus est la suivante : pour chaque terme, nous recherchons les termes contextuels dans une région délimitée par une taille de fenêtre avant et une taille de fenêtre après le terme. De plus, nous attribuons moins de poids aux mots prononcés plus loin [26].

Naturellement, il existe de nombreux "contextes" car leur magnitude est essentiellement combinatoire. Ainsi, en factorisant cette matrice, on crée une matrice de dimension inférieure, dont chaque ligne contient désormais une représentation vectorielle de chaque mot. En général, on y parvient en réduisant une "perte de reconstruction". Cette perte recherche des modèles de dimension inférieure qui peuvent rendre compte de la majorité de la variance des données de haute dimension [26].

1.8 Conclusion

Dans ce chapitre, nous avons introduit plusieurs concepts clés tels que l'apprentissage automatique, l'apprentissage profond, l'apprentissage fédéré et le plongement lexical (Word embedding). Ces concepts constituent des fondements importants dans le domaine de l'intelligence artificielle et ont des applications variées.
Dans le prochain chapitre, nous aborderons un autre domaine essentiel :l'ingénierie sociale et l'hameçonnage. Nous examinerons en détail les différentes formes d'hameçonnage, y compris l'hameçonnage par SMS, ainsi que les travaux de recherche associés à ce domaine.

Chapitre 2

Généralités sur le phishing

2.1 Introduction

Le mode de vie des gens a changé avec l'arrivée d'Internet. De nos jours, les gens passent beaucoup de temps sur internet [28]. Malheureusement, cette exposition accrue à l'environnement numérique expose également les utilisateurs aux risques de cybercriminalité. L'ingénierie sociale, une technique de manipulation qui fait partie des crimes en ligne [28], consiste à l'utilisation de réseaux et d'ordinateurs pour commettre des actes criminels.

L'hameçonnage par SMS (smishing) est l'un des type d'attaque par ingénierie sociale, et une méthode de vol d'informations sensibles dans laquelle les SMS (Short Message Service) sont utilisés dans les cyberattaques. Lorsqu'ils reçoivent des SMS, Les individus ont tendance à être plus enclins à révéler des informations sensibles telles que des identifiants de compte et des mots de passe. Ces informations peuvent être exploitées pour escroquer des personnes ou des entreprises, leur faisant perdre des informations confidentielles et également de l'argent.

Dans ce chapitre, nous allons présenter l'ingénierie sociale, le phishing et ses types, et les travaux connexes pour la détection de smishing.

2.2 L'ingénierie sociale

Les méthodes d'ingénierie sociale sont actuellement le moyen le plus populaire pour infiltrer les systèmes informatiques et les infrastructures de technologies de l'information (TI) et commettre des cybercrimes [29].

L'ingénierie sociale est une attaque courante des pirates informatiques et une menace sérieuse, universelle et persistante pour la cybersécurité. Dans le contexte de la cybersé-

curité, l'ingénierie sociale désigne un type d'attaque dans laquelle l'attaquant exploite les vulnérabilités humaines (en utilisant des techniques comme la tromperie, la manipulation, l'influence, la persuasion, et l'incitation) pour violer les objectifs de sécurité (disponibilité, contrôlabilité, confidentialité, intégrité, et auditabilité) d'éléments du cyberespace (ressources, utilisateurs, infrastructure, données, et opérations) [30].

L'ingénierie sociale exploite les vulnérabilités humaines pour contourner ou franchir les barrières de sécurité, contrairement aux attaques plus conventionnelles comme le craquage de mots de passe par force brute et l'exploitation de vulnérabilités logicielles, qui nécessitent un codage profond pour combattre le pare-feu ou l'antivirus [30]. Les méthodes d'attaque sont les moyens par lesquels une attaque est réalisée, elles sont l'attaquant l'élabore et l'exécute pour réaliser une attaque spécifique [30] .

Une taxonomie courante dans la littérature consiste à diviser les attaques d'ingénierie sociale en deux catégories : les attaques informatiques et les attaques humaines. La figure 2.1 présente une vue d'ensemble de ces catégories et des instances correspondantes.

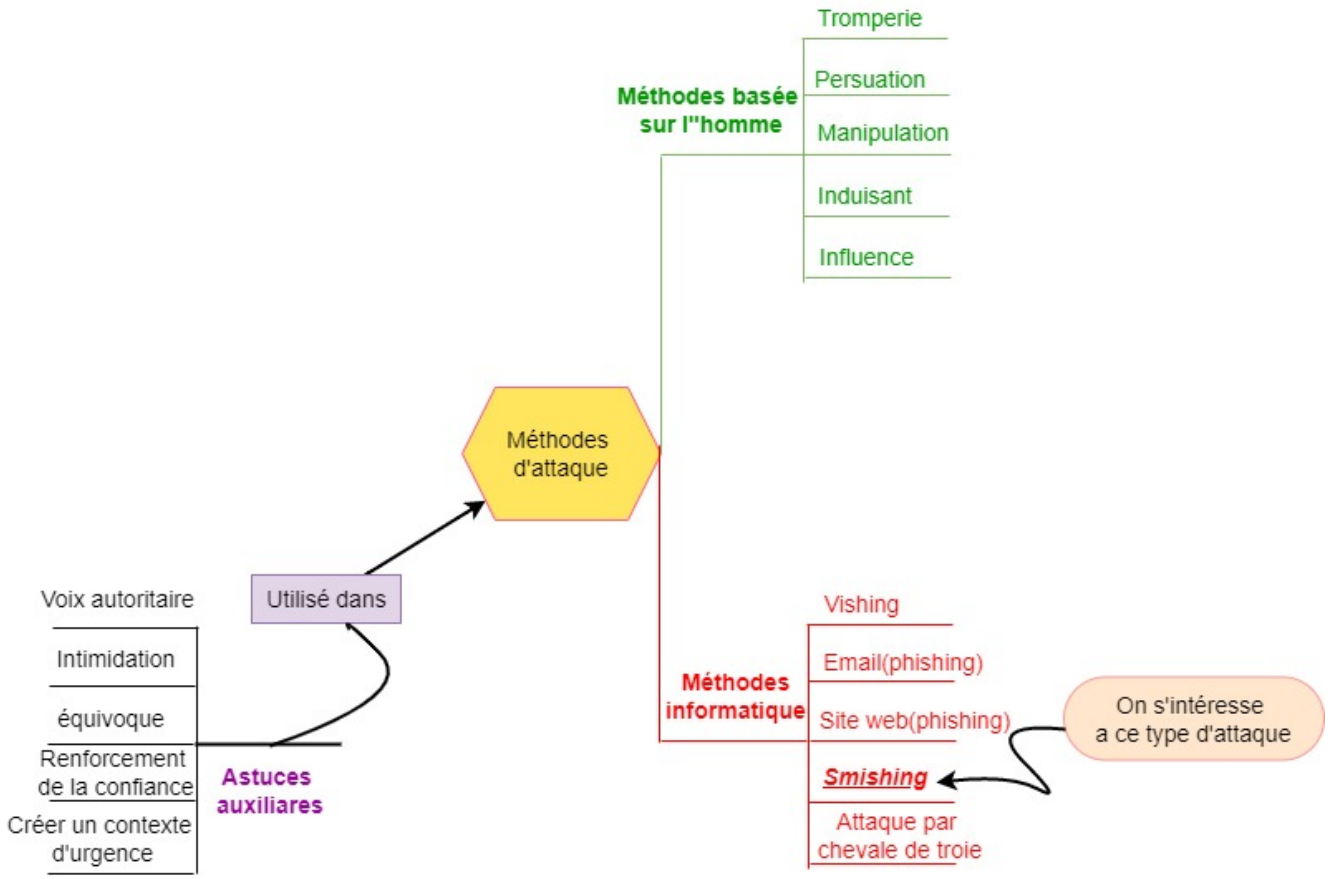


Figure 2.1: Méthodes d'attaque par ingénierie sociale

Les plus connues sont peut-être les escroqueries par hameçonnage (phishing), dans lesquelles des utilisateurs peu méfiants sont invités à cliquer sur un lien malicieux et, ce faisant, ils permettent aux pirates d'installer des logiciels malveillants et de pénétrer dans le système [29].

Les professionnels de la sécurité informatique et les observateurs politiques se sont récemment intéressés à un certain nombre de tentatives d'ingénierie sociale très médiatisées. Par exemple, en 2020 [29], des pirates ont ciblé la plateforme de médias sociaux Twitter, en particulier les profils publics de l'ancien président américain Barack Obama et du candidat démocrate de l'époque Joe Biden. Ils ont également ciblé les comptes de célébrités tels que Bill Gates, Elon Musk et Kanye West [29].

2.3 L'hameçonnage (phishing)

L'hameçonnage (phishing) est une approche d'ingénierie sociale qui vise à influencer une cible pour révéler des informations personnelles, telles qu'une adresse électronique, un mot de passe, un nom d'utilisateur, ou des informations financières en utilisant diverses méthodologies [31]. Ces informations sont ensuite utilisées par l'attaquant au détriment de la victime.

Le premier cas de cette technique a été signalé en 1995 [31], lorsque des attaquants ont utilisé le phishing pour persuader des victimes de divulguer les données de leur compte AOL (Online service provider company) [31].Par la suite, le phishing s'est développé, il est actuellement l'un des principaux vecteurs d'attaque utilisés par les pirates informatiques en raison des nouvelles stratégies d'attaque et de l'utilisation de nouveaux médias par les attaquants [31].

2.3.1 L'attaque par hameçonnage

L'attaque par hameçonnage comporte trois composantes : le vecteur, le support et l'approche technique. Pour le support du phishing, il y a trois supports qui sont la voix, l'internet et le service de messagerie court (SMS) [31]. L'Internet est le moyen le plus couramment utilisé pour le phishing, car il a ouvert de grandes possibilités pour le phisher de déployer l'attaque de phishing. Le vecteur est un endroit à partir duquel une attaque de phishing peut être lancée. Les courriers électroniques, les pages web et les réseaux sociaux sont des exemples de vecteurs sur l'internet [31].

Les méthodes de l'attaque par hameçonnage peuvent être classées en deux catégories : l'ingénierie sociale et l'attaque par logiciel malveillant [31]. L'ingénierie sociale exploite l'émotion de l'utilisateur qui craint de perdre quelque chose de précieux, ce qui l'amène

à révéler ses informations personnelles au phisher [31]. Pour l'attaque de phishing basée sur un logiciel malveillant, il installe secrètement des programmes malveillants afin de permettre au phisher d'accéder à l'ordinateur de l'utilisateur [31].

La figure 2.2 montre les étapes d'une attaque de phishing :

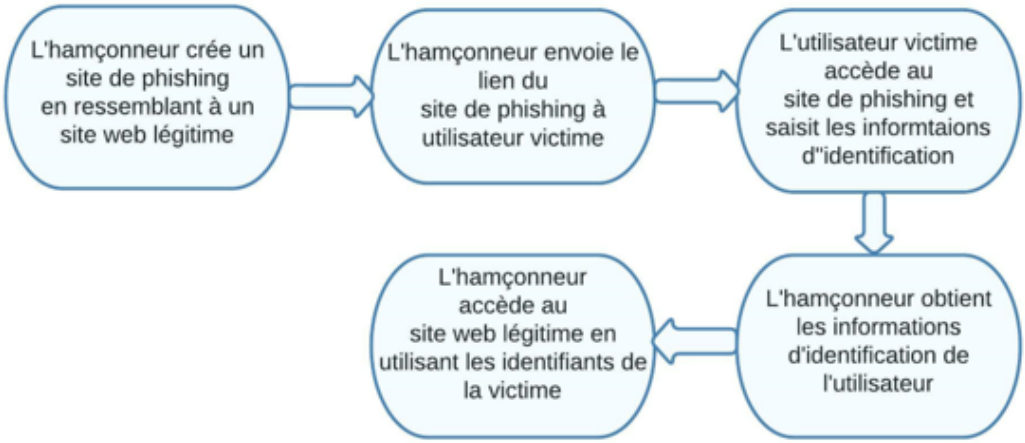


Figure 2.2: Les étapes d'une attaque de phishing [28]

2.3.2 Les vecteurs de l'hameçonnage

Les vecteurs dépendent du support utilisé par l'attaquant et constituent le canal par lequel l'attaque de phishing est menée [31], la figure 2.3 montre Comment les médias se transforment en vecteurs :

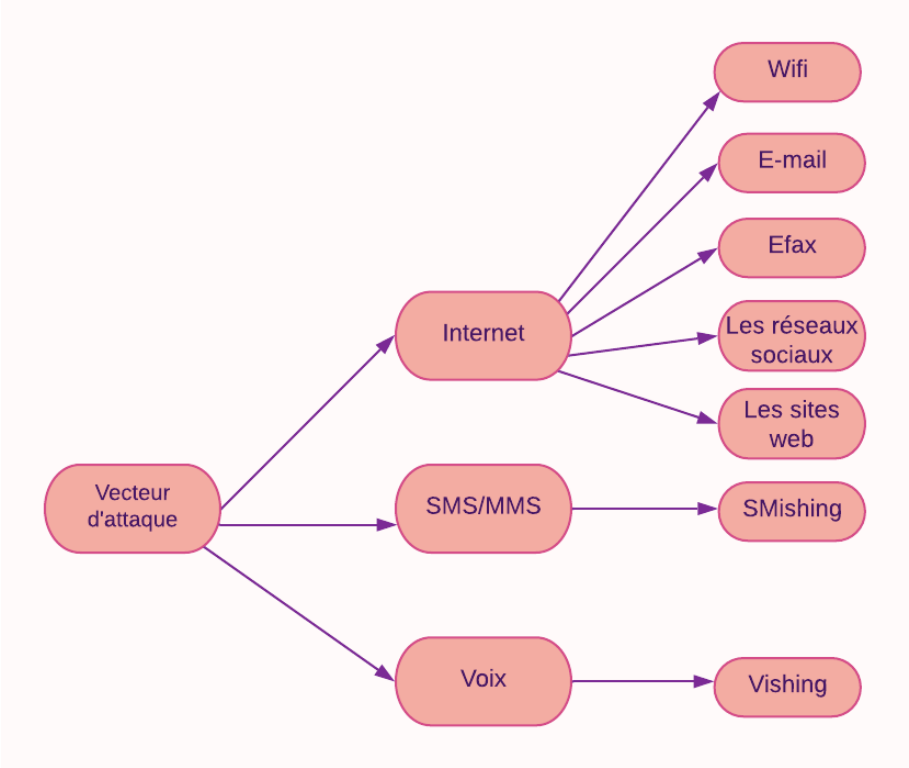


Figure 2.3: Transformation des médias sous forme de vecteurs [31]

2.3.2.1 E-mail

Le support qui comprend le plus grand nombre de vecteurs est l'internet [31]. Le courrier électronique(email) est le premier facteur à prendre en compte. Avec ce vecteur, des courriels spécialement conçus sont distribués aux cibles pour les inciter à effectuer des actions qui mettront leurs données personnelles à la disposition de l'attaquant. Comme les e-mails peuvent être facilement transférés à de nombreux destinataires, ils constituent un outil utile pour les hameçonneurs [31].

En outre, il permet à l'emplacement géographique de l'expéditeur de rester inconnue.

2.3.2.2 Efax

L'eFax est similaire à un fax traditionnel, mais sans la nécessité d'un télécopieur. Le protocole Internet (IP) est utilisé par des sites web comme efax.com pour transférer les fax, alors que les méthodes traditionnelles utilisaient des lignes téléphoniques [31].

L'avantage de cette méthode est que les télécopies peuvent être envoyées sur la machine du destinataire comme des courriels, ce qui élimine le besoin d'un télécopieur. Cependant, en raison de la nature en ligne de cette méthode de communication, elle ouvre une nouvelle voie pour les attaques de phishing visant à obtenir des informations personnelles sur les

victimes [31].

2.3.2.3 Les réseaux sociaux

Les réseaux sociaux se sont considérablement développés depuis le début du 21e siècle [31] et permettent aux gens d'interagir, de communiquer et de partager leurs expériences.

Citons par exemple Twitter, Facebook et LinkedIn, qui permettent aux utilisateurs de se connecter et d'identifier d'autres utilisateurs partageant les mêmes intérêts, les mêmes perspectives de vie, ou passe-temps. Toutefois, la principale utilisation de ces plateformes consiste à suivre les publications d'identités réelles. La nature du partage des données personnelles en ligne est une excellente ressource pour les hameçonneurs, qui peuvent ainsi identifier des groupes de cibles et potentiellement approcher les victimes [31].

2.3.2.4 Les sites Web

Les sites Web frauduleux sont une autre source d'attaque par hameçonnage. Ces sites semblent légitimes, mais ils sont utilisés pour collecter des informations personnelles auprès des victimes qui tentent de se connecter [31].

Comme Les internautes sont plus enclins à croire que les attaques de phishing sont principalement menées par le biais de courriers électroniques et d'autres services de messagerie, ils ont tendance à être moins conscients de la sécurité lorsqu'ils visitent des sites Web et à être plus attentifs aux attaques de phishing , ce qui les rend vulnérables à ces types de phishing [31].

2.3.2.5 Vishing

Le Vishing est la méthode de phishing qui implique l'utilisation de la voix. Il utilise la capacité à usurper un numéro de sorte qu'un appel semble provenir d'une source légitime [31].

L'introduction de la technologie de la voix sur IP (VoIP) a entraîné une augmentation de cette pratique, qui est utilisée pour masquer l'emplacement physique réel d'où provient l'appel, et la victime est alors manipulée pour révéler des informations.

2.3.2.6 SMishing

Le vecteur du smishing est imputable au support des SMS/MMS. Il s'agit de l'utilisation d'un service de messages courts pour mener des attaques de phishing [31].

Dans notre cas, nous sommes intéressés par le Smishing.

2.4 Smishing

Les attaques de type "smishing" utilisent les services de messages courts ou SMS, plus connus sous le nom de messages texte, il s'agit d'un type d'attaque par ingénierie sociale. Cette forme d'attaque est devenue de plus en plus populaire, car les gens sont plus susceptibles de faire confiance à un message qui leur parvient par le biais d'une application de messagerie sur leur téléphone qu'à un message délivré par courrier électronique [32].

Les hameçonneurs cherchent à voler les données personnelles, qu'ils peuvent ensuite utiliser pour commettre des fraudes ou d'autres cybercrimes, ils utilisent souvent l'une des deux méthodes suivantes pour voler ces données :

I. Malware : Le lien de l'URL de smishing peut vous inciter à télécharger un malware - un logiciel malveillant - qui s'installe sur votre téléphone. Ce logiciel malveillant par SMS peut se faire passer pour une application légitime, vous incitant à saisir des informations confidentielles et à les envoyer aux cybercriminels. La figure 2.4 illustre les étapes de l'attaque par le biais de Smishing Malware :

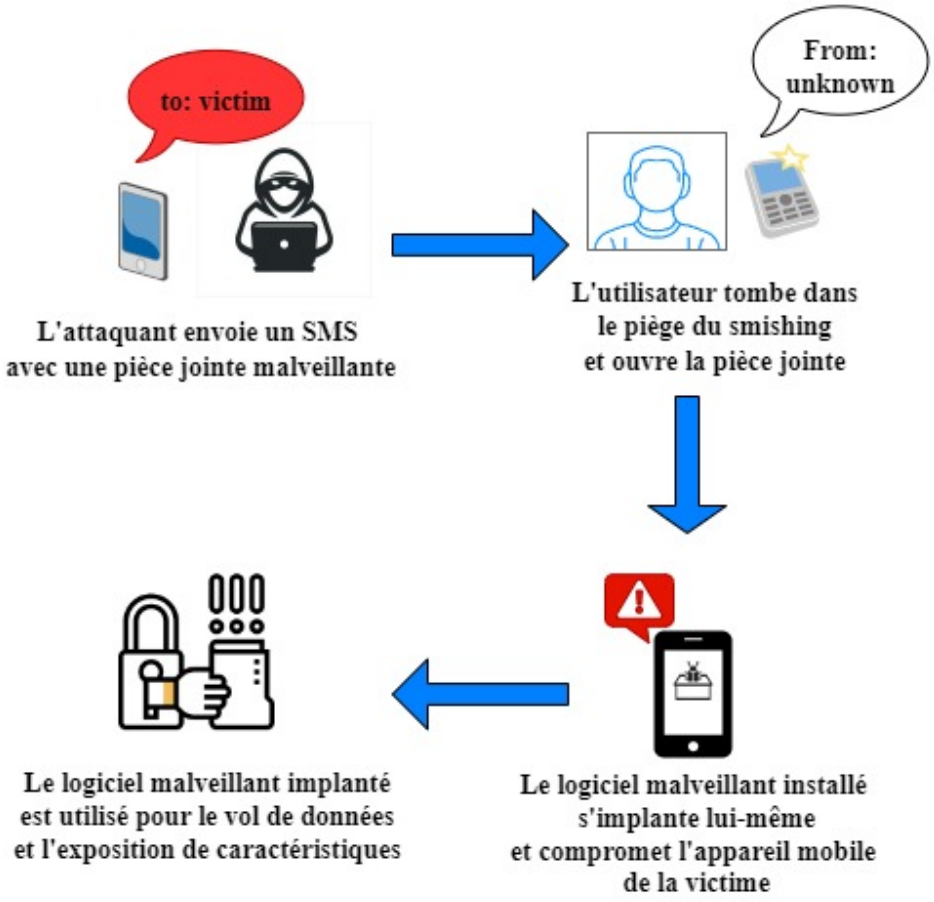


Figure 2.4: les étapes de l'attaque par le biais de Smishing Malware

II. Site web malveillant : Le lien contenu dans le message de smishing peut mener à un faux site qui vous demande de saisir des informations personnelles sensibles. Les cybercriminels utilisent des sites malveillants personnalisés, conçus pour imiter des sites réputés, ce qui facilite le vol de vos informations. La figure 2.5 illustre les étapes de l'attaque par le biais d'un Site web malveillant :

Figure 2.5: les étapes de l'attaque par le biais d'un Site web malveillant

2.4.1 Structure de SMS

SMS, qui signifie "Short Message" ou "Short Messaging Service", est simplement un message texte que vous pouvez envoyer et recevoir en utilisant votre smartphone tous les jours.

Les utilisateurs de téléphones mobiles échangent fréquemment de brefs textes au moyen du service de messages courts (SMS). Des entreprises renommées utilisent les SMS pour communiquer avec leurs clients et diffuser des informations, car elles peuvent atteindre un grand nombre de personnes pour un coût raisonnable avec un forfait SMS [33].

Un message SMS contient :

- Pièce jointe : Chaîne de caractères qui correspond au chemin d'association de la pièce jointe avec le message. Le message envoyé sera un MMS si cette variable est définie.

- Préfixe de pays : Chaîne de caractères contenant un préfixe international (213 pour l'Algérie). Si un préfixe national est donné et que le numéro du récepteur commence par "0", le "0" sera remplacé par le préfixe.

- Message : Chaîne de caractères qui représente le message envoyé ou reçu (jusqu'à 160 caractères).

- Le numéro : Chaîne de caractères contenant le numéro de téléphone de l'expéditeur ou du destinataire.

- Date de réception : Heure et date de réception du SMS.

2.4.2 Exemples de messages d'attaques par smishing

Les attaques de type "smishing" sont courantes, et certaines de leurs caractéristiques sont reconnaissables [34] :

• « Félicitations ! Vous avez gagné ! » Il s'agit d'un message d'escroquerie courant qui fait croire aux victimes qu'elles ont gagné un prix. Des liens ou des numéros de téléphone supplémentaires nécessitent souvent des données personnelles supplémentaires. Si vous n'avez pas participé au concours, vous n'avez probablement encore rien gagné.

• Message envoyé à un horaire suspect : La plupart des commerces sont ouverts entre 8h et 18h. Procédez avec prudence si vous recevez un message d'une source "légitime" tard dans la nuit ou tôt le matin.

• Message urgent de votre banque : Votre banque vous appellera très probablement personnellement en cas de demande urgente ou en cas de problème. Dans ces cas, la banque confirmera généralement vos données personnelles par téléphone. Si vous recevez un message urgent concernant votre activité bancaire, veuillez appeler votre banque pour plus d'informations.

• Erreurs d'orthographe et de grammaire : Les entreprises légitimes utilisent des écrivains professionnels. Si le message contient des fautes d'orthographe ou de grammaire, il est très probablement frauduleux.

La figure 2.6 représente un exemple d'un message de smishing :

Figure 2.6: Exemple d'un message de smishing

2.5 Les meilleures politiques pour éviter le Smishing

Comme de nombreuses personnes utilisent des appareils personnels tels que des smartphones, le smishing présente un danger pour les informations personnelles de l'utilisateur. Les meilleures politiques pour lutter contre les attaques de smishing sur les appareils mobiles sont :

- Vérifier l'authenticité du message reçu : Jouer les détectives peut valoir la peine si vous recevez un SMS de quelqu'un qui prétend être votre banque ou PayPal. Vous devez vérifier si : L'entreprise est propriétaire du numéro, un représentant de l'entreprise qui prend contact avec vous, Le numéro de référence du message est réel.

- N'oubliez pas que les entreprises de bonne réputation ne demandent pas d'informations personnelles par SMS.

- Ne saisissez jamais d'informations sensibles et ne cliquez jamais sur les hyperliens qui apparaissent dans les messages.

- Ne pas envoyez les informations personnelles à des numéros inconnus et non vérifiés.

- N'appelez jamais un numéro de téléphone mentionné dans un texte que vous n'attendiez pas.

- Il est évident que SmiShing est impliqué si le message indique "Cher utilisateur, Félicitations, vous avez gagné...". Parce que rien dans la vie n'est gratuit.

Malgré les conseils susmentionnés peuvent sans aucun doute aider à prévenir le smishing dans une certaine mesure, ils ne sont pas totalement infaillibles. Il est difficile de reconnaître et d'arrêter de telles attaques, car les stratégies des escrocs sont de plus en plus sophistiquées. C'est pourquoi les chercheurs se tournent vers l'intelligence artificielle (IA) pour mettre au point de meilleures techniques de détection du smishing.

Ces techniques impliquent l'utilisation d'algorithmes d'apprentissage automatique, apprentissage profond, et d'autres méthodes pour examiner le contenu des messages textuels, les détails de l'expéditeur et les métadonnées afin de repérer les schémas et les anomalies qui indiqueraient une attaque par smishing. Ces outils de pointe permettent d'identifier des tentatives de smishing qui auraient pu passer inaperçues en utilisant des procédures plus conventionnelles.

Étant donné que le smishing reste une menace importante pour les personnes et les entreprises du monde entier, investir dans des techniques de détection basées sur l'IA peut ajouter une couche supplémentaire de sécurité et protéger vos informations critiques des pirates informatiques.

2.6 Techniques utilisées pour détecter le Smishing

Il existe plusieurs méthodes de détection du smishing, parmi eux :

2.6.1 L'apprentissage automatique

La plupart des méthodes de détection de phishing par SMS reposent sur des techniques d'apprentissage automatique, et l'objectif essentiel de l'apprentissage automatique est de déterminer la relation entre les objets et leurs catégories pour la prédiction et la découverte de connaissances.

Une stratégie basée sur des règles pour reconnaître les communications de smishing a été proposée par [35] . Pour séparer les messages de smishing des messages authentiques, les auteurs ont défini neuf règles. De plus, ces règles ont été enseignées à l'aide de diverses méthodes de catégorisation, notamment PRISM, RIPPER et l'arbre de décision. Dans l'évaluation des performances de l'approche, plus de 99 % des messages négatifs réels sont négatifs.

[36] ont proposé une méthode basée sur le contenu pour identifier les messages de smishing dans un autre travail qui a été suggéré. Les mots les plus courants utilisés dans les SMS de smishing sont déterminés à l'aide d'un système d'apprentissage automatique. En outre, ce modèle évalue l'aspect de la page de connexion et le téléchargement du fichier .apk pour vérifier si l'URL est malveillante.

Un algorithme basé sur l'apprentissage automatique pour identifier les messages de smishing a été proposé par [37]. Le modèle classe les communications comme valides ou comme des messages de smishing en utilisant une combinaison d'algorithmes d'ingénierie des caractéristiques et d'apprentissage automatique. Au cours du processus d'ingénierie des caractéristiques, les caractéristiques pertinentes des messages sont extraites, telles que l'existence de mots ou de phrases particuliers. Le modèle utilise des arbres de décision, des forêts aléatoires et des machines à vecteurs de support comme algorithmes d'apprentissage automatique. Le modèle a été entraîné et testé sur un ensemble de données de messages authentiques et de messages de smishing.

[38] a utilisé quatre techniques de corrélation pour classer les caractéristiques dans l'effort d'étude le plus récent pour la détection du smishing, notamment la corrélation de rang de Pearson, la corrélation de rang de Spearman, la corrélation de rang de Kendall et la corrélation de rang de Point biserial. L'ensemble optimal de caractéristiques est finalement choisi pour identifier le smishing avec une précision de 98,40 %.

[39] ont utilisé un ensemble de données réelles de trois mois provenant de Chine (31,97 millions) pour détecter le smishing. Sur la base de trois résultats importants d'une enquête

empirique, ils ont développé et mis en œuvre un nouvel algorithme de détection, qui a réussi à identifier 90 801 messages de harponnage par SMS dans l'ensemble de données avec un taux de précision de 96,16 %.

Un système de détection du smishing en deux phases - vérification du domaine et classification des SMS - a été créé et testé avec succès par [40]. Pour évaluer la malice du SMS, chaque phase s'est concentrée sur une partie différente du message. Ils sont principalement concentrés sur l'examen de la légitimité de l'URL dans le SMS tout en minimisant la complexité du système. L'approche de Backpropagation a été utilisée pour développer le système, et la précision résultante était de 97,93%.

[41] ont présenté une technique de détection de la fraude par smishing basée sur l'apprentissage automatique. Sur la base des résultats de l'outil, ils ont découvert que Naive Bayes est un meilleur classificateur lorsque le score des faux négatifs est pris en compte, même si Random Forest produit de meilleures mesures d'évaluation. la précision résultante était de 90.59% avec l'algorithme Naive Bayes, et 98.15 % avec l'algorithme Random Forest.

Une technique de détection de smishing qui combine un classificateur de texte et un classificateur d'URL a été proposé par [42]. Le message a été analysé par le modèle, et si une URL a été trouvée, elle a été envoyée à un classificateur d'URL. Les classificateurs de vote sont utilisés dans la méthode proposée pour déterminer si un message est du smishing ou non. Les classificateurs de vote combinent les résultats de différents modèles. L'approche proposée produit un taux de précision de 98,94%.

[43] ont proposé une méthodologie basée sur l'apprentissage automatique pour détecter le Smishing. Le meilleur modèle, avec une précision de 99,86 %, est un modèle hybride utilisant la sélection de caractéristiques du classificateur Extratree et Random Forest employant la vectorisation TFIDF (Term Frequency Inverse Document Frequency). Les résultats sont comparés à un modèle Naive Bayes multinomial comme base de référence. Une base de données de 32259 messages en swahili est utilisée pour évaluer les performances.

2.6.2 L'apprentissage Profond

L'apprentissage profond a récemment attiré l'attention pour son efficacité dans l'évaluation de la cybersécurité, un système qui permet des évaluations complètes et concluantes de la cybersécurité. Il est utilisé pour détecter le phishing par SMS.

[44] ont utilisé Un modèle C-LSTM pour classification des sentiments et des questions à partir d'un ensemble de base de données spécifié. Le CNN et le RNN (réseau neuronal récurrent) sont fusionnés. Le RNN est utilisé pour créer des phrases à partir des phrases

récupérées une fois que les phrases sont extraites à l'aide du CNN.

[45] ont utilisé un modèle d'apprentissage profond basé sur CNN et LSTM pour la classification des communications de spam dans l'étude de recherche. Leur méthodologie est mise en action en trois étapes : d'abord, une matrice de mots est produite, puis des caractéristiques sont identifiées, et enfin, la classification est effectuée dans la troisième étape. La précision rapportée est de 99,44 %.

[46] a proposé un modèle de détection de smishing par l'apprentissage profond. L'approche proposée est mise à l'épreuve en utilisant plusieurs modèles de catégorisation. Le modèle récurrent à mémoire à long terme (LSTM), KNeighbors, Stochastic Gradient Descent (SGD), Decision Tree, Naive Bayes et Random Forest Classifier sont des exemples de modèles de classification utilisés. Le LSTM donne de meilleurs résultats que les autres classificateurs. La précision du modèle est de 95,11 %.

[47] ont utiliser un réseau neuronal pour créer un système efficace de détection du smishing. En outre, à l'aide d'un réseau neuronal, les sept principales caractéristiques des SMS de smishing sont extraites. La précision obtenue est comparée aux résultats de la classification des algorithmes d'apprentissage automatique. Avec une différence de 1,11 %, la comparaison démontre que les réseaux neuronaux ont produit des résultats d'une plus grande précision.

2.6.3 Autres méthodes

Il existent d'autres méthodes pour la détection des SMS phishing, tel que :

2.6.3.1 Méthodes heuristiques

Les chercheurs utilisent également la technique de catégorisation basée sur l'heuristique pour classer les SMS à l'aide d'algorithmes d'apprentissage automatique. Les chercheurs de ces travaux extraient un ensemble de caractéristiques de l'ensemble de données et catégorisent les SMS à l'aide de ces attributs.

Une technique basée sur l'heuristique pour identifier les messages de smishing a été proposée par [48]. Les auteurs ont utilisé des méthodes de classification pour classer les messages sur la base des 10 caractéristiques qu'ils avaient choisies dans les messages de smishing. Les auteurs ont testé leur stratégie à l'aide d'un ensemble de données éditées manuellement. Les résultats de leur évaluation indiquent un taux de précision de 98,74

2.6.3.2 Réseau neuronal moyen

Pour classer les communications dans les catégories spam ou authentique, un modèle de réseau neuronal moyen a également été utilisé pour les caractéristiques récupérées.

[49] ont utilisé des réseaux neuronaux moyennés avec une couche cachée pour créer un modèle précis permettant de reconnaître les messages de spam SMS sur la base de critères fondés sur le contenu. Les résultats de l'évaluation indiquent que les caractéristiques extraites ont une forte association avec la classe de messages et que la technique de réseau neuronal moyenné est capable de classer avec précision la classe de messages avec un taux de mesure F élevé. Les résultats indiquent un taux de précision de 98.8%.

2.6.3.3 Liste noire

Dans cette technique, les sites web dignes de confiance tiennent une liste d'URL et de domaines douteux qui peuvent être utilisés pour repérer les sites web frauduleux. Cette stratégie est utilisée par diverses applications intuitives [46].

Dans [50] les auteurs ont proposé l'utilisation d'une technique appelée "classificateur de smishing" pour classer les messages de smishing. Les trois phases de cette méthodologie sont l'analyse des SMS, la normalisation des SMS et la phase de classification des SMS. Après l'étude de l'URL contenue dans le message lors de la phase d'analyse du SMS, le message est traité de manière plus approfondie. Enfin, l'étape de classification des SMS permet de classer les SMS à l'aide de l'algorithme de classification Naive Bayes. La phase de normalisation des SMS normalise, ou convertit le texte en forme de racine, le texte présent dans le SMS. Le cadre proposé recherche l'URL et le numéro de téléphone mobile de l'expéditeur dans les listes noires.

Le tableau 2.1 synthétise les travaux connexes sur la détection de smishing, mettant en évidence les méthodes utilisées et les résultats obtenus :

Tableau 2.1: *Une comparaison entre les travaux connexes de détection des SMS phishing*

Recherche	Technique	Méthode	Topologie du modèle	Dataset	Résultats
[44]	Apprentissage Profond	CNN et RNN	Centralisé	11855 SMS	87.8%
[45]	Apprentissage Profond	CNN et LSTM	Centralisé	5574 SMS	99.4%
[46]	Apprentissage Profond	LSTM, KNeighbors, SGD, Decision Tree, Naive Bayes, Random Forest Classifier	Centralisé	5572 SMS	95.11%
[47]	Neural network	ANN	Centralisé	5858 SMS	97.91%.
[35]	Apprentissage Automatique	PRISM, RIPPER et l'arbre de décision	Centralisé	5574 SMS	99%
[36]	Apprentissage Automatique	Naive Bayes Classifier Random Forest classifier Decision Tree classifier	Centralisé	5572 SMS	96.29 %
[37]	Apprentissage Automatique	Decision trees, random forests, support vector machines	Centralisé	/ SMS	/
[38]	Apprentissage Automatique	Random Forest, DecisionTree Classifer ,AdaBoostClassifer ,Support Vector Machine	Centralisé	5578 SMS	98.40%
[39]	Apprentissage Automatique	Logistic regression	Centralisé	31,97 millions SMS	96.16%
[40]	Apprentissage Automatique	Backpropagation Algorithm, Random Forest, Naive Bayes, Decision Tree	Centralisé	5858 SMS	97.93%
[41]	Apprentissage Automatique	Naive Bayes, Random Forest	Centralisé	5000 SMS	98.15%
[42]	Apprentissage Automatique	XGB, GBDT, RF, BgC, KNN, ETC, DT, LR, AdaBoost, BNB, MNB, SVC, and GNB	Centralisé	5179 SMS, 507195 URL	98,94%
[43]	Apprentissage Automatique	Random Forest, Naive Bayes, SVM, KNN, Adaboost, Logistic Regression, Extra tree	Centralisé	32259 SMS	99.86%
[50]	liste noire	Naïve Bayesian Classifier, Bayesian Classifier	Centralisé	/	/
[48]	heuristique	SVM, Logistic Regression, Neural Network, Naive Bayes, Random Forest	Centralisé	5574 SMS	98,74 %
[49]	réseau neuronal moyen	/	Centralisé	5574SMS	98.8%.

La figure 2.7 démontre un résumé concis du tableau détaillé décrivant les travaux liés à la détection du Smishing, offrant une vue rapide et complète des différentes méthodologies employées dans la détection du Smishing, ainsi que leurs auteurs respectifs et l'année de publication :

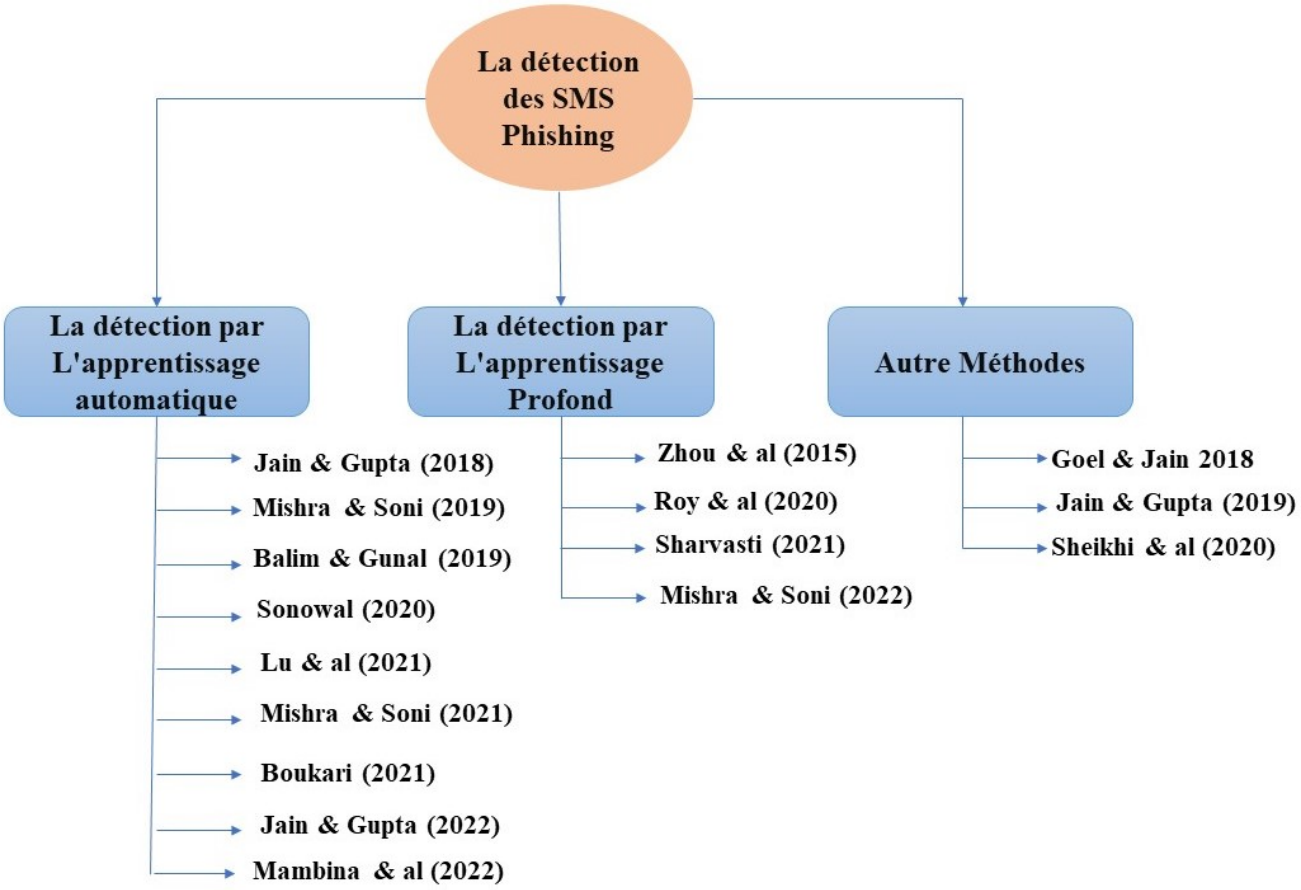


Figure 2.7: Aperçu des méthodologies de détection de smishing : Résumé des auteurs et des années

2.7 Synthèse

La technique la plus utilisée pour la détection des SMS Phishing est L'apprentissage Automatique. L'apprentissage Profond est rarement utilisé.

[47] Ont comparé entre les algorithmes de l'apprentissage automatique et les réseaux neuronaux. La comparaison entre ces deux méthodes révèle que les réseaux de neurones ont présenté une précision supérieure, avec une différence de précision de 1,11

[41] ont conclu que Naive Bayes est un meilleur classificateur lorsque le score des faux négatifs est pris en compte, même si Random Forest produit des résultats meilleurs.

Auprès les travaux connexes vu étudiés, nous remarquons que la topologie de modèle décentralisé n'a pas été utilisée auparavant.

2.8 Conclusion

Dans ce chapitre, nous avons présenté l'ingénierie sociale et ses diverses méthodes utilisées par les attaquants pour obtenir des informations sensibles. En nous concentrant plus particulièrement sur les attaques de phishing et de smishing. Nous avons examiné les vecteurs et les structures de ces attaques et discuté des travaux connexes sur la détection de l'hameçonnage par SMS.

Dans le chapitre suivant, nous présenterons l'architecture générale de la solution que nous proposons, ainsi que la démonstration de toutes les étapes.

Chapitre 3

La solution proposée

3.1 Introduction

La fréquence croissante des attaques de phishing par SMS et les lacunes des techniques de détection actuelles ont été démontrées dans le chapitre précédent. Dans ce chapitre, nous allons présenter la solution proposée pour la détection des SMS phishing en utilisant des algorithmes d'apprentissage fédéré et des données contextuelles pour identifier précisément les messages d'hameçonnage par SMS, ainsi que la démonstration de toutes les étapes.

3.2 Architecture de la solution proposée

Le smishing, ou hameçonnage par SMS, est une forme d'hameçonnage qui utilise des messages textuels pour tromper et manipuler les victimes afin qu'elles divulguent des informations sensibles, telles que des mots de passe, des numéros de carte de crédit ou des informations d'identité personnelle.

Pourtant, il peut être difficile de repérer les messages d'hameçonnage par SMS car ils ressemblent souvent à de vrais messages et ne comportent pas de liens sur lesquels il est possible de cliquer. Les techniques actuelles de détection du phishing par SMS se concentrent sur des stratégies basées sur des règles ou des signatures, qui sont inefficaces et que les attaquants peuvent facilement contourner. Il est donc nécessaire de trouver une solution fiable et précise de détection de l'hameçonnage par SMS qui permette d'identifier ces types d'attaques.

Dans notre quête de développement d'un système efficace de détection du smishing, nous avons exploré différents modèles d'apprentissage, y compris des approches d'appren-

tissage non fédérées et fédérées. La figure 3.1 montre l'architecture générale de la solution proposée :

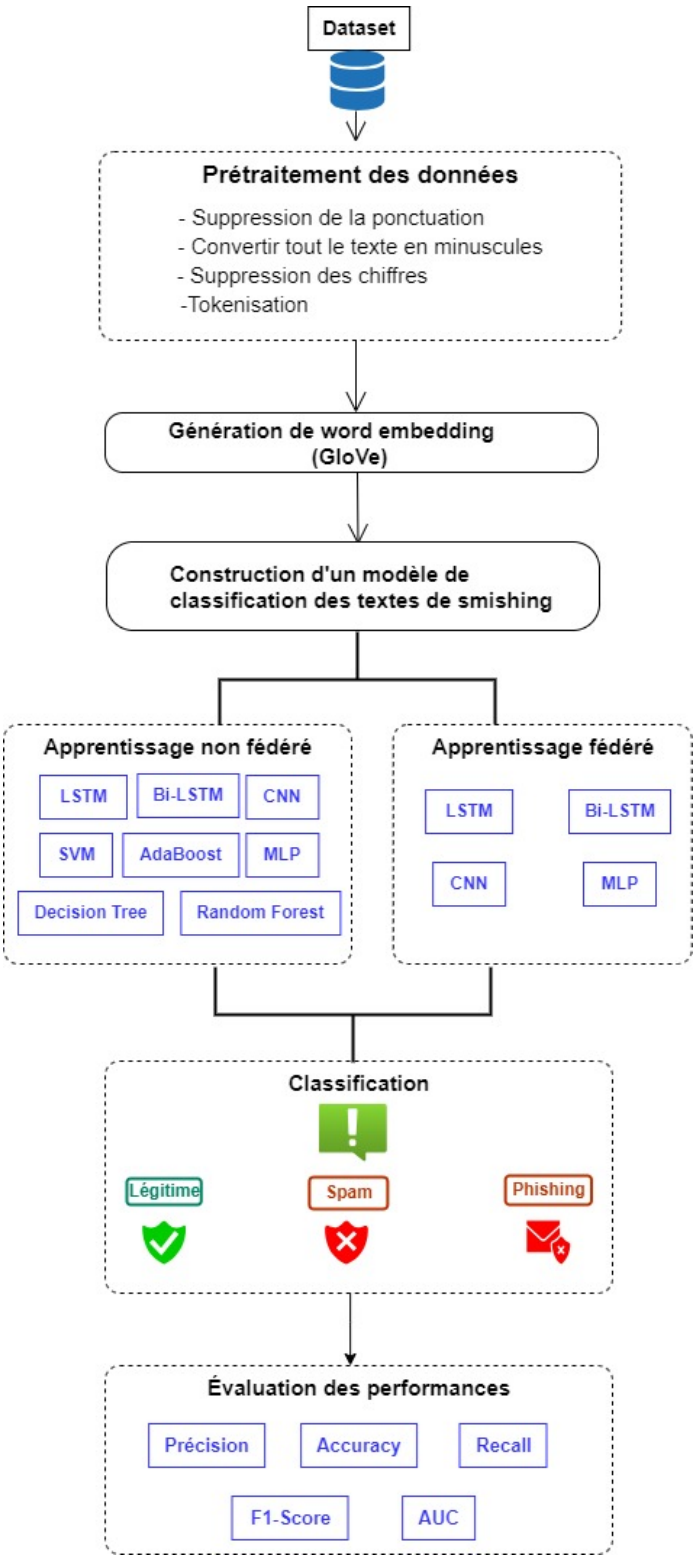


Figure 3.1: Pipeline de notre modèle

La première étape de l'approche proposée consiste à prétraiter l'ensemble de données. Diverses techniques de prétraitement telles que la tokenisation, la suppression des ponctuations et La conversion de tout le texte en minuscules sont appliquées aux données textuelles téléchargées. La deuxième phase consiste à transformer les données textuelles prétraitées en un espace vectoriel à haute dimension à l'aide de l'approche d'intégration GloVe. Dans la troisième étape, un modèle de classification est construit pour la classification des textes. Les données prétraitées et intégrées à GloVe sont utilisées pour entraîner le modèle de classification. Enfin, les performances du modèle de classification sont évaluées à l'aide de diverses mesures telles que la précision, le rappel, le score F1 et l'AUC afin de valider l'efficacité du système dans la détection des messages de smishing.

Initialement, nous avons mené des expériences en utilisant des modèles d'apprentissage non fédérés. Ces modèles ont été formés en utilisant divers algorithmes tels que LSTM, CNN, SVM, Decision Tree, Random Forest, MLP et AdaBoost. Nous avons évalué l'efficacité de ces modèles sur un ensemble de données de test distinct, en tenant compte de mesures telles que l'exactitude, la précision, le rappel et le score F1. Les connaissances obtenues à partir de ces expériences nous ont aidés à comprendre les capacités et les limites de l'apprentissage non fédéré.

La figure 3.2 présente l'architecture de modèle de l'apprentissage non fédéré :

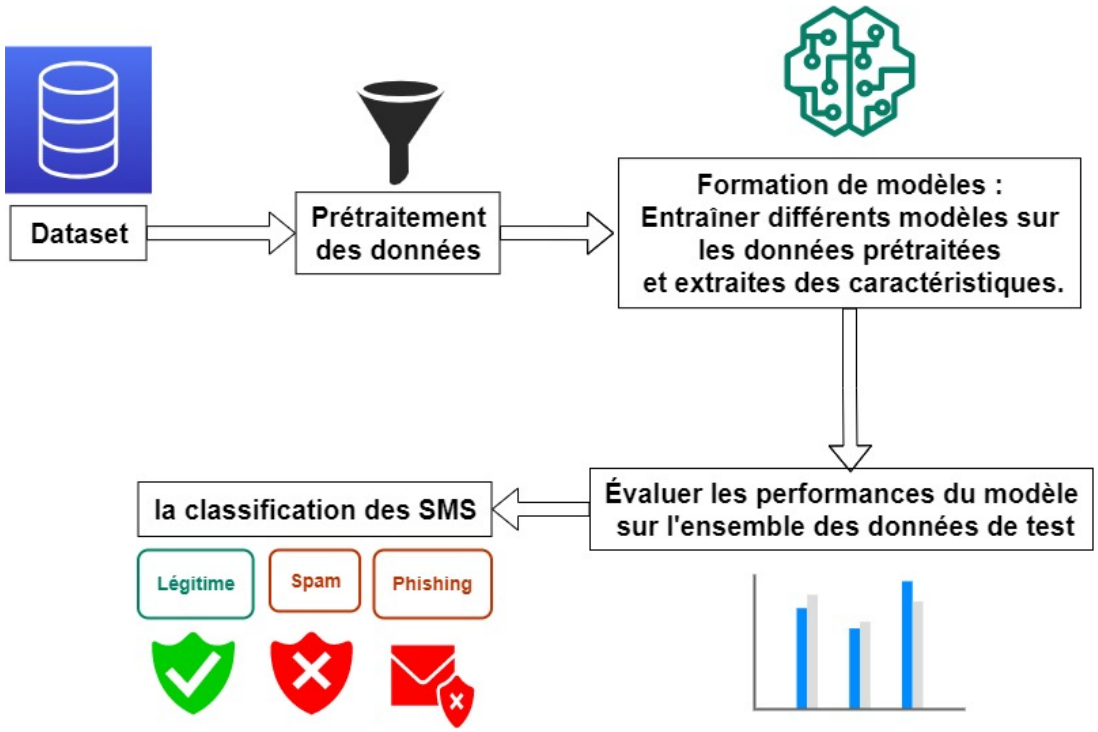


Figure 3.2: Architecture de modèle de l'apprentissage non fédéré pour la détection de smishing

Par la suite, nous avons exploré l'approche de l'apprentissage fédéré. L'apprentissage fédéré, une technique d'apprentissage automatique distribuée qui permet à de nombreux appareils ou serveurs de former ensemble un modèle d'apprentissage automatique sans partager leurs données brutes, est utilisé dans l'approche que nous proposons pour la détection de l'hameçonnage par SMS.

Pour catégoriser les phrases de smishing, nous avons formé un modèle d'apprentissage profond à l'aide d'une approche d'apprentissage fédéré. La confidentialité des données est assurée en entraînant le modèle localement et en n'envoyant que les modifications au serveur principal. La solution proposée répond aux problèmes de distribution inégale des données et de confidentialité des données dans la classification des textes de smishing.

La figure 3.3 présente l'architecture de modèle de l'apprentissage fédéré :

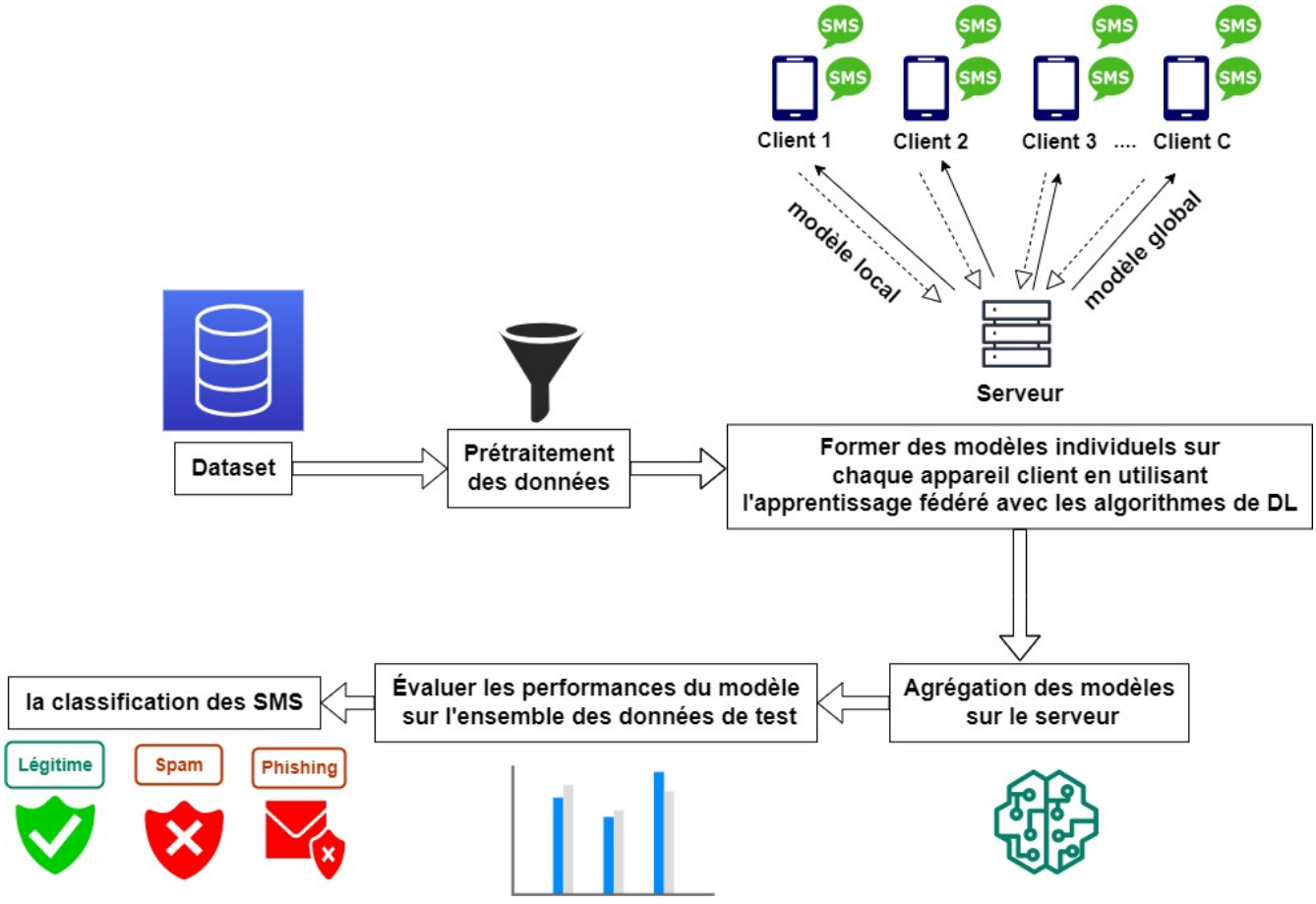


Figure 3.3: Architecture de modèle de l'apprentissage fédéré pour la détection de smishing

Dans l'ensemble, la solution que nous proposons peut améliorer la précision de la détection de l'hameçonnage par SMS et différencier les messages légitimes (ham) des messages indésirables (spam), tout en identifiant les messages d'hameçonnage. Toutefois,

l'efficacité du modèle dépendra de la qualité et de la représentativité de l'ensemble de données d'entraînement, ainsi que de la robustesse de l'algorithme d'apprentissage profond utilisé.

Dans l'algorithme 1, nous présentons l'algorithme de détection de Smishing, qui vise à classer les messages textuels comme "ham", "smishing" ou "spam" :

Algorithm 1 SMiShing Detection Algorithm

Require: Messages : A list of n text messages (Messages = [Message1, Message2, ..., MessageN])
Ensure: Message labels ("ham," "smishing," or "spam")

```
for all Message in Messages do
    if Message contains known SMiShing patterns then
        return "smishing"
    else if Message found in spam database then
        return "spam"
    else
        return "ham"
    end if
end for=0
```

3.3 Dataset

Cet ensemble de données sur l'hameçonnage par SMS, développé par Sandhya Mishra et Devpriya Soni [51], comprend un total de 5971 messages SMS qui ont été classés comme légitimes (Ham), spam ou hameçonnage. Il contient 489 messages de spam, 638 messages de phishing et 4844 messages légitimes.

La figure 3.4 représente le nombre de messages classés en tant que "smishing", "spam" et "ham" dans notre dataset.

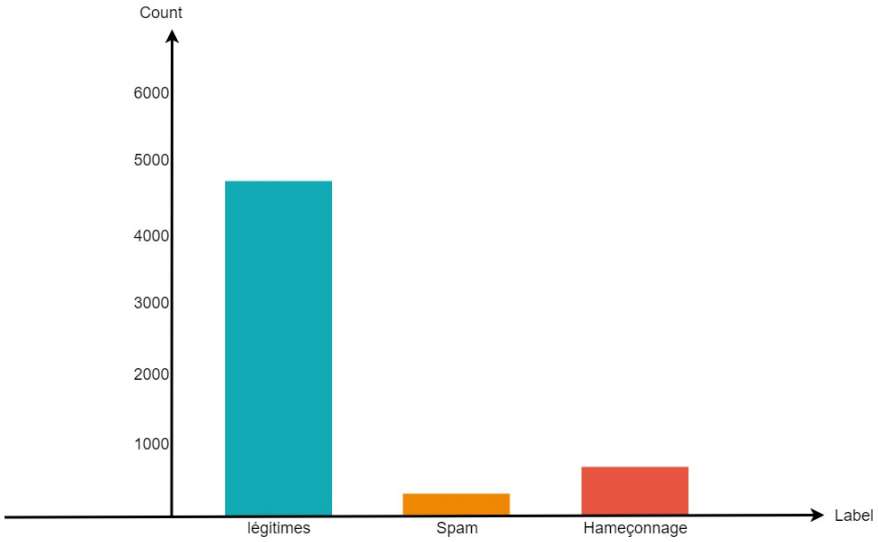


Figure 3.4: Représentation du Dataset

Le contenu brut du message dans l'ensemble de données, qui est conçu pour être utilisé dans la recherche sur le phishing par SMS, peut être utilisé pour le Deep Learning ou pour extraire des attributs supplémentaires. Il contient également des attributs extraits de messages malveillants qui peuvent être utilisés pour déterminer si un message est malveillant ou légitime. La figure 3.5 représente un exemple d'une partie de notre dataset :

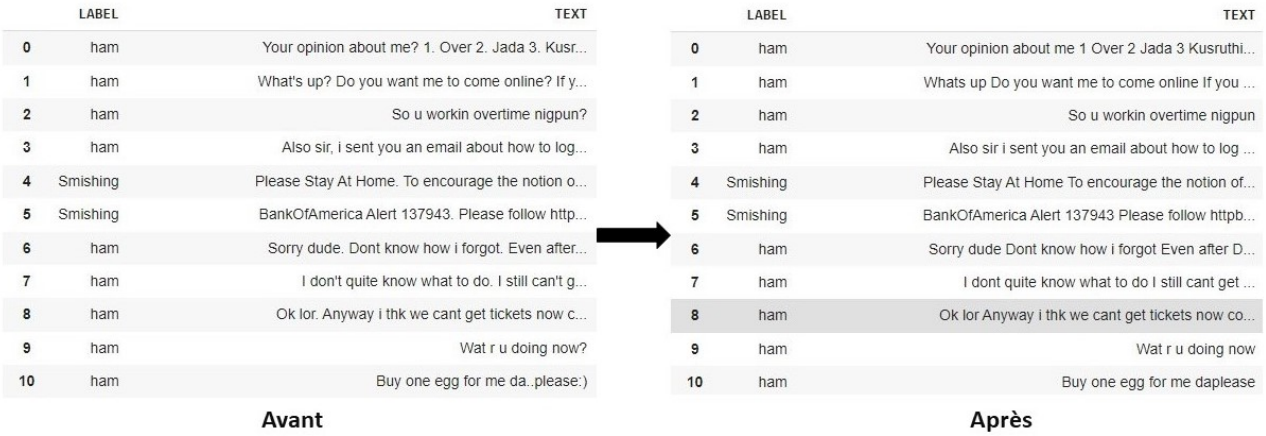
index	LABEL	TEXT
0	ham	Your opinion about me? 1. Over 2. Jada 3. Kusruthi 4. Lovable 5. Silent 6. Spl character 7. Not matured 8. Stylish 9. Simple Pls reply..
1	ham	What's up? Do you want me to come online? If you are free we can talk sometime🔲
2	ham	So u workin overtime nigpun?
3	ham	Also sir, i sent you an email about how to log into the usc payment portal. I.ll send you another message that should explain how things are back home. Have a great weekend.
4	Smishing	Please Stay At Home. To encourage the notion of staying at home. All tax-paying citizens are entitled to 🔲305.96 or more emergency refund. smsg.io/fCVbD
5	Smishing	BankOfAmerica Alert 137943. Please follow http://bit.do/cgjK-and re-activate
6	ham	Sorry dude. Dont know how i forgot. Even after Dan reminded me. Sorry. Hope you guys had fun.
7	ham	I don't quite know what to do. I still can't get hold of anyone. I cud pick you up bout 7.30pm and we can see if they're in the pub?
8	ham	Ok lor. Anyway i thk we cant get tickets now cos like quite late already. U wan 2 go look 4 ur frens a not? Darren is wif them now...
9	ham	Wat r u doing now?
10	ham	Buy one egg for me da..please:)
11	ham	Is there a reason we've not spoken this year? Anyways have a great week and all the best in your exam
12	ham	Stop the story. I've told him i've returned it and he's saying i should not re order it.
13	ham	No. I dont want to hear anything
14	ham	Waqt se pehle or naseeb se zyada kisi ko kuch nahi milta,Zindgi wo nahi he jo hum sochte hai Zindgi wo hai jo ham jeetey hai.....
15	ham	I'm e person who's doing e sms survey...
16	ham	Mm i had my food da from out
17	ham	4 tacos + 1 rajas burrito, right?
18	ham	House-Maid is the murderer, coz the man was murdered on 26th January..

Figure 3.5: Extrait de notre Dataset

3.4 Prétraitement des données

La première phase de la méthode proposée consiste à nettoyer et à convertir les messages SMS non traités dans un format approprié pour l'analyse. Ce processus comprend la suppression de la ponctuation, la conversion de tout le texte en minuscules, la suppression des chiffres et la tokenisation .

-Suppression de la ponctuation :Les points, les virgules et les points d'exclamation n'apportent pas grand-chose au texte, mais ils peuvent interférer avec les opérations de traitement du langage naturel telles que la tokenisation et l'analyse syntaxique et provoquer du bruit, ce qui rend difficile l'analyse des données textuelles. Par conséquent, l'élimination de la ponctuation peut contribuer à simplifier le texte et à le rendre plus facile à comprendre.



	LABEL	TEXT
0	ham	Your opinion about me? 1. Over 2. Jada 3. Kusr...
1	ham	What's up? Do you want me to come online? If y...
2	ham	So u workin overtime nigpun?
3	ham	Also sir, i sent you an email about how to log...
4	Smishing	Please Stay At Home. To encourage the notion o...
5	Smishing	BankOfAmerica Alert 137943. Please follow http...
6	ham	Sorry dude. Dont know how i forgot. Even after...
7	ham	I don't quite know what to do. I still can't g...
8	ham	Ok lor. Anyway i thk we cant get tickets now c...
9	ham	Wat r u doing now?
10	ham	Buy one egg for me da..please:)

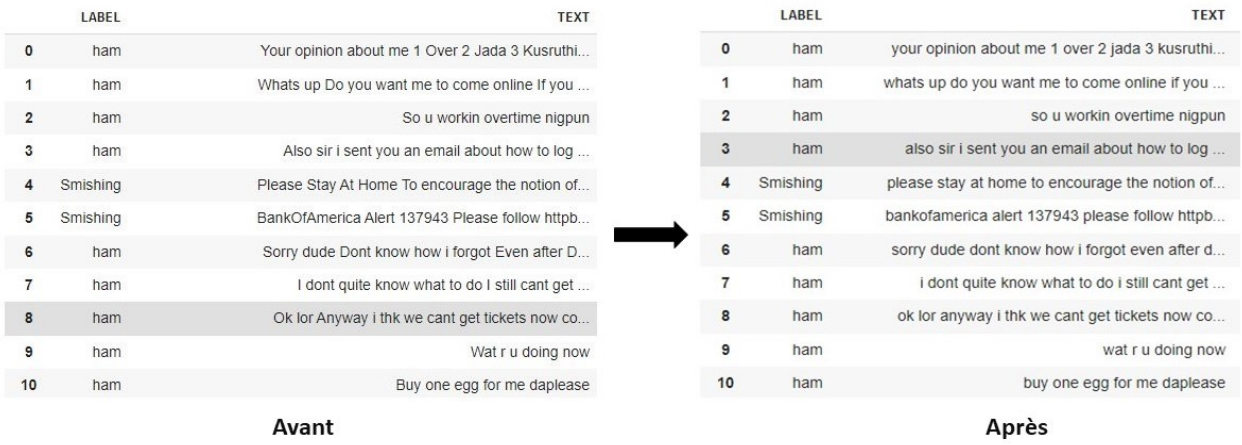
Avant

	LABEL	TEXT
0	ham	Your opinion about me 1 Over 2 Jada 3 Kusruthi...
1	ham	Whats up Do you want me to come online If you ...
2	ham	So u workin overtime nigpun
3	ham	Also sir i sent you an email about how to log ...
4	Smishing	Please Stay At Home To encourage the notion of...
5	Smishing	BankOfAmerica Alert 137943 Please follow httpb...
6	ham	Sorry dude Dont know how i forgot Even after D...
7	ham	I dont quite know what to do I still cant get ...
8	ham	Ok lor Anyway i thk we cant get tickets now co...
9	ham	Wat r u doing now
10	ham	Buy one egg for me daplease

Après

Figure 3.6: Extrait de Supression de la ponctuation

-La conversion de tout le texte en minuscules :Il s'agit de remplacer chaque caractère du texte par son équivalent en minuscules, ce qui permet de normaliser les données et de réduire le nombre de mots de vocabulaire avec des jetons uniques.



Avant

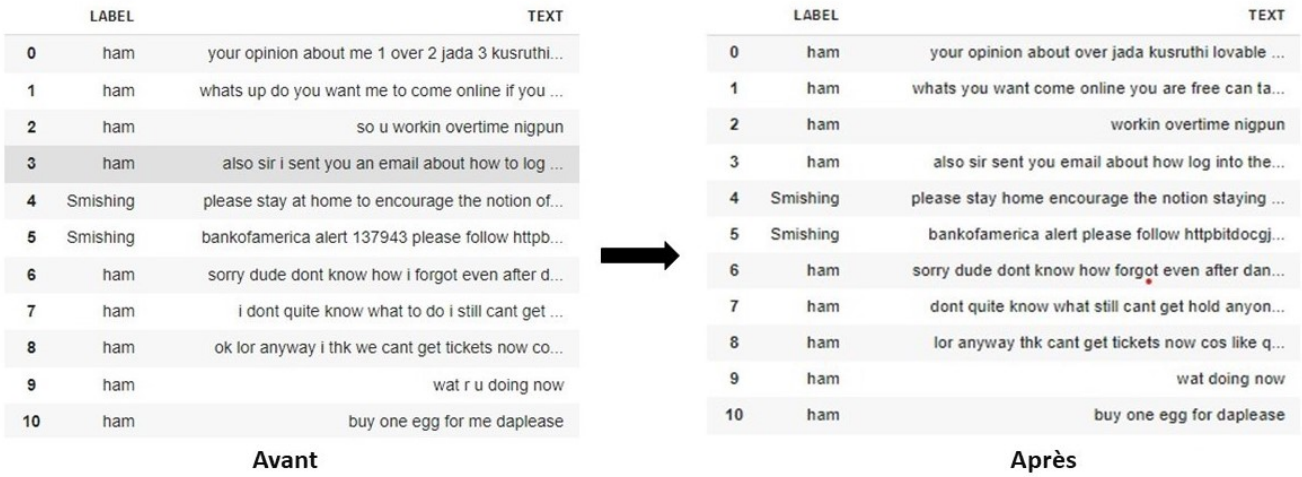
	LABEL	TEXT
0	ham	Your opinion about me 1 Over 2 Jada 3 Kusruthi...
1	ham	Whats up Do you want me to come online If you ...
2	ham	So u workin overtime nigpun
3	ham	Also sir i sent you an email about how to log ...
4	Smishing	Please Stay At Home To encourage the notion of...
5	Smishing	BankOfAmerica Alert 137943 Please follow httpb...
6	ham	Sorry dude Dont know how i forgot Even after D...
7	ham	I dont quite know what to do I still cant get ...
8	ham	Ok lor Anyway i thk we cant get tickets now co...
9	ham	Wat r u doing now
10	ham	Buy one egg for me daplease

Après

	LABEL	TEXT
0	ham	your opinion about me 1 over 2 jada 3 kusruthi...
1	ham	whats up do you want me to come online if you ...
2	ham	so u workin overtime nigpun
3	ham	also sir i sent you an email about how to log ...
4	Smishing	please stay at home to encourage the notion of...
5	Smishing	bankofamerica alert 137943 please follow httpb...
6	ham	sorry dude dont know how i forgot even after d...
7	ham	i dont quite know what to do i still cant get ...
8	ham	ok lor anyway i thk we cant get tickets now co...
9	ham	wat r u doing now
10	ham	buy one egg for me daplease

Figure 3.7: Extrait de La conversion de texte en minuscules

-La suppression des chiffres : Cette opération est effectuée pour supprimer toute donnée numérique qui pourrait ne pas être pertinente pour l'objectif précis de la détection de smishing. En supprimant les chiffres des messages, il est plus facile de se concentrer sur le texte et les modèles linguistiques.



Avant

	LABEL	TEXT
0	ham	your opinion about me 1 over 2 jada 3 kusruthi...
1	ham	whats up do you want me to come online if you ...
2	ham	so u workin overtime nigpun
3	ham	also sir i sent you an email about how to log ...
4	Smishing	please stay at home to encourage the notion of...
5	Smishing	bankofamerica alert 137943 please follow httpb...
6	ham	sorry dude dont know how i forgot even after d...
7	ham	i dont quite know what to do i still cant get ...
8	ham	ok lor anyway i thk we cant get tickets now co...
9	ham	wat r u doing now
10	ham	buy one egg for me daplease

Après

	LABEL	TEXT
0	ham	your opinion about over jada kusruthi lovable ...
1	ham	whats you want come online you are free can ta...
2	ham	workin overtime nigpun
3	ham	also sir sent you email about how log into the...
4	Smishing	please stay home encourage the notion staying ...
5	Smishing	bankofamerica alert please follow httpbitdocgj...
6	ham	sorry dude dont know how forgot even after dan...
7	ham	dont quite know what still cant get hold anyon...
8	ham	lor anyway thk cant get tickets now cos like q...
9	ham	wat doing now
10	ham	buy one egg for daplease

Figure 3.8: Extrait de La suppression des chiffres

-Tokenisation : La tokenisation est le processus qui consiste à séparer les mots individuels ou les jetons d'un document textuel ou d'une phrase. Plusieurs applications de traitement du langage naturel, telles que la traduction automatique et la catégorisation des textes, l'utilisent comme étape cruciale de prétraitement. La tokenisation peut être effectuée de différentes manières, par exemple en séparant le texte sur la base des espaces blancs ou de la ponctuation, ou en employant des méthodes plus complexes telles que les enchâssements de mots. Les tokens générés servent d'entrée pour les traitements de texte supplémentaires, tels que l'extraction de caractéristiques ou le traitement du langage na-

turel. En général, en divisant les données textuelles en unités plus petites et plus faciles à manipuler, la tokenisation facilite l'extraction d'informations utiles à partir des données textuelles.

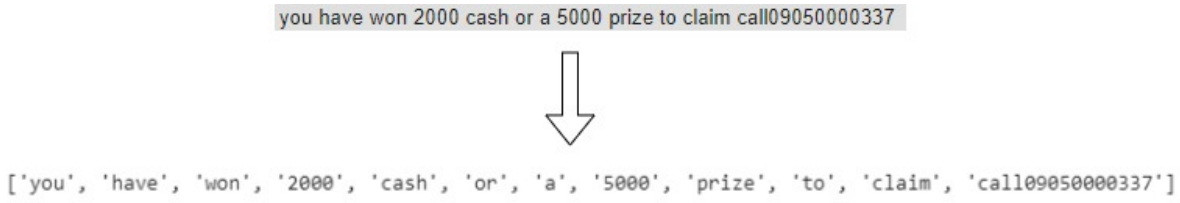


Figure 3.9: Extrait de La tokenisation

Voici le résultat des étapes de prétraitement appliquées à la base de données, qui ont permis d'obtenir un ensemble de données plus propre, plus cohérent et plus informatif, mieux adapté à l'analyse et à la prise de décision fondée sur les données :

index	LABEL	TEXT
0	2	your opinion about over jada kusruthi lovable silent spl character not matured stylish simple pls reply
1	2	whats you want come online you are free can talk sometime
2	2	workin overtime nigpun
3	2	also sir sent you email about how log into the usc payment portal ill send you another message that should explain how things are back home have great weekend
4	1	please stay home encourage the notion staying home all taxpaying citizens are entitled more emergency refund smsgiofcvbd
5	1	bankofamerica alert please follow httpbitdocgjkand reactivate
6	2	sorry dude dont know how forgot even after dan reminded sorry hope you guys had fun
7	2	dont quite know what still cant get hold anyone cud pick you bout 730pm and can see theyre the pub
8	2	lor anyway thk cant get tickets now cos like quite late already wan look frens not darren wif them now
9	2	wat doing now
10	2	buy one egg for daplease
11	2	there reason weve not spoken this year anyways have great week and all the best your exam
12	2	stop the story ive told him ive returned and hes saying should not order
13	2	dont want hear anything
14	2	waqt pehle naseeb zyada kisi kuch nahi miltazindgi nahi hum sochte hai zindgi hai ham jeetey hai
15	2	person whos doing sms survey
16	2	had food from out
17	2	tacos rajas burrito right

Figure 3.10: Extrait de notre Base de données aprés le Pré-traitement

3.5 Word Embedding

L'intégration de mots est une méthode utilisée dans le traitement du langage naturel pour représenter les mots dans un espace vectoriel à haute dimension. Word2vec, GloVe et fastText sont quelques-unes des techniques d'intégration de mots les plus répandues.

La méthode GloVe a été utilisée dans notre cas pour construire les ancrages de mots, car elle peut capturer les contextes globaux et locaux et a fait ses preuves dans une variété de tâches de traitement du langage naturel. En apprenant des représentations

vectorielles de mots à l'aide de données de cooccurrence, GloVe crée des enchâssements qui contiennent à la fois des informations syntaxiques et sémantiques. Pour que nos modèles d'apprentissage automatique puissent effectuer la tâche de détection du smishing, ces enchâssements seront utilisés comme caractéristiques d'entrée.

3.5.1 Application de Glove Embedding

Nous avons construit un dictionnaire de vecteurs d'intégration qui contient chaque mot du fichier Glove Embedding (840B, 100d) afin d'utiliser l'intégration Glove dans notre modèle. Les mots du fichier Glove servent de clés au dictionnaire, et la valeur de chaque clé correspond à un vecteur d'intégration découvert dans le même fichier. Ensuite, à l'aide des vecteurs d'intégration du fichier Glove, nous avons créé une matrice contenant uniquement les mots de notre vocabulaire. La couche d'intégration de notre modèle d'apprentissage profond est initialisée à l'aide de cette matrice.

```
ignoring [ 0.19791     0.237      -0.026908    0.1796     -0.43083999  0.35929
  0.16117001 -0.35975999  0.41068    -0.085815    0.63441998 -0.82369
 -2.97350001 -1.19980001  0.033558    0.40619999 -0.89915001 -0.15260001
 -0.63497001 -0.62651998  0.030563    0.36221999 -0.57604003 -0.043078
 -0.23289999  0.86098999  0.20212001 -0.057865   -0.0064906   0.48975
  0.46336001 -0.46292001 -0.29047    -0.43123999 -0.78365999  0.0054959
 -0.4716      0.023128    0.30489001 -0.14902     0.47819     0.96829998
  0.65781999 -0.40116999 -0.15279     0.27810001  0.031858   -0.10408
 -0.41486999  0.57216001 -0.66011    -0.112       0.77915001 -1.27170002
  0.56971002  0.82137001 -0.71513999 -0.18745001  0.14004999 -0.38569
  0.35861999  0.21256     0.24321     0.41802999 -0.27568999  0.83723003
  0.1885      0.97196001 -0.19437    -0.57336003 -0.97595     0.70213002
  0.31630999 -0.39329001  0.69594002  0.26659     0.15797    -0.21017
 -0.86550999  0.28487     0.13447     0.61588001 -0.43393999 -0.11094
 -0.024712    0.088142   -0.17455     0.14442     0.32767001  0.018175
 -0.25742    -0.30371001 -0.10781     0.65068001 -0.29802001  0.051217
 -0.5011     -0.55912    -0.029058    0.025709  ]
```

Figure 3.11: Exemple de l'application du Glove Embedding

3.6 Construction d'un modèle de classification

Nous étudions deux méthodes dans la construction d'un modèle de classification pour les textes de smishing : l'apprentissage non fédéré et l'apprentissage fédéré.

3.6.1 Apprentissage non fédéré

Pour la première méthode, nous expérimentons un certain nombre de modèles d'apprentissage profond et d'apprentissage automatique bien connus dans la technique d'apprentissage non fédéré, notamment LSTM, Bi-LSTM, CNN, SVM, Decision Tree, Random

Forest, MLP et AdaBoost. Ces modèles ont été testés à l'aide d'une variété de mesures de performance, après avoir été formés sur un ensemble de données assez important de textes de smishing prétraités.

3.6.2 Apprentissage fédéré

Dans la deuxième technique, nous avons utilisé une approche d'apprentissage fédéré qui nous permet de former un modèle sur des données décentralisées sans transférer les données entre les appareils. Dans cette méthode, nous nous sommes concentrés sur l'utilisation de LSTM, Bi-LSTM, CNN et MLP, qui sont des algorithmes d'apprentissage profond. Ces algorithmes ont été choisis en raison de leur propension à gérer les liens complexes et les modèles observés dans les données textuelles de smishing.

Les algorithmes d'apprentissage en profondeur, par contre aux algorithmes d'apprentissage automatique , sont excellents pour identifier des modèles et des connexions complexes dans les données en utilisant de nombreuses couches de réseaux neuronaux. Cela est particulièrement avantageux dans le cas de la détection du smishing, car les messages textuels contiennent souvent des informations linguistiques et contextuelles subtiles que les modèles d'apprentissage profond peuvent mieux saisir.

L'algorithme 2 présente une vue d'ensemble d'apprentissage fédéré pour la détection de l'hameçonnage, qui se compose d'un serveur de coordination et de plusieurs dispositifs clients. Chaque client, représenté par C, où C $\in$ [1, 2, 3...C] possède son propre ensemble de données locales relatives à l'hameçonnage.

Algorithm 2 Federated Learning for Smishing Detection

Input : Smishing dataset

Output : Model performance (e.g., Accuracy, F1-score, and Precision)

/* **Server-side** */ **Server :** Initialize and send global model W_t to all C clients

for *each epoch* $e \in E$ **do**

  for *each client* $c \in \{1, 2, ..., C\}$ *in parallel* **do**

    $W_{c_t} \leftarrow \text{ClientUpdate}(W_{c_t})$

    ; /local updates

  Perform weighted averaging and update the global model : $W_{t+1} \leftarrow \sum \left(\frac{n_c}{n}\right) \cdot W_{c_t}$, where n_c is the number of samples in client c and n is the total number of samples

  Send the updated global model W_{t+1} to all clients

/* **Client-side at each client** c */ **ClientUpdate**(W_{c_t}) **:** /* **Runs once at the beginning** */ Prepare smishing dataset :

— Data extraction

— Data preprocessing

— Perform an 80 :20 train-test split

— Tokenization

/* **Runs repetitively during training/testing** */ **while** *global model* W_t *is received from the server* **do**

  Set $W_{c_t} = W_t$ /* **Training/testing on the local smishing dataset** X **with** n_c **samples** */ **for** *each batch* $b \in B$ **do**

    $W_{c_t} \leftarrow W_{c_t} - \eta \cdot O_f(W_{c_t}; X)$, for $X \sim P_c$

    /* B **is the batch size,** η **is the learning rate, and** $O_f(W_{c_t}; X)$ **represents gradients with respect to the cost function** */

  Send locally trained W_{c_t} to the server for aggregation

La procédure d'apprentissage fédéré se déroule comme suit :

1- Chaque client entraîne un modèle à l'aide de son ensemble local de données de détection de smishing, désigné par D_C. À l'instant t, cette procédure de formation produit un modèle local, WC_t. Chaque client forme son modèle indépendamment en utilisant ses propres données et ressources.

2-Une fois les modèles locaux formés, tous les clients soumettent leurs modèles respectifs au serveur et les agrègent. En tenant compte de la contribution du modèle de chaque client, le serveur procède alors à une moyenne pondérée ou à une agrégation des modèles locaux reçus. Cette phase d'agrégation aboutit à la mise à jour W_{t+1} du modèle global.

3-Après l'agrégation, le modèle global mis à jour est communiqué à tous les clients participants. Cela garantit que tous les clients ont accès à la version la plus récente du modèle. La synchronisation des modèles facilite l'alignement des modèles des clients, ce qui favorise la cohérence et l'apprentissage collaboratif.

4-Processus itératif : Les clients répètent les étapes de formation du modèle local,

de téléchargement du modèle, d'agrégation et de synchronisation du modèle au fur et à mesure que le processus d'apprentissage fédéré se poursuit de manière itérative. Chaque itération équivaut à une époque globale. Le processus se poursuit jusqu'à ce que le modèle global converge et atteigne les performances souhaitées en matière de détection de l'hameçonnage.

Dans le cadre d'apprentissage fédéré pour la détection de l'hameçonnage, le serveur coordonne le processus de formation tout en préservant la confidentialité des données et en protégeant la vie privée. Pendant la formation, les clients, tels que les institutions financières, ne sont pas tenus de partager leurs données originales relatives à l'hameçonnage avec le serveur ou d'autres clients. Au lieu de cela, seuls les paramètres du modèle (c'est-à-dire les informations apprises) sont transmis entre les clients et le serveur, ce qui garantit la confidentialité des données locales tout au long du processus de formation.

3.6.3 Long Short-Term Memory (LSTM)

• **Apprentissage non fédéré** : La technique LSTM est utilisée dans le cadre d'apprentissage non fédéré pour examiner la nature séquentielle des messages de smishing. Le modèle LSTM commence par une couche d'intégration qui transforme les entrées de séquences en une représentation vectorielle dense. Le composant principal du modèle LSTM est la couche LSTM, qui comprend 50 cellules de mémoire. Cette couche est chargée de capturer et de stocker les dépendances à long terme dans les séquences d'entrée. La couche LSTM traite séquentiellement les données d'entrée, ce qui permet de modéliser les relations temporelles dans les messages de smishing. Une fonction d'activation softmax est utilisée pour dériver des probabilités pour chaque catégorie sur la base de la sortie de la couche LSTM. En utilisant la LSTM pour l'apprentissage non fédéré, le modèle peut apprendre efficacement à partir du caractère séquentiel des données de smishing et faire des prédictions précises.

• **Apprentissage fédéré** : La technique LSTM est modifiée pour l'apprentissage distribué dans le cadre de l'apprentissage fédéré, où différents appareils clients fournissent leurs données locales pour l'apprentissage sans partager les données brutes elles-mêmes. Sur son ensemble de données locales, chaque appareil client entraîne individuellement son propre modèle LSTM . Les modèles clients transmettent ensuite leurs modifications à un serveur central, où elles sont combinées à d'autres mises à jour pour produire un modèle LSTM global à l'aide de federated averaging algorithme .

L'architecture LSTM est composée d'une seule couche LSTM qui contient 250 cellules de mémoire. Cette couche est chargée de capturer et de stocker les dépendances à

long terme dans les messages de spam. La couche LSTM traite séquentiellement les données d'entrée, ce qui permet de modéliser les relations temporelles dans les données de smishing. En employant la LSTM dans l'apprentissage fédéré, le modèle peut apprendre efficacement à partir du caractère dispersé et confidentiel des données. La couche LSTM permet au modèle de gérer la nature séquentielle des données d'hameçonnage et de faire des prédictions précises tout en maintenant la sécurité et la confidentialité des données sur plusieurs appareils ou clients participant au processus d'apprentissage fédéré.

L'équation d'une cellule LSTM est donnée par [52] :

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3.1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3.2}$$

$$\hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{3.3}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \hat{C}_t \tag{3.4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{3.5}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{3.6}$$

$$y_t = softmax(W_y \cdot h_t + b_y) \tag{3.7}$$

D'où :

- x_t : est le vecteur d'entrée à l'instant t.

- h_{t-1} : est l'état caché (Hidden State) à l'instant précédent .

- i_t, f_t, o_t : sont les vecteurs de portes d'entrée (Input Gate) , d'oubli (Forget Gate) et de sortie (Output Gate) respectivement.

- $\hat{C}_t$: est le vecteur de candidat de cellule mémoire (Candidate Cell State).

- C_t : est le vecteur de cellule mémoire à l'instant t (Cell State).

- σ : représente la fonction d'activation sigmoïde.

- tanh : représente la fonction d'activation tangente hyperbolique.

-Softmax : est une fonction d'activation utilisée pour obtenir une distribution de probabilité sur les sorties.

- W_f W_i W_c W_o W_y sont des matrices de poids.
- b_f b_i b_c b_o b_y sont des vecteurs de biais.

3.6.4 Bi-LSTM

• **Apprentissage non fédéré** : La technique Bi-LSTM est utilisée dans l'apprentissage non fédéré pour examiner les propriétés séquentielles des messages textuels de smishing. Le modèle Bi-LSTM se compose d'une couche d'intégration suivie d'une couche LSTM bidirectionnelle à 250 cellules de mémoire. La couche d'intégration transforme les messages de smishing en représentations numériques, ce qui permet au modèle de traiter efficacement les données. La couche LSTM bidirectionnelle est composée de deux LSTM, dont l'une traite la séquence d'entrée dans le sens direct et l'autre dans le sens inverse. Cela permet au modèle de capturer des informations contextuelles à la fois passées et futures, améliorant ainsi sa capacité à comprendre les modèles séquentiels dans les données de smishing. En exploitant la nature bidirectionnelle de la LSTM, le modèle peut effectivement apprendre des relations temporelles dans les messages d'hameçonnage, ce qui améliore la précision de la détection des attaques d'hameçonnage.

• **Apprentissage fédéré** : En revanche, l'algorithme Bi-LSTM est modifié pour un environnement décentralisé dans le contexte de l'apprentissage fédéré. Chaque appareil client entraîne individuellement son propre modèle Bi-LSTM à l'aide de son propre ensemble de données local. Pour créer un modèle Bi-LSTM global, le serveur central combine les mises à jour des modèles à l'aide de federated averaging algorithme . Le modèle Bi-LSTM se compose d'une couche d'intégration suivie d'une couche LSTM bidirectionnelle de 50 cellules. La couche d'intégration transforme les messages de smishing en représentations numériques pour faciliter le traitement des données du modèle. Combinant deux couches LSTM, la couche LSTM bidirectionnelle traite la séquence d'entrée dans les deux sens. Cela permet au modèle de capturer les informations contextuelles des séquences passées et futures, améliorant ainsi sa compréhension des modèles séquentiels dans les données de smishing. Les messages de smishing sont classés en trois catégories à l'aide d'une couche dense avec activation softmax. En utilisant la nature bidirectionnelle de la LSTM et de la couche d'intégration, le modèle Bi-LSTM en apprentissage fédéré améliore la précision de détection des attaques de smishing.

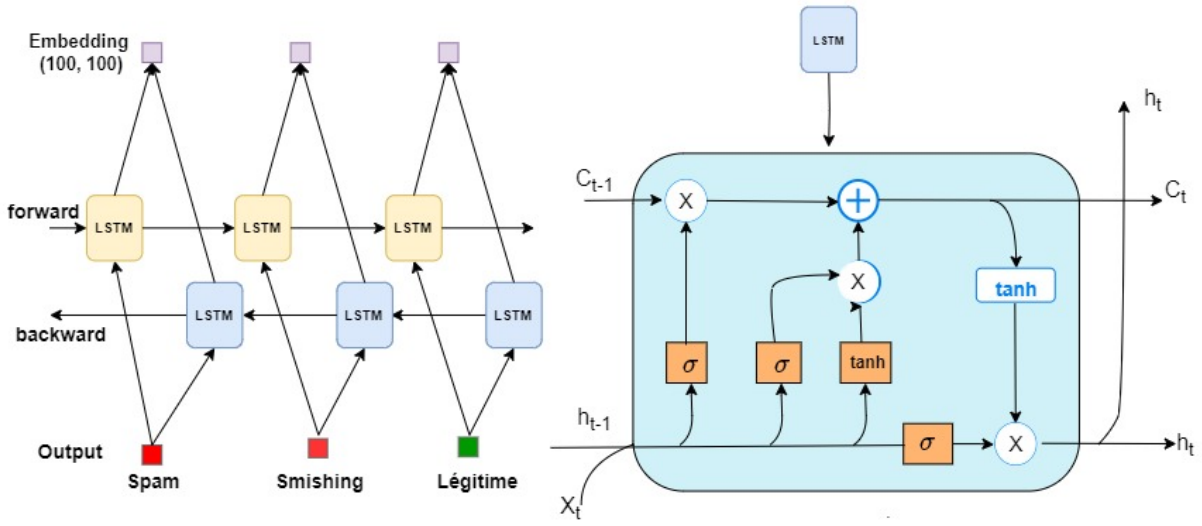


Figure 3.12: Un modèle de réseau Bi-LSTM

3.6.5 Réseaux de neurones convolutifs CNN

• **Apprentissage non fédéré** : L'algorithme CNN (réseau neuronal convolutif) est utilisé dans le cas d'apprentissage non fédéré pour identifier le smishing en extrayant les caractéristiques pertinentes des messages textuels. En commençant par une couche d'intégration qui fait correspondre les séquences d'entrée à une représentation vectorielle dense, la procédure passe ensuite à l'étape d'extraction. L'architecture du CNN comprend une couche Conv1D avec 50 filtres, une taille de noyau de trois, et un remplissage " équivalent ". Afin de réduire les dimensions spatiales, cette couche est suivie d'une couche MaxPooling1D avec une taille de pool de 2. En utilisant la fonction d'activation ReLU, la sortie est ensuite mise à plat et transmise à travers 3 couches Dense avec 512, 128 et 64 unités, respectivement. Ces couches permettent d'extraire des attributs significatifs des données d'entrée. Le modèle CNN est entraîné à l'aide de l'initialisateur de noyau "he_uniform", et la dernière couche utilise l'activation softmax pour classer les messages de smishing en trois catégories différentes.

• **Apprentissage non fédéré** : L'algorithme CNN est modifié pour un environnement distribué dans le cadre de l'apprentissage fédéré. Chaque appareil client entraîne son propre modèle CNN. Les couches de convolution et de mise en commun extraient les caractéristiques locales dans les modèles clients, qui ont une architecture similaire à celle du CNN centralisé. Le serveur central est ensuite informé de toute modification du modèle. En combinant les paramètres du modèle ,le serveur central agrège les mises à jour du

modèle à l'aide de federated averaging algorithm. Le modèle CNN global final incorpore diverses caractéristiques provenant de nombreux ensembles de données et représente les connaissances mises en commun de tous les appareils clients.

En commençant par une couche d'intégration qui fait correspondre les séquences d'entrée à une représentation vectorielle dense, la procédure passe ensuite à l'étape d'extraction. L'architecture du CNN comprend une couche Conv1D avec 128 filtres et une taille de noyau de 3, activée par la fonction ReLU. Afin de réduire les dimensions spatiales, cette couche est suivie d'une couche MaxPooling1D avec une taille de pool de 4. La sortie est ensuite aplatie pour produire un vecteur de caractéristiques à une seule dimension. Les messages de smishing sont classés en trois catégories à l'aide d'une couche dense de 3 unités et d'une fonction d'activation softmax. Ce modèle CNN, lorsqu'il est incorporé dans le cadre de l'apprentissage fédéré, permet une formation collaborative sur des ensembles de données distribuées sans compromettre la sécurité ou la confidentialité des données.

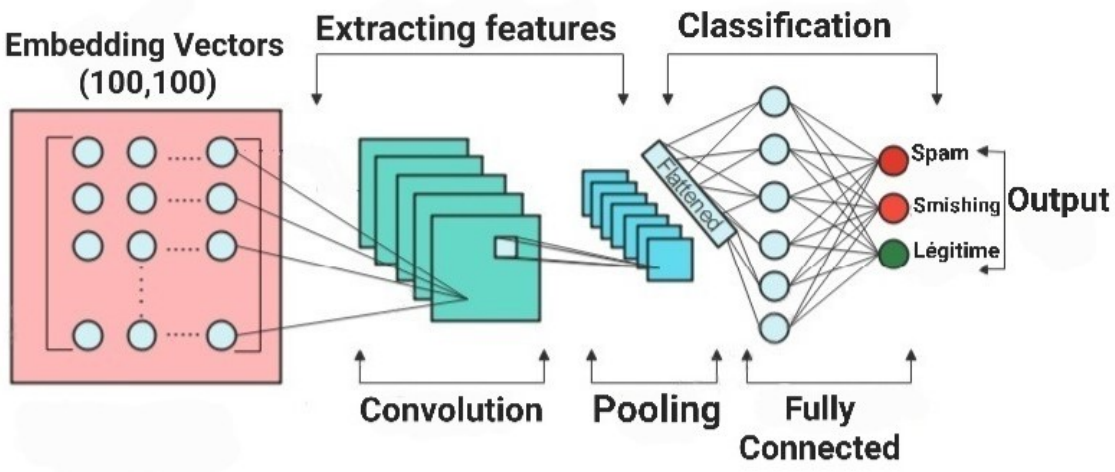


Figure 3.13: Un modèle de réseau neuronal convolutif

3.6.6 Perceptron multicouche (MLP)

• **Apprentissage non fédéré** : L'approche MLP est utilisée dans le contexte de l'apprentissage non fédéré pour identifier le smishing en construisant un réseau neuronal . Le modèle MLP comporte une couche cachée contenant 10 unités. La couche d'intégration fournit les données d'entrée à la couche cachée, où chaque unité effectue une somme pondérée des entrées et applique une fonction d'activation. Cela permet au modèle de saisir

les relations non linéaires complexes entre les caractéristiques et de faire des prédictions. La couche cachée est suivie d'une couche de sortie équipée d'une fonction d'activation softmax qui attribue des probabilités à chaque classe. La classe dont la probabilité est la plus élevée est celle pour laquelle le message de smishing est prédit. En utilisant les capacités des réseaux neuronaux profonds et l'adaptabilité des architectures MLP, le modèle MLP dans l'apprentissage non fédéré fournit une méthode fiable pour la détection de l'hameçonnage.

• **Apprentissage fédéré** : L'algorithme MLP est modifié pour le cas décentralisé dans l'apprentissage fédéré. Chaque appareil client entraîne son propre modèle MLP. L'architecture des modèles locaux, qui comprend des couches interconnectées et des fonctions d'activation, est comparable à celle du modèle MLP centralisé.

Grâce à federated averaging algorithm, le serveur central combine les mises à jour du modèle provenant des appareils clients. Un modèle MLP global est produit à la suite de ce processus d'agrégation, qui bénéficie des diverses informations et connaissances des dispositifs clients. En commençant par la couche d'intégration, le modèle MLP est construit à l'aide d'une architecture séquentielle. La couche d'intégration transforme les informations textuelles en une représentation numérique dense. L'aplatissement de la sortie de la couche d'intégration crée une entrée unidimensionnelle pour le modèle MLP. Le MLP se compose d'une couche cachée de 128 unités et d'une fonction d'activation ReLU pour introduire la non-linéarité. La dernière couche du MLP est une couche de sortie softmax qui attribue des probabilités à chaque classe. La classe ayant la probabilité la plus élevée est celle pour laquelle le message de smishing est prédit. L'apprentissage fédéré facilite la détection des attaques de smishing tout en préservant la confidentialité des données grâce à l'architecture MLP. L'équation de base de MLP est donnée par [53] :

$$f(x) = \text{softmax}(W_2 \cdot \text{relu}(W_1 \cdot x + b_1) + b_2) \tag{3.8}$$

- f(x) : représente la sortie prédite pour l'échantillon d'entrée x.
- x : est l'échantillon d'entrée.
- W_1 : est la matrice de poids entre la couche d'entrée et la première couche cachée.
- b_1 : est le vecteur de biais de la première couche cachée.
- relu() : est la fonction d'activation ReLU (Rectified Linear Unit), qui est appliquée à la sortie de la première couche cachée.
- W_2 : est la matrice de poids entre la première couche cachée et la couche de sortie.
- b_2 : est le vecteur de biais de la couche de sortie.
-softmax() : est la fonction d'activation softmax, qui est appliquée à la sortie de la couche de sortie pour obtenir des probabilités normalisées pour chaque classe.

3.6.7 Support Vector Machine (SVM)

L'algorithme des machines à vecteurs de support (SVM) est une technique d'apprentissage automatique très appréciée pour les problèmes de classification. Nous avons utilisé les SVM comme l'un des modèles de classification dans notre méthode d'apprentissage non fédérée pour former et tester notre dataset.

Pour la détection du smishing dans l'apprentissage non fédéré, la technique de la machine à vecteur de support (SVM) est utilisée. Le modèle SVM est mis en œuvre à l'aide du classificateur LinearSVC. Le SVM est un algorithme puissant qui cherche à identifier l'hyperplan optimal pour classer les données dans un espace de caractéristiques à haute dimension. Dans le contexte de la détection du smishing, le modèle SVM apprend à partir de données étiquetées pour établir une limite de décision qui distingue les messages légitimes des messages de smishing. Le classificateur LinearSVC utilise une fonction de noyau linéaire pour exécuter la tâche de classification. L'apprentissage non fédéré fournit une méthode efficace pour détecter les attaques de smishing sur la base de la séparation apprise entre des classes distinctes en tirant parti de la méthode SVM.

L'équation de base de SVM est donnée par [54] :

$$f(x) = \arg\max_{c} \left(\sum_{i=1}^{N} \alpha[i] \cdot y[i] \cdot K(x[i], x) + b_c \right) \tag{3.9}$$

- f(x) : représente la sortie prédite pour l'échantillon d'entrée x.

- $\arg\max_c$: sélectionner la classe c qui maximise l'expression qui suit. N est le nombre d'échantillons d'entraînement.

- $\alpha[i]$: représente les multiplicateurs de coefficients des vecteurs de support associés à chaque échantillon d'entraînement.

- y[i] représente les étiquettes de classe correspondantes des échantillons d'entraînement. on a trois étiquettes : +1 pour smishing, 0 pour spam et -1 pour légitime.

- K(x[i], x) est la fonction de noyau qui calcule la similarité entre l'échantillon d'entraînement x[i] et l'échantillon d'entrée x.

- b_c représente le terme de biais pour chaque classe.

3.6.8 Decision Tree

L'algorithme de l'arbre de décision est utilisé pour la détection du smishing dans le cadre de l'apprentissage non fédéré. Le modèle d'arbre de décision est mis en œuvre à l'aide de l'algorithme DecisionTreeClassifier. L'arbre de décision est un algorithme bien connu d'apprentissage automatique qui construit un modèle arborescent pour prendre

des décisions basées sur les valeurs des caractéristiques. Dans la détection du smishing, le modèle d'arbre de décision utilise des données étiquetées pour construire une série de conditions "si" et "alors" qui permettent de classer efficacement les messages comme légitimes, spam ou smishing. Chaque nœud interne représente une caractéristique ou un attribut, tandis que chaque nœud secondaire représente un identifiant de classe. Le modèle peut classer efficacement les messages de smishing en parcourant l'arbre de décision en fonction des valeurs des différentes caractéristiques. Dans le cadre de l'apprentissage non fédéré, l'algorithme DecisionTreeClassifier fournit une méthode interprétable et efficace pour la détection du smishing.

3.6.9 Random Forest

Random Forest est une technique d'apprentissage d'ensemble puissante qui mélange de nombreux arbres de décision afin d'améliorer la détection de smishing dans l'apprentissage non fédéré. Le modèle Random Forest est construit avec la procédure d'ensemble RandomForestClassifier. Random Forest est un algorithme d'apprentissage automatique puissant qui combine plusieurs arbres de décision pour augmenter la précision et la robustesse des prédictions. Chaque arbre est formé sur un sous-ensemble aléatoire des données avec remplacement. Le modèle de forêt aléatoire fournit une classification plus précise des messages de smishing en agrégeant les prédictions de tous les arbres. Le modèle de forêt aléatoire de notre étude a une profondeur maximale de 20 et nécessite un minimum d'un échantillon à chaque nœud de feuille. Ces paramètres déterminent la profondeur d'échantillonnage requise et le nombre minimum d'échantillons pour que le modèle puisse faire des prédictions précises.

3.6.10 AdaBoost

Adaboost est une stratégie d'apprentissage automatique permettant de catégoriser efficacement les textes de smishing, qui combine de nombreux apprenants faibles pour générer un apprenant fort. Dans cette méthode, le classificateur DecisionTreeClassifier est utilisé comme estimateur principal. AdaBoost entraîne les classificateurs faibles, tels que les arbres de décision, de manière itérative en ajustant les poids des échantillons mal classés après chaque itération. Dans notre étude, le paramètre n_estimateurs a été fixé à 50, ce qui détermine le nombre de classificateurs faibles utilisés. AdaBoost produit un modèle de classification robuste pour la détection du smishing en intégrant les prédictions de ces classificateurs faibles à l'aide d'un système de vote pondéré. AdaBoost en apprentissage non fédéré améliore la performance de la détection du smishing en utilisant les forces de

plusieurs classificateurs faibles pour augmenter la précision et la robustesse du modèle.

L'équation de base de Adaboost est donnée par [55] :

$$f(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) \tag{10}$$

- f(x) : représente la sortie prédite pour l'échantillon d'entrée x.

- sign() : est la fonction de signe qui attribue la classe -1 pour les valeurs négatives et +1 pour les valeurs positives.

- T : est le nombre d'itérations utilisés dans Adaboost.

- α_t est le poids associé au classifieur faible h_t à l'itération t.

- $h_t(x)$ est la prédiction du classifieur faible h_t pour l'échantillon d'entrée x.

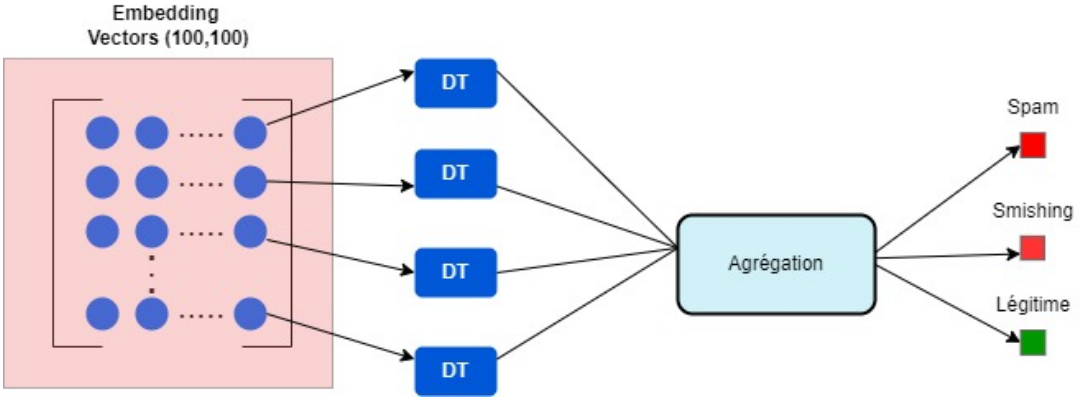


Figure 3.14: L'algorithme AdaBoost proposé

3.7 La classification

La classification des messages SMS dans l'approche d'apprentissage non fédéré implique la formation d'un modèle sur un serveur centralisé à l'aide de l'ensemble de données disponible. Le modèle est formé sur le serveur à l'aide de diverses méthodes, notamment LSMT, Bi-LSTM, CNN, MLP, SVM, adaboost, Decision Tree et Random forest. Le serveur a accès à l'ensemble des données. L'ensemble de données est utilisé par le serveur pour ajuster les performances du modèle et optimiser les paramètres du modèle pendant la phase d'entraînement. Une fois entraîné, le modèle peut être utilisé pour classer les messages SMS sur les appareils des consommateurs.

Le serveur centralisé doit avoir accès aux données de l'utilisateur, et cette solution pose donc des questions concernant la confidentialité des données. Le serveur doit collecter et stocker les messages SMS, qui peuvent contenir des données sensibles. Les risques liés aux

violations de données ou à l'accès non autorisé à des informations personnelles découlent de ce stockage centralisé des données de l'utilisateur. Les utilisateurs peuvent être réticents à fournir leurs données SMS à un serveur central pour des raisons de confidentialité.

L'apprentissage fédéré offre une solution à ces problèmes qui préserve la vie privée. L'apprentissage fédéré permet de répartir l'apprentissage du modèle entre les appareils clients. Les appareils clients effectuent un apprentissage local en utilisant les données facilement disponibles localement plutôt que d'envoyer les données brutes au serveur. Le serveur agrège ensuite les paramètres du modèle d'apprentissage afin de mettre à jour le modèle global. Seules les modifications du modèle sont partagées, les données brutes restant sur chaque appareil client. L'apprentissage fédéré réduit les risques d'atteinte à la vie privée et améliore la confidentialité des données en les gardant décentralisées.

En conclusion, les techniques d'apprentissage non fédéré peuvent catégoriser les messages SMS avec un haut degré de précision, mais elles posent des problèmes de confiance des utilisateurs et de confidentialité des données. En répartissant le processus de formation entre les appareils clients, l'apprentissage fédéré garantit que les données brutes restent sur les appareils clients et que seules les mises à jour du modèle sont partagées, protégeant ainsi la vie privée de l'utilisateur. Cette technique est un choix intéressant pour la classification des SMS, car elle permet de maintenir des niveaux de précision élevés et de résoudre les problèmes de confidentialité.

3.8 Conclusion

Dans ce chapitre, nous avons abordé l'architecture du modèle que nous proposons pour la détection du smishing, nous avons présenté aussi une analyse détaillée des algorithmes employés dans les techniques d'apprentissage non fédéré et d'apprentissage fédéré . Les résultats de la solution que nous proposons seront présentés dans le chapitre suivant.

Chapitre 4

Test et résultats

4.1 Introduction

Les utilisateurs sont plus vulnérables aux attaques de smishing en raison de l'utilisation généralisée des appareils mobiles. Dans ces attaques, les utilisateurs sont incités à divulguer des informations personnelles ou à cliquer sur des liens dangereux par le biais de messages textuels. Dans ce chapitre, nous allons présenter les algorithmes d'apprentissage fédéré que nous proposons pour la détection du smishing et évalue leur efficacité à l'aide de dataset réels, ainsi que l'utilisation finale de ces algorithmes, qui offre une méthode proactive pour identifier et éviter les attaques de smishing.

4.2 Environnement et Outils de Travail

Nous définirons dans cette section les outils logiciels et l'environnement matériel utilisés dans la création et la mise en œuvre du système de détection de l'hameçonnage proposé :

• **Google Colaboratory** :Google Colaboratory, parfois connu sous le nom de Colab, est un service gratuit qui associe la fonctionnalité Jupyter Notebook à des machines virtuelles hébergées par Google et à une technologie de pointe [56]. Colab a été créé pour les chercheurs en IA et en science des données afin d'échanger des expériences reproductibles et des explications techniques.

En utilisant les 12 Go de mémoire vive de Google Colab, nous avons maximisé son potentiel. De plus, en utilisant l'accélération GPU offerte par Google Colab, nous avons pu exécuter des algorithmes d'apprentissage automatique plus rapidement et plus efficacement, ce qui nous a permis d'accélérer notre travail de recherche et d'analyse.

• **Python** : Python est un langage de programmation général de haut niveau dont la

philosophie de conception met fortement l'accent sur la lisibilité du code.

La syntaxe de Python permet aux programmeurs d'exprimer des concepts en moins de code qu'ils ne pourraient le faire dans des langages comme le C, et le langage possède des constructions destinées à permettre la compréhension des systèmes à petite et à grande échelle. Python prend en charge une variété de paradigmes de programmation, y compris la programmation impérative, fonctionnelle et orientée objet [57] .

Le processus d'apprentissage est effectué à l'aide d'un ensemble de bibliothèques Python telles que :

• **TensorFlow** :TensorFlow est une boîte à outils logicielle évolutive et adaptable pour les calculs numériques basés sur les graphes de flux de données. Les utilisateurs peuvent rapidement développer, tester et déployer des réseaux neuronaux et d'autres modèles d'apprentissage automatique à l'aide de cette bibliothèque et des outils associés [58].

•**Keras** :Une boîte à outils d'apprentissage profond de haut niveau en Python appelée Keras, qui peut être utilisée au-dessus de TensorFlow , est petite et simple à apprendre. Il permet aux développeurs de s'occuper des subtilités des tenseurs, de leurs formes et de leurs détails mathématiques tout en se concentrant sur les idées clés de l'apprentissage profond, telles que la construction de couches pour les réseaux neuronaux. Le back-end de Keras doit être TensorFlow, Theano ou CNTK [59].

Dans notre travail, nous avons mis en pratique les modèles de classification proposés en utilisant un ordinateur aux caractéristiques suivantes :

• **RAM** : 8,00 Go.
• **Processeur** : Intel(R) Core(TM) i7-4600U CPU @ 2.10GHz 2.70 GHz.
• **Système** : Système d'exploitation 64 bits, processeur x64.

4.3 Mésures de comparaison

Différentes mesures de comparaison ont été utilisées pour évaluer les performances des modèles proposés, tels que : Accuracy, la Précision, Recall, F1-Score, la matrice de confusion et AUC.

4.3.1 La matrice de confusion

La matrice de confusion est une méthode d'évaluation des performances, permet de visualiser et de répertorier les résultats d'un modèle de classification. Le nombre de vrais

positifs, de vrais négatifs, de faux positifs et de faux négatifs est indiqué.

•**Vrais positifs (VP)** : Les vrais positifs (TP) sont le nombre de messages de smishing que notre modèle de classification a réussi à détecter comme étant du smishing.

•**Vrais négatifs (VN)** : Les vrais négatifs (TN) sont un terme utilisé pour décrire le nombre de cas qui ont été correctement identifiés comme négatifs et pour lesquels la classe réelle et la classe projetée étaient toutes deux négatives.

• **Faux positifs (FP)** : Les faux positifs (FP) sont des instances que le modèle prédit comme positives (smishing), mais qui sont en fait négatives (légitimes).

• **Faux négatifs (FN)** : Les faux négatifs (FN) sont des cas où un échantillon est positif alors que le modèle indiquait qu'il serait négatif. Dans notre cas, il s'agit de messages de smishing qui n'étaient pas classés comme tels et qui auraient pu exposer des informations sensibles.

		Classe Réelle	
		Négatif	**Positif**
Classe Prédite	**Négatif**	**Vrais Négatifs**	**Faux Négatifs**
	Positif	**Faux positifs**	**Vrais Positifs**

Figure 4.1: La matrice de confusion [60]

4.3.2 La précision

La précision est une statistique de performance qui évalue le rapport entre les prédictions positives exactes et l'ensemble des prédictions positives d'un modèle. Elle démontre la précision du modèle dans l'identification des échantillons positifs [61].

$$\textbf{PRECISION} = \frac{VP}{VP + FP} \tag{4.1}$$

4.3.3 Exactitude

Une mesure de la capacité de modèle à catégoriser chaque point de données dans un ensemble de données. Elle est calculée en divisant le nombre total de points de données par le nombre de points de données correctement catégorisés [62].

$$\textbf{Exactitude} = \frac{VP + VN}{VP + FP + VN + FN} \tag{4.2}$$

4.3.4 Recall

Il s'agit de la proportion de vrais positifs par rapport au total des vrais positifs et des faux négatifs.Il évalue le pourcentage de cas positifs réels que le modèle a détectés avec précision [61].

$$\textbf{RAPPEL} = \frac{VP}{VP + FN} \tag{4.3}$$

4.3.5 F1-Score

Le score F1 est la moyenne harmonique de la précision et du rappel, où le rappel est la proportion de vrais positifs parmi tous les positifs réels et la précision est la proportion de vrais positifs parmi tous les positifs anticipés [62].

$$\textbf{F1-Score} = \frac{2 \times Precision \times Rappel}{Precision + Rappel} \tag{4.4}$$

4.3.6 AUC

L'aire sous la courbe de la caractéristique de fonctionnement du récepteur (ROC), qui représente le taux de vrais positifs (sensibilité) par rapport au taux de faux positifs (1-spécificité), est utilisée comme statistique pour évaluer l'efficacité d'un modèle de classification binaire. L'aire sous la courbe a une valeur comprise entre 0 et 1, les chiffres les plus élevés indiquant une meilleure performance [62].

4.4 Expérimentation

Cette partie couvre les résultats expérimentaux et l'évaluation de l'architecture de détection de smishing présentée dans le chapitre précédent.

Pour mesurer la performance globale de chaque modèle, nous avons calculé des paramètres d'évaluation tels que la précision, le rappel et le score F1. Ces mesures donnent des informations sur la capacité des algorithmes à classer correctement les messages de smishing.

4.4.1 Apprentissage non fédéré

Nous examinerons dans cette section les résultats des algorithmes d'apprentissage non fédérés utilisés dans notre étude.

Dans le cadre de notre étude, nous avons évalué plusieurs modèles d'apprentissage profond et d'apprentissage automatique pour détecter le smishing dans notre dataset, tel que : LSTM, Bi-LSTM, CNN, SVM, Decision Tree, Random Forest, MLP et AdaBoost.

•**Résultats des algorithmes d'apprentissage profond** :

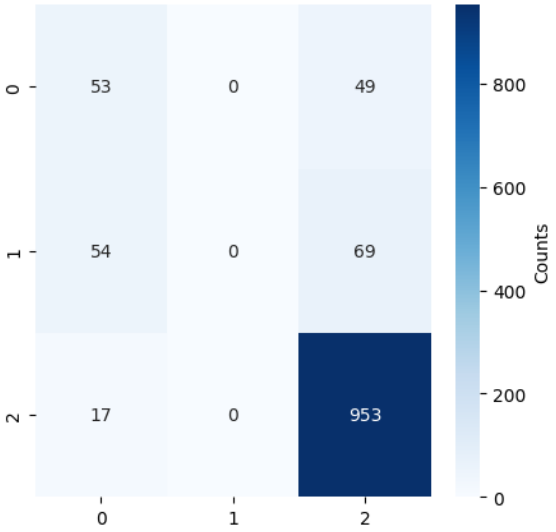
Pour l'apprentissage non fédéré, nous avons évalué l'efficacité de diverses techniques d'apprentissage profond pour la détection du smishing. Le modèle Bi-LSTM a surpassé les autres modèles avec une valeur moyenne d'exactitude de 92,3 %. Nous avons ensuite utilisé le modèle LSTM, qui a atteint une valeur d'exactitude de 87,86%. Le modèle CNN a également été utilisé, avec une valeur moyenne d'exactitude de 91.54 %. En outre, le modèle MLP a été mis en œuvre, ce qui a permis d'obtenir une valeur d'exactitude de 80.5 %.

Le tableau 4.1 présente les mésures d'évaluations des algorithmes d'apprentissage profond dans le cas d'apprentissage non fédéré :

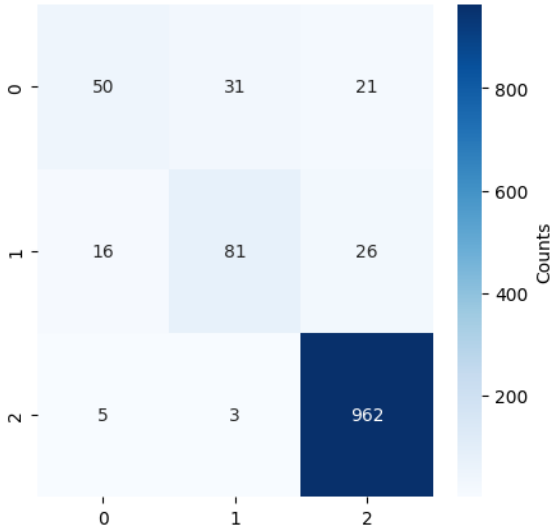
Tableau 4.1: *Mesures d'évaluation des des algorithmes d'apprentissage profond dans le cas d'apprentissage non fédérés*

Algorithme	Exactitude	La précision	Recall	F1-Score
LSTM	87.86%	81%	88%	84%
Bi-LSTM	**92.3%**	**92%**	**91%**	**91%**
CNN	91.54%	91%	92%	91%
MLP	80.5%	69%	81%	74%

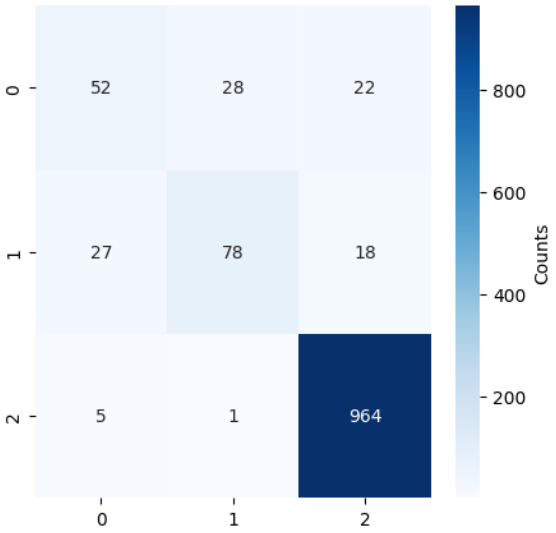
La figure 4.2 représente les matrices de confusion de chaque algorithmes :



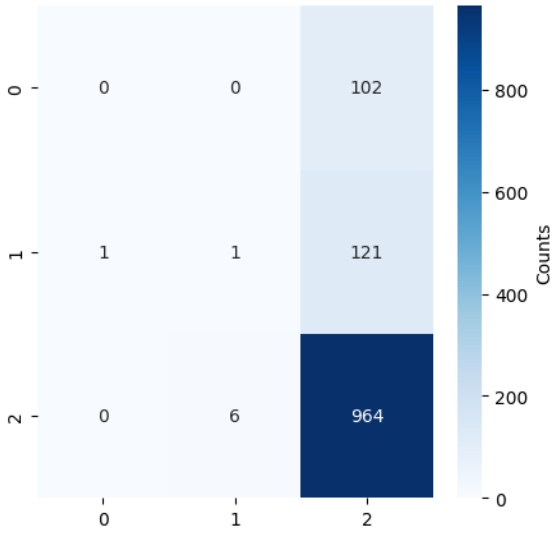
(a) La matrice de confusion du modèle LSTM



(b) La matrice de confusion du modèle Bi-LSTM



(c) La matrice de confusion du modèle CNN



(d) La matrice de confusion du modèle MLP

Figure 4.2: Les matrices de confusions des modèles LSTM, Bi-LSTM, CNN et MLP

•**Résultats des algorithmes d'apprentissage automatique** :

Nous avons évalué l'efficacité de divers algorithmes d'apprentissage automatique pour la détection du smishing dans le scénario d'apprentissage non fédéré.

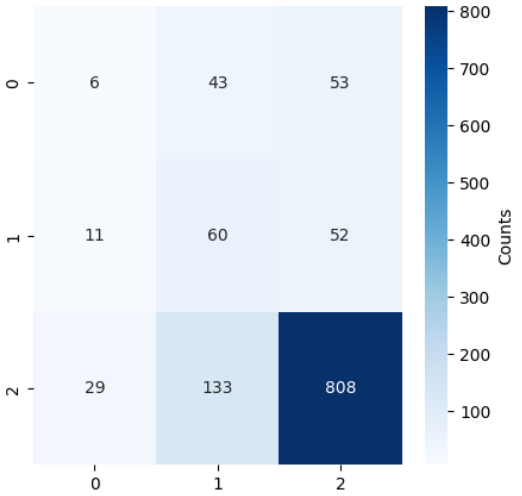
La valeur moyenne d'exactitude du modèle SVM était de 73.13 %. De plus, le modèle Decision Tree a été utilisé, avec valeur moyenne d'exactitude de 80.08 %. Le modèle Random Forest a quant à lui obtenu une valeur d'exactitude de 88.95 %. Le modèle AdaBoost a obtenu une valeur moyenne d'exactitude de 86.61 %.

Le tableau 4.2 présente les mésures d'évaluations des algorithmes d'apprentissage automatique dans le cas d'apprentissage non fédéré :

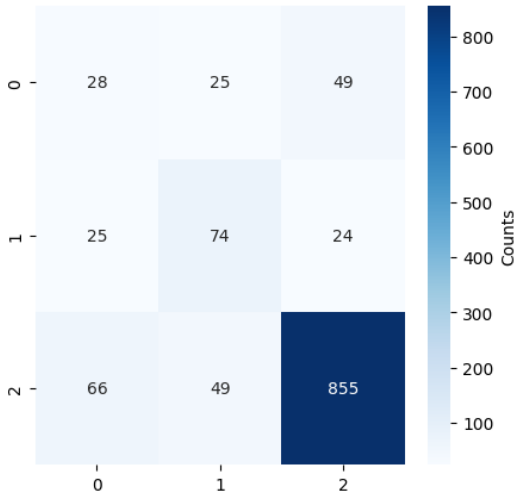
Tableau 4.2: *Mesures d'évaluation des algorithmes d'apprentissage automatique dans le cas d'apprentissage non fédérés*

Algorithme	Exactitde	La précision	Recall	F1-Score
SVM	73.13%	76%	73%	74%
Decision Tree	80.08%	82%	80%	80 %
Random Forest	**88.95%**	**87%**	**89%**	**87%**
AdaBoost	86.61%	85%	87%	86%

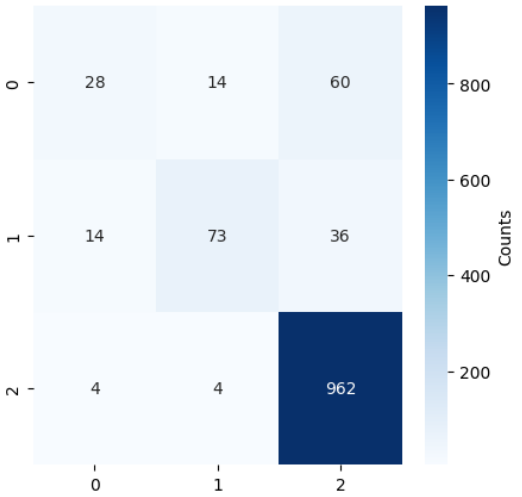
La figure 4.3 représente les matrices de confusion de chaque algorithmes :



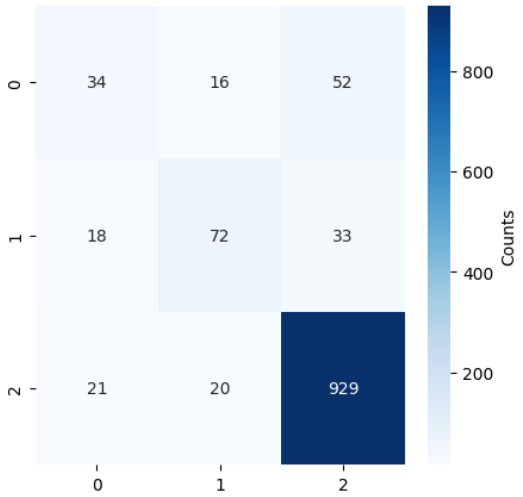
(a) La matrice de confusion du modèle SVM



(b) La matrice de confusion du modèle Decision Tree



(c) La matrice de confusion du modèle Random Forest



(d) La matrice de confusion du modèle AdaBoost

Figure 4.3: Les matrices de confusions des modèles SVM, Decision Tree, Random Forest et AdaBoost

4.4.2 Apprentissage fédéré

Nous examinerons dans cette section les résultats des algorithmes d'apprentissage fédérés utilisés dans notre étude.

Nous avons évalué les mêmes modèles d'apprentissage profond utilisés dans l'apprentissage non fédéré pour détecter le smishing dans notre dataset : LSTM, Bi-LSTM, CNN, MLP, qui comprenait un ensemble de données provenant de 10 clients différents.

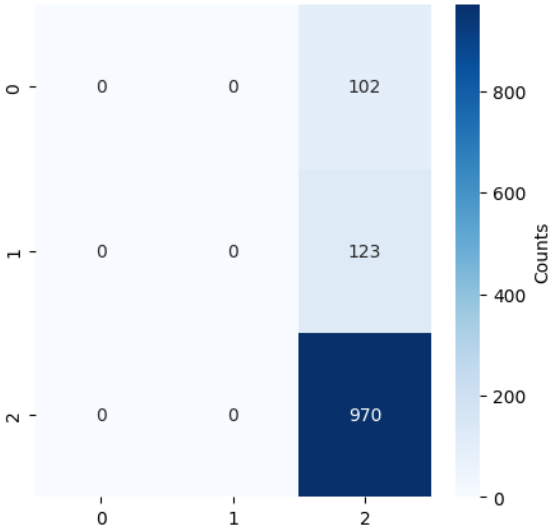
Parmi ces modèles, l'approche CNN a obtenu la valeur moyenne d'exactitude de détection des attaques de smishing le plus élevé, 92,38 %, ce qui démontre son efficacité. Les autres modèles ont obtenu des moyennes valeurs d'accuracy de 81,17 % pour LSTM, 88,78 % pour Bi-LSTM et 88,87 % pour MLP.

Le tableau 4.3 représente les mésures d'évaluations des modèles d'apprentissage fédérés :

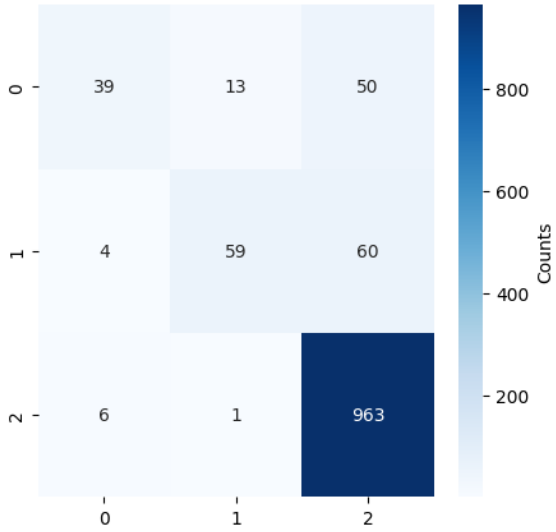
Tableau 4.3: *Mesures d'évaluation des modèles d'apprentissage fédérés*

Algorithme	Exactitude	La précision	Recall	F1-Score
LSTM	81.17%	66%	81%	73%
Bi-LSTM	88.78%	88%	89%	87%
CNN	**92.38%**	**92%**	**92%**	**92%**
MLP	88.87%	88 %	89%	87%

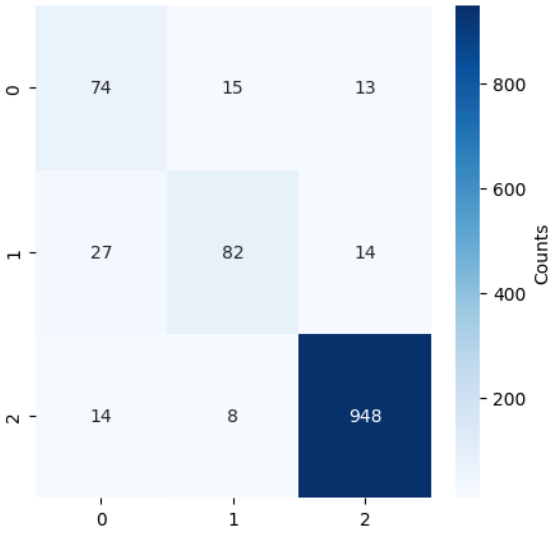
La figure 4.4 représente les matrices de confusion de chaque algorithmes :



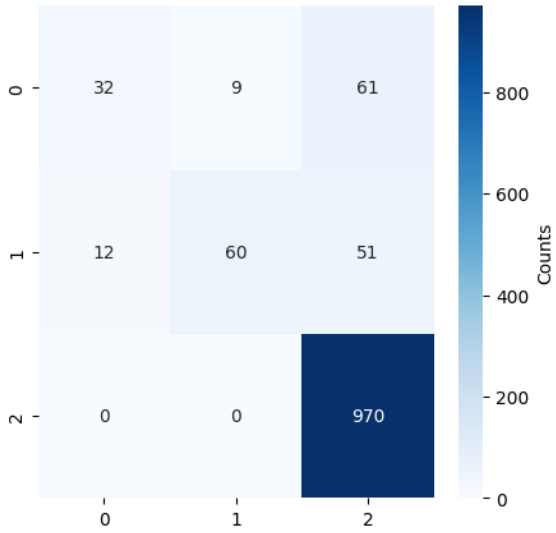
(a) La matrice de confusion du modèle LSTM



(b) La matrice de confusion du modèle Bi-LSTM



(c) La matrice de confusion du modèle CNN



(d) La matrice de confusion du modèle MLP

Figure 4.4: Les matrices de confusions des modèles LSTM, Bi-LSTM, CNN et MLP

4.5 Synthèse

Notre travail consiste à évaluer les modèles de détection du smishing en utilisant des approches d'apprentissage fédéré et non fédéré. En comparant les résultats, nous avons constaté que l'approche d'apprentissage fédéré utilisant le CNN avait la plus grande exactitude 92,38 %, alors que l'approche d'apprentissage non fédéré utilisant le Bi-LSTM présentait la moins bonne exactitude 92,3 %. Cela démontre l'efficacité des techniques d'apprentissage fédéré et non fédéré pour obtenir une grande précision dans la détection du smishing.

Le tableau 4.4 montre Une comparaisons entre les résultats de la méthode proposée et les travaux de la littérature :

Tableau 4.4: *Une comparaisons entre les résultats de la méthode proposée et plusieurs techniques de classification*

Recherche	Algorithme	La précision	Exactitude	Recall	F1-Score	AUC
[40]	Backpropagation Algorithm	84%	97.93%	94%	/	98.8%
[63]	Naive Bayes	93%	91.6%	92%	92%	/
[47]	ANN	/	94%	/	86.72%	/
Notre Proposition	**CNN**	**92%**	**92.38%**	**92%**	**92%**	/

En comparant avec les travaux de la littérature utilisant la même dataset et l'algorithme de Backpropagation, où la meilleure valeur d'exactitude obtenue était de 97,93 %, nos modèles d'apprentissage fédéré ont montré des précisions un peu plus faibles. Cependant, il est important de noter que les travaux de la littérature ont utilisé une approche centralisée avec un accès direct au dataset, alors que nos modèles d'apprentissage fédéré ont fonctionné dans un cadre décentralisé avec un accès limité aux données individuelles des clients.

Le principal avantage de l'apprentissage fédéré est la garantie de la sécurité et de la confidentialité des données. Chaque appareil client dans notre configuration d'apprentissage fédéré a entraîné son propre modèle local avec ses propres données privées, et seules les mises à jour du modèle ont été partagées avec le serveur central pour l'agrégation. Cette approche décentralisée garantit que les données de l'utilisateur restent sécurisées et privées, atténuant ainsi les problèmes de confidentialité associés au stockage central des données.

Bien que nos modèles d'apprentissage fédéré aient obtenu une précision légèrement inférieure à ceux de la littérature, ils présentent un certain nombre d'avantages. L'ap-

prentissage fédéré permet d'utiliser divers ensembles de données provenant de différents clients, augmentant ainsi les capacités de généralisation des modèles. En outre, l'apprentissage fédéré est bien adapté aux situations où les données ne peuvent pas être facilement centralisées, comme dans le cas des données sensibles des utilisateurs ou des systèmes distribués.

Notre recherche démontre le potentiel de l'apprentissage fédéré dans le contexte de la détection de l'hameçonnage. Même si nos modèles ne surpassent pas la précision des approches centralisées, les avantages de la préservation de la confidentialité et de la sécurité des données font de l'apprentissage fédéré une technique précieuse dans les applications du monde réel où la confidentialité est de la plus haute importance.

4.6 Conclusion

Dans ce chapitre, nous avons examiné l'environnement de travail et les outils utilisés dans le cadre de notre recherche sur la détection du smishing. En outre, nous avons comparé les méthodes de mesure utilisées pour évaluer nos modèles. Ensuite, nous avons présenté les résultats de nos approches d'apprentissage non fédéré et fédéré, démontrant la moyenne valeurs d'accuracy des différents modèles. Enfin, nous avons fourni un résumé de nos conclusions, y compris une comparaison entre l'apprentissage non fédéré et l'apprentissage fédéré, ainsi qu'une comparaison entre l'apprentissage fédéré avec les travaux de la littérature.

Conclusion Générale

Cette recherche a contribué de manière significative au domaine de la détection du smishing en tirant parti du concept d'apprentissage fédéré. Notre travail a consisté en une étude exhaustive de divers modèles d'apprentissage automatique et d'apprentissage profond, en mettant l'accent sur la mise en œuvre de l'apprentissage fédéré pour améliorer la confidentialité.

Le modèle CNN avec apprentissage fédéré a atteint une valeur moyenne d'exactitude de 92,38 %, ce qui constitue l'un des résultats les plus significatifs de notre étude. Bien que ce résultat soit supérieur aux valeurs d'accuracy obtenues par d'autres modèles d'apprentissage non fédéré, il est à préciser qu'il n'a pas dépassé la valeur d'exactitude rapportée dans les travaux de la littérature .

Même si nous n'avons pas atteint la précision la plus élevée rapportée dans la littérature, notre objectif principal était d'assurer la sécurité et la confidentialité des données des utilisateurs. L'apprentissage fédéré nous a permis d'entraîner le modèle sans centraliser l'acquisition des données, protégeant ainsi les informations sensibles et préservant la vie privée des utilisateurs.

L'utilisation de l'apprentissage fédéré dans le contexte de la détection du smishing est une nouvelle approche qui n'a pas fait l'objet d'études approfondies auparavant. Cette étude comble ce vide et démontre l'immense potentiel de l'apprentissage fédéré pour améliorer le cadre de sécurité des communications mobiles.

En formant le modèle localement sur les appareils des utilisateurs, l'apprentissage fédéré garantit la confidentialité et la sécurité des données. Cette stratégie distribuée permet non seulement de protéger les données sensibles, mais aussi de réduire la dépendance à l'égard des serveurs centralisés, ce qui se traduit par un système plus évolutif et plus efficace. En outre, la réduction de la charge des serveurs se traduit par une diminution de la consommation d'énergie, ce qui fait de l'apprentissage fédéré une solution de protection de l'environnement.

Il existe de nombreuses possibilités de recherche prospective pour l'avenir. L'accuracy et la précision des modèles de détection du smishing peuvent être améliorées en intégrant des algorithmes d'apprentissage automatique supplémentaires dans l'application de l'apprentissage fédéré. En outre, le développement d'une application mobile intégrant notre modèle entraîné offrirait aux utilisateurs une protection en temps réel contre les attaques de smishing.

En plus, l'évolution continue des attaques de smishing nécessite le développement et l'adaptation continus de nos systèmes de détection. L'entraînement du modèle à la détection des attaques de type "zero-day", qui sont des menaces émergentes sans signatures de détection préalables, serait un domaine de recherche intéressant à l'avenir.

Afin d'accroître l'impact de notre travail, nous avons l'intention de prendre en charge d'autres langues, telles que l'arabe et le français, avec nos modèles. Cela permettrait de prendre en compte un plus grand nombre d'utilisateurs et de répondre à la nature globale des attaques de smishing.

Cette recherche a effectivement démontré l'efficacité de l'apprentissage fédéré dans la détection de l'hameçonnage, offrant une confidentialité, une évolutivité et une sécurité accrues par rapport aux approches traditionnelles non fédérées. L'importance que nous accordons à la sécurité et à la confidentialité des données pose les bases de futures avancées dans la prévention des attaques par smishing, même si nous n'avons pas atteint la précision la plus élevée rapportée dans la littérature.

Bibliographie

[1] A. Kang, J. Dong Lee, W. M. Kang, L. Barolli, and J. H. Park, "Security considerations for smart phone smishing attacks," in *Advances in Computer Science and its Applications : CSA 2013*, pp. 467–473, Springer, 2014.

[2] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[3] A. S. Shitole and I. Priyadarshini, "Survey of machine learning algorithms & its applications," *Journal of Advances in Computational Intelligence Theory*, vol. 3, no. 2, 2021.

[4] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, pp. 381–386, 2020.

[5] B. Liu and B. Liu, *Supervised learning*. Springer, 2011.

[6] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, 2021.

[7] P. Bajaj, "What is reinforcement learning ?." https://www.geeksforgeeks.org/what-is-reinforcement-learning/, Accessed 2023.

[8] R. Warlop, "Petit guide du machine learning – partie 4 : l'apprentissage par renforcement." https://teahouse.fifty-five.com/fr/petit-guide-du-machine-learning-partie-4-lapprentissage-par-renforcement/, 2019.

[9] P. Sarkar, "Boosting and adaboost in machine learning." https://www.knowledgehut.com/blog/data-science/boosting-and-adaboost-in-machine-learning, 2023.

[10] A. W. Trask, *Grokking deep learning*. Simon and Schuster, 2019.

[11] A. D. Jagtap, Y. Shin, K. Kawaguchi, and G. E. Karniadakis, "Deep kronecker neural networks : A general framework for neural networks with adaptive activation functions," *Neurocomputing*, vol. 468, pp. 165–180, 2022.

[12] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks : analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, 2021.

[13] V. H. Phung and E. J. Rhee, "A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets," *Applied Sciences*, vol. 9, no. 21, p. 4500, 2019.

[14] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv :1511.08458*, 2015.

[15] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv :1409.2329*, 2014.

[16] S. Latif, M. Driss, W. Boulila, S. S. Jamal, Z. Idrees, J. Ahmad, *et al.*, "Deep learning for the industrial internet of things (iiot) : A comprehensive survey of techniques, implementation frameworks, potential applications, and future directions," *Sensors*, vol. 21, no. 22, p. 7518, 2021.

[17] E. Zvornicanin, "Differences between bidirectional and unidirectional lstm." https ://www.baeldung.com/cs/bidirectional-vs-unidirectional-lstm, 2022.

[18] S. Abirami and P. Chitra, "Energy-efficient edge based real-time healthcare support system," in *Advances in computers*, vol. 117, pp. 339–368, Elsevier, 2020.

[19] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.

[20] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning : Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[21] T. Gemmo, "Federated learning : Predictive model without data sharing." https://gemmo.ai/federated-learning, 2021.

[22] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning : Challenges, methods, and future directions," *IEEE signal processing magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[23] RapidMiner-report, "What is federated learning ?." https ://rapidminer.com/glossary/federated-learning/, 2021.

[24] Devfi-report, "Federated learning : Benefits, applications implementation tips." https ://www.devfi.com/federated-learning-benefits-applications-implementation-tips/.

[25] C. Dilmegani, "What is federated learning ? use cases benefits in 2023." https ://research.aimultiple.com/federated-learning/, 2022.

[26] Great-Learning-Team, "What is word embedding | word2vec | glove." https ://www.mygreatlearning.com/blog/word-embedding/glove, 2020.

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[28] S. H. Apandi, J. Sallim, and R. M. Sidek, "Types of anti-phishing solutions for phishing attack," in *IOP Conference Series : Materials Science and Engineering*, vol. 769, p. 012072, IOP Publishing, 2020.

[29] N. Klimburg-Witjes and A. Wentland, "Hacking humans ? social engineering and the construction of the "deficient user" in cybersecurity discourses," *Science, Technology, & Human Values*, vol. 46, no. 6, pp. 1316–1339, 2021.

[30] Z. Wang, H. Zhu, P. Liu, and L. Sun, "Social engineering in cybersecurity : a domain ontology and knowledge graph application examples," *Cybersecurity*, vol. 4, pp. 1–21, 2021.

[31] R. Alabdan, "Phishing attacks survey : Types, vectors, and technical approaches," *Future internet*, vol. 12, no. 10, p. 168, 2020.

[32] Kaspersky-report, "What is smishing and how to defend against it." https ://www.kaspersky.com/resource-center/threats/what-is-smishing-and-how-to-defend-against-it, 2021.

[33] The-SMS-structure, "The sms structure." https ://doc.pcsoft.fr/en-US/ ?3068003name=the$_s ms_s tructure$, 2022.

[34] Keeper-report, "Qu'est-ce que le smishing ? un guide sur l'hameçonnage par sms," 2023.

[35] A. K. Jain and B. Gupta, "Rule-based framework for detection of smishing messages in mobile environment," *Procedia Computer Science*, vol. 125, pp. 617–623, 2018.

[36] S. Mishra and D. Soni, "A content-based approach for detecting smishing in mobile environment," in *Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur-India*, 2019.

[37] C. Balim and E. S. Gunal, "Automatic detection of smishing attacks by machine learning methods," in *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, pp. 1–3, IEEE, 2019.

[38] G. Sonowal, "Detecting phishing sms based on multiple correlation algorithms," *SN computer science*, vol. 1, no. 6, p. 361, 2020.

[39] M. Liu, Y. Zhang, B. Liu, Z. Li, H. Duan, and D. Sun, "Detecting and characterizing sms spearphishing attacks," in *Annual Computer Security Applications Conference*, pp. 930–943, 2021.

[40] S. Mishra and D. Soni, "Dsmishsms-a system to detect smishing sms," *Neural Computing and Applications*, pp. 1–18, 2021.

[41] B. E. Boukari, A. Ravi, and M. Msahli, "Machine learning detection for smishing frauds," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–2, IEEE, 2021.

[42] A. K. Jain, B. B. Gupta, K. Kaur, P. Bhutani, W. Alhalabi, and A. Almomani, "A content and url analysis-based efficient approach to detect smishing sms in intelligent systems," *International Journal of Intelligent Systems*, vol. 37, no. 12, pp. 11117–11141, 2022.

[43] I. S. Mambina, J. D. Ndibwile, and K. F. Michael, "Classifying swahili smishing attacks for mobile money users : A machine-learning approach," *IEEE Access*, vol. 10, pp. 83061–83074, 2022.

[44] C. Zhou, C. Sun, Z. Liu, and F. Lau, "A c-lstm neural network for text classification," *arXiv preprint arXiv :1511.08630*, 2015.

[45] P. K. Roy, J. P. Singh, and S. Banerjee, "Deep learning to filter sms spam," *Future Generation Computer Systems*, vol. 102, pp. 524–533, 2020.

[46] S. S. Shravasti and M. Chavan, "Smishing detection : Using artificial intelligence,"

[47] S. Mishra and D. Soni, "Implementation of 'smishing detector' : an efficient model for smishing detection using neural network," *SN Computer Science*, vol. 3, no. 3, p. 189, 2022.

[48] A. K. Jain and B. B. Gupta, "Feature based approach for detection of smishing messages in the mobile environment," *Journal of Information Technology Research (JITR)*, vol. 12, no. 2, pp. 17–35, 2019.

[49] S. Sheikhi, M. T. Kheirabadi, and A. Bazzazi, "An effective model for sms spam detection using content-based features and averaged neural network," *International Journal of Engineering*, vol. 33, no. 2, pp. 221–228, 2020.

[50] D. Goel and A. K. Jain, "Smishing-classifier : a novel framework for detection of smishing attack in mobile environment," in *Smart and Innovative Trends in Next Generation Computing Technologies : Third International Conference, NGCT 2017, Dehradun, India, October 30-31, 2017, Revised Selected Papers, Part II 3*, pp. 502–512, Springer, 2018.

[51] D. S. sandhya mishra, "Sms phishing dataset for machine learning and pattern recognition." https ://data.mendeley.com/datasets/f45bkkt8pr/1, 2022.

[52] K. Smagulova and A. P. James, "A survey on lstm memristive neural network architectures and applications," *The European Physical Journal Special Topics*, vol. 228, no. 10, pp. 2313–2324, 2019.

[53] H. Ramchoun, Y. Ghanou, M. Ettaouil, and M. A. Janati Idrissi, "Multilayer perceptron : Architecture optimization and training," 2016.

[54] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions : a local svm approach," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 3, pp. 32–36, IEEE, 2004.

[55] R. E. Schapire, "Explaining adaboost," *Empirical Inference : Festschrift in Honor of Vladimir N. Vapnik*, pp. 37–52, 2013.

[56] M. J. Nelson and A. K. Hoover, "Notes on using google colaboratory in ai education," in *Proceedings of the 2020 ACM conference on innovation and Technology in Computer Science Education*, pp. 533–534, 2020.

[57] G. Van Rossum *et al.*, "Python programming language.," in *USENIX annual technical conference*, vol. 41, pp. 1–36, Santa Clara, CA, 2007.

[58] B. Pang, E. Nijkamp, and Y. N. Wu, "Deep learning with tensorflow : A review," *Journal of Educational and Behavioral Statistics*, vol. 45, no. 2, pp. 227–248, 2020.

[59] N. K. Manaswi and N. K. Manaswi, "Understanding and working with keras," *Deep learning with applications using Python : Chatbots and face, object, and speech recognition with TensorFlow and Keras*, pp. 31–43, 2018.

[60] A. Bhandari, "Understanding interpreting confusion matrices for machine learning." https ://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/, 2023.

[61] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240, 2006.

[62] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, f-score and roc : a family of discriminant measures for performance evaluation," in *AI 2006 : Advances in Artificial Intelligence : 19th Australian Joint Conference on Artificial Intelligence, Hobart, Australia, December 4-8, 2006. Proceedings 19*, pp. 1015–1021, Springer, 2006.

[63] S. Mishra and D. Soni, "Smishing detector : A security model to detect smishing through sms content analysis and url behavior analysis," *Future Generation Computer Systems*, vol. 108, pp. 803–815, 2020.