RÉPUBLIQUE DU CAMEROUN

Paix – Travail – Patrie

****

UNIVERSITÉ DE DSCHANG

****

ECOLE DOCTORALE

REPUBLIC OF CAMEROON

Peace – Work– Fatherland

****

UNIVERSITY OF DSCHANG

****

POST GRADUATE SCHOOL

ÉCOLE DOCTORALE DES SCIENCES ET TECHNOLOGIES DE DSCHANG

Unité de Recherche en Informatique Fondamentale, Ingénierie et Application (URIFIA)

THÈME :

DÉTECTION AUTOMATIQUE DES DISCOURS HAINEUX POUR LA PROTECTION DES ENFANTS EN LIGNE

Mémoire soutenu publiquement en vue de l'obtention du diplôme de Master/ MSc en Informatique

Option : Réseaux et Services Distribués

Présenté par :

DASSI SIME ISMAELLE GLORIA

Matricule : CM-UDS-20SCI0842

Licence en informatique fondamentale

Sous la direction de :

Pr BOMGNI Alain Bertrand

Maitre de Conférences, Université de Dschang

ANNEE ACADEMIQUE 2024-2025

CERTIFICATION

Je soussigné, **DASSI SIME Ismaelle Gloria**, matricule CM-UDS-20SCI0842, étudiante en Master 2, option Reseaux et Services Distribués au Département de Mathématiques et Informatique, Faculté des Sciences, Université de Dschang, certifie que les présents travaux de recherche intitulés *« DÉTECTION AUTOMATIQUE DES DISCOURS HAINEUX POUR LA PROTECTION DES ENFANTS EN LIGNE »*, réalisés au sein de l'Unité de Recherche URIFIA de l'Université de Dschang sous la direction du **Pr BOMGNI Alain Bertrand**, Maître de Conférences de l'Université de Dschang, est original. Il n'a jamais été soumis antérieurement pour l'obtention d'un diplôme.

Candidat
DASSI SIME Ismaelle Gloria

.....

Superviseur
Pr BOMGNI Alain Bertrand
(Maître de Conférences, Université de Dschang)

.....

DÉDICACE

À mes parents SIME TCHAKOUNTE PIERRE *et* SIME TCHOUANZEU COLLETTE.

REMERCIEMENTS

Je rends grâce à Dieu Tout-Puissant, qui a produit en moi le vouloir et le faire selon Son bon plaisir.

J'exprime toute ma gratitude à mon encadreur, **Pr. BOMGNI Alain Bertrand**, pour sa patience, sa disponibilité et la richesse de ses conseils, qui ont grandement nourri ma réflexion et guidé l'orientation de ce travail.

En témoignage de ma profonde reconnaissance, j'adresse mes remerciements :

— À tous les membres du jury, pour l'honneur qu'ils me font en acceptant d'évaluer ce mémoire ;

— À tous les enseignants du département de Mathématiques et Informatique pour leur présence, leur encadrement et les connaissances qu'ils m'ont transmises ;

— Au Dr MFOGO Volviane Saphir, pour ses critiques éclairées, ses analyses approfondies, ses remarques détaillées, et sa capacité de m'emmener à sortir de ma zone de confort ;

— À mon père spirituel Mr NGANDEU Joseph pour ses prières, ses conseils et sa bienveillance envers moi ;

— À mon Pasteur Dr TEBONG Kenneth ainsi qu'à tous les membres de ma famille spirituelle, pour leur foi en moi, leurs prières et leur grand amour ;

— À mes parents, Mr SIME TCHAKOUNTE Pierre et Mme SIME TCHOUANZEU Collette, pour leurs soutiens inconditionnel, moral et financier qui ont rendu possible la poursuite de mes études et la réalisation de ce mémoire ;

— À mon grand frère Mr MOUAFO Samuel Aquilas et son épouse Mme MOUAFO ELANGA Claude, pour leur présence et leur soutien à tous les niveaux ;

— À mon grand frère Mr NGOZEU TCHAKOUNTE Prophette Nathan, pour son attention motivants, sa bienveillance et ses encouragements ;

— À mon petit frère Mr DJALEU TCHAKOUNTE Idriss, pour ses nombreux services, sa présence et ses encouragements constants ;

— À mon voisin, Mr KAMCHOUM LÉOPOLD Junior, pour ses observations constructives, ses remarques détaillées et sa bienveillance ;

— À Mme ETCHIEKE Jeanisse, pour son précieux partage d'expérience ;

— À mon ami, Mr TEYI KODJO Jerôme Sedowo, pour sa présence, ses observations constructives, ses remarques détaillées et sa bienveillance ;
— À toutes les personnes qui, de près ou de loin, ont contribué à la réussite de ce mémoire ;

TABLE DES MATIÈRES

LISTE DES ACRONYMES

ALBERT	A Lite BERT for Self-supervised Learning of Language Representations
AUC	Area Under the Curve (Surface sous la courbe ROC)
BERT	Bidirectional Encoder Representations from Transformers
BoW	Bag of Words (Sac de mots)
CBOW	Continuous Bag of Words
CSV	Comma-Separated Values (Fichier à valeurs séparées par virgule)
CNN	Convolutional Neural Network (Réseau de neurones convolutifs)
FastText	Fast Word Embeddings with Subword Information
GloVe	Global Vectors for Word Representation
GPT	Generative Pre-trained Transformer
IA	Intelligence Artificielle
LBFGS	Limited-memory Broyden–Fletcher–Goldfarb–Shanno (méthode d'optimisation)
LR	Logistic Regression (Régression logistique)
ML	Machine Learning (Apprentissage automatique)
MLP	Multilayer Perceptron (Perceptron multicouche)
NLP	Natural Language Processing (Traitement automatique du langage naturel)
PCA	Principal Component Analysis (Analyse en Composantes Principales)
RNA	Réseaux de Neurones Artificiels
RNN	Recurrent Neural Network (Réseau de neurones récurrents)
RoBERTa	Robustly Optimized BERT Pretraining Approach
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine (Machine à vecteurs de support)
TALN	Traitement Automatique du Langage Naturel
TF–IDF	Term Frequency–Inverse Document Frequency
Word2Vec	Word to Vector (Apprentissage de vecteurs de mots)
XGBoost	Extreme Gradient Boosting

LISTE DES TABLEAUX

TABLE DES FIGURES

RÉSUMÉ

Avec l'essor du numérique et l'accès généralisé aux plateformes en ligne, les enfants sont de plus en plus exposés à des contenus textuels potentiellement nuisibles, tels que les discours haineux, les insultes ou les propos discriminatoires. Ce mémoire s'attaque de front à cette problématique urgente de protection des enfants sur Internet, en proposant une solution innovante, accessible et performante pour la détection automatique de contenus textuels nuisibles via l'apprentissage automatique. Pour répondre à cet enjeu, nous avons mis en place un modèle finement optimisé, composé de plusieurs étapes clés. Tout d'abord, les textes extraits des jeux de données ont été nettoyés, puis transformés en représentations numériques à l'aide de la technique dite CountVectorizer, qui permet de convertir les mots en vecteurs exploitables par un algorithme. Ensuite, un rééquilibrage des classes a été réalisé grâce à la méthode d'oversampling, afin d'éviter que certaines catégories soient sur-représentées. Le cœur du système repose sur un modèle statistique appelé régression logistique multinomiale, dont les paramètres ont été ajustés avec soin pour améliorer la précision et maximiser les performances de classification. Ce modèle a été entraîné sur une partie des données, puis testé sur un ensemble séparé pour évaluer sa performance. Les résultats parlent d'eux-mêmes : notre système, bien que simple, atteint et surpasse même les modèles plus complexes avec un F1-score de 96% et une précision de 97% sur le jeu Davidson, et un F1-score de 88% avec une précision de 89% sur le jeu MetaHate. Ces performances, obtenues avec une architecture simple, montrent qu'un modèle transparent, facile à comprendre et à déployer peut être aussi efficace, voire plus, que des approches beaucoup plus complexes.

Mots-clés : Discours haineux, Apprentissage automatique, Sécurité en ligne, CountVectorizer, Régression logistique, Oversampling.

AUTOMATIC DETECTION OF HATE SPEECH FOR THE PROTECTION OF CHILDREN ONLINE

ABSTRACT

With the rapid rise of digital technologies and widespread access to online platforms, children are increasingly exposed to potentially harmful textual content, such as hate speech, insults, or discriminatory remarks. This thesis directly addresses this urgent issue of online child protection by proposing an innovative, accessible, and high-performing solution for the automatic detection of harmful textual content using machine learning. To tackle this challenge, we have implemented a finely optimized model composed of several key steps. First, the texts extracted from the datasets were cleaned and then converted into numerical representations using the CountVectorizer technique, which transforms words into vectors usable by an algorithm. Next, class imbalance was corrected using the *oversampling* technique, to prevent overrepresentation of certain categories. The core of the system is based on a statistical model known as multinomial logistic regression, whose parameters were carefully fine-tuned to improve accuracy and maximize classification performance. The model was trained on a portion of the data and then tested on a separate set to evaluate its effectiveness. The results speak for themselves : despite its simplicity, our system achieves and even outperforms more complex models, with an F1-score of 96% and a precision of 97% on the Davidson dataset, and an F1-score of 88% with a precision of 89% on the MetaHate dataset. These performances, obtained with a lightweight and interpretable architecture, demonstrate that a transparent and easy-to-deploy model can be just as effective if not more so than far more sophisticated approaches.

Keywords : Hate speech, Machine learning, Online safety, CountVectorizer, Logistic regression, Oversampling.

INTRODUCTION GÉNÉRALE

CONTENTS

Contexte du travail

L'essor rapide du numérique à travers le monde, et plus particulièrement en Afrique, a profondément modifié les usages sociaux, éducatifs et culturels des enfants. Grâce à la démocratisation des smartphones, à l'amélioration de la connectivité Internet et à l'introduction progressive des technologies dans les établissements scolaires, les plus jeunes accèdent désormais très tôt aux plateformes en ligne[10]. Au Cameroun, cette évolution s'observe dans toutes les couches de la société. Si cette transformation offre des opportunités d'apprentissage et d'ouverture, elle soulève également des inquiétudes majeures en matière de sécurité numérique, notamment concernant l'exposition précoce des enfants à des contenus inappropriés comme les discours haineux, les menaces ou les propos offensif. Ces contenus peuvent altérer leur développement psychologique et renforcer les vulnérabilités sociales.

Problématique

En dépit des efforts entrepris par certaines institutions publiques et privées[11, 12], la régulation des contenus dangereux demeure insuffisante. Les enfants peuvent ainsi être exposés à des messages violents ou discriminatoires sur les réseaux sociaux, sans barrière de protection adéquate. Ces réalités soulèvent des interrogations urgentes sur la capacité de nos systèmes à prévenir et filtrer les contenus dangereux. Comment protéger efficacement les enfants africains et camerounais en particulier contre les dérives du numérique tout en préservant leur droit à l'accès à l'information ? Les solutions classiques de filtrage, souvent conçues dans des contextes occidentaux, sont-elles adaptées aux réalités socioculturelles africaines ? Des travaux récents qui proposent des modèles de classification de texte, jusque-là, demeurent limités dans l'analyse de texte, comme solution à ce défi [13, 14].

Questions de recherche

Afin de répondre efficacement à la problématique posée, ce travail de recherche s'articule autour des questions suivantes :

- ⋆ Comment concevoir un modèle de classification textuelle capable de détecter efficacement des discours haineux, offensifs ou non haineux sur les réseaux sociaux ?
- ⋆ Quels types de représentations textuelles et quels algorithmes de classification offrent le meilleur compromis entre performance, simplicité et interprétabilité ?
- ⋆ Dans quelle mesure des techniques d'équilibrage des classes et d'optimisation des paramètres peuvent-elles améliorer les résultats du modèle ?

Objectifs du travail

Ce mémoire vise à concevoir un modèle automatique capable de détecter les discours haineux dans des textes en ligne, en particulier sur les réseaux sociaux. Ce travail se concentre exclusivement sur les contenus textuels, afin de proposer une solution légère, efficace et facilement interprétable. Le modèle repose sur deux piliers : une représentation vectorielle des textes via *CountVectorizer* et une classification multiclasse assurée par la **régression logistique**. L'objectif est d'attribuer automatiquement à chaque message l'une des catégories suivantes : *non haineux*, *haineux* ou *offensif*. Ce choix permet d'obtenir de bonnes performances tout en

conservant une faible complexité algorithmique, facilitant ainsi l'intégration du système dans des plateformes de modération en ligne.

Contributions

Ce travail apporte une solution concrète et facilement réutilisable pour détecter automatiquement les discours haineux en ligne, dans le but de mieux protéger les enfants sur Internet. Notre première contribution réside dans la construction d'un pipeline de classification efficace, combinant une technique simple de vectorisation textuelle (*CountVectorizer*) avec un modèle de régression logistique multiclasse, optimisé pour la reconnaissance de trois types de discours : *non haineux*, *offensif* et *haineux*. Nous avons ensuite intégré un prétraitement textuel robuste adapté au langage informel des réseaux sociaux, sans dépendance à des bibliothèques externes, ce qui améliore la portabilité du système. Par ailleurs, nous avons appliqué une stratégie d'**oversampling** pour corriger le déséquilibre des classes dans les jeux de données, renforçant ainsi la robustesse de l'apprentissage. Une attention particulière a été portée au réglage des hyperparamètres, notamment le paramètre de régularisation `C`, afin d'ajuster finement les performances du modèle. Enfin, nous avons validé notre approche sur deux jeux de données de référence, *Davidson (2017)* et *MetaHate (2024)*, en démontrant que notre solution atteint des scores élevés (F1-score de 96% et 0.88% respectivement), dépassant parfois des modèles plus complexes tout en restant plus simples à interpréter et déployer.

Structure du mémoire

Afin de mener à bien cette recherche, le mémoire est organisé en trois chapitres principaux, chacun abordant une facette complémentaire de notre travail. Tout d'abord, le chapitre I, intitulé fondements de l'apprentissage automatique pour la sécurité des enfants en ligne, introduit les notions fondamentales, en présentant les approches supervisées, la régression logistique ainsi que les principales techniques d'extraction de caractéristiques. Ensuite, le chapitre II, consacré à la revue de la littérature sur la sécurité des enfants en ligne via l'apprentissage automatique, examine les travaux existants portant sur la détection de discours haineux, en mettant l'accent sur les méthodes utilisées et les jeux de données disponibles. Enfin, le chapitre III, intitulé contribution à la sécurité des enfants en ligne par un modèle de classification multiclasse, détaille la solution proposée, en décrivant les

choix techniques, les expérimentations menées, les résultats obtenus, ainsi que les perspectives futures de ce travail.

I

CHAPITRE

NOTION D'APPRENTISSAGE AUTOMATIQUE

SOMMAIRE

I.1. Introduction

L'apprentissage automatique, et plus spécifiquement le traitement automatique du langage naturel (TALN), offre un cadre pertinent pour répondre à ces défis. Ce chapitre présente les fondements théoriques sur lesquels repose notre travail de détection automatique de discours haineux dans les textes. Nous débutons par une présentation des grandes catégories d'apprentissage automatique, notamment l'apprentissage supervisé, au cœur de notre démarche. Ensuite, nous abordons les techniques de représentation des textes, comme la vectorisation par sac de mots (*CountVectorizer*) et la pondération TF-IDF, qui permettent de transformer un corpus linguistique en données exploitables par des algorithmes.

Une attention particulière est accordée aux modèles de classification, en particulier à la régression logistique, choisie dans notre étude pour sa simplicité, sa robustesse et sa capacité à produire des résultats interprétables sur des jeux de données réels. Nous explorerons également les défis liés à la qualité des données textuelles, comme l'équilibrage des classes, la préservation du contexte sémantique, ou encore les biais linguistiques.

Enfin, nous replacerons ces notions dans le cadre de la protection des enfants en ligne, en mettant en lumière les implications pratiques de la détection automatique de contenus haineux, et en illustrant les apports potentiels de ce type de solution dans des environnements éducatifs, familiaux ou communautaires. Pour mieux situer notre approche dans le champ global de l'intelligence artificielle, la Figure 1 ci-dessous illustre les principaux sous-domaines de l'IA, incluant notamment le traitement du langage naturel (NLP), domaine auquel appartient notre contribution.

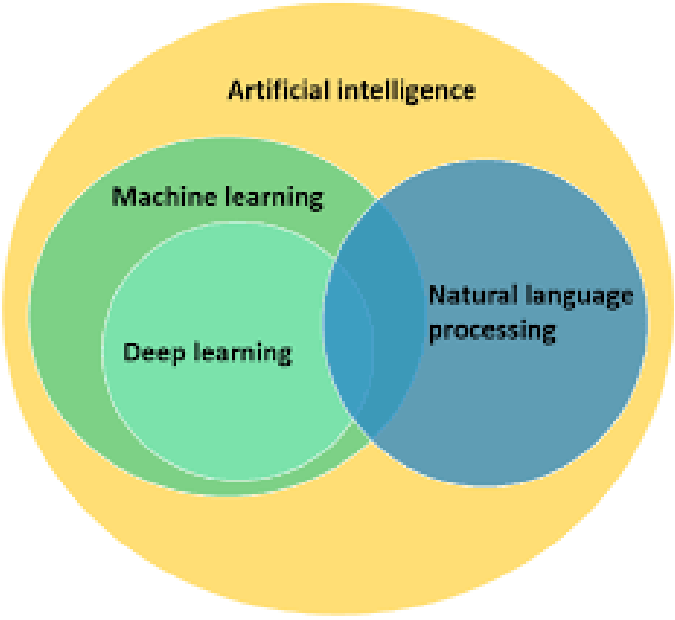

FIGURE 1 – *Sous-domaines de l'intelligence artificielle [1]*

I.2. Concepts de base en apprentissage automatique

L'apprentissage automatique désigne un ensemble de méthodes statistiques et d'algorithmes permettant à un système informatique d'apprendre à partir de données et de réaliser des prédictions ou des classifications sans être explicitement programmé pour chaque tâche spécifique. Dans cette section, nous présenterons les fondements de cette discipline, notamment les approches d'apprentissage supervisé, non supervisé, semi-supervisé et par renforcement ; l'accent sera mis sur l'apprentissage supervisé, qui constitue le socle de notre approche de détection des discours haineux. Nous aborderons également les étapes clés du processus de modélisation, depuis la représentation des données (vectorisation) jusqu'à l'entraînement et à l'évaluation des modèles, en détaillant les choix opérés dans notre

pipeline. L'attention sera portée principalement sur les méthodes de classification textuelle, en particulier la régression logistique multiclasse, qui a été retenue dans notre système pour sa simplicité, son efficacité et sa capacité à produire des résultats interprétables. Bien que des techniques plus complexes comme les réseaux de neurones profonds ou les SVM soient brièvement mentionnées, notre objectif est de montrer qu'un modèle plus sobre peut s'avérer pertinent dans un contexte appliqué à la protection des enfants en ligne.

Ce cadre théorique fournira ainsi les bases nécessaires à la compréhension des décisions techniques prises dans les chapitres suivants. La Figure 2 illustre les principaux paradigmes de l'apprentissage automatique, en distinguant clairement les trois grandes approches : l'apprentissage supervisé, non supervisé et par renforcement. On y voit également comment ces paradigmes s'articulent autour du concept central d'« apprentissage à partir de données », avec des objectifs différents selon le type de supervision disponible. Ce schéma met en lumière l'importance de choisir le paradigme le plus adapté à la tâche visée. Dans le cadre de notre mémoire, l'apprentissage supervisé est privilégié, car il permet d'exploiter les étiquettes présentes dans nos jeux de données (tweets annotés) pour entraîner un classificateur multiclasse.

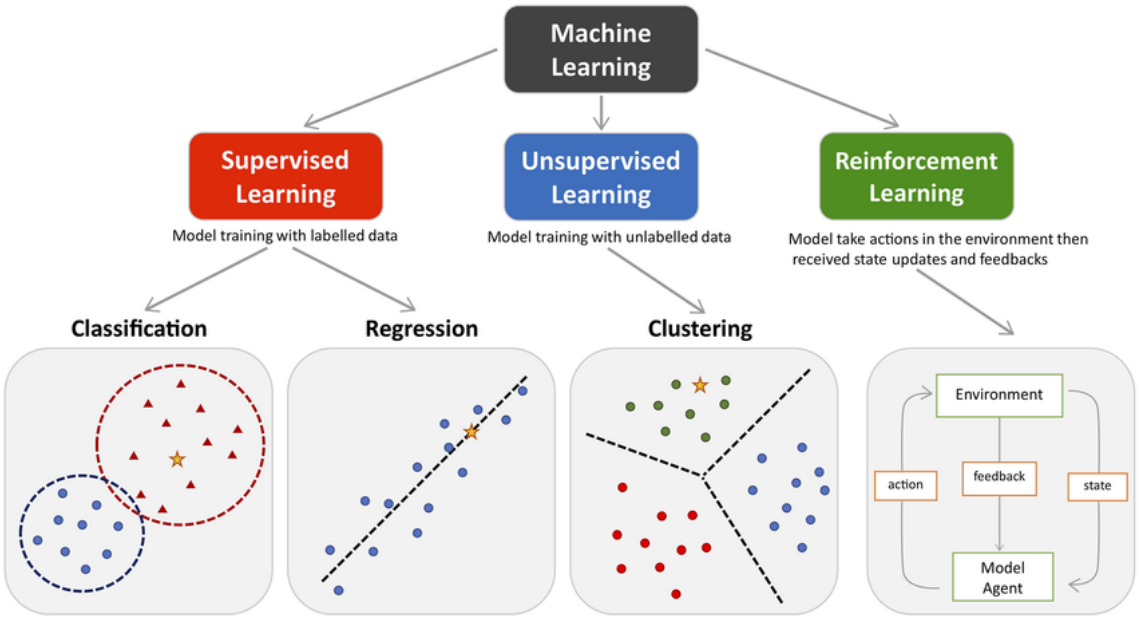

FIGURE 2 – *Apprentissage automatique [2]*

I.2.1. Type d'apprentissage

L'apprentissage automatique (*machine learning*) se décline en plusieurs paradigmes, chacun correspondant à un mode d'apprentissage spécifique en fonction

de la nature des données disponibles et de la tâche à accomplir. Ces paradigmes guident la manière dont un algorithme apprend à partir d'exemples, ajuste ses paramètres et produit des prédictions. Les quatre formes principales sont : l'apprentissage supervisé, l'apprentissage non supervisé, l'apprentissage semi-supervisé et l'apprentissage par renforcement.

I.2.1.1. Apprentissage supervisé

L'apprentissage supervisé est sans doute le paradigme le plus utilisé en apprentissage automatique, particulièrement adapté aux tâches de classification et de régression. Il repose sur la disponibilité d'un ensemble de données étiquetées, c'est-à-dire pour lesquelles chaque observation est associée à une sortie attendue (appelée aussi étiquette ou label) ce qui permet à l'algorithme d'apprendre une fonction qui associe les entrées aux sorties[15] afin de construire un modèle capable de prédire la sortie associée à de nouvelles observations encore inconnues.

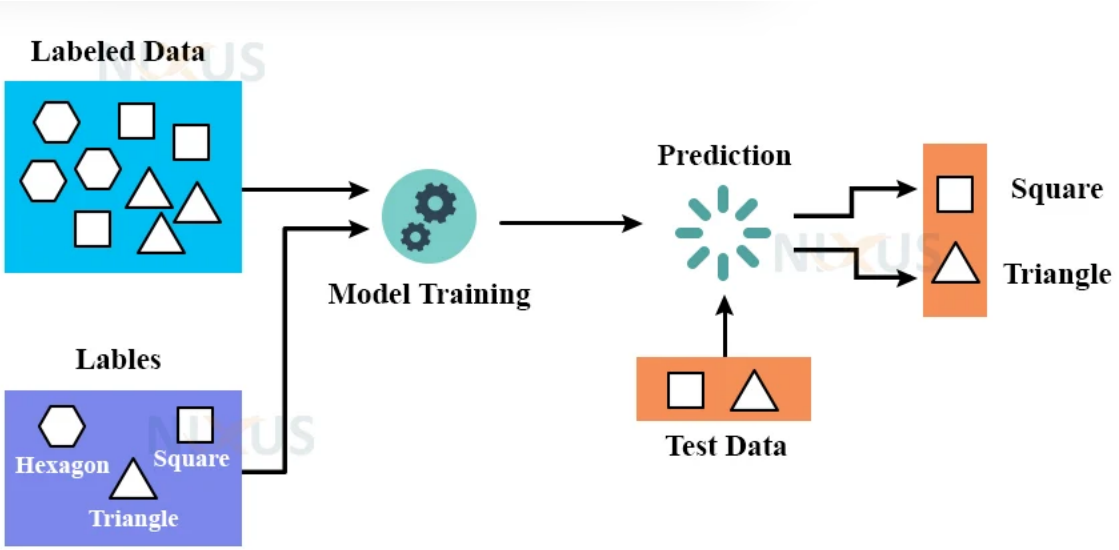

FIGURE 3 – *Illustration de l'apprentissage supervisé[3]*

Formellement, on considère un ensemble de données d'apprentissage constitué de n exemples :

$$\mathcal{D} = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_n, y_n)\}[16] \tag{1}$$

où chaque $\vec{x}_i \in \mathbb{R}^d$ représente un vecteur de caractéristiques décrivant un exemple (par exemple, un tweet vectorisé), et y_i est son étiquette appartenant à un espace de sortie Y. Le but de l'apprentissage supervisé est d'estimer une fonction $f : \mathbb{R}^d \to Y$ telle que :

$$y_i = f(\vec{x}_i) + \varepsilon_i, [16] \tag{2}$$

où ε_i est un terme d'erreur aléatoire supposé suivre une distribution centrée réduite [16].

La Figure 3 illustre clairement ce principe : à partir d'un ensemble d'apprentissage constitué de couples (entrée, sortie), le modèle est entraîné à minimiser l'erreur entre les prédictions et les sorties réelles. Une fois entraîné, ce modèle peut être utilisé pour effectuer des prédictions sur des données nouvelles et non étiquetées. Ce schéma reflète le processus général de tout système supervisé, en mettant en évidence les étapes clés : apprentissage, généralisation et prédiction. Ce paradigme couvre principalement deux types de tâches comme le montre la Figure 4 :

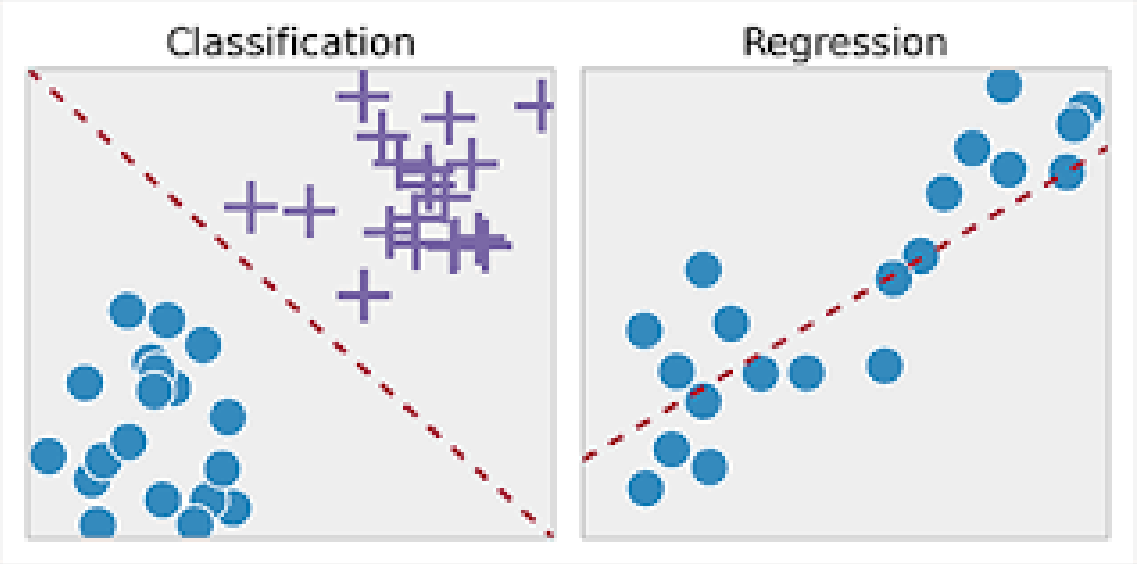

FIGURE 4 – *Différence entre la régression et la classification [3]*

- ⋆ **La régression** : l'objectif est ici de prédire une variable numérique continue. Le modèle apprend à estimer une fonction f continue telle que $f(\vec{x}) \approx y$. Un exemple classique est la prédiction du prix d'un bien immobilier à partir de ses caractéristiques comme la surface, le nombre de pièces ou l'emplacement.
- ⋆ **La classification** : le but est d'assigner une observation à une ou plusieurs catégories. Par exemple, dans le cadre de notre mémoire, il s'agit d'assigner un tweet à l'une des classes suivantes : *non haineux*, *haineux*, ou *offensif*. Le modèle prédit alors une variable discrète $y \in \{0, 1, 2\}$.

La performance des modèles supervisés est généralement mesurée à l'aide d'indicateurs tels que l'exactitude (accuracy), la précision, le rappel, ou encore le score F_1

. Il existe plusieurs algorithmes d'apprentissage supervisé utilisés pour résoudre

une variété de problèmes en science des données et en intelligence artificielle tels que présenté dans le Tableau I. Ces algorithmes permettent de modéliser des relations entre variables, en vue de réaliser des prédictions ou des classifications.

TABLE I – *Résumé comparatif de quelques algorithmes d'apprentissage supervisé*

Algorithme	**Description**	**Type**	**Avantages**
Régression linéaire[17]	Modélise une équation linéaire aux données observées, permettant ainsi de prédire et d'interpréter les effets des variables prédictives entre les variables indépendantes et dépendante.	Régression	Simple, rapide, interprétable, Peu de ressources, Fonctionne bien pour des relations linéaires.
Arbres de décision[18]	Les arbres de décision sont des modèles d'apprentissage supervisé qui utilisent une structure arborescente de règles conditionnelles pour classer des données ou prédire des valeurs cible à partir de caractéristiques numériques.	Classification/ Régression	Interprétable, non linéaire, Robuste vis-à-vis des valeurs aberrantes et des données manquantes, Préparation minimale des données.
Forêt aléatoire [19]	Ensemble d'arbres de décision construits sur des sous-échantillons, avec agrégation.	Classification/ Régression	Robuste, réduit la variance, efficace.
SVM (Support Vector Machine) [20]	Recherche une frontière optimale en projetant les données dans un espace de plus grande dimension.	Classification/ Régression	Performant pour petits jeux de données et marges claires.
Réseaux de neurones [21]	Architecture inspirée du cerveau humain, capable d'apprendre des représentations complexes.	Classification/ Régression	Très puissant, adapté aux grandes bases de données.

F. Régression logistique

La régression logistique : est un modèle probabiliste de classification, initialement conçu pour les problèmes binaires et étendu à la classification multiclasse (on parle alors de softmax regression ou logit multinomial) ; elle est particulièrement adaptée aux situations où la variable cible est binaire, c'est-à-dire qu'elle prend uniquement deux valeurs possibles (par exemple, succès/échec). Elle per-

met de modéliser la probabilité d'appartenance à une classe donnée à partir d'un ensemble de variables explicatives [22]

F.1 Principes et formulation

La régression logistique est un algorithme d'apprentissage supervisé utilisé pour résoudre des problèmes de classification. Contrairement à son nom, elle n'est pas conçue pour les tâches de régression (au sens prédictif continu), mais bien pour prédire l'appartenance d'une observation à une ou plusieurs classes. Ce modèle repose sur une fonction mathématique appelée **fonction sigmoïde** (ou logit), qui permet de transformer une combinaison linéaire des variables d'entrée en une probabilité comprise entre 0 et 1. Pour un problème de classification binaire, la probabilité qu'une observation x appartienne à la classe 1 s'écrit :

$$P(y = 1 \mid \vec{x}) = \frac{1}{1 + e^{-\theta^\top \vec{x}}} [16] \tag{3}$$

où :

— $\vec{x}$ est le vecteur représentant les caractéristiques de l'entrée (par exemple, les mots vectorisés dans un tweet),

— θ est le vecteur des coefficients appris par le modèle,

— $\theta^\top \vec{x}$ est le produit scalaire entre les poids et les variables d'entrée.

Ce modèle renvoie une probabilité, et une règle de décision est appliquée pour attribuer une classe. Par exemple, si $P(y = 1 \mid \vec{x}) > 0.5$, l'instance est classée dans la classe 1.

Dans notre cas, la tâche ne se limite pas à une simple classification binaire, mais implique plusieurs classes (non haineux, haineux, offensif). Pour cela, la régression logistique est généralisée via la **fonction softmax**, qui permet de gérer des sorties multiclasse en calculant une probabilité pour chaque classe possible :

$$P(y = j \mid \vec{x}) = \frac{e^{\theta_j^\top \vec{x}}}{\sum_{k=1}^{K} e^{\theta_k^\top \vec{x}}} [16] \tag{4}$$

où :

— K est le nombre total de classes,

— θ_j est le vecteur des paramètres associé à la classe j,

— la somme au dénominateur garantit que les probabilités sur toutes les classes s'additionnent à 1.

Cette formulation probabiliste offre un avantage majeur : elle permet non seulement de prédire la classe la plus probable, mais aussi d'obtenir une mesure de

confiance dans la prédiction, ce qui est particulièrement utile pour les systèmes de filtrage automatique de contenu sensible.

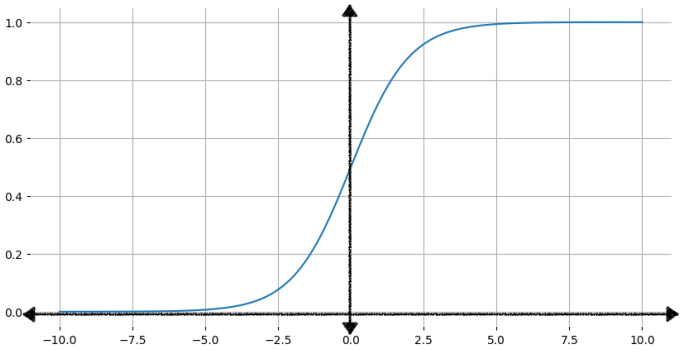

FIGURE 5 – *Illustration des fonctions sigmoïde[4].*

F.2 Apprentissage des paramètres

Une fois la structure de la régression logistique définie, il est essentiel de déterminer les valeurs optimales des paramètres (appelés aussi *poids*) qui permettront au modèle de faire des prédictions précises. Ces paramètres sont notés θ dans les équations. Un point fondamental à considérer est le mécanisme par lequel le modèle apprend les paramètres. Le processus repose sur un principe essentiel : fournir au modèle un ensemble d'exemples annotés, appelés *données d'apprentissage*. Pour chaque exemple, on connaît à la fois les informations d'entrée (par exemple, le texte d'un tweet vectorisé) et la bonne réponse (la classe à laquelle il appartient : haineux, offensif ou non-haineux).

Le modèle compare alors sa prédiction avec la vraie réponse. L'écart entre les deux est mesuré à l'aide d'une fonction appelée **fonction de coût** (ou fonction de perte). Dans la régression logistique, on utilise généralement une fonction appelée *entropie croisée* (*cross-entropy* en anglais).

Définition de la fonction de coût pour le cas multiclasse :

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{K} 1(y_i = j)\cdot \log P(y_i = j \mid \vec{x}_i)[5] \tag{5}$$

où :

— n est le nombre total d'exemples d'apprentissage,
— K est le nombre de classes (ici, 3),
— $1(y_i = j)$ est une fonction qui vaut 1 si l'étiquette réelle est j, 0 sinon,
— $P(y_i = j \mid \vec{x}_i)$ est la probabilité prédite par le modèle pour la classe j sur l'exemple i.

Cette fonction mesure à quel point les prédictions du modèle sont proches des vraies réponses. Plus les prédictions sont mauvaises, plus la valeur de $J(\theta)$ est élevée. Le but est donc de **minimiser** cette fonction de coût.

Une question centrale est celle de la réduction de l'erreur du modèle. Pour ce faire, on cherche à identifier les paramètres optimaux θ, on utilise une technique d'optimisation appelée la **descente de gradient**.

— Le principe est de modifier petit à petit les valeurs de θ pour que la fonction de coût $J(\theta)$ diminue.
— À chaque étape, on calcule la pente (ou le *gradient*) de la fonction de coût, qui indique la direction dans laquelle il faut aller pour réduire l'erreur.
— Le modèle ajuste alors ses paramètres dans le sens opposé à cette pente, à l'aide d'un pas d'apprentissage appelé `learning rate`.

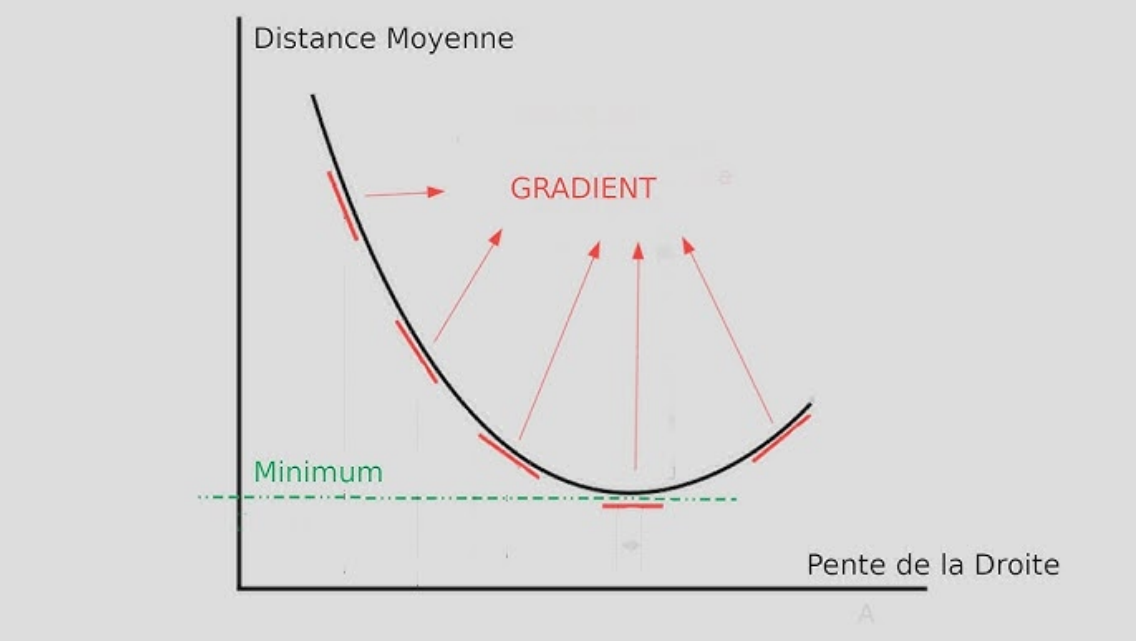

FIGURE 6 – *Illustration du principe de descente de gradient [5]*

le modèle de la descente de gradient commence avec des valeurs aléatoires pour ses paramètres, puis les ajuste progressivement à chaque exemple, pour apprendre à prédire correctement les classes à partir des données textuelles. Une fois entraîné, il peut alors généraliser à de nouveaux tweets jamais vus auparavant. Dans notre cas, nous avons utilisé le solveur `lbfgs`, une version plus avancée et rapide de la descente de gradient classique. Il permet d'accélérer la convergence du modèle, c'est-à-dire de trouver plus vite les bons paramètres, même avec un grand nombre de caractéristiques comme c'est le cas avec les textes vectorisés. Contrairement à d'autres modèles plus complexes, la régression logistique permet non seulement

de prédire une classe, mais aussi d'expliquer clairement les facteurs qui influencent cette décision.

F.3 Interprétation des résultats

Une fois le modèle entraîné, chaque mot du vocabulaire (ou caractéristique textuelle) est associé à un **poids** noté θ_j, qui indique l'importance de ce mot dans la prédiction d'une classe.

- ⋆ Un poids **positif** signifie que la présence du mot augmente la probabilité que le tweet soit classé dans une certaine catégorie (par exemple : *haineux*).
- ⋆ Un poids **négatif** indique au contraire que ce mot diminue la probabilité d'appartenance à cette classe.

Cela permet d'identifier de manière précise les termes qui influencent le plus le jugement du modèle, ce qui est essentiel pour la vérifiabilité et la confiance dans les systèmes de filtrage automatique.

Régularisation : éviter le suraprentissage Lorsqu'un modèle est trop complexe ou exposé à un trop grand nombre de variables peu pertinentes, il risque d'**apprendre par cœur** les exemples du jeu d'entraînement, sans être capable de généraliser à de nouvelles données. C'est ce qu'on appelle le **suraprentissage** (ou *overfitting*).

Pour éviter cela, on applique une technique appelée **régularisation**. Dans le cadre de la régression logistique, cela consiste à pénaliser les poids extrêmes θ_j dans la fonction de coût, en ajoutant un terme de régularisation.

On distingue principalement deux types de régularisation :

- ⋆ La **régularisation L2** (ou ridge), qui pénalise la somme des carrés des poids : $\sum \theta_j^2$. Elle tend à « lisser » les poids, sans en annuler totalement.
- ⋆ La **régularisation L1** (ou lasso), qui pénalise la somme des valeurs absolues des poids : $\sum |\theta_j|$. Elle peut annuler certains poids et donc simplifier le modèle en supprimant des caractéristiques jugées inutiles.

Dans notre modèle, nous avons opté pour une **régularisation L2**, associée à un hyperparamètre $C = 100.0$. Cet hyperparamètre joue un rôle d'équilibre :

- ⋆ Une valeur de C **élevée** réduit l'effet de la régularisation, laissant au modèle plus de liberté pour ajuster les poids.
- ⋆ Une valeur de C **faible** augmente la pénalité sur les grands poids, rendant le modèle plus simple et moins sujet à l'overfitting.

Le choix de $C = 100.0$ représente ici un compromis entre la complexité du modèle et la capacité de généralisation, permettant à notre classificateur de capturer

efficacement les caractéristiques discriminantes sans tomber dans un excès d'ajustement.

L'apprentissage supervisé constitue une approche pertinente lorsque des données annotées sont disponibles, comme c'est le cas dans notre étude avec des tweets étiquetés. Toutefois, dans des contextes à faible ressource, l'accès à des données labellisées peut s'avérer limité. L'apprentissage non supervisé devient alors une alternative précieuse, permettant d'exploiter les données brutes pour révéler des structures sous-jacentes, identifier des regroupements ou réduire la dimensionnalité.

I.2.1.2. Apprentissage non supervisé

L'apprentissage non supervisé est une branche du machine learning qui s'applique aux données non étiquetées, c'est-à-dire sans indication préalable de classes cibles. Contrairement à l'apprentissage supervisé, son objectif n'est pas de prédire une sortie, mais de découvrir des structures latentes dans les données. Il s'agit notamment de regrouper des observations similaires, de détecter des schémas récurrents ou encore de réduire la dimensionnalité pour faciliter l'interprétation [23]. Ces mécanismes sont illustrés à la Figure 7, qui représente différentes techniques appliquées à des données non étiquetées pour révéler leur organisation intrinsèque.

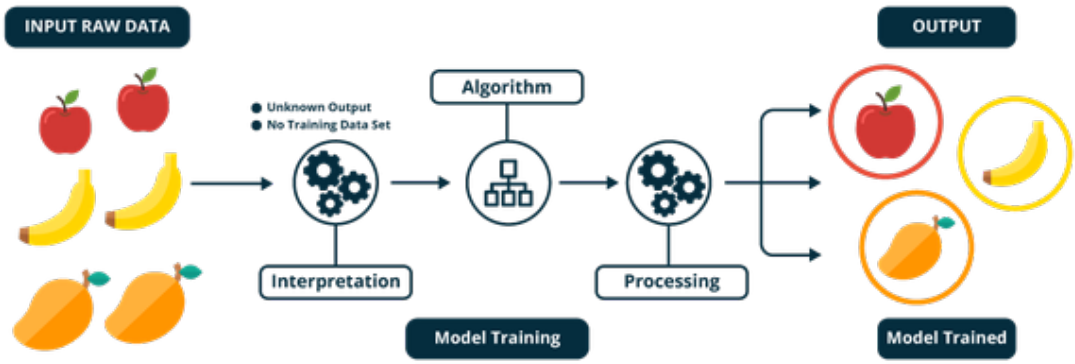

FIGURE 7 – *Apprentissage non supervisé [3]*

Le Tableau II, présente une description des méthodes couramment utilisé.

TABLE II – *Méthodes d'analyse exploratoire des données*

Méthode	Description
K-means	Méthode de *clustering* qui partitionne les données en K groupes en minimisant la distance intra-cluster [24].
Règles d'association	Utiles pour découvrir des relations fréquentes entre variables, notamment en marketing et dans les systèmes de recommandation [25].
Réduction de dimensionnalité (PCA)	Technique visant à simplifier les données tout en préservant l'information essentielle, afin de faciliter leur visualisation[26].

I.2.1.3. Apprentissage semi-supervisé

L'apprentissage semi-supervisé est une approche hybride qui combine à la fois des données annotées et un grand volume de données non étiquetées dans le processus d'entraînement d'un modèle [27]. Contrairement à l'apprentissage supervisé, où chaque observation est associée à une étiquette, cette méthode repose sur un petit sous-ensemble de données annotées, tandis que la majorité des exemples disponibles sont dépourvus d'étiquettes. Autrement dit, dans un ensemble d'observations $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$, seules quelques-unes sont associées à une sortie y_i. L'objectif consiste alors à exploiter la similarité entre les données étiquetées et non étiquetées pour propager les étiquettes manquantes et renforcer la performance prédictive du modèle [28]. L'apprentissage semi-supervisé s'inscrit donc à mi-chemin entre l'apprentissage supervisé et l'apprentissage non supervisé, en tirant parti des deux types d'informations pour améliorer la robustesse et la généralisation du système, comme illustré à la Figure 8.

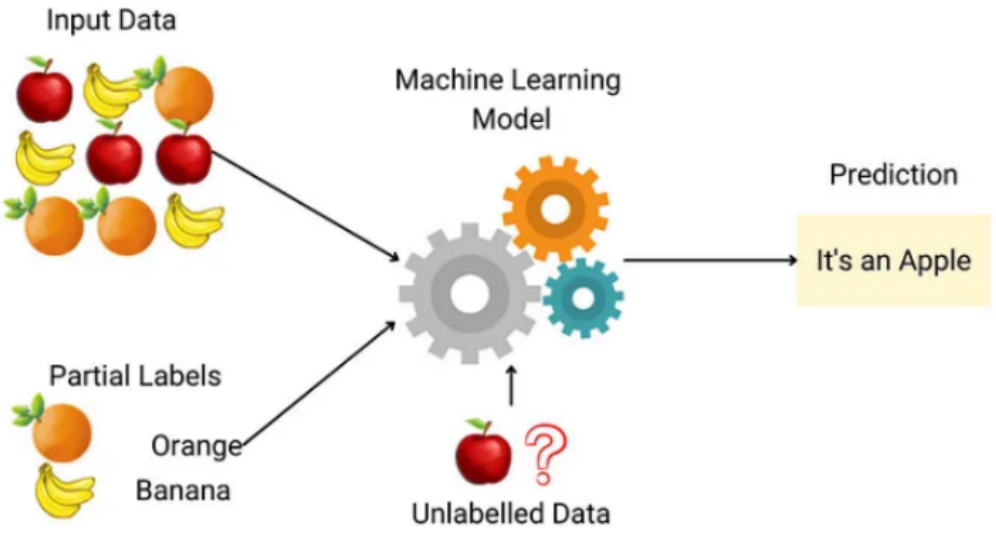

FIGURE 8 – *Apprentissage semi-supervisé [3]*

I.2.1.4. Apprentissage par renforcement

L'apprentissage par renforcement est une approche de l'IA dans laquelle un agent apprend à prendre des décisions en interagissant avec son environnement, sans supervision directe. Plutôt que de recevoir des étiquettes de classes, l'agent agit, observe les conséquences de ses actions et reçoit un retour sous forme de récompense ou de punition [29]. Son objectif est de maximiser les gains cumulés au fil du temps en développant une stratégie appelée politique optimale.

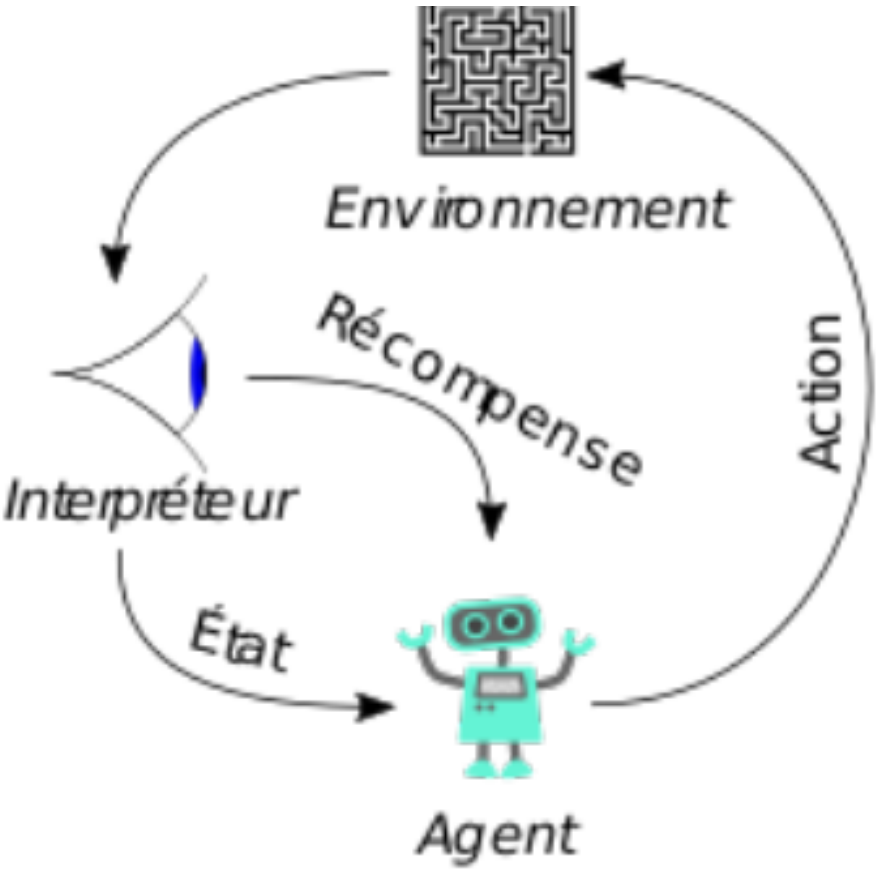

FIGURE 9 – *Apprentissage par renforcement [6]*

La Figure 9 illustre les composants clés de cette méthode : un agent qui observe un état s, choisit une action a, et reçoit en retour une récompense $r(s,a)$, ce qui influence ses choix futurs. Cette boucle d'interaction repose sur le principe d'essai-erreur.

I.3. Techniques d'extraction de caractéristiques

Dans le cadre de notre mémoire, l'extraction de caractéristiques textuelles constitue une étape cruciale pour transformer les tweets bruts en représentations exploi-

tables par les algorithmes d'apprentissage automatique. L'objectif de cette phase est de convertir chaque message textuel en un vecteur numérique qui capture l'information sémantique nécessaire à la tâche de classification. On distingue trois grandes catégories de techniques :

⋆ **Méthodes basées sur le comptage et les statistiques**
Parmi les premières approches utilisées pour transformer des textes en représentations numériques exploitables par les modèles de classification, les méthodes basées sur le comptage et les statistiques occupent une place centrale. Elles reposent sur des principes simples mais efficaces pour extraire l'information lexicale la plus significative d'un corpus.

— *Bag-of-Words (sac de mots)* : Cette méthode consiste à représenter chaque document sous la forme d'un vecteur de fréquence, où chaque dimension correspond à un mot du vocabulaire extrait du corpus. Elle ignore la syntaxe et l'ordre des mots, ne tenant compte que de leur présence ou fréquence. Par exemple, les phrases « le chat dort » et « dort le chat » auront des représentations identiques. Malgré cette simplicité, Bag-of-Words reste une technique performante pour de nombreuses tâches de classification de texte, notamment lorsqu'elle est utilisée sur des corpus homogènes et bien nettoyés. Sa popularité repose sur sa facilité de mise en œuvre, son efficacité computationnelle et sa robustesse dans les contextes supervisés classiques [30].

— *TF–IDF (Term Frequency–Inverse Document Frequency)* : Cette méthode affine l'approche Bag-of-Words en pondérant chaque mot selon son importance dans un document donné relativement à l'ensemble du corpus. Le poids d'un terme augmente proportionnellement à sa fréquence dans un document (TF) mais est atténué par sa fréquence dans l'ensemble des documents (IDF), ce qui permet de minimiser l'impact des mots très fréquents mais peu discriminants comme « le », « et », « un », etc. Formellement, le poids TF–IDF d'un terme t dans un document d est défini par :

$$\text{TF-IDF}(t,d) = \text{tf}(t,d) \times \log\left(\frac{N}{\text{df}(t)}\right) \text{[31]} \tag{6}$$

où $\text{tf}(t,d)$ est la fréquence du terme t dans le document d, N est le nombre total de documents dans le corpus, et $\text{df}(t)$ est le nombre de documents contenant le terme t. Cette pondération améliore la précision des modèles en valorisant les termes réellement informatifs [32].

⋆ **Word embeddings statiques**
En complément des approches basées sur le comptage, les word embeddings statiques offrent une alternative plus riche en capturant des relations sémantiques entre les mots à partir de leur contexte global, à l'aide de représentations vectorielles denses apprises sur de grands corpus.
— *Word2Vec* : apprend, via un modèle Skip-Gram ou CBOW, des vecteurs de mots tels que les termes sémantiquement proches possèdent des représentations voisines [33].
— *GloVe* : combine statistiques globales et co-occurrences locales pour produire des embeddings robustes sur de grands corpus [34].
— *FastText* : enrichit Word2Vec en décomposant les mots en n-grammes de caractères, améliorant la gestion des mots rares et des morphologies complexes [35].

⋆ **Embeddings contextuels profonds** Contrairement aux représentations statiques qui attribuent un vecteur fixe à chaque mot, les embeddings contextuels profonds ajustent dynamiquement la représentation d'un mot en fonction de son contexte d'apparition dans la phrase, offrant ainsi une compréhension plus fine et nuancée du langage naturel.
— *BERT (Bidirectional Encoder Representations from Transformers)* : génère pour chaque mot un vecteur dépendant de son contexte gauche et droit, permettant de saisir les multiples sens d'un même terme [36].
— *RoBERTa, ALBERT, DistilBERT* : variantes optimisées de BERT, améliorant la vitesse ou la taille du modèle tout en conservant une qualité d'encodage élevée [37, 38].
— *XLNet, GPT-3* : modèles autoregressifs ou permutationnels offrant des performances de pointe sur les tâches de classification de texte et de génération [39].

Dans le cadre de ce travail, nous avons opté pour des méthodes plus légères et facilement interprétables, particulièrement adaptées aux ressources disponibles et à la nature des données traitées. En particulier, deux techniques classiques ont été explorées et comparées :

— **CountVectorizer** : Cette méthode repose sur le modèle du sac de mots (Bag-of-Words). Elle construit un vocabulaire à partir du corpus, puis représente chaque texte par un vecteur dont chaque dimension correspond à la fréquence d'apparition d'un mot dans le document. Cette approche simple mais robuste permet de capter efficacement les termes discriminants sans introduire de biais sémantiques complexes.

— **TfidfVectorizer (Term Frequency – Inverse Document Frequency)** : Cette technique affine la représentation Bag-of-Words en pondérant chaque mot par son importance relative. Les termes fréquents dans un document mais rares dans l'ensemble du corpus reçoivent un poids plus élevé. Cela permet d'atténuer l'impact des mots très courants et d'accentuer ceux qui portent une valeur informative significative pour la classification.

Ces vecteurs creux de grande dimension sont ensuite utilisés comme entrées du classificateur de régression logistique.

I.4. Problème de la sécurité des enfants en ligne

À l'ère du numérique, les enfants sont de plus en plus exposés à des contenus en ligne potentiellement dangereux, tels que les discours haineux, les propos violents ou les situations de cyberintimidation. Les systèmes de modération traditionnels, basés sur des règles statiques ou l'intervention humaine, se révèlent souvent inefficaces face à la complexité, la subtilité et la masse des données publiées en ligne. Dans ce contexte, l'objectif de ce travail est de concevoir un système automatique de détection des contenus textuels inappropriés à destination des enfants, reposant sur des techniques d'apprentissage automatique supervisé. Plus précisément, nous proposons un modèle de classification multiclasse utilisant la régression logistique, combinée à des méthodes de vectorisation comme `CountVectorizer` ou `TfidfVectorizer`, afin d'identifier les discours problématiques de manière fiable, rapide et interprétable. Cette approche vise à renforcer la protection des jeunes utilisateurs en ligne en fournissant une solution simple, accessible et efficace pour prévenir leur exposition à des contenus nuisibles.

I.4.1. Risques en ligne

L'accès croissant des enfants à Internet les expose à plusieurs formes de contenus textuels nuisibles, pouvant compromettre leur développement personnel. Quatre grands types de risques se distinguent :

⋆ **Discours haineux** : propos discriminatoires ou violents visant des groupes ou individus en fonction de leur identité. Souvent subtils ou ironiques, ces messages normalisent la haine et influencent négativement la perception sociale des jeunes.

- ⋆ **Cyberharcèlement textuel** : attaques répétées par messages ou publications visant à humilier, menacer ou isoler psychologiquement les jeunes, avec des effets graves sur leur santé mentale.
- ⋆ **Propagande idéologique** : contenus incitant à la radicalisation ou à l'intolérance, qui exploitent la crédulité des enfants pour diffuser des idéologies extrémistes sans qu'ils en aient pleinement conscience.
- ⋆ **Désinformation** : fausses informations ou rumeurs présentées de manière persuasive, auxquelles les jeunes, souvent peu outillés pour vérifier les sources, sont particulièrement sensibles.

La diversité, la subtilité et la rapidité de diffusion de ces contenus rendent indispensable l'utilisation de systèmes automatiques capables de les détecter et de les filtrer efficacement, afin de protéger les enfants dans leurs usages numériques quotidiens.

I.4.2. Rôle des technologies dans la protection des enfants en ligne

Dans un contexte numérique où les enfants sont quotidiennement exposés à des contenus textuels potentiellement dangereux, les technologies d'analyse automatique du langage jouent un rôle central dans leur protection. Plusieurs approches sont aujourd'hui mobilisées :

- ⋆ **Les modèles de classification supervisée**, comme la régression logistique ou les SVM, permettent d'identifier automatiquement les contenus inappropriés à partir de données annotées.
- ⋆ **Les word embeddings**, tels que Word2Vec ou FastText, offrent une compréhension plus fine du sens des mots et des formulations indirectes utilisées dans les discours haineux.
- ⋆ **Les modèles de langage contextuels (Transformers)**, notamment BERT et ses variantes, améliorent la détection de messages subtils ou ambigus grâce à leur compréhension du contexte sémantique.
- ⋆ **Les systèmes de détection en temps réel** permettent une modération instantanée des contenus publiés, limitant ainsi l'exposition immédiate des jeunes utilisateurs.
- ⋆ **Les contrôles parentaux intelligents** offrent des restrictions dynamiques et des rapports d'activité, s'adaptant au profil et à l'âge de l'enfant.
- ⋆ **Les approches hybrides** combinent règles linguistiques et modèles statistiques pour mieux repérer les contenus explicites comme implicites.

⋆ **La prévention assistée par apprentissage automatique**, via des outils éducatifs ou ludiques, sensibilise les enfants aux bons comportements numériques.

Ces technologies, bien configurées et intégrées, constituent un levier essentiel pour détecter, filtrer ou anticiper les menaces textuelles en ligne, contribuant ainsi à un environnement plus sûr pour les jeunes internautes.

I.5. Conclusion

Ce premier chapitre a posé les bases théoriques indispensables à la compréhension de notre démarche sur la sécurité des enfants en ligne. Il a présenté les fondements de l'apprentissage automatique, en mettant l'accent sur l'apprentissage supervisé et la régression logistique, choisie pour sa simplicité et son efficacité en classification multiclasse. Les principales techniques de vectorisation textuelle (`CountVectorizer`, TF–IDF) ont été décrites, aux côtés d'approches plus avancées comme les embeddings sémantiques (Word2Vec, BERT). Nous avons également identifié les risques textuels majeurs du web (discours haineux, harcèlement, désinformation) et montré l'apport des technologies de traitement automatique du langage pour leur détection. Enfin, le rôle essentiel des outils d'IA dans la protection des jeunes publics a été souligné, préparant ainsi les fondements nécessaires aux développements méthodologiques et expérimentaux des chapitres suivants.

II

CHAPITRE

REVUE DE LA LITTÉRATURE POUR LA SÉCURITÉ DES ENFANTS EN LIGNE

SOMMAIRE

II.1. Introduction

La multiplication des contenus textuels partagés sur les réseaux sociaux, les forums, les jeux en ligne ou encore les plateformes éducatives représente une source d'inquiétude grandissante. En effet, les messages textuels sont le vecteur principal de nombreuses formes de toxicité numérique : intimidation, menaces, incitation à la haine, propos discriminatoires, et bien d'autres. Ces contenus, en apparence anodins, peuvent avoir un impact psychologique profond sur les enfants, notamment lorsque ceux-ci ne disposent pas des compétences cognitives nécessaires pour en comprendre le danger ou y réagir de manière appropriée [40].

C'est dans ce contexte que les chercheurs se sont tournés vers les méthodes d'intelligence artificielle, et plus spécifiquement vers le traitement automatique du langage naturel (TALN), pour développer des outils de détection automatique de ces contenus donnant ainsi aux machines la capacité de comprendre et d'interpréter le langage humain. Ces dernières années, l'essor des modèles de type transformers, et notamment Bidirectional Encoder Representations from Transformers (BERT), a profondément renouvelé les approches classiques de classification de texte. Grâce

à sa capacité à prendre en compte le contexte bidirectionnel des mots et à encoder de manière fine les relations sémantiques, Bidirectional Encoder Representations from Transformers (BERT) est devenu un standard dans les tâches de détection de toxicité, de haine ou de cyberviolence en ligne [36, 41].

Dans ce chapitre, nous proposons une revue critique de la littérature scientifique traitant de la sécurité des enfants en ligne à travers le prisme de l'apprentissage automatique, avec un accent particulier sur les approches de classification de texte basées sur Bidirectional Encoder Representations from Transformers (BERT). Après une exploration des travaux portant sur la détection de contenus inappropriés textuels, nous aborderons les approches multimodales, combinant texte et image. Nous examinerons ensuite les dispositifs actuels de filtrage et de surveillance, avant de mettre en évidence les limites méthodologiques et techniques des approches existantes. Cette revue a pour objectif de poser les bases conceptuelles et scientifiques qui orienteront la proposition de notre propre modèle de détection, présentée dans le chapitre suivant.

II.2. Travaux sur la détection de contenus inappropriés

Dans un contexte numérique où les enfants interagissent quotidiennement avec des contenus issus de réseaux sociaux, plateformes de vidéos, forums ou encore messageries instantanées, la détection automatique de contenus inappropriés est devenue une priorité pour garantir leur sécurité en ligne. Ces contenus, qu'ils soient textuels, visuels ou multimodaux, peuvent véhiculer des formes de haine, de harcèlement, de violences symboliques ou explicites, ou encore de désinformation, avec des conséquences parfois graves sur le bien-être psychologique et émotionnel des plus jeunes [40, 7].

Les travaux de recherche se sont particulièrement intensifiés autour de la détection automatique de ces contenus, grâce à l'essor de l'intelligence artificielle et du TALN. Les algorithmes d'apprentissage automatique, et plus récemment ceux issus du deep learning, ont permis d'obtenir des performances prometteuses dans l'identification de propos haineux, de cyberintimidation ou d'images à caractère choquant [42, 43].

Les approches actuelles combinent souvent des techniques classiques de classification supervisée avec des représentations vectorielles avancées comme les modèles de type BERT pour les textes, et les CNN pour les images. Ces modèles sont capables de repérer non seulement des termes ou objets explicites, mais aussi des messages à forte connotation négative, parfois exprimés de manière implicite et

contextuelle. La Figure 10 illustre les types de contenus que ces modèles cherchent à détecter.

FIGURE 10 – *Exemple de contenus inappropriés [7]*

II.2.1. Approches basées sur l'apprentissage automatique classique

Les modèles classiques, où les caractéristiques discriminantes sont extraites manuellement à partir des données brutes, ont longtemps constitué l'approche dominante pour la détection de contenus inappropriés. Avant l'essor des modèles d'apprentissage profond, la détection automatique de contenus violents ou haineux reposait principalement sur des méthodes classiques d'apprentissage supervisé. Ces approches incluaient notamment les machines à vecteurs de support (SVM),

les forêts aléatoires (Random Forest), ainsi que les k-plus proches voisins (k-NN). Elles étaient généralement couplées à des représentations de données construites manuellement, comme les histogrammes de couleurs, les gradients orientés (HOG), ou encore les descripteurs locaux tels que SIFT et SURF pour les images, et TF-IDF ou Bag-of-Words pour les textes [32, 30].

Dans le domaine de la détection textuelle de discours haineux, plusieurs études antérieures ont appliqué ces méthodes classiques avec succès. Par exemple, **Davidson et al. (2017)**[44] ont proposé un classifieur basé sur une régression logistique entraînée sur des représentations TF-IDF pour classifier les tweets en propos haineux, offensants ou neutres. Leur jeu de données annoté manuellement est devenu une référence dans le domaine et a été largement réutilisé par des études ultérieures.

De manière complémentaire, les modèles classiques ont également été explorés dans des cadres comparatifs ou en tant que baselines pour évaluer de nouveaux modèles. Par exemple, dans la revue de littérature menée par **Mansur et al. (2023)**[45], les auteurs rappellent que les méthodes comme Naive Bayes ou SVM ont souvent été utilisées comme points de référence dans la détection de discours haineux, avant l'arrivée des approches neuronales modernes. Cette rétrospective souligne le rôle fondamental joué par ces algorithmes dans la phase exploratoire du domaine.

Ces méthodes, bien qu'efficaces dans des contextes simples et faiblement dimensionnés, ont montré leurs limites dès lors qu'il s'agissait de capturer des nuances sémantiques complexes comme l'ironie, le sarcasme ou les formulations implicites. Toutefois, leur faible complexité computationnelle et leur grande interprétabilité en ont fait des outils de prédilection dans les environnements à ressources limitées ou dans les systèmes embarqués. Aujourd'hui encore, elles servent de base de comparaison pour valider l'intérêt des solutions basées sur l'apprentissage profond.

Nous abordons à présent les approches modernes reposant sur l'apprentissage profond, appliquées cette fois à la détection de contenus visuels inappropriés en particulier les images violentes où les modèles convolutifs (CNN) ont démontré des performances remarquables.

II.2.2. Approches basées sur les Images

L'émergence de l'apprentissage profond a profondément révolutionné le domaine de la vision par ordinateur, en rendant possible l'analyse automatique d'images complexes avec une grande précision. Dans le contexte spécifique de la détection de scènes violentes, les réseaux de neurones convolutifs (CNN) se sont imposés comme des solutions de référence, notamment grâce à leur capacité à extraire des

caractéristiques hiérarchiques telles que les bords, les textures, les formes et les objets.

La Figure 11 illustre une architecture type de réseau convolutionnel utilisée pour la détection de contenus visuellement violents. Ces modèles apprennent à classifier les images en exploitant des représentations visuelles obtenues à travers plusieurs couches de convolution et de regroupement.

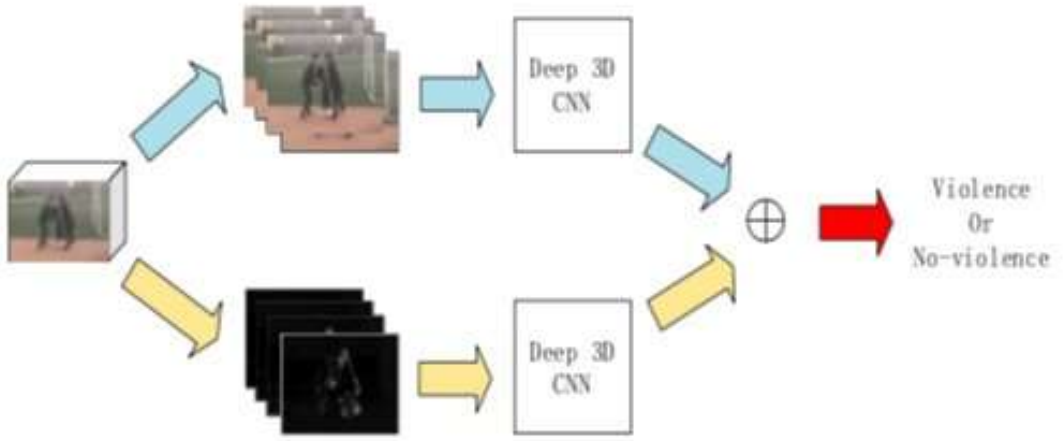

FIGURE 11 – *Méthode basée sur un CNN pour la détection d'images violentes [8]*

Les premières avancées ont été portées par des architectures convolutionnelles profondes telles que **VGGNet**, **ResNet** ou **InceptionV3**, qui ont démontré leur efficacité dans l'extraction de caractéristiques discriminantes pour la classification de contenus visuellement choquants [46, 47].

Plus récemment, l'attention s'est portée sur les modèles fondés sur l'architecture Transformer appliquée à la vision, tels que les **Vision Transformers (ViT)** et les modèles hybrides comme le **Swin Transformer**. Contrairement aux CNN traditionnels, ces modèles sont capables de capturer des relations à longue portée entre les éléments visuels d'une image, ce qui améliore significativement les performances dans des conditions dégradées (flou, obstruction partielle, faible résolution). Parmi les travaux notables combinant vision par ordinateur et traitement du langage naturel,L'étude de **Lee et al. (2023)** [48] illustre l'efficacité du *Swin Transformer* pour la détection de scènes violentes en temps réel, atteignant un taux de précision supérieur à 93 %. Dans une perspective similaire, Cao et al. (2023)[49] s'intéressent aux publications haineuses sur Instagram. Leur modèle repose sur une fusion tardive entre RoBERTa pour la composante textuelle (captions, hashtags, commentaires) et DenseNet pour l'analyse visuelle des images publiées. À la différence de la fusion intermédiaire, la fusion tardive combine les décisions indépendantes des deux branches après traitement, ce qui simplifie l'architecture tout

en conservant une bonne performance globale. Les auteurs montrent que cette approche est particulièrement efficace pour détecter des contenus haineux implicites diffusés sur les plateformes centrées sur l'image. Ainsi, les progrès dans les modèles de vision par ordinateur, et en particulier l'introduction des Transformers visuels, ont profondément renouvelé les méthodes de détection de contenus violents. Le recours à des stratégies multimodales et des architectures hybrides constitue une orientation prometteuse pour mieux répondre aux défis posés par la complexité des données visuelles en ligne. La diversité des modèles visuels, allant des CNN classiques aux architectures Transformer et hybrides, est synthétisée dans le Tableau III, qui met en lumière les contributions récentes les plus significatives en matière de détection automatique de contenus violents à partir d'images ou de vidéos.

TABLE III – *Synthèse des approches récentes basées sur les images pour la détection de contenus violents*

Étude	**Modèle utilisé**	**Type de contenu détecté**
Singh et al. (2022) [46]	CNN classique (VGGNet, InceptionV3)	Scènes de violence dans les images ou vidéos
Dhiman et al. (2021) [47]	ResNet + LSTM	Actes violents dans les vidéos à faible résolution
Lee et al. (2023) [48]	Swin Transformer (vision Transformer hiérarchique)	Détection de violence en temps réel dans des environnements complexes
Cao et al. (2023)[49]	RoBERTa + DenseNet + Late Fusion	Instagram hate content

Après avoir exploré les méthodes de détection de contenus inappropriés dans les images à l'aide d'architectures profondes, il convient désormais d'analyser les approches centrées sur les contenus textuels ; Dans cette optique, plusieurs travaux récents ont proposé des modèles avancés capables de capturer les subtilités du langage, allant au-delà des représentations traditionnelles. Ces contributions s'appuient principalement sur des modèles de type Transformer, tels que BERT, ou sur des architectures hybrides combinant plusieurs types de réseaux neuronaux. La suite de cette sous-section présente les principales avancées dans ce domaine, en mettant l'accent sur les modèles fondés sur les textes, souvent évalués à l'aide du corpus de référence proposé par Davidson et al. [44].

II.2.3. Approches basées sur le texte

Les modèles traditionnels de détection de discours haineux rencontrent des difficultés face à des expressions implicites comme le sarcasme, les doubles sens ou

les discours codés, souvent mal capturés par les approches supervisées classiques et les annotations humaines parfois incohérentes. Pour répondre à cette problématique, Yang et al. (2023)[50] ont proposé une nouvelle architecture appelée **HARE** (*Explainable Hate Speech Detection with Step-by-Step Reasoning*), qui intègre un mécanisme de raisonnement progressif basé sur des *chain-of-thought prompts*. Cette méthode permet de générer, pour chaque texte haineux détecté, une explication explicite appelée *rationale*, que ce soit de manière automatique (Fr-HARE) ou guidée par des annotations humaines (Co-HARE). Leur approche repose sur des modèles de langage pré-entraînés (Flan-T5 et GPT-2) appliqués aux jeux de données SBIC et Implicit Hate (issus du travail initial de Davidson et al.[44]), avec une phase de fine-tuning où les rationales sont intégrées dans la supervision. Les résultats montrent que HARE surpasse les modèles de classification traditionnels, atteignant des scores F1 supérieurs à 85 %, tout en renforçant l'interprétabilité des prédictions. Fr-HARE produit des raisonnements plus proches de la logique humaine tandis que Co-HARE se montre plus cohérent avec les annotations de référence. Cette capacité à combiner performance et transparence positionne HARE comme une avancée significative pour la détection automatisée de contenus textuels inappropriés.

Dans la continuité des efforts visant à renforcer la robustesse des systèmes de détection automatique, une autre étude marquante est celle de Mnassri et al. (2022)[51]. Ces auteurs se sont penchés sur les limites liées au déséquilibre des classes présentes dans les jeux de données existants, notamment celui de Davidson et al.[44], largement utilisé dans ce domaine. Pour pallier ce problème, les auteurs proposent une stratégie fondée sur des modèles d'ensemble basés sur BERT, intégrés via des techniques telles que le soft voting, le hard voting et le stacking. Leur principale contribution consiste en la création d'un nouveau corpus composite et équilibré nommé **DHO**, fusionnant trois ressources majeures : Davidson, OLID et HatEval. Chaque modèle BERT est affiné (fine-tuned) individuellement sur ces données, avant d'être combiné selon les différentes stratégies d'ensemble mentionnées. Les résultats expérimentaux révèlent que le modèle de type stacking surpasse nettement les autres configurations lorsqu'il est appliqué au dataset Davidson pur, atteignant un score impressionnant de **97 % de macro-F1**. Toutefois, sur le corpus DHO, les performances globales s'avèrent moindres, avec un score autour de **77 % de macro-F1**, ce qui témoigne d'une robustesse accrue mais aussi d'une plus grande difficulté liée à la variabilité sémantique des données issues de plusieurs sources. Cette étude met ainsi en évidence l'intérêt de combiner plusieurs

jeux de données et de recourir à des architectures ensemblistes pour améliorer la détection de contenus haineux textuels sur Twitter.

Dans une autre optique complémentaire à l'approche ensembliste, Putra et Wang (2024)[52] explorent la complémentarité entre les représentations contextuelles riches produites par BERT et la capacité des réseaux de neurones convolutionnels (CNN) à capter des motifs locaux discriminants. Leur approche repose sur une architecture hybride baptisée **Advanced BERT-CNN**, qui applique une couche CNN directement sur les embeddings issus de BERT, permettant ainsi de combiner des informations globales contextuelles et des structures syntaxiques locales présentes dans les tweets. L'expérimentation est menée sur le jeu de données de Davidson et al., utilisé comme référence dans de nombreux travaux sur la détection de propos haineux. Les résultats montrent que leur modèle atteint un **F1-score de 73 %**, ce qui représente une amélioration significative par rapport aux performances typiquement observées sur ce corpus (comprises entre 61 % et 90 % selon les configurations de base). Ce gain de performance est particulièrement manifeste en termes de précision sur la classe haineuse, démontrant la pertinence de combiner BERT à des couches convolutives pour enrichir la classification des contenus textuels inappropriés.

La synthèse des approches textuelles récentes illustrée dans le Tableau IV met en évidence la diversité des modèles exploités et des contenus haineux ciblés, notamment à partir du corpus de Davidson et al.[44], largement utilisé dans les travaux de référence sur le sujet.

TABLE IV – *Synthèse des études récentes sur la détection de discours haineux à partir de textes*

Étude	Modèle utilisé	Type de contenu détecté
Yang et al. (2023) [50]	HARE (Flan-T5/GPT-2 avec raisonnement par étapes)	Propos haineux implicites avec explications (rationales)
Mnassri et al. (2022) [51]	Modèles BERT ensemblistes (soft/hard voting, stacking)	Tweets haineux à partir de corpus fusionnés (Davidson, OLID, HatEval)
Putra et Wang (2024) [52]	Architecture hybride BERT + CNN	Propos haineux dans les tweets (dataset Davidson)

II.2.4. Approches multimodales

L'approche multimodale repose sur la complémentarité entre les informations visuelles et textuelles pour améliorer la détection automatique de contenus inappropriés. Tandis que le texte fournit des indices sémantiques explicites tels que des

insultes ou propos discriminants, l'image transmet des signaux implicites complexes à interpréter de manière isolée. L'exploitation conjointe de ces deux modalités permet d'enrichir la compréhension contextuelle et d'accroître la robustesse des systèmes de classification.

Dans cette perspective, plusieurs architectures ont été développées afin de fusionner efficacement les représentations extraites par des modèles spécialisés en traitement du langage naturel (comme **BERT**) et en vision par ordinateur (tels que **VGG19** ou **ResNet**). Ces systèmes de fusion reposent généralement sur trois stratégies principales : *fusion précoce* (early fusion), *fusion intermédiaire* (intermediate fusion) et *fusion tardive* (late fusion), chacune ayant ses avantages selon la tâche ciblée. Kumar et al. (2022)[53] propose une approche multimodale dédiée à la détection de mèmes haineux, un format de plus en plus répandu sur les réseaux sociaux. Leur architecture repose sur une fusion intermédiaire entre deux modèles spécialisés : ResNet pour l'extraction des caractéristiques visuelles à partir des images, et BERT pour l'analyse sémantique du texte intégré dans les mèmes. Cette fusion intermédiaire permet d'établir une interaction plus fine entre les deux modalités avant la classification finale, ce qui améliore la compréhension du contexte global du message. L'approche s'est montrée performante sur des ensembles de mèmes haineux, soulignant l'importance de capter à la fois les signaux implicites de l'image et les indices explicites du langage. dans ce même schéma, l'étude de **Vijayaraghavan et al. (2021)** [54] propose un modèle interprétable basé sur une attention croisée entre les représentations textuelles et visuelles, appliquée à la dé-Gestion de messages haineux. Le système atteint une précision de 84,2 % sur un corpus annoté manuellement, démontrant l'intérêt d'une interaction directe entre les modalités. Par ailleurs, la revue de littérature menée par **Zhou et al. (2023)** [7] passe en revue les principales méthodes de fusion multimodale, et souligne notamment l'efficacité de la fusion intermédiaire, qui permet une interaction plus fine entre les représentations encodées des deux modalités avant la classification. De plus, la tendance actuelle va vers une approche multimodale combinant les informations issues des images et des textes. Le travail de **Osei et Wang (2024)** [55] propose une architecture intégrant un *Visual Transformer* avec un modèle NLP pour renforcer la détection de contenus explicites dans les publications sur les réseaux sociaux. Leur système atteint un score F1 de 91 %, confirmant l'intérêt des approches croisées image/texte. Le tableau V présente une synthèse des travaux récents les plus représentatifs utilisant des modèles multimodaux pour la détection de contenus inappropriés dans les environnements en ligne. Il met en lumière les différentes combinaisons de modèles, ainsi que les corpus employés.

TABLE V – *Travaux récents sur la détection de contenus inappropriés en utilisant des approches multimodales*

Étude	Modèles utilisés	Données
Vijayaraghavan et al. (2021)[54]	BERT + CNN	Multimodal Twitter dataset
Zhou et al. (2023)[7]	ViLT (Vision+Language Transformer)	Hateful Memes Dataset (Facebook)
Osei et Wang (2024) [55]	Vision Transformer + NLP (approche multimodale)	Contenus explicites (images + textes) sur les réseaux sociaux
Kumar et al. (2022)[53]	ResNet + BERT + Fusion Intermédiaire	mèmes haineux
Shah et al. (2024)[56]	CLIP + Transformer multimodal	Multimodal meme dataset

II.2.5. Analyse comparative des approches

Les travaux récents dans le domaine de la détection de contenus violents ont fait émerger plusieurs tendances technologiques significatives. **Analyse sémantique fine des textes**

Les modèles comme BERT [57] ont démontré leur efficacité dans la compréhension du langage naturel grâce à leur capacité à analyser les relations contextuelles entre les mots dans une phrase. Contrairement aux filtres traditionnels qui se basent uniquement sur des mots-clés, BERT est capable de détecter un contenu haineux, sexuellement explicite ou violent même lorsque celui-ci est exprimé de manière implicite, humoristique ou détournée [58]. Cela permet de réduire significativement les faux négatifs.

Reconnaissance visuelle avancée

Dans le cas des images, les architectures CNN comme VGG19, ResNet ou EfficientNet permettent d'extraire des caractéristiques hiérarchiques (formes, textures, objets) pour identifier des éléments visuels offensants tels que des scènes de violence, des armes ou de la nudité [59]. Ces modèles peuvent détecter des éléments subtils non repérables par une simple analyse de métadonnées ou de balises.

Approche multimodale intégrée

Les contenus les plus problématiques, comme les mèmes haineux ou les vidéos combinant images et propos agressifs, nécessitent une approche multimodale. La combinaison de modèles de traitement de texte (comme BERT) et d'image (comme CNN) dans un seul pipeline permet d'analyser le message global porté par un contenu. Des recherches comme celles de Zhou et al. [7] ou Miraftabzadeh et al. [15] montrent que les approches multimodales surpassent nettement les systèmes

unimodaux en termes de précision et de pertinence, en particulier dans les contextes complexes.

Fonctionnement en temps réel et automatisé
Une fois intégrés dans un système de navigation ou de messagerie, ces modèles peuvent fonctionner de façon automatisée, en filtrant les contenus en amont (avant affichage) ou en déclenchant une alerte. Des plateformes comme Facebook ou TikTok utilisent déjà ce type de détection à grande échelle [43].

Modèles hybrides : L'association des réseaux de neurones convolutifs (CNN) avec des architectures récurrentes telles que les GRU ou LSTM permet de tirer parti à la fois des caractéristiques spatiales (issus des images) et temporelles (issus des séquences vidéo). Cette combinaison améliore notablement les performances de détection des comportements violents, notamment dans les flux vidéo continus. Par exemple, Haque et al. (2024)[60] ont proposé le modèle *BrutNet*, une architecture hybride combinant un DCNN avec une cellule GRU, qui a obtenu d'excellents résultats sur des vidéos de surveillance en capturant efficacement les dynamiques temporelles.

Efficacité computationnelle : L'adoption de modèles légers comme EfficientNet-B0, enrichis par des mécanismes d'attention, constitue une solution efficace pour le traitement en temps réel. Ces modèles offrent un bon équilibre entre précision et rapidité, ce qui les rend particulièrement adaptés à des environnements contraints en ressources, tels que les dispositifs embarqués. C'est le cas du travail de Li et al. (2024)[61], qui introduisent une architecture combinant EfficientNet-B0 à un module attentionnel bidirectionnel (*Bi-LTMA*), permettant de détecter efficacement les scènes violentes tout en maintenant une faible charge computationnelle.

Transformers pour la vision : Les architectures inspirées des Transformers, comme ViViT, ont démontré une capacité prometteuse à modéliser les relations à long terme dans les vidéos. Leur structure permet de capturer des dépendances globales dans le temps et l'espace, bien que leur complexité computationnelle élevée reste un frein à une adoption large, notamment en contexte opérationnel. À ce titre, l'étude de Singh et al. (2022)[62] met en avant les performances de ViViT pour la détection de comportements violents dans des environnements complexes, tout en soulignant les défis liés à l'inférence temps réel.

II.3. Méthodes de protection existantes (filtrage, surveillance)

Afin d'assurer une sécurité renforcée aux enfants lors de l'utilisation d'Internet, il est indispensable de mettre en œuvre des méthodes capables de bloquer

directement les contenus violents, haineux ou à caractère offensant. Contrairement aux approches générales de sensibilisation ou d'éducation, ces solutions techniques visent une protection immédiate au moment de la navigation ou de l'usage d'applications.

1. Filtrage de contenu

Le filtrage constitue la première barrière mise en place pour protéger les enfants de contenus inappropriés en ligne, tels que discours haineux, la violence, la pornographie. Initialement fondé sur des listes noires de mots-clés ou d'URL, ce type de filtrage statique montre rapidement ses limites face à la subtilité et à la diversité des contenus numériques. Pour y remédier, les approches modernes s'appuient sur des algorithmes d'apprentissage automatique capables de traiter plusieurs modalités (texte, image, audio). Des modèles puissants comme BERT pour le texte [57], ResNet pour l'image [63], ou des réseaux CNN pour l'audio [64] sont désormais intégrés dans des systèmes intelligents de détection contextuelle. Des solutions avancées telles que SafeNet (Meta) exploitent la fusion multimodale pour bloquer automatiquement les contenus problématiques avant leur affichage [7].

2. Applications intelligentes de surveillance comportementale et contextuelle

La surveillance s'appuie sur le suivi des interactions numériques des enfants afin d'identifier des comportements suspects, comme l'exposition répétée à du contenu sensible ou des interactions anormales avec d'autres utilisateurs. Ces méthodes reposent sur l'analyse de logs de navigation, de conversations, ou encore sur l'utilisation de capteurs d'activité, et font intervenir des modèles prédictifs pour anticiper les risques.

Des plateformes comme comme *Bark*, *SafeToNet* et *Canopy AI* utilisent par exemple des modèles NLP avancés pour détecter des signaux de détresse, d'intimidation ou de manipulation et analyser temps réel ; Ces applications peuvent détecter des signes de harcèlement, de nudité, de propos haineux ou violents et les bloquer avant qu'ils ne soient visibles par l'enfant. Cette approche est illustrée dans l'étude de Alam et al. (2024)[65], qui développe un modèle de deep learning adapté à l'âge pour la détection automatique de contenus inappropriés dans des vidéos les contenus affichés à l'écran (images, vidéos, textes), aussi, dans [43], les auteurs proposent un système d'alerte basé sur le deep learning qui identifie en temps réel des schémas de harcèlement dans des flux de messages sur les réseaux sociaux. La Figure 12 illustre les différentes composantes de l'écosystème de protection des enfants en ligne, en mettant en évidence les méthodes de prévention, de détection et d'intervention coordonnées entre les parties prenantes.

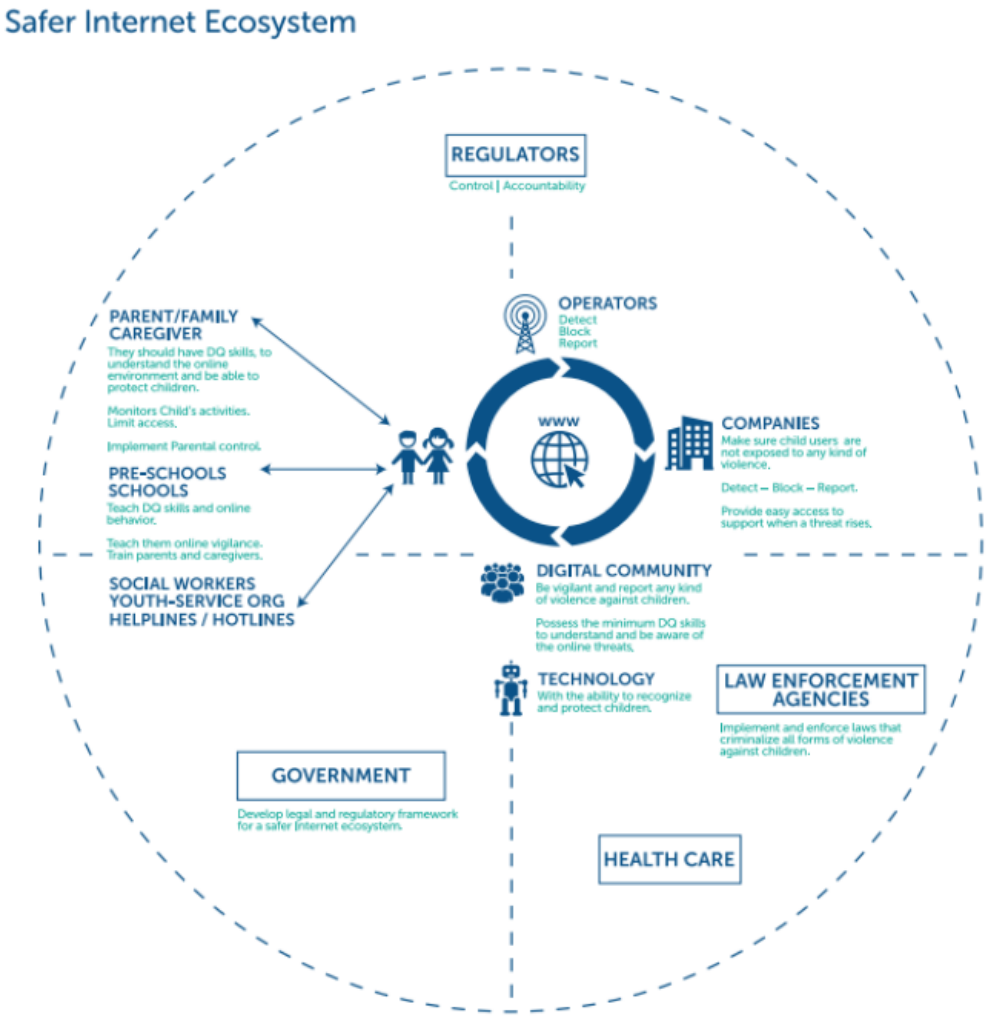

FIGURE 12 – *Écosystème de protection des enfants en ligne [9]*

3. Contrôle parental intégré

Les systèmes d'exploitation modernes (Windows, Android, iOS) intègrent des outils natifs de contrôle parental. Une étude approfondie sur 75 applications Android de contrôle parental montre que la majorité des fonctionnalités privilégie les stratégies de restriction et de surveillance familiale plutôt que l'autonomisation de l'adolescent, ce qui rejoint les capacités natives offertes par les réglages de temps d'écran sur iOS et Windows(Wisniewski et al.,2017)[66] ; Ils offrent des fonctionnalités telles que le blocage de sites web, le filtrage automatique de contenus (textes ou images violents, pornographiques ou haineux), la limitation du temps d'écran et le suivi des activités en ligne ; Par exemple, les paramètres de « Temps d'écran » sur iOS permettent de bloquer les contenus pour adultes. Des solutions comme *Qustodio*, *Norton Family*, ou encore *Google Family Link* sont largement utilisées à cet effet. Zhu, Deng et Bai (2023)[67] explorent comment le contrôle parental,

qu'il soit comportemental ou psychologique, modère l'addiction à Internet chez l'adolescent, en tenant compte des relations parents-enfants.

4. Navigateurs sécurisés pour enfants

Des navigateurs spécialisés proposent un environnement de navigation limité à une sélection de sites web adaptés à l'âge. Ils bloquent l'accès à tout contenu non validé par une liste blanche. Par exemple, *Kiddle* (un moteur de recherche sécurisé pour enfants) et *YouTube Kids* filtrent automatiquement les vidéos et résultats selon des critères de sécurité renforcés. Al-Ghamdi Al-Dala'in (2016)[68] proposent un navigateur web éducatif et sécurisé pour enfants, intégrant un système de filtres pour restreindre les sites nocifs et n'afficher que du contenu approprié

5. DNS filtrant au niveau du réseau

Une méthode efficace consiste à configurer un DNS sécurisé au niveau du routeur Wi-Fi ou de l'appareil. Des services de DNS filtrants comme *CleanBrowsing*, *OpenDNS Family Shield* ou *NextDNS* bloquent l'accès à des catégories entières de sites (violents, pornographiques, haineux) sans avoir besoin d'installer de logiciel. Ces services, souvent appelés Protective DNS, adoptent des politiques de réécriture DNS pour empêcher la résolution des domaines non autorisés, tout en maintenant une latence minimale. Une analyse récente (Liu et al., 2024)[69] souligne l'ampleur du déploiement de ces services et discute de leurs implications en termes de sécurité et de fiabilité.

6. Messageries sécurisées pour mineurs

Pour limiter les risques de harcèlement ou d'exposition à des propos violents dans les échanges en ligne, certaines messageries dédiées aux enfants offrent des fonctions de surveillance. *Messenger Kids*, par exemple, permet aux parents d'approuver les contacts de leur enfant et de consulter les messages échangés. Certaines plateformes détectent également automatiquement des mots-clés sensibles (ex. : insultes, menaces) et alertent les responsables légaux. Une revue récente souligne que, bien que ces outils répondent aux attentes des familles en matière de protection, leur efficacité dépend fortement du contexte d'usage, de la transparence des alertes et du potentiel traumatisant d'une surveillance excessive (Stoilova & Livingstone,2024)[70].

En combinant ces méthodes, il est possible d'établir un écosystème numérique sécurisé comme le montre le Tableau VI, adapté à l'âge de l'enfant, tout en respectant son développement numérique progressif. Ces approches, loin de remplacer les dispositifs techniques, viennent les enrichir pour proposer une stratégie de protection plus globale, respectueuse des droits de l'enfant et mieux ancrée dans les réalités sociales et culturelles.

TABLE VI – *Exemples de systèmes actuels de protection des enfants en ligne*

Système	**Approche utilisée**	**Type de contenu ciblé**	**Technologie**
Google SafeSearch[71]	Filtrage automatique par IA	Images, sites explicites	CNN, NLP
Bark[72]	Analyse des échanges textuels	Cyberharcèlement, suicide	NLP supervisé (SVM, BERT-based)
Facebook SafeNet[73]	Détection multimodale	Violence, haine, nudité	Multimodal (BERT+ResNet)
Qustodio[66]	Surveillance des usages	Temps d'écran, contenus inappropriés	Apprentissage supervisé

Bien que ces outils contribuent significativement à renforcer la sécurité des enfants en ligne, leur efficacité reste conditionnée à une mise à jour continue et à une adaptation aux réalités culturelles et linguistiques variées.

II.4. Limites des approches actuelles

Malgré les progrès, les méthodes actuelles présentent des limites importantes. D'une part, les modèles d'apprentissage automatique ou profond exigent de vastes jeux de données annotés pour être fiables. Comme le soulignent Samal et al.[74], l'absence de grands corpus bien étiquetés pour le contenu « malsain » conduit à de l'overfitting et à des performances instables. Les données disponibles sont souvent restreintes (par exemple, peu d'images pédo-pornographiques légales) ou déséquilibrées, ce qui restreint la généralisation des modèles. D'autre part, l'aspect évolutif du web pose problème : les contenus inappropriés changent avec le temps et les fraudeurs développent constamment de nouveaux stratagèmes (termes codés, nouvelles mèmes), ce qui rend un apprentissage statique rapidement obsolète. Enfin, les systèmes automatisés peinent à saisir les nuances contextuelles.**Chawki (2023)** [75] souligne que, les systèmes de modération automatique basés sur l'intelligence artificielle présentent encore de nombreuses limites, L'auteur montre que les outils algorithmiques actuels ont du mal à interpréter, à comprendre certains types de messages complexes, comme les blagues ironiques, les doubles sens ou les expressions liées à une culture particulière. Il note également que la nature en temps réel et générée par les utilisateurs (par exemple dans les environnements virtuels comme Roblox) complique la modération automatique. Chawki De plus, la sur-application des filtres peut générer des contenus bloqués par erreur (faux posi-

tifs) ce qui peut empêcher les jeunes de s'exprimer librement et leur donner l'impression d'être censurés sans raison. Il existe des enjeux de vie privée : filtrer les communications des enfants (par exemple via messagerie instantanée) soulève des questions éthiques et légales, imposant souvent des techniques de protection des données (fédération de l'apprentissage, anonymisation). Ces défis montrent que, malgré l'efficacité croissante des modèles, une modération humaine supervisée et des garde-fous réglementaires demeurent nécessaires.

Réduction des faux positifs et négatifs

Face aux limites observées dans les systèmes actuels notamment, le risque de censure injustifiée ou d'inefficacité face à des contenus ambiguës, il est essentiel de concevoir des solutions capables de mieux équilibrer la précision et la sensibilité. Les modèles d'apprentissage supervisé peuvent contribuer à cette amélioration, en réduisant à la fois les *faux positifs* et les contenus dangereux non détectés *faux négatifs*). La solution développée dans ce mémoire, fondée sur un modèle de classification supervisée par régression logistique multiclasse appliquée à des textes vectorisés (via CountVectorizer) permet de garantir une meilleure protection des enfants tout en préservant leur liberté d'expression et leur accès à des contenus utiles, notamment éducatifs. Cette solution s'inscrit donc comme une réponse pragmatique aux contraintes techniques et contextuelles évoquées précédemment. En misant sur un algorithme léger, interprétable et facile à implémenter, elle répond aux besoins spécifiques des acteurs éducatifs, institutionnels ou associatifs intervenant dans des régions où l'accès à des infrastructures technologiques avancées reste limité. Elle constitue une première étape vers la mise en place de mécanismes de filtrage fiables, compréhensibles et culturellement adaptés, tout en posant les bases pour des extensions futures, notamment vers des modèles plus complexes et multimodaux lorsque les ressources le permettront.

II.5. Conclusion

Ce chapitre a présenté une vue d'ensemble des principales approches basées sur l'intelligence artificielle pour la protection des enfants en ligne. Nous avons d'abord examiné les méthodes de détection automatique de contenus inappropriés, qu'il s'agisse d'images choquantes ou de textes haineux, en mettant en évidence l'efficacité des modèles profonds comme les réseaux convolutifs (VGG19, ResNet) pour l'analyse visuelle, et des modèles de type BERT pour le traitement du langage. Nous avons également souligné l'émergence des approches multimodales, qui combinent plusieurs types de signaux (textes, images, parfois audio) et offrent

de meilleures performances dans des environnements numériques complexes tels que les réseaux sociaux. En parallèle, différents systèmes existants qu'ils reposent sur des filtres, du DNS sécurisé, ou des applications intelligentes ont été passés en revue pour montrer la diversité des mécanismes de protection disponibles tout en soulignant les défis techniques, éthiques et contextuels qui persistent dans la mise en œuvre de solutions réellement efficaces et adaptées à la protection des jeunes sur les réseaux sociaux.Le prochain chapitre sera consacré à la mise en œuvre expérimentale de notre modèle de détection automatique. Il décrira les différentes étapes du processus : préparation et nettoyage des données, construction du modèle, entraînement, évaluation des performances, ainsi que les perspectives d'amélioration envisagées.

III

CHAPITRE

CONTRIBUTION À LA SÉCURITÉ DES ENFANTS EN LIGNE VIA LA DÉTECTION AUTOMATIQUE DE DISCOURS HAINEUX À L'AIDE D'UN MODÈLE DE RÉGRESSION LOGISTIQUE

SOMMAIRE

III.1. Introduction

L'augmentation exponentielle des contenus textuels sur les plateformes numériques a amplifié le risque d'exposition à des discours haineux, en particulier pour les publics vulnérables comme les enfants et les adolescents. Ces propos, souvent dissimulés derrière des formes linguistiques ambigües, représentent une menace grave pour la cohésion sociale, le bien-être psychologique des jeunes et leur développement personnel. Dans ce contexte, la détection automatique des discours haineux en ligne devient un enjeu majeur pour garantir un environnement numérique plus sûr.

Les approches basées sur l'apprentissage automatique, notamment la classification de texte, se sont révélées particulièrement efficaces pour cette tâche. Toutefois,

la nature ambiguë, contextuelle et en constante mutation des discours haineux rend leur identification difficile, exigeant des modèles d'apprentissage rigoureux, efficaces et faciles à interpréter. Dans cette optique, le présent chapitre présente une contribution technique à la détection de discours haineux sur les réseaux sociaux, reposant sur une architecture sobre et performante, qui intègre une étape de tokenisation suivie d'une vectorisation des textes via *CountVectorizer*, puis l'utilisation d'un classifieur par régression logistique multiclasse pour la prédiction [76].

Le modèle proposé suit un pipeline complet : à partir d'un corpus de tweets annotés (jeu de données de Davidson et al. [44]), il applique un nettoyage systématique des textes, suivi d'une étape de tokenisation, puis d'une suppression des biais lexicaux. Les textes ainsi préparés sont ensuite vectorisés par la méthode du sac de mots à l'aide de *CountVectorizer*, avant d'être utilisés pour l'entraînement d'un classifieur par régression logistique. L'objectif est d'assigner automatiquement à chaque message une des trois classes suivantes : *non haineux*, *haineux* ou *offensif*.

Ce chapitre s'articule autour des étapes suivantes :

- ⋆ Présentation de l'architecture.
- ⋆ Justification des technologies employées pour l'implémentation.
- ⋆ Résultats expérimentaux
- ⋆ Discussion et analyse
- ⋆ Recommandations et perspectives

À travers cette contribution,nous démontrons qu'il est possible, même avec des techniques simples et interprétables, de développer un modèle de détection de discours haineux performant pour renforcer la protection des jeunes utilisateurs dans les espaces numériques à fort risque.

III.2. Présentation de l'architecture

Ici, nous décrivons l'architecture globale du système de détection de contenus textuels inappropriés mis en œuvre dans le cadre de ce mémoire. Notre solution repose sur une chaîne de traitement modulaire, dans laquelle chaque composant remplit une fonction précise allant de la préparation des données à la classification finale. L'objectif est de concevoir un pipeline clair, reproductible et cohérent avec les principes de l'apprentissage supervisé présentés dans la partie du chapitre I.2.1.1. Avant de détailler le fonctionnement de chaque module, nous proposons d'abord un aperçu global du système dans son ensemble.

III.2.1. Aperçu global du système

La Figure 13 illustre l'architecture générale du système de détection automatique de discours haineux développé dans le cadre de ce travail. Ce système repose sur une chaîne de traitement modulaire, allant de la préparation initiale des données à la classification finale via un modèle d'apprentissage supervisé.

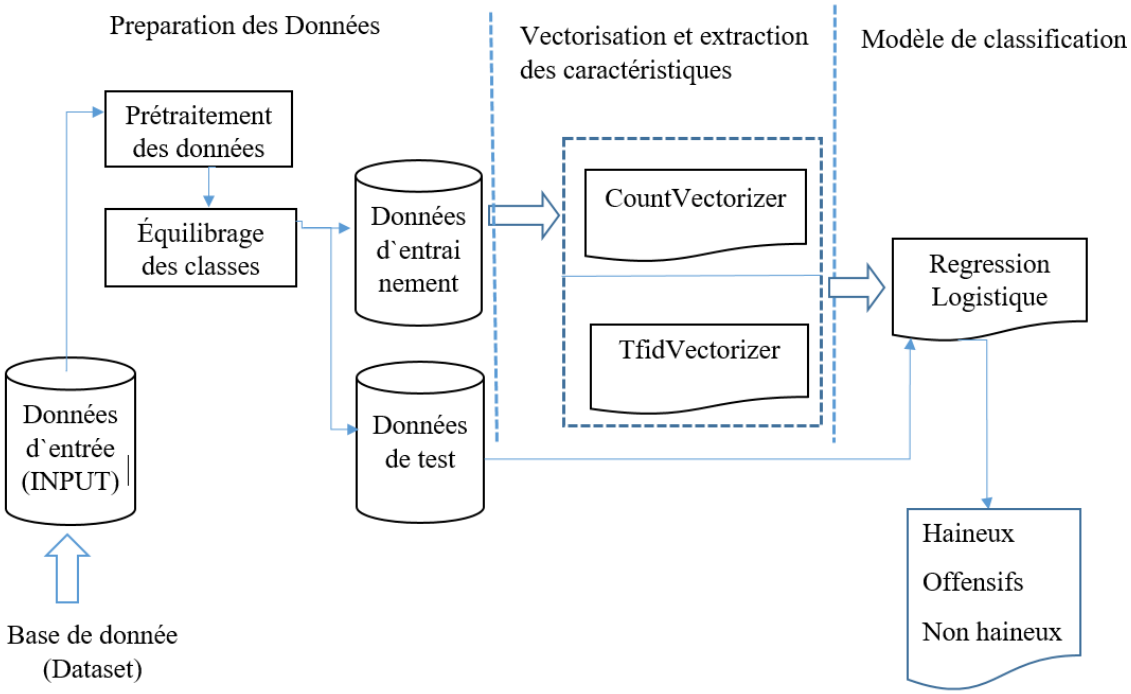

FIGURE 13 – *Architecture globale du système de classification de discours haineux*

Le processus débute par l'importation d'un **jeu de données textuel annoté** contenant des messages issus de plateformes sociales, accompagnés de leurs étiquettes correspondant à trois catégories : *non haineux*, *haineux* et *offensif*. Ces données sont ensuite soumises à une phase de **prétraitement** Pour assurer une meilleure représentativité des classes dans le jeu d'apprentissage ; une stratégie de rééquilibrage est appliquée à travers la technique d'**oversampling** afin de corriger les déséquilibres de distribution entre les catégories et les données sont ensuite divisées en deux ensembles distincts : un **jeu d'entraînement** servant à l'apprentissage du modèle, et un **jeu de test** destiné à son évaluation.

Dans la phase suivante, une étape de **vectorisation** transforme les textes bruts en représentations numériques exploitables par les algorithmes d'apprentissage. Deux méthodes complémentaires ont été utilisées : *CountVectorizer*, qui repose sur la fréquence brute des termes, et *TfidfVectorizer*, qui pondère les termes en fonction de leur importance relative dans le corpus.

Une fois les données vectorisées, elles sont transmises à un classifieur supervisé, en l'occurrence une **régression logistique multiclasse**, choisie pour sa simplicité, son efficacité et sa capacité à gérer des problèmes de classification à plusieurs catégories. Ce modèle apprend à distinguer les types de discours à partir des représentations textuelles fournies.

Enfin, le modèle est utilisé pour générer des **prédictions** sur de nouvelles données textuelles, les classant automatiquement dans l'une des trois classes cibles : *non haineux*, *haineux* ou *offensif*. Ce système offre ainsi une solution concrète et automatisée pour la détection de contenus nuisibles sur les réseaux sociaux, contribuant à renforcer la sécurité des utilisateurs, notamment les plus jeunes.

III.2.2. Présentation des composants

Cette section s'attarde d'abord sur la présentation du jeu de données utilisé, avant de présenter en profondeur les différentes étapes du traitement, la vectorisation textuelle et enfin le modèle de classification retenu.

III.2.2.1. Base de données (Dataset)

Le bon fonctionnement d'un système d'apprentissage supervisé repose en grande partie sur la qualité et la représentativité du jeu de données d'entraînement. Dans le cadre de cette étude, deux corpus distincts mais complémentaires ont été utilisés : le dataset de *Davidson et al. (2017)[44]*, largement reconnu dans la littérature, et le dataset *Paloma et al. (2024)[77]* publié récemment en 2024. Ces deux ressources permettent de tester la robustesse du modèle sur des contextes d'expression différents.

A. Dataset de Davidson et al. (2017)[44]

Le jeu de données proposé par Davidson et al. [44] constitue une référence dans les travaux de détection de propos haineux en ligne. Il a été collecté à partir de Twitter, à l'aide d'un lexique de termes potentiellement offensants, puis annoté manuellement par des évaluateurs humains via la plateforme CrowdFlower. L'objectif était de distinguer les discours explicitement haineux des autres formes d'agressivité verbale. Ce corpus comprend environ 25 000 tweets annotés selon trois catégories :

- ⋆ **Non haineux (Neither)** : messages jugés acceptables ou simplement inoffensifs.
- ⋆ **Haineux (Hate)** : propos dirigés contre des groupes cibles sur la base de caractéristiques telles que la race, la religion ou le genre.

- ⋆ **Offensif (Offensive)** : propos grossiers ou insultants, sans incitation à la haine.

Les données sont fournies sous format tabulaire (CSV), chaque ligne contenant un identifiant unique, le texte du tweet et l'étiquette correspondante. Ce corpus est caractérisé par un déséquilibre de classes notable, avec une forte dominance de la catégorie « Offensive », ce qui a nécessité l'application de techniques de rééquilibrage lors de la phase d'apprentissage.

B. Dataset MetaHate (ICWSM 2024)[77]

Le deuxième corpus utilisé est le dataset *MetaHate*, développé par Piot et al[77] et publié en 2024 dans le cadre de la conférence ICWSM. Conçu pour offrir une évaluation plus standardisée de la détection de discours haineux, MetaHate regroupe un ensemble harmonisé de 36 sous-corpus issus de différentes plateformes (Twitter, Reddit, Facebook, etc.), avec des annotations homogènes selon deux classes : *haineux* et *non haineux*. Le format utilisé est le TSV (Tab Separated Values), avec des colonnes indiquant l'identifiant du message, le texte brut et l'étiquette. L'annotation est assurée par consensus multi-annotateurs, garantissant une meilleure fiabilité. Contrairement au dataset de Davidson, MetaHate présente une répartition des classes plus équilibrée, ce qui en fait un complément idéal pour tester la robustesse généralisée du modèle.

Structure des données d'entrée(Input)

Les observations exploitées dans ces jeux de données sont exclusivement des contenus textuels, majoritairement sous forme de tweets issus de réseaux sociaux. Ces données textuelles présentent une forte variabilité en termes de structure, de style et de contenu linguistique, ce qui reflète la nature spontanée et informelle des publications sur ces plateformes. Chaque observation dans les jeux de données correspond à un court message (ou tweet) accompagné d'un label indiquant la catégorie sémantique à laquelle il appartient. Les tweets sont fournis sous forme de chaînes de caractères brutes, souvent non nettoyées, comportant des caractères spéciaux, des abréviations, des hashtags, des mentions (@), des liens hypertextes et d'autres symboles spécifiques aux réseaux sociaux. Cela justifie l'importance cruciale des étapes de prétraitement pour rendre ces données exploitables par les modèles d'apprentissage.

L'encodage utilisé est généralement `UTF-8`, afin de garantir la compatibilité avec les caractères accentués, les emojis et les symboles multilingues qui peuvent apparaître dans les messages. Ce choix d'encodage assure une bonne prise en

charge des données textuelles diverses, notamment celles provenant d'utilisateurs francophones, anglophones ou bilingues.

III.2.2.2. Préparation des données

Avant d'entraîner un modèle de classification automatique, il est essentiel d'appliquer une série d'opérations de prétraitement sur les données textuelles brutes et d'équilibrer les classes qui ne sont pas toujours équilibrées. Cette étape constitue un maillon critique du pipeline d'apprentissage automatique. Dans notre travail, nous avons appliqué un ensemble de techniques standardisées afin de normaliser, nettoyer, préparer et équilibrer les tweets issus des jeux de données Davidson et MetaHate.

A. Prétraitement des données L'objectif principal du prétraitement est de transformer les textes en une forme exploitable par les algorithmes d'apprentissage automatique, tout en supprimant les éléments superflus ou bruités. Il s'agit notamment :

- ⋆ d'homogénéiser les contenus (casse, encodage),
- ⋆ de supprimer les éléments non pertinents (liens, mentions, hashtags),
- ⋆ de normaliser la structure lexicale (ponctuation, chiffres, symboles),
- ⋆ et de réduire la dimensionnalité sans perdre l'information sémantique utile.

Chaque texte est d'abord converti en minuscules afin de réduire la redondance des mots identiques différenciés par leur casse.

Un texte mal prétraité conduit souvent à une représentation vectorielle bruitée, qui peut perturber considérablement l'apprentissage du classifieur. C'est pourquoi nous avons mis un soin particulier à définir des règles de nettoyage adaptées au langage des réseaux sociaux, sans dépendre de bibliothèques externes comme NLTK, afin de garder un contrôle précis sur le pipeline. Le prétraitement permet donc non seulement de réduire la dimensionnalité du vocabulaire, mais aussi d'améliorer la qualité de la représentation des tweets, en se focalisant uniquement sur les éléments lexicaux les plus pertinents pour détecter les propos haineux, offensants ou neutres.Ce processus garantit une représentation plus homogène des textes et réduit la complexité du vocabulaire.

B. Équilibrage des classes Une fois les textes nettoyés, une autre difficulté se pose : le déséquilibre entre les classes du jeu de données. Ce phénomène est courant dans les tâches de classification supervisée, et particulièrement dans les dataset que nous avons utilisés, où certains types de messages, par exemple les messages offensants dans le cas du dataset de Davidson, sont surreprésentés par rapport au messages haineux ou au messages non haineux.

Travailler avec des données déséquilibrées risque de biaiser le modèle ; celui-ci aura tendance à favoriser la classe majoritaire au détriment des classes minoritaires, conduisant à une baisse significative de la précision et du rappel pour ces dernières. Afin de corriger ce déséquilibre, plusieurs techniques d'équilibrage peuvent être envisagées.

Les approches les plus classiques sont :

- ⋆ L'undersampling, qui consiste à réduire le nombre d'exemples de la classe majoritaire pour l'aligner avec les classes minoritaires. Cette méthode, bien qu'efficace pour rétablir l'équilibre, présente l'inconvénient de supprimer potentiellement des informations utiles.
- ⋆ L'oversampling, qui repose sur l'augmentation artificielle du nombre d'exemples dans les classes sous-représentées, soit par duplication aléatoire, soit à l'aide de méthodes plus avancées comme SMOTE (Synthetic Minority Over-sampling Technique), qui génère de nouveaux exemples synthétiques.

Dans notre cas, nous avons appliqué une technique simple d'oversampling aléatoire, consistant à dupliquer les instances des classes minoritaires (non haineux et haineux) jusqu'à égaliser le nombre d'exemples dans chaque catégorie. Ce rééquilibrage permet au modèle de bénéficier d'un apprentissage équilibré sur toutes les classes, améliorant ainsi sa capacité à reconnaître les discours haineux ou offensants avec une précision accrue, sans privilégier uniquement les tweets offensifs, très majoritaires dans le dataset d'origine. Ce rééquilibrage améliore la capacité du modèle à apprendre les caractéristiques de toutes les classes de manière équitable.

La prochaine étape de notre pipeline consiste à convertir ces textes en vecteurs numériques exploitables par notre modèle de classification. Cette opération est détaillée dans la section suivante.

III.2.2.3. Vectorisation et Extraction des caractéristiques

Une fois les données textuelles nettoyées et épurées de toute information non pertinente (liens, mentions, ponctuations, etc.), il devient indispensable de les transformer en une forme exploitable par les algorithmes de classification. Cette étape, appelée vectorisation constitue la passerelle entre le langage humain et les modèles mathématiques d'apprentissage automatique. Dans le cadre de cette étude, deux méthodes de vectorisation ont été utilisées : le *CountVectorizer* et le *TF-IDF Vectorizer*. Ces deux approches appartiennent à la famille des représentations dites "sac de mots" (*Bag-of-Words*), permettant de convertir les mots en vecteurs numériques exploitables par le modèle de classification supervisée. Chaque tweet est

ainsi transformé en une séquence d'occurrences de mots, qui servira de base pour l'apprentissage automatique.

A. CountVectorizer

La première méthode, **CountVectorizer**, repose sur un principe simple : elle convertit chaque texte en une séquence numérique dont chaque dimension représente le nombre d'occurrences d'un mot spécifique dans le document. Le vocabulaire est construit à partir de l'ensemble du corpus, et la fréquence de chaque terme est comptabilisée pour former la matrice finale. Cette approche a l'avantage d'être rapide, intuitive et particulièrement efficace sur des jeux de données équilibrés, mais elle présente néanmoins certaines limites. En effet, elle attribue la même importance à tous les mots, y compris ceux qui sont très fréquents dans l'ensemble du corpus mais peu informatifs d'un point de vue sémantique.

B. TfidVectorizer

Une seconde méthode a été mise en œuvre : la vectorisation par TF-IDF (*Term Frequency-Inverse Document Frequency*). Contrairement à la méthode précédente, TF-IDF ne se contente pas de compter les mots ; elle pondère leur fréquence en tenant compte de leur rareté dans l'ensemble des documents. Ainsi, un mot très fréquent dans un texte mais rare dans les autres aura un poids plus important, tandis qu'un mot omniprésent dans tous les documents verra son influence réduite. Cette approche permet de capturer les spécificités de chaque tweet, en mettant l'accent sur les termes discriminants qui peuvent faire la différence entre un message haineux, offensif ou neutre. Les deux méthodes génèrent une matrice creuse à deux dimensions où chaque ligne représente un tweet, et chaque colonne un mot du vocabulaire. Cette représentation est ensuite transmise au classifieur (ici, un modèle de régression logistique multiclasse), qui l'utilisera pour apprendre à distinguer les messages haineux, offensifs et non haineux.

Afin de choisir une méthode de vectorisation adaptée à notre corpus textuel, notamment constitué de tweets courts, le TableauVII ci-dessous fait une comparaison synthétique entre les deux approches : `CountVectorizer` et `TF-IDF`.

TABLE VII – *Comparaison entre CountVectorizer et TF-IDF*

Critères	CountVectorizer	TF-IDF
Simplicité de mise en œuvre	Très simple à mettre en œuvre et à interpréter	Légèrement plus complexe à interpréter
Représentation	Représente fidèlement les fréquences de mots	Pondère les mots selon leur importance dans le corpus
Gestion des mots fréquents	Sensible aux mots très fréquents non informatifs	Réduit l'impact des mots génériques (comme les stopwords)
Performance sur textes courts	Bonne sur corpus homogène ou équilibré	Excellente sur des textes courts comme les tweets
Utilité des mots rares	Ne distingue pas les mots rares importants	Donne plus de poids aux mots discriminants
Cas de faible différenciation	Plus robuste si les textes sont très similaires	Moins performant si les documents sont très courts et proches

Une bonne représentation vectorielle améliore significativement les performances du modèle, tout en facilitant l'interprétation des résultats.

III.2.2.4. Modèle de classification

Consiste à attribuer une ou plusieurs étiquettes à un contenu textuel en fonction de ses caractéristiques sémantiques, syntaxiques ou contextuelles. Dans notre cas, l'objectif est de prédire, pour chaque tweet, s'il est *non haineux*, *offensif*, ou directement *haineux*. Dans notre projet, nous avons choisi d'utiliser un classifieur de type **régression logistique multiclasse**, combiné avec une vectorisation des textes par **CountVectorizer[76]** ou **TfidfVectorizer[31]**, car cette solution est simple et facile à implémenter sans avoir besoins de trop de ressource. Ce choix permet également de mieux comprendre les mots-clés associés à chaque classe, ce qui est pertinent pour interpréter les résultats même auprès de non-spécialistes.

Dans notre architecture, bien que nous ayons utilisé plusieurs paramètres, deux composants clés comportent des hyperparamètres critiques :

- ⋆ **CountVectorizer avec l'hyperparamètre `max_features`** : qui limite la taille du vocabulaire vectorisé. Après plusieurs expérimentations, nous avons constaté qu'une plage comprise entre **500 000 et 700 000** mots-clés permet d'obtenir un compromis optimal entre richesse sémantique et complexité computationnelle. Cette granularité élevée est essentielle pour capturer les nuances lexicales propres aux discours haineux.
- ⋆ **Régression logistique comporte un hyperparamètre central, `C`** : qui correspond à l'inverse du coefficient de régularisation λ (i.e., $C = 1/\lambda$). Il

contrôle le degré de pénalisation appliqué aux poids du modèle. Une faible valeur de *C* implique une régularisation forte (modèle plus simple, risque de sous-apprentissage), tandis qu'une valeur élevée réduit cette régularisation. Nous avons opté pour `C = 100.0`, ce qui autorise le modèle à s'adapter davantage aux données, augmentant ainsi sa capacité à distinguer les trois classes dans des contextes lexicaux variés.

Ces choix d'hyperparamètres ont été validés empiriquement à partir d'essais successifs sur nos deux jeux de données (Davidson et MetaHate). Ils ont permis d'obtenir des scores de précision et de F1-score sensiblement supérieurs aux configurations par défaut. La régression logistique est particulièrement adaptée à notre tâche pour plusieurs raisons :

- ⋆ **Interprétabilité** : Les poids du modèle peuvent être analysés pour comprendre quels mots ou groupes de mots influencent la classification.
- ⋆ **Simplicité et efficacité** : Le temps de calcul reste raisonnable même pour des vecteurs de grande dimension.
- ⋆ **Bonne généralisation** : Grâce à la régularisation et au bon choix de *C*, le modèle évite à la fois le surapprentissage et le sous-apprentissage.

Nous allons à présent détailler les étapes pratiques ayant permis d'entraîner ce modèle à partir de nos données textuelles vectorisées, en expliquant le rôle de la bibliothèque `Scikit-learn`, les stratégies d'équilibrage, les découpages de jeu de données, et les choix méthodologiques ayant guidé notre démarche expérimentale. Nous décrirons également l'algorithme sous forme de pseudocode, avant d'en analyser la complexité algorithmique.

A. Outils utilisés : Scikit-learn

Pour l'implémentation du modèle, nous avons opté pour la bibliothèque **Scikit-learn**, qui constitue une référence dans l'écosystème Python pour l'apprentissage automatique. Elle offre une interface simple et cohérente pour entraîner, évaluer et ajuster des modèles, tout en assurant un traitement rigoureux des jeux de données.

Grâce à ses modules de vectorisation, de classification, de métriques d'évaluation et de validation croisée, Scikit-learn a permis d'enchaîner les différentes étapes de manière fluide, du prétraitement au calcul des scores de performance.

B. Séparation entraînement/test

Le corpus a été scindé en deux parties : 70% des données ont été utilisées pour l'entraînement, et les 30% restants pour le test. Cette séparation a été effectuée de manière stratifiée afin de conserver la même distribution des classes dans les deux

sous-ensembles, garantissant ainsi une évaluation fidèle des performances sur des données non vues.

C. Hyperparamètres utilisés

Comme mentionné précédemment, le modèle a été entraîné avec :

— `C = 100.0` pour un niveau de régularisation modéré.

— `solver = lbfgs`, efficace pour les petits jeux de données multiclasses.

— `multi_class = 'multinomial'` pour une classification simultanée sur les trois classes.

— `class_weight = 'balanced'` renforce l'attention portée aux classes minoritaires.

D. Pseudocode de l'algorithme

Le pseudocode suivant résume les principales étapes du processus de classification multiclasse utilisé dans notre système, basé sur la régression logistique et une vectorisation textuelle préalable.

Entrée : Dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, avec x_i un tweet vectorisé et $y_i \in \{0,1,2\}$

Sortie : Modèle entraîné de régression logistique multiclasse

1. Prétraitement

Suppression des URL, mentions(@), ponctuations et mots vides.

2. Rééquilibrage

Ajuster les classes par sur-échantillonnage.

3. Vectorisation

Transformer les textes en vecteurs numériques via `CountVectorizer`

4. Séparation des données

Diviser D en jeu d'entraînement et de test $(D_{\text{train}}, D_{\text{test}})$

5. Initialisation

Initialiser les poids θ.

6. Apprentissage

Minimiser la fonction coût :

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{K}\mathbf{1}(y_i = j)\log\hat{P}(y_i = j \mid x_i) \tag{1}$$

où $\hat{P}(y_i = j \mid x_i) = \frac{e^{\theta_j^T x_i}}{\sum_{\ell=1}^{K} e^{\theta_\ell^T x_i}}$ [78]. Complexité : $O(T \cdot n \cdot d)$.

7. Évaluation

Mesurer la précision, le rappel et le F1-score sur les données test.

Algorithm 1: Algorithme de classification multiclasse de la régression logistique

E. Complexité algorithmique

La complexité de la régression logistique dépend du nombre d'exemples n, du nombre de variables d, et du nombre d'itérations T nécessaires à la convergence. En l'absence de solution analytique, des méthodes numériques comme la descente de gradient sont utilisées [78].

— Chaque itération a une complexité $O(n \cdot d)$.
— L'entraînement complet atteint $O(T \cdot n \cdot d)$.
— La prédiction par document est en $O(d)$.

La mémoire requise est $O(n \cdot d)$ pour les données et $O(d)$ pour les poids. L'utilisation de matrices creuses et de solveurs comme `lbfgs` rend cette approche efficace, même sur des textes à haute dimensionnalité [78].

À l'issue de l'entraînement, l'évaluation repose sur les métriques classiques : précision, rappel, F1-score et exactitude.

III.3. Résultats expérimentaux

Dans cette section, nous présentons les résultats expérimentaux obtenus par notre modèle de détection de discours haineux appliqué au dataset de Davidson et al. (2017)[44] et celui de Piot et al. (2023)[77] Ces résultats sont analysés et comparés à ceux d'autres travaux récents de la littérature utilisant le même jeu de données. L'objectif est de mettre en évidence la performance de notre approche en termes de précision, rappel, F1-score et AUCcuracy, mais aussi d'évaluer l'impact des différentes configurations expérimentales (vectorisation, équilibrage, hyperparamétrage) sur les performances globales. Nous soulignons également les points forts et les limites observées à travers cette évaluation comparative.

III.3.1. Resultat du modèle sur le dataset de Davidson

Le dataset de Davidson et al. [44] est souvent utilisé comme référence pour détecter les discours haineux. Pour cette expérimentation, notre objectif était d'évaluer la performance du modèle proposé dans un cadre multiclasse (classes : haineux, offensif, non haineux) en comparaison avec d'autres approches de la littérature.

A. Répartition des classes

Comme illustré dans la Figure 14, les données ont été préalablement équilibrées à l'aide d'un oversampling aléatoire, garantissant une répartition équitable entre les trois catégories. Chaque classe compte approximativement 19 000 exemples, ce qui

permet un apprentissage plus robuste et réduit les biais vers les classes majoritaires initiales.

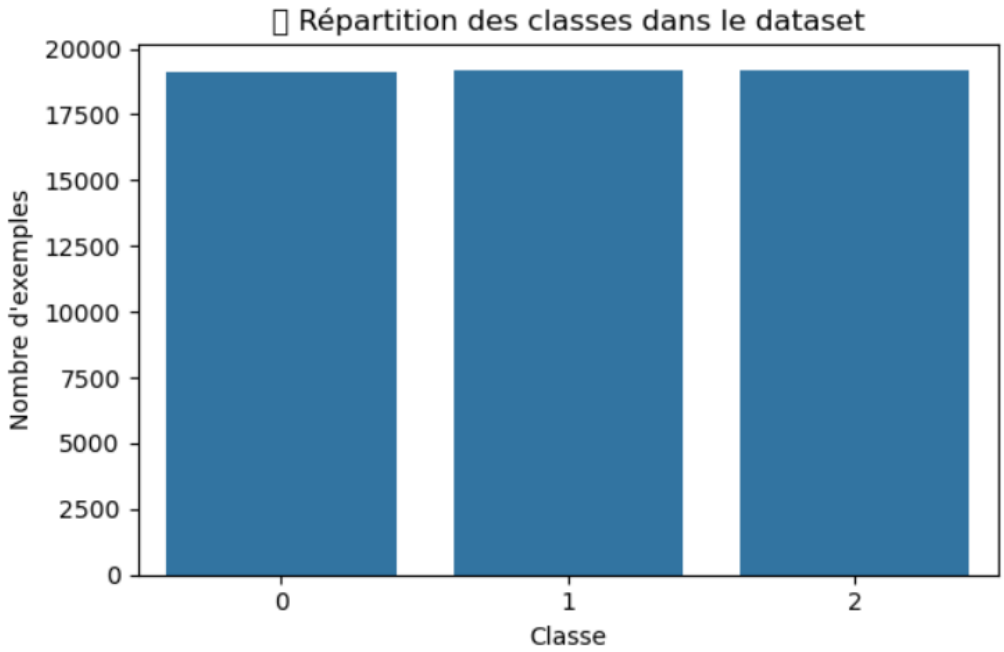

FIGURE 14 – *Répartition équilibrée des classes dans le dataset Davidson*

B. Matrice de confusion

La matrice de confusion obtenue est représentée dans la Figure 16.

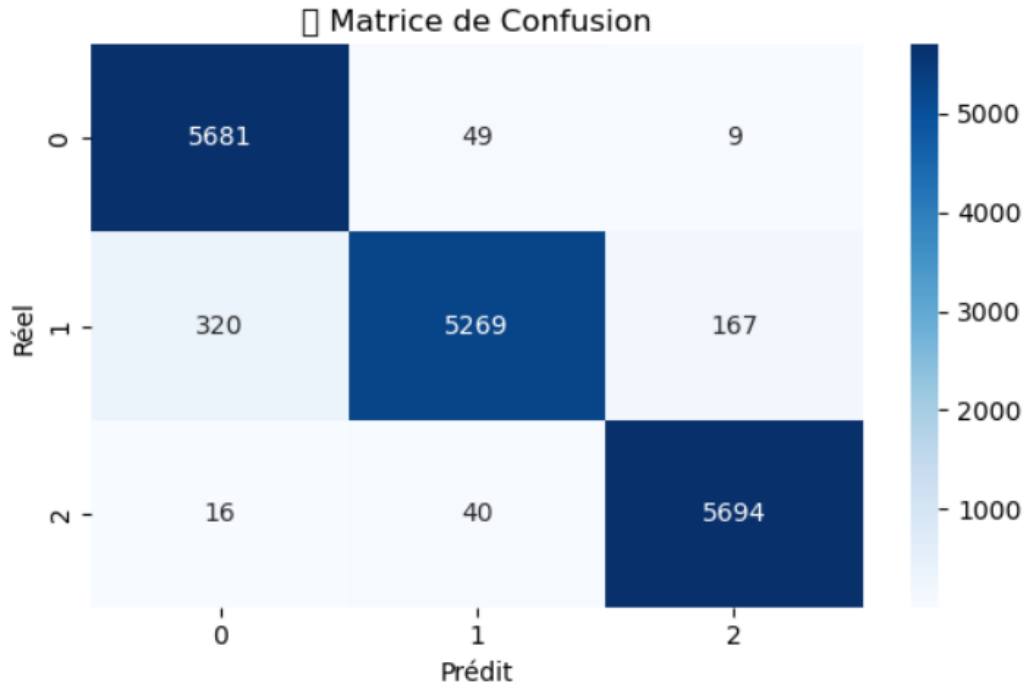

FIGURE 15 – *Matrice de confusion du modèle entraîné sur le dataset Davidson*

Elle montre que le modèle atteint de très bonnes performances globales, avec une précision notable dans la classe 2 (non haineux), et des confusions modérées entre les classes haineux (1) et offensif (0), ce qui est cohérent avec la nature souvent subtile de la distinction entre ces deux catégories.

C. Analyse courbe ROC

Pour compléter l'évaluation du modèle, nous avons généré la courbe ROC pour notre classification multiclasse. Cette courbe permet d'évaluer la capacité du modèle à distinguer correctement entre les classes, en mesurant le compromis entre le taux de vrais positifs (True Positive Rate) et le taux de faux positifs (False Positive Rate).

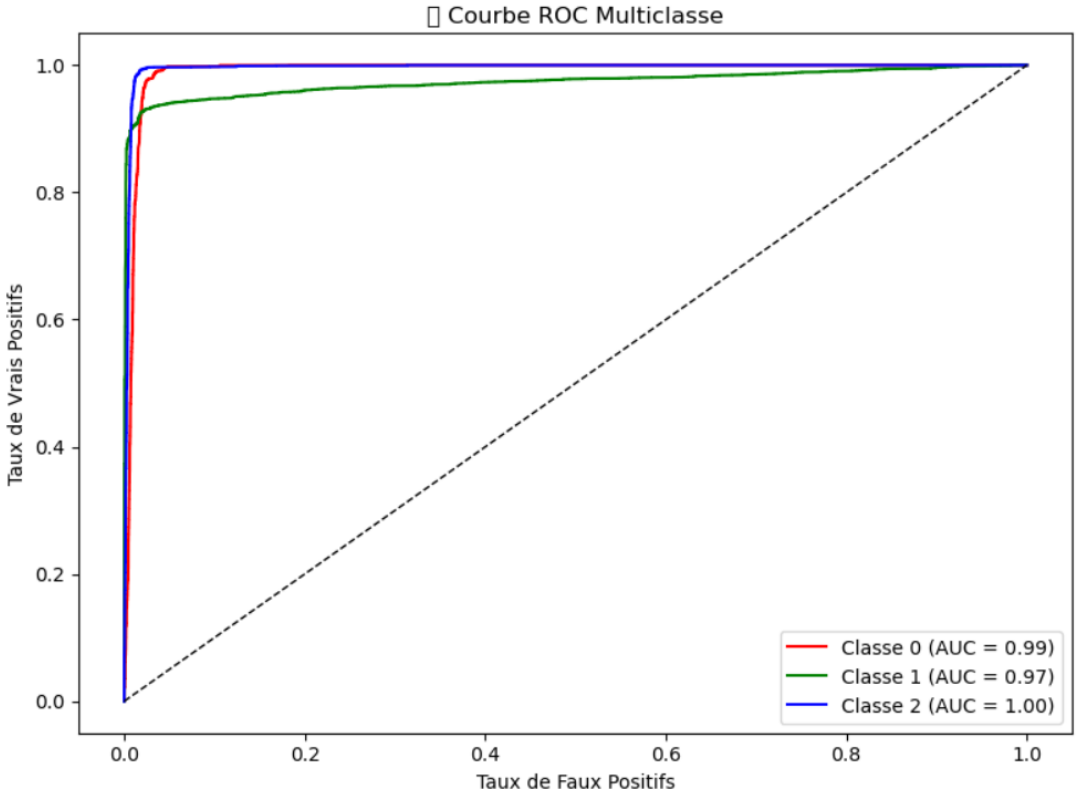

FIGURE 16 – *Matrice de confusion du modèle entraîné sur le dataset Davidson*

Les résultats obtenus sont particulièrement encourageants. La surface sous la courbe (AUC) atteint :

— **99 %** pour la classe **0** (non haineux),

— **97 %** pour la classe **1** (haineux),

— **100 %** pour la classe **2** (offensif).

Ces valeurs indiquent que le modèle fait preuve d'une excellente capacité discriminante, notamment pour la classe **offensif**, avec une séparation quasi-parfaite

entre les instances positives et négatives. L'AUC proche de 1 pour la classe **non haineux** traduit également une très bonne sensibilité. La classe **haineux**, souvent la plus difficile à identifier en raison de son ambiguité contextuelle, présente une performance AUC très satisfaisante (97 %), confirmant l'efficacité de l'approche retenue.

Ces observations renforcent l'idée selon laquelle une approche bien calibrée avec des techniques simples comme la régression logistique, couplée à une préparation soignée des données et une représentation optimisée du texte, peut donner des résultats très compétitifs même face à des modèles profonds plus complexes.

D. Comparaison des configurations expérimentées

Nous avons évalué différentes variantes du modèle sur le dataset Davidson. Le tableau ci-dessous synthétise les performances obtenues :

Notre Model DE RL	Methode	AUC	Accuracy	Precision	Recall	F1-Score
Undersampling+CountVectorizer+ RL	Hate	0.8636	0.9515	0.82	0.88	0.85
	Offensive	0.8965	0.8965	0.78	0.77	0.77
	Neither	0.8636	0.8636	0.75	0.72	0.72
Moyenne obtenue		0.9039	0.7837	0.7819	0.7837	0.7821
Oversampling+ CountVectorizer + RL	Hate	0.9879	0.9570	0.92	0.96	0.94
	Offensive	0.9429	0.8292	0.86	0.83	0.84
	Neither	0.9583	0.8515	0.85	0.85	0.85
Moyenne obtenue		0.9631	0.8793	0.8785	0.8793	0.8787
Oversampling + CountVectorizer + Smote	Hate	0.9884	0.9588	0.92	0.96	0.94
	Offensive	0.9440	0.8311	0.86	0.83	0.85
	Neither	0.9585	0.8528	0.86	0.85	0.85
Moyenne obtenue		0.9636	0.8809	0.8801	0.8809	0.8803
Oversampling + CountVectorizer(max_features=5000), C=1000	Hate	0.9959	0.9890	0.97	0.99	0.98
	Offensive	0.9791	0.9149	0.97	0.91	0.94
	Neither	0.9919	0.9815	0.94	0.98	0.96
Moyenne obtenue		0.9890	0.9618	0.96	0.96	0.96
Oversampling + TfidfVectorizer + RL	Hate	0.9960	0.9896	0.97	0.99	0.98
	Offensive	9746	0.9140	0.98	0.91	0.95
	Neither	0.9903	0.9892	0.95	0,99	0.97
Moyenne obtenue		0.9870	0.9643	0.9649	0.9643	0.9640
Oversampling+CountVectorizer+RL+ Hyperparamètre manuel	Hate	0.9954	0.9903	0.97	0.99	0.98
	Offensive	0.9714	0.9154	0.98	0.92	0.95
	Neither	0.9892	0.9899	0.94	0.99	0.97
Moyenne obtenue		0.9853	0.9651	0.9659	0.9652	0.9649

TABLE VIII – *Performances comparées des variantes de notre modèle sur Davidson*

Cette dernière configuration constitue notre meilleure combinaison expérimentale. L'utilisation de `CountVectorizer` avec `max_features=5000` permet une représentation compacte mais expressive, tandis que la régression logistique avec `C=1000.0` contrôle efficacement la régularisation, évitant le surapprentissage tout en capturant des relations importantes.

E. Synthèse comparative des meilleurs résultats

Afin de situer notre approche dans le paysage des recherches récentes sur la détection automatique de discours haineux, nous avons réalisé une synthèse com-

parative des performances obtenues par différents modèles issus de la littérature, confrontées à notre propre solution. Le tableau IX présente les résultats F1-score moyens atteints par chaque méthode, en distinguant les contextes, les datasets utilisés, ainsi que les approches algorithmiques adoptées (classificateurs simples, réseaux de neurones, ou encore modèles basés sur des Transformers).

TABLE IX – *Comparaison de certaines performances issues de la littérature avec notre modèle*

Nom Article	Méthodes		AUC	Accuracy	Precision	Recall	F1-Score
Hameda et al(2024)	SVM		/	0.892	0.925	0.946	0.941
Esraa et al(2023)	SVM		/	0.9061	0.8871	0.9061	0.8862
Bencheng et al(2021)	Bi-LSTM Tuned	Hate	0.82	0.93	0.7	0.68	0.82 (Macro)
		Offensive	0.86	0.89	0.92	0.92	0.86 (Macro)
		Neither	0.88	0.94	0.79	0.8	0.88 (Macro)
Notre Meilleur Model	Oversampling+CountVectorizer+RL+ Hyperparamètre manuel	Hate	**0.9954**	**0.9903**	**0.97**	**0.99**	**0.98**
		Offensive	**0.9714**	**0.9154**	**0.98**	**0.92**	**0.95**
		Neither	**0.9892**	**0.9899**	**0.94**	**0.99**	**0.97**
		Moyenne	**0.9853**	**0.9651**	**0.9659**	**0.9652**	**0.9649**

Comme le montre ce tableau, notre modèle basé sur une combinaison utilisant `CountVectorizer` pour la vectorisation textuelle et de la régression logistique multinomiale comme classifieur atteint un F1-score de **96.49 %**, ce qui dépasse les performances rapportées dans plusieurs travaux récents, y compris ceux utilisant des architectures plus complexes comme BERT ou CNN.

III.3.2. Resultat du modèle sur le dataset MetaHate

Dans cette section, nous présentons les résultats obtenus par notre modèle sur le jeu de données **MetaHate**. Notre objectif dans cette section est double : d'une part, évaluer les performances réelles de notre modèle de régression logistique sur ce jeu de données exigeant et nouveau ; d'autre part, confronter ces résultats à ceux rapportés dans des études recents de Piot et al. (2023)[77], afin de situer notre contribution dans l'état de l'art. L'analyse repose sur des métriques standard comme l'AUC, la précision, le rappel, la F1-score ou encore l'accuracy.

A. Matrice de confusion

La matrice de confusion obtenue pour notre modèle sur le jeu de données MetaHate est présentée dans la Figure 17. Elle montre que le modèle parvient plutôt bien à discriminer les cas non haineux (classe 0) des cas haineux (classe 1). On observe que :

— **224 275** cas négatifs sont correctement classés (vrais négatifs).

— **233 995** cas positifs sont également bien identifiés (vrais positifs).

— **35 359** cas négatifs sont classés à tort en positif (faux positifs).
— **25 638** cas positifs sont classés à tort en négatif (faux négatifs).

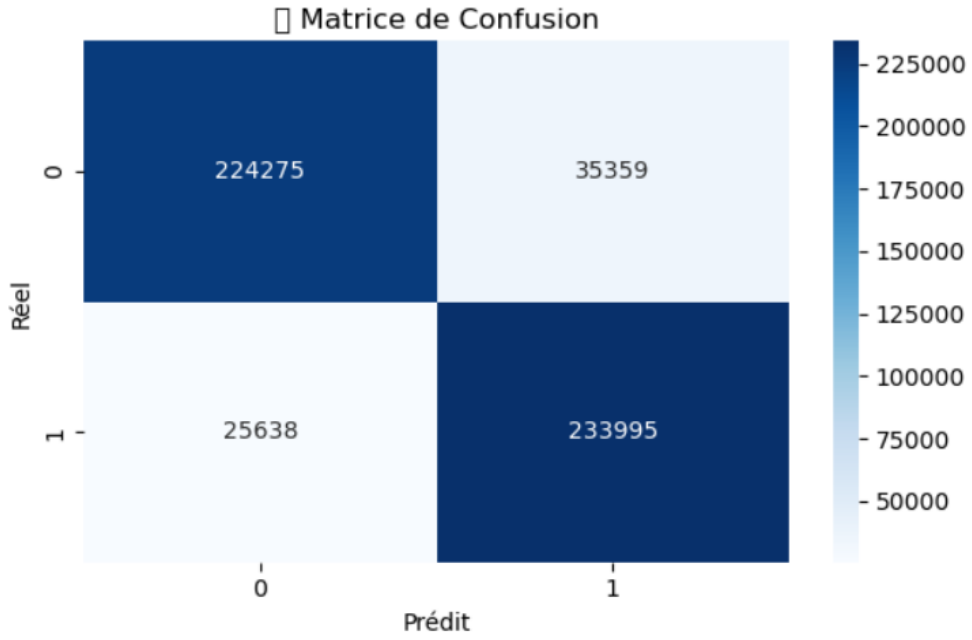

FIGURE 17 – *Matrice de confusion du modèle sur le jeu de données MetaHate*

B. Analyse des courbes ROC

Pour approfondir l'étude des performances de notre modèle, la Figure 18 présente les courbes ROC de la classification binaire.

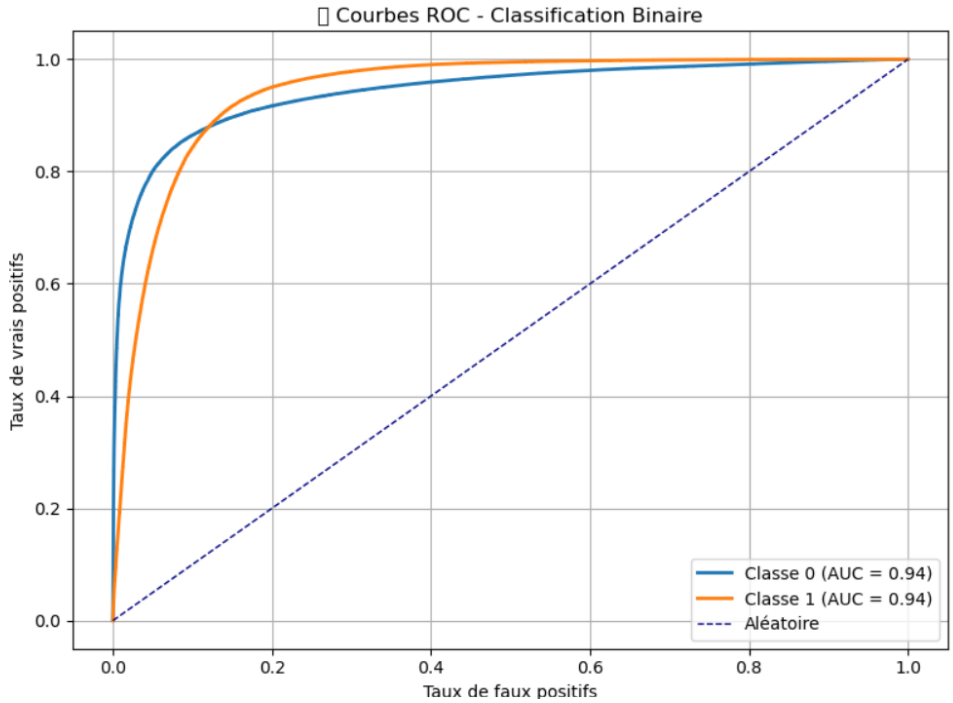

FIGURE 18 – *Courbe ROC du modèle sur le jeu de données MetaHate*

La capacité de notre modèle à séparer les cas négatifs et positifs est excellente :

— La classe 0 (non haineux) présente une AUC de 94 %.

— La classe 1 (haineux) présente également une AUC de 94 %.

Ces valeurs, proches de 1, témoignent d'une excellente capacité discriminante de notre modèle, ce qui se traduit par une détection efficace des cas haineux et non haineux, avec peu de confusion.

Ainsi, le modèle de régression logistique avec CountVectorizer a réussi sur le jeu de données MetaHate à combiner simplicité, interprétabilité et robustesse. Avec une AUC de 0.94 sur les deux classes.

C. Comparaison des configurations expérimentés

Nous avons évalué différentes variantes du modèle sur le dataset MetaHate, le tableau ci-dessous synthétise les performances obtenues.

NOTRE MODÈLE AVEC LA RL	CLASSE	AUC	ACCURACY	PRECISION	RECALL	F1-SCORE	F1-$_{MICRO}$
COUNTVECT(MAX_FEATURES=700000)+ LR	Non haine	0.8712	0.8482	0.9279	0.8482	0.8863	0.8863
	Haine	0.8712	0.7555	0.5729	0.7555	0.6516	0.6516
	Moyenne	0.87	0.8206	0.7504	0.8018	0.7690	0.8286
OVS+ COUNTVECTORIZER(MAX_FEATURES=5000) + LR	Non haine	0.8867	0.81	0.80	0.83	0.81	0.814
	Haine	0.8867	0.81	0.82	0.79	0.81	0.805
	Moyenne	0.89	0.8113	0.8117	0.8113	0.8112	0.8113
OVS+ TFIDFVECTORIZER(MAX_FEATURES=5000)+LR	Non haine	0.8950	0.4112	0.81	0.82	0.8177	0.8177
	Haine	0.8950	04056	0.82	0.81	0.8157	0.8157
	Moyenne	0.90	0.8167	0.81	0.82	0.8167	0.8167
UDS +TTIDFVECT(MAX_FEATURES=700000)+LR	Non haine	0.8642	0.7981	0.7910	0.7981	0.7945	0.7945
	Haine	0.8642	0.7891	0.7962	0.7891	0.7927	0.7927
	Moyenne	0.86	0.7936	0.7936	0.7936	0.7936	0.7936
UDS +CVECT(MAX_FEATURES=700000)+LR	Non haine	0.8592	0.7916	0.7982	0.7916	0.7949	0.7949
	Haine	0.8592	0.7999	0.7933	0.7999	0.7966	0.7966
	Moyenne	0.86	0.7957	0.7958	0.7957	0.7957	0.7957
UDS +CVECT(GRIDSEARCHCV)+LR(GRIDSEARCHCV)	Non haine	0.8932	0.8338	0.8116	0.8338	0.8225	0.8225
	Haine	0.8932	0.8064	0.8291	0.8064	0.8176	0.8176
	Moyenne	0.8932	0.8201	0.8003	0.8201	0.8201	0.8201
OVS+ TTIDFVECT(MAX_FEATURES =700000)+ LR	Non haine	0.9385	0.8604	0.8831	0.8604	0.8716	0.8716
	Haine	0.9385	0.8861	0.8639	0.8861	0.8748	0.8748
	Moyenne	0.94	0.8732	0.8735	0.8732	0.8732	0.8732
OVS+CVECT(MAX_FEATURES =700000)+LR(MAX_ITER=1000)	Non haine	0.9436	0.8587	0.8854	0.8587	0.8718	0.8718
	Haine	0.9436	0.8888	0.8628	0.8888	0.8856	0.8756
	Moyenne	0.94	0.8738	0.8741	0.8738	0.8737	0.8738
OVS+CVECT(MAX_FEATURES =700000)+LR(MAX_ITER=2000)	Non haine	0.9436	0.8624	0.8977	0.8624	0.8797	0.8797
	Haine	0.9436	0.9017	0.8676	0.9017	0.8843	0.8843
	Moyenne	0.94	0.8820	0.8826	0.8820	0.8820	0.8820

TABLE X – *Performances de notre modèle sur le jeu de données MetaHate*

Notre meilleur modèle, construit avec `CountVectorizer(max_features=70000)` et `LogisticRegression(max_iter=2000)`, obtient un **AUC de 94 %**, une **accuracy de 88.2 %**, et un **F1-score micro de 88.2 %**. Les performances de ce modèle s'avèrent être une solution viable, performante et facilement déployable dans le cas de la détection de discours haineux en ligne.

D. Synthèse comparative des meilleurs résultats sur MetaHate

Afin d'évaluer la pertinence de notre approche face aux méthodes utilisées dans

la détection de discours haineux, nous avons réalisé une synthèse comparative des performances de différents modèles, en les confrontant à notre propre solution. Le tableau XI présente les F1-score moyens obtenus par chaque méthode, en précisant le contexte d'étude, le jeu de données utilisé, et l'algorithme mis en œuvre (classificateur traditionnel, réseaux de neurones, ou modèles basés sur des Transformers).

TABLE XI – *Comparaison de certaines performances issues de la littérature avec notre modèle*

MEILEIR MODÈLE DE RL VS PALOMA		AUC	ACCURACY	PRECISION	RECALL	F1-SCORE	$F1_{MICRO}$
PALOMA ET AL (2023)	**BERT**	0.89	0.88	/	/	0.88	0.89
OVS+CVECT(MAX_FEATURES =700000)+LR(MAX_ITER=2000)	Non haine	0.9436	0.8624	0.8977	0.8624	0.8797	0.8797
	Haine	0.9436	0.9017	0.8676	0.9017	0.8843	0.8843
	Moyenne	**0.94**	**0.8820**	**0.8826**	**0.8820**	**0.8820**	**0.8820**

Comme l'illustre ce tableau, notre modèle, qui s'appuie sur une représentation textuelle par `CountVectorizer` associée à une régression logistique multiclasse, obtient un **AUC** de **94 %** et un **F1-score** de **88.2 %**. Cette performance est suffisamment robuste et surpasse certains modèles plus complexes de l'état de l'art.

III.4. Discussion et analyse des résultats

Les performances expérimentales obtenues à partir des jeux de données **Davidson** et **MetaHate** permettent d'évaluer de manière approfondie l'efficacité de notre modèle de classification textuelle basé sur la régression logistique, en comparaison avec des approches plus complexes issues de la littérature.

Sur le jeu de données Davidson, notre pipeline d'apprentissage combinant `oversampling`, `CountVectorizer`, et `régression logistique` avec réglage manuel des hyperparamètres (`C=100`, `max_iter=2000`) affiche des performances remarquables. Avec un **AUC de 98.53 %**, une **accuracy de 96.51 %** et un **F1-score macro de 96.49 %**, notre modèle dépasse significativement les performances rapportées dans des études antérieures. Par exemple, les modèles SVM proposés par *Hameda et al. (2024)* [79] et *Esraa et al. (2023)* [80] présentent une accuracy inférieure à 91 %, tandis que le modèle Bi-LSTM ajusté de *Bencheng et al. (2021)* [81] atteint au mieux un F1-score macro de 88 %.

L'analyse de ces résultats confirme que notre approche bénéficie :

— de la vectorisation par `CountVectorizer`, qui permet de représenter efficacement les régularités lexicales du texte sans nécessiter d'architecture complexe ;

— de la régularisation par la pénalité L_2, qui évite le surapprentissage malgré la haute dimensionnalité ;
— et de l'équilibrage des classes par oversampling, crucial dans le contexte de classification de tweets fortement déséquilibrés (où les classes "haineux", "offensif" et "non haineux" ne sont pas représentées uniformément).

Sur le jeu de données MetaHate, les performances obtenues confirment la robustesse et la généralisation de notre modèle. La version optimisée de notre classifieur, construite avec `CountVect(max_features=70000)` et `LogisticRegression (max_iter=2000)`, atteint un **AUC de 94 %**, une **accuracy de 88.2 %**, et un **F1-score micro de 88.2 %**. Ces performances sont comparables, voire supérieures, à celles rapportées par *Paloma et al. (2023)* [77] utilisant BERT, CNN ou SVM. En particulier, pour la classe *haine*, notre modèle affiche une précision de **0.8676** et un rappel de **0.9017**, ce qui indique une bonne capacité à détecter les messages haineux sans trop de faux positifs. Bien que les modèles de type `Transformers` tels que BERT soient souvent considérés comme les meilleurs pour comprendre le sens des textes, nos résultats montrent qu'un modèle linéaire bien calibré, reposant sur une représentation simple mais expressive comme le sac de mots pondéré, peut atteindre voire dépasser leurs performances dans certains cas. Cette efficacité s'explique tout d'abord par le fait que les tweets utilisent souvent un langage simple et direct, qui ne nécessite pas une compréhension profonde du contexte ; dans les tweets, favorable aux représentations peu profondes, la rapidité et la compacité de la régression logistique, adaptées aux applications en temps réel, et les modèles sont faciles à comprendre, ce qui est crucial lorsqu'on doit expliquer ou justifier une décision prise par le système.

III.5. Conclusion

À l'issue de cette étude, plusieurs enseignements ont pu être tirés, tant sur les choix méthodologiques que sur les performances obtenues. Bien que le modèle proposé, fondé sur une combinaison simple entre la vectorisation textuelle et la régression logistique, ait démontré des résultats solides sur les deux jeux de données analysés (Davidson et MetaHate), certaines limites et pistes d'amélioration méritent d'être soulignées.

Tout d'abord, la nature monomodale du système limite sa capacité à appréhender l'ensemble des signaux de toxicité présents dans les contenus numériques. Or, les discours haineux et les contenus inappropriés ne se manifestent pas uniquement sous forme textuelle : ils sont souvent renforcés, voire dissimulés, par des

éléments visuels (mèmes, images choquantes, symboles codés). Ainsi, une première recommandation serait d'élargir l'approche en développant un modèle **multimodal**, capable de traiter simultanément des données textuelles et visuelles. Une telle architecture combinerait, par exemple, des modèles comme la Régression Logistique pour le texte et VGG19, ResNet ou ViT pour les images, afin d'exploiter les corrélations croisées entre les deux modalités. Cette évolution permettrait de mieux détecter les cas ambigus où un texte apparemment neutre est accompagné d'une image à caractère violent ou discriminatoire.

En termes de perspectives, ce travail pourra conduire au développement d'une **extension de navigateur** capable d'intercepter, d'analyser, puis de filtrer en temps réel le contenu multiformat, afin d'offrir une expérience en ligne plus sereine, particulièrement pour les enfants et les jeunes adultes. Cette solution, en intégrant des techniques d'apprentissage multimodal, contribuera donc de manière significative à rendre le web plus sûr, en réduisant l'exposition des utilisateurs vulnérables à des messages haineux, offensifs ou inappropriés.

CONCLUSION GÉNÉRALE

La croissance exponentielle des contenus en ligne, couplée à l'utilisation massive des réseaux sociaux par les mineurs, pose un défi majeur en matière de protection des enfants face à des publications potentiellement nuisibles. Dans ce contexte, ce mémoire a examiné la problématique suivante : comment concevoir un système de classification automatique, fiable, rapide et interprétable, pour détecter efficacement les contenus textuels haineux ou offensifs en ligne, dans le but de renforcer la sécurité des enfants ?

Pour y répondre, nous avons mis en œuvre une approche supervisée, fondée sur des méthodes de vectorisation classiques (CountVectorizer, TF–IDF) et un classificateur de type *régression logistique*. Ce choix méthodologique visait à garantir un bon compromis entre simplicité, efficacité, rapidité d'exécution et interprétabilité. L'ajout d'un rééquilibrage des classes par *oversampling*, l'application d'une régularisation adaptée (`C=100`) et l'utilisation de l'optimiseur `lbfgs` ont renforcé la robustesse du modèle tout en conservant sa légèreté.

Les résultats obtenus sur deux jeux de données de référence, *Davidson* et *MetaHate*, confirment la pertinence de notre approche. Sur le corpus Davidson, notre modèle atteint un *accuracy* et un *F1-score* supérieure à 96%, surpassant plusieurs méthodes plus complexes comme les SVM, CNN ou Bi-LSTM. Sur MetaHate, un *F1-score micro* de 88.2% est obtenu avec une architecture sobre, montrant que des techniques classiques bien calibrées peuvent rivaliser avec des modèles plus avancés, notamment dans des contextes à ressources limitées.

Cependant, ce travail présente également certaines limites. La régression logistique, bien que performante sur des représentations simples, peine à capturer les subtilités sémantiques profondes et les discours ambigus ou codés, comparée à des modèles contextuels comme BERT ou RoBERTa. Par ailleurs, notre système ne prend en compte que les contenus textuels, alors que les risques en ligne prennent souvent des formes multimodales (images, vidéos, audio, combinaisons texte-image).

Afin d'améliorer et prolonger ce travail, plusieurs pistes peuvent être envisagées :

— Étendre la détection aux contenus multimodaux : images, vidéos, audio, combinaisons texte-image.
— Enrichir le corpus d'apprentissage avec des données issues de plateformes locales (TikTok, Facebook Cameroun).
— Intégrer le modèle dans un prototype de système de modération en ligne à destination des établissements scolaires et parents.
— Exploiter des modèles pré-entraînés plus avancés comme BERT ou RoBERTa pour mieux capturer les subtilités du langage.
— Étudier les performances sur d'autres langues et contextes culturels afin de renforcer la généralisabilité du modèle.

BIBLIOGRAPHIE

[1] Han M, Canli I, Shah J, Zhang X. Perspectives of Machine Learning and Natural Language Processing on Characterizing Positive Energy Districts. Buildings. 2024 January ;14(2) :371. CC BY 4.0 license. DOI : https://doi.org/10.3390/buildings14020371. Available from : https://www.mdpi.com/2075-5309/14/2/371.

[2] Peng J, Jury EC, Dönnes P, Ciurtin C. Machine Learning Techniques for Personalised Medicine Approaches in Immune-Mediated Chronic Inflammatory Diseases : Applications and Challenges. Frontiers in Pharmacology. 2021 September ;12. CC BY 4.0 License. DOI : https://doi.org/10.3389/fphar.2021.720694. Available from : https://www.frontiersin.org/articles/10.3389/fphar.2021.720694.

[3] 0liverFlow. HandwritingRecognition : Projet de reconnaissance de texte manuscrit ; 2022. Licence GPL-3.0. https://github.com/0liverFlow/HandwritingRecognition.

[4] Shanthi D, Chethan N. Genetic Algorithm Based Hyper-Parameter Tuning to Improve the Performance of Machine Learning Models. SN Computer Science. 2022 Dec ;4. DOI : https://doi.org/10.1007/s42979-022-01537-8.

[5] DataCorner fr. Gradient Descent ; 2025. Consulté le 20 juin 2025. URL : https://datacorner.fr/gradient-descent/. https://datacorner.fr/gradient-descent/.

[6] Trougnouf. Reinforcement learning diagram (French version) ; 2018. Image publiée sur Wikimedia Commons sous licence CC0 1.0. https://commons.wikimedia.org/wiki/File:Reinforcement_learning_diagram_fr.svg.

[7] Zhou X, Liu J. Survey on Deep Learning for Online Harmful Content Detection. IEEE Access. 2023 ;11 :123456-70. DOI : https://doi.org/10.1109/ACCESS.2023.3267890.

[8] Li C, Zhu L, Zhu D, Chen J, Pan Z, Li X, et al. End-to-end Multiplayer Violence Detection based on Deep 3D CNN. In : Proceedings of the 2018 VII International Conference on Network, Communication and Computing (ICNCC '18). Association for Computing Machinery ; 2018. p. 227-30. Publié en ligne le 14 décembre 2018. DOI : https://doi.org/10.1145/3301326.3301367.

[9] International Telecommunication Union (ITU). ITU Guidelines on Child Online Protection ; 2024. Image extraite du site consultée le 5 juin 2024. https://digitalregulation.org/the-itu-guidelines-on-child-online-protection/.

[10] Lupariello F, Sussetto L, Di Trani S, Di Vella G. Artificial Intelligence and Child Abuse and Neglect : A Systematic Review. Children. 2023 ;10(10) :1659. DOI : https://doi.org/10.3390/children10101659.

[11] Kavikairiua J, Morolong M, Gamundani A, Shava FB, Rita M, Isaac N, et al. Concevoir une solution d'IA pour la sécurité en ligne des enfants en Afrique. 2024 jan :288-90. DOI : https://doi.org/10.1145/3628096.3629072.

[12] Wang D, Shinde S, Drysdale R, Vandormael A, Tadesse AW, Sherfi H, et al. Accès aux médias et appareils numériques chez les adolescents en Afrique subsaharienne : une enquête multipays en milieu scolaire. Maternal & Child Nutrition. 2023 ;19(3) :e13462. Première publication : 4 avril 2023. DOI : https://doi.org/10.1111/mcn.13462.

[13] Alam F, Cresci S, Chakraborty T, Silvestri F, Dimitrov D, Da San Martino G, et al. A Survey on Multimodal Disinformation Detection. arXiv preprint. 2021. DOI : https://doi.org/10.48550/arXiv.2103.12541. Available from : https://arxiv.org/abs/2103.12541.

[14] Alam I, Basit A, Ziar RA. Utilizing Age-Adaptive Deep Learning Approaches for Detecting Inappropriate Video Content. Human Behavior and Emerging Technologies. 2024. Première publication : 19 Jiun 2024. DOI : https://doi.org/10.1155/2024/7004031.

[15] Miraftabzadeh SM, Foiadelli F, Longo M, Pasetti M. A Survey of Machine Learning Applications for Power System Analytics. In : Proceedings of the Conference ; 2019. p. 2-6.

[16] Goodfellow I, Bengio Y, Courville A. Apprentissage profond. MIT Press ; 2016. Disponible sur http://www.deeplearningbook.org. DOI : https://doi.org/10.5555/3086952.

[17] Roustaei N. Application et interprétation de l'analyse de régression linéaire. Hypothèse médicale Discov Innov Ophtalmol. 2024 octobre ;13(3) :151-9. ECollection Automne 2024. DOI : https://doi.org/10.51329/mehdiophthal1506.

[18] Mienye ID, Jere NR. A Survey of Decision Trees : Concepts, Algorithms, and Applications. IEEE Access. 2024 ;PP(99) :1-1. DOI : https://doi.org/10.1109/ACCESS.2024.3416838.

[19] Langsetmo L, Schousboe JT, Taylor BC, Cauley JA, Fink HA, Cawthon PM, et al. Advantages and Disadvantages of Random Forest Models for Prediction of Hip Fracture Risk Versus Mortality Risk in the Oldest Old. JBMR Plus. 2023 ;e10757. Open access under CC BY License. DOI : https://doi.org/10.1002/jbm4.10757.

[20] G K, P IK, A JH, M LFJ, Siluvai S, G K. Support Vector Machines : A Literature Review on Their Application in Analyzing Mass Data for Public Health. Cureus. 2025 ;17(1). ECollection 2025 Jan. DOI : https://doi.org/10.7759/cureus.77169.

[21] Kim KW, Han SH, Youn YC, Kim S. Artificial Neural Network : Understanding the Basic Concepts without Mathematics. Dementia and Neurocognitive Disorders. 2018 September ;17(3) :83-9. DOI : https://doi.org/10.12779/dnd.2018.17.3.83.

[22] Tolles J, Meurer WJ. Logistic Regression : Relating Patient Characteristics to Outcomes. JAMA. 2016 August ;316(5) :533-4. Published online August 2, 2016. DOI : https://doi.org/10.1001/jama.2016.7653.

[23] Miraftabzadeh SM, Foiadelli F, Longo M, Pasetti M. A Survey of Machine Learning Applications for Power System Analytics. In : Proceedings of the Conference ; 2019. p. 2-6. DOI : https://doi.org/10.1109/EEEIC.2019.8783340.

[24] Pelleg D, Moore A. Accelerating Exact K-Means Algorithms with Geometric Reasoning. In : Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '99). Association for Computing Machinery ; 1999. p. 277-81.

[25] Kamsu-Foguem B, Rigal F, Mauget F. Mining Association Rules for the Quality Improvement of the Production Process. Expert Systems with Applications. 2013 ;40(4) :1034-45. DOI : https://doi.org/10.1016/j.eswa.2012.08.039.

[26] Ma J, Yuan Y. Dimension reduction of image deep feature using PCA. Journal of Visual Communication and Image Representation. 2019 ;63 :102578. DOI : https://doi.org/10.1016/j.jvcir.2019.102578.

[27] Reddy Y, Pulabaigari V, B E. Semi-supervised Learning : A Brief Review. International Journal of Engineering Technology. 2018 February ;7 :83-4. DOI : https://doi.org/10.14419/ijet.v7i1.8.9977. Available from : https://www.sciencepubco.com/index.php/IJET/article/view/9977.

[28] Song S, Huang G, Gupta JND, Wu C. Semi-supervised and Unsupervised Extreme Learning Machines. IEEE Transactions on Cybernetics. 2014 December ;44(12) :2405-17. DOI : https://doi.org/10.1109/TCYB.2014.2307349.

[29] Janiesch C, Zschech P, Heinrich K. Machine learning and deep learning. Electronic Markets. 2021 ;31(3) :685-95. DOI : https://doi.org/10.1007/s12525-021-00475-2.

[30] Zhang Y, Jin R, Zhou Z. Understanding bag-of-words model : a statistical framework. International Journal of Machine Learning and Cybernetics. 2010 ;1(1) :43-52. DOI : https://doi.org/10.1007/s13042-010-0001-0.

[31] Ramos J. Using tf-idf to determine word relevance in document queries. In : Proceedings of the first instructional conference on machine learning. vol. 242. New Jersey ; 2003. p. 133-42.

[32] Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. Information Processing & Management. 1988 ;24(5) :513-23. DOI : https://doi.org/10.1016/0306-4573(88)90021-0.

[33] Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space. In : ICLR Workshop ; 2013. DOI : https://doi.org/10.4236/apm.2016.66030. Available from : https://arxiv.org/abs/1301.3781.

[34] Pennington J, Socher R, Manning CD. GloVe : Global Vectors for Word Representation. In : Moschitti A, Pang B, Daelemans W, editors. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar : Association for Computational Linguistics ; 2014. p. 1532-43. DOI : https://doi.org/10.3115/v1/D14-1162.

[35] Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching Word Vectors with Subword Information. vol. 5 ; 2017. p. 135-46. DOI : https://doi.org/10.1162/tacl_a_00051.

[36] Devlin J, Chang M, Lee K, Toutanova K. BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019 Jun :4171-86. DOI : https://doi.org/10.18653/v1/N19-1423.

[37] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. RoBERTa : A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv :190711692. 2019. DOI : https://doi.org/10.48550/arXiv.1907.11692.

[38] Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R. ALBERT : A Lite BERT for Self-supervised Learning of Language Representations. arXiv preprint arXiv :190911942. 2019. DOI : https://doi.org/10.48550/arXiv.1909.11942.

[39] Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al. Language Models are Few-Shot Learners. 2020 :1877-901. DOI : https://doi.org/10.5555/3495724.3495883.

[40] Mukta MSH, Akter A, Ahmed M, Islam S. A Review on Deep-Learning-Based Cyberbullying Detection. Future Internet. 2023 ;15(5) :179. DOI : https://doi.org/10.3390/fi15050179. Available from : https://www.mdpi.com/1999-5903/15/5/179.

[41] Philipo AG, Sarwatt DS, Ding J, Daneshmand M, Ning H. Assessing Text Classification Methods for Cyberbullying Detection on Social Media Platforms. arXiv preprint arXiv :241219928. 2024. Available from : https://arxiv.org/abs/2412.19928.

[42] Kumar Pea. Multimodal Approaches for Toxic Content Detection : A Survey. ACM Computing Surveys. 2022. Available from : https://dl.acm.org/doi/10.1145/3542703.

[43] Lee MJ, Park JW. Real-Time Violence Detection in Video using Swin Transformer. Expert Systems with Applications. 2024 ;213 :119056.

[44] Davidson T, Warmsley D, Macy M, Weber I. Automated Hate Speech Detection and the Problem of Offensive Language. arXiv preprint. 2017 March. DOI : https://doi.org/10.48550/arXiv.1703.04009.

[45] Mansur Z, Omar N, Tiun S. Twitter Hate Speech Detection : A Systematic Review of Methods, Taxonomy Analysis, Challenges, and Opportunities. IEEE Access. 2023 ;11 :42012-37. DOI : https://doi.org/10.1109/ACCESS.2023.3260417. Available from : https://ieeexplore.ieee.org/document/10099992.

[46] Singh K, Sharma R. Violence detection in videos using CNN and keyframe extraction. Journal of Visual Communication and Image Representation. 2022.

[47] Dhiman K, Rana A. Detecting violence in videos using ResNet50 and LSTM. Multimedia Tools and Applications. 2021.

[48] Lee H, Kim J. Swin Transformer for Real-Time Detection of Violent Scenes in Social Media Videos. IEEE Access. 2023.

[49] Cao R, Hee MS, Kuek A, Chong W, Lee RK, Jiang J. Pro-Cap : Leveraging a Frozen Vision-Language Model for Hateful Meme Detection. In : Proceedings of the 31st ACM International Conference on Multimedia (MM '23) ; 2023. DOI : https://doi.org/10.1145/3581783.3612498.

[50] Yang Y, Kim J, Kim Y, Ho N, Thorne J, Yun S. HARE : Explainable Hate Speech Detection with Step-by-Step Reasoning. In : Findings of the Association for Computational Linguistics : EMNLP2023. Association for Computational Linguistics ; 2023. p. 5490-505. DOI : https://doi.org/10.18653/v1/2023.findings-emnlp.365.

[51] Mnassri K, Rajapaksha P, Farahbakhsh R, Crespi N. BERT-based Ensemble Approaches for Hate Speech Detection. In : Proceedings of IEEE Global Communications Conference (GLOBECOM) ; 2022. DOI : https://doi.org/10.1109/GLOBECOM48099.2022.10001325.

[52] Putra CD, Wang HC. Advanced BERT-CNN for Hate Speech Detection. In : Procedia Computer Science. vol. 234 ; 2024. p. 239-46. DOI : https://doi.org/10.1016/j.procs.2024.02.170.

[53] Kumar GK, Nandakumar K. Hate-CLIPper : Multimodal Hateful Meme Classification based on Cross-modal Interaction of CLIP Features. arXiv preprint. 2022. DOI : https://doi.org/10.48550/arXiv.2210.05916.

[54] Vijayaraghavan P, Larochelle H, Roy D. Interpretable multi-modal hate speech detection. arXiv preprint arXiv :210300000. 2021. DOI : https://doi.org/10.48550/arXiv.2103.01616.

[55] Osei J, Wang L. Multimodal Detection of Explicit Content Using Vision Transformers and BERT. ACM Transactions on Multimedia Computing, Communications, and Applications. 2024.

[56] Shah SB, Shiwakoti S, Chaudhary M, Wang H. MemeCLIP : Leveraging CLIP Representations for Multimodal Meme Classification. In : Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics ; 2024. p. 17320-32. DOI : https://doi.org/10.18653/v1/2024.emnlp-main.959. Available from : https://aclanthology.org/2024.emnlp-main.959/.

[57] Devlin J, Chang MW, Lee K, Toutanova K. BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv :181004805. 2018.

[58] Srivastava P, Bej S, Yordanova K, Wolkenhauer O. Self-Attention-Based Models for the Extraction of Molecular Interactions from Biological Texts. Biomolecules. 2021 ;11(11) :1591.

[59] Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. Communications of the ACM. 2017 May ;60(6) :84-90.

[60] Haque M, Nyeem H, Afsha S. BrutNet : A novel approach for violence detection and classification using DCNN with GRU. The Journal of Engineering. 2024 ;2024(4) :e12375. DOI : https://doi.org/10.1049/tje2.12375.

[61] Li Y, Wang X, Zhang Q. Lightweight Violence Detection Model Based on 2D CNN with Bi-Directional Motion Attention. Applied Sciences. 2024 ;14(11) :4895. DOI : https://doi.org/10.3390/app14114895.

[62] Singh S, Dewangan S, Krishna GS, Tyagi V, Reddy S, Medi PR. Video Vision Transformers for Violence Detection. arXiv preprint arXiv :220903561. 2022.

[63] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016 :770-8.

[64] Chung YA, Hsu WN, Glass J. Learning Word Embeddings from Speech. In : Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) ; 2018. p. 1009-19.

[65] Alam J, Basit A, Ziar RA. Utilizing Age-Adaptive Deep Learning Approaches for Detecting Inappropriate Video Content. Human Behavior and Emerging Technologies. 2024. DOI : https://doi.org/10.1155/2024/7004031.

[66] Wisniewski P, Ghosh AK, Xu H, Rosson MB, Carroll JM. Parental Control vs. Teen Self-Regulation : Is there a middle ground for mobile online safety ? In : Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW) ; 2017. DOI : https://doi.org/10.1145/2998181.2998352.

[67] Zhu X, Deng C, Bai W. Parental control and adolescent internet addiction : the moderating effect of parent-child relationships. Frontiers in Public Health. 2023 ;11 :1190534. DOI : https://doi.org/10.3389/fpubh.2023.1190534.

[68] Al-Ghamdi JA, Al-Dala'in T. Educational Entertaining Web Browser for Children using Security System Design. Communications on Applied Electronics. 2016 ;5(5) :1-4. DOI : https://doi.org/10.5120/cae2016652290.

[69] Liu M, Zhang Y, Li X, Lu C, Liu B, Duan H, et al. Understanding the Implementation and Security Implications of Protective DNS Services. In : NDSS Symposium 2024 ; 2024. DOI : https://doi.org/10.14722/ndss.2024.24782.

[70] Stoilova M, Livingstone S. Do parental control tools fulfil family expectations for child protection ? A rapid evidence review of the contexts and outcomes of use. Journal of Children and Media. 2024 ;18(1) :29-52. DOI : https://doi.org/10.1080/17482798.2023.2265512.

[71] Google. Using AI to keep Google Search safe ; 2023. Description of SafeSearch leveraging BERT and machine learning for filtering explicit content. Blog post, Google. Available from : https://blog.google/products/search/using-ai-keep-google-search-safe/.

[72] Van Hee C, Jacobs G, Emmery C, Desmet B, Lefever E, Verhoeven B, et al. Automatic Detection of Cyberbullying in Social Media Text. Social

Network Analysis and Mining. 2018. DOI : https://doi.org/10.1007/s13278-018-0533-2.

[73] Rana A, Jha S. Emotion Based Hate Speech Detection using Multimodal Learning. arXiv preprint. 2022. DOI : https://doi.org/10.48550/arXiv.2202.06218. Available from : https://arxiv.org/abs/2202.06218.

[74] Samal S, Nayak R, Jena S, Balabantaray BK. Obscene Image Detection using Transfer Learning and Feature Fusion. Multimedia Tools and Applications. 2023;82 :28739-67. DOI : https://doi.org/10.1007/s11042-023-14437-7.

[75] Chawki M. AI Moderation and Legal Frameworks in Child-Centric Social Media : A Case Study of Roblox. Laws. 2023;14(3) :29. DOI : https://doi.org/10.3390/laws14030029.

[76] Ahmad T, Usman M, Noor R. An Improved Logistic Regression Framework for Hate Speech Detection in Online Texts. International Journal of Advanced Computer Science and Applications. 2023;14(3) :12-20. DOI : https://doi.org/10.14569/IJACSA.2023.0140302.

[77] Piot P, Martín-Rodilla P, Parapar J. MetaHate : A Dataset for Unifying Efforts on Hate Speech Detection. Proceedings of the International AAAI Conference on Web and Social Media. 2024;18(1) :2025-39. DOI : https://doi.org/10.1609/icwsm.v18i1.31445.

[78] Wikipedia contributors. Logistic Regression – Wikipedia ; 2024. Consulté en juin 2025. https://en.wikipedia.org/wiki/Logistic_regression.

[79] Hameda A, Bouchama Y. Improving Hate Speech Detection Using Support Vector Machines on Davidson Dataset. International Journal of Computational Linguistics. 2024;10(1) :22-35.

[80] Omran E, Al Tararwah E, Al Qundus J. A comparative analysis of machine learning algorithms for hate speech detection in social media. Online Journal of Communication and Media Technologies. 2023;13(4) :e202348. DOI : https://doi.org/10.30935/ojcmt/13603.

[81] Luo B, Zhang M. Hate Speech Detection with a Tuned Bi-LSTM on Twitter Data. In : Workshop on Abusive Language Online ; 2021. p. 15-24.

A

ANNEXE

IMAGES ILLUSTRATIVES

Dans cette annexe, la Figure 19 présente la fréquence des mots les plus courants dans le corpus de Davidson, ce qui permet d'avoir un aperçu global des termes les plus utilisés dans les différents types de discours analysés.

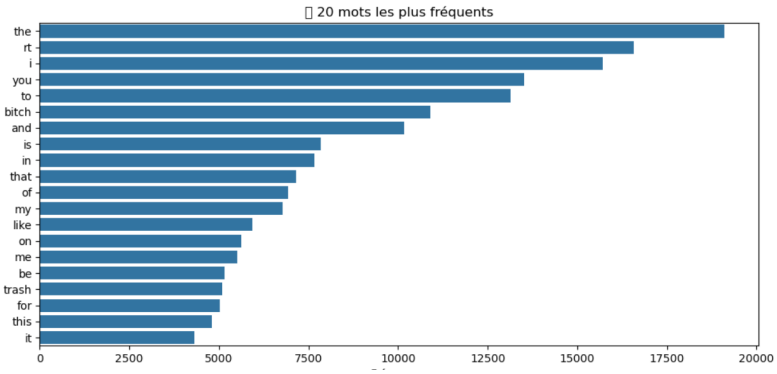

FIGURE 19 – *Fréquence des mots sur Davidson*

De même, la Figure 20 illustre un nuage de mots généré à partir du même corpus. Cette représentation visuelle facilite l'identification rapide des mots saillants en fonction de leur fréquence, tout en offrant une lecture intuitive pour interpréter les thématiques dominantes dans les données textuelles.

FIGURE 20 – *Nuage des mots sur Davidson*