الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة غرداية
Université de Ghardaïa

كلية العلوم والتكنولوجيا
Faculté des Sciences et de la Technologie

قسم الرياضيات والاعلام الآلي
Département de Mathématiques et d'Informatique

N° d'enregistrement
/...../...../...../...../.....

Mémoire

Pour l'obtention du diplôme de master

Domaine: Mathématiques et Informatique
Filière: Informatique
Spécialité: Systèmes Intelligents pour l'Extraction des Connaissances (SIEC)

Thème

Filtrage des contenus indésirables (Texte) sur Twitter : traditionnel vs bio-inspiré algorithmes

Soutenue publiquement le 19/06/2022

Par

Mahfoud BADJELMANE & Nacereddine OUALI

Devant le jury composé de

M. **Nacera BRAHIM**	**MCB**	**Univ.Ghardaïa**	**Examinateur**
M. **Ahmed SAIDI**	**MCB**	**Univ.Ghardaïa**	**Examinateur**
M. **Abdelkader BOUHANI**	**MCB**	**Univ.Ghardaïa**	**Encadreur**

Année universitaire 2021/2022

Résumé

Twitter est l'un des plus grands sites de réseaux sociaux au monde, où la quantité d'informations et des messages qui appelé "Tweet" échangés entre les utilisateurs augmente ; Malheureusement, cela a accru la propagation des spammeurs qui partagent des messages indésirables, qui contiennent souvent des publicités ou des programmes ou des liens qui contiennent des sites malveillants, donc ces messages texte peuvent être détectés et classés comme spam , pour cela il existe plusieurs algorithmes et techniques a utilisé .
C'est ce que nous avons abordé dans notre thèse, où nous avons utilisé une base de données unifiée et on a appliqué des techniques et des algorithmes traditionnels, et d'autre bio inspirés pour classer le message électronique comme spam ou non, et suivent les résultats, nous avons effectué un processus de comparaison où nous a conclu que l'algorithme Naïve Bayes donnait une précision de 83% bien meilleure que K-Nearest Neighbor, qui donnait une précision de 66% au mieux, avec plus de temps que n'en prenait Naïve Bayes, pour les algorithmes traditionnels, alors que pour les algorithmes d'optimisation bio inspiré, nous avons essayé d'appliquer une approche qui inclut l'algorithme Grey Wolf Optimization (GWO) combiné avec la classification K-NN, mais nous n'avons pas atteint de résultats logiques.

Mots Clés : Spam, Twitter, Algorithmes bio inspirés.

Abstract

Twitter is one of the largest social networking sites in the world, where the amount of information and messages exchanged between users who called "Tweet", increases ; Unfortunately, this has increased the spread of spammers who share unwanted messages, which often contain advertisements or programs or links that contain malicious sites, so these text messages can be detected and classified as spam or not, for this there is several algorithms and techniques used. This is what we tackled in our thesis, where we used a unified database and applied traditional, and other bio inspired techniques and algorithms to it to classify mail as spam or not, and track the results, we performed a comparison process where we concluded that the Naive Bayes algorithm gave 83% accuracy much better than K-Nearest Neighbor, which gave 66% accuracy at best, with more time than Naive took Bayes, for traditional algorithms, while for bio-inspired algorithms,

Keywords : Spam, Twitter, Bio Inspired Algorithms.

الملخص

يعتبر تويتر أحد أكبر مواقع التواصل الاجتماعي انتشارا حول العالم حيث تزداد فيه كمية المعلومات والرسائل المتبادلة بين المستخدمين والتي تسمى بـ »التغريدات«، هذا ولسوء الحظ ما زاد في انتشار مرسلي البريد العشوائي الذين يشاركون الرسائل غير المرغوب فيها والتي غالبا ما تحتوي على اعلانات أو برامج أو روابط تحتوي على مواقع ضارة، وبالتالي يمكن الكشف عن هذه الرسائل النصية وتصنيفها على أنها رسائل مرغوب فيها أو لا، وذلك باستعمال خوارزميات وتقنيات عديدة. هذا ما تناولناه في اطروحتنا هذه، حيث قمنا باستخدام قاعدة بيانات موحدة طبقنا عليها تقنيات وخوارزميات تقليدية ومستوحاة من الطبيعة المستخدمة لتصنيف البريد على أنه عشوائي أو لا، وعلى ضوء النتائج قمنا بعملية مقارنة، حيث أن خوارزمية Naïve Bayes أعطت دقة بلغت 83% أفضل بكثير من K-Nearest Neighbor التي أعطت دقة قدرها وصلت لـ 66% في أفضل الأحوال مع وقت أكثر من الذي استغرقته Naïve Bayes ، هذا فيما يخص الخوارزميات التقليدية، فيما يخص الخوارزميات المستوحاة من الحيوية

الكلمات الرئيسية : تويتر، الخوارزميات المستوحاة من الحيوية، البريد العشوائي.

Dédicace

*Tout d'abord je tiens à remercier **ALLAH** le tout puissant de m'avoir donné la force et la volonté et m'avoir permis d'arriver la*

Je dédie ce mémoire

A ma chère mère,

A mon cher père,

Pour l'amour qu'ils m'ont toujours donné, qui n'ont jamais cessé, de formuler des prières à mon égard, de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs.

A ma chère Epouse et mes chères enfants

Avec tous mes sentiments de respect, d'amour, de gratitude pour tous les sacrifices déployés pour assurer mon éducation dans les meilleures conditions

A ma sœur et frères.

A tous mes amis qui m'ont toujours encouragé, a tous ceux que j'aime.

Merci !

Mahfoud

Dédicace

*Tout d'abord je tiens à remercier **ALLAH** le tout puissant de m'avoir donné la force et la volonté et m'avoir permis d'arriver la*

Je dédie ce mémoire à mes parents

Pour l'amour qu'ils m'ont toujours donné, qui n'ont jamais cessé, de formuler des prières à mon égard, de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs.

***A mes frères, qui m'ont toujours encouragé, a tous ceux que j'aime**.*

A ma famille, mes proches et a tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plus de succès.

Merci !

Nacereddine

Reconnaissance

Tout d'abord, nous tenons à remercier Allah de nous avoir donné la force le courage et la patience de mener à bien ce projet

Mes vifs remerciements sont d'abord adressés à monsieur ***M. Bouhani Abdelkader****, Je tiens à lui exprimer ma gratitude et mon profond respect, a ses précieux conseils*

Nous tenons à remercier également les membres de jury : ***M. Nacera BRAHIM*** *et* ***M. Ahmed SAIDI*** *pour avoir accepté d'examiner ce travail.*

Je tiens également à remercier tous les professeurs de département de mathématiques et d'informatique.

Nous tenons également exprimer nos sincère remerciements aux professeurs qui nous ont enseigné et pour tout le soutien qu'ils nous ont apporté.

TABLE DES MATIÈRES

Table des figures

Liste des tableaux

Liste des Algorithmes

Liste des sigles et acronymes

NB	*Naïve Bayes Algorithm*
KNN	*K-Nearest Neighbor Algorithm*
GWO	*Grey Wolf Optimization Algorithm*
FOA	*Firefly Optimization Algorithm*
NLP	*Natural Language Processing*
BIC	*Bio Inspired Computing*
TT	*Trending Topics*
URL	*Uniform Resource Locator*

Introduction

Twitter est l'une des plate-formes de médias sociaux les plus populaires qui fournit un réseau social d'utilisateurs qui publient des messages jusqu'à 280 caractères [13] appelés «tweet», seuls du texte et des liens HTTP peuvent être inclus dans les tweets Ces échanges de tweets permettent aux amis/collègues de communiquer et de rester connectés [14], il est le site qui connaît une croissance la plus rapide parmi tous les sites de réseautage social, fournit une liste des sujets les plus discutés appelées "Trending Topics (TT)" pour permettre aux utilisateurs d'être au courant de la plupart sujets populaires sur Twitter [15].
En plus "Hashtag" un terme qui commence par le caractère "#" est couramment utilisé pour mentionner le sujet du tweet et permettre aux utilisateurs de suivre les sujets qui les intéressent [15].
Les utilisateurs de Twitter ont différents niveaux de sensibilisation en ce qui concerne les menaces de sécurité cachées dans les sites de réseaux sociaux. Par exemple, une étude précédente a montré que 45% des utilisateurs d'un site de réseau social clinquant facilement sur les liens publiés par n'importe quel ami même s'ils ne connaissent pas cette personne [14].
Les pirates utilisent une méthode pour atteindre leurs objectifs malveillants publier des messages alléchants comme « Je viens de voir cette photo de vous » suivis d'un lien URL, lorsque vous cliquez dessus, vous amène à un site qui télécharge des logiciels malveillants.[15] Malheureusement, ce phénomène croissant permet aux spammeurs de diffuser des tweets malveillants. twitter propose plusieurs méthodes permettant aux utilisateurs de signaler ces signalements qui sont examinés par Twitter, ou suspendus en cas de spam.
Ces méthodes ne sont pas très utiles pour les sujets tendance car le processus de suspension est lent, il est nécessaire de mettre en place des mécanismes de protection automatiques qui permettent de protéger les utilisateurs, plusieurs algorithmes ont été implémentés à cette fin.
Notre mémoire s'intéresse a définir quelques algorithmes qui filtrent le contenu des tweets(texte) indésirables (spams) dans twitter,parmi ces algorithmes traditionnels : Naïve Bayes (NB) Algorithme, K-Nearest Neighbor Algorithme (K-NN) et bio inspiré : Grey Wolf Optimization Algorithm (GWO), Firefly Optimization Algorithm (FOA), on a détaillé la structure de chaque approche et ensuite, on a modélisé deux approches une traditionnelle et une autre bio inspirée,en plus on a fait une comparaison entre les deux approches.

Nous avons organisé notre mémoire en quatre chapitres :
Chapitre 1 : Généralité, on a défini quelques concepts de base comme :Réseau social, Twitter, Spam.
Chapitre 2 : Algorithmes de détection des spams,on a analysé et discuté les quatre approches suivantes : Naïve Bayes (NB), (K-NN),Traditionnel, et (GWO), (FOA) Bio inspiré, pour (K-NN) on a utilisé trois distances avec elle : Enclidienne, Manhattanet, et Minkowski.
Chapitre 3 : État de l'art,on a étudié quelques exemples les plus importants approches traditionnelles et bio inspirés traitent le problème de détection des spam,avec la présentation et la discution des différents résultats obtenus.
Chapitre 4 : Expérimentation, ce chapitre comprend l'environnement de travail, la structure proposée pour détecter le spam dans Twitter avec une explication de ses étapes, puis la présentation et la discussion des résultats obtenus à l'aide des algorithmes proposés.

Chapitre 1

Généralité

1.1 Réseau social

1.1.1 Définition

Depuis le début des années 2000, la présence des réseaux sociaux, également appelés réseaux communautaires, dans le monde virtuel, un réseau social désignait principalement des personnes ayant une affinité ou un intérêt commun, permettant de regrouper diverses personnes afin de créer un échange sur un sujet, et de partager des informations, indépendamment de leur situation géographique chaque utilisateur doit créer un profil pour publier et consulter différents contenus : texte, photos, vidéos ; Il existe de nombreux médias sociaux WhatsApp, Facebook, Messenger, Instagram, Twitter. . . etc.[13]

1.1.2 Twitter (définition)

Est un outil qui permet à un utilisateur d'envoyer des messages simples gratuitement ces messages sont appelés tweets, on peut publier ses tweets à partir d'un smart phone ou un ordinateur portable...etc, par une application via internet, généralement limité à 280 caractères.[16]
Twitter Contient 1,3 milliard de comptes dans le monde, 321 millions d'utilisateurs actifs par mois, et 126 millions d'utilisateurs actifs par jour.[16]

1.2 Contenu indésirable (Spams)

1.2.1 Historique de mot spam

En va parler brièvement sur la création du mot spam :
• En 1937une société américaine Hormel Foods a lancé un concours,pour trouver un nouveau nom au produit « jambon épicé », Finalement, le mot a été choisi « SPicedhAM », ensuite a été abrégé au mot « SPAM ».[14]
• En 1985, les participants aux (rares) systèmes de chat commencent à utiliser le terme « SPAM » pour définir les messages nuisibles.[14]
• Le premier spam à l'échelle mondiale envoyé en janvier 1994 sur tous les groupes de discussion Usenet.[14]

1.2.2 Définition de spam

Le spam est tout type de message électronique non sollicité, généralement envoyé à des fins « publicitaires, commerciales ou malveillantes ou pour faciliter la fraude » sont envoyés sur les réseaux sociaux, les e-mails, et même les commentaires ou les "J'aime" sur les publications, y compris les SMS,étant donné que l'envoi des messages spam actuellement ne demande pas beaucoup d'effort en taille et en contenu qui touche plusieurs domaine d'une manière d'effarante (Figure 1.1).[17]

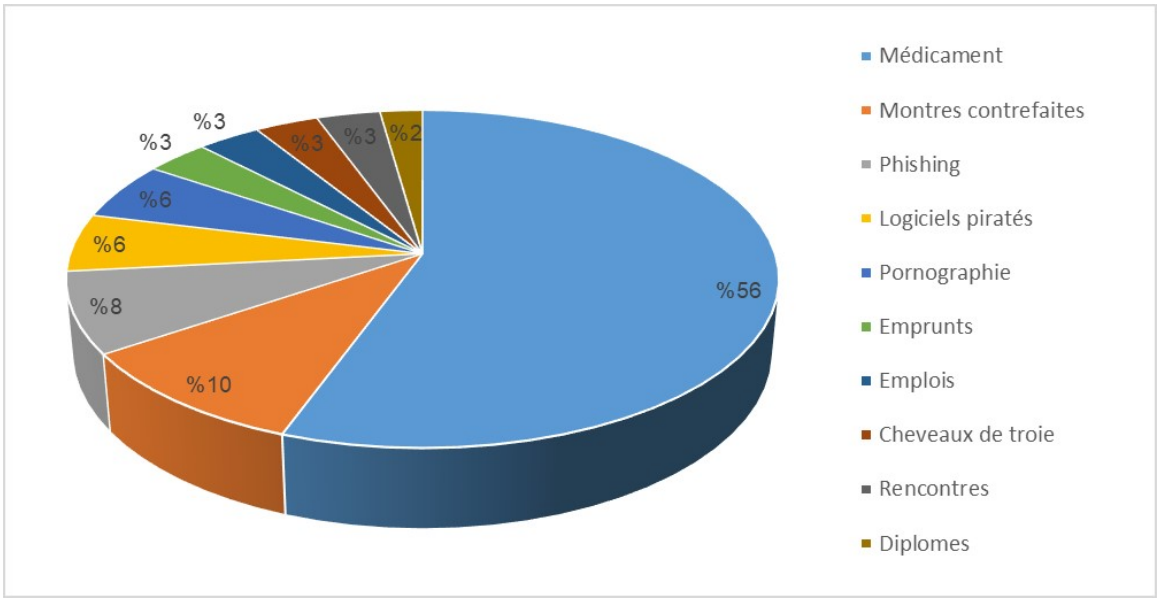

FIGURE 1.1 – Répartition des spam par contenu.[1]

1.2.3 But du spam

Dans twitter les spammeurs utilisent de nombreuses méthodes pour atteindre leurs objectifs, parmi eux :(1) la publication des virus et des malwares, (2) la diffusion des publicités pour générer des ventes et des profits illégaux, (3)création de sites Web de phishing pour révéler des informations sensibles, (4) accéder aux données (5) la diffusion du matériel pornographique.[18]

1.3 Dangers de contenu non sollicité dans twitter

Dans les média sociaux et particulièrement dans twitter le contenu indésirable crée une nuisance majeure telle que : (1) dégrader l'exactitude des statistiques obtenues,(2) polluer les résultats de recherche,(3) violer la vie privée des utilisateurs,par exemple,capables de faire des captures d'écran, de prendre le contrôle d'un ordinateur, et (4) consommer les ressources de stockage,c'est parce le nombre important des spams prend un espace sur les ressources de stockage .[18]

Chapitre 2

Algorithmes de détection des spams

L'augmentation des utilisateurs en médias sociaux est une opportunité pour les spammeurs de diffuser leurs messages indésirables, afin d'attirer l'attention des utilisateurs, c'est pour cette raison les développeurs sont poussés à penser de développer des algorithmes pour limiter la propagation de ces types de message.
les Algorithmes de détection des spam sur twitter peuvent être classer en trois catégories
(1) Algorithme basées sur les tweets.
(2) Algorithme basées sur le compte.
(3) Algorithme basées sur la relation entre l'expéditeur et le destinataire du tweet.
la détection des spam basées sur les tweets utilise les caractéristiques comme cité dans le (tableau 2.1) ces caractéristiques seront utilises par ces Algorithmes a fin de filtrer les spams.[9]

Feature	Description	Is User-controlled ?
Sender	The sender of the tweet	Yes
Mentions	The mention(s) used in the tweet	Yes
Hashtags	The hashtag(s) used in the tweet	Yes
Link	The link used in the tweet	Yes
Number of likes	The number of likes the tweet has	No
Number of retweets	The number of retweets the tweet has	No
Number of replies	The number of replies the tweet has received	No
Sent date	The date tweet is sent	Yes

TABLE 2.1 – Features de détection de spam basses sur Tweet.[9]

Nous avons étudié ces algorithmes de formes de deux angles différents, le premier sur les approches traditionnelles et le deuxième sur les approches bio inspirés .

2.1 Algorithmes traditionnels

Il existe plusieurs approches traditionnelles qui permettent de détecter les spams, on va analyser et discuter les deux suivants : Naïve Bayesian classifiers, et K-Nearest Neighbor classifiers (K-NN).

2.1.1 Naïve Bayesian classifiers (NB)

La classification naïve bayésienne s'apparente à une classification bayé-sienne probabiliste simple (dite naïve) ; Populaire en Machine Learning, elle repose sur le théorème de Bayes, il est particulièrement utile pour les problématiques de classification de texte.

- **Avantages de Naïve Bayésien classifier**

Ce type de classification simple permet à un modèle de machine Learning d'apprendre rapidement,en plus son exécution est très rapide, il n'est pas nécessaire de fournir un gros volume de données lors de la phase d'apprentissage, comparativement à d'autres méthodes de machine Learning autrement plus complexes, La classification naïve bayésienne offre des résultats très efficaces dans des domaines variés, par exemple la création de filtres anti spam, la classification de documents (par catégories) ou encore les moteurs d'indexation et de recherche.[19]

- **Théorème de Bayes**

Notons P la probabilité d'un événement le théorème de Bayes fournit un moyen de calculer la probabilité a posteriori comme suite : P(A|B) à partir de P(A),P(B) et P(B|A).
Voir l'équation (2.1) ci-dessous :

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \tag{2.1}$$

Le terme P(A|B) se lit : la probabilité que l'événement A se réalise sachant que l'événement B s'est déjà réalisé.
- P(A) est la probabilité a priori de la classe antérieure.
- P(B|A) est la probabilité du prédicateur pour une classe donnée.
- P(B) est la probabilité a priori du prédicateur.[1]

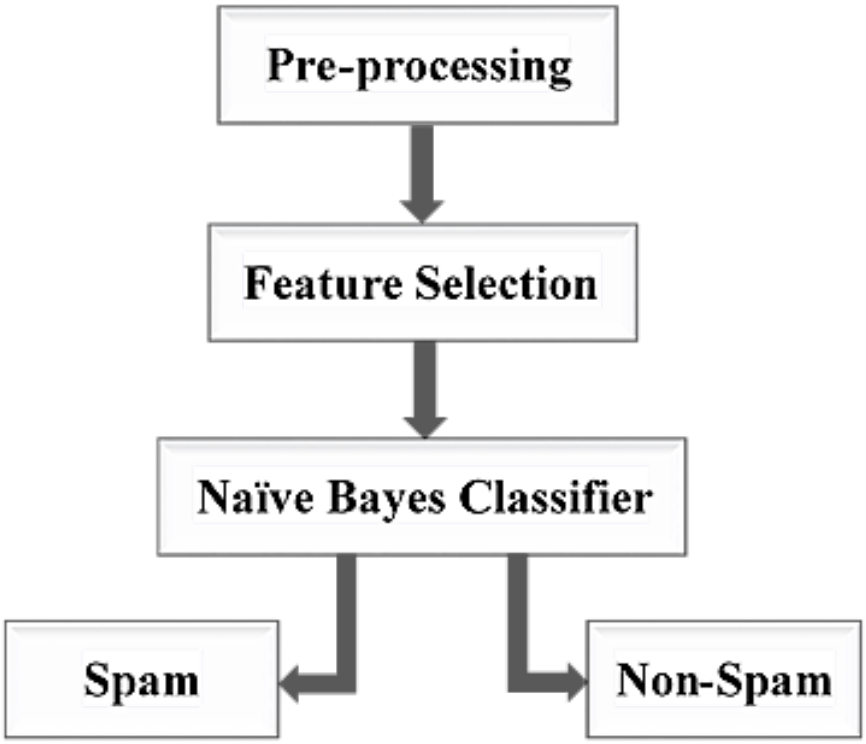

FIGURE 2.1 – Architecture de Naïve Bayes Classifier [2]

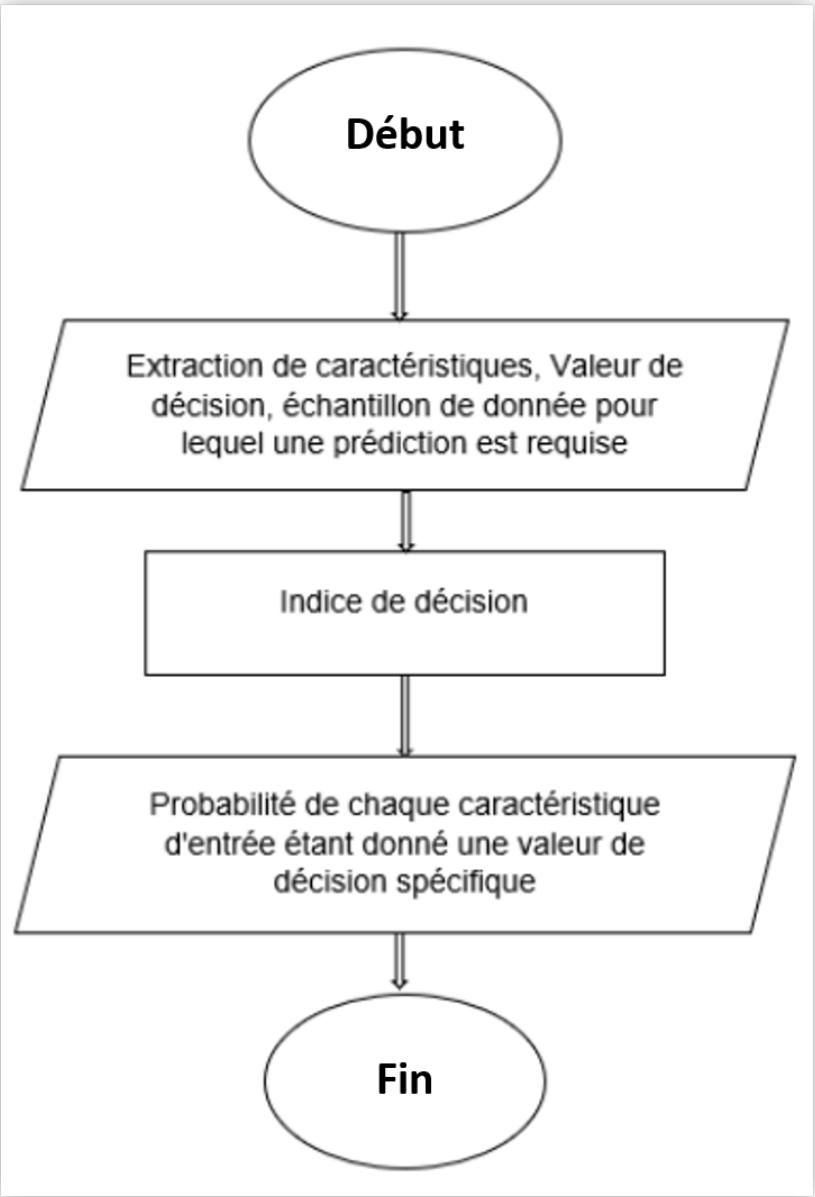

FIGURE 2.2 – Organigramme de l'algorithme Naïve Bayes [3]

Comment fonctionne vraiment Naïve Bayes :
Pour clarifier les choses on va l'expliquer à l'aide d'un exemple :
Supposons qu'on a un état d'E-mails on va les classer comme spam ou non spam.
Une analyse a été effectuée à l'ensemble des données contient 15 e-mails non spam et 10 e-mails spam, la fréquence de chaque mot enregistré comme indiqué au tableau ci-dessous :

D'abord une opération de prétraitement a été effectuée par l'application Natural Language Processing (NLP) pour éliminer les mots vides car ils n'ont pas de signification importante, La même chose s'applique aux nombres et aux ponctuations, les résultats sont dans (Tableau 2.2).

	Non Spam	Spam
Dear	8	3
Visit	2	6
Invitations	5	2
Link	2	7
Friend	6	1
Hello	5	4
Discount	0	8
Money	1	7
Click	2	9
Dinner	3	0
Total Words	34	47

TABLE 2.2 – Naïve Bayes Classifier Exemple.[12]

Quelques probabilités :

- P(Dear|Not Spam) = 8/34.
- P(Visit|Not Spam) = 2/34.
- P(Dear|Spam) = 3/47.
- P(Visit|Spam) = 6/47.[12]

on va utiliser le théorème de Bayes pour voir s'il s'agit d'un spam ou non le message "Hellofriend" Donc

$$\mathbf{P(Not\ Spam \mid Hello\ Friend) = \frac{P(Hello\ Friend \mid Not\ Spam) * P(Not\ Spam)}{P(Hello\ Friend)}}$$

P(Hello friend | Not Spam) = 0, car ce cas (Hello friend) n'existe pas dans notre donnée, c'est-à-dire qu'on va traiter des mots simples, et de même pour **P(Hello friend | Spam)** sera également égal à zéro, ce qui fait que les deux probabilités d'être un spam et non un spam seront nulles, ce qui n'a aucun sens !! [12]

Sans tenir compte du dénominateur :

$$\mathbf{P(Not\ Spam \mid Hello\ friend) = P(Hello\ friend \mid Not\ Spam) * P(Not\ Spam)}$$

Du moment que nous avons dit que le Naïve Bayes suppose que "les caractéristiques que nous utilisons pour prédire la cible sont indépendantes".

Donc :

$$\mathbf{P(Hello\ Friend \mid Not\ Spam) = P(Hello \mid Not\ Spam) * P(Friend \mid Not\ Spam)}$$

$$\mathbf{P(Not\ Spam \mid Hello\ Friend) = P(Hello \mid Not\ Spam) * P(Friend \mid Not\ Spam) * P(Not\ Spam)}$$

$$\mathbf{P(Not\ Spam \mid Hello\ Friend) = \frac{5}{34} * \frac{6}{34} * \frac{15}{25} = 0.0155}$$

La même procédure pour calculer la probabilité d'être un spam en utilisant :

$$\mathbf{P(Spam \mid Hello\ Friend) = \frac{4}{47} * \frac{1}{47} * \frac{10}{25} = 0.00072}$$

P(Not Spam| Hello friend)=0.0155 > P(Spam| Hello friend) ; Alors, le message
" Hello friend " n'est pas un spam.[12]

2.1.2 K-Nearest Neighbor classifiers (K-NN)

K plus proche voisin (K-NN) est une méthode de classification puissante qui permet de classer une instance inconnue en utilisant un ensemble d'instances classés.
Le but est de classer une nouvelle instance selon les classes des instances de la base d'apprentissage,

cette classification est basée sur le principe de voisinage afin de détecter la classe la plus fréquente parmi les k voisins de l'instance inconnue.
Le cas le plus simple de cet algorithme est lorsque k est égale à 1. (1-NN) est basée sur la classe du voisin le plus proche afin de classer l'instance inconnue.[20]
La classification k-NN comporte deux étapes, la première est de déterminer les voisins les plus proches et la seconde est de déterminer la classe, à l'aide de ces voisins.
Lorsqu'on parle de voisin cela implique la notion de distance, c'est à dire on va calculer la distance bien que théoriquement un nombre infini de mesures peuvent exister en faisant varier l'ordre de l'équation.[20]

Seulement trois mesures ont bénéficié de beaucoup d'attention, la plus souvent utilisée est la distance euclidienne définie par l'équation (2.2) [21] :

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.2}$$

la distance Manhattan (equations 2.3) [21] :

$$d(x,y) = \sum_{i=1}^{n}|x_i - y_i| \tag{2.3}$$

la distance de Minkowski (equations 2.4) [21] :

$$d(x,y) = \left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{1/p} \tag{2.4}$$

La distance de Minkowski est une généralisation des distances de Manhattan et euclidienne qui ajoute un paramètre p appelé ordre.
Lorsque p est égal à un, la distance de Minkowski est égale à la distance de Manhattan ,et lorsque p est égal à deux, la distance de Minkowski est égale à la distance de euclidienne, lorsque p est infini, la distance de Minkowski est égale à la distance de Chebyshev (equations 2.5). [21] :

$$d(x,y) = \max_i(|x_i - y_i|) \tag{2.5}$$

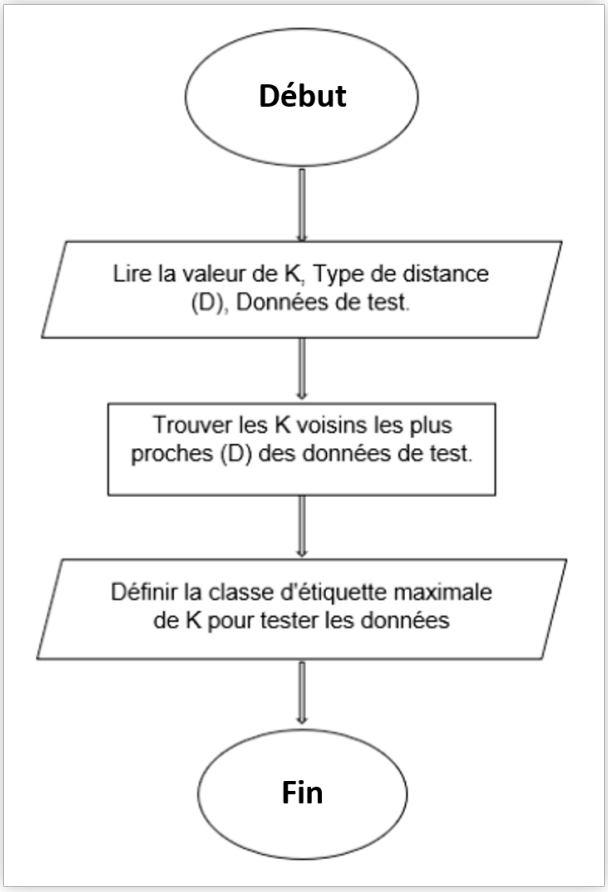

FIGURE 2.3 – Organigramme de l'algorithme K-NN [3].

Un exemple de classification K-NN est illustré par la figure 1 où le point inconnu (cercle) appartient soit à la première classe (carré) ou à la deuxième classe (triangle).
Si K = 3, le point inconnu est classé en deuxième classe parce qu'il y a deux triangles et un seul carré parmi les trois plus proches exemples à l'intérieur du cercle, illustrés dans la (Figure 2.4).
Si K = 5, il est classé dans la première classe, illustrée dans la (Figure) Exemple de classification K-NN (K=3, et K=5).[20]

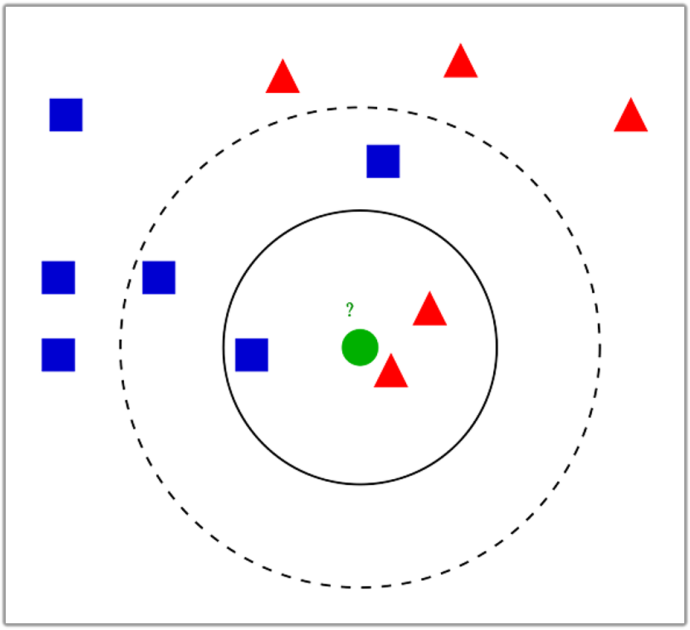

FIGURE 2.4 – Exemple de classification K-NN (K=3, et K=5)) [4].

- K-NN résumées par les étapes indiquées dans (algorithme 1)

Algorithme 1 : K-Nearest Neighbor Algorithm

```
for all the unknown samples UnSample(i)
   for all the known samples Sample(j)
      compute the distance between UnSamples(i) and Sample(j)
   end for
   find the k smallest distances
   locate the corresponding samples Sample(j1),..,Sample(jk)
   assign UnSample(i) to the class which appears more frequently
end for
```

2.2 Algorithmes bio-inspirées

2.2.1 Introduction

Les chercheurs depuis pas mal de temps sont tournés vers la nature et la biologie pour comprendre et modéliser des solutions à des problèmes complexes.
Les algorithmes bio-inspirés, généralement sont des algorithmes d'optimisation utilisent des nouvelles techniques inspirées de la nature pour développer des solutions.
Il existe plusieurs types de classification des algorithmes basés sur la source d'inspiration biologique, sont illustrés dans la (Figure 2.5)

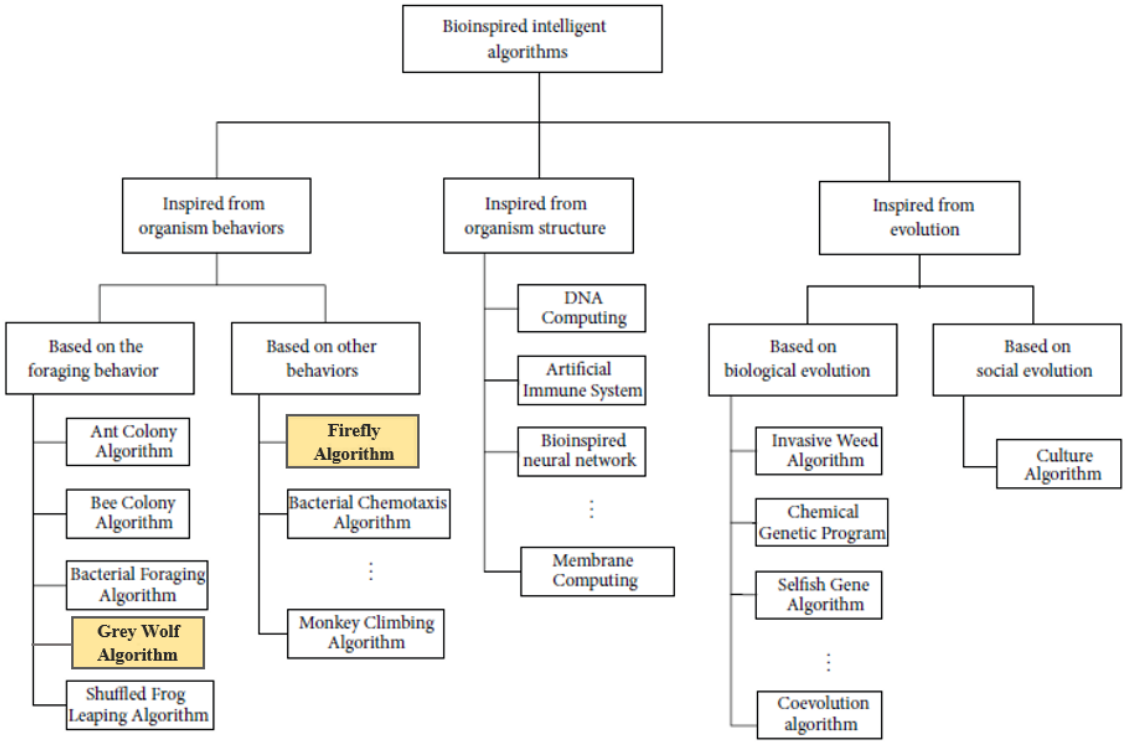

FIGURE 2.5 – La classification des bio-inspirée algorithmes à partir du mécanisme biomimétique[1].[5]

[1] Discipline scientifique qui consiste à étudier les différents processus qui permettent à la Nature de survivre et de se développer pour les appliquer à l'homme.https://www.linternaute.fr/dictionnaire/fr/definition/biomimetique/

En a choisir deux algorithmes d'optimisation bio-inspirés à définir :

2.2.2 Grey Wolf Optimization Algorithm (GWO)

L'algorithme d'optimisation du loup Gris (GWO), est un algorithme d'optimisation qui utilise une approche inspirée de la nature prédatrice, en 2014 Mirjalili et son groupe, présente une stratégie d'optimisation évolutive méta-heuristique[1], appelée le mécanisme de chasse des loups gris dans la nature, sont considérés comme des chasseurs au groupe comprise entre 5 et 12 loups, l'algorithme GWO dépend de la structure en fonction de leur ordre hiérarchique et leur leadership pour créer une technique de chasse.[6]

il considère quatre types de loups gris nommés alpha (α), bêta (β), delta (δ) et oméga (ω) qui sont reconnus pour comprendre leur hiérarchie ; Comme le montre le schéma pyramidal(figure 2.6).
- Niveau alpha (α) composé d'un seul loup leader.
- Niveau bêta (β) composé d'un seul loup.
- Niveau delta (δ) composé de plusieurs loups.
- Niveau oméga (ω) composé d'un seul loup.[6]

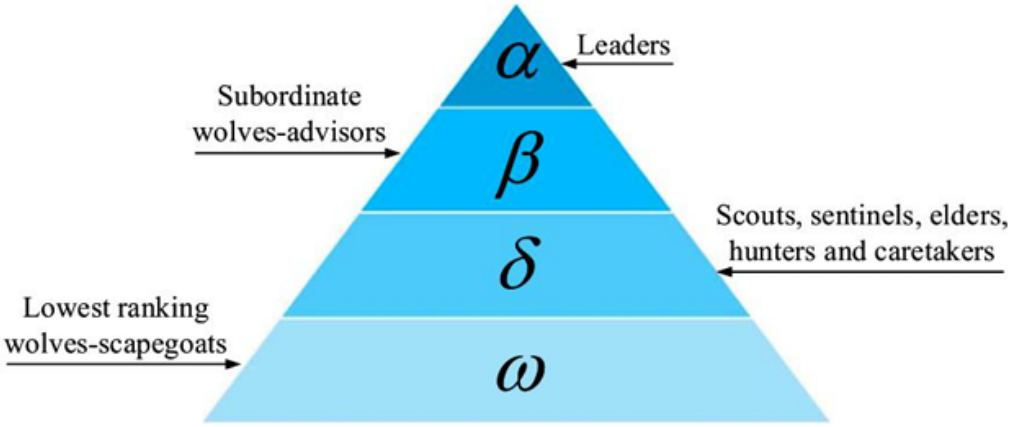

FIGURE 2.6 – Pyramide hiérarchie familiale de loups gris. [6].

La chasse en groupe est également l'un des comportements sociaux intéressants des loups gris.

La chasse des loups gris comprend les trois parties principales suivantes :
(1) Suivre, et approcher de la proie.
(2) Poursuivre, et encercler la proie jusqu'à ce qu'elle ne bouge pas.
(3) Attaquer la proie.

[1]un algorithme d'optimisation visant à résoudre des problèmes d'optimisation difficile.

GWO résumées par les étapes indiquées dans (algorithme 2)

Algorithme 2 : Wolf Optimization Algorithm

Initialize the grey wolf population

$$X_i, (i = 1, 2, ..., n)$$

Initialize a, A, and C
Calculate the fitness of each search agent, as: X_α =the best search agent, X_β =the second best search agent, X_δ =the third best search agent
while $t<Maxnumberofiterations$ **do**
  for <each agent> **do**
    Update the position of the current search agent
  end for
  Update a, A, and C
  Calculate the fitness of all search agents
  Update X_α, X_β, and X_δ
  $t = t + 1$
end while
return X_α

- Le pionnier est appelé alpha (α) est capable de faire le choix de chasse, lieu de couchage...etc, le second est nommé bêta (β) assiste l'alpha dans la prise de décision.[7]
- Le loup bêta (β) doit respecter α le loup gris le plus bas du point de vue du rang est oméga (ω) il soumet les données à tous les autres loups dominants.[7]
- Le reste des loups gris a nommé delta (δ) ce sont les prédominants.[7]

L'algorithme GWO utilise une population de loups gris dans laquelle chaque loup joue le rôle d'agent de recherche, il calcule l'aptitude pour chacun de ces agents de recherche à l'aide d'une fonction prédéfinie.[7]

Définition 1. Distance entre le loup gris et la proie qui est déterminé par l'équation (2.6) :

$$\overrightarrow{D} = |\overrightarrow{C}.\overrightarrow{X}_p(t) - \overrightarrow{X}(t)| \tag{2.6}$$

- D utilisé pour spécifier une nouvelle position du loup gris.
- **(t)** Nombre d'itérations.
- X_p Vecteur de positions des proies.
- X Vecteur des positions du loup gris.[7]

- C Coefficients de vecteurs qui est déterminé par l'équation (2.7) :

$$\overrightarrow{C} = 2.\overrightarrow{r}_1(2) \tag{2.7}$$

- r_1 est un vecteur aléatoire dans l'intervalle de [0.1].[7]

Définition 2. Reconnaissance de l'emplacement des proies par les équation (2.8) et (2.9) :

$$\overrightarrow{X}(t+1) = \overrightarrow{X}_p(t) - \overrightarrow{A}.\overrightarrow{D} \tag{2.8}$$

$$\overrightarrow{A} = 2.\overrightarrow{d}.\overrightarrow{r}_2 - \overrightarrow{d}(4) \tag{2.9}$$

- A Coefficients de vecteurs.
- r_2 Est un vecteur aléatoire dans l'intervalle de [0.1].

Les composantes de d diminué linéairement de 2 à 0 au cours des itérations.

Définition 3. Mise à jour de la position du loup gris

La distance entre chaque agent de recherche et α, β, δ peut être mesurée par les équation (2.10) et (2.11) et (2.12) :

$$\overrightarrow{D}_\alpha = |\overrightarrow{C}_1.\overrightarrow{X}_\alpha - \overrightarrow{X}| \tag{2.10}$$

$$\overrightarrow{D}_\beta = |\overrightarrow{C}_2.\overrightarrow{X}_\beta - \overrightarrow{X}| \tag{2.11}$$

$$\overrightarrow{D}_\delta = |\overrightarrow{C}_3.\overrightarrow{X}_\delta - \overrightarrow{X}| \tag{2.12}$$

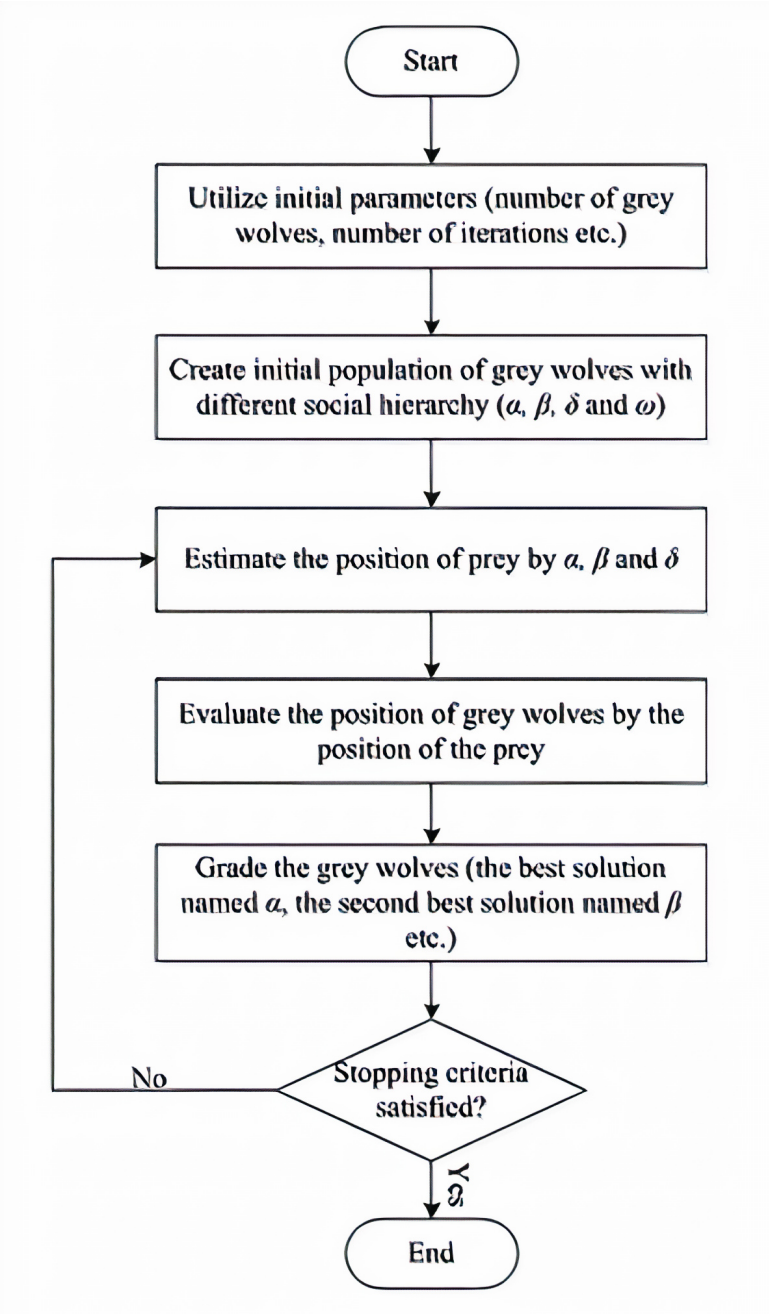

FIGURE 2.7 – L'organigramme de l'algorithme GWO [7].

Au cours de plusieurs interactions, il redéfinit les positions du loup pour chasser sa proie ; L'agent de recherche avec la meilleure valeur de fitness résultante est appelée (α), tandis que la deuxième meilleure valeur de fitness est appelée (β) et la troisième meilleure valeur de fitness est appelée (δ).[7]

2.2.3 Firefly Optimization Algorithm (FOA)

FOA est une technique Bio Inspired Computing (BIC), utilisée pour trouver des résultats approximatifs. (Johari et al., 2013), Présente une stratégie d'optimisation s'inspire de clignotante des lucioles, le but principal d'une luciole est d'attirer d'autres lucioles, et pour accomplir cette tâche, le clignotement de leur queue se comporte comme un système de signal.[11]
Les lucioles se déplacent en fonction de la luminosité de leurs voisins, où les lucioles brillantes sont plus attirantes.
Par conséquent, le mouvement de toute luciole dépend de celle de ceux qui l'entourent et le coefficient d'attractivité lumineuse d'une luciole η est donné par l'équation (2.13) suivante :

$$\eta = \eta_0 e^{(-\gamma d^2)} \tag{2.13}$$

Où d est la distance entre deux lucioles.[11]
γ est un coefficient variable tel que $\gamma \to 0$ et η_0 est l'attractivité lumineuse d'une luciole à d = 0.
La quantité de mouvement de luciole a vers une luciole b la plus brillante, est calculée à l'aide de (equation 2.14) :

$$X_{ab} = X_b + \eta_0 e^{(-\gamma d^2)}(X_b - X_a) + \lambda \epsilon_a \tag{2.14}$$

Les équations définissent le mouvement basé sur trois parties :

La première partie (X_b) indique l'emplacement actuel de la luciole b.
La deuxième partie $(\eta_0 e^{(-\gamma a b^2)}(X_b - X_a))$ reflète le mouvement de l'attraction entre la luciole a et b.
La troisième partie $(\lambda \epsilon_a)$ est ajoutée pour la randomisation à l'aide du vecteur de variables aléatoires ϵ_a qui peut être tiré de différentes répartitions, dans la troisième partie, lambda (λ) contrôle la taille du pas est appelée le paramètre de mise à l'échelle.[11]

Trois hypothèses doivent être remplies pour développer ce modèle de clignotante des lucioles.
• Toutes les lucioles sont considérées comme uni-sexuelles, chaque luciole se sent attirée par toutes les autres lucioles.[11]
• Pour deux lucioles, la luciole la moins brillante se déplacera dans la direction de la plus brillante en raison de l'attraction.[11]
• S'il n'y a pas de lucioles plus lumineuses qu'une luciole donnée une luciole se déplacera vers des positions aléatoires[11].

FIGURE 2.8 – Organigramme de l'algorithme FOA [8]

- FOA résumées par les étapes indiquées dans (algorithme 3)

Algorithme 3 : Firefly Optimization Algorithm

```
1  Randomly generate N fireflies (solutions) as an initial population
   {X_i | i = 1, 2, ..., N};
2  Calculate the fitness value of each firefly;
3  FEs = N;
4  while FEs ≤ MAX_FEs do
5      for i = 1 to N do
6          for j = 1 to N do
7              if f(X_j) < f(X_i) then
8                  Move firefly X_i towards X_j according to Eq. (2);
9                  Calculate the fitness value of the new solution;
10                 FEs++;
11             end
12         end
13     end
14 end
```

Chapitre 3

Etat de l'art

dans cette partie on va étudier quelques exemples les plus importents approches traditionnelles et bio inspirés traitent le problème de détection des spams dans l'eturatur qui à été proposé par les chercheurs ; et on va apercevoir les résultats

3.1 Approches twitter de détection de contenu non sollicité

Twitter a cherché à traquer les spammeurs et suspendre leurs comptes grâce à un mécanisme répertorié sur le site appelé (**Anti-Spam Twitter**) les utilisateurs peuvent signaler tout compte suspect en clinquant sur l'option **"Signaler : ils publient du spam"** sur Twitter (Figure 3.1).[18]
Il appartient aux responsables de Twitter de prendre la décision de suspendre ou non après avoir examiné manuellement le rapport de signalement, mais ce mécanisme nécessite un double effort des deux parties, l'utilisateur et l'administrateur, car la probabilité de découvrir un nouveau compte et de le suspendre dans les premiers jours ne dépasse pas 1%.[18]

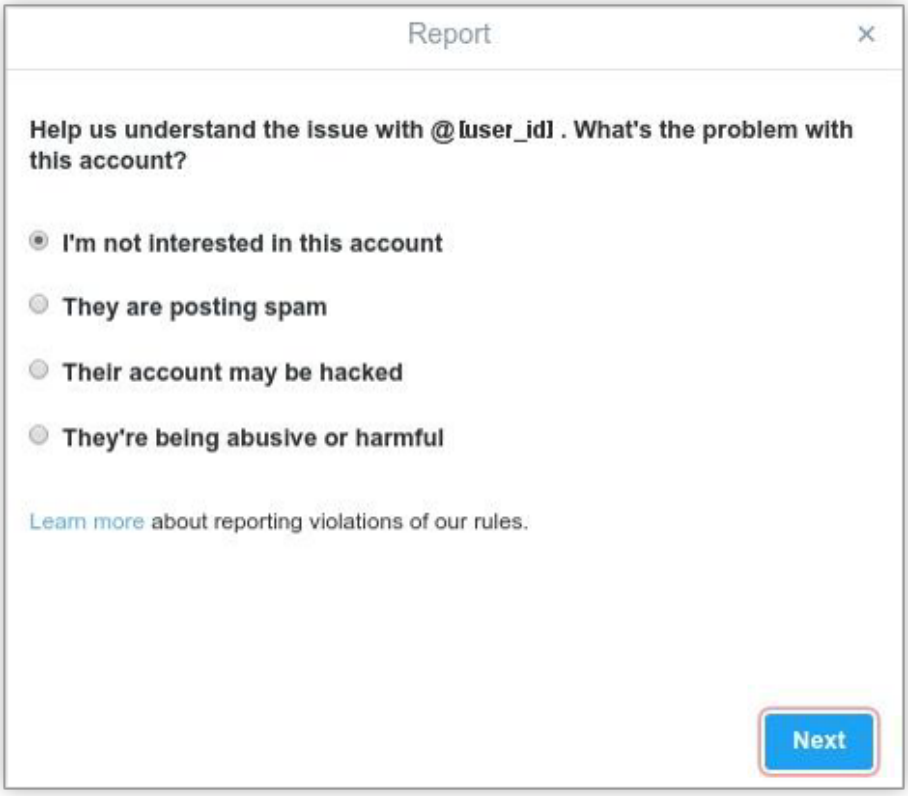

FIGURE 3.1 – L'interface utilisateur de Twitter qui est utilisée pour signaler un compte en sélectionnant la raison.[9]

Une statistique indique que 8% des 25 millions d'URL publiées sur Twitter sont considérées comme des adresses malveillantes incluses dans les listes noires de Twitter où lorsqu'un utilisateur publie l'une de ces adresses, son compte est suspendu, mais cette méthode est très lente à identifier de nouvelles menaces car plus de 90% des utilisateurs cliquent sur l'URL et entrent sur le site avant qu'il ne soit mis sur liste noire [18].

Ensuite, les chercheurs doivent trouver des méthodes avancées et plus efficaces qui s'appuient sur l'intelligence artificielle pour réduire les spams, car certains ont eu recours à des algorithmes de classification traditionnels inclus dans l'apprentissage automatique tels que Naïve Bayésien, K-Nearest Neighbor Classifier (K-NN), et d'autres.

3.2 Approches traditionnelles de détection de contenu non sollicité sur twitter

Juan Martinez et Al de [22] ont introduit une méthodologie pour détecter les spams Twitter concernant le contenu de 20 millions tweets Comprend 34 000 sujets populaires.
Ils ont analysé le langage utilisé dans ces tweets et suggéré des fonctionnalités qui seraient difficiles à manipuler pour les spammeurs, afin d'identifier en utilisant le programme Weka [23], qui contient un ensemble complet d'algorithmes d'apprentissage automatique, où ils ont choisi des algorithmes de classification traditionnels (Naïve Bayes et 5 autre), ils ont détecté entre 89,3% et 93,7% des tweets de spam et 6,3% étaient mal catégorisés, et ils ont également obtenu des meilleurs résultats lors d'un deuxième test d'évaluation avec un score de 94,5% et un taux de faux positifs de seulement 5,4%.

En 2017, Aryo Pinandito et Al tentent de détecter et de réduire le spam sur Twitter en s'appuyant sur deux algorithmes traditionnels, le premier est Naïve Bayes, un classificateur statistique basé sur le théorème de Bayes, a travers d'elle,Ils ont calculé la probabilité d'appartenir d'un document composé de plusieurs mots ($W_1 ... W_n$) à une certaine catégorie (C_i), puis choisissis la grande valeur V_{MAP} définie selon l'équation (3.1) suivante :

$$V_{MAP} = argmax P(C_i) \prod P(W_j|C_j) \tag{3.1}$$

Et ils ont également utilisé l'algorithme K-Nearest Neighbor Classifier K-NN, qui peut être combiné avec la méthode TF-IDF qui reflète la valeur d'un mot dans un document pour obtenir des résultats de notation plus élevés comme dans [24], où K-NN dépend du calcul de la distance de plusieurs manières, dont la plus célèbre est la distance **Euclidean** à travers laquelle ils ont classé les exemples en fonction de la catégorie de leurs voisins les plus proches, les résultats sont résumés dans (tableau 3.1) et (figure 3.2).

	Accuracy	**Training data set size**	**Processing Time (ms)**
Naïve Bayes (NB)	82 %	100 Exemple	140000 ms
K-Nearest Neighbor (K-NN)	71 %		160000 ms

TABLE 3.1 – Naïve Bayes et K-Nearest Neighbor Classification Accuracy results.[10]

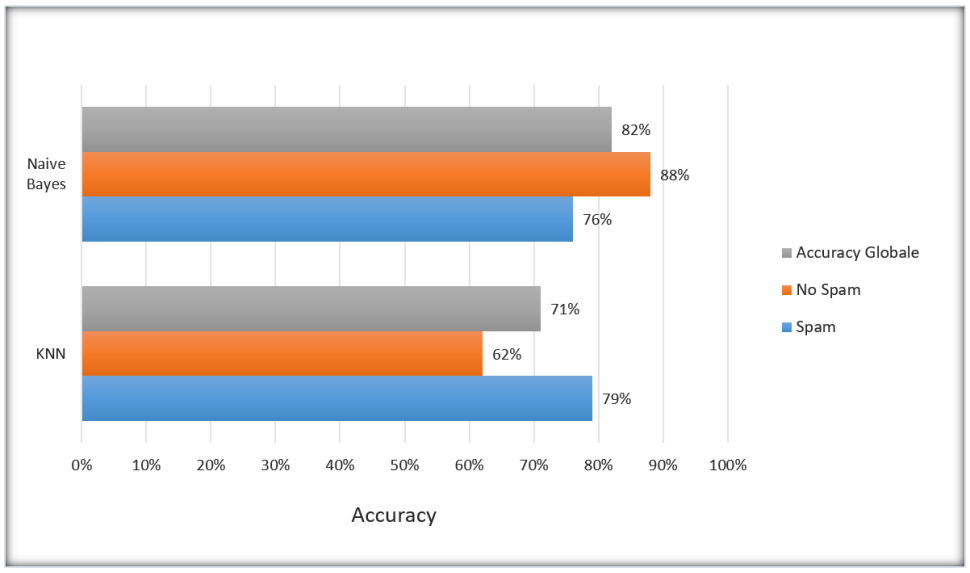

FIGURE 3.2 – Naïve Bayes et K-Nearest Neighbor Accuracy Comparaison.[10]

Ils ont également conclu que la classification Naïve Bayes pouvait obtenir une meilleure précision dans la détection du spam que la méthode de classification K-Nearest Neighbor ; Au contraire, la classification K-Nearest Neighbor utilisant la méthode de pondération TF-IDF a une meilleure précision dans la détection du spam que la classification Naïve Bayes.[10]

Certains chercheurs ont également proposé des algorithmes d'optimisation inspirés de la vie biologique des animaux, appelés (bio-inspired algorithms) pour détecter les spams tels que l'algorithme d'optimisation du loup Gris (GWO), Firefly Optimization Algorithm (FOA), et autres.

3.3 Approches bio-inspirées pour la détection de contenu non sollicité

Dans la lettrassions on n'a pas trouvé des travaux bio inspiré portant sur les problèmes des spam dans twitter, pour cela on a discuté des approches qui traitent le problème des spams dans les E-mails, car tous les deux ont la même structure de contenu (texte).

En 2021, Jai Batra et Al [11] ont cherché à détecté du spam dans les e-mails, proposant un mécanisme de classification basé sur l'algorithme K-Nearest Neighbor intégré à plusieurs techniques d'optimisation bio-inspirées, dont GWO et FOA, évaluant d'abord les performances de classification KNN avec trois métriques de distance : Euclidean, Manhattan et Chebyshev.
Ils ont ensuite comparé et évalué les performances d'algorithmes bio-inspirés intégrés à KNN sur la base de plusieurs métriques différentes telles que la précision, le temps d'exécution et d'autres critères, selon l'architecture (figure3.3) suivante :

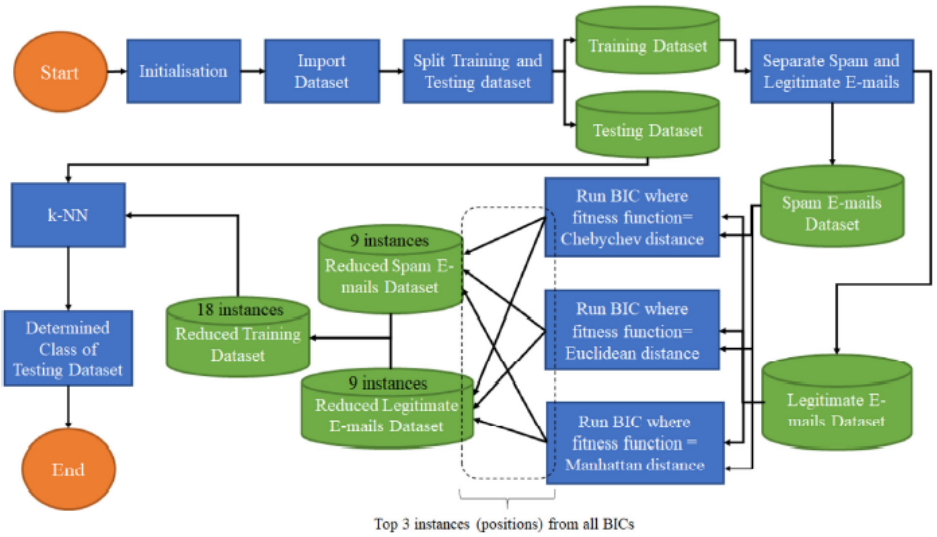

FIGURE 3.3 – Cadre du modèle proposé.[11]

Les résultats qu'ils ont reçus sont représentés dans le tableau et le graphique suivants (figure3.4).

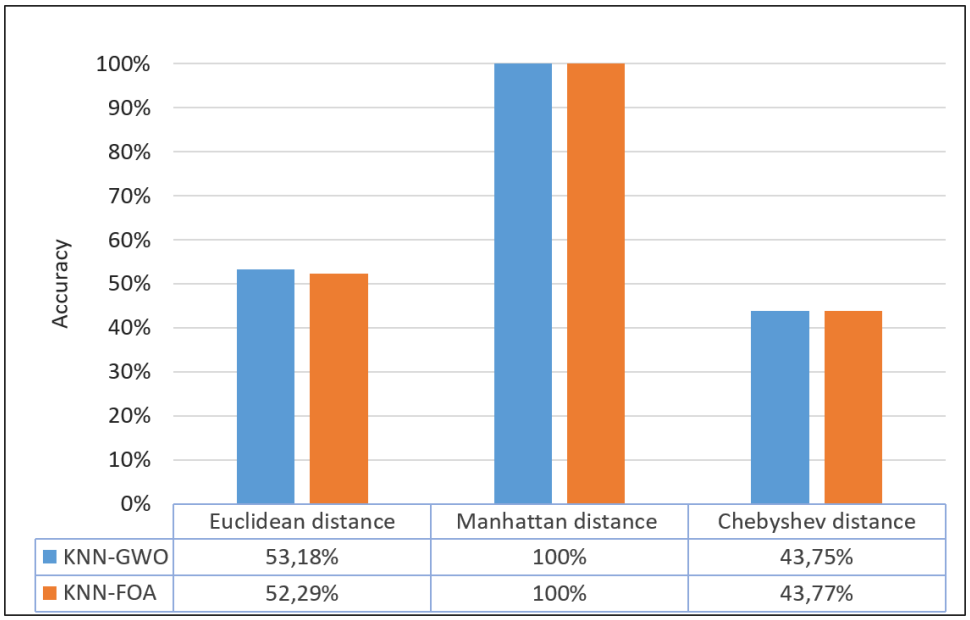

FIGURE 3.4 – Résultats de comparaison entre GWO et FOA avec les trois distances avec K=5.[11]

Chapitre 4

Expérimentation

4.1 Introduction

Dans ce chapitre, nous aborderons les mécanismes de détection des spams dans Twitter en classant les tweets comme spam ou non en utilisant plusieurs approche de classification, parmi eux les approches traditionnelles : Naïve Bayes Algorithme (NB), K-Nearest Neighbor Algorithme (K-NN), et bio-inspirées : Grey Wolf Optimization Algorithm (GWO) intégré à KNN, Firefly Optimization Algorithm (FOA) intégré à KNN.

4.2 Environnement

Nous avons développé notre modèle spécialisé dans la classification des tweets à l'aide de la plateforme **Google Colab)**, spécialisée dans, l'analyse des données, le développement des modèles d'apprentissage automatique, et l'affichage des résultats, à l'aide d'un ordinateur qu'il a les caractéristiques suivantes :

- RAM : 8 GB.
- Espace disque : 69 GB.
- CPU : Intel(R) Core(TM)i7-3632QM.
- CPU Freq : 2.20 GHz.
- Système d'exploitation : 64 Bits.
- GPU : AMD Radeon HD 8600 Series.
- Mémoire GPU : 6 GB.

Nous avons écrit notre code en utilisant : le langage de programmation python3 sous jupyter notebook, et en utilisant les librairies suivantes :

- **Scikit-learn**[1] **:** est une bibliothèque d'apprentissage automatique gratuite pour Python, il comporte divers algorithmes comme les Naïve bayes, Random forests, et k-neighbours, et il prend également en charge les bibliothèques numériques et scientifiques Python comme NumPy.

- **NumPy**[2] **:** acronyme de Numerical Python, est un package permettant d'effectuer efficacement des calculs scientifiques en Python, il a des capacités de généré des nombres aléatoires, des fonctions d'algèbre linéaire de base et encore plus.

- **NLTK**[3] **(Natural Language Toolkit) :** est une suite qui contient des bibliothèques et des programmes pour le traitement statistique d'une langue, c'est l'une des bibliothèques NLP les plus puissantes, qui contient des packages permettant aux machines de comprendre le langage humain et de répondre avec une réponse appropriée.

- **Matplotlib**[4] **:** est une bibliothèque de visualisation étonnante en Python pour les tracés 2D . et est une bibliothèque de visualisation des données multi-plateforme construite sur les matrice NumPy et conçue pour fonctionner avec la pile SciPy .

- **Pandas**[5] **:** est un outil d'analyse et de manipulation des données open source, rapide, puissant, flexible et facile à utiliser, construit pour le langage de programmation Python.

[1] Dataques, thttps ://www.dataquest.io/blog/sci-kit-learn-tutorial/
[2] Quantinsti, https ://blog.quantinsti.com/python-numpy-tutorial-installation-arrays-random-sampling/
[3] Guru99, https ://www.guru99.com/nltk-tutorial.html
[4] Geeksforgeeks, https ://www.geeksforgeeks.org/python-introduction-matplotlib/
[5] Pandas, https ://pandas.pydata.org/

4.3 Architecture

Notre mécanisme de détection des spams se décomposé en plusieurs étapes, on a proposé une architecte illustré dans la (figure 4.1) suivante :

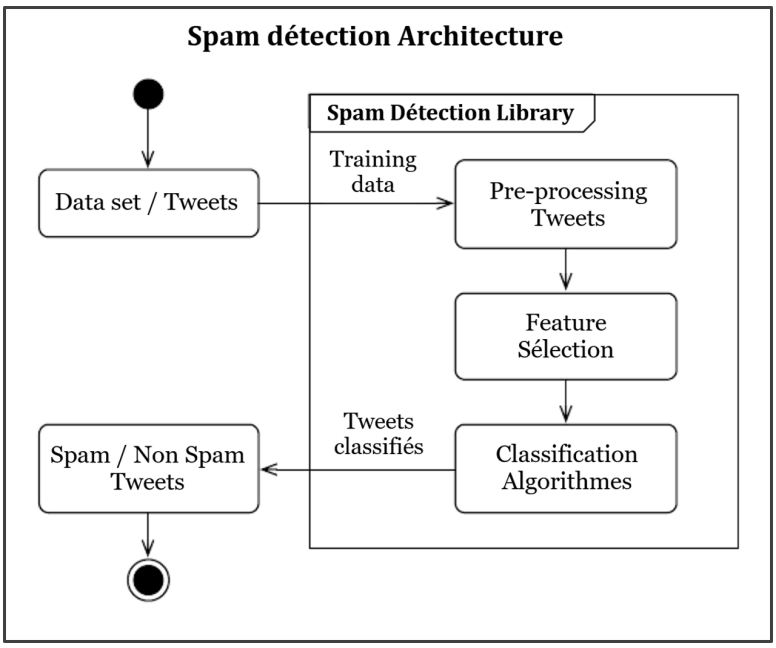

FIGURE 4.1 – Architecture de détection de spam.

4.3.1 Data set

Dans notre projet, nous avons utilisé la base de données (data set) se nommé (train), en tant que fichier CSV, qui se compose de 11 968 lignes (tweets) [25], dont 6153 sont des tweets légitimes à 51%, et 5815 sont des spams à 49%, et elle contient 8 colonnes disposées comme suit (tableau 4.1) :

Colonnes	Id	Tweet	following	followers	actions	is_retweet	location	Type
Type	integre	string	float	float	float	float	string	string

TABLE 4.1 – Data set informations.

- **Id :** est un entier qui représente le code du Tweet.

- **Tweet :** représente le contenu du Tweet ou ce qu'a été tweeté, le Tweet peut inclure des hashtags (#), des URLs (http) et des mentions (@).

Hashtag (#) : les tweets des spammeurs contiennent principalement des hashtags pour attirer l'attention des utilisateurs légitimes.
URL (http) : les tweets des spammeurs contiennent majoritairement des liens pour attirer l'attention des utilisateurs légitimes et les conduire à leurs fins malveillantes.
Signaux (@) : de nombreuses mentions dans un tweet indiquent que le tweet est un spam.

- **following :** le nombre de personnes suivies par le compte qui a tweeté.

- **followers :** le nombre de personnes suivant le compte qui ont tweeté.

Compte tenu du nombre de followers et de following, par rapport aux utilisateurs légitimes, les spammeurs sont plus susceptibles de suivre un très grand nombre de comptes légitimes afin d'attirer leur attention.
À l'inverse, les spammeurs sont suivis que par quelques utilisateurs légitimes, ce qui réduit la probabilité que leurs tweets reçois des likes et des réponses par rapport en d'utilisateurs légitimes.

- **actions :** Le nombre total de favoris, de réponses et de retweets dudit tweet.

- **is_retweet :** inclut deux valeurs,(0 ou 1) : si 0 ce n'est pas un retweet, si 1 c'est un retweet.

- **location :** indique l'emplacement de l'expéditeur du Tweet.

- **Type :** spécifie la nature du Tweet (Qualité ou Spam).

data_train.csv ×

1 to 10 of 7214 entries Filter

Id	Tweet	following	followers	actions	is_retweet	location	Type
10091	It's the everything else that's complicated. #PESummit #PXpic.twitter.com/Jsv6BAFQMl	0.0	11500.0		0.0	Chicago	Quality
10172	Eren sent a glare towards Mikasa then nodded and stood up to go help his lovely girlfriend @SincerePyrrhic. Once he arrived in the kitchen–	0.0	0.0		0.0		Quality
7012	I posted a new photo to Facebook http://fb.me/2Be7LiyuJ	0.0	0.0		0.0	Scotland, U.K	Quality
3697	#jan Idiot Chelsea Handler Diagnoses Trump With a Disease https://t.co/k8PrqcWTRI https://t.co/dRN35xtSJZ	3319.0	611.0	294.0	0.0	FBBIGBANG&2NE1TH	Spam
10740	Pedophile Anthony Weiner is TERRIFIED of Getting Beaten Up in Prison https://t.co/g3bU9Q4gAg	4840.0	1724.0	1522.0	0.0	www.instagram.com/fender	Spam
9572	EBMUD ending penalties for excessive water users https://t.co/D5a1FMVMHd	4435.0	16041.0	27750.0	0.0	Noida, NCR, India	Spam
10792	Big day. #WeTheNorth #yyz #thesix #sunset #skyline @ The Six https://www.instagram.com/p/BFgrA9gBZay/	0.0	0.0	0.0	0.0	Toronto, ON	Quality
11594	#UPA #scams to the tune of Rs 12 lakh Crore #Shame "Con"gress - Conning India since Independence! @INCIndia @IYCpic.twitter.com/0ZHSYhX5c2	0.0	193000.0		0.0	Mumbai	Quality

FIGURE 4.2 – Quelques exemples dans Data set.

4.3.2 Pre-processing

Ici, dans cette étape, les données brutes ont été organisées avant de construire le modèle formé, cette étape nous rend plus à l'aise avec les données, elle améliore également la qualité des données, voici donc les étapes du pré-traitement :

1. Tekonisation : dans cette étape, les données sensibles ont été échangées contre des données non sensibles. Ici, nous allons diviser le texte en morceaux, appelés jetons. Afin de l'utiliser pour supprimer les mots vides, le tweet a été tokenisé à l'aide de la fonction (word_tokenize) dans la bibliothèque NLTK.

2. Supprimez les mots vides et les symboles spéciaux du contenu des tweets : ici, dans cette étape, nous allons supprimer les mots vides pour la langue 'english' comme (are, the, is, am, then, a,..etc), en plus de supprimer les caractères spéciaux tels que (les signes de ponctuation, les URL, les espaces supplémentaires,..etc), à l'aide de la fonction (removeSpecialCharacter).

3. Lemmatisation : Dans cette étape, nous allons regrouper les différentes formes fléchies, elle vise normalement à supprimer uniquement les terminaisons flexionnelles et à retourner la forme de base ou de dictionnaire d'un mot, qui est connue sous le nom de lemme, ici, nous utilisons le vocabulaire et l'analyse morphologique des mots, ceci est fait en utilisant la fonction (WordNetLemmatizer) dans la bibliothèque NLTK.

Après ces trois étapes importantes, nous obtiendrons des tweets clairs préparés pour la formation et les tests, comme dans l'exemple suivant (tableau 4.2) :

Id	Tweet avant Pré-processing
3697	#jan Idiot Chelsea Handler Diagnoses Trump With a Disease https://t.co/k8PrqcWTRI https://t.co/dRN35xtSJZ
	Tweet après Pré-processing
	jan idiot chelsea handler diagnosis trump disease

Table 4.2 – Exemple de Pré-processing.

Après le pré-traitement, nous avons essayé de compter les mots qui composent les tweets de type (spam) et de type (Qualité) et leur fréquence, les résultats sont présentés dans le graphique suivant (figure 4.3) et (figure 4.4) :

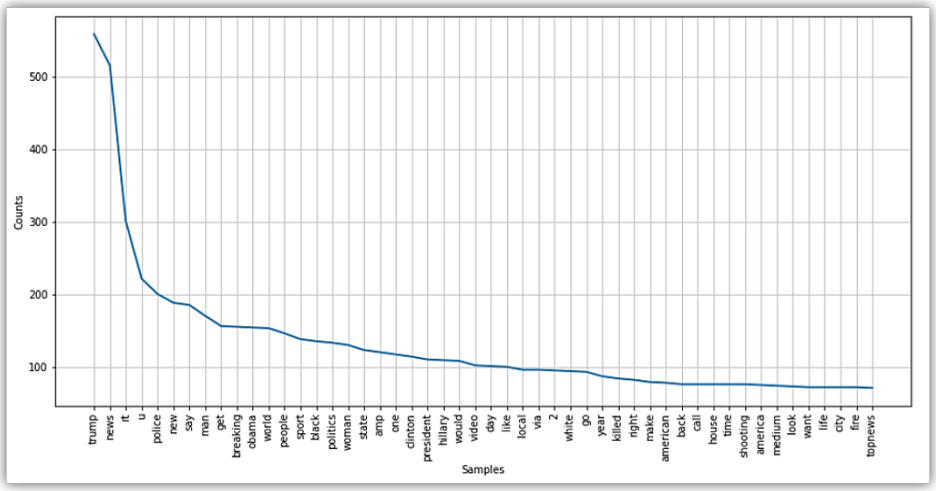

FIGURE 4.3 – 50 mots les plus couramment utilisés dans les tweets SPAM.

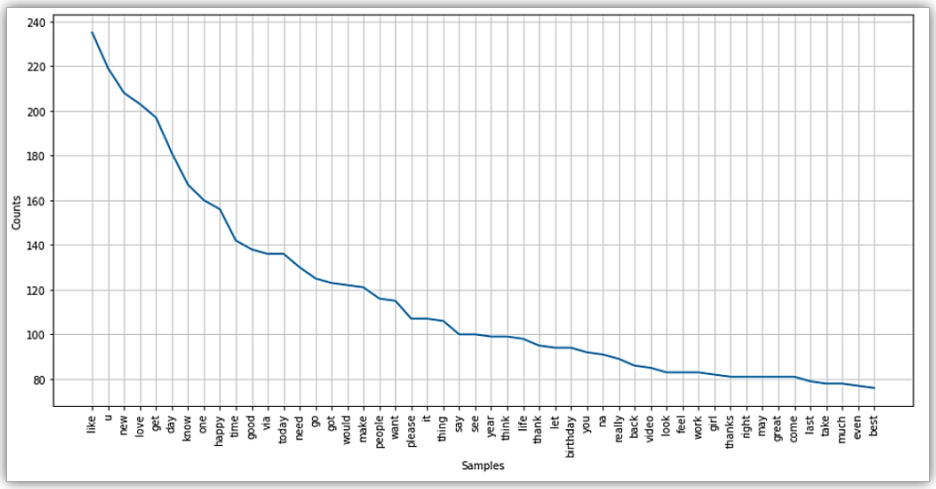

FIGURE 4.4 – 50 mots les plus couramment utilisés dans les tweets QUALITY.

4.3.3 Feature sélection

Les données brutes incluent des données redondantes, et comme notre objectif est de construire un bon classificateur, nous devons obtenir un ensemble complet et significatif des fonctionnalités nécessaires, l'extraction de fonctionnalités vise à créer et déduire de nouvelles fonctionnalités à partir de celles existantes.

Nous allons utiliser la fonction (CountVectorizer) incluse dans la bibliothèque Sklearn avec des paramètres (min_df = 0.02, pour éviter les termes qui ont une fréquence de document strictement inférieure au seuil min donné) pour créer un sac de mots.
Les feature résultantes (après application de la fonction CountVectorizer aux tweets traités) sont : {'new', 'trump', 'day', 'rt', 'people', 'news', 'like', 'get', 'one', 'say'}.

4.3.4 Classification algorithmes

Étant donné que les classes de sortie du classificateur sont 1 ou 0, la qualité a reçu une valeur de 1 et le spam a reçu une valeur de 0 ; Ensuite, l'ensemble des données a été divisé en 80% (9574 tweets) d'entraînement et 20% (2394 tweets) de tests.
Les données sont maintenant prêtes à être transmises aux classificateurs suivant :

Traditionnels :
- Naïve Bayesian classifiers (NB).
- K-Nearest Neighbor classifiers (K-NN).

Bio-inspirées :
- Grey Wolf Optimization Algorithm (GWO).
- Firefly Optimization Algorithm (FOA).

4.4 Implémentions d'une approche traditionnelle

4.4.1 Naïve Bayesian classifiers (NB)

Pour s'entraîner et testé notre données à l'aide de l'algorithme Naïve Bayes, nous avons utilisé la fonction GaussianNB (Gaussian Naïve Bayes) incluse dans la bibliothèque Sklearn.

- **GaussianNB**[1] **:** un algorithme Gaussian Naïve Bayes est un type spécial d'algorithme Naïve Bayes, il est spécifiquement utilisé lorsque les caractéristiques ont des valeurs continues. Il est également supposé que toutes les caractéristiques suivent une distribution gaussienne, c'est-à-dire une distribution normale.

[1] Dataaspirant, https ://dataaspirant.com/gaussian-naive-bayes-classifier-implementation-python/

Nous obtenons les résultats suivants (figure 4.5) :

- **Accuracy d'entraînement :** 84%.
- **Accuracy de validation :** 83%.

	precision	recall	f1-score	support
Quality 0	0.77	0.97	0.86	1221
Spam 1	0.95	0.69	0.80	1173
accuracy			0.83	2394
macro avg	0.86	0.83	0.83	2394
weighted avg	0.86	0.83	0.83	2394

FIGURE 4.5 – Naïve Bayesian classifiers résultats.

Comme nous pouvons le voir dans la sortie de la fonction (confusion_matrix) (figure 4.6), il y a 358 + 41 = 399 prédictions incorrectes et 1180 + 815 = 1995 prédictions correctes.

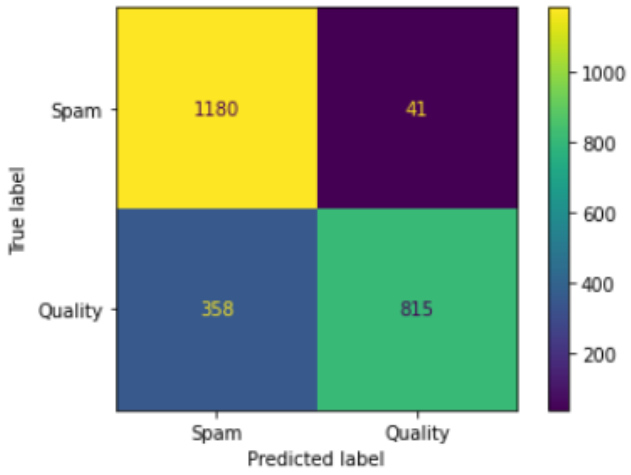

FIGURE 4.6 – Spam et Quality Nombre de tweets détectés à l'aide de Naïve Bayes.

True Positive (TP) est un résultat où le modèle prédit correctement la classe (Spam), TP = 1180.
True Negative (TN) est un résultat où le modèle prédit correctement la classe (Quallity), TN = 825.
False Positive (FP) est un résultat où le modèle prédit de manière incorrecte la classe (Spam), FP = 41.
False Negative (FN) est un résultat où le modèle prédit de manière incorrecte la classe (Quallity), FN = 358.[26]

Precision = $\frac{TP}{TP+FP}$ **Recall** = $\frac{TP}{TP+FN}$ **Accuracy** = $\frac{TP+TN}{TP+TN+FP+FN}$

F1-score = (2 * Precision * Recall) / (Precision + Recall)

4.4.2 K-Nearest Neighbor classifiers (K-NN)

Concernant l'application de l'algorithme KNN sur notre data-set nous avons utilisé la fonction (KNeighborsClassifier) avec les paramètres :

- (n_neighbors = 3) : 3 plus proche voisin.
- (metric = 'manhattan' et 'euclidean' et 'minkowski') : les distances appliquée.

On a obtenu les résultats suivants :

1. Avec la distance (Manhattan) (figure 4.7) :

- **Accuracy d'entraînement :** 83%.
- **Accuracy de validation :** 66%.

	precision	recall	f1-score	support
Quality 0	0.65	0.70	0.67	1221
Spam 1	0.66	0.62	0.64	1173
accuracy			0.66	2394
macro avg	0.66	0.66	0.66	2394
weighted avg	0.66	0.66	0.66	2394

FIGURE 4.7 – K-Nearest Neighbor classifiers validation résultats (Manhattan distance).

2. Avec la distance (Euclidean) (figure 4.8) :

- **Accuracy d'entraînement :** 79%.
- **Accuracy de validation :** 54%.

	precision	recall	f1-score	support
Quality 0	0.55	0.56	0.55	1221
Spam 1	0.53	0.52	0.53	1173
accuracy			0.54	2394
macro avg	0.54	0.54	0.54	2394
weighted avg	0.54	0.54	0.54	2394

FIGURE 4.8 – K-Nearest Neighbor classifiers validation résultats (Euclidean distance).

3. Avec la distance (Minkowski, p→ ∞ : chebyshev distance) (figure 4.9) :

- **Accuracy d'entraînement :** 75%.
- **Accuracy de validation :** 51%.

	precision	recall	f1-score	support
Quality 0	0.51	0.52	0.52	1221
Spam 1	0.49	0.49	0.49	1173
accuracy			0.51	2394
macro avg	0.50	0.50	0.50	2394
weighted avg	0.50	0.51	0.50	2394

FIGURE 4.9 – K-Nearest Neighbor classifiers validation résultats (Minkowski distance).

Comme nous pouvons le voir dans la sortie de la fonction (confusion_matrix) (figure 4.10)

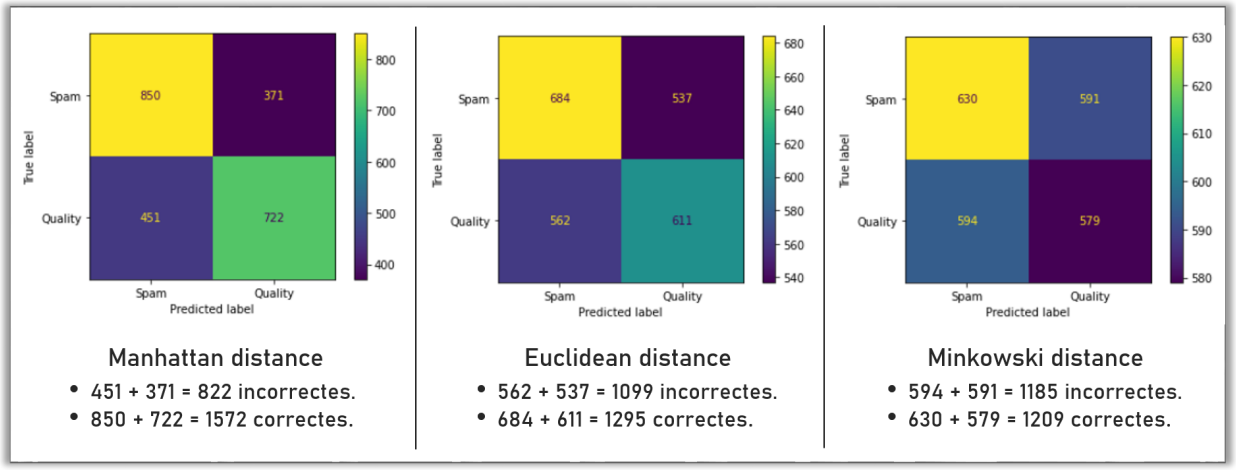

FIGURE 4.10 – Spam et Quality Nombre de tweets détectés à l'aide de KNN.

Grâce à la comparaison des résultats des trois distances (Manhattan, Euclidean, et Minkowski) on peut dire que la distance Manhattan est la meilleure, suivi de la distance Euclidean puis la dernier distance Minkowski.
Nous pouvons dire que l'order (p) a un impact direct sur les résultats,plus que la valeur P est faible,on obtenu meilleurs résultats de classification KNN.

4.5 Implémentions d'une approche bio-inspirée

Puisque dans littérature on a constaté que les méthodes bio sont généralement utiliser pour a finir les méthodes traditionnelles, et effet que produit la combinaison de deux algorithmes le premier traditionnel et le seconde bio inespéré, dans notre qu'a en va étudie la combinaison de KNN avec GWO. On est entraîne de codé cette idée :

Notre approche d'intégration de KNN avec GWO commence après avoir divisé notre ensemble de données en données d'entraînement et en données de test, où nous transmettons les données de test directement à l'algorithme KNN, tandis que les données d'entraînement que nous passerons et améliorerons d'abord en utilisant l'algorithme GWO selon les étapes suivantes :

1- Nous séparons les tweets (Spam) et les tweets (Qualité) pour avoir deux groupes d'entraînement.

2- Nous passons les données à l'algorithme GWO qui utilisent les trois mesures de distance (Euclidean, Chebyshev, Manhattan) comme fonctions de fitness.

3- De ce dernier on a obtenu :
• Les trois premiers tweets de spam a, b et g, respectivement, pour chacun des trois distances, pour un total (les 9 premiers tweets de spam).
• Les trois tweets de meilleure qualité, a, b et g, respectivement, pour chacun des trois distances, pour un total (les 9 premiers tweets de qualité).

4- Nous fusionnons les tweets ensemble pour obtenir un ensemble de données d'entraînement composé des 18 meilleurs tweets de Spam et Qualité, que nous transmettrons à l'algorithme KNN.

5- Nous appliquons l'algorithme KNN en utilisant les 3 plus proches voisins (k=3), plus la distance de Manhattan, pour obtenir les résultats.[11]

• Le but de l'algorithme GWO dans notre approche est de réduire (minimisation) la distance entre le tweet du milieu et tous les autres tweets pour obtenir les trois premiers tweets, qu'il s'agisse de tweets de spam ou de tweets de qualité, ou en d'autres termes d'améliorer les données d'entraînement en général.

Toutes les étapes précédentes sont illustrées dans la Modèle suivante (figure 4.11) :

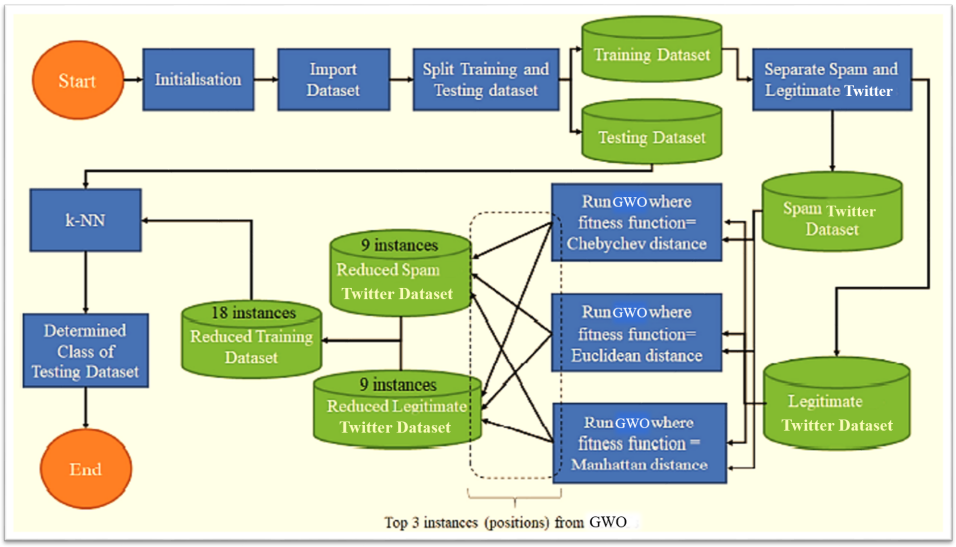

FIGURE 4.11 – Modèle de détection de spam utilisant l'algorithme GWO combiné avec KNN.[11]

4.6 Résultats et comparaison

4.6.1 Parmi les algorithmes traditionnels

Après avoir appliqué les algorithmes Naïve Bayesian et K-Nearest Neighbor au même ensemble de données (data set), nous obtenons les résultats suivants, (tableau 4.3) et graphique (figure 4.12) :

Les Modèles	Naïve Bayes Classifier	K-Neighbors Classifier
Precision	0,95	0,66
Recall	0,69	0,62
F1-Score	0,80	0,64
Accuracy	0,83	0,66
Spam tweets détection	1180 tweets	850 tweets

TABLE 4.3 – les résultats de Naïve Bayes et K-NN avec la distance (Manhattan).

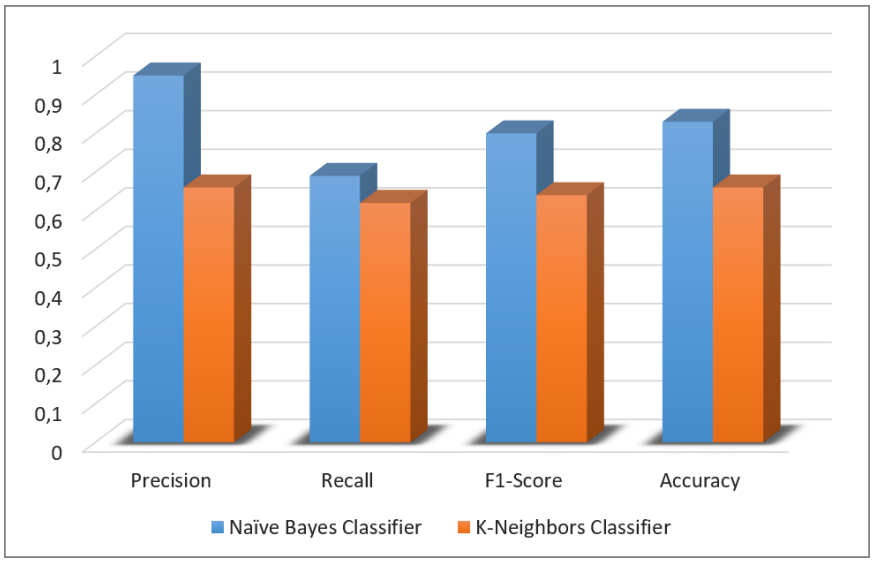

FIGURE 4.12 – Comparez les résultats de Naïve Bayes et K-NN avec la distance (Manhattan).

Grâce aux résultats ci-dessus, nous pouvons dire que l'algorithme Naïve Bayesian est meilleur que K-NN à tous égards, même le temps qu'il faut pour s'exécuter, car la classificateur Naïve Bayesian est meilleure pour gérer des grands ensembles de données que K-NN.

4.6.2 Entre algorithmes traditionnels et bio-inspirés

Pendant notre étude des approches traditionnelle, on a fait une comparaison sur l'utilisation de déférentes distances pour l'algorithme KNN, et on a abouti au résultat suivant :
La meilleure distance qui donne des bons résultats est la distance Manhattan, cette comparaison sera l'objet de l'étude de l'impact de ces distances su KNN combiné avec l'un des algorithmes Bio inspiré.
Nous avons essayé d'implémenter ces idées en Python, mais nous n'avons pas atteint de résultats logiques, mais il est probable que leurs résultats seront meilleurs par rapport aux algorithmes traditionnels seuls, car les algorithmes inspirés de la dynamique sont des algorithmes d'optimisation, qui améliorent les résultats des algorithmes traditionnels lorsque combiné avec eux.

Conclusion Générale

Dans ce mémoire nous concentrons sur les différentes approches de détection de spam dans twetter, approches traditionnelles et bio inspiré, nous avons utilisé une base de données (data set) se nommé (data_train), en tant que fichier CSV, qui se compose de 11 968 lignes (tweets) les données ont été organisées avant de construire le modèle formé, pour la classification de spam on a commencé par l'approche Naïve Bayesian qu'elle fais partie des algorithmes traditionnelle très connu dans la classification,en suite on a utilisé l'approche K-NN avec 3 plus proche voisin, on a appliqué les trois distances, manhattan, euclidean, et minkowski, suivent les résultat en peut dire que la distance Manhattan est la meilleure, quant aux algorithmes d'optimisation bio-inspirés, nous n'avons malheureusement trouvé aucun résultat.

Bibliographie

[1] S. Ray, "Learn naive bayes algorithm." https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/, Sept. 2017.

[2] N. F. Rusland, N. Wahid, S. Kasim, and H. Hafit, "Analysis of naïve bayes algorithm for email spam filtering across multiple datasets," 2017.

[3] Z. W.-B. A. Bienvenue, "Application des algorithmes d'apprentissage automatique pour la détection de défauts," 2020.

[4] Vitalflux,https://vitalflux.com/k-nearest-neighbors-explained-with-python-examples/. consulté le : 16/05/2022.

[5] J. Ni, L. Wu, X. Fan, and S. X. Yang, "Bioinspired intelligent algorithm and its applications for mobile robot control : A survey," *Computational Intelligence and Neuroscience*, vol. 2016, 2016.

[6] S. K. Mosavi, E. Jalalian, and F. S. Gharahchopog, "A comprehensive survey of grey wolf optimizer algorithm and its application."

[7] O. Bozorg-Haddad, ed., *Advanced Optimization by Nature-Inspired Algorithms*, vol. 720. Springer Singapore, 2018.

[8] H. Reddad, M. Zemzami, N. E. Hami, and N. Hmina, "Optimisation métaheuristique et en application mécatronique metaheuristic optimization and mechatronic application," 2022.

[9] A. T. Kabakus and R. Kara, "A survey of spam detection methods on twitter," *International Journal of Advanced Computer Science and Applications*, vol. 8, 2017.

[10] A. Pinandito, R. S. Perdana, M. C. Saputra, and H. M. Az-Zahra, "Spam detection framework for android twitter application using naive bayes and k-nearest neighbor classifiers," pp. 77–82, Association for Computing Machinery, 2 2017.

[11] J. Batra, R. Jain, V. A. Tikkiwal, and A. Chakraborty, "A comprehensive study of spam detection in e-mails using bio-inspired optimization techniques," 2021.

[12] Medium,https://medium.com/analytics-vidhya/na{i}ve-bayes-algorithm-5bf31e9032a2. consulté le : 11/05/2022.

[13] Futura-sciences,https://www.futura-scienes.com/tech/definitions/informatique-reseau-social-10255/. consulté le : 20/04/2022.

[14] P. Guillon, "Etat de l'art du spam, solutions et recommandations," 2008.

[15] "Detecting spam in a twitter network," *First Monday*, 12 2009.

[16] Futura-sciences,https://www.futura-sciences.com/tech/definitions/reseaux-sociaux-twitter-10997/,. consulté le : 20/04/2022.

[17] B. B. Naouel, "Détection de courriels indésirables par apprentissage automatique," Master's thesis, Université Ahmed Ben Bella - Oran 1, 2012.

[18] M. Washha, M. Mezghani, and F. Sèdes, "La qualité de l'information dans les réseaux sociaux en ligne : une approche non supervisée et rapide de détection de spam," in *INFORSID*, 2017.

[19] Journaldunet,https://www.journaldunet.fr/web-tech/guide-de-l-intelligence\protect\@normalcr\relax-artificielle/1501321-classification-naive-bayesienne-/. consulté le : 17/05/2022.

[20] H. AMEL, "Classification supervisée à base de knn avec pondération d'attributs par l'algorithme génétique," 02/02/2015.

[21] Towardsdatascience,https://towardsdatascience.com/9-distance-measures-in-data\protect\@normalcr\relax-science-918109d069fa. consulté le : 29/05/2022.

[22] J. Martinez-Romo and L. Araujo, "Detecting malicious tweets in trending topics using a statistical analysis of language," *Expert Systems with Applications*, vol. 40, no. 8, pp. 2992–3000, 2013.

[23] I. H. I. H. Witten and E. Frank, *Data mining : practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann, 2000.

[24] B. Trstenjak, S. Mikac, and D. Donko, "Knn with tf-idf based framework for text categorization," *Procedia Engineering*, vol. 69, pp. 1356–1364, 2014.

[25] kaggle,https://www.kaggle.com/competitions/twitter-spam/overview. consulté le : 30/05/2022.

[26] S. Shringi, H. Sharma, and D. L. Suthar, "Fitness-based grey wolf optimizer clustering method for spam review detection," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–15, 4 2022.