UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

ANALYSE HIERARCHIQUE ET MULTIMODALE EN UTILISANT L'APPRENTISSAGE PROFOND POUR LA DETECTION DE SPAMS

MÉMOIRE PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DU PROGRAMME DE MAÎTRISE EN SCIENCES ET TECHNOLOGIES DE

L'INFORMATION

PAR

Mahdi Tari

Juin 2024

Jury d’évaluation

Président du Jury :	Pr. Nguena Timo, Omer Landry
Membre du Jury :	Pr. Raphael Khoury
Directeur de recherche :	Pr. Allili Mohand Said

Mémoire accepté le : 24 Juin 2024

Remerciements

Cette réussite remarquable n'aurait pu se concrétiser sans la précieuse collaboration de nombreux individus, et c'est empreint d'une gratitude profonde que je souhaite exprimer mes remerciements.

Je souhaite témoigner ma profonde gratitude envers le Professeur Mohand Saïd Allili, mon directeur de recherche. Son encadrement avisé a été un facteur déterminant dans la réussite exceptionnelle de ce projet de maîtrise. Sa disponibilité, ses conseils judicieux et son soutien financier ont constitué des piliers essentiels tout au long de cette aventure académique.

Un immense merci est également destiné à l'ensemble des professeurs de l'UQO, dont les enseignements ont enrichi mon parcours académique.

À mes parents, piliers inébranlables de sagesse, patience et conseils éclairés, je dédie une gratitude infinie. Les encouragements constants et le soutien inconditionnel que j'ai reçus ont illuminé ma trajectoire, me permettant de franchir des caps significatifs. À ma mère, source inépuisable de persévérance, sagesse et bienveillance, je rends hommage pour m'avoir inculqué ces valeurs avec dévouement. À mon père, modèle de détermination, rigueur et ardeur au travail, je témoigne ma reconnaissance pour les éclairages essentiels qu'il a apportés à mon parcours. Grâce à eux, j'ai pu atteindre des jalons importants, et c'est avec une sincérité profonde que je leur exprime ma reconnaissance.

Ce travail de recherche n'aurait pu voir le jour sans le soutien indéfectible de ma chère épouse, Yamina. Sa présence constante et son appui moral ont été des piliers essentiels dans les moments les plus difficiles. À toi, Yamina, va ma reconnaissance profonde pour avoir rendu possible la concrétisation de ce projet.

Enfin, un merci chaleureux s'adresse à ma famille, qui a su trouver les mots justes pour m'encourager tout au long de ce parcours. Votre soutien inestimable demeure gravé dans mon cœur, et je vous exprime ma reconnaissance la plus sincère.

À mes précieux parents, Megdouda et AbdelKamel,

À l'amour de ma vie, ma femme Yamina.

Table des matières

Liste des figures

Liste des tableaux

Liste des abréviations, sigles et acronymes

TALN	traitement automatique du langage naturel
Glove	vecteurs globaux pour la représentation des mots
ELMO	incorporations à partir de modèles de langage
LLM	grands Modèles de Langage
USE	universel sentence encoder
Bert	représentations d'encodeur bidirectionnelles de transformers
XLNET	réseau neuronal extra-long
GPT	transformer générateur pré-entraîné
TF-IDF	fréquence des termes-inverse de la fréquence dans les documents
RL	régression logistique
RF	random forest
SVM	support vector machine
NB	naïve bayes
DT	arbres de décision
CNN	réseau neuronal convolutif
RNN	réseau neuronal récurrent
LSTM	mémoire à long terme courte
VGSL	apprentissage de la forme à géométrie variable
URL	localisateur uniforme de ressource
PCA	analyse en composantes principales

Résumé

Avec la croissance de l'utilisation de la messagerie électronique et des réseaux sociaux, le spam est devenu un défi majeur. Avec l'essor des technologies multimédias, la prévalence du spam multimodal contenant un mélange de texte et d'images a significativement augmenté. Or, la plupart des méthodes proposées pour détecter le spam dans le passé sont principalement basées sur l'analyse du texte. Le développement d'une approche multimodale de filtrage de spams revêt donc d'une importance capitale. Dans cette perspective, le mémoire vise à développer une méthode améliorée pour la détection de spams multimédias en utilisant l'analyse hiérarchique et multimodale des messages, combinée à l'apprentissage profond.

Notre approche repose sur l'extraction de plusieurs caractéristiques représentatives à partir de données multimodales (texte, liens, images) en utilisant des modèles basés sur les grands modèles de langage et les réseaux convolutifs. Ceci vise à obtenir une représentation sémantique fine mettant l'accent sur les parties cruciales des messages pour une meilleure classification de spams multimédias. Nous avons évalué notre approche sur un grand ensemble de données, en utilisant une analyse qualitative et quantitative pour mesurer la précision et la robustesse. Notre approche a démontré un grand potentiel pour combiner efficacement plusieurs modalités de données dans l'analyse des spams.

Abstract

With the growth of email and social media usage, spam has become a major challenge. With the rise of multimedia technologies, the prevalence of multimodal spam containing a mix of text and images has significantly increased. However, most of the methods proposed in the past for spam detection are primarily based on text analysis. Therefore, the development of a multimodal spam filtering approach is of paramount importance. In this perspective, this thesis aims to develop an improved method for multimedia spam detection using hierarchical and multimodal message analysis combined with deep learning.

Our approach relies on extracting multiple representative features from multimodal data (text, links, images) using models based on large language models and convolutional networks. This aims to obtain a fine semantic representation, focusing on the crucial parts of the messages for better multimedia spam classification. We evaluated our approach on a large dataset, using both qualitative and quantitative analysis to measure accuracy and robustness. Our approach has demonstrated great potential for effectively combining multiple data modalities in spam analysis.

.

1. INTRODUCTION GENERALE

1.1 Introduction

Le spam est un problème persistant sur Internet depuis les années 1990, caractérisé par des évolutions majeures au fil du temps. Outre le spam traditionnel, les spammeurs ont élaboré des techniques sophistiquées pour contourner les filtres anti-spam et accroître leur taux de succès. Par exemple, la pandémie de COVID-19 a entraîné une augmentation significative des spams liés au COVID-19. Les spams de ransomware ont également augmenté de 20 % en 2020 par rapport à l'année précédente, selon une étude de Kaspersky Lab [2]. Les spams de sextorsion représentent une autre forme de spam qui a connu une croissance notable ces dernières années, augmentant de 36 % en 2020 par rapport à l'année précédente, d'après une étude de Symantec [3].

De nos jours, les spammeurs utilisent des techniques sophistiquées, telles que l'adresses email usurpées, l'obfuscation, le phishing, les messages de type "scareware" ou publicitaires pour atteindre les boîtes de réception des utilisateurs [4]. Les fournisseurs de services Internet utilisent des filtres anti-spam pour lutter contre les spams, et ces filtres utilisent une variété de techniques telles que l'analyse des en-têtes de messages, le filtrage basé sur le contenu et l'utilisation de listes noires d'adresses IP connues pour envoyer du spam [3]. Malgré cela, les spammeurs continuent d'affiner leurs techniques pour contourner les mesures de sécurité, tandis que les fournisseurs de services Internet continuent d'investir dans des solutions anti-spam plus sophistiquées.

Une étude de Symantec a révélé que les spams étaient de moins en moins efficaces, mais que les attaques par phishing étaient plus sophistiquées et ciblées [4]. Le pourcentage de spams dans le trafic email mondial est passé de 72,9 % en 2015 à 55,2 % en 2020, montrant une baisse significative mais toujours présente [5]. Le marché des solutions anti-spam devrait atteindre 4,5 milliards de dollars d'ici 2023, selon une étude de Radicati [3].

Plusieurs techniques ont vu le jour pour la détection de spams. Récemment, les méthodes de détection de spams basées sur l'analyse du contenu ont reçu une attention particulière grâce aux progrès fulgurants des techniques de classification de texte basées sur l'apprentissage automatique [6]. Ces techniques se basent sur l'utilisation de caractéristiques textuelles extraites des messages. Ainsi, des algorithmes de classification supervisée sont souvent utilisés, où le modèle est entraîné sur un ensemble de données étiquetées, associant chaque message à une étiquette indiquant s'il s'agit d'un spam ou d'un message légitime [6]. Les premières techniques se reposent sur des méthodes d'analyse statistique et de modélisation, et sont développées pour résoudre des problèmes spécifiques de traitement du langage naturel [7]. Parmi ces approches classiques, on trouve les méthodes basées sur des arbres de décision, les modèles de langage probabilistes tels que les modèles de Markov cachés, ainsi que TF-IDF pour représenter les mots sous forme de vecteurs numériques. Bien que ces méthodes aient été utiles pour la classification de spams, elles demeurent limitées car elles se basent sur des caractéristiques conçues de manière manuelle et ne tiennent pas compte de la sémantique et du contexte profond des messages [8].

Récemment, l'apprentissage profond est devenu une approche prometteuse pour la classification de spams. L'apprentissage profond est une branche de l'intelligence

artificielle qui utilise des réseaux de neurones pour extraire des caractéristiques à partir des données d'entrée, telles que le texte et les images. En utilisant des techniques de traitement du langage naturel, l'apprentissage profond peut apprendre à détecter les caractéristiques des e-mails spam à partir de données multimédias, ce qui permet de les distinguer des e-mails légitimes avec une grande précision. Plusieurs études ont montré que l'apprentissage profond peut être très efficace pour la classification de spams. A titre d'exemple, AbdulNabi et Yaseen [9] ont utilisé un réseau de neurones pour classifier les e-mails en spam ou non spam avec une précision de 98,6 %. De même, Zhu et al. [10] a utilisé l'apprentissage profond pour classer les spams en plusieurs catégories, y compris les spams publicitaires et les spams de phishing.

1.2 Problématique

Bien que d'importants progrès aient été réalisés dans le domaine de la détection de spams, des défis de taille subsistent, compromettant l'efficacité des méthodes actuelles. Les spammeurs, toujours plus ingénieux, utilisent des tactiques sophistiquées pour dissimuler leurs e-mails malveillants, rendant ainsi la tâche des algorithmes d'apprentissage profond plus complexe. Par exemple, ils peuvent incorporer du texte dans des images ou inclure des liens malveillants conduisant souvent à des sites Web frauduleux conçus pour voler des informations personnelles. Ces stratagèmes sophistiqués rendent la détection de spam fondée uniquement sur l'analyse du texte brut insuffisante. De plus, les méthodes traditionnelles de détection, essentiellement basées sur l'analyse textuelle, peuvent être aisément contournées par des techniques d'obfuscation, telles que l'utilisation de caractères spéciaux ou de codage Unicode pour rendre le contenu difficile à analyser automatiquement. En outre, l'apprentissage profond exige fréquemment des ensembles de données volumineux et étiquetés pour

donner des résultats efficaces, ce qui représente une contrainte temporelle et financière non négligeable en termes de ressources.

Dans ce contexte, notre contribution à cette problématique repose sur une approche novatrice utilisant l'analyse hiérarchique et multimodale pour classifier les spams multimédias grâce à l'apprentissage profond. L'inefficacité résultant du manque de prise en compte simultanée de plusieurs modalités, telles que les URL, le texte et les images, engendrant des lacunes significatives dans la détection du spam. Les courriels malveillants, en constante évolution, recourent à des tactiques sophistiquées telles que l'obfuscation, l'intégration d'images, les URL trompeuses, le polymorphisme et l'ajout de fausses pièces jointes, complexifiant la tâche des filtres anti-spam en altérant constamment les signatures.

Face à ces enjeux, notre modèle HMSD (*Hierarchical and Multimodal Spam Detection*) propose une approche novatrice en combinant de manière synergique des informations provenant de diverses modalités, incluant la classification des URL, l'analyse de l'obfuscation, la catégorisation du texte, l'extraction du texte des images et l'analyse des caractéristiques des images. Ainsi, HMSD surmonte les défis intrinsèques aux approches actuelles, offrant une solution plus robuste, adaptable et précise dans la détection des spams.

La pertinence de notre approche hiérarchique et multimodale se démarque par sa remarquable capacité à améliorer considérablement la détection du spam. HMSD se positionne comme une solution essentielle pour anticiper et contrer les évolutions futures des tactiques employées par les cybercriminels. L'adoption de HMSD garantit une protection substantielle contre les menaces émergentes, consolidant ainsi HMSD comme le choix optimal pour toute stratégie avancée de sécurité informatique.

Le reste de cet ouvrage est organisé comme suit : le chapitre 2 présente un état de l'art des techniques de détection de spams. Le chapitre 3 détaille notre approche HMSD, en expliquant les concepts utilisés et en détaillant notre méthodologie hiérarchique et multimodale utilisée pour améliorer la détection du spam, ainsi que les résultats expérimentaux, en comparant les performances de HMSD avec MMTD. Enfin, le chapitre 4 conclut l'ouvrage en discutant des contributions de notre recherche, des limitations identifiées et des pistes pour les travaux futurs dans le domaine de la détection de spams.

2. ÉTAT DE L'ART SUR LA DÉTECTION DE SPAM

Dans ce chapitre, nous explorerons les concepts et les approches modernes de détection de spam basées sur l'apprentissage automatique. Nous aborderons la catégorisation de texte et son application spécifique à la détection de spam. Nous passerons en revue les approches traditionnelles, les techniques d'apprentissage profond, ainsi que les méthodes intégrant des données multimodales.

2.1 Introduction aux concepts clés

Pour faciliter la compréhension des aspects techniques, nous définirons les concepts clés suivants :

Caractéristiques : Ce sont des propriétés spécifiques ou des traits distinctifs des données qui sont utilisés pour décrire ou analyser ces données dans un contexte particulier.

Attributs : Ce sont des variables spécifiques qui caractérisent une entité ou un objet et qui peuvent être utilisés comme entrées dans un modèle ou un système d'analyse.

Représentations : Elles désignent les formes ou les formats utilisés pour représenter les données dans un modèle ou un système donné, souvent sous forme structurée ou vectorielle.

Jeux de données : Ils sont constitués d'ensembles structurés d'informations ou de données utilisées pour l'entraînement, la validation ou le test de modèles d'apprentissage automatique.

Étiquettes : Ce sont des balises ou des annotations attribuées aux données pour indiquer leur classe, leur catégorie ou leur état, essentielles pour la supervision et l'évaluation des modèles d'apprentissage automatique.

2.2 Catégorisation de texte

La catégorisation de textes est un problème qui intéresse les chercheurs depuis relativement longtemps. On retrouve des travaux portant sur ce sujet depuis au moins le début des années 1960. Même s'il est certain que des avancées importantes ont été observées depuis, la recherche dans ce domaine est toujours très pertinente, car les résultats obtenus aujourd'hui sont encore sujets à amélioration [12].

La catégorisation de texte consiste à chercher une liaison fonctionnelle entre un ensemble de textes et un ensemble de catégories (étiquettes, classes). Cette liaison fonctionnelle, que l'on appelle également modèle de prédiction, est estimée par un apprentissage automatique [13]. Pour ce faire, il est nécessaire de disposer d'un ensemble de textes préalablement étiquetés, dit ensemble d'apprentissage, à partir duquel nous estimons les paramètres du modèle de prédiction le plus performant possible, c'est-à-dire le modèle qui produit le moins d'erreur en prédiction.

Formellement, la catégorisation de texte consiste à associer une valeur booléenne à chaque paire $(d_j , c_i) \in D \times C$, où D est l'ensemble des textes et C est l'ensemble des catégories. La valeur V (Vrai) est alors associée au couple (d_j , c_i) si le texte d_j appartient à la classe c_i tandis que la valeur F (Faux) lui sera associée dans le

cas contraire. Le but de la catégorisation de texte est de construire un modèle associant une ou plusieurs étiquettes (catégories) à un document d_j telle que la décision donnée par cette procédure « coïncide le plus possible » avec la fonction $\Phi : D \times C \rightarrow \{V, F\}$, la vraie fonction qui retourne pour chaque vecteur d_j une valeur c_i [13].

Le processus de catégorisation de texte se décompose en deux phases distinctes, telles qu'illustrées dans la figure 1 [13]. Tout d'abord, la phase d'apprentissage, constituée de plusieurs étapes visant à élaborer un modèle de prédiction performant. Cette phase débute par l'utilisation d'un ensemble de textes préalablement étiquetés, où chaque texte est associé à une catégorie particulière. À partir de ce corpus, les termes les plus pertinents (t_1, t_2, ..., t_k) sont extraits en fonction de la nature du problème à résoudre. Ce processus aboutit à la création d'un tableau dans lequel chaque texte est caractérisé par ses descripteurs spécifiques ainsi que son étiquette respective. En fin de compte, un algorithme d'apprentissage est appliqué afin de construire un modèle de prédiction Φ de haute qualité.

Dans la phase de classement des nouveaux textes, deux étapes distinctes sont déroulées. En premier lieu, il y a la recherche et la pondération des occurrences des termes (t_1, t_2, ..., t_k) dans le texte à classer, désigné par d_x. Ici, on peut utiliser une approche utilisant une représentation. Ensuite, le modèle de prédiction Φ est utilisé sur ces occurrences pour prédire l'étiquette appropriée pour le texte d_x.

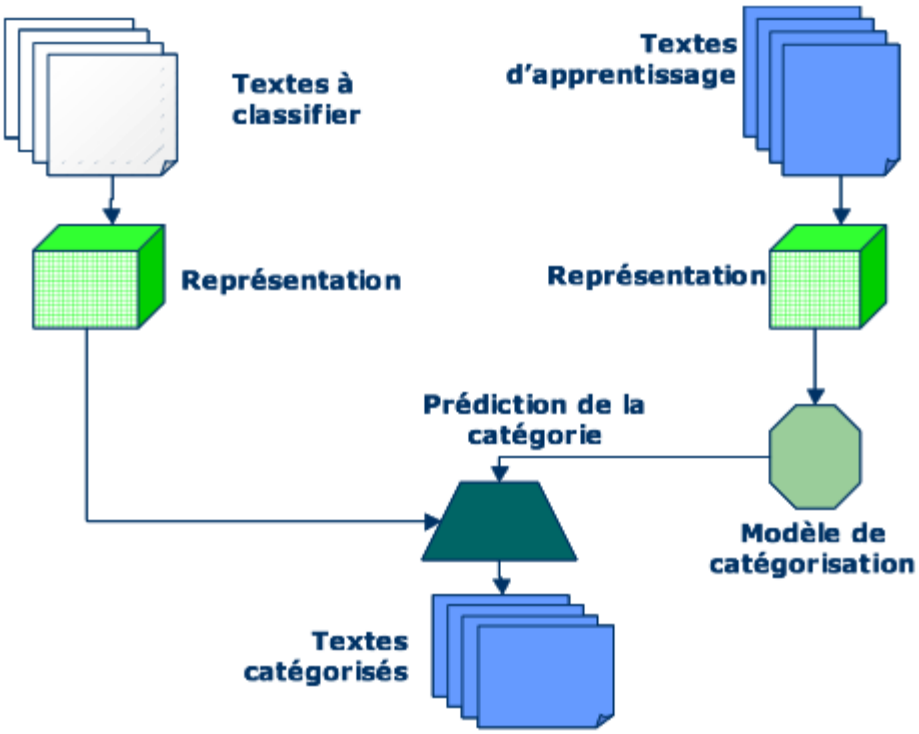

Figure 1 : Processus de catégorisation de textes [13].

En relation avec le processus de catégorisation de textes précédemment expliqué, il est essentiel de noter que le choix d'un mode de représentation devient crucial dans ce contexte, notamment lors de la sélection des attributs. Cette sélection implique l'élimination des attributs jugés inutiles pour la classification [12]. Comme illustré dans la figure 2, l'objectif de la catégorisation de textes consiste à apprendre à une machine comment classer un texte dans la catégorie appropriée en se basant sur son contenu [12]. Les catégories sont généralement associées aux sujets des textes, mais elles peuvent revêtir différentes formes pour des applications spécifiques. Par exemple, les techniques de catégorisation peuvent être utilisées pour résoudre des problèmes tels que l'identification de la langue d'un document, le filtrage de courrier

électronique pertinent ou indésirable, ainsi que la désambiguïsation de termes [66]. Un autre aspect qui varie en fonction des applications est la contrainte relative au nombre de catégories pouvant être assignées à un document donné.

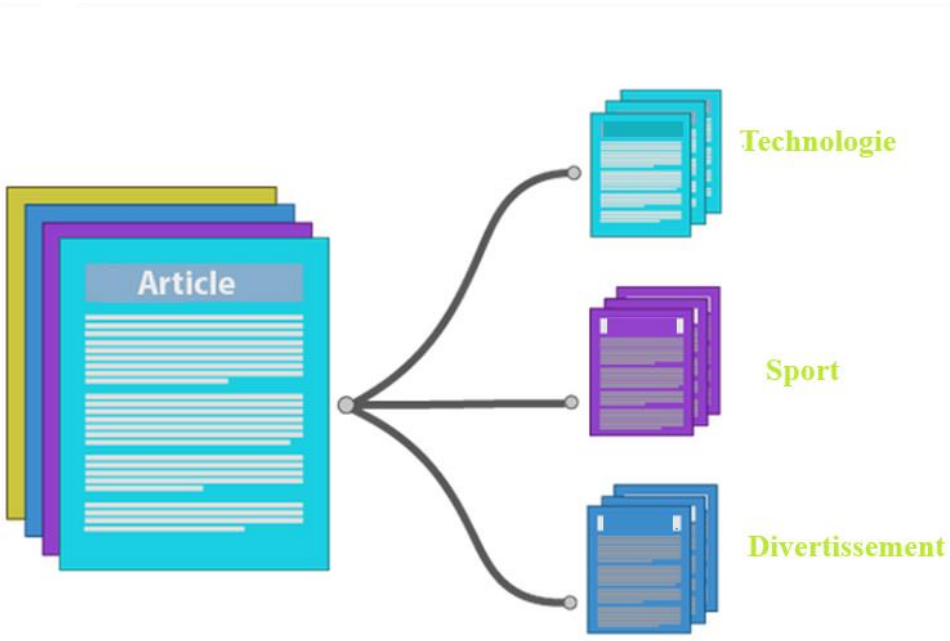

Figure 2 : La tâche de catégorisation automatique [13].

Parfois, il est souhaitable qu'un texte soit associé à une seule catégorie, tandis que dans d'autres cas, plusieurs catégories peuvent accueillir le même document. Il est important de noter que, dans le cadre de la catégorisation de textes, l'ensemble des catégories possibles est déterminé à l'avance. Le problème réside dans la nécessité de regrouper des documents en fonction de leur similarité. Toutefois, lorsque les groupes à former sont a priori inconnus, le processus, connu sous le nom de regroupement de textes, n'est pas pris en considération.

2.3 Détection de spams par classification de texte

L'apprentissage automatique est une discipline informatique qui se concentre sur le développement d'algorithmes et de modèles permettant aux machines d'apprendre à partir de données et d'améliorer leurs performances sur des tâches spécifiques sans être explicitement programmées.

Dans le domaine de la détection de spams basée sur la classification de texte, plusieurs approches ont été développées au fil des décennies. Parmi les approches traditionnelles, on trouve l'utilisation de méthodes telles que la régression logistique (RL) [55], la forêt aléatoire (RF) [53], machines à vecteurs de support (SVM) [52], Naïve Bayes (NB) [8]. Ces approches ont été développées avant l'avènement des architectures d'apprentissage profond, mais elles continuent de jouer un rôle essentiel dans certaines applications et servent souvent de base pour les méthodes plus récentes. Bien que les techniques traditionnelles d'apprentissage automatique fonctionnent bien dans de nombreux domaines, elles nécessitent encore beaucoup d'interférences ou de conseils de la part de spécialistes humains lorsque les gens essaient d'appliquer ces technologies pour résoudre des problèmes. Par exemple, extraire et représenter les caractéristiques des données est toujours un travail difficile mais indispensable pour les scientifiques de l'apprentissage automatique. En d'autres termes, la capacité insuffisante de nombreux classificateurs d'apprentissage automatique traditionnels est une limitation majeure à une application plus efficace et massive.

Avec l'avènement des techniques d'apprentissage profond, de nouvelles méthodes ont vu le jour pour aborder le problème du spam. Parmi ces approches figurent le *réseau neuronal convolutif* (CNN) [15], le *réseau neuronal récurrent* (RNN) [16], ainsi que

la mémoire à long court terme (LSTM) [17], une variante performante des RNN spécialement conçue pour gérer les dépendances à long terme dans les données séquentielles. Ces techniques d'apprentissage profond sont capables non seulement d'apprendre beaucoup plus de fonctionnalités, mais également d'extraire davantage de fonctionnalités de niveau supérieur qui sont formées par la composition de fonctionnalités de niveau inférieur.

Cependant, ce qui distingue fondamentalement l'apprentissage profond des méthodes traditionnelles réside dans sa capacité à automatiser le processus d'extraction de caractéristiques pertinentes à partir des données. Contrairement aux approches traditionnelles qui nécessitent souvent une ingénierie manuelle des caractéristiques, les algorithmes d'apprentissage profond peuvent apprendre de manière autonome des représentations hiérarchiques complexes à partir des données brutes.

Cette diversité d'approches offre des solutions adaptées à différentes situations, chaque catégorie ayant ses avantages et ses limites. Dans l'ensemble, la combinaison de ces approches traditionnelles et profond a enrichi le domaine de la détection de spams et continue de contribuer à son développement.

2.3.1 Approches traditionnelles de détection de spam

Les techniques traditionnelles en apprentissage automatique peuvent être classées en différentes catégories. Tout d'abord, on trouve les méthodes d'apprentissage supervisé, qui s'appuient sur un ensemble de données d'entraînement annotées, où chaque exemple est associé à une étiquette de classe ou une sortie désirée. Dans le contexte de la détection de spam, les classifieurs bayésiens ont été largement utilisés. Les classifieurs bayésiens se basent sur le théorème de Bayes pour estimer les

probabilités conditionnelles des classes et sont couramment utilisés pour filtrer les courriers électroniques indésirables. Des études telles que celle menée par Sahami et al. [7] ont démontré l'efficacité de ces méthodes dans la détection de spam en analysant les mots clés et les caractéristiques des messages.

Les machines à vecteurs de support constituent une autre technique populaire de classification supervisée dans la détection de spam. Les SVM cherchent à trouver un hyperplan qui sépare de manière optimale les différentes classes dans un espace multidimensionnel. Dans le contexte de la détection de spam, les SVM ont été utilisés avec succès pour la classification de courriers électroniques en spam et en non-spam, en se basant sur des caractéristiques telles que la fréquence des mots, les en-têtes et les liens hypertextes. Les travaux de Drucker et al. [18] ont montré l'efficacité des SVM dans la détection de spam en exploitant ces caractéristiques pour créer un classifieur robuste.

Ensuite, les techniques d'apprentissage non supervisé visent à découvrir des structures cachées ou des modèles intrinsèques dans les données sans avoir besoin d'étiquettes pour les données d'entraînement. Bien qu'elles soient moins courantes dans la détection de spam, certaines méthodes de regroupement (clustering) telles que le k-moyennes peuvent être utilisées pour regrouper les courriers électroniques en fonction de leurs similarités, permettant ainsi de détecter des tendances ou des comportements anormaux dans les données. L'analyse en composantes principales (PCA) est une autre technique non supervisée qui pourrait être appliquée pour réduire la dimensionnalité des caractéristiques des courriers électroniques dans le but de faciliter la détection de spam.

Par ailleurs, les méthodes d'apprentissage semi-supervisé pourraient être utilisées dans le contexte de la détection de spam pour tirer parti à la fois des courriers électroniques annotés et non annotés. Cela peut être particulièrement utile lorsque l'annotation manuelle des courriers électroniques est coûteuse ou difficile à obtenir. Des recherches telles que celles menées par Zhang et al. [19] ont montré que l'utilisation d'approches semi-supervisées peut améliorer la précision de la détection de spam en exploitant les informations non annotées pour mieux généraliser le modèle.

Enfin, les techniques d'apprentissage par renforcement n'ont pas été largement utilisées dans la détection de spam, mais elles pourraient potentiellement être explorées dans des contextes plus avancés de détection de menaces spécifiques, où l'agent doit apprendre à prendre des décisions séquentielles pour optimiser le filtrage des courriers électroniques indésirables.

2.3.2 Apprentissage profond pour la détection de spam

Dans la quête d'une détection de spam plus précise et adaptée aux défis croissants, les techniques d'apprentissage profond se sont imposées comme des solutions puissantes et polyvalentes. Ces approches ont introduit une révolution dans le domaine de l'intelligence artificielle en permettant aux machines d'assimiler des volumes massifs de données et d'extraire des informations significatives. Les réseaux de neurones profonds, caractéristiques clés de ces approches, ont le pouvoir de dévoiler des représentations hautement abstraites et complexes à partir de données brutes. Lorsqu'il s'agit de la catégorisation de texte, leur efficacité s'est manifestée à travers les réseaux neuronaux ainsi que les modèles de type Transformer [20], ceux-ci offrant des performances remarquables en matière de classification de texte.

Cette capacité à comprendre et à représenter des informations contextuelles s'étend également au domaine de l'analyse du contenu visuel [21]. Ici, les techniques d'apprentissage profond se révèlent très utiles pour extraire des caractéristiques visuelles et détecter des schémas subtils de spam cachés au sein d'images. Cependant, le vrai potentiel émerge lorsqu'on associe ces approches novatrices aux méthodes traditionnelles déjà éprouvées. Cette fusion stratégique crée une synergie qui peut non seulement améliorer considérablement la détection de spam, mais aussi ouvrir de nouvelles voies pour mieux gérer l'information dans un environnement complexe et saturé de données.

En effet, l'intégration intelligente de l'apprentissage profond aux méthodes classiques offre des avantages multiples. Les CNN se prêtent à l'analyse de texte, tandis que les modèles de type Transformer excellent dans la capture des relations complexes entre les mots. Les RNN et les architectures LSTM sont adaptés à l'exploitation des séquences de mots, permettant la détection de motifs dissimulés dans les messages.

Les RNN [16] et les LSTM [17], jouent un rôle crucial dans la détection de spam en exploitant les séquences de données, telles que les messages textuels. Les RNN sont une classe de réseaux neuronaux conçus pour traiter des données séquentielles, où chaque élément de la séquence est traité dans un ordre particulier, tenant compte des informations contextuelles antérieures. Cette capacité à conserver la mémoire du contexte passé est essentielle pour la détection de spam, car elle permet au modèle d'analyser la séquence des mots dans un message et d'identifier des motifs de spam.

Les LSTM sont une extension des RNN conçue pour résoudre le problème de dégradation du gradient qui se propage inefficacement sur de longues séquences, ce

qui est courant dans les RNN traditionnels. Les LSTM incorporent des mécanismes de porte qui permettent de contrôler le flux d'information à travers les cellules de mémoire. Cela leur permet de conserver des informations sur des intervalles temporels plus longs, ce qui est particulièrement important pour la détection de spam, car certains messages peuvent contenir des indices de spam dissimulés sur plusieurs étapes de la séquence.

L'impact des méthodes d'apprentissage profond ne se limite pas à la simple détection de spam. Elles abordent également les défis liés à la complexité des données et à l'absence d'interférence humaine directe. En facilitant l'apprentissage et l'extraction de caractéristiques avancées, ces méthodes ouvrent la voie à une amélioration significative de l'efficacité et à une adaptation continue dans la gestion des informations. En somme, l'intégration des techniques d'apprentissage profond dans la détection de spam engendre une perspective prometteuse pour relever les défis actuels et futurs de la gestion d'informations dans le panorama numérique en constante mutation.

2.4 Détection de spams basée sur l’analyse sémantique

L’approche présentée par Saidani et al. [22] examine les courriels à deux niveaux sémantiques distincts. Au premier niveau, les courriels sont classés selon leurs domaines pour offrir une perspective sémantique distincte des spams dans chaque domaine. À l'aide d'un ensemble sélectionné de caractéristiques, les auteurs catégorisent les courriels selon des domaines spécifiques tels que Santé, Finance, Adulte, etc. Plusieurs algorithmes de classification ont été comparés, et le classifier fournissant la catégorisation la plus précise des courriels, a été identifié.

Au deuxième niveau, un classificateur de spam est construit pour chaque domaine spécifique en utilisant des fonctionnalités sémantiques. Ces fonctionnalités combinent des règles spécifiées manuellement et générées automatiquement, construites à partir de courriels étiquetés. Les règles spécifiées manuellement représentent des connaissances d'expert, construites à l'aide d'expressions régulières. Les entités générées automatiquement sont obtenues à l'aide de la méthode CN2-SD. Chaque règle produit un résultat binaire et agit en tant que classificateur indépendant pour la détection de spam. Les caractéristiques sémantiques ainsi obtenues sont ensuite exploitées pour construire des classificateurs spécialisés destinés à détecter les spams propres à chaque domaine.

2.5 Ingénierie des caractéristiques (Feature Engineering)

L'ingénierie des caractéristiques est essentielle dans la détection de spam, permettant la sélection et l'extraction des attributs pertinents pour améliorer les modèles d'apprentissage automatique. Ce processus englobe la sélection, la création et la transformation des caractéristiques d'un ensemble de données pour optimiser les performances. En complément de l'analyse textuelle, diverses autres caractéristiques peuvent être exploitées pour renforcer la classification de spam.

Une des approches courantes est l'analyse des métadonnées des courriels. Les métadonnées fournissent des informations contextuelles sur les messages, telles que les informations d'expéditeur, les dates d'envoi, les en-têtes, et bien plus encore. Ces informations peuvent être utilisées pour détecter des schémas suspects. Par exemple, une adresse IP provenant d'une source non fiable peut être un indicateur de spam. En

exploitant ces métadonnées, les techniques d'ingénierie des caractéristiques permettent d'améliorer la précision des modèles de classification [23].

Les attributs textuels des courriels peuvent également être exploités. Par exemple, la longueur du message peut être un indicateur pertinent pour distinguer les courriels spam des courriels légitimes [23]. Les courriels spam ont souvent tendance à être courts et concis, tandis que les e-mails légitimes sont généralement plus détaillés. En extrayant cette caractéristique, il est possible d'améliorer la performance des modèles de classification.

La présence d'URLs dans les courriels est également un aspect crucial. Les courriels légitimes sont souvent associés à des liens vérifiés ou à des URLs réputées. En revanche, les courriels spam sont souvent associés à des liens suspects ou à des URLs non fiables. En analysant la présence de ces liens et en les considérant comme une caractéristique, il est possible de détecter plus efficacement les courriels de spam [24].

L'ingénierie des caractéristiques offre une approche puissante pour améliorer la détection de spam. En exploitant les métadonnées des courriels, ainsi que les attributs textuels tels que la longueur du message et la présence de liens, il est possible d'accroître la précision des modèles de classification. L'application de techniques d'ingénierie des caractéristiques dans la détection de spam constitue un domaine de recherche actif, offrant de nombreuses opportunités d'innovation et d'amélioration des systèmes existants.

Ces techniques traditionnelles ont joué un rôle essentiel dans l'évolution de l'apprentissage automatique appliqué à la détection de spam, et continuent d'être

utilisées dans certains systèmes de filtrage. Cependant, elles ont également ouvert la voie à l'émergence de méthodes plus avancées basées sur l'apprentissage profond, telles que les réseaux neuronaux, qui ont démontré des performances supérieures dans la détection de spam.

2.6 Détection de spam à l'aide de données multimodales (multimodal data)

La détection de spam à l'aide de données multimodales se réfère à l'utilisation d'ensembles de données intégrant plusieurs types de modalités d'information, comme le texte, l'image, l’obfuscation, l'audio ou d'autres formes de données, afin d'améliorer la précision des modèles de détection de spam. Au fil des décennies, le paysage de la détection de spam a évolué pour répondre à la sophistication croissante des méthodes de spamming. Parmi les approches émergentes qui ont enrichi ce domaine, les méthodes multimodales se sont imposées comme des solutions puissantes. L'approche Texte-Image [21], par exemple, transcende les limites de l'analyse textuelle seule en combinant habilement les éléments textuels et visuels des e-mails. En intégrant les informations extraites des images, comme les logos ou les publicités, cette méthode révèle des schémas de spam complexes qui échapperaient à une détection basée exclusivement sur le texte.

Parallèlement, l'approche Audio-Texte [25] offre une perspective novatrice en examinant les éléments audio présents dans les e-mails, tels que les messages vocaux ou les clips sonores. En fusionnant ces éléments audio avec le texte, cette méthode enrichit la détection en explorant une dimension jusqu'alors peu exploitée.

L'Analyse Temporelle [8], met en lumière les schémas temporels d'envoi de courriels. Par exemple, considérons une situation où un utilisateur normal envoie

régulièrement des courriels tout au long de la semaine, avec une activité relativement constante. Soudainement, une augmentation significative du nombre de courriels envoyés est observée les fins de semaine. Ce schéma temporel inhabituel pourrait être indicatif d'une activité suspecte, notamment dans le contexte de la détection de spams. Les schémas temporels peuvent inclure des variations régulières, des pics d'activité, des cycles ou d'autres comportements spécifiques à certaines périodes. Ce qui peut être utile pour détecter des activités anormales ou suspectes, comme celle associée à des envois de spam.

L'obfuscation est une technique sophistiquée utilisée pour rendre le contenu de données illisible ou difficile à comprendre sans en altérer le sens. Elle est largement utilisée dans divers domaines tels que la sécurité informatique, la protection de la propriété intellectuelle, la lutte contre le spam et les logiciels malveillants [26]. Dans le contexte de la classification de spam, l'obfuscation est couramment employée par les spammeurs pour échapper aux filtres anti-spam et tromper les mécanismes de détection. Les techniques d'obfuscation dans les e-mails indésirables peuvent être très variées. Elles incluent l'utilisation de mots-clés spécifiques pour attirer l'attention des utilisateurs, l'insertion de caractères spéciaux ou de fautes d'orthographe intentionnelles pour contourner les filtres, et même le masquage de liens et d'adresses e-mail pour déjouer les analyses automatisées [27]. Ces tactiques rendent la détection de spam plus complexe et exigent des approches avancées pour contrer ces tentatives de dissimulation.

La détection de l'obfuscation revêt une importance cruciale dans la lutte contre le spam. Les chercheurs et les experts en sécurité s'efforcent de développer des méthodes sophistiquées pour identifier ces schémas et améliorer l'efficacité des

systèmes de détection. Certaines approches [28] s'appuient sur l'apprentissage automatique, où des modèles sont formés en utilisant des exemples d'e-mails obfusqués et non obfusqués. D'autres méthodes [28] analysent linguistiquement et sémantiquement pour identifier les e-mails suspects.

La fusion de données multimodales, peut capitaliser sur la complémentarité des modèles spécifiques de différentes modalités, telles que le texte et l'image. Cette fusion permet d'agréger les prédictions de ces modèles pour obtenir une décision globale bénéficiant des avantages de chaque approche tout en minimisant leurs faiblesses individuelles [29].

L'analyse du contenu visuel constitue un aspect important de la détection de spam, notamment lorsqu'il s'agit de traiter des images. Les techniques d'analyse d'images offrent la possibilité d'extraire des caractéristiques visuelles et de détecter des schémas de spam. Par exemple, en utilisant des algorithmes de traitement d'images, il est possible d'identifier la présence de texte suspect dans les images, tels que des adresses URL non fiables ou des appels à l'action douteux. De plus, en analysant les logos de spam connus ou les motifs d'image couramment utilisés dans les e-mails indésirables, il devient possible de les repérer plus efficacement.

Les travaux de recherche ont démontré l'efficacité de ces techniques d'analyse d'images pour la détection de spam visuel. Zhang et al. ont proposé des méthodes avancées d'analyse d'images pour extraire des caractéristiques visuelles et les utiliser dans la détection de spam [21]. Leur étude a montré que l'analyse du contenu visuel peut fournir des informations complémentaires précieuses pour améliorer la précision des modèles de détection de spam. De même, Hanjalic a étudié spécifiquement l'analyse du contenu visuel pour la détection de spam dans les médias sociaux [5]. Leur

recherche a mis en évidence l'importance de considérer les caractéristiques visuelles dans les images partagées sur les plateformes de médias sociaux, et a proposé des approches pour identifier les contenus indésirables visuellement.

Il est important de souligner que l'analyse du contenu visuel ne se substitue pas aux méthodes traditionnelles d'analyse de texte, mais elle vient compléter et renforcer les techniques existantes de détection de spam.

2.7 Détection de spam avec les LLMs

Les grands modèles de langage (LLM) représentent une avancée majeure dans le domaine de l'intelligence artificielle et la détection de spam. Ces algorithmes d'apprentissage profond sont capables d'accomplir une variété de tâches liées au langage naturel, allant de la traduction à la génération de contenu en passant par la compréhension et la prédiction [45]. Ils reposent sur des architectures de modèles de transformateur et sont entraînés sur de vastes ensembles de données, ce qui leur permet d'apprendre les schémas et les relations entre les éléments du langage. Les principaux composants des LLMs comprennent des couches de réseaux de neurones récurrents, des couches à action directe, des couches de plongement et des couches d'attention [45]. Ces composants travaillent ensemble pour traiter les entrées textuelles et générer des sorties précises. Les LLMs se déclinent en plusieurs types, notamment les modèles génériques de langage, adaptés aux instructions et aux dialogues. Chaque type est adapté à des applications spécifiques, telles que la récupération d'informations, l'analyse des sentiments et la génération de texte ou de code.

Parmi les modèles de langage les plus populaires, on trouve quelques noms bien connus. PaLM [51], développé par Google, est un modèle de transformateur polyvalent capable d'effectuer des tâches variées telles que la génération de code, la compréhension de blagues, et même la traduction. BERT [40], également de Google, est un modèle basé sur des transformateurs qui excelle dans la compréhension du langage naturel et la réponse à des questions. Transformer-XL [43] se distingue par sa capacité à générer des prédictions dans un ordre aléatoire, offrant une approche différente de modélisation linguistique. Enfin, les transformateurs génératifs pré-entraînés tels que GPT, notamment GPT-3 et GPT-4 d'Open-AI, sont parmi les plus célèbres [45]. Ces modèles peuvent être adaptés à diverses tâches spécifiques, comme la gestion des relations clients ou les analyses financières, démontrant ainsi leur polyvalence et leur potentiel dans différents domaines d'application.

Dans la continuité des avancées dans le domaine de l'intelligence artificielle et de la détection de spam, le modèle MMTD (*A Multilingual and Multimodal Spam Detection Model Combining Text and Document Images*) [54] représente une étape significative. Il utilise deux grands modèles de langage de pointe : Beit pour l'analyse d’image et Bert pour le traitement textuel. Cette combinaison de modèles permet une classification plus précise des contenus. Les performances du modèle ont été évaluées à l'aide du jeu de données EDP, démontrant sa supériorité par rapport à d'autres approches multimodales [54]. Ce progrès souligne l'importance de ces modèles LLM dans l'amélioration des systèmes de détection de spam et de sécurité en ligne. L'introduction du jeu de données EDP, un ensemble de données multilingue de spam multimodal, a permis aux chercheurs d'évaluer de manière approfondie les performances du modèle dans des situations réelles. Les résultats des expérimentations indiquent clairement que le modèle MMTD surpasse de manière significative les autres

approches multimodales [54], soulignant ainsi son potentiel considérable pour renforcer la sécurité et la fiabilité des systèmes de communication en ligne.

Dans le chapitre suivant, nous approfondirons notre méthodologie d'analyse hiérarchique et multimodale dans le domaine de la détection de spam. Nous fournirons une explication détaillée de cette méthode et discuterons des résultats issus de nos expériences approfondies. Cette étude approfondie nous permettra de mieux appréhender l'impact de l'analyse de multiples modalités telles que le texte, les URLs, la catégorisation, le texte d'image et les caractéristiques d'image dans la détection de spam. De plus, nous évaluerons l'efficacité de notre approche en la comparant au modèle multimodal MMTD.

3. METHODOLOGIE

3.1 Définition

Dans cette section, nous examinerons en profondeur les concepts fondamentaux liés à l'utilisation des données multimodale et l'analyse thématique. Pour commencer, nous aborderons le traitement automatique du langage naturel (TALN) en contexte de détection de spam, mettant en évidence l'importance cruciale du TALN dans la manipulation de données textuelles. Ensuite, nous détaillerons les mécanismes d'attention utilisés dans notre approche, en expliquant leur rôle crucial dans le traitement du texte. Nous discuterons en particulier de l'attention avec produit scalaire mise à l'échelle et de l'attention à têtes multiples, en soulignant leur impact sur la performance de notre modèle. Enfin, nous explorerons les représentations vectorielles, en mettant en lumière l'évolution des techniques de plongements de mots telles que le modèle GPT-4, BERT et USE dans la représentation du langage naturel.

3.1.1 Traitement du langage naturel

Les données textuelles sont l'un des types de données les plus couramment utilisés par les entreprises aujourd'hui. Cependant, en raison de leur manque de structure claire, il peut être difficile et chronophage d'extraire des informations à partir de ces données. Le traitement des données textuelles relève du traitement du langage naturel, qui est l'un des sous-domaines de l'intelligence artificielle [1].

Le TALN est un sous-domaine de l'intelligence artificielle qui étudie comment les ordinateurs interagissent avec les langues humaines et comment les programmer pour traiter et analyser de grandes quantités de données textuelles [30]. La recherche en TALN est également menée dans des domaines tels que les sciences cognitives, la linguistique et la psychologie. La détection des spams est l'un des cas d'utilisation du TALN [9]. Le langage humain est fragmenté dans le traitement du langage naturel afin que la structure grammaticale des phrases et le sens des mots puissent être analysés et compris dans leur contexte [11]. Cela permet aux ordinateurs de lire et d'analyser un texte parlé ou écrit de façon avancée. Avec les avancées de l'apprentissage profond, le TALN a pris un essor remarquable durant les dernières années. Plusieurs architectures de réseaux de neurones ont été proposées pour analyser et interpréter le sens du texte, et même générer du contenu textuel, en s'entrainant sur des corpus constitués de millions de documents. L'une des techniques récentes qui a avancé significativement le TALN est l'utilisation de l'attention dans les modèles séquentiels [32, 33], et par la suite dans les Transformers [35]. L'utilisation de l'attention de mieux capter le sens des mots dans leur contexte, qui permet de mieux nuancer les contenus textuels et de les synthétiser.

3.1.2 Mécanisme d'attention

Les mécanismes d'attention peuvent être implémentés de différentes manières, selon la tâche à accomplir et la structure du modèle de réseau de neurones. Parmi les techniques les plus courantes, on peut citer l'auto-attention, l'attention multi-tête, l'attention paraplégique et l'attention contextuelle [34]. Concrètement, le mécanisme d'attention produit des poids qui représentent l'importance des éléments en fonction de leur corrélation avec le contexte. Le but principal du mécanisme d'attention est de

découvrir la partie la plus pertinente de la séquence d'entrée pour produire une bonne prédiction pour la suite.

Le mécanisme d'attention a été introduit comme une amélioration de l'état caché dans le modèle d'encodeur-décodeur RNN [31]. La contribution du mécanisme d'attention réside dans le fait qu'il calcule les poids en fonction de l'ensemble des états cachés générés par l'encodeur. Le décodeur utilise ensuite cette combinaison pondérée de tous les états cachés, plutôt que de se focaliser exclusivement sur le dernier état caché du réseau de neurones. Cet état est d'ailleurs appelé "Contexte". Par ailleurs, le mécanisme d'attention est appliqué au domaine de la vision par ordinateur par Xu et al. [32], et ils ont également proposé deux approches différentes de l'attention nommées "soft attention" et "hard attention". Luong et al. ont proposé d'utiliser une attention globale et une autre locale [33]. L'attention globale est similaire au modèle de Bahdanau et al. [31] avec une architecture plus simple, tandis que l'attention locale est une combinaison d'attention douce et dure de Xu et al. [32]. Dans ce qui suit, nous présentons quelques définitions de base se reportant au mécanisme d'auto-attention.

3.1.2.1 Définition de l'auto-attention

L'auto-attention, au cœur du traitement du langage naturel et de l'apprentissage profond a révolutionné la manière dont les modèles d'apprentissage machine traitent les séquences d'entrée. Contrairement aux approches statiques, l'auto-attention offre une adaptabilité dynamique, permettant aux modèles de se concentrer sur des parties spécifiques d'une séquence en fonction de leur pertinence contextuelle [34]. L'essence même de cette technique réside dans sa capacité à capturer les relations et les dépendances complexes entre différentes positions au sein d'une séquence. En d'autres termes, le modèle apprend à attribuer des poids différenciés à chaque élément de la

séquence, créant ainsi une représentation globale plus nuancée [34]. Cette flexibilité est particulièrement évidente dans le modèle Transformer, qui a joué un rôle déterminant dans la diffusion de cette approche.

Dans le contexte du modèle Transformer, l'auto-attention permet au modèle de hiérarchiser l'importance des éléments de la séquence en attribuant des pondérations spécifiques. Cette capacité à ajuster dynamiquement l'attention est particulièrement utile lors de l'exécution de tâche complexe telle que la détection de spam. Le modèle peut ainsi se concentrer de manière sélective sur des parties spécifiques de l'entrée, améliorant ainsi son aptitude à traiter des informations cruciales pour la tâche en cours. En favorisant la capture de dépendances à long terme au sein des données séquentielles, l'auto-attention élève la performance des modèles dans une multitude de domaines liés au traitement du langage naturel. L'auto-attention représente une pierre angulaire cruciale dans l'évolution des capacités des modèles d'apprentissage machine, offrant une flexibilité et une sophistication inégalées dans la compréhension et la manipulation de données séquentielles.

3.1.2.2 Représentation de l'auto-attention

L'auto-attention est un mécanisme essentiel utilisé dans les modèles de traitement du langage naturel. Les termes Q (Requête), K (Clé), et V (Valeur) sont fondamentaux. La Requête représente la question posée sur un élément de la séquence, la Clé est une représentation de cet élément qui détermine sa pertinence par rapport à la question, et la Valeur est l'information extraite de l'élément lorsque sa pertinence est confirmée. Ces concepts permettent au modèle d'attention de calculer comment les éléments d'une séquence interagissent.

Comme indiqué ci-dessus, l'auto-attention prend trois vecteurs en entrée : la clé K, la valeur V et la requête Q. Pour calculer l'attention, nous attribuons des poids à chaque élément de la séquence en utilisant une "fonction de compatibilité". Ces poids sont utilisés pour effectuer une moyenne pondérée des vecteurs de valeur V. Ces poids sont obtenus comme suit :

- Requête, $Q = EW_q$: La Requête Q est calculée en multipliant la représentation originale de chaque élément de la séquence par une matrice de poids W_q.
- Clé, $\mathrm{K} = EW_k$: La Clé K est obtenue de manière similaire en multipliant la représentation originale de chaque élément par une matrice de poids W_k.
- Valeur, $\mathrm{V} = EW_v$: La Valeur V est également calculée en multipliant la représentation originale de chaque élément par une matrice de poids W_v.

où $\mathrm{W}_q \in \mathbf{R}^{d_{\mathrm{model}} \times d_{\mathrm{model}}}$, $\mathrm{W}_k \in \mathbf{R}^{d_{\mathrm{model}} \times d_{\mathrm{model}}}$, $\mathrm{W}_v \in \mathbf{R}^{d_{\mathrm{model}} \times d_{\mathrm{model}}}$ et d_{model} est l'hyperparamètre de dimension du modèle.

Dans la formule, un produit scalaire du vecteur de requête Q et du vecteur de clé K est effectué pour mesurer la similarité de chaque paire de requête-clé. Plus la similarité est élevée, plus le poids attribué à la valeur correspondante sera élevé. La transposition est utilisée pour aligner correctement les dimensions des matrices afin de

permettre la multiplication matricielle et le calcul des scores d'attention. Pour normaliser ce produit scalaire, on le divise par la racine carrée d_k, qui représente la dimension de l'espace vectoriel dans lequel les clés K sont représentées. L'introduction de ce facteur d'échelle est essentielle pour éviter que les valeurs d'entrée du calcul softmax ne tombent dans une plage où la sortie deviendrait négligeable. Ensuite, une opération softmax est appliquée aux résultats. Le vecteur final obtenu est ensuite multiplié par le vecteur de valeur V, pour obtenir les scores d'attention finaux.

Cette formule permet au modèle d'attribuer des poids d'attention élevés aux paires de mots qui ont des relations importantes sur le plan sémantique ou grammatical. En revanche, elle diminue les poids pour les paires de mots moins pertinentes. Ainsi, elle permet de mettre en avant les informations cruciales et de capturer les relations significatives entre les mots d'une séquence, comme illustré dans la figure 3. L'architecture de l'auto-attention, implémentée à travers plusieurs étapes clés, permet à chaque mot de la séquence de se concentrer sur les autres mots de manière adaptative. Tout d'abord, il y a une multiplication matricielle MatMul entre les vecteurs Query Q et Key K, suivie d'une mise à l'échelle Scale pour stabiliser les gradients. Ensuite, ces scores d'attention sont normalisés via une fonction softmax, générant ainsi des poids de probabilité. Finalement, une dernière multiplication matricielle MatMul est effectuée avec les vecteurs Value V, produisant les représentations finales pondérées des mots.

L'intégration de cette architecture dans le modèle Transformer a marqué une avancée significative dans le TALN. Elle a considérablement amélioré la capacité du modèle à comprendre et à générer du texte de manière précise et cohérente, en capturant

efficacement les relations et les dépendances entre les éléments d'une séquence textuelle.

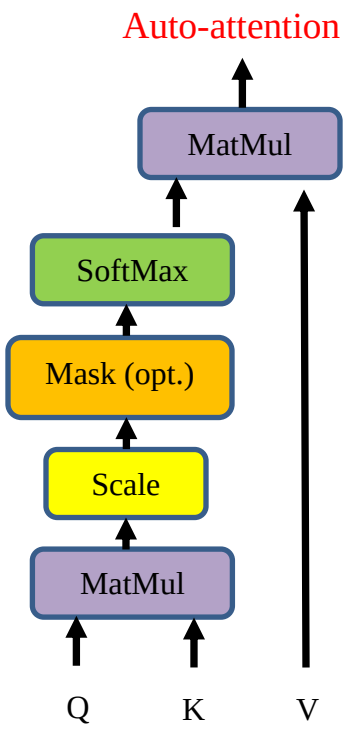

Figure 3 : Représentation de l'auto-attention pour une tête. [34].

Dans l'exemple de l'utilisation de l'auto-attention illustré dans la figure 4, le symbole "E" désigne le plongement, où chaque élément de la séquence, comme les mots "I am trying to understand", est transformé par une multiplication avec une matrice de poids spécifique. Cette opération de multiplication matricielle permet de générer une nouvelle représentation pour la Requête, la Clé et la Valeur de chaque élément de la séquence. Cette étape de transformation linéaire prépare ainsi les données pour le calcul des poids d'attention, lesquels jouent un rôle crucial dans la capture des

relations contextuelles entre les éléments de la séquence lors de l'application de l'auto-attention.

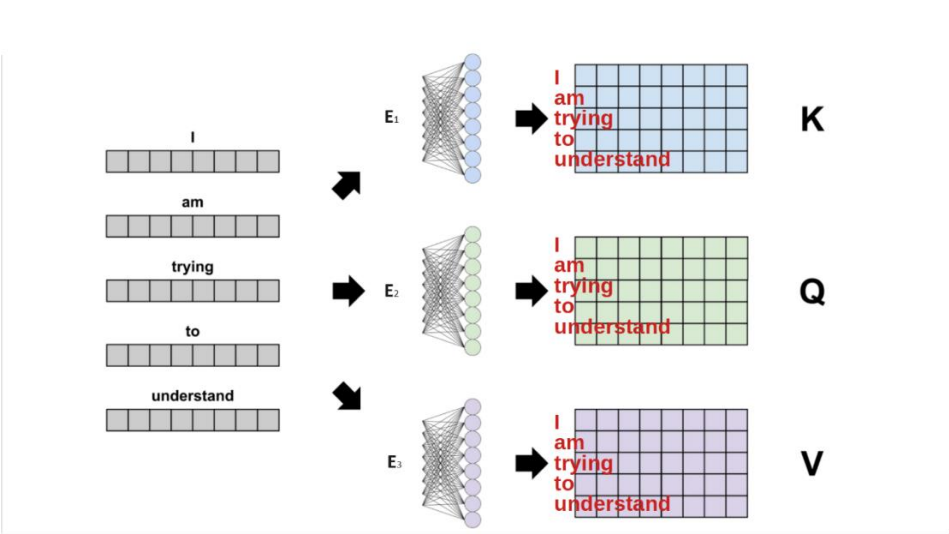

Figure 4 : Exemple de l'utilisation de l'auto-attention [34].

3.1.2.3 Auto-attention à têtes multiples

L'attention à têtes multiples est une technique essentielle en apprentissage automatique, principalement utilisée dans les modèles de type Transformer. Elle divise l'espace vectoriel en plusieurs sous-espaces, appelés "têtes", qui effectuent des calculs d'attention indépendants sur différentes parties des données d'entrée [34]. Cette approche exécute le mécanisme d'attention plusieurs fois en projetant les matrices Clé, Requête et Valeur dans des espaces de dimensions réduites, générant ainsi plusieurs

"têtes" d'attention, comme le montre la figure 5. Ces têtes sont ensuite combinées en utilisant une matrice de poids carrée, conformément à l'équation suivante :

$$\text{MultiHead(Q, K, V)} = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\text{W}^O$$

$$\textit{où } \text{head}_i = \text{Attention}(\text{QW}_i^Q, \text{KW}_i^K, \text{VW}_i^V)$$

$$\textit{Lorsque} : \text{W}_i^Q \in \mathbf{R}^{d_{\text{model}} \times d_k}, \text{W}_i^K \in \mathbf{R}^{d_{\text{model}} \times d_k}, \text{W}_i^V \in \mathbf{R}^{d_{\text{model}} \times d_v} \text{ et } W^O \in \mathbf{R}^{hd_v \times d_{\text{model}}}.$$

Cette formule est essentielle pour expliquer comment le modèle gère et traite des informations complexes dans les données d'entrée. L'opération d'attention est divisée en plusieurs "têtes" $(\text{head}_1, ..., \text{head}_h)$, chacune d'entre elles représentant une sous-opération d'attention indépendante. Chaque tête utilise des matrices de projection spécifiques (W_i^Q, W_i^K, W_i^V) pour transformer les données d'entrée (Q, K, V) dans des espaces de dimensions réduites, ce qui permet d'effectuer des calculs plus efficaces.

Une fois que chaque tête a effectué son calcul d'attention, les résultats sont concaténés pour former une représentation globale qui intègre des informations provenant de différentes perspectives. Ensuite, la matrice W^O, qui agit comme une matrice de transformation linéaire, intervient pour combiner cette représentation globale en la sortie finale du mécanisme d'attention multi-tête. La sortie finale désigne spécifiquement le vecteur ou la représentation numérique résultante qui capture de manière consolidée les informations importantes extraites par toutes les têtes d'attention. Cette étape est cruciale car W^O permet de pondérer judicieusement la contribution de chaque tête à cette sortie globale, assurant ainsi que le modèle exploite

pleinement les avantages de l'attention multi-tête tout en maintenant la cohérence et l'efficacité des prédictions ou des représentations finales.

La structure à têtes multiples de l'attention permet au modèle de capter efficacement les informations contextuelles en segmentant l'entrée en différentes parties. De plus, cette approche offre une capacité unique à corréler les mots et à saisir des informations contextuelles variées. Le modèle Transformer exploite cette capacité en utilisant des têtes d'attention indépendantes, ce qui lui permet de comprendre les entrées de manière approfondie et efficace. Ces caractéristiques clés ont permis au modèle Transformer de révolutionner de nombreux aspects du TALN.

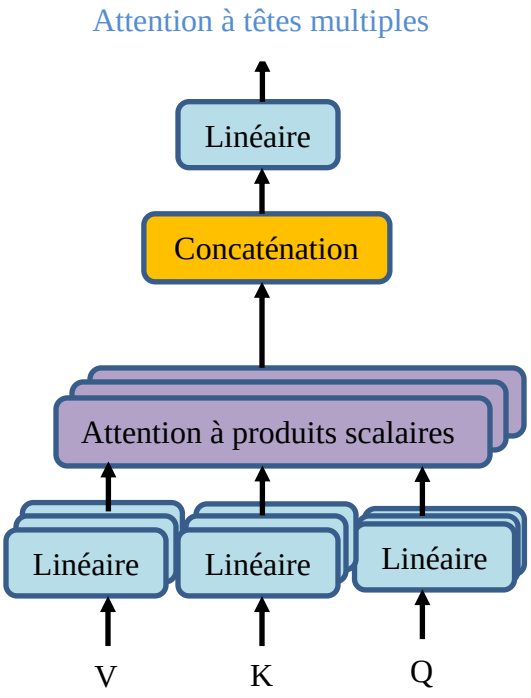

Figure 5 : Représentation de l'attention à tête multiples [34].

3.1.2.4 Les Transformers

La réduction de la charge de calcul séquentiel a longtemps été un problème majeur pour les applications de TALN [35]. Malgré les nombreuses solutions proposées au fil du temps pour atténuer la dépendance linéaire ou logarithmique dans le traitement des séquences, le TALN continue de souffrir de ce problème. Les modèles de Transformers offrent une architecture plus simple, sans couches convolutives ni récurrentes. Ce changement d'architecture a permis de résoudre ce problème en effectuant un nombre constant d'opérations grâce à l'attention positionnelle pondérée, qui peut être considérée comme une attention à têtes multiples, ainsi qu'aux intégrations positionnelles. Les modèles de Transformers surpassent les modèles existants avec un coût de formation réduit [34].

L'architecture du modèle Transformer, telle qu'illustrée dans la figure 5, repose sur une organisation en couches de multiples blocs Transformer [34]. Chaque couche de Transformer se compose d'une sous-couche d'auto-attention à plusieurs têtes et d'une sous-couche de réseau feedforward. La sous-couche feedforward, équipée de deux couches entièrement connectées, facilite l'extraction efficace de caractéristiques significatives. Quant à la couche d'auto-attention à plusieurs têtes, elle divise l'espace vectoriel en plusieurs sous-espaces appelés "têtes". Chaque tête réalise une opération d'auto-attention indépendante sur différentes parties des données d'entrée, permettant ainsi une compréhension contextuelle approfondie du texte et enrichissant sa représentation sémantique.

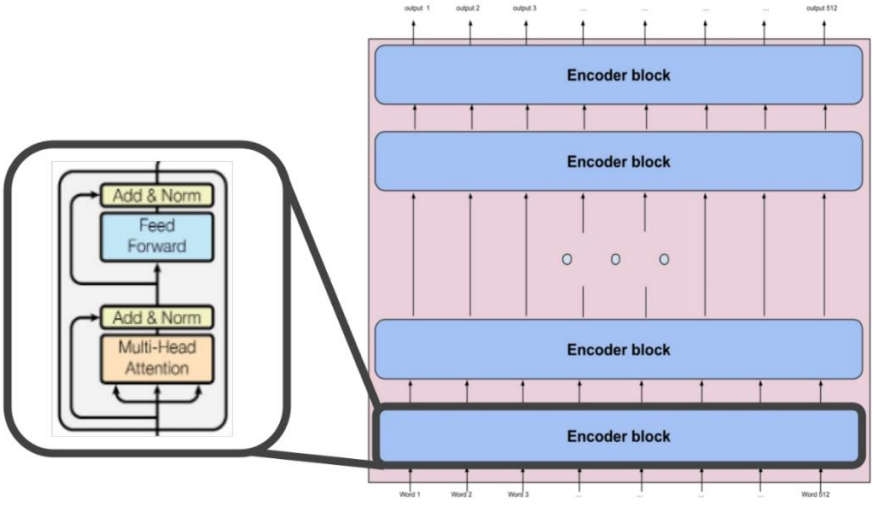

Figure 6 : Architecture d'un transformer [34].

3.1.3. Plongement de mots

Dans cette section, nous introduirons les représentations vectorielles de mots, un aspect fondamental du traitement automatique du langage naturel. Ensuite, nous approfondirons notre analyse en détaillant USE.

3.1.3.1 Introduction

En 2018, l'équipe Google AI a apporté une révolution dans le domaine du TALN en introduisant les Représentations de l'Encodeur Bidirectionnel à partir de Transformateurs (BERT). En raison de son approche hautement pragmatique et de ses performances supérieures, BERT est considéré comme ayant atteint les performances de pointe dans de nombreuses tâches de TALN [36].

Les plongements de mots sont un ensemble de techniques de modélisation linguistique et d'apprentissage de caractéristiques en traitement automatique du langage naturel où les mots ou expressions du vocabulaire sont associés à des vecteurs de nombres réels [36]. Les plongements de mots sont capables de capturer le contexte d'un mot dans un document, la similarité sémantique et syntaxique, la relation avec d'autres mots, etc. Les plongements de mots sont principalement utilisés comme caractéristiques d'entrée pour d'autres modèles construits pour des tâches personnalisées.

Le plongement de mots est une méthode populaire pour représenter les mots sous forme de vecteurs, permettant d'effectuer des opérations mathématiques sur les mots et de capturer leur sens sémantique. Chaque mot a un plongement ou vecteur unique, qui est simplement une liste de nombres pour chaque mot et sont généralement multidimensionnels, allant de 50 à 500 dimensions pour un bon modèle. Les plongements de mots capturent la signification du mot, et nous permet d'effectuer des opérations sur les mots telles que l'addition et la soustraction pour capturer des relations sémantiques. Enfin, des mots similaires ont des valeurs de plongements similaires, nous permettant d'identifier des mots liés dans un corpus.

Il existe plusieurs approches pour générer des plongements de mots, avec trois catégories : les approches indépendantes du contexte, les approches dépendantes du contexte et les approches des LLM, comme illustré dans la figure 7.

1. Les approches indépendantes du contexte, telles que Bag of Words [37], TF-IDF [38], GloVe [40], se concentrent sur la génération de plongements basés

uniquement sur la fréquence des mots dans un corpus, sans prendre en compte leur contexte.

2. Les approches dépendantes du contexte, telles que ELMo [41], et Word2Vec [39] génèrent des plongements en prenant en compte le contexte des mots, permettant une compréhension plus nuancée du sens des mots.
3. Les approches des LLM telles que Transformer-XL [43], BERT [40], GPT-2 [44], USE [42] utilisent de grands modèles pré-entraînés pour générer des plongements qui sont très dépendants du contexte et peuvent capturer des relations complexes entre les mots.

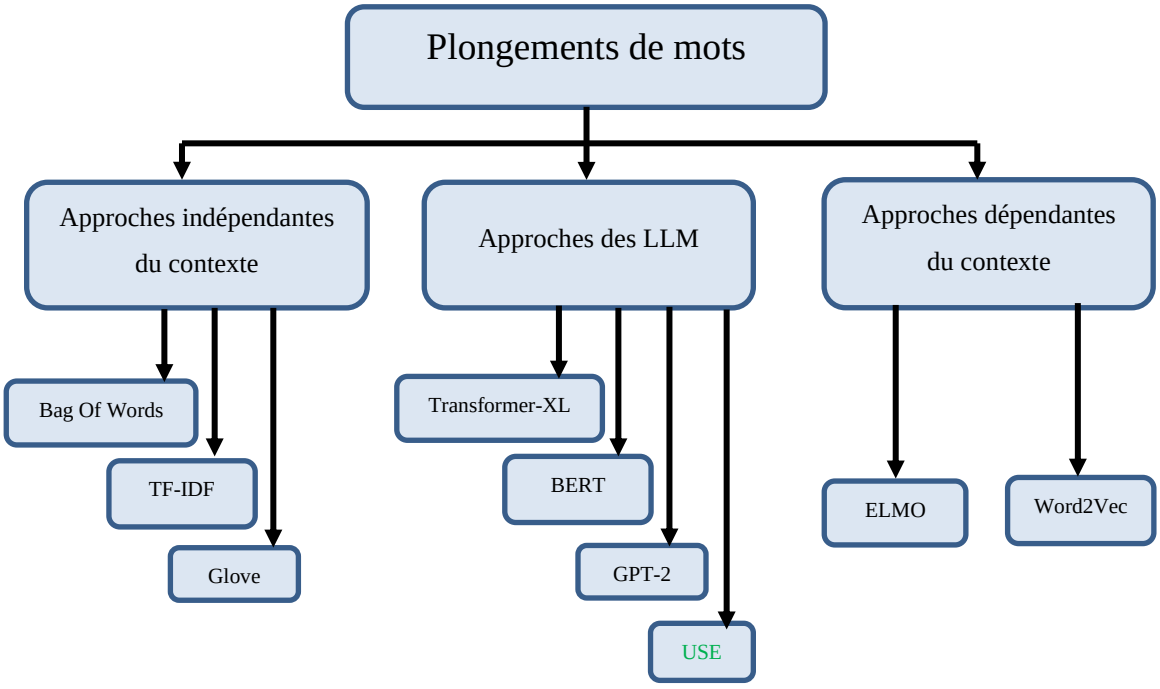

Figure 7 : Evolution des plongements de mots.

Dans notre étude, nous utiliserons USE pour générer des plongements de phrases. Cette approche prend en compte le contexte des phrases, ce qui permet une compréhension plus nuancée du sens du texte.

3.1.3.2 Plongement de texte avec USE (Universal Sentence Encoder)

Dans le domaine de plongement de mots et de phrases, USE a été largement reconnu comme une avancée majeure. Cette méthode novatrice, présentée par Cer et al. [42], s'appuie sur le puissant modèle de transfert de connaissances de Google. Pré-entraîné sur un vaste corpus de données provenant de diverses sources telles que Wikipedia et Common Crawl, USE utilise une architecture de transformer bidirectionnelle pour capturer les relations contextuelles et sémantiques entre les mots dans une phrase. Ce mécanisme permet de générer des représentations de phrases hautement informatives et riches en sémantique.

Le processus de fonctionnement de l'USE, tel qu'illustré dans la figure 8, peut être décomposé en cinq étapes :

1. **Tokenization** : La tokenization est le processus de division du texte en tokens, qui sont les unités de base utilisées pour former des phrases. Ces tokens peuvent être des mots individuels, des racines de mots, des caractères, ou même des éléments plus complexes comme des emojis ou des symboles de ponctuation. La tokenization crée une représentation structurée du texte, qui est ensuite utilisée pour l'analyse et le traitement ultérieurs.
2. **Plongement des tokens** : Chaque token est ensuite converti en un vecteur de nombres réels, également appelé plongement. Ces plongements sont des représentations numériques qui capturent le sens et le contexte de chaque token

dans le corpus de données sur lequel le modèle a été entraîné. Par exemple, les tokens qui ont des significations similaires ou qui apparaissent dans des contextes similaires auront des plongements similaires. Ce processus est généralement effectué à l'aide de réseaux de neurones spécialement conçus pour le plongement de mots, comme ceux utilisés dans les modèles de langage neuronaux.

3. **Agrégation des plongements :** Une fois que chaque token a été converti en un plongement, les plongements des tokens sont combinés pour former un plongement de la phrase entière. Cette agrégation se fait grâce à l'utilisation de réseaux de neurones pour combiner les plongements de manière plus complexe. L'objectif est de capturer la sémantique globale de la phrase à partir de ses composants individuels.
4. **Normalisation** : La représentation vectorielle obtenue après l'agrégation des plongements est souvent normalisée pour avoir une longueur unitaire, ce qui signifie que sa norme euclidienne est égale à un. Cela permet de rendre les vecteurs comparables et facilite les calculs de similarité entre les phrases.
5. **Sortie du plongement** : Enfin, la phrase est encodée en un vecteur de dimension 512. Cette dimension a été choisie de manière empirique pour capturer de manière efficace la sémantique des phrases dans un espace vectoriel dense. Ce vecteur peut alors servir de représentation de la phrase dans le processus de détection de spam, permettant ainsi au modèle de prendre en compte la sémantique sous-jacente lors de la classification des messages.

Un des avantages les plus marquants de l’USE réside dans sa capacité à capter les similitudes sémantiques entre des phrases similaires, même si elles sont formulées différemment. Cette caractéristique en fait un outil précieux pour des tâches de

classification de texte et de recherche de similarité. L'aspect le plus impressionnant de l'USE réside dans sa capacité à généraliser à des phrases qu'il n'a jamais rencontrées lors de la phase d'entraînement. Cela signifie qu'il est capable de comprendre des phrases inconnues et de les représenter dans un espace sémantique cohérent. Cette capacité de généralisation élargit considérablement son potentiel d'application dans des tâches de compréhension de texte complexes et variées.

Dans le domaine de la recherche en TALN, USE a été adopté et étendu dans de nombreuses études et applications. Des chercheurs l'ont utilisé pour des tâches variées telles que la classification de textes [46], la détection de similarité de phrases [47], la génération de résumés [48], et même pour améliorer la performance de modèles de compréhension du langage naturel tels que BERT [40].

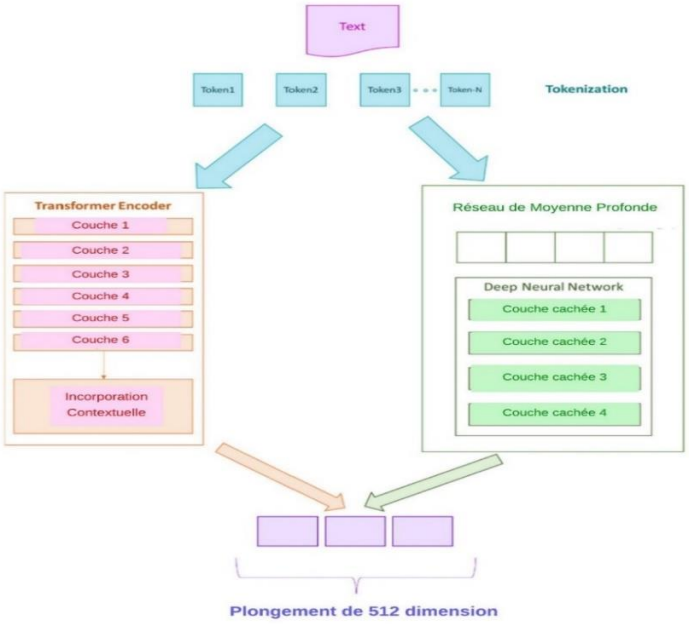

Figure 8 : Architecture de l'USE [42].

Les travaux de Cer et al. [42] ont donc ouvert la voie à de nouvelles perspectives dans le domaine du plongement des phrases. Depuis leur publication, USE a connu une adoption généralisée et continue d'attirer l'attention des chercheurs pour son potentiel et son efficacité remarquables dans la compréhension et la représentation du langage naturel.

3.1.4 Extraction de caractéristiques visuelles à partir d'images

Dans notre étude, nous avons utilisé l'extracteur de caractéristiques ResNet-50 [49], un réseau neuronal profond révolutionnaire dans le domaine de la vision par ordinateur. ResNet-50 est largement reconnu pour ses avancées significatives, notamment grâce à son architecture basée sur des couches résiduelles, comme illustré dans la figure 9. Cette architecture particulière lui confère des capacités exceptionnelles pour extraire des caractéristiques à partir d'images complexes. ResNet-50 peut capturer des motifs visuels détaillés grâce à ses 50 couches et à l'utilisation de blocs résiduels, ce qui facilite l'entraînement des réseaux profonds et améliore la précision globale.

En gelant les couches préalablement entraînées de ResNet-50, nous avons préservé les connaissances apprises lors de la phase d'entraînement initial, empêchant ainsi la modification de leurs poids pendant la phase de classification des spams. Pour l'extraction des caractéristiques, nous avons utilisé une approche comprenant une couche d'aplanissement suivie de deux couches denses, permettant ainsi une transformation efficace des informations visuelles en un format adapté à l'analyse.

L'utilisation de ResNet-50 est particulièrement pertinente pour la détection de spam car sa capacité à extraire des caractéristiques visuelles détaillées permet d'identifier des indices subtils et des anomalies dans les images, souvent présentes dans

les contenus de spam. Par la suite, une étape d'affinage a été appliquée à la dernière partie du modèle ResNet-50. Cette étape de fine-tuning a permis d'affiner les représentations apprises, en ajustant les poids des dix dernières couches ajoutées pour mieux s'adapter à la tâche spécifique de détection de spam. Le résultat de ce processus est un vecteur de sortie de 64 dimensions, consolidant ainsi les caractéristiques extraites pour une représentation plus robuste et discriminative des données visuelles.

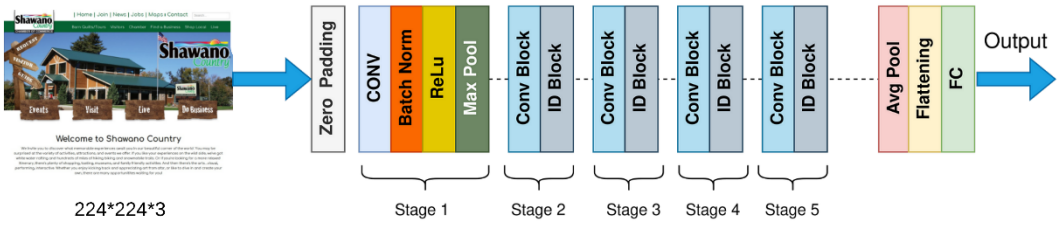

Figure 9 : Architecture de ResNet-50 [69].

3.1.5 Extraction de texte à partir d'images

En ce qui concerne l'extraction de texte à partir d'images, nous avons opté pour l'utilisation de l'OCR Tesseract, un moteur open source développé par Google capable de convertir des images de texte en texte éditable. Son architecture, telle que présentée dans la figure 10, repose sur une séquence de composants interdépendants œuvrant pour accomplir la tâche complexe de reconnaissance optique de caractères. Le prétraitement se consacre au nettoyage et à la préparation minutieuse de l'image source, effectuant des opérations telles que la normalisation de la luminosité, la conversion en niveaux de gris et la binarisation. La phase de segmentation analyse l'image pour

identifier les zones de texte potentielles, incluant la détection de lignes, de paragraphes et de mots. Au cœur du processus, la reconnaissance des caractères exploite un modèle de langage statistique soutenu par des modèles entraînés sur d'importants ensembles de données, renforçant ainsi la précision globale du système. Enfin, le post-traitement intervient pour améliorer la qualité des résultats, notamment par des corrections d'erreurs et la fusion de caractères.

Nous avons sélectionné PyTesseract pour son intégration fluide avec Tesseract et ses fonctionnalités avancées de reconnaissance optique de caractères. Ce système, compatible avec TensorFlow, exploite le langage de description de réseau VGSL. Pour l'identification d'images contenant un unique caractère, l'utilisation d'un CNN est courante, tandis que le traitement de texte à longueur variable requiert des LSTM. En mettant en œuvre des techniques telles que la binarisation et la segmentation des caractères, PyTesseract a considérablement rehaussé la précision de l'identification de texte dans les images. PyTesseract propose également des options de configuration avancées, permettant de définir la langue de reconnaissance et la résolution d'image. Ces fonctionnalités font de PyTesseract un outil polyvalent, idéal pour l'extraction de texte en vue de la détection de spam.

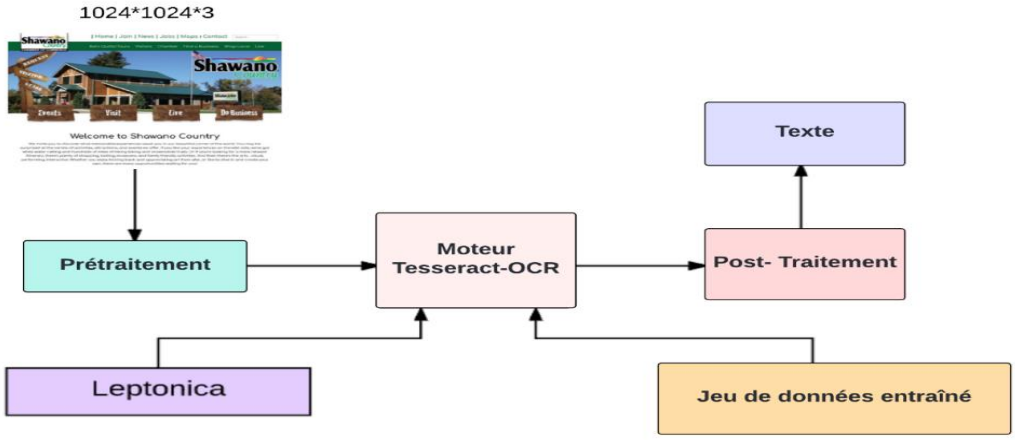

Figure 10 : Architecture du processus OCR Tesseract [50].

3.2 Détection de spam par analyse hiérarchique et multimodale

Dans cette section, nous présentons en détail notre modèle HMSD. Nous débutons par la préparation des données, puis nous détaillons la phase d'extraction des caractéristiques multimodales. Enfin, nous concluons en décrivant comment ces caractéristiques multimodales ont été fusionnées.

3.2.1 Préparation des données

La préparation des données se déroule en plusieurs phases distinctes : l'analyse de l'obfuscation, le nettoyage des données, l'encodage pour la catégorisation textuelle et la classification du spam.

3.2.1.1 Approche Analytique d'obfuscation dans la Classification de Spam

Dans le contexte de la classification de spam, les spammeurs utilisent des techniques d'obfuscation pour masquer ou dissimuler certaines informations dans le contenu des emails, afin de contourner les filtres anti-spam [26]. Pour détecter ces schémas d'obfuscation, nous avons sélectionné des motifs spécifiques adaptés à notre jeu de données, tels que des phrases liées à des thèmes couramment exploités dans les emails indésirables, comme l'argent, la perte de poids, les offres gratuites, les loteries, les médicaments, les cartes de crédit, les remboursements, les offres exclusives, les adresses e-mail, les URL, ainsi que des caractères Unicode suspects.

Notre approche consiste à parcourir les emails, extraire le corps de chaque email, et rechercher la présence de ces motifs d'obfuscation. Lorsqu'un de ces motifs est détecté, l'email est considéré comme obfusqué et ajouté à la liste des "obfuscated_emails". De plus, nous calculons le taux d'obfuscation pour chaque email en rapportant le nombre de motifs d'obfuscation détectés à la longueur totale du corps de l'email. Ce taux d'obfuscation, une mesure normalisée variant entre 0 et 1, reflète la proportion d'obfuscation présente dans l'email. Un taux d'obfuscation élevé indique une présence importante d'obfuscation, souvent associée à une gravité plus élevée.

Cette approche nous permet d'identifier les emails susceptibles d'être du spam en se basant sur la présence de schémas d'obfuscation. L'analyse du taux d'obfuscation nous permet d'évaluer l'ampleur de l'obfuscation dans notre jeu de données, contribuant ainsi à une classification plus précise et efficace du spam. En utilisant des motifs spécifiques adaptés à notre jeu de données, nous ciblons les schémas d'obfuscation les plus pertinents pour notre problématique, ce qui renforce la lutte contre les emails indésirables [26] [28]. De plus, cette approche offre plusieurs avantages. Premièrement, elle nous permet de mieux comprendre le degré d'obfuscation appliqué

à chaque exemple dans notre ensemble de données existant, en identifiant les modèles spécifiques d'obfuscation utilisés dans les emails indésirables. Deuxièmement, en intégrant la colonne de taux d'obfuscation dans notre ensemble de données, nous considérons l'obfuscation comme une caractéristique supplémentaire lors de l'entraînement du modèle, ce qui pourrait potentiellement améliorer la capacité du modèle à détecter les emails de spam avec des niveaux élevés d'obfuscation. Enfin, cette approche nous permet d'explorer l'impact de l'obfuscation sur la performance globale du modèle, en mettant en évidence les exemples les plus difficiles à détecter en raison de leur niveau d'obfuscation élevé.

Dans cette section, nous avons détaillé notre approche analytique d'obfuscation dans la détection de spam. Nous avons expliqué les motifs utilisés pour détecter l'obfuscation, ainsi que les techniques utilisées pour calculer le taux d'obfuscation. Nous avons également souligné les avantages de notre méthode, notamment de son potentiel à améliorer la détection de spam.

Les résultats de nos expériences approfondies seront discutés ultérieurement dans ce même chapitre, où nous évaluerons l'efficacité de notre approche par rapport aux approches traditionnelles.

3.2.1.2 Prétraitement pour la catégorisation de texte

Dans le cadre de la catégorisation de texte, le prétraitement revêt une importance capitale, contribuant ainsi à améliorer les performances des modèles de catégorisation et garantir la qualité des données utilisées en éliminant les éléments indésirables. Ces techniques ont permis de réduire le bruit, de simplifier le texte et de

focaliser l'attention sur les informations pertinentes, favorisant ainsi une meilleure généralisation de notre modèle de catégorisation.

Tout d'abord, nous avons normalisé les abréviations couramment utilisées dans les messages, ce qui permet de rendre le texte plus lisible et cohérent. Nous avons également supprimé les mentions d'utilisateurs, souvent présentes dans les réseaux sociaux, afin de préserver l'anonymat des individus.

Pour améliorer la généralisation et réduire les dimensions de l'espace des caractéristiques, nous avons remplacé les chiffres par des tokens. Cela permet de traiter tous les nombres de manière équivalente et de se concentrer sur d'autres aspects du texte.

Concernent les emojis, nous avons réalisé leurs transcriptions en mots clés spécifiques. Par exemple, nous avons remplacé les sourires par le terme "SMILE" et les expressions tristes par le terme "SADFACE" et les symboles spéciaux, tels que les cœurs représentés par "<3" par le terme "HEART". Cette transformation nous permet de capturer l'émotion véhiculée par ces symboles et de les intégrer dans l'analyse du texte.

Enfin, nous avons réalisé d'autres opérations de nettoyage, notamment la suppression des mots allongés et de la répétition de la ponctuation. Ces actions ont pour effet de réduire le bruit et de simplifier le texte, améliorant ainsi la capacité du modèle à extraire des informations pertinentes.

En combinant ces différentes techniques de prétraitement, nous avons réussi à obtenir des données textuelles de meilleure qualité pour la catégorisation des spams.

Ces étapes de préparation des données sont essentielles pour garantir des résultats précis et fiables lors de l'entraînement du modèle de catégorisation.

3.2.1.3 Prétraitement pour la classification de spam

Dans le processus de prétraitement en vue de la classification des spams, il est essentiel de traiter les adresses e-mail, les URL et les nombres de manière spécifique. Cette approche simplifie le texte en remplaçant ces éléments par des tokens distincts, réduisant ainsi la complexité et le bruit tout en préservant les informations pertinentes.

Chaque URL, en raison de sa variabilité, est transformée en un token unique. Cela permet de prendre en compte les caractéristiques uniques de chaque lien, préservant ainsi les détails spécifiques qui peuvent influencer la classification des spams. De manière similaire, chaque adresse e-mail reçoit son propre token pour conserver les informations relatives à l'expéditeur ou au destinataire du message. Cette approche garantit que ces informations essentielles sont préservées sans introduire de biais indésirables dans le modèle de classification. De plus, chaque nombre est également conservé sous forme de token distinct. Cette mesure préserve les informations numériques spécifiques qui peuvent jouer un rôle crucial dans la classification des spams, telles que des montants, des codes, des numéros de téléphone, ou d'autres détails pertinents. Finalement, les mots vides sont éliminés, ce qui permet de se concentrer uniquement sur les mots-clés significatifs. De plus, les caractères répétés sont supprimés pour éviter les redondances, et les mots qui ne figurent pas dans le dictionnaire anglais sont également éliminés.

L'ensemble de ces opérations de nettoyage et de prétraitement vise à maximiser la qualité des données utilisées pour la classification des spams. En préservant ce qui

est pertinent tout en éliminant les éléments inutiles, cette approche permet aux modèles de classification de détecter avec précision les caractéristiques distinctives associées aux URL, aux adresses e-mail et aux mots-clés du texte. Ainsi, elle contribue à une classification des spams plus efficace et fiable.

3.2.1.4 Plongement pour la catégorisation de texte et la classification de spam

La phase de plongement de mot joue un rôle central dans notre projet, et nous avons finalement utilisé Universal Sentence Encoder (USE) [42]. Le USE est renommé pour sa capacité à saisir les relations sémantiques dans le texte, ce qui en fait un choix idéal pour extraire des représentations riches et contextualisées à partir des données textuelles.

Le processus d'encodage comprend une étape de prétraitement où les données textuelles sont normalisées en utilisant le modèle USE pré-entraîné. Le USE utilise une architecture complexe qui apprend des représentations denses pour les phrases et les paragraphes, permettant une capture exhaustive des informations sémantiques et contextuelles. En utilisant le USE, nous avons obtenu des représentations textuelles enrichies et prêtes à être utilisées dans l'architecture de notre modèle. Ces représentations sont générées en prenant en compte les relations sémantiques entre les mots et en fournissant des vecteurs de haute qualité pour chaque élément d'entrée.

Nous avons intégré le "USE" dans notre modèle de classification de spam en utilisant les fonctionnalités de "TensorFlow Hub". Cela nous a permis de créer un modèle capable d'extraire des informations pertinentes pour des tâches spécifiques en utilisant les représentations générées par le "USE". Le "USE" encode le texte en vecteurs de haute dimension qui peuvent être utilisés pour la classification de texte, la

similarité sémantique, le regroupement et d'autres tâches liées au langage naturel. Le modèle est entraîné et optimisé pour des textes de longueur supérieure à un mot, tels que des phrases ou de courts paragraphes. Il est entraîné sur une variété de sources de données et de tâches dans le but de s'adapter dynamiquement à une grande variété de tâches de compréhension du langage naturel. L'entrée est un texte en Anglais de longueur variable et la sortie est un vecteur à 512 dimensions. Chaque dimension dans ce vecteur représente une caractéristique ou une information spécifique extraite du texte.

Notre projet a tiré profit des avantages du USE pour obtenir des représentations textuelles de haute qualité, prêtes à être utilisées dans notre modèle de traitement du langage naturel. Grâce à l'utilisation de cet encodeur, notre modèle est capable de saisir de manière précise les relations sémantiques et contextuelles dans le texte, améliorant ainsi la pertinence des résultats obtenus pour la classification de spam.

3.2.2 Extraction de caractéristiques multimodales

Nous avons développé notre modèle hiérarchique et multimodal de détection de spam HMSD en réponse aux limitations inhérentes des modèles actuels qui peinent à gérer efficacement diverses modalités de données. La notion de "hiérarchique" fait référence à la structure organisée et progressive du traitement des données, permettant une analyse en profondeur à différents niveaux. La notion de "multimodale" indique la capacité du modèle à traiter simultanément diverses modalités de données, telles que le texte, les images et les liens URL, pour une détection de spam plus efficace. Cette lacune est particulièrement évidente dans la gestion de plusieurs types d'informations dans le contexte de la détection de spam. Notre approche vise à combler cette lacune

en intégrant un modèle combiné capable de traiter simultanément ces modalités diverses.

La structure de notre modèle HMSD est présentée dans la Figure 11. Le processus peut être décomposé en cinq étapes distinctes, comme suit :

1	➢ Intégration des URL à l'aide du plongement de l'USE. ➢ Acheminement des intégrations vers le modèle Z, qui est chargé de l'entraînement pour générer quatre sorties correspondant aux catégories de phishing, d'URL malveillantes, de défiguration ou de catégories bénignes. ➢ Création d'un modèle intermédiaire et application du processus de gel pour préserver ses paramètres.
2	➢ Nettoyage du texte pour la catégorisation du texte. ➢ Intégration du texte à l'aide du plongement de l'USE. ➢ Acheminement des intégrations vers le modèle Y, responsable de la catégorisation basée sur des domaines tels que la politique, le sport, le divertissement, les affaires et la technologie. ➢ Création d'un modèle intermédiaire et application du processus de gel pour préserver ses paramètres.
3	➢ Prétraitement des images. ➢ Intégration des images dans le modèle ResNet 50. ➢ Application du processus de gel des couches du modèle ResNet 50 pour préserver ses connaissances antérieures. ➢ Extraction de caractéristiques visuelles des images en aplatissant la sortie du modèle ResNet 50, et des couches denses sont ajoutées pour affiner ces caractéristiques et les adapter spécifiquement à notre tâche, améliorant

	ainsi sa capacité à extraire des caractéristiques pertinentes des images pour la classification du spam. ➢ Extraction de texte des images à l'aide de PyTesseract.
4	➢ Nettoyage du texte pour la classification du spam. ➢ Nettoyage du texte extrait des images pour la classification du spam. ➢ Intégration du texte et du texte extrait des images en utilisant l'USE.
5	➢ Concaténation de trois entrées : le texte, la sortie du modèle intermédiaire de Z et la sortie du modèle intermédiaire de Y. ➢ Concaténation dans la couche de fusion entre les caractéristiques textuelles extraites des images et la sortie du modèle ResNet-50. ➢ Concaténation entre la sortie de X et la couche de fusion. Cela garantit que toutes les modalités, y compris le texte et la catégorie du texte, les URL, le texte des images et les caractéristiques visuelles des images, sont prises en compte lors de la classification du spam.

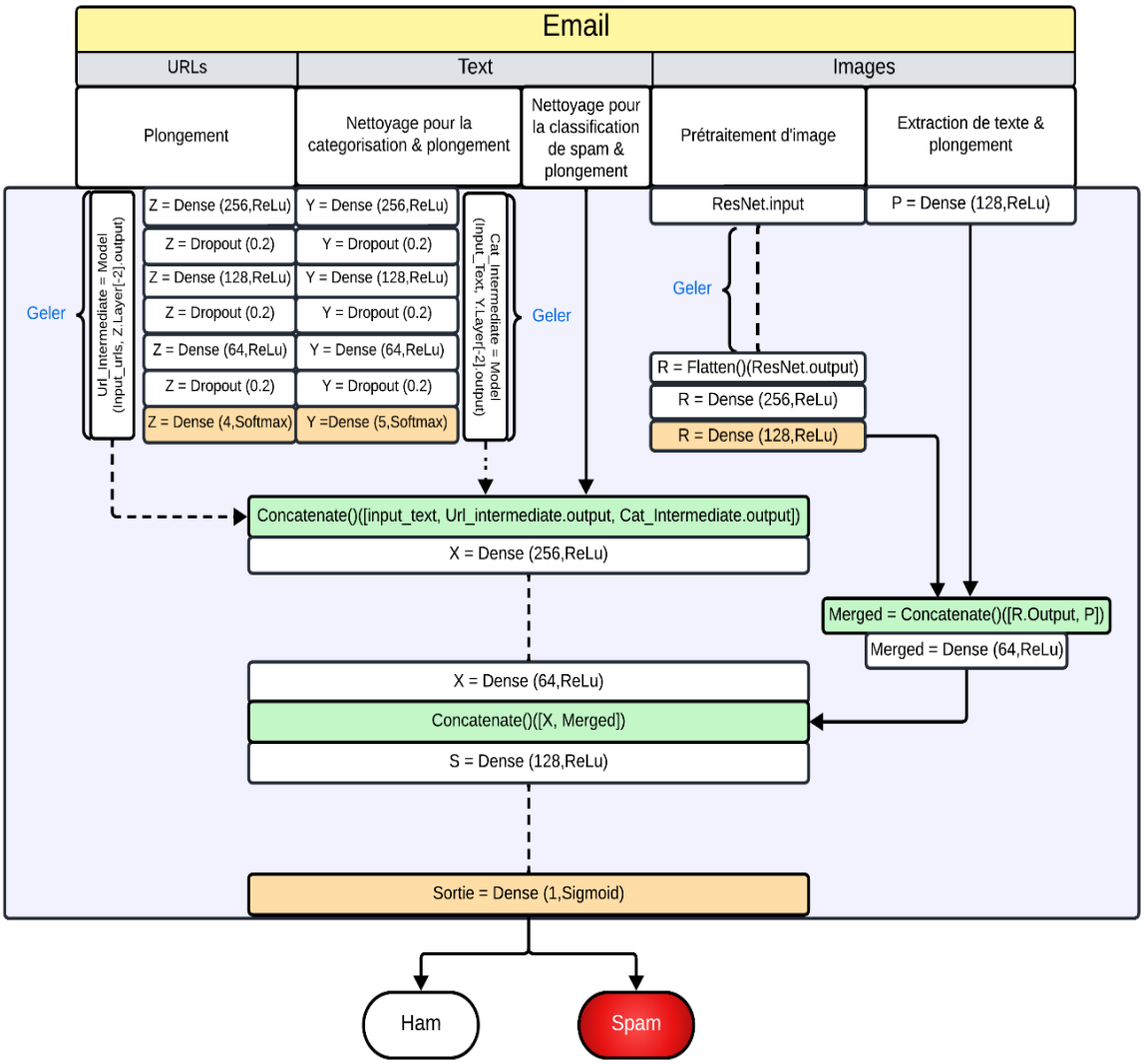

Figure 11 : Architecture du model HMSD.

L'architecture des deux encodeurs, URL Z et catégorisation de texte Y, est conçue avec des couches denses et des abandons successifs, générant des sorties activées par softmax. L'encodeur d'URL produit des sorties de dimension 4, tandis que

l'encodeur de catégorisation de texte génère des sorties de dimension 5, comme illustré dans la Figure 12. Dans les deux cas, la sortie de la couche avant-dernière, représentant une dimension de 64, est extraite. Ces représentations sont utilisées pour créer des modèles intermédiaires spécifiques à la classification des URLs et à la catégorisation de texte, respectivement.

Pendant l'entraînement, les couches des modèles intermédiaires sont gelées pour préserver les caractéristiques apprises. Les sorties de ces couches intermédiaires revêtent une importance particulière, surtout lors de la phase de concaténation des entrées dans la classification du spam. Cette approche garantit une considération précise des caractéristiques des URLs et de la catégorie de texte, assurant une gestion robuste des représentations sans altération, contribuant ainsi à une classification du spam fiable et efficace.

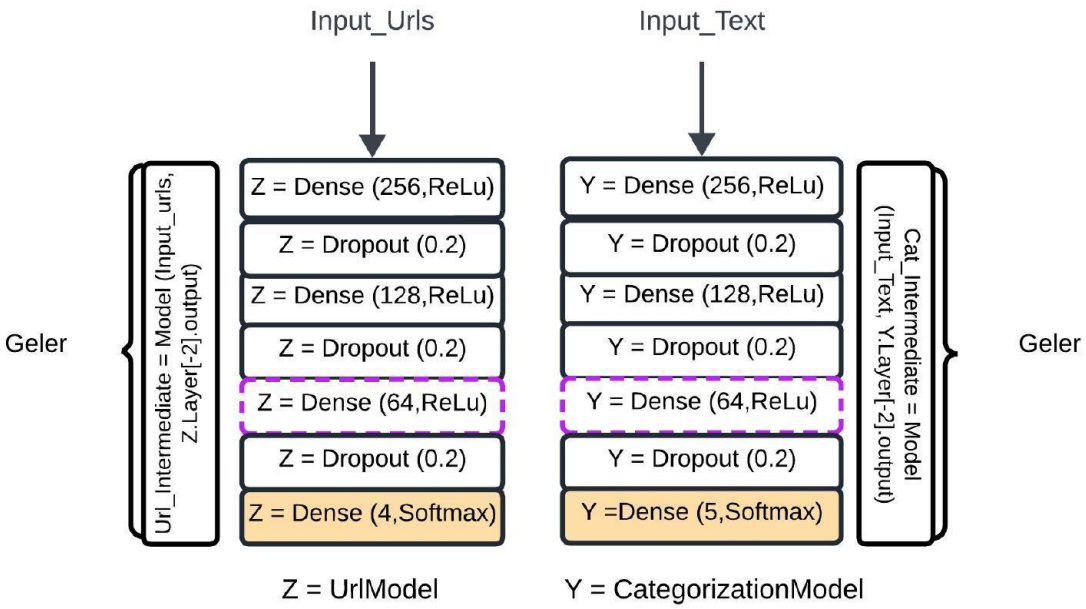

Figure 12 : Architecture de classification d'URLs et catégorisation de texte.

3.2.3 Fusion des caractéristiques multimodales

Le modèle HMSD représente une avancée innovante dans la détection de spams grâce à une architecture hiérarchique et multimodale. Notre approche novatrice consiste à concaténer diverses sources d'informations en tant qu'entrées, comprenant des données textuelles, des sorties du modèle URL intermédiaire et du modèle de catégorisation intermédiaire. Nous concaténons également le texte extrait des images et les caractéristiques d'une version affinée de ResNet-50 dans la couche Merged_Layer. Ces concaténations créent un espace d'information riche et diversifié, capitalisant sur la variété des formats de données.

L'évolution de l'architecture se poursuit avec des couches denses incorporant des mécanismes d'abandon. Une couche de concaténation, utilisant les sorties du modèle de classification du spam avec la couche Merged_Layer, est introduite avant la prise de décision finale, comme illustré dans la Figure 13, permettant au modèle de prendre en compte le texte des images ainsi que les caractéristiques pertinentes des images. Après cette concaténation, la sortie résultante est réintroduite dans l'entraînement du modèle, favorisant ainsi l'apprentissage des caractéristiques à la fois des images et du texte extrait des images. Cette approche hiérarchique et multimodale renforce significativement la capacité du modèle à traiter simultanément différentes sources d'informations.

La sortie du modèle est générée par une couche dense avec une activation sigmoïde, indiquant la probabilité qu'un élément soit un spam ou un ham. Cette conception garantit l'efficacité et la robustesse du modèle dans la classification du spam.

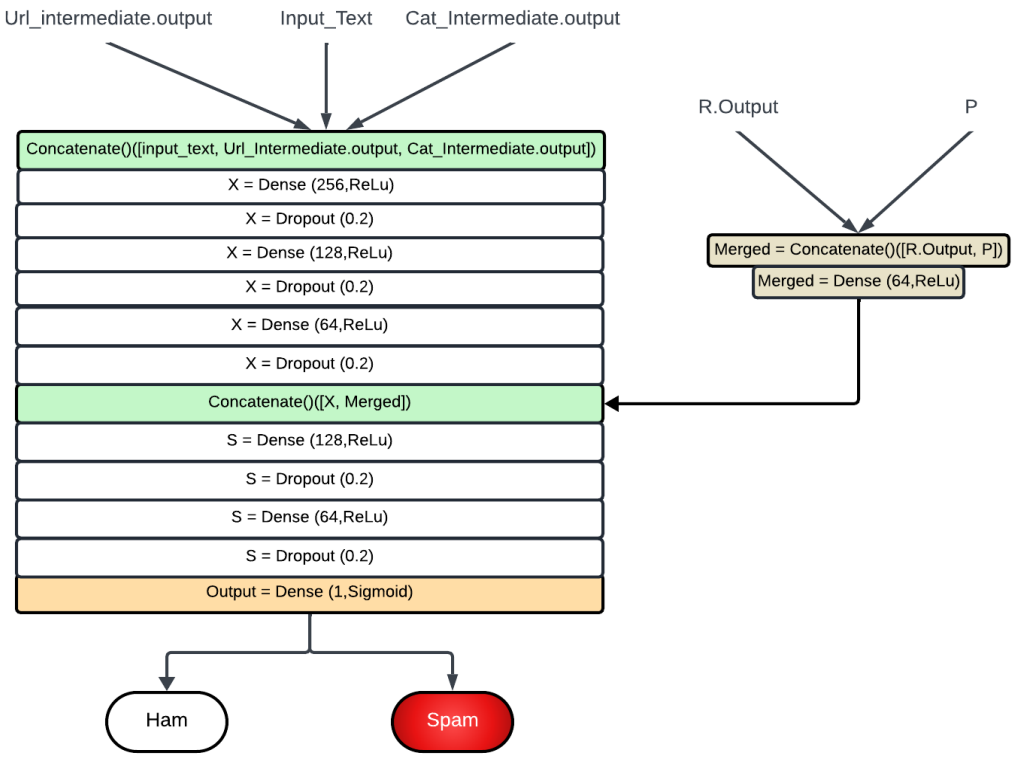

Figure 13 : Architecture final de classification de spam.

3.3 Expérimentation et évaluation

Dans cette section, nous abordons l'expérimentation et l'évaluation de notre modèle HMSD. Nous commençons par décrire la collecte de données, puis nous détaillons notre démarche d'expérimentation comparative, et enfin nous présentons les résultats de l'évaluation. Cette section est essentielle pour valider l'efficacité de notre méthodologie dans la résolution du problème de la détection de spam.

3.3.1 Jeu de données HMSD

Notre ensemble de données pour la détection de spam a été méticuleusement élaboré pour répondre aux besoins spécifiques de notre modèle HMSD. Tout d'abord, nous avons intégré le jeu de données des courriels Enron (https://www.kaggle.com/datasets/wcukierski/enron-email-dataset), Enron-Spam a été largement utilisé dans la recherche en détection de spam, notamment pour la comparaison de différentes méthodes de détection de spam. Ensuite, nous avons prédit les catégories de chaque courriel en utilisant le jeu de données des articles et catégories de la BBC bbc-text (https://www.kaggle.com/datasets/yufengdev/bbc-fulltext-and-category). Enfin, pour enrichir notre ensemble de données avec divers types d'URL, nous utilisons le jeu de données de malicious-urls-dataset (https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset).

Notre jeu de données, comme illustré dans la Figure 14, se compose de 8 colonnes :

1. **text**: Contient le texte des emails d'Enron.
2. **pics** : Chemins des images associées.
3. **labels** : Étiquettes pour le texte et les images : 1 pour spam et 0 pour ham.
4. **emailclean** : Texte extrait des images des emails.
5. **category** : Catégorie prédite pour chaque texte.
6. **obfuscation_rate** : Proportion d'obfuscation dans le texte.
7. **url** : Contient les divers URL de malicious-urls-dataset.
8. **type** : Type de chaque URL.

	text	pics	labels	emailclean	category	taux_obfuscation	url	type
0	royal replicas rr have you ever wanted to own ...	spam_2268.jpg	1	royal replicas rr have you ever wanted to own ...	tech	0.001443	manta.com/c/mm73jsg/don-ferguson	benign
1	your needed soffttwares at rock bottom prri ce...	spam_3786.jpg	1	your needed soffttwares at rock bottom prri ce...	tech	0.000264	digg.com/news/offbeat/Make_Your_Penis_Bigger_A...	benign
2	i ll have him mail you cv d be happy to speak ...	ham_2124.jpg	0	i ll have him mail you cv d be happy to speak ...	business	0.001742	bluesinthedigitalage.wordpress.com/2011/01/27/...	benign
3	hello welcome to medzonli moonbeam ne shop we ...	spam_4589.jpg	1	hello welcome to medzonli moonbeam ne shop we ...	business	0.001277	linkedin.com/in/heatherp	benign
4	viet vernon backstitch baritone pomposity work...	spam_1535.jpg	1	viet vernon backstitch baritone pomposity work...	entertainment	0.000000	recollectionbooks.com/siml/library/DonneSovver...	benign
5	good morning all below is an email from kevin ...	ham_2545.jpg	0	good morning all below is an email from kevin ...	politics	0.000713	bourbonlibrary.org/aahistory_thoroughbreds.htm	benign
6	hi everybody get ready for our lunch meeting t...	ham_1613.jpg	0	hi everybody get ready for our lunch meeting t...	politics	0.002398	http://www.natuurfotovanderhart.nl/foto-van-de...	defacement
7	subject thank you be on board vince have just ...	spam_904.jpg	1	subject thank you be on board vince have just ...	sport	0.000000	losangeles.craigslist.org/sgv/pts/2683488164.html	benign
8	to date my primary duties have been to oversee...	spam_3367.jpg	1	opt in email special offer unsubscribe me sear...	entertainment	0.000214	repetitionr.com/repetitions/proti-monopolu-lju...	benign
9	thee south letter explain explain taste page n...	spam_3515.jpg	1	thee south letter explain explain taste page n...	politics	0.001348	http://tuvanbatdongsan.vn/Tu-van-phong-thuy/in...	defacement

Figure 14 : Le jeu de données HMSD.

Dans notre étude, nous avons effectué un échantillonnage du jeu de données afin d'avoir une répartition équilibrée des catégories d'e-mails ainsi que des e-mails "spam" et "ham". En réduisant le nombre d'instances de chaque catégorie, nous nous sommes assurés de disposer d'un jeu de données plus gérable en termes de taille et de contraintes de ressources, telles que le CPU et l'espace mémoire.

Après avoir effectué cet échantillonnage, voici le nombre d'instances pour chaque catégorie :

- ✓ Catégorie : Entertainment - Nombre d'instances : 1556
- ✓ Catégorie : Business - Nombre d'instances : 1492
- ✓ Catégorie : Tech - Nombre d'instances : 1203
- ✓ Catégorie : Politics - Nombre d'instances : 1259
- ✓ Catégorie : Sport - Nombre d'instances : 1117

De plus, nous avons également réduit le nombre d'instances de chaque catégorie d'e-mails afin de gérer les contraintes de ressources, telles que le CPU et l'espace mémoire. Ainsi, nous avons obtenu un total de 3750 e-mails spam et 2985 e-mails ham.

Concernant les images, nous avons intégré le texte de nos e-mails dans celles-ci, les utilisant ainsi comme arrière-plan pour notre ensemble de données. Nous exploitons trois ensembles de données distincts afin de diversifier et enrichir notre ensemble de données de détection de spam. Le jeu de données Phish Iris (https://www.kaggle.com/datasets/saurabhshahane/phishiris) propose une variété d'images, servant de référence pour les études anti-hameçonnage basées sur la vision par ordinateur. Nous avons également intégré le jeu de données SMS Spam (https://www.kaggle.com/datasets/tapakah68/spam-text-messages-dataset), qui se concentre sur les messages texte indésirables couvrant des offres promotionnelles, des fraudes, de l'hameçonnage, et d'autres formes de communication non sollicitée. Enfin, le jeu de données Spam Image (https://www.kaggle.com/datasets/asifjamal123/spam-image-dataset) s'est avéré être une source précieuse pour les images naturelles et liées au spam. Dans la Figure 15, des exemples d'images spam et ham de notre ensemble de données peuvent être observés.

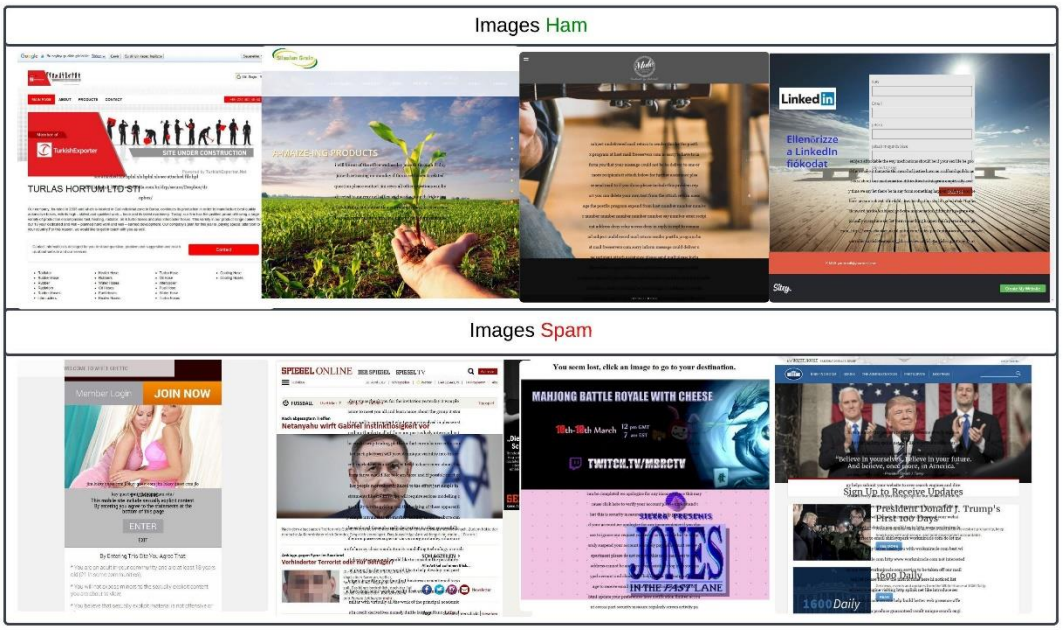

Figure 15 : Exemples d'images spam et ham.

3.3.2 Expérimentation comparative

Nous avons méthodiquement introduit des altérations dans notre ensemble de données pour évaluer la robustesse de notre modèle HMSD. Nous avons examiné cinq jeux de données différents, chacun soumis à des niveaux de perturbation croissants. Nous avons initialement remplacé 10% des données spam par des données ham, puis avons progressivement augmenté ce pourcentage jusqu'à atteindre 50%. Cette approche nous a permis d'évaluer de manière exhaustive l'efficacité de notre modèle face à divers scénarios de perturbation, tout en le comparant au modèle multimodal MMTD. Malgré ces variations, notre analyse a mis en évidence que le contenu textuel des images reste notre principale source d'information. Cette constatation souligne l'importance de notre approche dans la gestion des données, garantissant ainsi la qualité et la fiabilité de nos

analyses. En intégrant stratégiquement ce bruit, nous renforçons la capacité de notre modèle à généraliser efficacement aux données réelles et à résister aux tentatives de manipulation.

3.3.3 Évaluation

Pour évaluer notre modèle HMSD, nous avons appliqué un traitement différencié aux données textuelles, les préparant séparément pour la catégorisation et la classification de spam, tout en intégrant également les URL. Par la suite, une autre étape de prétraitement s'est avérée indispensable. Nous avons redimensionné et normalisé les données d'image à une dimension RGB de 224 × 224 pour l'intégration au modèle ResNet50, et à une dimension RGB de 1024*1024 pour l'extraction de texte avec Pytesseract.

Cette section propose une analyse comparative approfondie des résultats expérimentaux entre notre modèle HMSD et le modèle MMTD. Les configurations hyperparamétriques détaillées pour ces modèles sont présentées dans le Tableau 1. Il est essentiel de noter que la sélection des hyperparamètres dépend des contraintes matérielles, notamment de la capacité de la mémoire GPU, qui influence le choix de la taille du lot.

Définition des Termes Clés

- **Époque (Epoch)** : Une époque représente un cycle complet où le modèle traite chaque exemple de l'ensemble de données d'entraînement une fois. Pendant une époque, le modèle ajuste ses poids en fonction des erreurs constatées, ce qui permet d'améliorer progressivement sa précision sur les données d'entraînement.

- **Taille du Lot (Batch Size)** : La taille du lot désigne le nombre d'exemples d'entraînement utilisés pour calculer l'erreur et mettre à jour les poids du modèle en une seule fois. Une taille de lot plus grande fournit une estimation plus précise du gradient de la fonction de perte mais nécessite plus de mémoire, tandis qu'une taille de lot plus petite permet des mises à jour plus fréquentes des poids, ce qui peut aider à une convergence plus rapide.

- **Optimiseur (Optimizer)** : Un optimiseur est un algorithme qui ajuste les poids du modèle pour minimiser la fonction de perte en fonction des gradients calculés. Les optimiseurs couramment utilisés incluent Stochastic Gradient Descent (SGD), Adam, Adam W, RMSprop, et chacun a ses propres mécanismes et paramètres pour mettre à jour les poids. Par exemple, Adam est populaire pour sa capacité à ajuster le taux d'apprentissage pour chaque paramètre individuellement.
- **Taux d'Apprentissage (Learning Rate)** : Le taux d'apprentissage est un hyperparamètre qui détermine l'amplitude des mises à jour des poids du modèle à chaque itération. Un taux d'apprentissage trop élevé peut entraîner une convergence instable, tandis qu'un taux d'apprentissage trop bas peut ralentir considérablement l'entraînement, rendant difficile l'atteinte d'une solution optimale en temps raisonnable.

Modèle	Époque	Taille du Lot	Optimiseur	Taux d'Apprentissage
HMSD	20	32	Adam	0.001
MMTD	3	40	AdamW	0.0005

Tableau 1 : Les configurations hyperparamétriques.

Ensuite, nous avons évalué ces modèles en utilisant la matrice de confusion, comme illustré dans le tableau 3, ainsi que le rapport de classification, qui fournit des métriques essentielles telles que l'exactitude, la précision, le rappel et le score F1, présentées dans le tableau 2 ci-dessous :

Taux de perturbation	Modèle	Exactitude	Précision	Score-F1	Rappel
10%	HMSD	**0.925**	**0.934**	**0.927**	**0.940**
	MMTD	0.927	0.930	0.930	0.930
20%	HSMD	**0.911**	**0.904**	**0.913**	**0.922**
	MMTD	0.863	0.880	0.860	0.860
30%	HSMD	**0.921**	**0.909**	**0.923**	**0.938**
	MMTD	0.778	0.785	0.785	0.780
40%	HSMD	**0.898**	**0.893**	**0.901**	**0.909**
	MMTD	0.751	0.800	0.740	0.755
50%	HSMD	**0.930**	**0.922**	**0.932**	**0.941**
	MMTD	0.645	0.645	0.645	0.640

Tableau 2 : Analyse comparative du rapport de classification.

En examinant de près les résultats expérimentaux présentés dans le Tableau 2, une tendance significative émerge, mettant en évidence les performances remarquables du modèle HMSD. Bien que MMTD affiche également des performances solides, le modèle HMSD se distingue particulièrement par sa capacité à mieux généraliser aux

données perturbées. Cela suggère une plus grande robustesse et adaptabilité dans des contextes variés.

Taux de perturbation		**MMTD**		**HSMD**	
		Ham	Spam	Ham	Spam
10%	Ham	958	17	818	81
	Spam	127	888	55	870
20%	Ham	946	29	809	90
	Spam	245	770	72	853
30%	Ham	821	200	813	86
	Spam	240	729	57	868
40%	Ham	927	48	793	101
	Spam	447	568	85	845
50%	Ham	683	338	826	73
	Spam	368	601	54	871

Tableau 3 : Analyse des résultats des matrices de confusion.

L'analyse des matrices de confusion dans le Tableau 3 met en évidence la capacité remarquable du modèle HMSD à minimiser les erreurs de classification lors de perturbations. Cette observation est cohérente avec les performances globalement supérieures du modèle HMSD par rapport à MMTD.

Ces résultats soulignent la robustesse du modèle HMSD dans la détection de spams, suggérant que sa structure fondamentale capture efficacement des caractéristiques pertinentes, y compris les informations extraites à la fois des images et du texte des images. De plus, notre modèle excelle dans le traitement simultané de multiples caractéristiques, telles que le texte, les URL et la catégorisation, renforçant ainsi sa capacité à analyser de manière holistique les données multimodales.

En outre, l'analyse des résultats révèle également la capacité d'adaptation remarquable du modèle HMSD aux fluctuations des données, démontrant ainsi une flexibilité accrue face aux défis imprévus. Cette adaptabilité est cruciale dans un paysage numérique en constante évolution, où de nouveaux types de spams et des stratégies de dissimulation sophistiquées émergent régulièrement. La capacité du modèle HMSD à maintenir des performances élevées, même dans des conditions changeantes, souligne son potentiel à être déployé efficacement dans des environnements opérationnels réels de détection de spams.

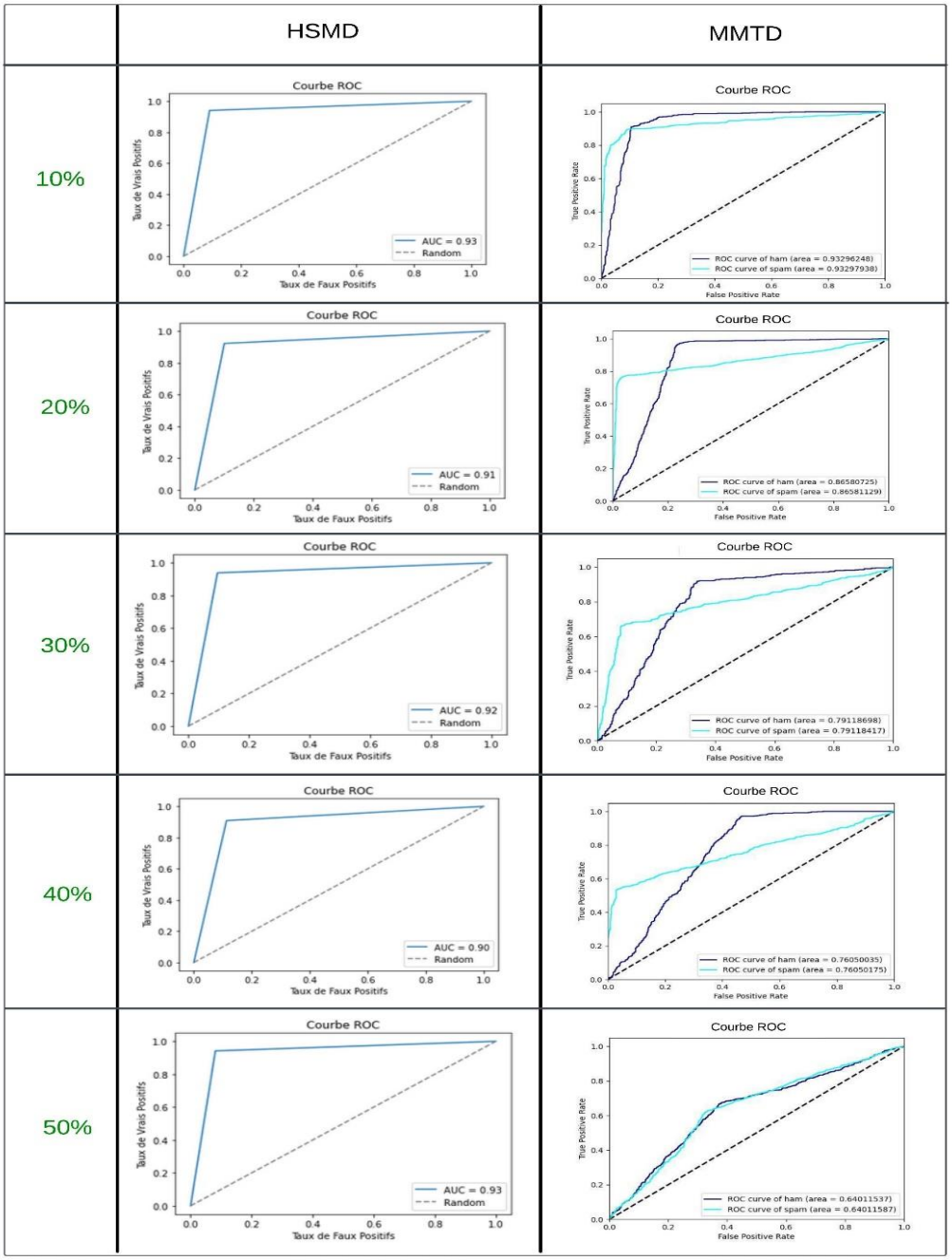

Figure 16 : Analyse Comparative des Courbes ROC.

L'évaluation comparative des courbes ROC entre les modèles HMSD et MMTD, comme illustré dans la figure 16, révèle des différences significatives dans leur aptitude à discriminer entre les classes positives et négatives à différents seuils de classification. Les courbes ROC associée au modèle HMSD démontrent une nette tendance à se rapprocher du coin supérieur gauche de l'espace ROC, suggérant une meilleure généralisation en condition de perturbation par rapport au modèle MMTD. Ces résultats mettent en évidence la capacité supérieure du modèle HMSD à minimiser à la fois les faux positifs et les faux négatifs.

Une observation récurrente est que, à mesure que le degré de perturbation augmente, le modèle MMTD rencontre des difficultés croissantes dans la classification. Cette limitation découle du fait que le modèle MMTD se repose principalement sur le texte et les caractéristiques de l'image, le rendant ainsi moins résilient aux perturbations. En revanche, le modèle HMSD semble mieux adapté à ces variations, grâce à son architecture plus complexe et à sa capacité à intégrer plusieurs modalités.

Cette analyse souligne de manière convaincante la pertinence du modèle HMSD dans le contexte de la détection de spams, mettant en lumière sa capacité à traiter efficacement plusieurs données multimodales et à fournir une précision supérieure dans des conditions de perturbation.

4. CONCLUSION

Cette recherche examine de manière exhaustive les défis et les avancées dans le domaine de la détection de spams, mettant en lumière la conception et l'évaluation du modèle HMSD. À travers une analyse rigoureuse des approches traditionnelles et des récentes avancées en modélisation multimodale, nous mettons en évidence les défis rencontrés par les méthodes uni-modales et l'importance croissante de fusionner diverses sources d'information pour une détection efficace des spams.

Le modèle HMSD développé dans cette étude a démontré sa capacité à surpasser les modèles existants dans des conditions perturbées, en intégrant de manière innovante l'analyse du texte et de sa catégorie, l'exploitation des URL, ainsi que des caractéristiques visuelles et textuelles des images. Cette approche améliore significativement la précision et la robustesse de la détection de spams. Les résultats obtenus indiquent clairement que le modèle HMSD offre des performances supérieures, notamment en termes de précision et de généralisation à de nouvelles données, par rapport au modèle multimodal MMTD.

En outre, cette recherche souligne l'importance cruciale de l'adaptabilité des modèles de détection de spams face à l'évolution constante des tactiques des spammeurs. En intégrant des techniques de prétraitement avancées, des méthodes d'apprentissage profond et une architecture multimodale flexible, le modèle HMSD représente une solution prometteuse pour contrer les spams dans divers contextes et environnements.

Cette étude contribue de manière significative à l'avancement de la recherche en détection de spams en proposant un modèle innovant et efficace. Elle met en évidence l'importance continue de l'innovation et de l'adaptabilité pour faire face aux menaces en ligne émergentes. Ces travaux ouvrent la voie à de futures recherches visant à améliorer les approches multimodales et à relever les défis en constante évolution dans ce domaine crucial, où convergent l'intelligence artificielle et la sécurité informatique.

Bibliographie

[1] A. F. Al-Qahtani and S. Cresci, "The COVID-19 scamdemic: A survey of phishing attacks and their countermeasures during COVID-19," *IET Information Security*, vol. 16, no. 5, pp. 324-345, Sep. 2022. doi: 10.1049/ise2.12073.

[2] Kaspersky Lab. (2021). "Spam and phishing in 2020.". Available : https://securelist.com/spam-and-phishing-in-2020/99872/. [Accessed: December 2023].

[3] M. Fossi and al. "Symantec internet security threat report trends for 2010." *Volume XVI*, 2011.

[4] D. Farhat and M. S. Awan, "A Brief Survey on Ransomware with the Perspective of Internet Security Threat Reports," *2021 9th International Symposium on Digital Forensics and Security (ISDFS)*, Elazig, Turkey, 2021, pp. 1-6, doi: 10.1109/ISDFS52919.2021.9486348.

[5] T. Chua, N. Fuhr, G. Grefenstette, K. Järvelin, and J. Peltonen, "User-Generated Content in Social Media", Dagstuhl Reports, Vol. 7, no. 7, pp. 110–154, January 2018, https://doi.org/10.4230/DagRep.7.7.110.

[6] J. R. Mendez, T. R. Cotos-Yanez, and D. Ruano-Ordas, "A new semantic-based feature selection method for spam filtering," *Applied Soft Computing*, vol. 76, pp. 89-104, 2019.

[7] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian approach to filtering junk e-mail," in AAAI Workshop on Learning for Text Categorization, Madison, Wisconsin, July 26–30, pp. 55-62, 1998.

[8] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam Filtering with Naive Bayes—Which Naive Bayes?" in *Proceedings of the CEAS 2006—Third Conference on Email and Anti-Spam*, Mountain View, CA, USA, 27-28 July 2006.

[9] Isra'a AbdulNabi, Qussai Yaseen, "Spam Email Detection Using Deep Learning Techniques," Procedia Computer Science, vol. 184, pp. 853-858, 2021. DOI:10.1016/j.procs.2021.03.107.

[10] T. Wu, S. Liu, J. Zhang, and Y. Xiang, "Twitter spam detection based on deep learning," in Proceedings of the australasian computer science week multiconference, pp. 1–8, 2017. https://doi.org/10.1145/3014812.3014815

[11] F. Yvon. "Une petite introduction au traitement automatique des langues naturelles." In : *Conference on Knowledge discovery and data mining*. p. 27-36, 2010.

[12] L.Yang, J. Zhai, W. Liu, X. Ji, H. Bai, G. Liu, and Y. Dai, "Detecting Word-Based Algorithmically Generated Domains Using Semantic Analysis", Symmetry, vol. 11, no. 2: 176. 2019, https://doi.org/10.3390/sym11020176

[13] R. Jalam, "Apprentissage automatique et catégorisation de textes multilingues", Thèse de doctorat en informatique, Université Lumière Lyon 2, Faculté de Sciences économiques et de gestion, Laboratoire ERIC, 4 juin 2003.

[14] F. Sebastiani. "A Tutorial on Automated Text Categorisation", in Analia Amandi and Alejandro Zunino (eds.), Proceedings, 1st Argentinian Symposium on Artificial Intelligence, Buenos Aires, AR, pp. 7-35, 1999.

[15] P. Bharati and A. Pramanik, "Deep learning techniques—R-CNN to mask R-CNN: a survey," *in Computational Intelligence in Pattern Recognition: Proceedings of CIPR'*2019', Springer, pp. 657-668, 2020.

[16] J. Koutník, K. Greff, F. Gomez, and J. Schmidhuber, "A Clockwork RNN," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, Beijing, China, vol. 5, pp. 3881–3889. 2014. http://proceedings.mlr.press/v32/koutnik14.pdf

[17] A. Graves, "Supervised sequence labelling with recurrent neural networks", Heidelberg, Springer Berlin, pp. 37-45, 2012.

[18] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *in IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1048-1054, 1999.

[19] M. L. Zhang and Z. H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," in Pattern Recognition, vol. 40, no. 7, pp. 2038-2048, 2007.

[20] M. Zaheer, G. Guruganesh, K. A. Dubey, et al., "Big bird: Transformers for longer sequences," in Advances in Neural Information Processing Systems, vol. 33, pp. 17283-17297, *December 6-12*, 2020. https://arxiv.org/abs/2007.14062v2.

[21] A. Chavda, "Image Spam Detection", Master's Projects, No. 543, 2017. https://scholarworks.sjsu.edu/etd_projects/543/. [Accessed: January 2024].

[22] N. Saidani, K. Adi, and M. S. Allili, "A semantic-based classification approach for an enhanced spam detection," Computers & Security, vol. 94, 101716, pp. 1-12, Jan 2020. https://doi.org/10.1016/j.cose.2020.101716.

[23] W. Hijawi, H. Faris, J. Alqatawna, A. M. Al-Zoubi, and I. Aljarah, "Improving email spam detection using content based feature engineering approach," in IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT 2017), Amman, Jordan, pp. 1-6, , 2017, doi: 10.1109/AEECT.2017.8257764.

[24] A. Gupta and R. Kaushal, "Improving spam detection in Online Social Networks," *2015 International Conference on Cognitive Computing and Information Processing (CCIP)*, Noida, India, pp. 1-6, , 2015, doi:10.1109/CCIP.2015.7100738.

[25] Kihal, M., Hamza, L. "Robust multimedia spam filtering based on visual, textual, and audio deep features and random forest." *Multimed Tools Appl* 82, 40819–40837, 2023. https://doi.org/10.1007/s11042-023-15170-x.

[26] P. Patil, R. Rane and M. Bhalekar, "Detecting spam and phishing mails using SVM and obfuscation URL detection algorithm," *2017 International Conference on Inventive Systems and Control (ICISC)*, Coimbatore, India, pp. 1-4, 19-20 January 2017, doi: 10.1109/ICISC.2017.8068633.

[27] C. Varol and H. M. T. Abdulhadi, "Comparision of String Matching Algorithms on Spam Email Detection," *in International Congress on Big Data, Deep Learning and*

Fighting Cyber Terrorism (IBIGDELFT), Ankara, Turkey, pp. 6-11, 03-04 December 2018, doi: 10.1109/IBIGDELFT.2018.8625317.

[28] D. E. Bakken, T. J. Palmer, A. A. Franz, D. M. Blough, R. Parameswaran, A. Genz, and M. Medidi, "Data Obfuscation: A New Class of Security Mechanism Providing Anonymity and Desensitization of Useable Data Sets," *IEEE Security & Privacy*, vol. 2, no 6, p. 34-41, 2004, doi: 10.1109/MSP.2004.97.

[29] T. Magallon, F. Béchet, and B. Favre, "Multimodal Image/Text Fusion Using Deep Neural Networks for Printed Document Classification," in Proceedings of the 15th Conference on Information Retrieval and Applications (CORIA), Rennes, France, May 2018.

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, no. 6088, pp. 533-536, 1986.

[31] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in Proc. 3rd Int. Conf. Learn. Represent. (ICLR), San diego, CA, USA, May, 7-9, 2015.

[32] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, et al., "Show, attend and tell: Neural image caption generation with visual attention," In: *International conference on machine learning*. PMLR. p. 2048-2057, 2015.

[33] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in : *Proceedings of the 2015 Conference on Empirical*

Methods in Natural Language Processing, pages 1412–1421, Lisbon, Portugal, 2015, doi:10.18653/v1/D15-1166.

[34] A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention is all you need", advanced in Neural Information Processing Systems NIPS, Long Beach, CA, USA, Vol.30, 2017.

[35] S. Hsu, Tida and V. Srinivas," Universal Spam Detection using Transfer Learning of BERT Model," in Proc. IEEE Conf. on Computer Communications, NY, USA, pp. 1-6, 02-05 May 2022.

[36] I. Rajapaksha, "BERT word embeddings deep dive," Medium, Oct. 11, 2020. Available: https://is-rajapaksha.medium.com/bert-word-embeddings-deep-dive-32f6214f02bf. [Accessed: May 2023].

[37] Z. S. Harris, "Distributional structure," Word, vol. 10, no. 2-3, pp. 146-162, 1954.

[38] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Information Processing & Management, vol. 24, no. 5, pp. 513-523, 1988, doi: https://doi.org/10.1016/0306-4573(88)90021-0.

[39] T. Mikolov, K. Chen, G. Corrado, and al., "Efficient estimation of word representations in vector space," 1st International Conference on Learning Representations, (ICLR 2013), Scottsdale, Arizona, USA, May 2-4, 2013.

[40] J. Devlin, M.-W. Chang, K. Lee, and al., "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proceedings of NAACL-HLT 2019, pp. 4171–4186, Minneapolis, Minnesota, June 2 - 7, 2019.

[41] M. Peters, M. Neumann, M. Iyyer, and al., "Deep Contextualized Word Representations," *in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics. 2018, doi:10.18653/v1/N18-1202.

[42] D. Cer, Y. Yang, S.-Y. Kong, and al., "Universal sentence encoder", Available : https://static.googleusercontent.com/media/research.google.com/fr//pubs/archive/46808.pdf. [Accessed: May 2024].

[43] Z. Dai, Z. Yang, Y. Yang, et al., "Transformer-XL: Attentive language models beyond a fixed-length context," *in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp 2978–2988, Florence, Italy, July 28 - August 2, 2019. https://aclanthology.org/P19-1285.pdf.

[44] A. Radford, J. Wu, R. Child, et al., "Language models are unsupervised multitask learners," in OpenAI, San Francisco, California, United States, 2019, https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf .

[45] Elasticsearch B.V., "What is Large Language Models," Available : https://www.elastic.co/fr/what-is/large-language-models. [Accessed: April 2024].

[46] M. Kerscher and S. Eger, "Vec2Sent: Probing Sentence Embeddings with Natural Language Generation," in *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain, Dec. 8-13, pp. 1729-1736, 2020. https://aclanthology.org/2020.coling-main.152.pdf.

[47] X. Sun, Y. Meng, X. Ao, et al., "Sentence similarity based on contexts," Transactions of the Association for Computational Linguistics, vol. 10, no. jan, pp. 573-588, May 2022, doi:10.1162/tacl_a_00477.

[48] D. Marcheggiani and I. Titov, "Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 1506-1515. Association for Computational Linguistics, 2017 doi:10.18653/v1/D17-1159

[49] L. Du, Z. Lu, and D. Li, "Broodstock breeding behaviour recognition based on Resnet50-LSTM with CBAM attention mechanism," *Computers and Electronics in Agriculture*, vol. 202 , Art. no. 107404, 2022, doi :10.1016/j.compag.2022.107404

[50] R. Smith, "An Overview of the Tesseract OCR Engine," in Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Brazil, pp. 629-633, 2007, doi:10.1109/ICDAR.2007.4376991.

[51] A. Mairey and M. Aouini, "PALM: Un modèle neuronal pour l'étiquetage morphosyntaxique des textes médiévaux," presented at the JADT 2020: 15èmes Journées Internationales d'Analyse statistique des Données Textuelles, June 2021.

[52] K. S. Prasanthi, T. Deepika, S. Anudeep, and M. S. Koushik, "An Efficient Email Spam Detection using Support Vector Machine," in International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, no. 2, December 2019. doi: 10.35940/ijitee.B9001.129219.

[53] S. Miao, X. Zhang, Y. Han, W. Sun, C. Liu, and S. Yin, "Random Forest Algorithm for the Relationship between Negative Air Ions and Environmental Factors in an Urban Park," Atmosphere, vol. 9, no. 12, p. 463, 2018. doi: 10.3390/atmos9120463.

[54] Z. Zhang, Z. Deng, W. Zhang, and L. Bu, "MMTD: A Multilingual and Multimodal Spam Detection Model Combining Text and Document Images," Applied Sciences, vol. 13, no. 21, p. 11783, 2023. doi:10.3390/app132111783.

[55] D. W. Hosmer Jr., S. Lemeshow, and R. X. Sturdivant, "Applied Logistic Regression". Hoboken, NJ: John Wiley & Sons, 2013, doi:10.1002/9781118548387.