

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ MOHAMED KHIDER - BISKRA

FACULTÉ DES SCIENCES EXACTES ET DE LA NATURE

DÉPARTEMENT D'INFORMATIQUE



# *Mémoire De Master*

## *Fake news detection using machine learning*

Réalisé par: **Baarir Nihel Fatima**

Soutenue publiquement le: .....

**Devant le jury composé de:**

Djeffal abdelhamid	MCA	Université de Biskra	Encadrant
		Université de Biskra	Examineur
		Université de Biskra	Examineur

*Année Universitaire 2019/2020*

# *Dédicace*



*mes chers parents,  
mes frères et sœurs,  
mes enseignants et mes chers amis,*

*je dédie ce modeste travail.*

# Remerciements



Je tiens premièrement à prosterner remerciant Allah le tout puissant de m'avoir donné le courage et la patience pour terminer ce travail.

Je voudrais dans un premier temps remercier mon directeur de mémoire, Monsieur *Abdelhamid Djefal*, de m'avoir encadré, orienté, aidé et conseillé.

J'adresse mes sincères remerciements aux membres du jury, qui ont accepté d'évaluer mon travail.

Je tiens à remercier spécialement mon grand frère *Soheib* pour ses conseils et ses critiques qui ont contribué à alimenter ma réflexion et qui m'ont aidé à affiner ce travail.

Je remercie mes très chers parents, qui ont toujours été là pour moi. Je remercie mes sœurs *Watfa et Raouia*, et mon frère *Okeil*, pour leurs paroles, leurs écrits et leurs encouragements qui ont flatté mon estime et confiance en soi.

Enfin, Je voudrais exprimer ma reconnaissance envers mon amie *Darine*, et tout membre de ma famille m'a apporté un soutien moral tout au long de ma démarche.

# Table des Matières

<i>Introduction Générale</i>	<b>2</b>
<b>1</b> <i>État de l'art sur: la détection des fausses nouvelles</i>	<b>4</b>
1.1 Introduction . . . . .	5
1.2 La signification des fausse nouvelles . . . . .	5
1.3 Les composants des fausses nouvelles . . . . .	6
1.3.1 Créateur/Diffuseur de fausses nouvelles . . . . .	6
1.3.2 Contenu d'actualité . . . . .	7
1.3.3 Contexte social . . . . .	7
1.4 Les modèles de détection des fausses nouvelles . . . . .	8
1.5 Impact global sur différents domaines . . . . .	9
1.6 Web Mining . . . . .	9
1.6.1 Processus du Web Mining . . . . .	10
1.6.2 Objectifs du Web Mining . . . . .	10
1.7 Opinion Mining . . . . .	10
1.7.1 Définition . . . . .	11
1.7.2 Méthodes d'analyse . . . . .	11
1.8 Travaux connexes . . . . .	12
1.9 Conclusion . . . . .	13
<b>2</b> <i>Machine Learning</i>	<b>14</b>
2.1 Introduction . . . . .	14
2.2 Description . . . . .	15
2.3 Types de Machine Learning . . . . .	15
2.3.1 Apprentissage supervisé . . . . .	15
2.3.2 Apprentissage non supervisé . . . . .	25
2.3.3 Apprentissage semi-supervisé . . . . .	25
2.3.4 Apprentissage par renforcement . . . . .	25

2.4	Conclusion . . . . .	26
<b>3</b>	<b><i>Conception du système</i></b>	<b>27</b>
3.1	Introduction . . . . .	27
3.2	Architecture générale . . . . .	28
3.3	Architecture détaillée . . . . .	28
3.3.1	Pré-traitement . . . . .	29
3.3.2	Apprentissage: . . . . .	33
3.3.3	Utilisation . . . . .	36
3.4	Conclusion . . . . .	37
<b>4</b>	<b><i>Implémentation et résultats</i></b>	<b>38</b>
4.1	Introduction . . . . .	38
4.2	Outils utilisés . . . . .	39
4.2.1	Logiciels utilisés . . . . .	39
4.2.2	Langages utilisés . . . . .	40
4.2.3	Bibliothèques utilisées . . . . .	41
4.2.4	Struts 2 . . . . .	44
4.2.5	Dataset . . . . .	45
4.3	Application: . . . . .	47
4.4	Résultats et discussion: . . . . .	52
4.4.1	Pré-traitement: . . . . .	52
4.4.2	Apprentissage: . . . . .	54
4.5	Conclusion . . . . .	57
	<b><i>Conclusion Générale</i></b>	<b>58</b>

# *Liste des Figures*

1.1	Processus de Web Mining . . . . .	10
2.1	Schéma explicatif de l'apprentissage supervisé . . . . .	16
2.2	classe binaire vs multiple-classes . . . . .	17
2.3	Réseau de neurones . . . . .	19
2.4	Machine à vecteurs de support . . . . .	20
2.5	Hyperplan séparateur . . . . .	20
2.6	Meilleure Hyperplan séparateur . . . . .	21
2.7	Astuce du noyau . . . . .	22
3.1	Architecture générale . . . . .	28
3.2	Pré-traitement . . . . .	29
3.3	encodage de la source . . . . .	31
3.4	Calcul des indices des auteurs . . . . .	32
3.5	Exemple de l'encodage des auteurs . . . . .	32
3.6	Degrés des mots . . . . .	33
3.7	Degré de confiance de la classification des nouvelles . . . . .	34
3.8	Utilisation . . . . .	37
4.1	Eclipse . . . . .	39
4.2	Weka . . . . .	40
4.3	Algorithme Analyzer en java . . . . .	41
4.4	Algorithme Porter en java . . . . .	42
4.5	LibSvm en java . . . . .	43
4.6	WhatsMate Translation API en java . . . . .	43
4.7	Modèle-vue-contrôleur . . . . .	45
4.8	Interface d'accueil de l'application . . . . .	47
4.9	Fenêtre de Pré-traitement . . . . .	48
4.10	Fenêtre d'entraînement . . . . .	49

4.11 Fenêtre de test . . . . .	50
4.12 Fenêtre d'utilisation . . . . .	51
4.13 Résultat de l'utilisation . . . . .	51
4.14 Influence des différentes caractéristiques sur le taux de reconnaissance . . .	54

# *Liste des Tableaux*

4.1	Taux de reconnaissance en fonction du nombre de mots fréquent pour le sac à mots. . . . .	52
4.2	Différents taux de reconnaissance des n-grams. . . . .	53
4.3	Différents taux de reconnaissance des $k^*(n\text{-grams})$ . . . . .	53
4.4	Valeur du taux par rapport au noyau. . . . .	54
4.5	Valeurs optimales des paramètres du noyau Gaussien . . . . .	57

# Introduction Générale

Ces dernières années, le phénomène des *Fake News* a connu une propagation très importante favorisée par les réseaux sociaux.

Ces fausses nouvelles peuvent être diffusées pour des objectifs différents. Certaines sont fabriquées seulement pour augmenter le nombre de clicks et de visiteurs sur un site, d'autres pour influencer l'opinion publique sur les décisions politiques ou sur les marchés financiers en impactant la réputation des entreprises et institutions sur le Web.

Une étude menée par l'économiste Roberto Cavazos de l'université de Baltimore permet de se rendre compte de l'impact direct de ces *Fake News* sur l'économie.

Selon l'étude donc, au moins 400 millions de dollars ont été dépensés pour la diffusion de *Fake News* dans les campagnes politiques lors des dernières élections en Inde (140 millions de dollars en 2019), au Brésil (34 millions de dollars en 2018), au Royaume-uni (1 million de dollars), en France (586.000 euros), en Australie (828.000 dollars), au Kenya (20 millions de dollars), en Afrique du Sud (2,7 millions de dollars) ou au Mexique (642.000 dollars) par exemple.

Et les États-Unis ne sont pas en reste, puisa une étude menée par l'université de Princeton sur la consommation de *Fake News* pendant la campagne américaine de 2016 a révélé que les fausses informations représentaient 2,6% de tous les articles d'actualité publiés lors de la fin de la dernière course à la présidence en 2016 remportée par *Donald Trump*. Ceci n'est qu'un échantillon témoignant de l'intérêt de détecter les *Fake News*.

Parmi les principales méthodes pour identifier ces fausses nouvelles : les règles linguistiques qui détermine le sentiment dégagé par les mots d'un commentaire puis classer ce dernier dans des catégories différentes; ou bien l'apprentissage automatique supervisé qui permet d'entraîner le système sur une base pour classer les nouvelles informations, cet apprentissage contient plusieurs algorithmes qui peuvent être utilisés en fonction de la demande.

Dans ce mémoire, nous allons présenter une méthode pour détecter les fausses nouvelles qui se base sur l'utilisation de:

- **Pré-traitement du texte:** constitué par le steaming et l'analyse du texte en

---

supprimant les mots vides et les caractères spéciaux.

- **Encodage du texte:** composé par le sac à mots et N-gram puis TF-IDF.
- **Extraction des caractéristiques:** qui permet de mieux identifier une fausse information, tel que la source, l'auteur, la date et le sentiment dégagé par le texte.
- **Machine à vecteurs de support:** un algorithme d'apprentissage automatique supervisé, permet de mieux classer les nouvelles informations.

Notre mémoire se compose de quatre chapitres:

1. **Etat de l'art sur la détection des fausses informations:** englobant les différentes définitions concernant les fausses nouvelles et le domaine du *Web Mining*.
2. **Machine learning:** expliquant son principe et les méthodes utilisées dans la détection des fausses nouvelles.
3. **Conception du système:** décrivant l'architecture générale et l'architecture détaillée de notre système.
4. **Implémentation et résultats:** présentant la mise en oeuvre de notre travail et les résultats obtenus.

# *État de l'art sur: la détection des fausses nouvelles*

□

---

1.1	Introduction . . . . .	5
1.2	La signification des fausse nouvelles . . . . .	5
1.3	Les composants des fausses nouvelles . . . . .	6
1.3.1	Créateur/Diffuseur de fausses nouvelles . . . . .	6
1.3.2	Contenu d'actualité . . . . .	7
1.3.3	Contexte social . . . . .	7
1.4	Les modèles de détection des fausses nouvelles . . . . .	8
1.5	Impact global sur différents domaines . . . . .	9
1.6	Web Mining . . . . .	9
1.6.1	Processus du Web Mining . . . . .	10
1.6.2	Objectifs du Web Mining . . . . .	10
1.7	Opinion Mining . . . . .	10
1.7.1	Définition . . . . .	11
1.7.2	Méthodes d'analyse . . . . .	11
1.8	Travaux connexes . . . . .	12
1.9	Conclusion . . . . .	13

---

## 1.1 Introduction

Le but de ce chapitre est de présenter d'une part quelques rappels indispensables et nécessaires à la compréhension de ce mémoire et d'autre part faire une synthèse bibliographique de quelques problèmes liés au domaine des *Fake News*, ceux dédiés plus précisément à l'étude de la gestion des réseaux sociaux.

Nous présentons dans un premier temps tout ce qui concerne les fausses nouvelles et ensuite, la façon de les détecter. Nous allons mettre l'accent sur le *Web Mining* en se basant sur *l'Opinion Mining*.

## 1.2 La signification des fausse nouvelles

Au cours des dernières années, le développement rapide et explosif des médias sociaux a entraîné une croissance considérable du nombre de fausses nouvelles. De nos jours, les fausses nouvelles sont ennuyeuses, envahissantes, distrayantes et omniprésentes. Il est donc important de mettre en place un système de détection efficace pour l'identification des fausses nouvelles.

**Fake News:** fausses histoires diffusées sur Internet ou d'autres médias. Conçues pour influencer les opinions politiques ou juste comme une blague. Elle sont caractérisées par ce qui suit [ZG20]:

- **Volume:** le volume des fausses nouvelles est très grand tel que tout le monde peut facilement écrire de fausses nouvelles sur Internet et sans aucune procédure de vérification.
- **Variété:** il existe plusieurs sources des fausses nouvelles: les rumeurs, les nouvelles satiriques, les fausses critiques, les fausses informations, les fausses publicités, les théories du complot, les fausses déclarations des politiciens,...etc.

- **Rapidité:** l'apparition des fausses nouvelles est très rapide, Ceci complexifie leur détection par les systèmes de contre-mesures.

## 1.3 Les composants des fausses nouvelles

Le terme *Fake News* se base sur quatre composants principaux: créateur/épandeur, victime cible, contenu de l'actualité et contexte social [ZG20].

- **Créateur / Diffuseur:** les créateurs de fausses nouvelles en ligne peuvent être des humains ou non.
- **Victimes ciblées:** les victimes des fausses nouvelles peuvent être des utilisateurs de médias sociaux ou d'autres récentes plates-formes. Selon les objectifs de la nouvelle, les cibles peuvent être des étudiants, des électeurs, des parents, des personnes âgées...etc.
- **Contenu de l'actualité:** le contenu de l'actualité fait référence au corps de la nouvelle. Il contient à la fois un contenu physique (par exemple, titre, corps de texte, multimédia) et contenu non physique (par exemple, but, sentiment, sujets).
- **Contexte social:** le contexte social indique comment les nouvelles sont diffusées sur Internet. L'analyse du contexte social inclut l'utilisateur, l'analyse du réseau (comment les utilisateurs en ligne sont impliqués dans les actualités) et l'analyse du modèle de diffusion.

### 1.3.1 Créateur/Diffuseur de fausses nouvelles

Il est important de montrer qui est derrière la fausse nouvelle et pourquoi elle est écrite et partagée tout au long de la vie sociale. Le créateur/diffuseur des fausses nouvelles peut être soit un humain, soit non.

- **Humain:** les robots sociaux ou les *Cyborgs* ne sont que les porteurs de fausses nouvelles sur les médias sociaux. Ces comptes automatiques sont programmés pour diffuser de faux messages par des humains, qui cherchent à perturber la crédibilité de la communauté sociale en ligne.

- **Non-humain:** les robots sociaux ou les *Cyborgs* sont les créateurs non-humains les plus courants, de fausses nouvelles. Ce sont des programmes informatiques conçus pour afficher des comportements similaires à ceux des humains, et pour produire automatiquement du contenu et interagir avec les humains via les médias sociaux, diffuser des rumeurs, des spams, des logiciels malveillants, des informations erronées.

### 1.3.2 Contenu d'actualité

Chaque nouvelle est constituée d'un contenu d'actualité physique et d'un contenu d'actualité non-physique.

- **Contenu d'actualité physique:** le contenu physique des fausses nouvelles contient le titre de la nouvelle, le corps de la nouvelle, des images, des vidéos, des hashtags, des signaux de mention, des emojis,...etc. En raison de certaines significations et fonctionnalités, ces composants sont des fonctionnalités importantes pour la détection de fausses nouvelles.
- **Contenu d'actualité non-physique:** le contenu non physique concerne les opinions, les émotions, attitudes et sentiments que les créateurs de nouvelles veulent exprimer.

Exemple, chaque jour, des millions de commentaires sont publiés sur des plateformes de vente en ligne, ces avis biaisés sont de gros problèmes pour les marques et les clients en ligne, non seulement ils vont affecter la décision du processus de la fabrication, mais aussi ils peuvent facilement détruire la réputation d'une marque.

### 1.3.3 Contexte social

Le contexte social fait référence à l'ensemble du système d'activité et de l'environnement social dans lequel la diffusion de la nouvelle a lieu. Aujourd'hui, les modes de partage de la nouvelle sont de plus en plus dominés par les technologies interactives sur les médias sociaux. Les utilisateurs en ligne peuvent non seulement apprendre d'avantage sur les tendances, mais aussi partager leurs histoires et défendre leurs intérêts. S'ils partagent ces expériences et interactions au sein de certains groupes sociaux, et les membres de ces groupes partageant les mêmes idées, l'influence de ces dernières peut être amplifiée. Ceci facilite la diffusion de fausses nouvelles.

## 1.4 Les modèles de détection des fausses nouvelles

Nous allons maintenant discuter des modèles de détection des fausses nouvelles [AHC19].

- **Modèles du contenu d'actualités:** les modèles de contenu des actualités peuvent être classifiés en ce qui suit:
  1. **Modèle basé sur les connaissances:** l'approche fondée sur les connaissances a pour objectif d'utiliser des sources pour vérifier les faits dans le contenu des actualités.
  2. **Modèle basé sur le style:** les éditeurs des fausses nouvelles utilisent certains styles d'écriture spécifique nécessaire pour faire appel à un véritable article de presse, tel que les mots neutres, décrivant les événements avec des faits. Une meilleure qualité d'écriture (en tenant compte des mots épinglés, ponctuation et longueur des phrases).
- **Modèles du contexte social:** les réseaux sociaux fournissent des ressources supplémentaires aux chercheurs pour compléter et améliorer les modèles de contexte d'actualité. Les modèles du contexte social sont aussi utilisés pour la détection des rumeurs et l'identification des faux contenus sur *Facebook*. Ces modèles sont en fonction de la position et de la propagation.
  1. **Basé sur la position:** c'est un processus qui peut déterminer les sentiments dégagés par l'utilisateur, en faveur, contre ou neutre. Il existe deux façons pour représenter la position de l'utilisateur explicite ou implicite. Les positions explicites sont celles où les lecteurs ont donné des expressions directes, comme le pouce vers le haut ou le pouce vers le bas. Les positions implicites sont les positions où les sentiments extraits des publications sur les réseaux sociaux.
  2. **Basé sur la propagation:** c'est un processus qui peut déterminer la relation entre les événements pertinents sur les publications. Il existe deux catégories, propagation homogène qui contient une entité unique, comme un message ou un événement, ou propagation hétérogène qui contient plusieurs entités en même temps.

3. **Détection des rumeurs:** la détection des rumeurs a pour objectif de classer une nouvelle comme rumeur ou non. C'est un modèle en quatre étapes: détection, suivi, position et véracité.

## 1.5 Impact global sur différents domaines

Les nouvelles visent des situation réels actuelles et des histoires complètes, couvrant différentes questions, (ex; la criminologie, la santé, le sport, la politique...etc). Voici quelques impactes des fausses nouvelles sur des différents domaines [ZG20].

1. Dans les médias en ligne, les fausses nouvelles permettent de donner de l'ampleur aux faux comptes, (ex; récupération de maximum de clicks et d'adhérents).
2. Influence les propagandes politiques, en affectant les décisions de vote lors des élections;
3. Influencer les marchés financiers, où des millions pourraient être perdus.

## 1.6 Web Mining

Oren Etzioni a appliqué la technologie du *data Mining* sur le Web pour l'extraction d'informations pertinentes à partir des ressources disponibles dans le Web, qui peuvent être un document ou un service Web. Depuis 1996, plusieurs autres travaux et publications se sont intéressés au sujet. Et donc le *Web Mining* a touché plusieurs autres domaines, (ex;recherche d'information, bases de données, intelligence artificielle). En fonction du type de données à fouiller, on distingue trois grandes classes: *Web Content Mining*, *Web Structure Mining* et *Web Usage Mining* [MBNL99].

### 1.6.1 Processus du Web Mining

Le processus du *Web Mining* se déroule en trois étapes [MBNL99]:

- La collecte des données sur l'utilisateur;
- La classification et la personnalisation de ses données;
- La présentation à l'utilisateur d'un contenu ciblé.

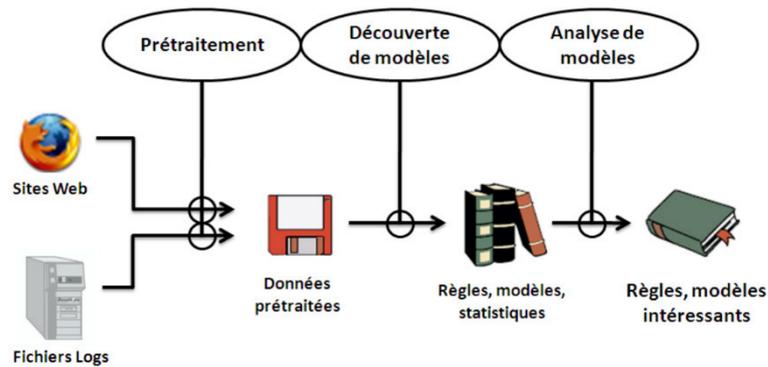


Figure 1.1: Processus de Web Mining.

[MBNL99]

### 1.6.2 Objectifs du Web Mining

Le *Web Mining* poursuit deux principaux objectifs [Cha05]:

- **L'amélioration et la valorisation des sites Web:** le suivi du comportement des utilisateurs des sites Web permet de valoriser leurs contenus en améliorant l'organisation et les performances des sites.
- **La personnalisation:** les techniques de *Data mining* appliquées aux données collectées sur le Web permettent d'extraire des informations intéressantes relatives à l'utilisation du site par les internautes. L'analyse de ces informations permet de personnaliser les utilisateurs, leurs préférences et leurs profils.

## 1.7 Opinion Mining

Aujourd'hui, les internautes ont pris l'habitude de consulter les commentaires déposés par les autres dès qu'ils doivent prendre une décision d'achat pour un produit technique,

ou encore pour une réservation d'hôtel. Et donc un commentaire peut détruire facilement la réputation d'une marque ou un produit. D'où les outils de *Web Mining* s'intéresse à analyser les opinions des internautes qui s'y expriment spontanément et en temps réel. [BL12].

### 1.7.1 Définition

L'*Opinion Mining* c'est l'étude qui analyse les opinions, les sentiments, les évaluations, les attitudes et les émotions des gens à partir de leurs écrits. L'importance croissante de l'analyse des sentiments coïncide avec la croissance des réseaux sociaux tels que Facebook et Twitter. Pour la première fois dans l'histoire de l'humanité, nous avons maintenant un énorme volume de données appelé *big data* d'opinion enregistrées sous forme numérique pour les classer en trois catégories: positive, négative et neutre [Liu12].

### 1.7.2 Méthodes d'analyse

Le but de l'analyse d'opinion est de déterminer si le sentiment dégagé par une phrase est positif ou négatif, qui dépend du contexte dans lequel elle est utilisée, du type de langage, ainsi que de la personne qui l'a écrite. Il existe plusieurs façons d'analyser automatiquement les sentiments. Parmi les méthodes les plus populaires, on cite [VC12]:

- **Apprentissage automatique supervisé:** l'objectif ici est de faire entraîner le système sur une base de telle façon qu'il puisse classer par la suite de façon automatique les nouvelles informations. Par exemple, un expert détermine pour chaque commentaire si l'utilisateur est « très énervé », « moyennement énervé » et « pas du tout énervé ». Pour que l'apprentissage automatique fonctionne, il faut annoter un grand jeu de données. Ensuite, un algorithme analyse cette annotation pour en faire ressortir des règles qui permettront de classer d'autres commentaires selon le degré d'énervement.
- **Règles linguistiques:** dans ce cas, un linguiste analyse un échantillon de commentaires pour déterminer quels sont les mots ou les expressions qui indiquent l'énervement et ses différents degrés. Une fois cette analyse linguistique effectuée, il crée des règles de grammaire qui permettront au système de classer chaque commentaire dans l'une ou l'autre des catégories.

## 1.8 Travaux connexes

- Les auteurs de [CRC15] proposent une typologie de plusieurs méthodes d'évaluation de la véracité émergeant de deux catégories principales: les approches de repères linguistiques (avec apprentissage automatique) et les approches d'analyse de réseau, pour la détection des fausses nouvelles.
- Les auteurs de [GM17] présentent une approche simple pour la détection de fausses nouvelles en utilisant un classificateur Bayésien naïf. Cette approche est testée par rapport à un ensemble de données extraites des messages d'actualités Facebook. Ils proclament pouvoir atteindre une précision de 74%.
- Les auteurs de [ATS17] proposent un modèle de détection de fausses nouvelles qui utilise l'analyse n-gram et les techniques d'apprentissage automatique en comparant deux techniques d'extraction de caractéristiques différentes et six techniques de classification différentes. Les expérimentations menées montrent que les meilleures performances sont obtenues en utilisant la méthode d'extraction des caractéristiques dite (TF-IDF). Le classifieur utilisé est la machine à vecteurs de support linéaire (LSVM), qui a donné une précision de 92%.
- Les auteurs de [Sau18] décrivent comment les utilisateurs des réseaux sociaux peuvent s'assurer de la vérité des informations, comment les critères du vrai et du faux se définissent. Ils décrivent aussi les mécanismes qui permettent leur validation et le rôle des journalistes ou ce qu'il faut attendre des chercheurs et des institutions officielles.
- Les auteurs de [MKC18] proposent plusieurs stratégies et types d'indices relevant de différentes modalités (texte, image, informations sociales). Ils explorent également l'intérêt de combiner et fusionner ces approches pour évaluer et vérifier l'information partagée.

## 1.9 Conclusion

Dans ce chapitre, les différents concepts des fausses nouvelles ont été présentés. Nous avons vu leur signification, caractéristiques ainsi que les différents modèles utilisés pour leur détection. L'application du *Data Mining* dans le Web et particulièrement *l'Opinion Mining* a permis d'apporter des solutions intéressantes a ce problème. Dans le chapitre suivant, nous allons nous concentrer précisément sur les méthodes d'apprentissage automatique et choisir la méthode qui convient le mieux au problème de détection des fausses nouvelles.

# Machine Learning

---

2.1	Introduction . . . . .	14
2.2	Description . . . . .	15
2.3	Types de Machine Learning . . . . .	15
2.3.1	Apprentissage supervisé . . . . .	15
2.3.2	Apprentissage non supervisé . . . . .	25
2.3.3	Apprentissage semi-supervisé . . . . .	25
2.3.4	Apprentissage par renforcement . . . . .	25
2.4	Conclusion . . . . .	26

---

## 2.1 Introduction

Les fausses nouvelles en ligne et les rumeurs étant souvent réduites mais de grandes conséquences. Les traitements par les techniques linguistiques traditionnelles et d'incorporation tel que le sac à mots ou le n-gram constituent une bonne approche pour leur détection et analyse. Cependant, ceci n'est pas suffisant pour extraire les fausses nouvelles, vu l'évolution continue de la langue utilisée ainsi que les limites présentés par ces méthodes pour la prise en charge de plusieurs aspects tel que les relations entre les mots et certaines caractéristiques relatives au réseaux sociaux tel que l'url, auteur, le temps,...etc. Pour en remédier, les techniques d'apprentissage automatique, sont des algorithmes puissants d'intégration du langage naturel, utilisés dans la détection de fausses nouvelles et même dans la détection des rumeurs ou des satires. Dans ce chapitre, nous allons voir dans un premier temps, la description, les types et la modélisation de l'apprentissage automatique ainsi que ses méthodes les plus utilisées dans les systèmes de détection des

fausses nouvelles et les différences majeures entre elles.

## 2.2 Description

Le *Machine Learning*, ou apprentissage automatique en français, est un sous-domaine de l'intelligence artificielle (IA). Il a pour objectif de comprendre la structure des données et de les intégrer dans des modèles qui peuvent être compris et utilisés pour tester d'autres données. Le *Machine Learning* est né grâce aux technologies de reconnaissance de pattern et à la théorie selon laquelle les ordinateurs peuvent apprendre sans être programmés pour effectuer des tâches spécifiques (ex; classification, régression). Les chercheurs intéressés par l'intelligence artificielle souhaitent vérifier si les ordinateurs pouvaient apprendre à partir de données et s'adapter avec les nouvelles données [L17].

## 2.3 Types de Machine Learning

Les algorithmes de *Machine Learning* ne sont pas une nouveauté, mais ce n'est que depuis peu qu'il est possible d'appliquer des calculs mathématiques complexes de plus en plus vite au *Big Data*; (un ensemble très volumineux de données qui nécessite un nouvel outil de gestion des bases de données). Le *Machine Learning* est aujourd'hui utilisé dans de nombreux domaines, tel que le développement de véhicules autonomes, les systèmes de recommandations en ligne (ex; Netflix et Amazon), l'analyse des sentiments des clients, ou encore la détection de fraude [L17]. Les modèles d'apprentissage automatique peuvent être résumés comme suit: modèles supervisés, non-supervisés, semi-supervisé et par renforcement.

### 2.3.1 Apprentissage supervisé

Ce type d'apprentissage utilise comme entrée des données qui sont étiquetées avec les sorties souhaitées. L'algorithme apprend en comparant la sortie réelle de l'entrée avec la sortie prédite, pour trouver des erreurs et modifier le modèle en conséquence jusqu'à arriver au meilleur taux de reconnaissance. L'apprentissage supervisé utilise donc des modèles pour prédire les valeurs d'étiquettes sur des données non-étiquetées dites données de test [L17].

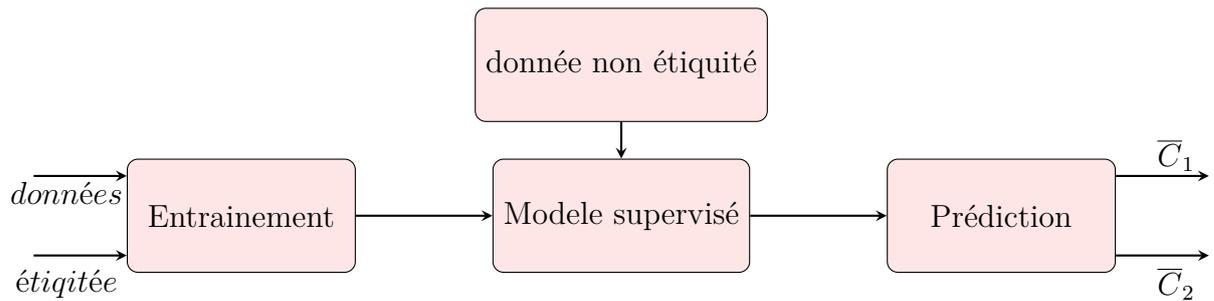


Figure 2.1: Schéma explicatif de l'apprentissage supervisé

L'apprentissage supervisé est le concept derrière plusieurs applications très courantes: reconnaissance faciale de nos photos par les smartphones, filtres anti-spam des emails...etc. Plus formellement, étant donné un ensemble de données  $D$ , décrit par un ensemble de caractéristiques  $X$ , un algorithme d'apprentissage supervisé va trouver une fonction de *Mapping* entre les variables prédictives en entrée  $X$  et la variable à prédire  $Y$ . La fonction de *Mapping* décrivant la relation entre  $X$  et  $Y$  s'appelle un modèle de prédiction [L17]:

$$f : x \rightarrow Y$$

$$f(x) = y$$

Les caractéristiques  $X$  peuvent être des valeurs numériques, alphanumériques, des images...etc, quant à la variable prédite  $Y$ , elle peut être de deux catégories [Ben17]:

- **Variable discrète:** la variable à prédire peut prendre une valeur d'un ensemble fini de valeurs (qu'on appelle des classes). Par exemple, pour prédire si un mail est un spam ou non, la variable  $Y$  peut prendre deux valeurs possible:

$$Y \in \{spam, non - spam\}$$

- **Variable continue:** la variable  $Y$  peut prendre des valeurs dans un domaine continu. Pour illustrer cette notion, on peut penser à un algorithme qui prend en entrée des caractéristiques d'un véhicule, et tentera de prédire le prix de ce dernier (la variable  $Y$ ).

Ainsi, la catégorie de la variable prédite  $Y$  fait décliner l'apprentissage supervisé en deux sous catégories:

- La classification

- La régression

1. **Algorithmes de classification:** quand la variable à prédire est discrète, on parle d'un problème de classification. Les algorithmes de classification ont pour objectif de prédire à quelle classe appartient une donnée (ex; un email est un spam ou non). Quand les valeurs de la variable prédite  $Y$  dépasse deux éléments, on parle de classification multi-classes. L'exemple suivant illustre les deux types de classifications [Ben17].

Exemple:

Dans l'image ci-dessus, les ronds en bleu représentent la classe des mail non-spam, et les croix rouges peuvent représenter des spam. Par contre, l'image à droite est une classification multi-classes , car nous avons trois classes possibles: les triangles, les croix et les carrés.

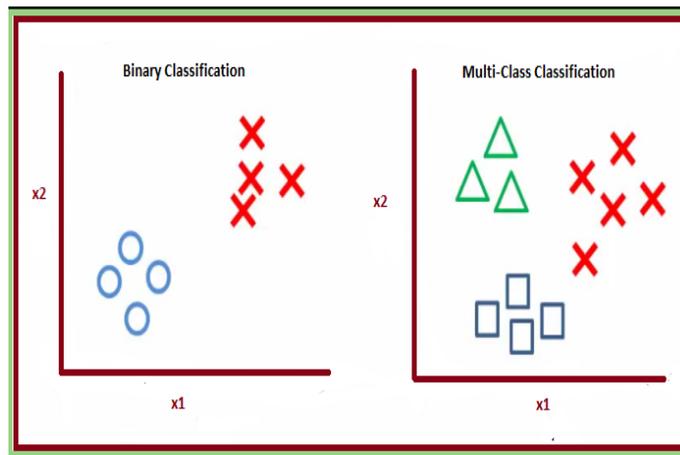


Figure 2.2: classe binaire vs multiple-classes

[Ben17]

Parmi les algorithmes de classification, on retrouve : machine à vecteurs de support, réseaux de neurones, réseaux bayésiens, arbre de décision, forêt aléatoires,...etc.

- **Arbre de décision** [Abd19]

Les arbres de décision cherche à exploiter une représentation hiérarchique de la structure des données sous forme des séquences des tests en vue de la prédiction d'une classe, tel que chaque donnée entrante est décrite par un ensemble de variables qui sont testées dans les nœuds de l'arbre. Donc, les tests s'effectuent dans les nœuds internes et les décisions sont prise dans les nœuds feuilles.

**Avantages:**

- (a) La rapidité d'entraînement par rapport à d'autres méthodes;
- (b) Passage simple et facile aux règles;
- (c) Une précision comparable à d'autres méthodes.

**Inconvénients:**

- (a) Il n'existe pas un troisième choix entre le oui et le non;
- (b) Applicable aux données symboliques seulement.

• **Réseaux bayésiens [Abd19]**

Le réseau bayésien cherche à déterminer,  $P(H/X)$ , la probabilité de vérification de  $H$  après l'observation de  $X$ . En utilisant le théorème de Bayes.

$$P(H|X) = [P(X|H) \cdot P(H)] / P(X)$$

**tel que:**

- **X**: des données non-étiquetée.
- **H**: l'hypothèse;  $X$  appartient à la classe  $C$ .
- **P(H/X)**: probabilité d'appartenance de  $X$  à la classe  $C$ .
- **P(H)**: la probabilité d'apparition de la classe  $C$  dans la population.
- **P(X)**: la probabilité d'apparition de l'attribut  $X$  dans la population .
- **P(X/H)**: probabilité d'apparition de chaque valeur des attributs de  $X$  dans les attributs des échantillons appartenant à la classe  $C$ :

$$P(X|H) = \prod P(\bar{A}_i = \bar{V}_i | H)$$

- $\bar{A}_i$  i ème attribut de  $X$  et  $\bar{V}_i$  sa valeur.

**Avantages:**

- (a) Une méthode simple et moins coûteuse en temps de calcul.
- (b) Prise en charge d'une nouvelle donnée ne nécessitant pas de refaire tous les calculs.

**Inconvénients:**

- (a) Nécessite l'indépendance entre les attributs ce qui est rare dans la réalité.

(b) Applicable aux données symboliques (catégoriel) seulement, ce qui nécessite une normalisation pour les données numériques.

- **Le réseau de neurones** [Abd19]

Un réseau de neurones artificiel est définie par ses neurones et leurs interconnexions. Sa structure est spécifiée par le nombre d'entrées, de sorties, de noeuds et comment sont inter-connectés et organisés. Ces derniers forment une couche où il n'y a pas des connexions intermédiaires. Il utilise des variables **qualitatives ou quantitatives**.

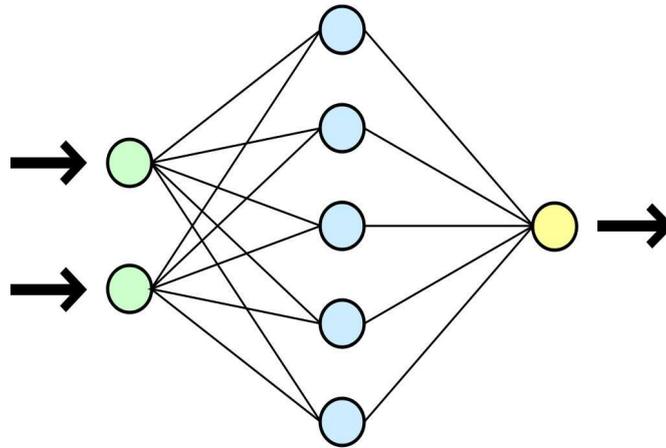


Figure 2.3: Réseau de neurones

[Mar20]

**Avantages:**

- (a) Utilisation des variables qualitatives ou quantitatives, mais résultat meilleur pour les derniers;
- (b) Utilisation pour la classification et la régression;
- (c) Traitement des problèmes non-structurés, sans aucune information au préalable.
- (d) Pas de nécessité pour l'indépendance entre les variables.
- (e) Pas de problème avec des données incomplètes ou bruitées.

**Inconvénients:**

- (a) L'ordre de présentation des données entrantes est important;
- (b) Nécessité de l'encodage des données catégorielles;
- (c) Sensibilité aux minimas locaux;

(d) Ambiguïté de leur fonctionnement, ce qui empêche de choisir la meilleure structure adaptée face à un problème donné.

- **Machine à vecteurs de support (SVM)**

L'SVM est une technique d'apprentissage supervisé introduite par *Vladimir Vapnik* au début des années 90. Cette technique classe les données d'une base linéairement, en cherchant un hyperplan qui sépare le mieux ces données en deux classes. Elle est appliquée dans la détection faciale, catégorisation de texte et hypertexte, classification des images et la bio-informatique [Abd19].

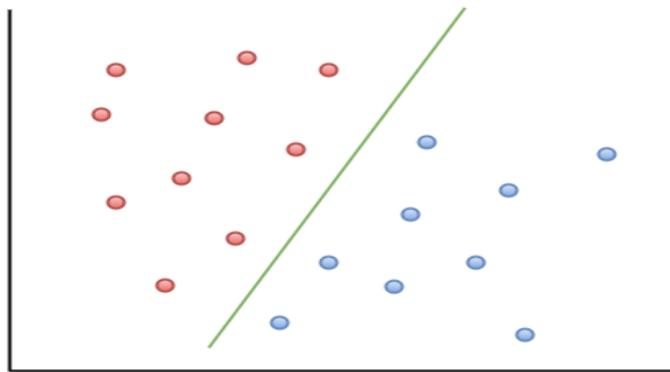


Figure 2.4: Machine à vecteurs de support

[Gan18]

Le problème ici est de savoir comment tracer cette droite?

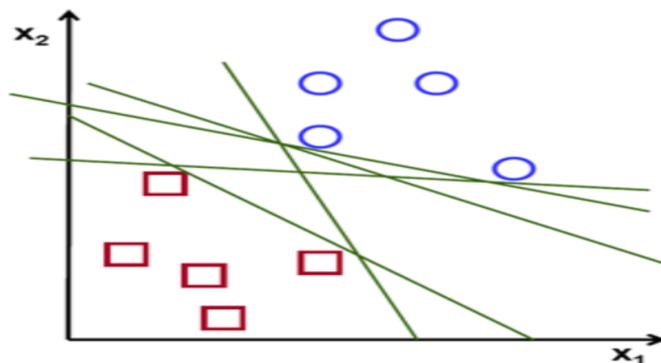


Figure 2.5: Hyperplan séparateur

[Gan18]

On va choisir tout simplement la droite qui maximise l'écart entre les deux classes. Dans l'exemple suivant, on prend le carré rouge le plus proche du cercle bleu puis, on trace une droite qui maximise la distance entre eux de sorte que la droite soit exactement au milieu et que la généralisation de l'algorithme soit optimale.

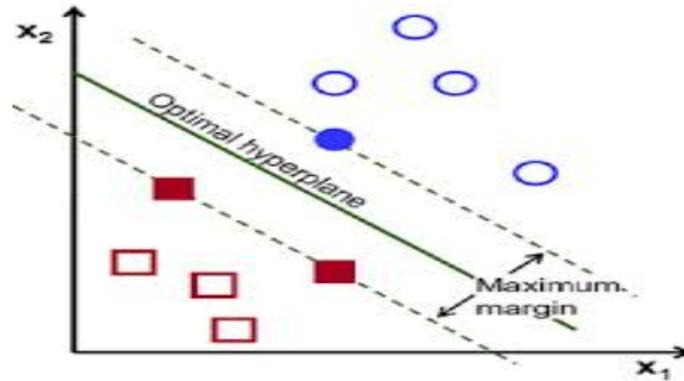


Figure 2.6: Meilleure Hyperplan séparateur

[Gan18]

Cet hyperplan séparateur est représenté par l'équation suivante:

$$H(x) = w^T x + b$$

Où:

- $w$  est un vecteur qui représente la pente de la droite séparant les exemples des deux classes.
- $b$  est un terme représentant le décalage de cette droite.
- $H(x)$  représente la distance entre le point  $x$  et la droite de séparation et appelée fonction de décision; si elle est positive l'exemple  $x$  appartient à la classe positive, si elle est négative il appartient à la classe négative.

Il reste toujours un problème: que faire lorsque notre jeu de données n'est pas linéairement séparable?

Dans ce cas, la méthode SVM utilise une astuce appelée Noyau.

**Astuce du noyau** l'idée est d'appliquer une transformation sur nos données pour pouvoir les séparer linéairement. L'algorithme transforme l'espace d'entrée, en un espace de dimension supérieure, appelé espace de redescription,

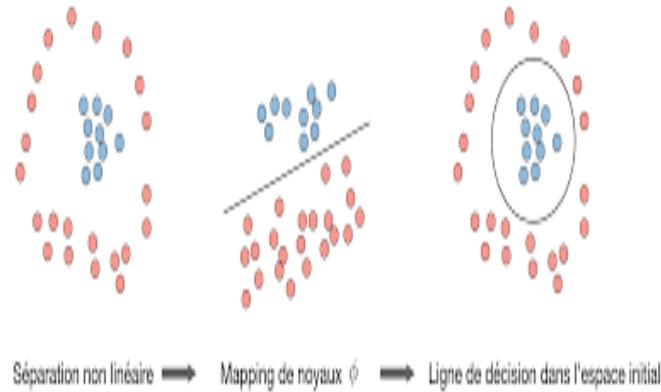


Figure 2.7: Astuce du noyau

[Gan18]

L'hyperplan séparateur aura donc la fonction suivante:

$$h(x) = w^t * \phi(x) + b \quad (2.1)$$

Où  $\phi$  est une fonction qui donne la projection de l'exemple  $x$  dans l'espace de redescription.

Les valeurs de  $w$  et  $b$  optimums sont celles maximisant la marge  $\frac{2}{\|w\|}$ : et bien classant les exemples des deux classes. pour les déterminer, on doit donc résoudre le problème quadratique suivant:

$$\left\{ \begin{array}{l} \text{Minimiser} \quad \frac{1}{2} \|w\|^2 \\ \text{sous contraintes} \\ y_i(w^T x_i + b) \geq 1 \quad \forall i = 1..n \end{array} \right. \quad (2.2)$$

Où  $x_i$  représente l'exemple numéro  $i$  et  $y_i$  sa classe égale à  $+1$  ou  $-1$ .  $n$  étant le nombre d'exemples d'entraînement.

Ce système dit primal est résolu par la méthode de Lagrange en le transformant en un problème dual introduisant les multiplacateurs de Lagrange:

$$\left\{ \begin{array}{l} \text{Maximiser } L(\alpha) = \sum_{k=1}^n \alpha_k - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \phi(x_i), \phi(x_j) \rangle \\ \text{Avec} \\ \alpha_i \geq 0 \quad \text{et } \sum_{k=1}^n \alpha_k y_k = 0 \end{array} \right. \quad (2.3)$$

Le problème de cette formulation est qu'elle implique un produit scalaire entre vecteurs dans l'espace de redescription, de dimension élevée, ce qui est coûteux en termes de calculs. Pour résoudre ce problème, on utilise une astuce connue sous le nom de *Kernel Trick*, qui consiste à utiliser une fonction noyau, qui vérifie:

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \quad (2.4)$$

L'expression de l'hyperplan séparateur en fonction de la fonction noyau :

$$h(x) = \sum_{k=1}^n \alpha_k^* y_k K(x_k, x) + b \quad (2.5)$$

Plusieurs noyaux sont utilisés dans la littérature:

- **Noyau linéaire:** si les données sont linéairement séparables, on n'a pas besoin de changer d'espace, et le produit scalaire suffit pour définir la fonction de décision:

$$k(x_i, x_j) = x_j^t * x_i \quad (2.6)$$

- **Noyau polynomial:** si les données ne sont pas linéairement séparables, le noyau polynomial élève le produit scalaire à une puissance naturelle  $d$  :

$$k(x_i, x_j) = ((x_j^t * x_i) + c)^d \quad (2.7)$$

- **Noyau RBF:** les noyaux RBF *Radial Basis functions* sont des noyaux qui peuvent être écrits sous la forme :  $K(x_i, x_j) = f(d(x_i, x_j))$  où  $d$  est une métrique sur  $x$  et  $f$  est une fonction dans  $<$ .

Un exemple des noyaux RBF est le noyau *Gaussien* :

$$k(x, y) = e^{-\frac{|x-y|^2}{2*\gamma^2}} \quad (2.8)$$

Encore deux problèmes: que faire lorsque on a plusieurs classes?

***Multiple-classes:*** les méthodes des machines à vecteur support multiple-classe, réduisent le problème multi-classes à une composition de plusieurs hyperplans permettant de tracer les frontières de décision entre les différentes classes. Ces méthodes décomposent l'ensemble des exemples en plusieurs sous ensembles représentant chacun un problème de classification binaire. Pour chaque problème un hyperplan de séparation est déterminé par la méthode SVM binaire. On construit lors de la classification une hiérarchie des hyperplans binaires, qui est parcourue de la racine jusqu'aux feuilles pour décider de la classe d'un nouvel exemple [Abd19].

**Avantages de la méthode SVM:**

- (a) Utilisation des données qualitatives et quantitatives;
- (b) Grande précision de prédiction;
- (c) Meilleur fonctionnement sur les *datasets* de taille réduite.

**Inconvénients:**

- (a) Temps d'entraînement long avec les bases volumineuses;
- (b) Moins efficace sur les jeux de données contenant du bruits.

2. **Algorithmes de régression:** un algorithme de régression permet de trouver un modèle pour permettre d'estimer la valeur d'une donnée qui ne faisait pas partie des données d'entraînement et (non pas la classifier). Donc on prend comme entrée une variable continue (et non pas discrète). Les algorithmes de régression peuvent prendre plusieurs formes, en fonction du modèle qu'on souhaite construire. La régression linéaire est le modèle le plus simple: il consiste à trouver la meilleure droite qui s'approche le plus des données d'apprentissage. La fonction de prédiction sera donc une droite [Ben17].

### 2.3.2 Apprentissage non supervisé

Ce type d'apprentissage utilise comme entrée des données non-étiquetées, de sorte que l'algorithme d'apprentissage trouve tout seul des points communs entre ses données. L'objectif de l'apprentissage non supervisé peut être aussi simple que de découvrir des modèles cachés dans un ensemble de données, mais il peut aussi avoir un objectif d'apprentissage des caractéristiques, qui permet à la machine intelligente de découvrir automatiquement les représentations nécessaires pour classer les données brutes. L'apprentissage non-supervisé est principalement utilisé en matière de clusterisation, procédé destiné à regrouper un ensemble d'éléments hétérogènes sous forme de sous groupes homogènes ou liés par des caractéristiques communes. (ex; le système peut recommander un livre ou un film à un utilisateur en fonction des goûts d'utilisateurs partageant des caractéristiques communes) [L17].

### 2.3.3 Apprentissage semi-supervisé

Ce type d'apprentissage utilise à la fois des données étiquetées et non-étiquetées pour s'entraîner. En règle générale, une petite quantité de données étiquetées est utilisée avec une grande quantité de données non-étiquetées. Pour cause, les données non étiquetées sont moins chères et plus faciles à obtenir. Ce type d'apprentissage peut être utilisé avec des méthodes comme la classification, la régression et la prédiction. L'apprentissage semi-supervisé est utile quand les coûts associés à l'étiquetage sont trop élevés pour permettre un processus d'apprentissage entièrement étiqueté. Par exemple il est utilisé pour identifier le visage d'une personne sur une *Webcam* [L17].

### 2.3.4 Apprentissage par renforcement

Ce type d'apprentissage multiplie les tentatives pour tenter de découvrir quelles actions apportent les plus grandes récompenses. Ce type d'apprentissage regroupe trois principaux composants: l'agent (qui apprend ou prend les décisions), l'environnement (tout ce avec quoi l'agent interagit), et les actions (ce que peut faire l'agent). L'objectif était que l'agent choisisse les actions qui maximisent les récompenses attendues sur une période donnée. L'agent atteindra ce but plus rapidement en suivant des règles bien établies. L'objectif de l'apprentissage par renforcement est donc d'apprendre les meilleures règles

[L17].

## 2.4 Conclusion

Dans ce chapitre, nous avons vu la description et le principe de fonctionnement des différents types de l'apprentissage automatique, ainsi que les définitions, les avantages et inconvénients des algorithmes de l'apprentissage supervisé les plus utilisées dans la détection des fausses nouvelles.

Dans le chapitre suivant, nous allons présenter la conception de notre solution pour le problème.

## *Conception du système*

---

3.1	Introduction . . . . .	27
3.2	Architecture générale . . . . .	28
3.3	Architecture détaillée . . . . .	28
3.3.1	Pré-traitement . . . . .	29
3.3.2	Apprentissage: . . . . .	33
3.3.3	Utilisation . . . . .	36
3.4	Conclusion . . . . .	37

---

### **3.1 Introduction**

Dans ce chapitre nous aborderons une description générale de notre système, en mettant en évidence son côté conceptuel du pré-traitement qui constitue une étape fondamentale avant la mise en oeuvre de notre système, ensuite nous détaillerons chaque phase en citant les principaux algorithmes et techniques utilisées dans chacune des phases.

## 3.2 Architecture générale

Notre système se base sur l'utilisation du *Machine Learning* pour détecter les *Fake News*. Le système prend en entrée une base brute de commentaires et leurs caractéristiques et la transforme en une base de caractéristiques utilisable par la phase d'apprentissage. Cette transformation est appelée pré-traitement, elle effectue une série d'opérations telles que le nettoyage, le filtrage et l'encodage. La base pré-traitée est subdivisée en deux parties; une pour l'entraînement et l'autre pour le test. Le module d'entraînement utilise la base d'entraînement et un algorithme d'apprentissage pour fournir un modèle de décision qui est appliqué sur la base de test. Si le modèle est accepté, c-à-d a pu atteindre un taux de reconnaissance acceptable, il sera conservé et utilisé par le module d'utilisation et l'entraînement se termine. Dans le cas contraire, les paramètres de l'algorithme d'apprentissage sont révisés dans le but d'améliorer le taux de reconnaissance.

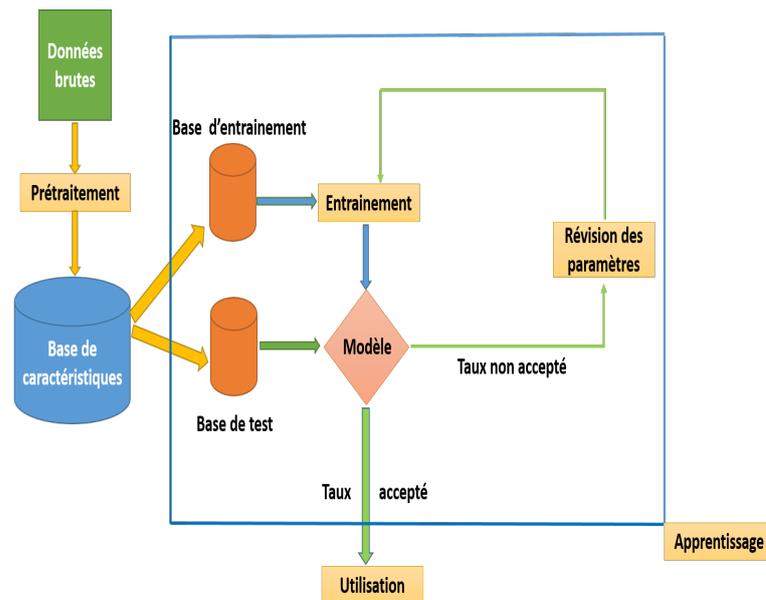


Figure 3.1: Architecture générale

## 3.3 Architecture détaillée

Dans ce qui suit, nous détaillons chacune des phases de notre système.

### 3.3.1 Pré-traitement

Notre objectif est d'extraire les meilleures caractéristiques permettant de détecter une *Fake News*. On commence par le pré-traitement des données du *dataset* brute qui sont subdivisées en trois catégories: les données textuelles, les données catégorielles et les données numériques qui représentent respectivement: le texte de la nouvelle, la source de la nouvelle avec son auteur enfin la date et le sentiment dégagé par la nouvelle. Le pré-traitement de chaque catégorie est effectué à travers un ensemble d'opérations:

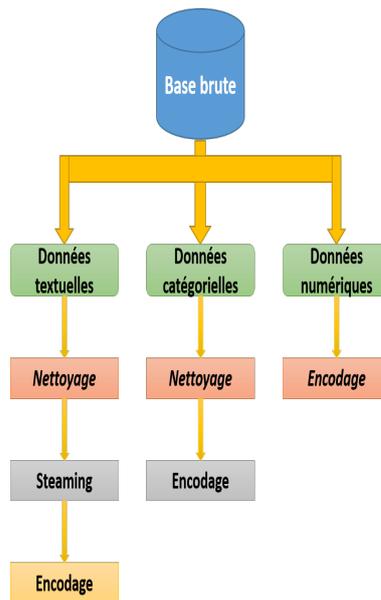


Figure 3.2: Pré-traitement

- **Données textuelles:** représentent le texte brute rédigé par l'auteur et pré-traité par les opérations suivantes:
  - **Nettoyage:** consiste à éliminer les mots vides tel que *a, about, am, you, are...* et les caractères spéciaux tel que *!, ?, :, ;, , ...* et toute information non utile, dans notre cas il s'agit des chiffres dans le texte.
  - **Stemming:** consiste à transformer les mots utiles en des racines par exemple, les mots *actor, acting, reenact* sont tous transformés en la même racine *act*.
  - **Encodage:** transformer l'ensemble des mots du commentaire en un vecteur numérique en passant par deux étapes: la combinaison des deux techniques: sac à mots et *N-gram*, puis l'application de la méthode TF-IDF sur le résultat.

\* Sac à mots: dans ce modèle, le texte est représenté sous forme de vecteur contenant ses mots, sans tenir compte de leur ordre, mais en gardant la multiplicité. Cette technique est principalement utilisée pour calculer différentes mesures qui caractérisent le texte par exemple, le mot le plus répété dans l'ensemble des documents c-à-d le *dataset* appelé corpus. Mais le problème ici est que, un mot qui se répète dans tout le corpus ne veut pas dire qu'il est vraiment important ou il caractérise un document précis. Pour résoudre ce problème, la fréquence du terme est pondérée par l'importance du document dans le corpus. Dans notre système nous avons utilisé la méthode TF-IDF [AZ12].

TF-IDF (*Term Frequency-Inverse Document Frequency*) est une mesure statistique permet d'évaluer l'importance d'un terme contenu au sein d'un document, dans un corpus. Le poids augmente proportionnellement au nombre d'occurrences du mot dans le document. Cette méthode est utilisée dans des moteurs de recherche pour apprécier la pertinence d'un document à une requête [AZ12].

Plusieurs formules de calcul ont été proposé pour cette méthode. Dans notre travail nous avons utilisé la formule suivante:

TF(t) = Nombre d'apparition du terme t dans le document (n) / Nombre total de termes dans le document, en gardant la multiplicité de chaque terme (k).

$$TF_t = \frac{n}{k}$$

IDF(t) = Nombre totale des documents (D) / Nombre des documents citant ce terme ( $\dot{D}_t$ ).

$$IDF_t = \log \frac{D}{\dot{D}_t}$$

\* N-gram: dans cette technique, le texte est représenté sous forme de vecteur contenant des blocs de mots, en tenant compte de leur ordre, et en gardant la multiplicité [AZ12].

- **Données catégorielles:** représentent dans notre étude la source et l'auteur. Le pré-traitement de ces données passe par deux étapes:

- **Nettoyage:** consiste à éliminer les caractères spéciaux et la transformation

des lettres en minuscules.

- **Encodage:** pour les source nous avons utilisé une simple numérotation tel que dans l'exemple de la figure suivante:

Source(Catégorielle)	Source(Numérique)
100percentfedup.com	1
CNN	2
New York Times	3
amtvmmedia.com	4

Figure 3.3: encodage de la source

Pour les auteurs, nous avons créé un encodage spécial et évité d'utiliser la numérotation simple, car l'inconvénient de cet encodage est que, les algorithmes d'apprentissage peuvent considérer que la source '1' et la source '2' sont proches, alors qu'en réalité ça peut être faux. Pour cette raison nous avons créé notre propre encodage pour convertir le nom de l'auteur en numérique, de telle sorte que les auteurs de la même source soient proches entre eux par rapport aux auteurs des autres sources.

Nous avons créé alors une liste contenant deux champs le premier pour la source et le deuxième pour ses auteurs, puis nous avons remplacé chaque auteur par son numéro d'index en rajoutant la somme des tailles des sources précédentes plus un. La figure suivante illustre un exemple où:

- \*  $T$ : nombre des auteurs de la source (taille).
- \*  $i_k$ : indice de l'auteur numéro  $k$ .

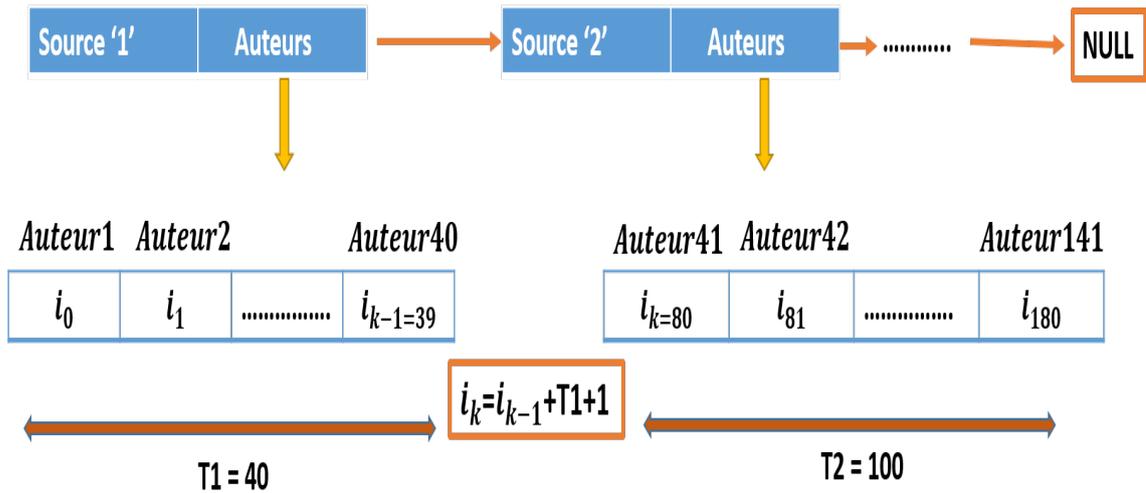


Figure 3.4: Calcul des indices des auteurs

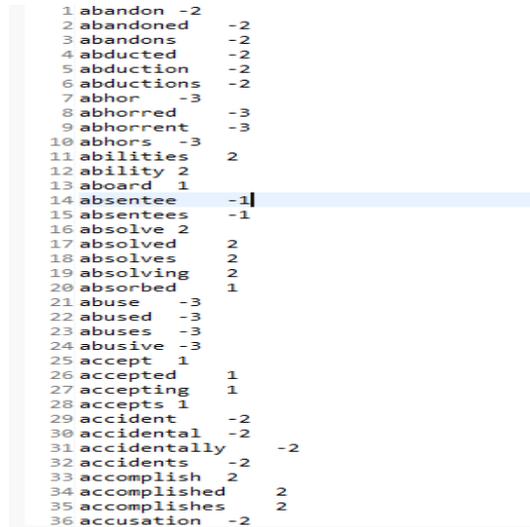
Source (catégorielle)	Source (numérique)	Taille de la source	Auteur (catégorielle)	Auteur (numérique)
100percentfedup.com	<b>1</b>	<b>3</b>	A1	1
100percentfedup.com			A2	2
100percentfedup.com			A3	3
CNN	<b>2</b>	<b>2</b>	A4	3+3+1=7
CNN			A5	8
New York Times	<b>3</b>	<b>2</b>	A6	8+2+1=11
New York Times			A7	12

Figure 3.5: Exemple de l'encodage des auteurs

- **Données numériques:** représentent la date de dépôt du commentaire et le sentiment dégagé par le texte. Puisque la date est déjà une valeur numérique nous l'avons seulement découpée en trois valeurs uniques: jour, mois et année. Pour le sentiment dégagé par la nouvelle, nous avons calculé la somme des degrés des mots. Selon les experts chaque mot a un degré de sentiment qui permet de le classer en trois classe [PL08]:
  - Si le degré est inférieur à 0, le sentiment est négatif.
  - Si le degré est supérieur à 0, le sentiment est positif.

- Si le degré est 0, le sentiment est neutre.

Voici un échantillon de la liste des mots et leurs degrés que nous avons utilisé pour calculer le sentiment dégagé par chaque information.



1	abandon	-2
2	abandoned	-2
3	abandons	-2
4	abducted	-2
5	abduction	-2
6	abductions	-2
7	abhor	-3
8	abhorred	-3
9	abhorrent	-3
10	abhors	-3
11	abilities	2
12	ability	2
13	aboard	1
14	absentee	-1
15	absentees	-1
16	absolve	2
17	absolved	2
18	absolves	2
19	absolving	2
20	absorbed	1
21	abuse	-3
22	abused	-3
23	abuses	-3
24	abusive	-3
25	accept	1
26	accepted	1
27	accepting	1
28	accepts	1
29	accident	-2
30	accidental	-2
31	accidentally	-2
32	accidents	-2
33	accomplish	2
34	accomplished	2
35	accomplishes	2
36	accusation	-2

Figure 3.6: Degrés des mots

### 3.3.2 Apprentissage:

il regroupe deux modules, l'entraînement et la validation utilisant chacun une partie de la base des caractéristiques subdivisée en deux parties, base d'entraînement et base de test. Le module d'entraînement utilise la base d'entraînement pour fournir un modèle de décision tandis que le module de validation utilise la base de test pour mesurer la performance du modèle fourni.

- **Entraînement:** pour entraîner notre modèle, nous avons choisis l'algorithme que nous avons déjà présenté dans le chapitre précédent; la machine à vecteurs de support, pour deux raison:
  1. Parce qu'il donne les meilleures résultats au niveau du *Texte Mining* [ATS17].
  2. Pour utiliser la valeur de la fonction de décision comme un degré de confiance pour la classification des nouvelles. Une valeur positive de la fonction de décision désigne en même temps une nouvelle vraie ainsi que son degré de vérité et vise versa, une valeur négative de la fonction de décision désigne une fausse nouvelle ainsi que son degré de faux.

On calcule donc, lors de l'entraînement le maximum et le minimum de la fonction de décision. Lors de l'utilisation le degré de vérité ou de faux est calculé par la fonction suivante:

$$\begin{cases} Dec > 0 & p = \frac{Dec}{Max_{dec}} * 100 \\ Sinon & p = \frac{Dec}{Min_{dec}} * 100 \end{cases} \quad (3.1)$$

Où:

- $Dec$ : est la valeur de la fonction de décision.
- $Max_{dec}$  et  $Min_{dec}$ : représentent les valeurs maximales et minimales de la fonction de décision.
- $p$ : est le pourcentage de vérité ou de faux.

La figure suivante illustre cette idée:

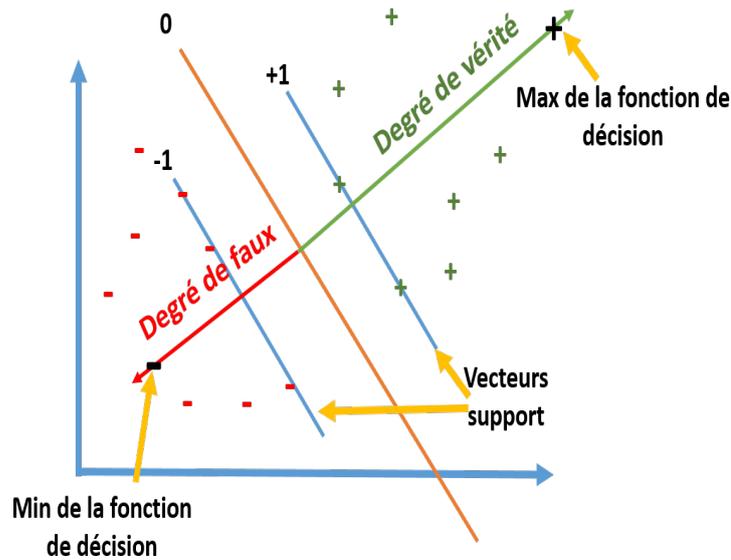


Figure 3.7: Degré de confiance de la classification des nouvelles

Le résultat de l'entraînement est un modèle ou pattern, qui représente l'analyse des données et leur transformation en informations utiles, en établissant des relations entre elles. Plusieurs métriques sont utilisées pour estimer la qualité du modèle qui se basent sur les valeurs suivantes:

1. VP: les exemples positifs classés correctement.
2. FP: les exemples positifs mal classés .

3. VN: les exemples négatifs classés correctement.
  4. FN: les exemples négatifs mal classés.
- **Précision:** proportion des exemples positifs correctement classés dans l'ensemble des exemples positifs.

$$P = \frac{VP}{VP + FP} \quad (3.2)$$

- **Rappel:** proportion des exemples positifs correctement classés VP par rapport aux exemples classés positifs (VP +FN).

$$R = \frac{VP}{VP + FN} \quad (3.3)$$

- **Fmesure:** moyenne harmonique de la précision et du rappel. Elle mesure la capacité du système à donner toutes les solutions pertinentes et à refuser les autres.

$$F = \frac{2 * P * R}{P + R} \quad (3.4)$$

- **Validation:** consiste à mesurer la capacité du modèle a reconnaître des nouveaux exemples. Pour cela, on écarte dès le départ une partie des exemples pour les utiliser pour le test du modèle. La base des caractéristiques est alors subdivisée en deux parties, une partie d'entraînement et une partie de test. Son utilité consiste a éviter le sur-apprentissage, c-à-d tester le modèle sur la même base d'entraînement. La subdivision n'est pas faite au hasard mais selon un échantillonnage particulier [PLH09]:

1. **Holdout method:** le *dataset* de taille n est subdivisé en deux partie, la première généralement de 60% ou plus: apprentissage, et la deuxième de 40%: test.

2. ***K-fold cross-validation***: le *dataset* est subdivisé en  $m$  parties,  $m-1$  parties pour l'apprentissage et une pour le test. Cette opération est répétée  $m$  fois et à chaque fois on obtient un taux de reconnaissance. À la fin on calcul la moyenne et l'écart types de ces taux pour estimer la performance du modèle.
  3. ***Leave-one-out cross-validation***: le *dataset* est subdivisé en  $m$  partie tel que  $m=k$  exemples où  $k$  représente le nombre d'exemples total de la base. À chaque opération l'apprentissage se fait sur  $k-1$  et le test sur l'exemple qui reste. C'est un cas particulier de la validation croisée.
- **Révision des paramètres**: cette opération a pour objectif d'améliorer le modèle, avec le *tuning* ou le réglage des paramètres de la machine à vecteurs de support et de changer la variante de validation croisée ou la valeur de  $k$  au cas de *k-folds cross validation*. Il existe de nombreux paramètres de l'svm mais, les plus important sont:
    - **Cost**: ce paramètre désigne l'optimisation de l'svm pour éviter de mal classer les données d'entraînement. Pour des valeurs élevées de  $C$ , l'optimisation choisira un hyper-plan à plus petite marge, inversement, une très petite valeur de  $C$  amènera l'optimisation à rechercher un hyper-plan de séparation à plus grande marge.
    - **Gamma**: ce paramètre définit jusqu'où l'influence d'une seule donnée d'entraînement atteint, avec des valeurs faibles une signification «loin» et des valeurs élevées une signification «proche».
    - **degré**: ce paramètre présente le degré du noyau.
    - **Epsilon**: ce paramètre détermine la tolérance du critère de terminaison. C'est le taux d'erreur permis.

### 3.3.3 Utilisation

C'est la dernière phase et la plus importante dans notre système. Après être arrivé au meilleur taux de reconnaissance, ou après avoir construit le meilleur modèle dans la phase précédente, nous devons l'utiliser sur des nouvelles informations non étiquetées, et le modèle nous permet de prédire la classe de la nouvelle si elle est fausse ou vraie avec un degré de confiance comme suit:

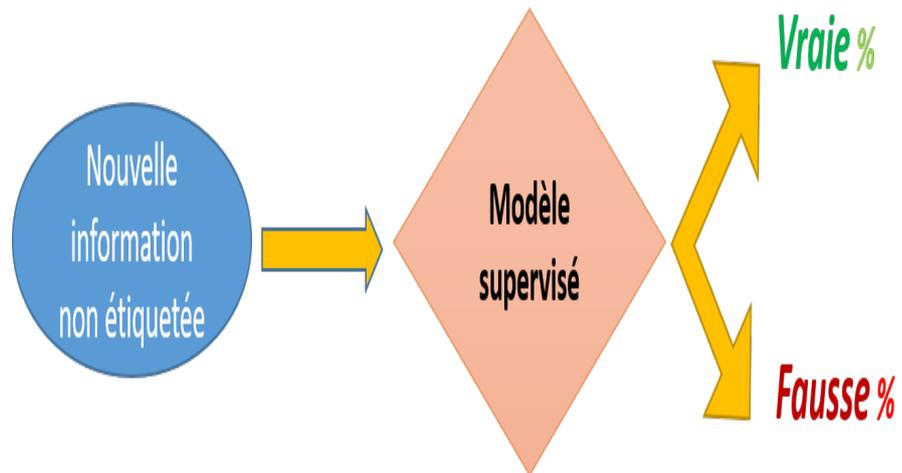


Figure 3.8: Utilisation

### 3.4 Conclusion

Ce chapitre a décrit la conception de notre système et il a présenté la démarche suivie dans ses différentes phases. Dans le chapitre suivant, nous allons décrire le fonctionnement de notre application mettant en oeuvre le système proposé.

## *Implémentation et résultats*

---

4.1	Introduction . . . . .	38
4.2	Outils utilisés . . . . .	39
4.2.1	Logiciels utilisés . . . . .	39
4.2.2	Langages utilisés . . . . .	40
4.2.3	Bibliothèques utilisées . . . . .	41
4.2.4	Struts 2 . . . . .	44
4.2.5	Dataset . . . . .	45
4.3	Application: . . . . .	47
4.4	Résultats et discussion: . . . . .	52
4.4.1	Pré-traitement: . . . . .	52
4.4.2	Apprentissage: . . . . .	54
4.5	Conclusion . . . . .	57

---

### 4.1 Introduction

L'implémentation d'un système informatique, désigne sa réalisation et sa mise en oeuvre, pour passer à l'interaction avec les utilisateurs. L'objectif de ce chapitre est de présenter les outils (les logiciels, les langages, les bibliothèques et les données utilisés dans notre système, ensuite nous aborderons l'application, puis nous discuterons les résultats.

## 4.2 Outils utilisés

### 4.2.1 Logiciels utilisés

Dans notre travail nous avons utilisé l’environnement Eclipse pour la phase du pré-traitement et le logiciel Weka pour l’entraînement et la validation. Pour l’utilisation nous avons créé une application Web, en intégrant le Weka et le *Framework*<sup>1</sup> Struts2 dans l’environnement Eclipse IDE pour Java EE Developers.

- **Eclipse:** c’est un logiciel qui vise à développer un environnement de production de logiciels libre qui soit extensible, en s’appuyant principalement sur Java [Bur05].

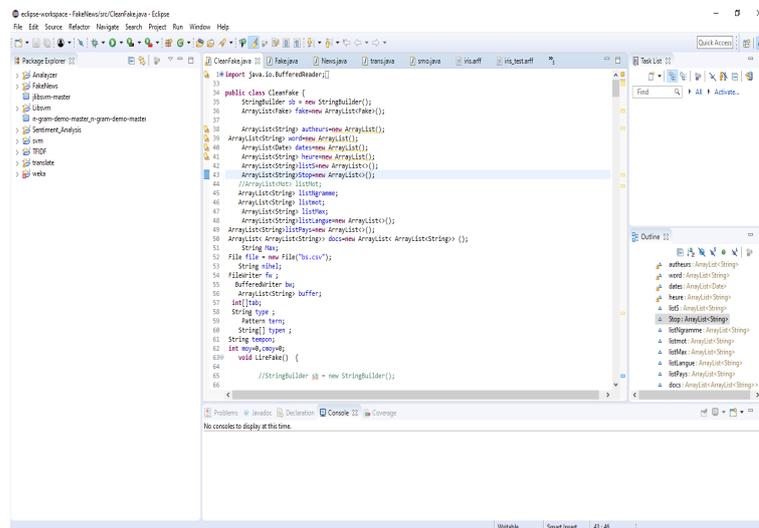


Figure 4.1: Eclipse

- **Eclipse IDE pour Java EE Developers:** c’est un logiciel pour les développeurs Java créant des applications JavaEE et Web [Bur05].
- **Weka:** c’est un logiciel d’apprentissage automatique *Open Source* approuvé, à travers lequel on peut accéder à des applications de terminal standard ou une API Java. Il est largement utilisé pour l’enseignement, la recherche et les applications

<sup>1</sup>Framework désigne un ensemble cohérent de composants logiciels structurels, qui sont utilisés pour modéliser l’architecture des logiciels applicatifs et des applications Web.

industrielles, il contient aussi un accès à des boîtes bien connues telles que *Scikit-Learn*, R et *Deeplearning4j* [HDW94].

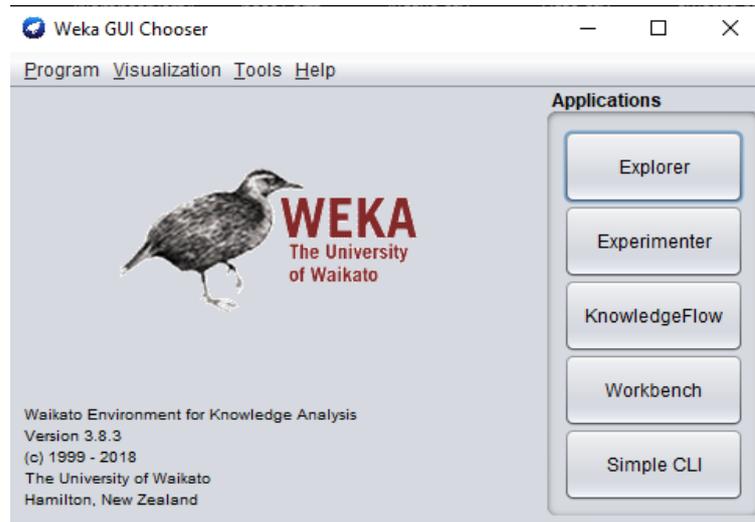


Figure 4.2: Weka

## 4.2.2 Langages utilisés

Dans notre travail nous avons le langage Java en premier lieu pour le pré-traitement de la base ensuite pour l'application web nous avons intégré le *Framework Struts2* qui englobe les langages: HTML, CSS et JavaScript pour l'interface, ainsi que le langage JavaEE pour le traitement.

- **Java:** c'est une technique informatique développée initialement par *Sun Microsystems* puis acquise par Oracle. C'est un langage de programmation orienté objet qui regroupe plusieurs logiciels, frameworks, bibliothèques, ...etc [AGHH00].
- **JavaEE:** (Java Enterprise Edition) est une évolution du Java destinée à une utilisation "côté serveur" pour un système d'information de l'entreprise [JCNE<sup>+</sup>14].
- **HTML:** *HyperText Markup Language*, c'est un langage de balisage conçu pour représenter les pages Web, permettant d'écrire de l'hyper-texte et de structurer sémantiquement la page [HA98].

- **CSS:** *Cascading Style Sheets*, c'est un langage qui décrit la présentation des documents HTML et XML [Mey04].
- **JavaScript:** c'est un langage de programmation de scripts principalement employé dans les pages Web avec les technologies HTML et CSS [JMT09].

### 4.2.3 Bibliothèques utilisées

Dans chaque phase de notre travail nous avons eu besoin d'intégrer quelques bibliothèques, d'abord, en pré-traitement où nous avons utilisé deux bibliothèques *Analyzer et Porter*, ensuite dans la deuxième phase nous avons entraîné les données de notre *dataset* avec l'algorithme de la machine à vecteurs de support en utilisant la bibliothèque LibSvm que nous avons intégré dans Weka. A la fin, à l'étape de l'utilisation nous avons fait appel à *WhatsMate Translation API* pour traduire les informations entrées par les utilisateurs en anglais.

- **Analyzer:** une annexe de la bibliothèque *Apache Lucene*, les analyseurs gèrent le travail d'analyse du texte en un vecteur de *token*, il existe plusieurs analyseurs qui se différencient de leurs fonctions, nous avons choisis *standard analyzer* qui élimine les caractères spéciaux et les numéros et même quelques mots vides [Ana].

```

1 import java.io.IOException;
2 import java.io.StringReader;
3 import java.util.ArrayList;
4 import java.util.List;
5 import org.apache.lucene.analysis.Analyzer;
6 import org.apache.lucene.analysis.TokenStream;
7 import org.apache.lucene.analysis.standard.StandardAnalyzer;
8 import org.apache.lucene.analysis.tokenattributes.CharTermAttribute;
9 import org.apache.lucene.util.Version;
10 public final class TokenizewithAnalyzer {
11     private TokenizewithAnalyzer() {}
12
13     public static String tokenizeString(Analyzer analyzer, String str) {
14         String result = null ;
15         try {
16             TokenStream stream = analyzer.tokenStream(null, new StringReader(str));
17             stream.reset();
18             while (stream.incrementToken()) {
19                 result=stream.getAttribute(CharTermAttribute.class).toString();
20             }
21         } catch (IOException e) {
22             // not thrown b/c we're using a string reader...
23             throw new RuntimeException(e);
24         }
25         return result;
26     }
27 }

```

Figure 4.3: Algorithme Analyzer en java

- **Porter:** un des algorithmes les plus connus pour le *steaming* d'un mot c'est à dire le transformer à son radical en supprimant ses préfixes et suffixes, développé par Julie Beth Lovins en 1968 et Martin Porter en 1980 [Wil06].

```
public void mot(String Smot){
    listmot=new ArrayList();
    String[] typen=Smot.split("\\s");
    for(String mot:typen) {
        if(main.Analayser(mot)!=null) {
            if(!main.BooleanstopWords(main.Analayser(mot))) {

                PorterStemmer stemmer = new PorterStemmer();
                stemmer.setCurrent(main.Analayser(mot));
                stemmer.stem();
                String racine=stemmer.getCurrent();

                listmot.add(racine);
            }
        }
    }
}
```

Figure 4.4: Algorithme Porter en java

- **SMO:** L'optimisation minimale séquentielle inventée par John Platt en 1998 chez Microsoft Research, un algorithme pour résoudre le problème de programmation quadratique (QP) qui se pose lors de la formation des machines à vecteurs de support (SVM) [Pla98].
- **LibSvm:** est une bibliothèque de *machine learning open source* populaire, développée à la *National Taiwan University*, écrite en C++ avec une API C. LibSvm implémente l'algorithme d'optimisation minimale séquentielle (SMO) pour les machines à vecteurs de support (SVM), prenant en charge la classification(one-classe/binaire/multi-classes) et la régression [CL11].

```

InputStream st= ServletActionContext.getServletContext().getResourceAsStream("/arff/BaseE.arff");
BufferedReader datafile = new BufferedReader(new InputStreamReader(st));
Instances data = new Instances(datafile);
data.setClassIndex(data.numAttributes() - 1);
WekaPackageManager.loadPackages( false, true, false );
LibSVM smo=new LibSVM();
String options = ( "-S 0 -K "+k+" -D 3 -G "+G+" -R "+cf+" -N 0.5 -M 40.0 -C "+C+" -E "+E+" -P 0.1" );
String[] optionsArray = options.split( " " );
smo.setOptions( optionsArray );
smo.buildClassifier(data);
int seed=1,folds=10;
Random rand=new Random(seed);
Evaluation eval=new Evaluation(data);
eval.crossValidateModel(smo, data, folds, rand);

```

Figure 4.5: LibSvm en java

- **LIBLINEAR**: implémente des SVM linéaires et des modèles de régression logistique formés à l'aide d'un algorithme de descente de gradient [CL11].
- **WhatsMate Translation API** : une *API Rest* qui permet aux applications de traduire facilement des textes d'une langue à une autre [Wha].

```

public String CLIENT_ID = "FREE_TRIAL_ACCOUNT";
public String CLIENT_SECRET = "PUBLIC_SECRET";
public String ENDPOINT = "http://api.whatsmate.net/v1/translation/translate";
public String translate(String fromLang, String toLang, String text) throws Exception {
    String jsonPayload = new StringBuilder()
        .append("{")
        .append("\"fromLang\":\"")
        .append(fromLang)
        .append("\",")
        .append("\"toLang\":\"")
        .append(toLang)
        .append("\",")
        .append("\"text\":\"")
        .append(text)
        .append("\"}")
        .toString();

    URL url = new URL(ENDPOINT);
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setDoOutput(true);
    conn.setRequestMethod("POST");
    conn.setRequestProperty("X-WM-CLIENT-ID", CLIENT_ID);
    conn.setRequestProperty("X-WM-CLIENT-SECRET", CLIENT_SECRET);
    conn.setRequestProperty("Content-Type", "application/json");
    OutputStream os = conn.getOutputStream();
    os.write(jsonPayload.getBytes());
    os.flush();
    os.close();
    System.out.println("os: " + os);
    int statusCode = conn.getResponseCode();
    System.out.println("Status Code: " + statusCode);
    BufferedReader br = new BufferedReader(new InputStreamReader(
        (statusCode == 200) ? conn.getInputStream() : conn.getErrorStream()
    ));
    String output=null;
    String sb = null;
    while ((output = br.readLine()) != null) {
        if(sb==null) {
            sb=output;
        }else {
            sb=sb+" "+output;
        }
        System.out.println(output);
    }

    conn.disconnect();
    return sb; }

```

Figure 4.6: WhatsMate Translation API en java

## 4.2.4 Struts 2

Un *Framework* qui permet de construire des applications Web Java EE respectant le modèle d'architecture MVC (Modèle-vue-contrôleur) [Baa17].

### 4.2.4.1 Modèle-vue-contrôleur:

Paradigme de construction d'IHM, composé de trois types de modules ayant trois responsabilités différentes: les modèles, les vues et les contrôleurs [Baa17].

- **Modèle:** c'est l'ensemble des actions, les données à afficher. C'est les java beans, les servlets, les bases de données.
- **Vue:** c'est l'aspect graphique du système. L'utilisateur interagit avec cette vue. C'est les fichiers JSP <sup>2</sup>.
- **Contrôleur:** c'est le traitement des requêtes du client fournis par *Apache Tomcat*, c'est un des nombreux projets de l'Apache Software Foundation. Il implémente les spécifications des servlets et des JSP du Java Community Process9, paramétrable par des fichiers XML et des propriétés. Il comporte également un serveur HTTP. C'est les Servlets, c'est une classe Java qui permet de créer dynamiquement des données au sein d'un serveur HTTP. Ces données sont le plus généralement présentées au format HTML, mais elles peuvent également l'être au format XML ou tout autre format destiné aux navigateurs Web.

---

<sup>2</sup>Code html qui encapsule du code java comme traitement, il peut intégrer le code css et java script aussi. en étant un code html

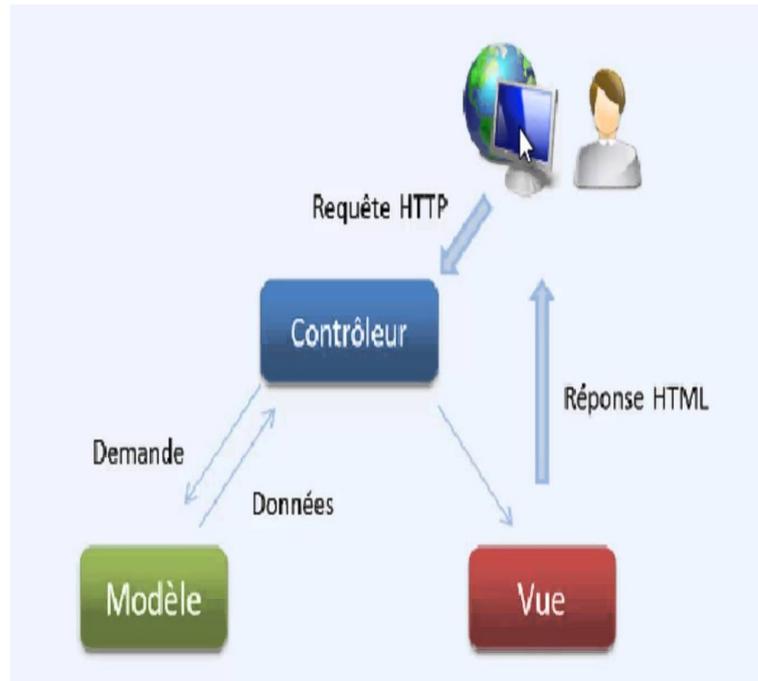


Figure 4.7: Modèle-vue-contrôleur

[Baa17]

### 4.2.5 Dataset

Comme n'importe quel système utilisant le *Machine Learning*, nous avons besoin d'un *dataset* pour l'entraîner sur les données en utilisant ses algorithmes pour prédire la classe des données de la base de test.

Le *dataset* de *Fake News* brute que nous avons utilisé et préparé, chaque information a été transformée en un vecteur de caractéristiques et enregistrée dans un fichier CSV. Le fichier est ensuite subdivisé en deux parties, une consacrée à l'entraînement et l'autre au test et validation. Après la validation, nous sommes passées à l'étape de l'utilisation qui consiste à prédire la classe des nouvelles informations non étiquetées (vraie/fausse).

À notre connaissance, et d'après nos recherches un *dataset* publique pour les *Fake News* contenant à la fois des fausses et vraies nouvelles contenant suffisamment de caractéristiques n'existe pas, ce qui nous a poussé à construire notre propre base. Nous avons alors fusionné deux bases existantes "Getting Real about Fake News" contenant des fausses nouvelles et "All the news" contenant les vraies nouvelles. Ces bases ont été obtenues du site Kaggle qui est une plate-forme Web organisant des compétitions en science des don-

nées. Les entreprises proposent des problèmes en science des données et offrent un prix aux data-logistes obtenant les meilleures performances. L'entreprise a été fondée en 2010 par *Anthony Goldbloom*.

1. **Getting Real about Fake News:** contient du texte et des méta-données extraits de 244 sites Web marqués comme fausse par le détecteur *BS Detector Chrome* de Daniel Sieradski, extraites à l'aide de l'API *webhose.io*. Cette base contient environ 12,999 publications sur les réseaux sociaux, divisés en 20 colonnes ayant des types différents; catégorielles, numériques et textuelles.
2. **All the news:** contient du texte et des méta-données extraits de New York Times, Breitbart, CNN, Business Insider, Atlantique, Fox News, Talking Points Memo, Buzzfeed News, National Review, New York Post, The Guardian, NPR, Reuters, Vox et le Washington Post, récupérés à l'aide de BeautifulSoup et stockées dans Sqlite, découpé en trois fichiers CSV distincts. Cette base contient du texte et des méta-données subdivisés en 10 colonnes ayant des types différents; catégorielles, numériques et textuelles.

Après avoir pré-traité les deux bases et testé les caractéristiques une par une jusqu'à arrivé au meilleur taux de précision. Nous avons obtenu un *dataset* qui contient 900 publications fausses et 975 publications réelles. Les caractéristiques finales:

- 5 mots obtenus par la méthode sac à mots,
- 3 mots composés obtenus par la méthode N-gram,
- date: jour, mois et année,
- sentiment,
- source,
- auteur,
- classe: fausse ou vraie.

### 4.3 Application:

Dans notre application Web nous avons quatre fenêtres principales représentant les différentes tâches effectuées pour la détection des fausses nouvelles à savoir: le pré-traitement, l'entraînement, le test et l'utilisation.



Figure 4.8: Interface d'accueil de l'application

- **Fenêtre de pré-traitement:** c'est une fenêtre qui contient une brève description de notre *dataset* et les étapes principales suivies pour le pré-traitement. Elle contient aussi deux liens vers la source de *dataset*, pour mieux comprendre la description des données, si l'utilisateur avait des données de la même structure, la fenêtre lui donnera la main pour entrer les deux bases *Fake* et *Real* pour commencer le pré-traitement, ensuite sauvegarder le résultat dans un fichier CSV.



Figure 4.9: Fenêtre de Pré-traitement

- Fenêtre d'entraînement:** c'est une vue qui permet aux utilisateurs d'entraîner les données et voir les performances du modèle (Rappel, Précision, Fmesure, Matrice de confusion, taux correctes et incorrectes), en entrant un *dataset* ayant la même structure que le notre, et en réglant les valeurs de SVM (Cost, gamma, kernel...). L'utilisateur peut à la fin enregistrer le meilleur modèle obtenu.



PRÉTRAITEMENT

ENTRAÎNEMENT

TEST

UTILISATION

**Parametres SVM**

kernel  
RBF exp(-gamma\*|U-V|^2)

C  
1

gamma  
0

degree  
3

Coeff  
0

Eps  
0.001

Cross validation folds  
10

**Entraîner**

S'entraîner avec votre base : Choisir un fichier Aucun fichier choisi

**Resultat :**

Correcte :  
Incorrecte :  
Précision :  
Repelle :  
Fmesure :  
matrice de confision :

Pour telecharger la base traitée : [Download](#)

Activer Windows  
Accédez aux paramètres pour activer Windo

Figure 4.10: Fenêtre d'entraînement

- **Fenêtre de test:** c'est une fenêtre qui permet aux utilisateurs de tester des modèles sur des bases autres que celles utilisées lors de l'entraînement. L'utilisateur doit alors fournir un modèle entraîné et une base contenant des nouvelles déjà classées pour obtenir les performance du modèle sur cette base.

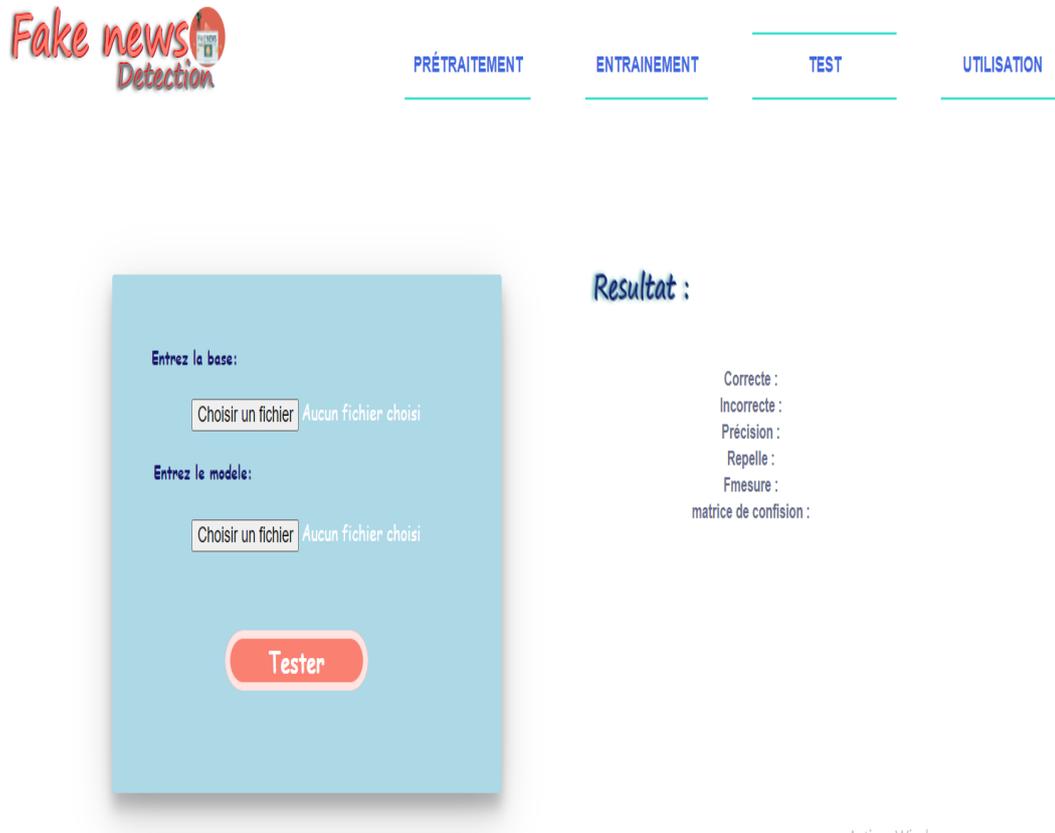


Figure 4.11: Fenêtre de test

- **Fenêtre de l'utilisation:** contrairement aux fenêtres précédentes destinées aux utilisateurs connaissant le *machine learning*, c'est une fenêtre destinée à tout types d'utilisateurs. L'utilisateur peut entrer une nouvelle avec ses caractéristiques sans la classe et un modèle déjà entraîné pour obtenir la classe de la nouvelle vraie ou fausse avec un degré de vérité.



Figure 4.12: Fenêtre d'utilisation

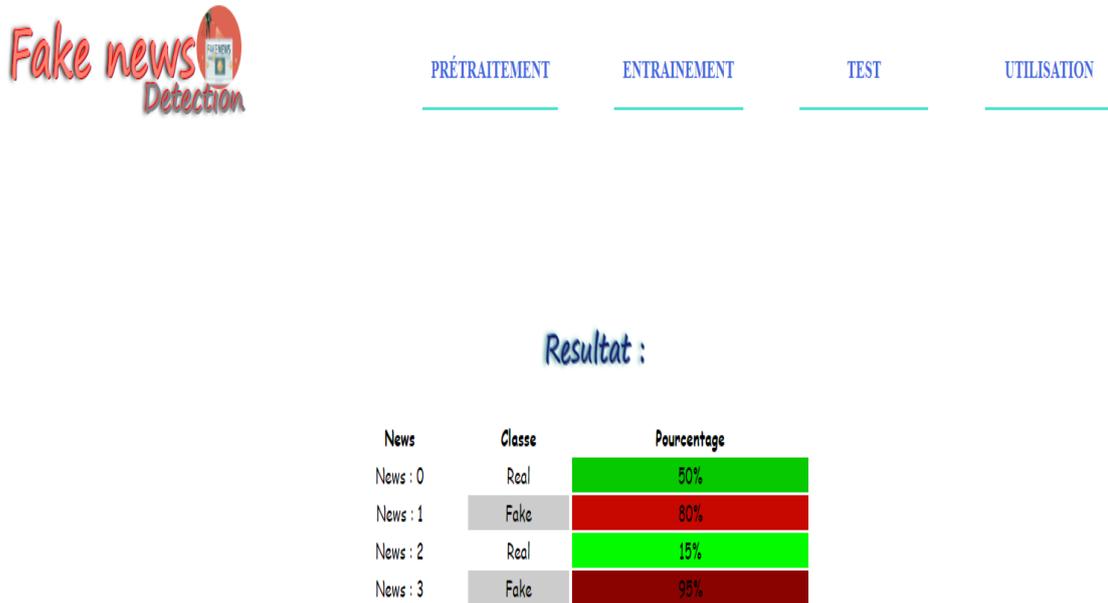


Figure 4.13: Résultat de l'utilisation

## 4.4 Résultats et discussion:

Pour obtenir le meilleur modèle de décision permettant de classer le mieux les nouvelles, nous avons procédé à un *tunning* des paramètres dans nos deux phases de pré-traitement et d'apprentissage.

### 4.4.1 Pré-traitement:

Dans cette phase nous avons essayé d'obtenir les meilleurs paramètres des deux techniques sac à mot et n-gram qui donnent le meilleur taux de reconnaissance sur notre base.

1. **Sac a mots:** pour cette technique nous avons aiguillé le nombre de mots les plus fréquents pris de chaque commentaire. Cette opération est répétée plusieurs fois jusqu'à arrivé au meilleure taux, en augmentant à chaque fois le nombre des mots les plus fréquents. Sur le logiciel Weka et en utilisant la bibliothèque SM0, et avec la validation croisée pour 10 parties, nous avons obtenues es résultats suivants:

Mots	3	5	7	10	15	20	25
<i>Taux%</i>	72.33	74.80	75.81	75.85	77.84	81.10	81.88

Tableau 4.1: Taux de reconnaissance en fonction du nombre de mots fréquent pour le sac à mots.

Comme le montre la figure (4.1) le taux de reconnaissance augmente avec le nombre de mots les plus fréquents, jusqu'à 25 mots, puis commence à diminuer, ce que nous expliquons par le phénomène de sur-apprentissage.

2. **N-gram:** pour cette technique, nous avons aiguillé le nombre de grams. Cette opération a été aussi répétée plusieurs fois en augmentant la valeur du n à chaque fois. Nous avons obtenu ces résultats:

n-grams	2-grams	3-grams	4-grams
<i>Taux%</i>	68.95	66.59	63.77

Tableau 4.2: Différents taux de reconnaissance des n-grams.

Après 2-grams le taux de reconnaissance a commencé à diminuer ce qui est très logique, à cause de la petite taille du texte de la nouvelle; un bloc de mots de plus de 2 ne va pas se répéter plusieurs fois dans la même nouvelle qui ne va pas dépassée 5 lignes au plus.

Nous nous somme arrêtées à 2-grams et nous avons procédé à l'aiguillage du nombre de blocs de mots les plus fréquents k. Nous avons obtenu les résultats suivants:

k*(2-grams)	3*(2-grams)	5*(2-grams)	10*(2-grams)	25*(2-grams)	30*(2-grams)
<i>Taux%</i>	68.95	71.76	73.56	73.67	74.62

Tableau 4.3: Différents taux de reconnaissance des k\*(n-grams).

Ici le taux a continué à augmenter sans dépasser le taux obtenu par la technique sac à mots. Ceci est dû à notre avis à deux possibilités: soit la petite taille du texte de l'information, soit l'incompatibilité des n-grams avec la méthode TF-IDF. Nous avons pensé alors à combiner les deux techniques.

3. Les autres caractéristiques:

Nous avons commencé par combiner 5 mots fréquents avec les 3\*(2-gram) fréquents ce qui nous a donné un taux de 52,30%. Ensuite, nous avons ajouté les autres caractéristiques pour mesurer l'influence de chacune sur le taux de reconnaissance.

La figure suivante illustre les résultats:

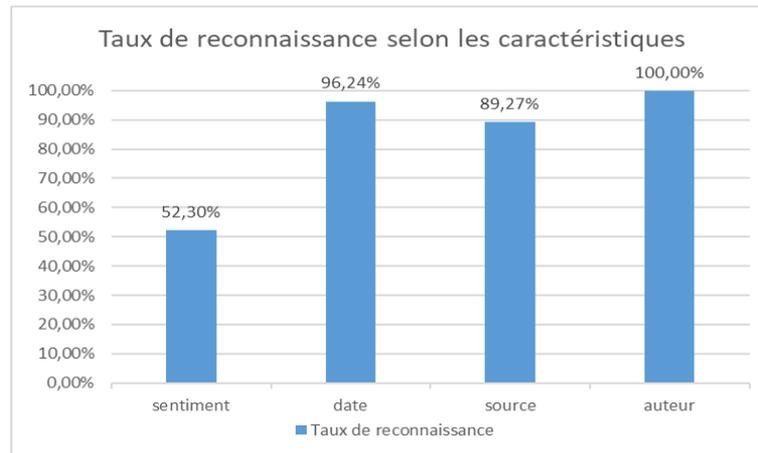


Figure 4.14: Influence des différentes caractéristiques sur le taux de reconnaissance

Nous remarquons sur ce graphe que l'influence de la caractéristique "sentiment" sur le taux de reconnaissance est presque négligeable, cependant la caractéristique "source" l'a augmenté jusqu'à 89.27%, et "date" jusqu'à 96,24%. Alors que la caractéristique auteur l'a poussé jusqu'à 100%, ce qui montre l'efficacité de l'encodage que nous avons proposé.

#### 4.4.2 Apprentissage:

Après le test des différents noyaux et leur tuning. Le tableaux suivant montre les résultats obtenus par les différents noyaux.

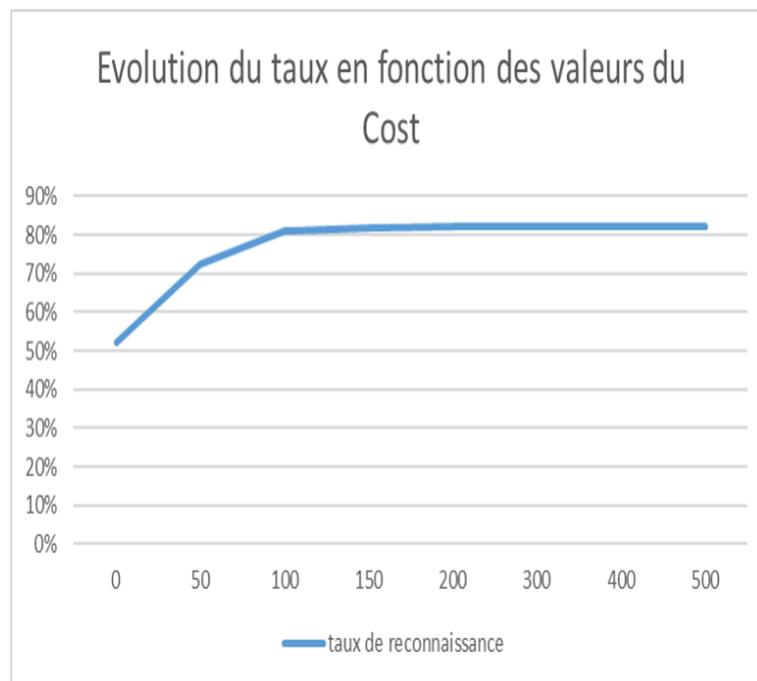
Noyau	Linière	Gaussien	Polynomiale
<i>Taux</i>	100%	99.73%	100%

Tableau 4.4: Valeur du taux par rapport au noyau.

Il est clair que les noyaux linéaire et polynomiale donnent les meilleurs résultats. Le noyau linéaire est sans paramètres et plus rapide cependant, théoriquement il n'arrive pas à modéliser les cas de chevauchement compliqué des deux classes. Par contre le noyau Gaussien permet de modéliser tout type de chevauchement mais son taux dépend des paramètres  $C$  (Cost),  $\epsilon$  et  $\gamma$ .

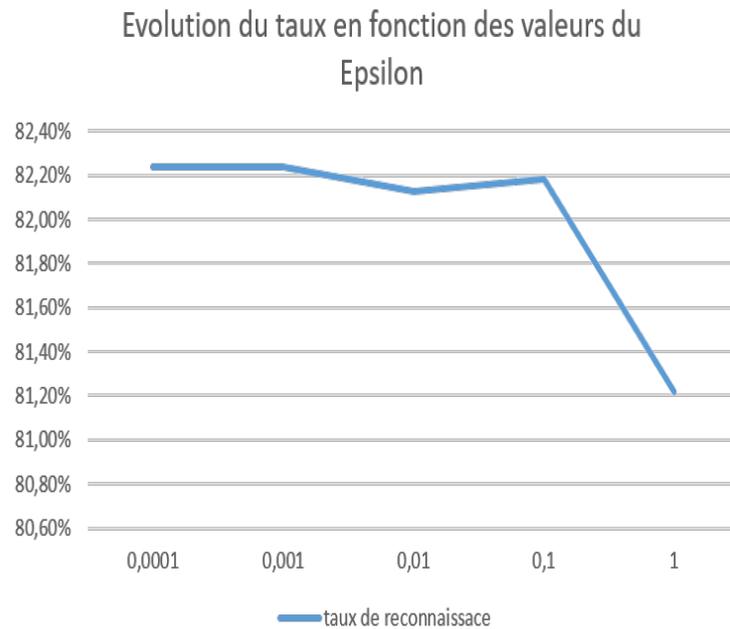
Nous avons étudié l'influence de ces paramètres sur la précision de l'apprentissage.

- **Influence au Cost C:** le graphe suivant donne l'évolution du taux de reconnaissance en fonction de  $C$ .



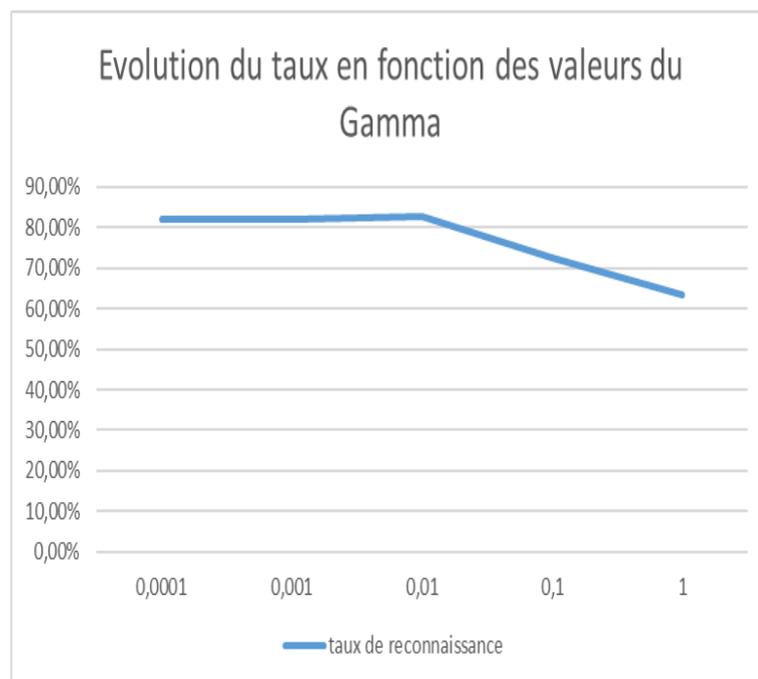
On observe une augmentation rapide du taux jusqu'à la valeur de 150, puis un léger ralentissement avec stabilisation aux alentours du taux de 82% puis une baisse à partir de la valeur de  $C=500$ .

- **Influence de  $\epsilon$** : le graphe suivant donne l'évolution du taux de reconnaissance en fonction de  $\epsilon$ .



Avec un C fixé à 300, le taux diminue en augmentant  $\epsilon$  à partir de la valeur 0.001.

- **Influence de  $\gamma$** : le graphe suivant donne l'évolution du taux de reconnaissance en fonction de  $\gamma$ .



Avec un  $C=100$  et un  $\epsilon = 0.001$ , le taux de reconnaissance augmente l'égermant jusqu'à la valeur de 0.001, puis une stabilisation aux alentours du taux 82% puis on observe une baisse rapide à partir de la valeur de  $\gamma = 0,01$ .

Les paramètres optimaux du noyaux Gaussien sont donc les suivants:

Paramètres	Cost	$\gamma$	$\epsilon$
<i>Valeur</i>	300	0.001	0.0001

Tableau 4.5: Valeurs optimales des paramètres du noyau Gaussien

## 4.5 Conclusion

Nous avons détaillé dans ce chapitre les différents outils utilisés pour la mise en oeuvre de notre proposition ainsi que les résultats obtenus et leur discussion, Nous avons montré qu'une représentation du texte des nouvelles par une combinaison des techniques de sac à mot et N-gram avec un encodage adéquat des auteurs et le noyaux Gaussien de la méthode SVM donnent des taux de reconnaissance très élevés.

## Conclusion Générale

Ce mémoire présente une méthode de détection des fausses nouvelles en utilisant la machine à vecteurs de support, en essayant de déterminer les meilleures caractéristiques et techniques permettant de mieux identifier une fausse nouvelle.

Nous avons commencé par étudier le domaine des fausses nouvelles, leur impact et les méthodes de leur détection.

Nous avons ensuite conçu et implémenté une solution qui se base sur l'utilisation des techniques de nettoyage, *steaming*, encodage par N-gram, sac à mots et TF-IDF pour le pré-traitement des textes brutes des nouvelles, ensuite l'extraction des autres caractéristiques permettant d'identifier une fausse nouvelle. À la fin nous avons appliqué l'algorithme de la machine à vecteurs de support sur notre base de caractéristiques pour construire un modèle permettant la classification des nouvelles informations.

A travers les recherches menées lors de cette étude, nous avons obtenus les résultats suivants:

- les meilleures caractéristiques permettant de détecter une fausse information sont par ordre: le texte, l'auteur, la source, la date et le sentiment.
- la démarche suivie a aboutit à un taux de reconnaissance atteignant jusqu'à 100%.
- l'analyse du sentiment dégagé par le texte est intéressante, toutefois elle serait plus influente dans le cas de *l'opinion mining*.
- la méthode des N-gram donne un résultat meilleur que le sac à mots avec des bases volumineuse et avec des textes de grandes tailles.
- la machine à vecteurs de support semble le meilleur algorithme pour détecter une fausse information, car il a donné un meilleur taux de reconnaissance, et a permis de donner à chaque information un degré de confiance pour sa classification.
- les paramètres influençant la machine à vecteurs de support sont par ordre: Cost C, gamma  $\gamma$  et epsilon  $\epsilon$ .

Toutefois, notre recherche n'a porté que sur un *dataset* de petite taille. a cause de la conjoncture actuelle due a la pandémie du covid19, qui nous a empêché de profiter des outils puissant de notre laboratoire d'informatique.

Le travail que nous avons réalisé pourrait être complété et poursuivi sous différents aspects. Il serait pertinent d'étendre cette étude avec un *dataset* plus volumineux, et d'évoluer son apprentissage supervisé par un autre en ligne pour une mise à jour continue et intégration automatique des nouvelles fausses informations.

# Bibliography

- [Abd19] Djeffal Abdelhamid. *Cour de classification*, 2019. [http://abdelhamid-djeffal.net/index.php?p=1\\_18\\_Cours](http://abdelhamid-djeffal.net/index.php?p=1_18_Cours), Université de Biskra. (pages 17, 18, 19, 20, 24).
- [AGHH00] Ken Arnold, James Gosling, David Holmes, and David Holmes. *The Java programming language*, volume 2. Addison-wesley Reading, 2000. (page 40).
- [AHC19] Sajjad Ahmed, Knut Hinkelmann, and Flavio Corradini. Combining machine learning with knowledge engineering to detect fake news in social networks-a survey. In *Proceedings of the AAAI 2019 Spring Symposium*, volume 12, 2019. (page 8).
- [Ana] Class analyzer. [https://lucene.apache.org/core/8\\_0\\_0/core/org/apache/lucene/analysis/Analyzer.html](https://lucene.apache.org/core/8_0_0/core/org/apache/lucene/analysis/Analyzer.html). (page 41).
- [ATS17] Hadeer Ahmed, Issa Traore, and Sherif Saad. Detection of online fake news using n-gram analysis and machine learning techniques. In *International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, pages 127–138. Springer, 2017. (pages 12, 33).
- [AZ12] Charu C Aggarwal and ChengXiang Zhai. *Mining text data*. Springer Science & Business Media, 2012. (page 30).
- [Baa17] Soheib Baarir. *Programmation Web: le framework Struts*. Université Paris ouest nentere la defense, 2017. (pages 44, 45).
- [Ben17] Younes Benzaki. L'apprentissage supervisé machine learning, 2017. <https://mrmint.fr/apprentissage-supervise-machine-learning>. (pages 16, 17, 24).
- [BL12] Dominique Boullier and Audrey Lohard. *Opinion mining et Sentiment analysis: Méthodes et outils*. OpenEdition Press, 2012. (page 11).

- [Bur05] Ed Burnette. *Eclipse IDE Pocket Guide: Using the Full-Featured IDE*. " O'Reilly Media, Inc.", 2005. (page 39).
- [Cha05] Malika Charrad. Techniques d'extraction de connaissances appliquées aux données du web. *Transformation*, 56:5–2, 2005. (page 10).
- [CL11] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011. (pages 42, 43).
- [CRC15] Niall J Conroy, Victoria L Rubin, and Yimin Chen. Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4, 2015. (page 12).
- [Gan18] Rohith Gandhi. Support vector machine — introduction to machine learning algorithms, 2018. <https://towardsdatascience.com/support-vector-machine>. (pages 20, 21, 22).
- [GM17] Mykhailo Granik and Volodymyr Mesyura. Fake news detection using naive bayes classifier. In *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, pages 900–903. IEEE, 2017. (page 12).
- [HA98] Yih-Shiuan Hu and Tejwansh S Anand. Hypertext markup language (html) extensions for graphical reporting over an internet, May 5 1998. US Patent 5,748,188. (page 40).
- [HDW94] Geoffrey Holmes, Andrew Donkin, and Ian H Witten. Weka: A machine learning workbench. In *Proceedings of ANZIIS'94-Australian New Zealand Intelligent Information Systems Conference*, pages 357–361. IEEE, 1994. (page 40).
- [JCNE<sup>+</sup>14] Eric Jendrock, Ricardo Cervera-Navarro, Ian Evans, Kim Haase, and William Markito. *The java ee 7 tutorial*, volume 1. Addison-Wesley Professional, 2014. (page 40).
- [JMT09] Simon Holm Jensen, Anders Møller, and Peter Thiemann. Type analysis for javascript. In *International Static Analysis Symposium*, pages 238–255. Springer, 2009. (page 41).

- [L17] Bastien L. Machine learning définition fonctionnement et secteurs d'application, 2017. <http://www.artificiel.net/machine-learning-definition>. (pages 15, 16, 25, 26).
- [Liu12] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012. (page 11).
- [Mar20] Search Engine Marketing. Réseau de neurones artificiels : quelles sont leurs capacités ?, 2020. <https://www.ionos.fr/digitalguide/web-marketing/search-engine-marketing/quest-ce-quun-reseau-neuronal-artificiel/>. (page 19).
- [MBNL99] Sanjay Kumar Madria, Sourav S Bhowmick, W-K Ng, and Ee-Peng Lim. Research issues in web data mining. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 303–312. Springer, 1999. (pages 9, 10).
- [Mey04] Eric A Meyer. *Cascading style sheets: The definitive guide*. " O'Reilly Media, Inc.", 2004. (page 41).
- [MKC18] Cédric Maigrot, Ewa Kijak, and Vincent Claveau. Fusion par apprentissage pour la détection de fausses informations dans les réseaux sociaux. *Document numérique*, 21(3):55–80, 2018. (page 12).
- [PL08] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis foundations and trends in information retrieval vol. 2, 2008. (page 32).
- [Pla98] John Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998. (page 42).
- [PLH09] Refaeilzadeh Payam, Tang Lei, and Liu Huan. Cross-validation. *Encyclopedia of database systems*, pages 532–538, 2009. (page 35).
- [Sau18] Florian Sauvageau. *Les fausses nouvelles, nouveaux visages, nouveaux défis. Comment déterminer la valeur de l'information dans les sociétés démocratiques?* Presses de l'Université Laval, 2018. (page 12).

- [VC12] G Vinodhini and RM Chandrasekaran. Sentiment analysis and opinion mining: a survey. *International Journal*, 2(6):282–292, 2012. (page 11).
- [Wha] Whatsmate translation api. <https://www.whatsmate.net/translation-api.html>. (page 43).
- [Wil06] Peter Willett. The porter stemming algorithm: then and now. *Program*, 2006. (page 42).
- [ZG20] Xichen Zhang and Ali A Ghorbani. An overview of online fake news: Characterization, detection, and discussion. *Information Processing & Management*, 57(2):102025, 2020. (pages 5, 6, 9).