MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM

Faculté des Sciences Exactes et d'Informatique

Département de Mathématiques et informatique

Filière : Informatique

MEMOIRE DE FIN D'ETUDES

Pour l'Obtention du Diplôme de Master en Informatique

Option : **Ingénierie des Systèmes d'Information**

Présenté par :

BERROUACHEDI Abd Elhak

BOUZID Mohamed El Amine

THEME :

Contribution d'une approche hybride Firefly bio-inspirée pour le diagnostic des données médicales

Soutenu le : 19-06-2021

Devant le jury composé de :

HENNI Fouad	Grade	Université de Mostaganem	Président
KAID SLIMANE B	Grade	Université de Mostaganem	Examinatrice
DJAHAFI Fatiha	Grade	Université de Mostaganem	Encadrante

Année Universitaire 2020-2021

Résumé

Ce travail s'inscrit dans le contexte général de l'aide à la classification et à l'analyse des données médicales par les techniques bio-inspirées. Les algorithmes bio-inspirés sont un nouveau domaine de recherche utilisé pour le diagnostic médical. Il existe plusieurs méthodes bio-inspirées pour résoudre le problème de classification telles que les réseaux de neurones, les algorithmes génétiques et les systèmes immunitaires artificiels. Dans notre travail, nous allons appliquer l'algorithme bio-inspiré Firefly à la classification des données médicales et nous proposons une hybridation entre cet algorithme Firefly et un autre algorithme qui s'inspire de la nature pour améliorer la précision de classification obtenu par l'algorithme Firefly.

Abstract

This work falls within the general framework of assistance in the classification and analysis of medical data using bio-inspired techniques. Bio-inspired algorithms are a new area of research used for medical diagnostics. There are several bio-inspired methods to solve the classification problem such as neural networks, genetic algorithms and artificial immune systems. In our work, we will apply the bio-inspired Firefly algorithm to the classification of medical data and we propose a hybridization between this Firefly algorithm and another nature-inspired algorithm to improve the classification accuracy obtained by the Firefly algorithm.

ملخص

يقع هذا العمل ضمن السياق العام للمساعدة في تصنيف وتحليل البيانات الطبية من خلال تقنيات مستوحاة من الحيوية. تعد الخوارزميات المستوحاة من الحيوية مجالًا جديدًا من الأبحاث المستخدمة في التشخيصات الطبية. هناك العديد من الطرق المستوحاة من الأحياء لحل مشكلة التصنيف مثل الشبكات العصبية والخوارزميات الجينية وأنظمة المناعة الاصطناعية. في عملنا، سنطبق خوارزمية اليراع المستوحاة من الأحياء لتصنيف البيانات الطبية ونقترح تهجينًا بين خوارزمية اليراع هذه وخوارزمية أخرى مستوحاة من الطبيعة لتحسين دقة التصنيف التي حصلت عليها خوارزمية اليراع

Dédicaces

Je dédie ce modeste travail :

A toute ma famille…

A tous mes amis…

A tous ceux qui m'ont aidé…

Remerciements

Nous voulons exprimer par ces quelques lignes de remerciements nos gratitudes envers tout d'abord à nos parents et nos familles pour leur aides, et notre encadrant Madame DJAHAFI Fatiha pour ses conseils et son encadrement.

Après à tous ceux en qui par leur présence, leur soutien, leur disponibilité et leurs conseils nous avons trouvé le courage afin d'accomplir ce projet.

En fin, nous ne pouvons pas achever ce projet sans exprimer nos gratitudes à tous les enseignants de l'Université ABDELHAMID IBN BADIS- MOSTAGANEM, pour leur dévouement et leur assistance tout au long de cette année.

Liste des figures

Liste des tableaux

Liste des abréviations

Abréviation	Expression Complète	Page
IHFD	Ireland Hip Fracture Database	10
AG	Algorithme génétique	24
ACO	Ant Colony Optimization	24
ABC	Artificial Bee Colony	26
AE	Abeille Employée	26
AS	Abeilles Spectateur	26
PSO	Particle Swarm Optimization	27
FA	Firefly Algorithm	29
BFA	Bacteria Forage Algorithm	29
PID	Pima Indian Diabetes	44
BMI	Body mass index	45

Table des matières

Introduction Générale

Allah a créé la nature et lui a donné une grande capacité d'adaptation pour faire face à diverses situations complexes. En effet, quels que soient les problèmes à résoudre et les défis auxquels il est confronté l'être humain, il peut toujours trouver la meilleure solution. Cela a motivé les gens à résoudre divers problèmes rencontrés dans la vie quotidienne.

Au cours des deux dernières décennies, le nombre d'études a augmenté effectuer sur des animaux vivant en groupes ou sociétés, en particulier insectes sociaux. Ces études sur la théorie de l'auto-organisation ont inspiré les gens De nombreux chercheurs développent une nouvelle méthode appelée métaheuristique. L'algorithme métaheuristique convient pour toutes sortes de problèmes. Ils semblaient être la meilleure solution problèmes d'optimisation, ils sont généralement inspirés par la nature.

L'informatique a été introduite dans le domaine médical. Grâce à la puissance de calcul élevée du processeur et à des algorithmes bien définis, l'identification des maladies et même le traitement deviennent plus faciles, et à partir de là, nous disposons de plusieurs méthodes pour classer et analyser les données médicales, dont la plus récente sont les techniques bio-inspiré qui sont des méthodes métaheuristique. Les algorithmes bio-inspiré constituent un nouveau domaine de recherche pour le diagnostic médical.

Après l'introduction générale, nous définissons le contexte général du document et ces objectifs.

Le premier chapitre présente la généralité des techniques de classification et de l’apprentissage, et ses différents types.

Dans le deuxième chapitre, nous allons détailler les différents méthodes bio-inspirés. Il s'agit de présenter les principaux algorithmes et l'état de l'art des dernières méthodes bio-inspirés, ce dernier étant divisé en deux grandes classes qui sont les algorithmes évolutionnaires (inspirés de la sélection naturelle) et les algorithmes basés essaim (inspirés du comportement collectif chez les animaux).

Le troisième chapitre donnée une description théorique de la méthode bio-inspirée appliquée Firefly.

Finalement, le dernier chapitre va résumer les résultats obtenus par l'algorithme appliqué sur la base médicale du diabète.

Chapitre 1

Les Techniques de Classification

1.1 Introduction

Ce chapitre introduit de l'apprentissage, également appelé apprentissage machine, les algorithmes sont les moteurs qui fait avancer l'apprentissage automatique, en général ils existent deux types d'algorithmes qui sont utilisé aujourd'hui : l'apprentissage supervisé et l'apprentissage non supervisé.

La différence entre les deux réside par la façon dont comment chacun faire apprendre les données Pour faire les décisions.

L'objectif de cette introduction est également de dresser un panorama de l'apprentissage et d'expliciter l'articulation entre les chapitres du mémoire.

1.2 Apprentissage

L'apprentissage fait référence au développement, à l'analyse et à la mise en œuvre de méthodes permettant aux machines de résoudre une classe de problèmes tout au long du processus d'apprentissage, résolvant ainsi ces problèmes et exécutant des tâches difficiles ou impossibles à réaliser avec les algorithmes traditionnels. L'apprentissage est un vieux problème qui a été résolu dans de nombreux domaines, dont la psychologie, les statistiques, la pédagogie, l'intelligence artificielle, etc. [28]

Le terme apprentissage correspond aux caractéristiques de la machine. Plus précisément, il s'agit de leur capacité à organiser, construire et résumer les connaissances

pour une utilisation ultérieure, et leur capacité à utiliser l'expérience pour améliorer leurs compétences en résolution de problèmes. L'apprentissage est utilisé au lieu d'obtenir des règles de classification d'experts dans le domaine, évitant ainsi la tâche difficile d'acquérir des connaissances. Attribué par des maladies spécifiques.

L'apprentissage comprendra ici des règles de classification « d'apprentissage» à partir d'un ensemble de descriptions de patients. Nous nous intéressons à un type particulier d'apprentissage, à savoir l'apprentissage inductif, également appelé apprentissage à partir d'exemples d'observation. Le processus d'apprentissage inductif peut être vu comme une recherche de descriptions générales possibles, qui peuvent expliquer les données d'entrée, et prédire de nouvelles descriptions (résultats, données terminologiques, etc.) et des tentatives inductives pour obtenir une description complète et l'exactitude La tentative est très utile. Un phénomène donné dérivé d'une observation spécifique de ce phénomène ou d'une partie de celui-ci. La formation permettra d'utiliser les données d'apprentissage pour optimiser les paramètres du classifieur pour le problème à résoudre. L'endroit où les données d'entraînement sont classées plus tôt. L'apprentissage peut être utilisé pour permettre à l'ordinateur de percevoir l'environnement informatique, comme la reconnaissance d'objets (visages, motifs, formes, etc.). Après l'âge adulte, les créatures vivantes ont de nombreuses capacités pour permettre aux ordinateurs de survivre dans l'environnement. Apprentissage de base, apprentissage sous la supervision d'autrui.

Par conséquent, l'apprentissage automatique d'une machine est un ensemble de tâches que la machine doit effectuer. Il utilise "des expériences telles que l'amélioration de ses performances, et une partie du domaine principal des applications d'apprentissage automatique est l'exploration de données et l'intelligence artificielle qui existent dans l'ensemble de données, Ils peuvent prendre des décisions basées sur des échantillons de ces problèmes pour résoudre automatiquement les problèmes [22].

1.2.1 Les types d’apprentissage

Il existe trois grands types d’apprentissage : apprentissage supervisé, apprentissage non supervisé et apprentissage semi supervisé

1.2.1.1 Apprentissage supervisé

L'apprentissage supervisé regroupe les tâches de classification, de régression et de classement. Il Habituellement, cela est lié au traitement des problèmes de prévision.

L'exemple suivant nous permettra de Comprenez le paradigme de l'apprentissage supervisé. [30] Imaginez une base de données immobilière. Chaque ligne représente une observation (appartement, maison, studio, etc.), représentée par les variables dites prédictives (colonnes : type de propriété, nombre de pièces, superficie en mètre carré, proximité ou non des transports en commun, etc.) [31], Encodé par des chaînes, des nombres, des valeurs booléennes, etc., et encodé par des chaînes, des nombres, des valeurs booléennes, etc. Le but est de prévoir les prix de l'immobilier. Ces lignes sont appelées échantillons, et ces colonnes sont appelées : caractéristiques et ce que nous voulons prédire, variables à prédire : cible. Une fois que cet exemple de base de données (ensemble de données anglais) est appelé apprentissage (train anglais), nous récupérons des échantillons sans prix (échantillons de test) et essayons de prédire les prix de ces nouveaux attributs en apprenant sur l'ensemble de données du train. Mathématiquement parlant, supposons que nous ayons un ensemble de données D, décrit par un ensemble de caractéristiques X.

L'algorithme d'apprentissage supervisé trouve une fonction appelée fonction de mappage ou fonction objectif entre la caractéristique d'entrée X et la variable de sortie Y à prédire. Cette fonction constitue le modèle prédictif. (Cible en anglais) : f (X) 7 → Y divise la variable de sortie Y en deux types de catégories de valeurs. Il peut prendre une valeur continue (l'infini dans un nombre infini de valeurs). Dans ce cas, nous devons traiter des tâches de régression ou des valeurs discrètes (l'infini dans un ensemble fini de valeurs), puis nous devons traiter de la classification Tâches. [25]

1.2.1.1.1 Applications d'apprentissage supervisé

Les applications conçus pour l'apprentissage supervisé sont nombreux nous citerons ci-dessous :

- Vision par ordinateur
- Reconnaissance de formes
- Reconnaissance de l'écriture manuscrite
- Reconnaissance vocale
- Traitement automatique de la langue
- Bio-informatique

1.2.1.2 L'apprentissage non supervisé

L'apprentissage non supervisé n'a pas besoin d'étiqueter les données de base pour l'apprentissage. Il tente de former des classes dans un ensemble d'observations non étiquetées, car ces connaissances sans nom ne sont pas disponibles pour leur appartenance à différentes classes. Ce type de formation peut être utilisé pour découvrir des clusters formés par des ensembles de données. Il doit découvrir personnellement la structure des données. [27] Le système doit ici déterminer la cible des données en fonction des attributs disponibles des données dans l'espace de description (la somme des données), afin de la classer en exemples similaires. Cette forme d'apprentissage utilise un ensemble de données non annotées afin de permettre l'extraction de connaissances organisées à partir de ces données. Par exemple, les épidémiologistes peuvent essayer de proposer des hypothèses explicatives dans une population suffisamment importante de patients atteints d'un cancer du foie, L'ordinateur peut distinguer différents groupes, puis les classer, tels que ceux provenant de l'alcoolisme, de l'exposition à des métaux lourds ou à des toxines telles que l'aflatoxine [23].

1.2.1.2.1 Liste des algorithmes d'apprentissage non supervisé

Il existe plusieurs algorithmes d'apprentissage non supervisé dont :

- K-means clustering (K-moyenne)
- Dimensionality Reduction (Réduction de la dimensionnalité)

- Neural networks (Réseaux de neurones) / Deep Learning
- Principal Component Analysais (Analyse des composants principaux)
- Singulair Value Décomposition (Décomposition en valeur singulière)
- Independent Component Analysais (Analyse en composantes indépendantes)
- Distribution models (Modèles de distribution)
- Hierarchical clustering (Classification hiérarchique)

1.2.1.3 L’apprentissage semi supervisé

Il s'agit d'un apprentissage à mi-chemin entre l'apprentissage supervisé, qui utilise un ensemble de données qui n'est que partiellement annoté (généralement petit).

Dans l'apprentissage semi-supervisé, l'idée de base est de réduire le coût de l'étiquetage en utilisant une petite quantité de données étiquetées et en utilisant des données non étiquetées. Par exemple, une méthode d'apprentissage semi-supervisé très basique consiste à apprendre un modèle de classification à partir de données étiquetées et à prédire la catégorie de données non étiquetées. Ensuite, les données non étiquetées les plus sûres (avec le moins d'incertitude quant à leur classification) et leur étiquette prédite sont ajoutées à la bibliothèque d'apprentissage pour apprendre un nouveau modèle de classification [23]. Il utilise l'ensemble d'apprentissage étiqueté pour estimer la fonction d'appartenance de chaque classe connue, et utilise l'apprentissage non supervisé pour améliorer la précision de cette estimation, détecte les nouvelles classes et apprend sa fonction d'appartenance.

Cette combinaison est particulièrement efficace lorsque l'ensemble de formation étiqueté contient très peu de scores par classe. C'est généralement le cas pour les classes représentant des modes de fonctionnement en échec. De plus, les modes de fonctionnement sont rarement utilisés dans les ensembles de formation, en particulier les modes de fonctionnement en panne. [24]

En effet, pour des raisons de coût ou de sécurité, certains défauts ou modes de fonctionnement dangereux ne seront pas provoqués, donc ce n'est pas dans le centre d'apprentissage. Par conséquent, l'apprentissage non supervisé peut détecter et apprendre de nouveaux modes de fonctionnement qui émergent pour enrichir les connaissances initiales.

L'apprentissage est adapté à un grand nombre d'activités humaines et est particulièrement adapté aux problèmes de prise de décision automatisée, tout comme ces exemples : [26]

- D'établir un diagnostic médical à partir de la description clinique d'un patient.
- De donner une réponse à la demande de prêt bancaire de la part d'un client sur la base de sa situation personnelle.
- De déclencher un processus d'alerte en fonction de signaux reçus par des capteurs.
- De la reconnaissance des formes.
- De la reconnaissance de la parole et du texte écrit.
- De contrôler un processus et de diagnostiquer des pannes.

1.3 Classification

Dans l'analyse des données, les catégories d'exemples de regroupement (généralement dans des conditions non supervisées) sont divisées en plusieurs catégories.

Ces classes sont généralement organisées par structure (groupe ou cluster). La classification automatique est la classification algorithmique des objets. Il s'agit d'attribuer des catégories ou des catégories à chaque objet (ou individu) à classer sur la base de données statistiques. Il utilise toujours l'apprentissage automatique et est utilisé pour la reconnaissance de formes.

1.3.1 Processus de classification

Dans cette section, nous examinerons l'article intitulé « Clustering-Aided Approach for Predicting Patient Outcomes with Application to Elderly Healthcare in Ireland » par Mahmoud Elbattah et Owen Molloy. Cet article est un excellent exemple de l'utilisation d'algorithmes d'apprentissage supervisé et non supervisé dans un projet réel.

Les chercheurs voulaient être en mesure de mieux comprendre les patients âgés ayant subi une fracture de la hanche entrant et sortant d'un hôpital, pensant qu'ils pourraient être en mesure d'optimiser la capacité des patients/lits dans ce dernier en planifiant et en attribuant

correctement les patients en fonction de certaines qualités ou caractéristiques, comme la durée du séjour d'un patient, la date d'enregistrement à l'hôpital, l'âge, les antécédents médicaux et bien plus encore. Pour ce faire, l'hôpital avait besoin de classer les patients en différentes classes pour avoir une meilleure vision des patients et être en mesure de mieux les gérer. [36]

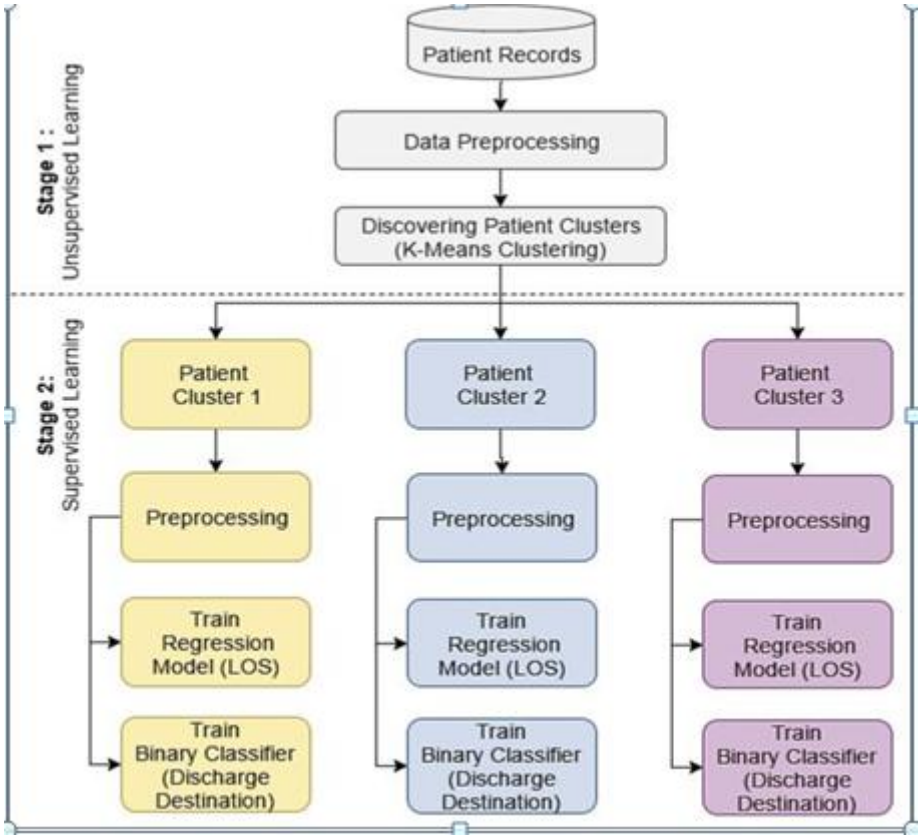

Figure 1 – Aperçu de l'approche. L'approche est décrite sur la base de l'existence de trois groupes de patients [36]

1.3.2 Utilisation de classification supervisé

Maintenant que les chercheurs avaient étiqueté les données. Ils pourraient utiliser les nouvelles bases de données à 3 classes afin de former un modèle de classification. Ce modèle

sera utilisé afin de classer un établissement de soins pour personnes âgées comme une maison de retraite ?

Au cours de cette étape, les nouveaux patients dans 1 des 3 catégories créées lors de la phase d'apprentissage non supervisé. Cette classification permettra à l'hôpital de savoir par exemple où le patient se dirigera après avoir quitté l'hôpital. Est-ce qu'il rentrera chez lui ou devrait-il se rendre dans chercheurs ont utilisé Random Forest pour créer le modèle de classification nécessaire. Le tableau 1 montre le résultat de l'algorithme Random Forest, qui montre quelles qualités contrôlent à quelle catégorie appartient le nouveau patient. [35]

Tableau 1 – Modèle de prédiction LOS : importance des caractéristiques sélectionnées par rapport aux trois clusters. [36]

Feature	Feature Importance Score (≈)		
	Cluster1	Cluster2	Cluster3
Age	0.84	0.49	0.61
Patient Gender	0.14	0.23	0.18
Fracture Type	0.38	0.44	0.21
Hospital Admitted To	0.78	0.93	0.56
ICD-10 Diagnosis	0.48	0.52	0.29
Fragility History	0.44	0.10	0.09
Time To Surgery	0.15	0.27	0.64

1.3.3 Utilisation de classification non supervisé

En utilisant le populaire algorithme Kmeans, les chercheurs ont proposé de classer les patients en 3 catégories ou groupes différents. Ils ont utilisé le dataset IHFD (Ireland Hip Fracture Database) qui contient des données de 4773 patients âgés ayant subi une fracture de la hanche, et ils ont étiqueté cet ensemble de données en utilisant 3 étiquettes (classes) différentes en utilisant l'algorithme KMeans (K=3). Le tableau 1 montre les variables ou qualités explorées dans le jeu de données afin de créer les 3 différents clusters. La figure B montre le résultat de l'algorithme de clustering. [37][38]

Tableau 2 – Variables explorées comme caractéristiques possibles. [36]

Variables Explored	
Source Hospital	Admission Type
Discharge Code	Patient Gender
Residence Area	Discharge Status
Admission Source	Hospital Transferred From
Hospital Transferred To	LOS
Age	ICD-10 Diagnosis
Admission Trauma Type	Admission via ED
Pre-Fracture Mobility	Fracture Type
Fragility History	Specialist Falls Assessment
Multi-Rehabilitation Assessment	

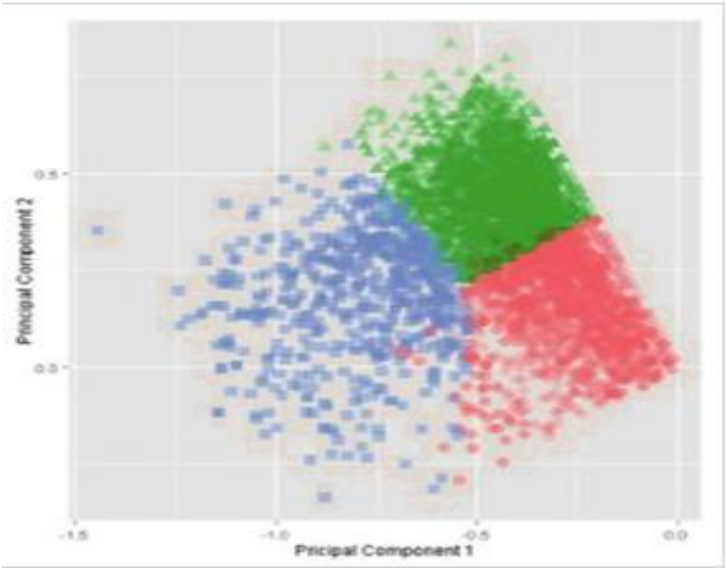

Figure 2 – Regroupement des expériences avec un certain nombre de groupes (K) 3/7

Les groupes sont présentés sur la base des deux composants principaux. Une incohérence de couleur indique moins de séparation entre les groupes. Les visualisations ont été produites à l'aide du package R ggplot (Wickham 2009). [32]

1.3.4 Exemple de problèmes de classification

Il existe deux principaux types de problèmes de classification. S'il y a une classe a priori associée à chaque observation, et que le but de la classification est de respecter au mieux ces leçons a priori, alors nous serons confrontés à la discrimination, à l'encadrement de la classification ou à l'apprentissage avec les enseignants. S'il n'y a pas de classification préalable, le but de la classification est de classer ces personnes dans la même catégorie en fonction de l'ensemble de variables gagnantes. Ce type de problème est le problème de la classification supervisée ou de l'apprentissage non-machine ou de l'apprentissage de la classification non-enseignant [29].

1.3.4.1 Classifications supervisées

Les problèmes suivants sont des exemples de classification supervisée [33] :

- La reconnaissance de formes dans une image (reconnaissance et identification de caractères manuscrits par exemple) ou une vidéo.
- Reconnaissance de parole.
- Reconnaissance de l'auteur ou de la thématique d'un texte.
- Détection des spam.
- Aide au diagnostic médical.
- Détection de gènes dans une séquence ADN.
- Attribution ou non d'un prêt à un client d'une banque.
- La classification des produits boursiers pour tenter de savoir s'ils sont ou non sous-évalués.

Parmi les méthodes de résolution utilisées en dehors des algorithmes évolutionnaires, il existe :

- La méthode des k plus proches voisins.
- Le classifieur Bayésien naïf.
- Les arbres de décision (ID3, C45, CART...).
- Les réseaux de neurones
- Les machines à vecteurs de support.

1.3.4.2 Classifications non supervisées

Les applications possibles sont notamment [34] :

- En biologie, pour la classification des plantes et des animaux.
- Dans le domaine des assurances, pour identifier les assurés les moins rentables ou pour identifier les cas possibles de fraudes.
- Dans la gestion des villes, pour identifier des groupes d'habitations.
- Dans l'étude des tremblements de terre, pour identifier les dangereux.

Parmi les méthodes de résolution utilisées en dehors des algorithmes évolutionnaires, il existe :

- L'algorithme des k – moyennes.
- Les modèles de mélange.
- Le regroupement hiérarchique.
- Le modèle de Markov caché.
- Les cartes auto adaptatives.

1.3.5 Avantages de la classification

- La capacité de prédire l'existence de classes a priori.

- Spécifier certains paramètres relatifs à la distance entre les classes et la variance à l'intérieur même d'une classe.
- Le caractère automatique : appréciable dans le cas d'un grand nombre d'images à traiter.
- La classification est effectuée selon les critères mathématiques indépendants de l'application.
- La séparation et l’ajout des caractéristiques de chacune de ces sous-classes.

1.4 Conclusion

Dans ce chapitre, nous avons présenté l'apprentissage automatique, les types d'apprentissage automatique et certaines applications qui reposent sur l'apprentissage automatique. Nous avons également trouvé deux types de classifications. Nous identifions des catégories basées sur des données statistiques et les classons pour organiser des catégories afin d'identifier des modèles.

Dans le deuxième chapitre, nous définirons des méthodes bio-inspirées pour classer et analyser des données médicales.

Chapitre 2

Les méthodes bio-inspirées

2.1 Introduction

Au fil du temps, des problèmes complexes sont apparus en informatique et la technologie existante n'est pas adaptée à la situation à gérer. Par conséquent, nous devons proposer des nouvelles technologies de résolution. Parmi ces méthodes, nous avons des méthodes bio-inspirées, qui sont des algorithmes inspirés de la nature. Ces méthodes deviennent plus intéressantes en raison de leur efficacité à résoudre des problèmes d'optimisation combinatoire.

Les méthodes bio-inspiré ont été largement utilisées en ingénierie, en mathématiques et en particulier dans le domaine médical. Ces méthodes ont un impact important sur le domaine médical et peuvent aider les gens à diagnostiquer de nombreuses maladies.

2.2 Définition

2.2.1 Métaheuristique

Une Métaheuristique est l'extension d'une heuristique ce qui la rends plus complète d'une part et plus complexe d'une autre part, le but de l'extension est l'obtention d'une solution de qualité supérieure, elle est caractérisée par son comportement stochastique itératif qui vise à progresser vers un optimum global quel que soit le point de départ, et s'inspire essentiellement des systèmes naturels.

Les métaheuristiques ont comme caractéristiques communes de par leurs caractères stochastiques, c.à.d. qu'une partie de la recherche est conduite de façon aléatoire, elles sont

inspirées d'analogies avec la réalité : physique (recuit simulé,…), biologie (algorithmes évolutionnaires, recherche tabou,...) ou éthologie (colonies de fourmis,...). [1]

2.2.2 Bio-inspiration

La bio-inspiration est une approche qui conduit à s'inspirer de la nature pour développer de nouveaux systèmes. La bio-inspiration s'appuie souvent sur le biomimétisme. Comme lui, elle peut puiser son inspiration tant dans le monde des végétaux que des animaux et des champignons, ou des bactéries et des virus. Elle est utilisée dans la conception de logiciels, d'algorithmes génétiques et/ou d'algorithmes évolutionnistes [2].

2.2.3 Processus d'un modèle inspiration

Pour passer d'un modèle naturel à un système artificiel, il y a quelques étapes à suivre. Tout d'abord, la nature inspire les humains à développer une observation d'un phénomène naturel particulier. Ensuite, ils créent un modèle et le testent à l'aide de simulations mathématiques, ce qui permet d'affiner le modèle original. Ensuite, le modèle affiné est utilisé pour extraire un méta heuristique qui pourra servir de base pour finalement concevoir et régler un algorithme inspiré de la nature.

Le processus de modèle inspiration est montré dans la figure suivante :

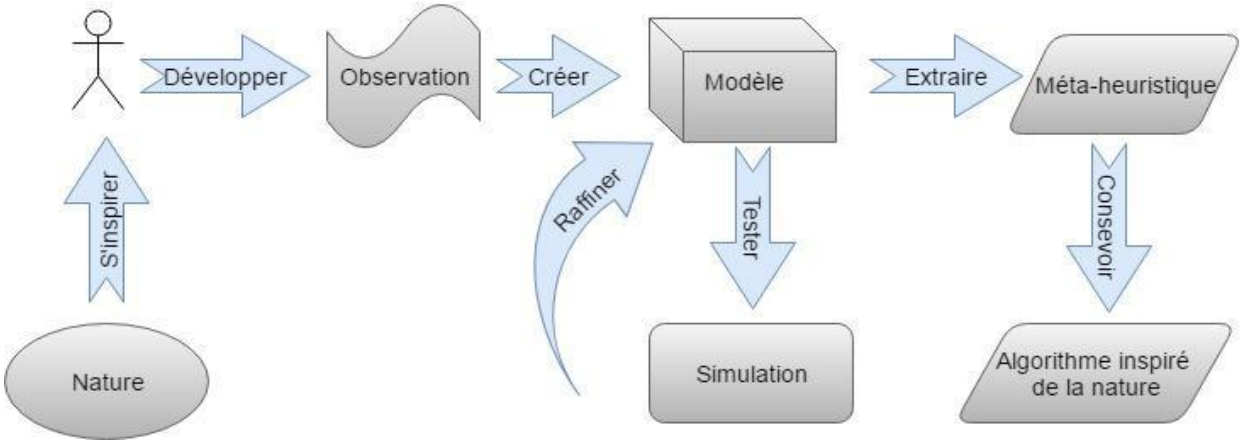

Figure 3 – Processus d'inspiration d'un phénomène naturel [17]

Ce processus se déroule en trois phases principales [17] :

- Observation : Avant toute chose, il faut observer et étudier la nature. Cette observation s'effectue en collaboration avec des biologistes. Lorsqu'un phénomène, une 'astuce' de la nature est observée, vient ensuite l'étape de la compréhension.
- Modélisation : L'observation ayant eu lieu, l'objectif est maintenant de mathématiser ce qui a été observé. Il ne suffit pas de copier, il faut comprendre. Il s'agit là du point essentiel pour permettre d'exploiter le phénomène observé.
- Implémentation : L'étape de la compréhension étant parfaitement réalisée, il devient alors possible d'implémenter l'observation première sur des dispositifs artificiels.

2.3 Classification des méthodes bio-inspirées

Il existe de nombreux algorithmes bio-inspirés développés à partir de diverses bio-inspirations. Donc selon la source d'inspiration, Les méthodes bio-inspirés peuvent être réparties en deux grandes classes qui sont les algorithmes évolutionnaires (inspirés de la sélection naturelle) et les algorithmes basés essaim (inspirés du comportement collectif chez les animaux).

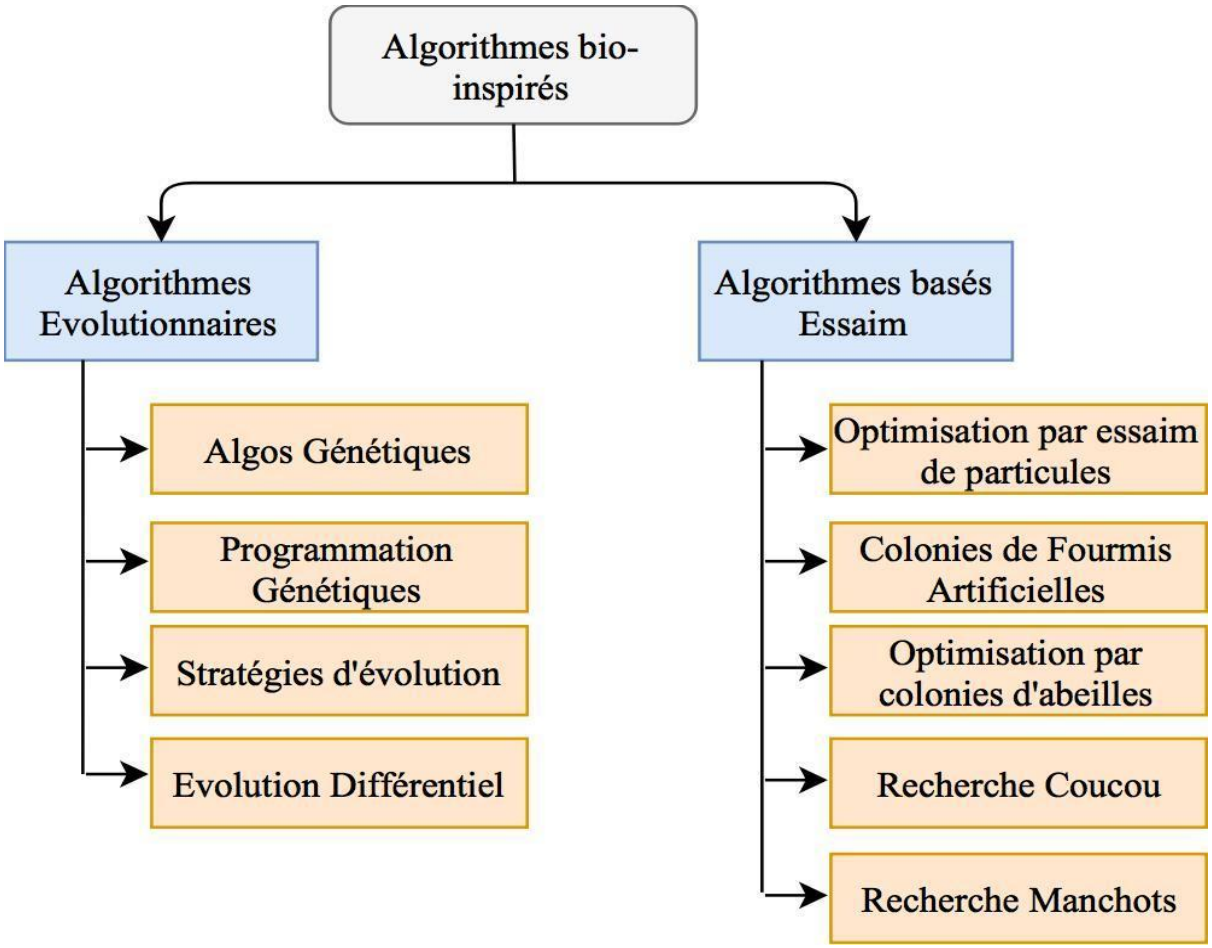

Figure 4 – Classification de méthodes bio-inspirées [6]

2.3.1 Algorithmes évolutionnaires

Les algorithmes évolutionnaires sont des techniques de recherche inspirées par l'évolution biologique des espèces. Ils s'inspirent de l'évolution des êtres vivants (la théorie Darwinienne de la sélection naturelle des espèces) pour résoudre des problèmes d'optimisation.

L'idée ici est que, les individus qui ont hérité des caractères bien adaptés à leur milieu ont tendance à vivre assez longtemps pour se reproduire, alors que les plus faibles ont tendance à disparaître [5]. Ceux-ci comprennent les algorithmes génétiques, programmation génétique, stratégies d'évolution et évolution différentielle.

2.3.2 Algorithmes basés essaim

Les algorithmes basés essaim sont des techniques d'optimisation inspirés du comportement collectif chez les espèces sociaux comme les fourmis, les abeilles, les guêpes, les termites (fourmis blanches), les poissons et les oiseaux. Qui sont des populations d'agents extrêmement simples, interagissant et communiquant indirectement à travers leur environnement, constituent des algorithmes distribués pour résoudre les problèmes réels difficiles. Parmi les algorithmes d'optimisation inspirés de l'intelligence en essaim les plus réussis, sont les colonies de fourmis et l'optimisation par essaim de particules, optimisation par la colonie d'abeille et récemment la recherche coucou.

2.4 Quelques algorithmes bio-inspirés

2.4.1 Algorithme génétique

2.4.1.1 Définition

Les algorithmes génétiques sont des algorithmes d'optimisation s'appuyant sur des techniques d´dérivées de la génétique et de l'´évolution naturelle : croisements, mutations, sélection, etc. Les algorithmes génétiques ont déjà une histoire relativement ancienne puisque les premiers travaux de John Holland sur les systèmes adaptatifs remontent à 1962 [8]. L'ouvrage de David Goldberg [9] a largement contribué à les vulgariser.

2.4.1.2 Principe

Un algorithme génétique recherche le ou les extrema d'une fonction d´définie sur un espace de données. Pour l'utiliser, on doit disposer des cinq éléments suivants :

1. Un principe de codage de l'élément de population. Cette étape associe à chacun des points de l'espace d'état une structure de données. Elle se place généralement après une phase de modélisation mathématique du problème traite. La qualité du codage des données conditionne le succès des algorithmes génétiques. Le codage binaires ont été très utilises à l'origine. Les

codages réels sont désormais largement utilisés, notamment dans les domaines applicatifs pour l'optimisation de problèmes à variables réelles.

2. Un mécanisme de génération de la population initiale. Ce mécanisme doit être capable de produire une population d'individus non homogène qui servira de base pour les générations futures. Le choix de la population initiale est important car il peut rendre plus ou moins rapide la convergence vers l'optimum global. Dans le cas où l'on ne connaît rien du problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche.

3. Une fonction à optimiser. Celle-ci retourne une valeur de R+ appelée fitness ou fonction d'évaluation de l'individu.

4. Des opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace d'état. L'opérateur de croisement recompose les gènes d'individus existant dans la population, l'opérateur de mutation a pour but de garantir l'exploration de l'espace d'états.

5. Des paramètres de dimensionnement : taille de la population, nombre total de générations ou critère d'arrêt, probabilités d'application des opérateurs de croisement et de mutation.

Le principe général du fonctionnement d'un algorithme génétique est représenté sur la figure5 :

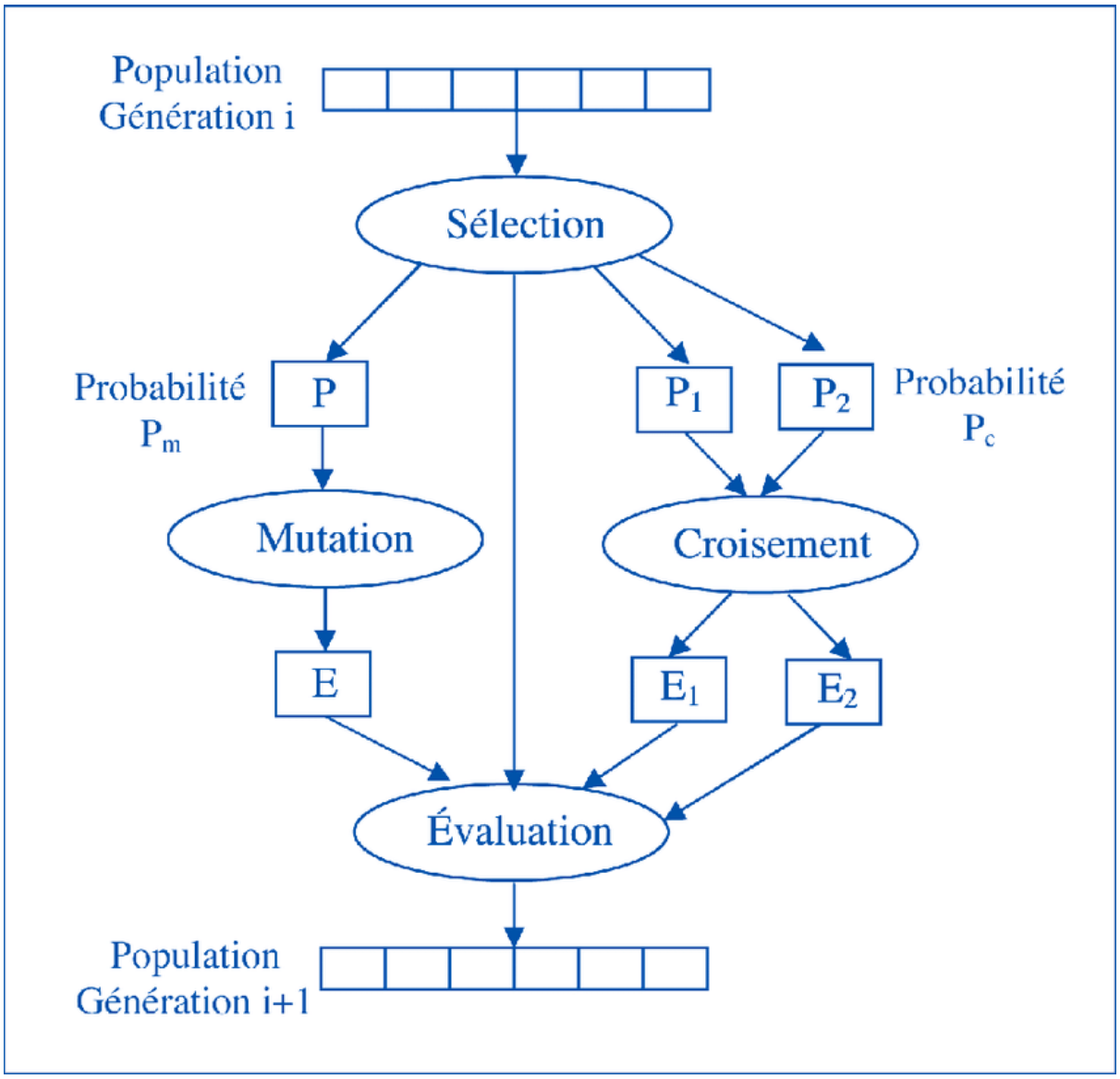

Figure 5 – Principe général des algorithmes génétiques [20]

On commence par générer une population d'individus de façon aléatoire. Pour passer d'une génération **i** à la génération **i+1**, les trois opérations suivantes sont répétées pour tous les éléments de la population **i**. Des couples de parents $\mathbf{P_1}$ et $\mathbf{P_2}$ sont **sélectionnés** en fonction de leurs adaptations. L'opérateur de **croisement** leur est appliqué avec une probabilité $\mathbf{P_c}$ et génère des couples d'enfants $\mathbf{E_1}$ et $\mathbf{E_2}$. D'autres éléments **P** sont **sélectionnés** en fonction de

leur adaptation. L'opérateur de **mutation** leur est appliqué avec la probabilité $\mathbf{P_m}$ ($\mathbf{P_m}$ est généralement très inférieur à $\mathbf{P_c}$) et génère des individus mutés **E**. Le niveau d'adaptation des enfants ($\mathbf{E_1}$, $\mathbf{E_2}$) et des individus mutés **E** sont ensuite **évalués** avant insertion dans la nouvelle population.

Différents critères d'arrêt de l'algorithme peuvent être choisis :

- Le nombre de générations que l'on souhaite exécuter peut être fixé à priori. C'est ce que l'on est tenté de faire lorsque l'on doit trouver une solution dans un temps limité.
- L'algorithme peut être arrêté lorsque la population n'évolue plus ou plus suffisamment rapidement.

2.4.1.3 Avantage

L'avantage des AG est leur simplicité. Néanmoins, les AG seuls ne sont pas très efficaces dans la résolution d'un problème. Cependant, ils apportent une solution acceptable assez rapidement. Sa combinaison avec un algorithme déterministe peut améliorer la solution d'une manière assez efficace.

2.4.2 Colonie des fourmis artificielle

2.4.2.1 Définition

Les algorithmes de colonies de fourmis (en anglais, ant colony optimization, ou ACO) sont des algorithmes inspirés du comportement des fourmis, ou d'autres espèces formant un super organisme, et qui constituent une famille de métaheuristiques d'optimisation.

Initialement proposé par Marco Dorigo et al [10,11] dans les années 1990 pour la recherche de chemins optimaux dans un graphe, le premier algorithme s'inspire du comportement des fourmis recherchant un chemin entre leur colonie et une source de nourriture. L'idée originale s'est depuis diversifiée pour résoudre une classe plus large de problèmes et plusieurs algorithmes ont vu le jour, s'inspirant de divers aspects du comportement des fourmis.

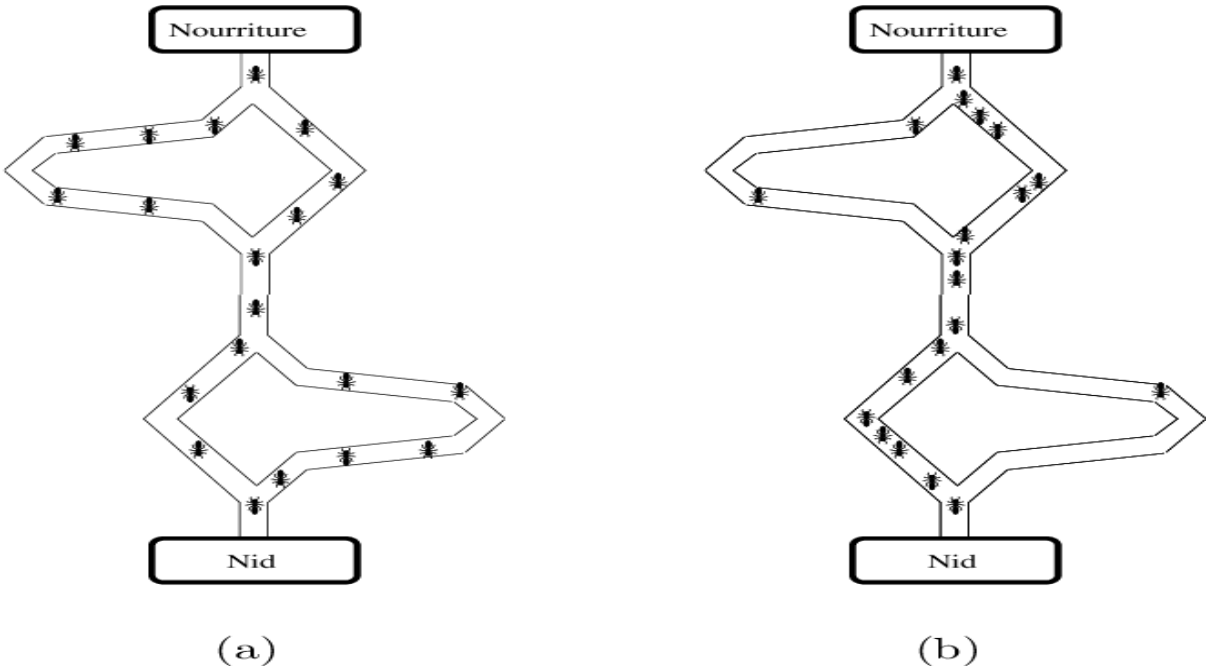

Figure 6 – Recueil de ressources par des fourmis

2.4.2.2 Principe

Le premier algorithme conçu selon ce modèle était destiné à résoudre le problème du voyageur de commerce. Le principe consiste à « lancer » des fourmis, et à les laisser élaborer pas à pas la solution, en allant d'une ville à l'autre. Au début, plusieurs chemins de longueurs différentes sont possibles, et les fourmis les empruntent tous en s'orientant au hasard. Ce faisant, elles laissent derrière elles des traces de phéromones. Sachant que ces traces s'évaporent petit à petit, et que le chemin le plus court permet par définition un plus grand nombre d'aller-retour. Les traces de phéromone se concentrent assez rapidement sur l'itinéraire optimal. C'est donc un algorithme qui repose sur la construction progressive de solutions, un peu comme dans la méthode GRASP (pour Greedy Randomized Adaptive Search 14 Procedure en Anglais), qui inclut également une phase de construction. Afin de ne pas revenir sur ses pas, une fourmi tient à jour une liste Tabou, qui contient la liste des villes déjà visitées.

2.4.2.3 Avantage

L'algorithme de colonies de fourmis offre beaucoup de souplesse, et il est possible de l'adapter à tous les grands problèmes combinatoires classiques. Par ailleurs, l'algorithme de colonies de fourmis se parallélise de façon très naturelle, en affectant par exemple un processus différent pour traiter la marche de chaque fourmi, et un autre pour mettre à jour les pistes de phéromones. L'algorithme, de par son dynamisme intrinsèque, s'adapte aussi très bien aux espaces de solutions qui varient dynamiquement dans le temps. Malgré ces avantages, il comporte certaines limites. Par exemple, il ne fonctionne pas bien quand un grand nombre d'arcs sur le graphe de construction font partie des bons chemins qui ont des valeurs de probabilité égale.

2.4.3 Colonie d'abeille artificielle

2.4.3.1 Définition

L'algorithme de colonie d'abeille artificielle (Artificial Bee Colony ABC en Anglais) est l'un des plus récemment introduit dans la base des Algorithmes basés essaim. ABC simule le comportement intelligent de recherche de nourriture d'un essaim d'abeilles. L'algorithme de colonie d'abeille artificielle a été introduit par Karaboga [12].

2.4.3.2 Principe

Chaque solution représente une position de nourriture potentielle dans l'espace de recherche et la qualité de la solution correspond à la qualité de la position alimentaire. Agents (abeilles artificielles) recherche à exploiter les sources de nourriture dans l'espace de recherche.

L'ABC utilise trois types d'agents : les abeilles employés (Employed Bee en Anglais), les abeilles spectateur (Onlookers Bee en Anglais), et les scouts (Scouts Bee en Anglais). Les abeilles employées (AE) sont associés à des solutions actuelles de l'algorithme. À chaque étape de l'algorithme un AE tente d'améliorer la solution, il la représente en utilisant une étape de recherche locale, après il va essayer de recruter des abeilles spectateur (AS) pour sa position actuelle. Les ASs choisissent parmi les postes promus en fonction de leur qualité, ce

qui signifie que de meilleures solutions attireront plus d'AS. Une fois un AS a choisi un AE et donc une solution, il cherche à optimiser la position de l'AE par une étape de recherche locale. AE mise à jour sa position si un AS recruté était en mesure de repérer une meilleure position, sinon il reste sur sa position actuelle. En outre, un AE va abandonner sa position si elle n'était pas en mesure d'améliorer sa position pour certain nombre d'étapes. Quand une AE abandonne sa position, il devient une Abeille éclaireuse, ce qui signifie qu'elle sélectionne une position aléatoire dans l'espace de recherche et devienne une Abeille employée à cette position.

2.4.3.3 Avantage

L'algorithme de la colonie d'abeille artificiel présente les avantages d'une grande robustesse, une convergence rapide, une grande flexibilité et moins de paramètres de contrôle [13].

2.4.4 Optimisation par essaim de particules

2.4.4.1 Définition

L'optimisation par essaim de particules (en anglais, PSO Particle Swarm Optimization) est un modèle stochastique, une technique d'optimisation basée population qui peut être appliquée à un large éventail de problèmes.

Le PSO qui a été introduite par Kennedy et Eberhart [14] en 1995 est l'un des paradigmes les plus importants de la Swarm Intelligence

2.4.4.2 Principe

Le PSO utilise un mécanisme simple qui imite les comportements d'essaims comme les volées d'oiseaux et les bancs de poissons pour guider les particules à la recherche de solutions globales optimales. Comme PSO est simple à mettre en œuvre, il a beaucoup progressé ces dernières années dans la résolution de problèmes d'optimisation du monde réel [15].

Le PSO s‘inspire du comportement social basé sur l‘analyse de l‘environnement et du voisinage qui constitue une méthode de recherche d‘optimum par l‘observation des tendances des individus voisins. Chaque individu cherche à optimiser ses chances en suivant une tendance qu‘il modère par son propre vécu. L‘ensemble d‘individus originellement disposés de façon aléatoire et homogène, que nous appellerons des particules, se déplacent dans l‘espace de recherche et constituent, chacune, une solution potentielle. Chaque particule dispose d‘une mémoire concernant sa meilleure solution visitée ainsi que la capacité de communiquer avec les particules voisines. A partir de ces informations, la particule va suivre une tendance basée, d‘une part, sur sa volonté à retourner vers sa solution optimale, et, d‘autre part, sur son mimétisme par rapport aux solutions trouvées chez les particules voisines. La figure suivante représente un élément du PSO.

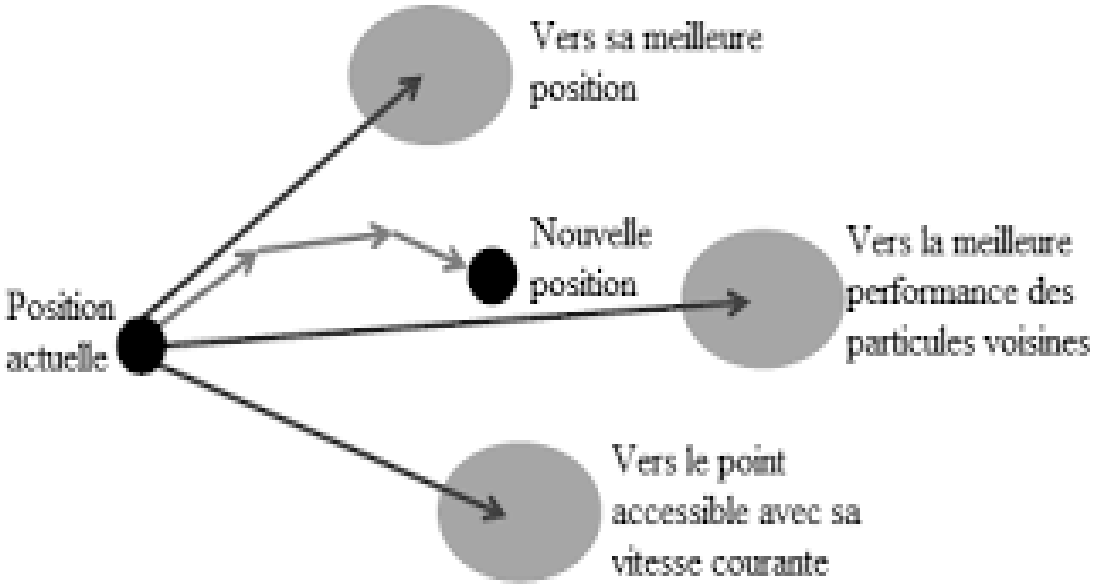

Figure 7 – Eléments du comportement des particules d‘un essaim [21]

2.4.4.3 Avantage

PSO utilise la mémoire pour stoker l’historique de la meilleure position locale et la meilleure position globale des essaims, qui aide non seulement chaque particule pour

sauvegarder leur expérience locale, mais aussi pour aident les autres particules de communiquer par leur expérience social entre eux, ce qui permet de converger vers les régions les plus prometteuses dans l'espace de recherche, et accélère le processus d'optimisation vers la solution optimal.

2.4.5 Algorithme des lucioles (Firefly)

2.4.5.1 Définition

L'algorithme Firefly (FA) est une métaheuristique, bio-inspirée, introduite par Dr Xin-She Yang à l'université Cambridge en 2007, c'est une technique récente inspirée de la nature conçu pour résoudre les problèmes d'optimisation non linéaire. Cet algorithme est basé sur le comportement des insectes sociaux (lucioles) [3].

2.4.5.2 Principe

L'algorithme est basé sur le principe d'attraction entre les lucioles et simule le comportement d'un essaim de lucioles dans la nature, ce qui lui donne beaucoup de similarités avec d'autres métaheuristiques basées sur l'intelligence collective du groupe tel que l'algorithme PSO (Particle Swarm Optimisation), l'algorithme d'optimisation par colonies d'abeilles (ABC), et l'algorithme des bactéries de fourrages (BFA) [4].

L'algorithme prend en considération les trois points suivant [4] :

- Toutes les lucioles sont unisexe, ce qui fait l'attraction entre celles-ci n'est pas en fonction de leur sexe.

- L'attraction est proportionnelle à leurs luminosités, donc pour deux lucioles, la moins lumineuse se déplacera vers la plus lumineuse. Si aucune luciole n'est lumineuse qu'une luciole particulière, cette dernière se déplacera aléatoirement.

- La luminosité des lucioles est déterminée en fonction d'une fonction objective (à optimiser).

2.4.5.3 Avantage

Toutefois, un scénario dans lequel FA est très utile est lorsque la fonction objectif être réduits au minimum a des solutions multiples. Même si ce n'est pas tout à fait évident, il s'avère que, FA automatiquement s'organise en essaims secondaires qui peuvent trouver des solutions multiples simultanément. [16]

L'inconvénient majeur de ces méthodes est qu'elles n'ont pas toujours trouvé la solution exacte vu que l'espace de recherche limité par la solution initiale.

2.5 Etat de l'art des méthodes bio-inspirées

Plusieurs travaux sur la classification des données médicales par les méthodes bio-inspirées ont été recensés. Dans [7], les auteurs ont développé un algorithme basé sur les colonies de fourmis (ACO) pour extraire un ensemble de règles floues pour la classification du diabète. La précision de classification obtenue est de 84,24%, ce qui révèle que cette méthode surpasse plusieurs méthodes célèbres et récentes en matière de précision de classification pour le diagnostic des maladies du diabète.

Dans [18], les auteurs ont proposé des algorithmes bio-inspirées pour mesurer Les performances de l'ensemble de données UCI du cancer du sein diagnostique du Wisconsin, et les résultats ont été calculés à l'aide de différents classificateurs sur les caractéristiques sélectionnées. Après l'expérience, on voit que le PSO a montré une précision maximale de 96,45%, le FA a montré des résultats considérables de 95,81%, l'ABC a montré une précision de 94% et pour l'ACO a montré une précision de 95.18%.

Dans [19], les auteurs ont obtenu l'ensemble de données du référentiel d'apprentissage machine de l'Université de Californie, Irvine (UCI) (Bache et Lichman, 2002), et ils en ont pris des données dermatologiques et hépatites. Sur la base de la classification de l'algorithme Firefly, les résultats obtenus à partir de la précision sont respectivement de 90% et 82%.

Le tableau suivant représente quelques travaux réalisés par les méthodes bio-inspirées pour différentes données médicales :

Tableau 3 – Travaux réalisés par les méthodes bio-inspirées

Auteurs (Année)	Méthodes (Type de donnée médicale)	Taux de précision(%)
M.F. Ganji et M.S. Abadeh. (2011) [7]	ACO (Pima Indian Diabetes)	79.48
Sharma, M., Gupta, S., Sharma, P. and Gupta, D. (2019) [18]	PSO (Cancer de sein)	96.45
	FA (Cancer de sein)	95.81
	ABC (Cancer de sein)	94
	ACO (Cancer de sein)	95.18
E. M. Mashhour, E. M. F. El Houby, K. T. Wassif et al. (2018) [19]	FA (Dermatologie)	90
	FA (Hépatites)	82

2.6 Conclusion

Dans ce chapitre, nous avons fait une vue global sur l'utilisation des modèles bio-inspiré en citant les différents algorithmes évolutionnaires et les algorithmes basés essaim ainsi que leurs avantages. En outre nous avons abordé une synthèse des travaux réalisés sur différents type de données médicales par les plusieurs algorithmes bio-inspirés.

Dans le prochain chapitre, nous allons présenter une hybridation de notre choix de l'algorithme firefly avec d'autre algorithme bio-inspiré et expliciter comment l'hybridation pourrait améliorer les performances de classification.

Chapitre 3

Étude théorique de la méthode appliquée

3.1 Introduction

L'algorithme de luciole (Firefly ‘FA’ en anglais) est une technique récente inspirée de la nature conçu pour résoudre les problèmes d'optimisation non linéaire. Cet algorithme est basé sur le comportement des insectes sociaux (lucioles) [47]. Dans les colonies d'insectes sociaux, chaque individu semble avoir son propre ordre du jour et pourtant le groupe dans son ensemble qui semble être très organisé. Cet Algorithmes basés sur la nature a été mis en évidence pour montrer l'efficacité et l'efficience de résoudre des problèmes d'optimisation difficiles. Un essaim des lucioles est un groupe de systèmes multi-agents dans laquelle des simples agents ce coordonner leurs activités afin de résoudre les problèmes complexe de l'attribution de la communication à des sites multiples fourragères dans des environnements dynamiques [56].

Dans ce chapitre, nous présenterons cette méthode bio-inspirée qui est un algorithme évolutif. Pour cela, nous commencerons par expliquer les aspects biologiques de l'algorithme Luciole .Ensuite, nous décrirons le fonctionnement de l'algorithme.

3.2 Lucioles naturelles

Les lucioles (Firefly) sont des insectes appartenant à la famille abeilles, la famille des lucioles et des lamproies (Lampyridae) comprend plus de 2000 presque toutes les espèces connues de coléoptères produisent de la lumière (du jaune au verdâtre, longueur d'onde de 510 à 670 nanomètres), sous forme de larves et / ou d'adultes, répartis sur tous les continents.

Ces insectes, en tant que petits prédateurs des couches herbacées et touffues, jouent un rôle important dans leur niche écologique, notamment en limitant reproduction de chenilles, d'escargots et d'escargots [39].

Bien que ces espèces fassent partie des coléoptères, la plupart des femelles ne sont pas ils ne peuvent pas voler, ils ressemblent à leurs propres larves, d'où le nom de « ver ». Les plus le lampre ou le ver luisant commun (Lampyris noctiluca) est connu. Dans les lucioles, par exemple en Europe, Luciola lusitanica Charpentier, la femelle a des ailes mais ne vole pas [40].

3.2.1 Description de Luciole

La luciole est en fait un coléoptère de la famille des insectes Lampyridae, qui en grec signifie « lumineux ». Cette famille comprend également d'autres espèces brillantes. Bien que Lampyris noctiluca soit souvent appelé un ver luisant, ce n'est pas du tout un vermifuge. D'autres noms que vous avez peut-être entendus pour la famille des Lampyrides en général sont les lucioles et les paratonnerres. Lampyris noctiluca est généralement de couleur brunâtre à noirâtre. La femelle adulte mesure de 12 à 20 mm de long, tandis que les mâles sont beaucoup plus petits. Les larves ne mesurent souvent que quelques millimètres de long [41].

Les mâles ont deux paires d'ailes, mais n'utilisent que la deuxième paire pour voler. La première paire d'ailes, les élytres, forme une couverture sur la seconde paire. Les femelles ne volent pas [42].

Ils sont doux et allongés. Sa tête est cachée d'en haut par un pronuto et ses antennes sont des fils. Seuls les derniers segments abdominaux sont brillants. [43]

La femme adulte est responsable de la lueur la plus active même si la larve, qui ressemble beaucoup à la femme avec respect, brille également. Le mâle peut briller légèrement, mais il est très différent de la femme qui utilise ses organes brillants pour attirer et stimuler le mâle. Les larves brillent beaucoup plus faiblement et seulement par

intermittence, pendant quelques secondes à la fois. Ils ne semblent pas non plus des versets, mais ils ont des corps segmentés et six jambes au bout de la tête, assez semblables aux adultes. Cependant, quand ils s'aident eux-mêmes avec leurs files d'attente, ils apparaissent un peu comme les chenilles [44]

Figure 8 – Corps de luciole

3.2.2 Cycle vitale Luciole

Le cycle vital varie dans les lucioles, mais l'exemple suivant s'applique à différentes espèces. L'accouplement a lieu sur l'obscurité, le printemps et le début de l'été. L'émission de signaux lumineux, associée au comportement de la Cour, permet aux partenaires de se rencontrer. Dans plusieurs espèces, le mâle émet les éclairs de lumière en vol et est attiré par la réponse brillante de la femelle immobile.

Après accouplement, la femelle goutte ses œufs dans un endroit humide. Il meurt peu de temps après la pose, tandis que le mâle s'éteint après la période de couplage. Les œufs

éclosent environ un mois plus tard. Les larves ont un corps aplati et allongé. Le nombre de Mouti varie selon les espèces. Les lucioles passent la majeure partie de leur vie à la phase larvaire. Au Québec, hiver sous cette forme.

Pour compléter cette étape, Larva s'installe dans un abri qui a creusé dans le sol. Au printemps, métamorphose dans une nymphe immobile. Après une douzaine de jours, l'insecte divise la cuticule et le bois de chauffage à l'âge adulte émerge. Il reste quelques jours sous terre puis faisant partie d'un partenaire. L'espèce Luciole la plus connue a un cycle de vie de deux ans [46].

3.2.3 Habitat Luciole

Les lucioles ont besoin d'une zone ouverte où les femelles peuvent afficher pour attirer un mâle en Juin, Juillet, et Août. Ils se retirent dans le sol pendant la journée. Ils préfèrent l'herbe ouverte ou les haies aux forêts, mais on les trouve rarement sur des terres qui ont été améliorées pour l'agriculture.

3.2.4 Alimentation vital du Luciole

Les adultes se nourrissent rarement. Malgré sa petite taille, les larves sont des prédateurs féroces. Ils ont erré dans la litière des feuilles à la recherche de petits escargots et des limaces, qu'ils mordent et injectent avec une neurotoxine qui immobilise et liquéfiait leur repas. Ils aspirent ensuite votre proie vide [41].

3.2.5 Comportement de Luciole

La luciole est active la nuit et passe ses journées dans des endroits humides sous les décombres. Les larves sont également nocturnes et sont rarement vues chaque fois que les conditions sont bonnes pour les escargots, généralement entre avril et octobre, ils peuvent être découverts. L'étape adulte, bien qu'elle soit courte, est la plus facile à reconnaître. Ils allument quelques heures à la fois et écoutent généralement brillant peu de temps après l'embrayage. Le meilleur moment pour voir que l'insecte est compris entre 22h00 et la nuit

sèche d'été de minuit (la pluie a tendance à regarder les femmes sur le sol et dans une végétation plus épaisse). [45]

3.3 Luciole artificiel

3.3.1 Présentation

L'algorithme Firefly est un algorithme bio-inspiré, introduite par Dr Xin-She Yan à l'université Cambridge en 2007. L'algorithme est basé sur le principe d'attraction entre les lucioles et simule le comportement d'un essaim de lucioles dans la nature, ce qui lui donne beaucoup de similarités avec d'autres méta heuristiques basées sur l'intelligence collective du groupe, tel que l'algorithme PSO (Particle Swarm Optimisation), l'algorithme d'optimisation par colonies d'abeilles (ABC), et l'algorithme des bactéries de fourrages (BFA) [3, 51]. Selon des bibliographies récentes, les performances de l'algorithme Firefly dans la résolution des problèmes d'optimisation dépassent celles des autres algorithmes, tel que les algorithmes génétiques. Ceci a été justifié par des recherches récentes, où les performances de cet algorithme ont été comparées avec celles de quelques algorithmes connus [52, 53, 54].

L'algorithme prend en considération les trois points suivants [53] [3] :

1. Toutes les lucioles sont unisexes, ce qui fait l'attraction entre celles-ci n'est pas en fonction de leur sexe.
2. L'attraction est proportionnelle à leurs luminosités, donc pour deux lucioles, la moins lumineuse se déplacera vers la plus lumineuse. Si aucune luciole n'est lumineuse qu'une luciole particulière, cette dernière se déplacera aléatoirement.
3. La luminosité des lucioles est déterminée en fonction d'une fonction objective (à optimiser).

En se basant sur ces trois règles, nous pouvons présenter le pseudo-code de l'algorithme Firefly comme suit :

```
Définir une fonction objective f(x), x = (x1, …, xd) T
Générer une population de lucioles xi (i = 1, 2, …, n)
  Définir l'intensité de lumière I à un point xi par la fonction objective f (xi)
  Déterminer le coefficient d'absorption γ
      Tant que (t < Max Génération)
          Pour i = 1 jusqu'à n
            Pour j = 1 jusqu'à n
                Si (Ii < Ij)
                    Déplacer la luciole i vers la luciole j
                Fin Si
             Varier l'attraction en fonction de la distance r via exp[-γr]
          Evaluation des nouvelles solutions et mettre à jour l'intensité de lumière
            Fin Pour j
          Fin Pour i
        Classer les lucioles et trouver la meilleure solution
       Fin Tant que
Afficher les résultats
```

3.3.2 Les Paramètres de l'algorithme

L'algorithme Firefly est formulé avec deux choses importantes : La variation de l'intensité de la lumière et la formulation de l'attraction. Pour simplifier, l'attraction des lucioles est déterminée en fonction de la luminosité, où la luminosité est déterminée avec la fonction objective. Donc nous citons quatre points importants dans l'algorithme de firefly : Intensité de lumière, attractivités, distance et mouvement.

3.3.2.1 Intensité de lumière

Dans le cas d'un problème de minimisation, la luminosité I d'une luciole à une position x peut être définie comme $I(x) \alpha f(x)^{-1}$. Cependant, l'attraction β est relative à la position des autres lucioles. Par conséquent, elle varie en fonction de la distance $r_{.j}$ entre la luciole i et la luciole j. D'un autre côté, l'intensité de la lumière diminue avec la croissance de la distance par rapport à la source. Ce qui fait que l'attraction peut varier selon le degré d'absorption. Pour simplifier, l'intensité de la lumière I(r) varie en fonction de la loi $I(r)=I s/r^2$ où Is est l'intensité à la source. Pour une valeur constante de Y, l'intensité varie en fonction de la distance r, ce qui donne $I =I_0 e^{-\gamma r}$, où I_0 est l'intensité de la lumière de source.

La combine des deux effets de la loi carrée inverse et l'absorption peut être approximer avec la formule Gaussienne suivante [47] :

$$I(r) = I_0 e^{-\gamma r2} \quad \text{(III. 1)}$$

Parfois il est nécessaire d'utiliser une fonction mono tonique décroissante. Dans ce cas, nous pouvons utiliser l'approximation :

$$I(r) = \frac{Io}{(1+\gamma r2)} \quad \text{(III. 2)}$$

3.3.2.2 Attractivities

L'attraction d'une luciole est proportionnelle à l'intensité des lucioles adjacentes, La formule de cette attractivité β d'une luciole peut être définie comme :

$$\beta(r) = \beta o e^{-yr^2} \quad \text{(III. 3)}$$

Où β_0 est l'attraction à r = 0. Pour généraliser, le calcul de β(r) est défini par :

$$\beta(r) = \beta o e^{-yr^m}, (m \geq 1) \quad \text{(III. 4)}$$

3.3.2.3 Distance

La distance entre deux lucioles i et j à des positions x_i et x_j est définie par la distance Cartésienne suivante :

$$r_{ij} = \left|\left|x_i - x_j\right|\right| = \sqrt{\sum_{k=1}^{d}(x_{i,k} - x_{j,k})} \quad \text{(III. 5)}$$

3.3.2.4 Mouvement

Le mouvement d'une luciole i attirée par une autre luciole j (plus lumineuse que i) est déterminé par :

$$x_i = x_i + \beta_0 e^{yr^2 ij}\left(x_i - x_j\right) + \alpha(rand - \tfrac{1}{2}) \quad \text{(III. 6)}$$

Le second terme dans l'équation est dû à l'attraction. Tant dis que le troisième terme rajoute de l'aléatoire à l'équation, où a est aléatoire, rand une fonction de génération de nombre aléatoire uniforme dans l'intervalle [0, 1]. Le paramètre y caractérise la variation de l'attractivité, sa valeur est cruciale dans la détermination de la vitesse de convergence et le comportement de l'algorithme

3.4 Conclusion

Dans ce chapitre, nous avons présenté la vie des lucioles dans la nature, ainsi que l'inspiration de l'algorithme Firefly en expliquant chaque paramètre en détail. Le problème de classification des données médicales peut être considéré comme étant un problème d'optimisation. Pour cela, nous essayons dans ce projet d'adapter la méthode méta-heuristique des Firefly conçu principalement pour résoudre le problème de classification des ensembles de données médicales.

Chapitre 4

Implémentation

4.1 Introduction

Dans ce dernier chapitre, nous allons décrire la phase d'implémentation et les résultats de notre application qui est d'automatiser les données médicales représentées dans une base de données « PID » du diabète par l'algorithme Firefly. Dans un premier temps, nous allons commencer par analyser et visualiser les données du diabète afin de décrire la corrélation en ses attributs, ensuite nous allons présenter les résultats obtenus par l'algorithme Firefly.

4.1 Environnement du travail

4.1.1 Matériel

Pour développer cette application on a utilisé une machine, configurées comme suit : Machine hp sa mémoire vive est de 4 Go, disque dure est de 500 Go et son processeur « Intel(R) Core(TM) i3-3110M CPU @ 2.40GHz 2.40 GHz» avec un système « Windows 10 Professionnel».

4.1.2 Outils de développement

Notre application a été réalisée avec le langage de programmation 'Python', et un outil de mise en œuvre de la base de données 'Fichier.csv'.

4.1.3 Description de langage de programmation

Le langage de programmation Choisit pour le développement de notre système est le Python, et comme éditeur nous avons choisir le JupyterLab.

4.1.3.1 Python

Python est un langage de programmation interprété, orienté objet, de haut niveau avec une sémantique dynamique. Ses structures de données intégrées de haut niveau, combinées à un typage dynamique et à une liaison dynamique, le rendent très attrayant pour le développement rapide d'applications, ainsi que pour une utilisation en tant que langage de script ou de collage pour connecter des composants existants entre eux. La syntaxe simple et facile à apprendre de Python met l'accent sur la lisibilité et réduit donc le coût de maintenance du programme. Python prend en charge les modules et les packages, ce qui encourage la modularité du programme et la réutilisation du code. L'interpréteur Python et la vaste bibliothèque standard sont disponibles sous forme source ou binaire sans frais pour toutes les principales plates-formes et peuvent être distribués gratuitement.

Souvent, les programmeurs tombent amoureux de Python en raison de la productivité accrue qu'il offre. Comme il n'y a pas d'étape de compilation, le cycle édition-test-débogage est incroyablement rapide. Le débogage des programmes Python est simple : un bogue ou une mauvaise entrée ne provoquera jamais un défaut de segmentation. Au lieu de cela, lorsque l'interpréteur découvre une erreur, il lève une exception. Lorsque le programme n'attrape pas l'exception, l'interpréteur imprime une trace de pile. Un débogueur de niveau source permet d'inspecter les variables locales et globales, d'évaluer des expressions arbitraires, de définir des points d'arrêt, de parcourir le code ligne par ligne, etc. Le débogueur est écrit en Python lui-même, témoignant de la puissance introspective de Python [48].

4.1.3.2 JupyterLab

JupyterLab est un environnement de développement interactif basé sur le Web pour les blocs-notes, le code et les données Jupyter. JupyterLab est flexible : configurez et organisez l'interface utilisateur pour prendre en charge un large éventail de flux de travail en science des données, en calcul scientifique et en apprentissage automatique. JupyterLab est extensible et modulaire : écrivez des plugins qui ajoutent de nouveaux composants et s'intègrent à ceux existants [57].

4.2 Description de la base de données utilisée

Dans ce mémoire, nous avons utilisé la base Pima Indian Diabetes (PID), cet ensemble de données provient à l'origine de l'Institut national du diabète et des maladies digestives et rénales. L'objectif de l'ensemble de données est de prédire de manière diagnostique si un patient est diabétique ou non, sur la base de certaines mesures diagnostiques incluses dans l'ensemble de données. Plusieurs contraintes ont été imposées à la sélection de ces instances à partir d'une base de données plus large. En particulier, tous les patients ici sont des femmes d'au moins 21 ans d'origine indienne Pima [49].

La base Pima Indian Diabetes est constituée de 768 cas dont 268 sont diabétiques et 500 non diabétiques. Chaque cas est formé de 9 attributs, dont 8 représentent des facteurs de risque et le 9eme représente la classe du patient, la figure 9 présente un échantillon de la base Pima Indians avec ces attributs :

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	35	168	43.1	2.288	33	1
6	5	116	74	0	0	25.6	0.201	30	0
7	3	78	50	32	88	31	0.248	26	1
8	10	115	0	0	0	35.3	0.134	29	0
9	2	197	70	45	543	30.5	0.158	53	1
10	8	125	96	0	0	0	0.232	54	1
11	4	110	92	0	0	37.6	0.191	30	0
12	10	168	74	0	0	38	0.537	34	1
13	10	139	80	0	0	27.1	1.441	57	0
14	1	189	60	23	846	30.1	0.398	59	1
15	5	166	72	19	175	25.8	0.587	51	1
16	7	100	0	0	0	30	0.484	32	1
17	0	118	84	47	230	45.8	0.551	31	1
18	7	107	74	0	0	29.6	0.254	31	1
19	1	103	30	38	83	43.3	0.183	33	0
20	1	115	70	30	96	34.6	0.529	32	1

diabetes.csv

Figure 9 – Tableau de base de données

L’ensemble de données se compose de plusieurs variables prédictifs médicaux et une variable cible ‘Outcome’ qui définit le type de classe, les variables sont :

Pregnancies : Nombre de fois enceinte
Glucose : Concentration plasmatique de glucose à 2 heures dans un test oral de tolérance au glucose
BloodPressure : Tension artérielle diastolique (mm Hg)
SkinThickness : Épaisseur du pli cutané du triceps (mm)
Insulin : Insuline sérique 2 heures (mu U / ml)
BMI : Indice de masse corporelle (poids en kg / (taille en m) ^ 2)
DiabetesPedigreeFunction : Fonction généalogique du diabète
Age : Années d'âge)
Outcome : Variable de classe (0 ou 1) 268 sur 768 sont 1, les autres sont 0

4.3 Méthodologie de travail

Avant d'appliquer l'algorithme Firefly sur la base de données du diabète, nous devons passer par plusieurs étapes pour la préparation des données dont les plus importantes sont : Collecte de donnée, pré-traitement de données, visualisation des données, entrainement du modèle.

4.3.1 Collecte de donnée

Nous allons d'abord charger nos données et afficher les cinq premières lignes, et voilà le résultat suivant :

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Figure 10 – Cinq premières lignes de la base de données

4.3.2 Prétraitement de données

Tout d'abord, nous allons vérifier s'il n'y a pas de valeurs nulles dans nos données, nous allons donc afficher les colonnes et leurs valeurs nulles et non nulles. La figure suivante affiche les attributs et ses valeurs.

```
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
```

Figure 11 – Les attributs de diabète

Ensuite, nous allons générer des statistiques descriptives pour le jeu de données du diabète comme il est décrit dans la figure 12.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Figure 12 – statistiques descriptives de jeu de données

Après, nous allons traiter les donnée pertinente du jeu de donnée et les analyser de plus près (Hypertension, glucose, Insuline ...), ces valeurs ne peuvent être nulles ou négatives, c'est pour cela qu'on va changer les valeurs à 0 avec le null. Pour qu'on puisse après réaliser le calcul de génération du modèle. Afficher le nombre de valeur Null dans le dataset, et voilà le résultat

```
Pregnancies                 0
Glucose                     5
BloodPressure              35
SkinThickness             227
Insulin                   374
BMI                        11
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Figure 13 – Nombre de valeur Null dans le dataset

4.3.3 Visualisation des données

Nous allons afficher en graphe le jeu de données pour comprendre la distribution des valeurs, le résultat sera visualiser sous forme des différents diagrammes. La figure 14 montre cette distribution des attributs.

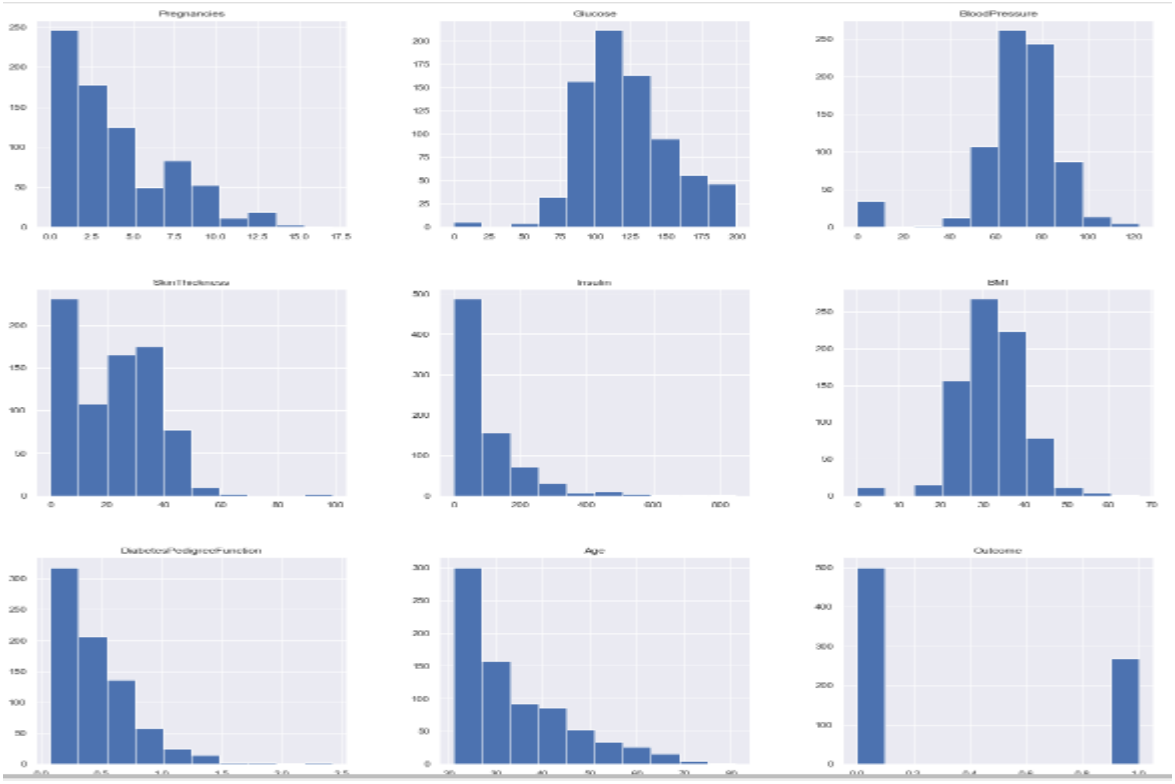

Figure 14 – Distribution des valeurs de jeu de donnée

Nous allons viser à changer les valeurs nan pour les colonnes en fonction de leur distribution

Nous allons Tracer les nouveaux graphes après l'élimination de Nan. Cette distribution représente dans la figure 15.

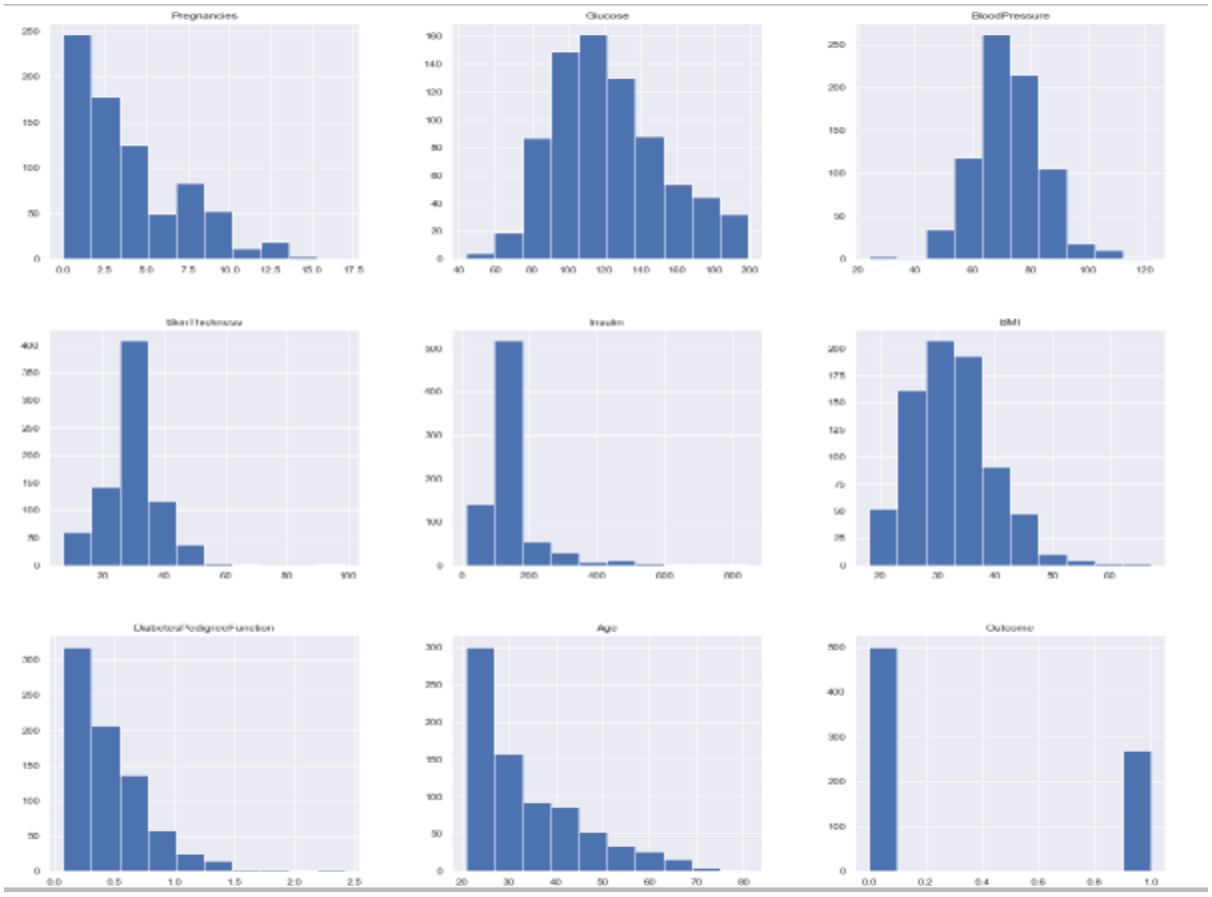

Figure 15 – Distribution des valeurs de jeu de donnée après l'élimination de Nan

Nous allons vérifier l'équilibre des données en traçant le nombre de résultats par leur valeur la figure suivante montre les classes de type d'analyse :

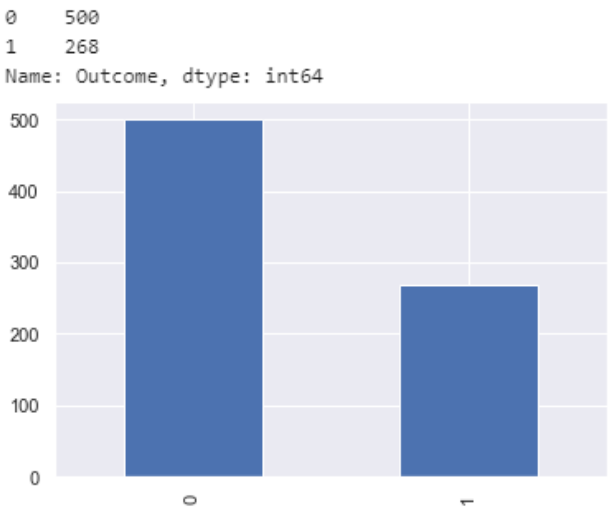

Figure 16 – Graphe de l'attribut 'outcome'

La matrice de corrélation suivante décrit le lien entre les différents attributs :

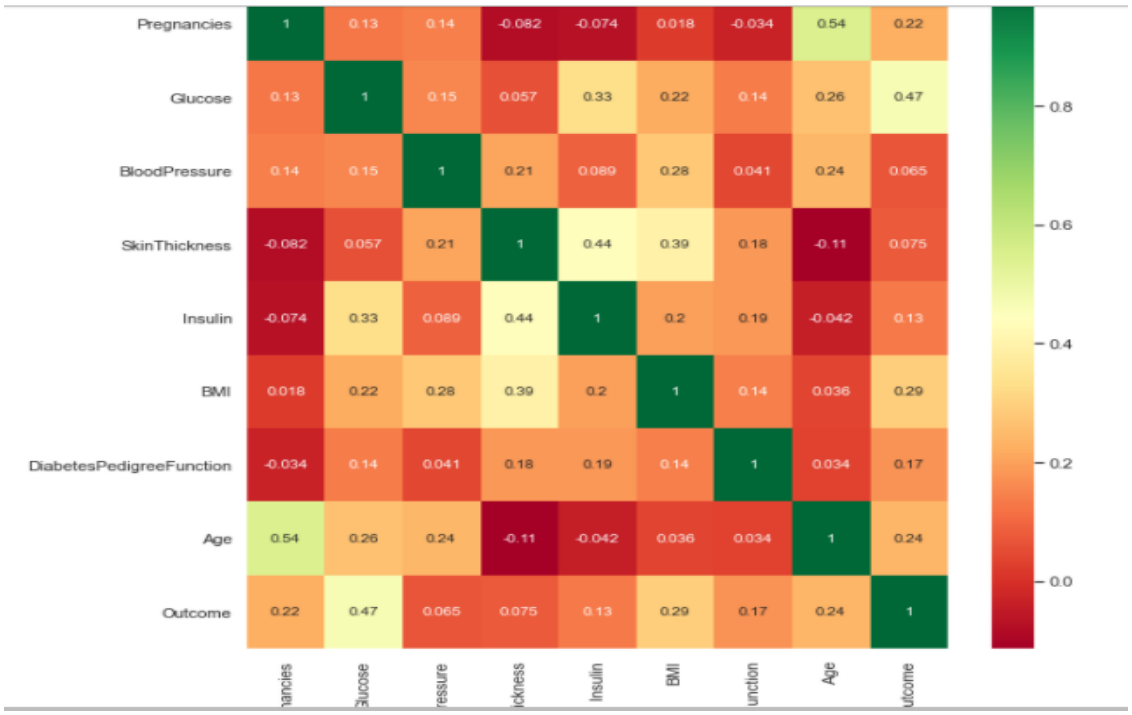

Figure 17 – Relation entre les attributs

4.3.4 Entrainement du modèle

Mettre à l'échelle du dataset, objectif est de changer les valeurs des dates pour faciliter et réduire le temps de calcul.

Nous allons afficher les cinq lignes du nouveau jeu de données, et voilà le résultat suivant :

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	0.639947	0.865108	-0.033518	0.670643	-0.181541	0.166619	0.468492	1.425995
1	-0.844885	-1.206162	-0.529859	-0.012301	-0.181541	-0.852200	-0.365061	-0.190672
2	1.233880	2.015813	-0.695306	-0.012301	-0.181541	-1.332500	0.604397	-0.105584
3	-0.844885	-1.074652	-0.529859	-0.695245	-0.540642	-0.633881	-0.920763	-1.041549
4	-1.141852	0.503458	-2.680669	0.670643	0.316566	1.549303	5.484909	-0.020496

Figure 18 – Cinq lignes du nouveau jeu de données

4.4 Résultat et discussion

Dans le tableau 4, nous décrivons les différents paramètres que nous avons utilisés pour obtenir nos résultats :

Tableau 4 – Paramètre de l'algorithme Firefly

Paramètre	Description	valeur
N	Nombre de luciole	768
β 0	Attractivité	1.0
γ	Coefficient d'absorption lumineuse	0.5

Maintenant que l'ensemble de données a été séparé en données d'entraînement et le reste pour le test, nous avons implémenté l'algorithme Firefly sur la base PID, et nous avons obtenu les best fireflies comme un résultat initial. Cette figure représente un exemple de meilleur fireflies obtenu après plusieurs génération :

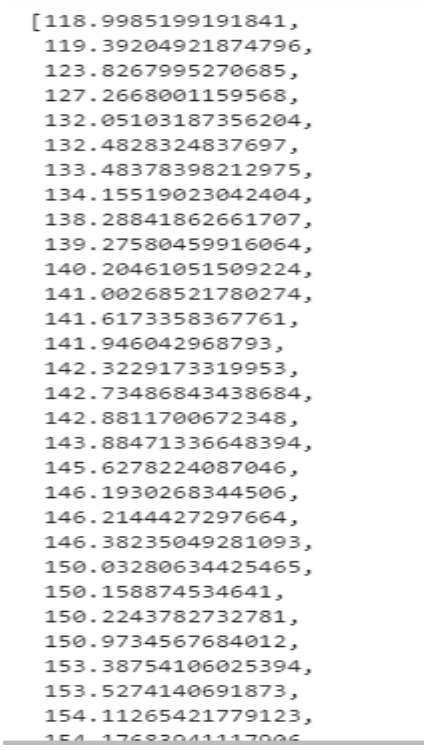

Figure 19 – Les bests fireflies

Et comme un résultat final retourne le minimum fireflie

```
best_firefly
```

```
118.9985199191841
```

Figure 20 – Best fireflie

La figure suivante représente les différentes valeurs d'attributs de champ sélectionnée

```
Pregnancies                 4.000
Glucose                    97.000
BloodPressure              60.000
SkinThickness              23.000
Insulin                   125.000
BMI                        28.200
DiabetesPedigreeFunction    0.443
Age                        22.000
Outcome                     0.000
Name: 118, dtype: float64
```

Figure 21 – Les attribut de best fireflies

Dans l'étape suivante, nous allons faire une hybridation de notre algorithme avec l'algorithme des k -plus proches voisins afin d'améliorer la taux de précision. Finalement, nous avons obtenu les résultats montrés dans la figure 22 :

Figure 22 – Graphe de résultat

Cette figure représente les deux courbes l'une pour l'entrainement et l'autre pour le test.

La figure suivante représente le taux de précision :

Figure 23 – Taux de précision

Dans nos expérimentations, nous appliquons l'algorithme Firefly avec ses paramètres proposées de cet algorithme sur la base PID, Dans chaque essai, nous appliquons une approche de l'algorithme et nous varions les indices de validité afin de comparer les résultats. Le résultat atteint un taux de précision de 76%. Avec cette précision obtenue, le programme est capable de distinguer correctement un patient diabétique d'un patient non diabétique.

4.5 Conclusion

Dans ce chapitre, nous avons appliqué l'algorithme Firefly sur la base du diabète. Nous prévoyons d'implémenter une deuxième méthode pour comparer les résultats avec ceux de l'algorithme Firefly, puis d'hybrider les deux méthodes pour améliorer la précision. Mais faute de temps, nous n'avons pas pu le faire. Cela reste un de nos futurs projets.

Conclusion Générale

Dieu a créé la nature et en a fait une source d'inspiration pour de nouvelles méthodes d'analyse. Parmi ces méthodes nous avons la méthode bio-inspiré qui est une approche qui conduit à s'inspirer de la nature pour développer de nouveaux systèmes

Il a été prouvé que les méthodes bio-inspiré résolvent efficacement les problèmes Divers complexes dans de multiples domaines de recherches surtout dans le domaine de la médecine.

Dans ce mémoire nous avons vue c'est quoi les technique de classification, ses types et ses avantages comme première chapitre, dans le deuxième chapitre nous avons fait un aspect générale sur les méthodes bio-inspiré, ses classifications et nous avons défini quelque algorithme bio-inspiré et un état de l'art de quelque travaux de développeurs, et dans le troisième chapitre nous avons étudié la théorique de la méthode appliquée ‘Firefly’ sur le diabète qui est une maladie chronique très grave et progressive caractérisée par un dysfonctionnement du système de régulation de la glycémie. Dans le dernier chapitre, nous avons analysé les différentes données du diabète dans la première étape. La deuxième se focalise sur le processus d’apprentissage par l’algorithme d’optimisation bio-inspiré firefly afin de distinguer les patients malades de patient en bonne santé.

Bibliographie

Livre, monographie

[2] X.-S. Yang, “Firefly Algorithm, Levy Flights and Global Optimization,” Research and Development in Intelligent Systems XXVI (Eds M. Bramer, R. Ellis, Petridis), Springer London, 2010, pp. 209-218.

[3] Yang, X. S., “Firefly Algorithm, Stochastic Test Functions and Design Optimisation”, Int. J. Bio-Inspired Computation, Vol. 2, No. 2, 2010, pp.78—84

[9] D.E Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Reading MA Addison Wesley, 1989.

[17] Hélène Horsin Molinaro : Bio-inspiration, la Nature comme modèle, 2017.

[53] Yang, X. S., “Firefly Algorithm, Stochastic Test Functions and Design Optimisation” , Int. J. Bio-Inspired Computation, Vol. 2, No. 2, 2010, pp.78—84

[54] X. S. Yang, “Firefly algorithms formultimodal optimization,” in Proceedings of the Stochastic Algorithms: Foundations and Applications (SAGA â€™09), vol. 5792 of Lecture Notes in Computing Sciences, pp. 178-178, Springer, Sapporo, Japan, October 2009.

[55] Xin-She Yang. Firefly algorithm for multimodal optimization. Stochastic Algorithms: Foundation and Application, 5th, 2009.

[56] X.S.YANG: “Firefly Algorithms for Multimodal Optimization, Stochastic Algorithms: Foundations and Applications”, SAGA 2009, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 5792:169-178.(2009).

[47] X.S. YANG " Nature-Inspired Metaheuristic Algorithms ". Luniver Press, UK.(2008)

Article d'actes de conférence

[1] Hanaa Hachimi. Hybridations d'algorithmes métaheuristiques en optimisation globale et leurs applications. Autre [cond-mat.other]. INSA de Rouen, 2013. Français. ffNNT: 2013ISAM0017ff. fftel-00905604ff

Articles de revue

[7] M.F. Ganji and M.S. Abadeh. A fuzzy classification system based on ant colony optimization for diabetes disease diagnosis. Expert System with Application 38, pp 14650–14659, 2011.

[10] A. Colorni, M. Dorigo et V. Maniezzo, Distributed Optimization by Ant Colonies, actes de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing, 134-142, 1991.

[14] Kennedy J., Eberhart R. C., Particle swarm optimization , proceeding of the IEEE conference on neural networks, IV, Piscataway, NJ, pp. 1942-1948, 1995.

[15] Zhan, Z. H., Zhang J. Li. Y., Chung H. H., Adaptive Particle Swarm Optimization, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 39, issue 6, pp. 1362– 1381, 2009.

[18] Sharma, M., Gupta, S., Sharma, P. and Gupta, D. (2019) 'Bio-inspired algorithms for diagnosis of breast cancer', Int. J. Innovative Computing and Applications, Vol. 10, Nos. 3/4, pp.164–174.

[41] Tweit, Susan J., juillet/août 1999. Danse des Lucioles. *Audubon*, 101 (4): 26-31.

[42] 1998. *The 1998 World Book Encyclopedia, International Edition*. World Book, Inc.

[43] Borror, Donald J., W. 1970. *Peterson Field Guides: Insectes*. Boston: Houghton Mifflin Company.

[44] Alliston, Mark, 22 juillet 1998. « Glow Worm Lampyris noctiluca » (en ligne). Consulté le 25 avril 2021 à http://www.the-timeless-dimension.com/in008.htm.

[46] espacepourlavie.ca/insectes-arthropodes/lucioles. Consulté le 29 avril 2021

[52] GARETH REES, The Remote Sensing Data Book , Cambridge University Press 1999

Ouvrage colectif

[13] K. Balasubramani And K. Marcus, *A Comprehensive review of Artificial Bee Colony Algorithm*, International Journal of Computers & Technology Volume 5, No. 1, May - June, 2013.

[35] Barga, R., Fontama, V., Tok, W.H. and Cabrera-Cordon, L., 2015. Predictive analytics with Microsoft Azure machine learning. Apress.

[51] S. Lukasik and S. Zak, "Firefly algorithm for con-tinuous constrained optimization tasks" in Proceedings of the International Conference on Computer and Computational Intelligence (ICCCI "09), N. T. Nguyen, R. Kowalczyk, and S.-M. Chen, Eds., vol. 5796 of LNAI, pp. 97â€"106, Springer, Wroclaw, Poland, October 2009.

Rapport Technique

[12] D. Karaboga, *An idea based on honey bee swarm for numerical optimization*, Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[4] El Dor, A. (2012). Perfectionnement des algorithmes d'optimisation par essaim particulaire : applications en segmentation d'images et en électronique. PhD thesis, Université Paris-Est.

[8] John Holland. Outline for a logical theory of adaptive systems. Journal of the Association of Computing Machinery, 3, 1962

[11] M. Dorigo, Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italie, 1992.

[22] Bouras salima «utilisation d'une méthode bio-inspirée pour l'analyse des données médicales» Mémoire de Master

[23] Marref Nadia «Apprentissage Incrémental & Machines à Vecteurs Supports» Mémoire de Magister, Batna : 2015.

[24] Melle. BENYETTOU Assia « Contribution en apprentissage semi-supervisé sous contexte multi-label » Mémoire de doctorat, Oran 2018

[36] Mahmoud Elbattah, Owen Molloy, National University of Ireland Galwaym. elbattah1@nuigalway.ie, owen.molloy@nuigalway.ie « Clustering-Aided Approach for Predicting Patient Outcomes with Application to Elderly Healthcare in Ireland »

[25] cours «Apprentissage automatique» Julien Ah-Pine (julien.ah-pine@univ-lyon2.fr), Université Lyon 2,M2 DM 2019/2020, Date de consultation: 11/03/2021

[27] experiences.microsoft.fr/business/intelligence-artificielle-ia-business/apprentissage-supervise-et-non-supervise-quelles-differences

[32] Wickham, H., 2009. ggplot2: elegant graphics for data analysis. Springer Science & Business Media.

[33] Sylvain Arlot. Classification supervisée: des algorithmes et leur ca libration automatique, 2009. www.di.ens.fr/~éarlot / enseigner / 2009Centrale / cours

[34] Matteo Matteucci. Un tutoriel sur les algorithmes de dépoussiérage. accueil deib.polimi.it/Matteucci/Clustering/tutoriel html

[37] Jain, A.K., 2010. Data clustering: 50 years beyond K-means.

Documents web

[2] https://datafranca.org/wiki/Bio-inspiration

[5] https://www.google.com/url?sa=i&url=https%3A%2F%2Fdocplayer.fr%2F107119697-Contributions-a-la-resolution-du-probleme-de-routage-dans-les-reseaux-mobiles-ad-hoc-par-les-methodes-bio-inspirees-akram-kout.

[19] E. M. Mashhour, E. M. F. El Houby, K. T. Wassif et al., A Novel Classifier based on Firefly Algorithm, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2018.11.009

[16] https://docs.microsoft.com/fr-fr/archive/msdn-magazine/2015/june/test-run-firefly-algorithm-optimization

[20] https://www.google.com/search?q=algorithme+g%C3%A9n%C3%A9tique&sxsrf=ALeKk03H5R-T2C80vgcYeRPq-xuwMqD7iQ:1615729433836&source=lnms&tbm=isch#imgrc=lQlsK3GLaTzglM

[21] https://www.google.com/imgres?imgurl=x-raw-image%3A%2F%2F%2Fc1701f75bc31907711f68e41b46b1a05d4a8b920055e4ca667775521730e3562&imgrefurl=https%3A%2F%2Fhal.archives-ouvertes.fr%2Fhal-01260694%2Fdocument&tbnid=d5AxvHUfcMFC8M&vet=12ahUKEwib9OiG-a_vAhUVRhoKHRWZAa4QMygvegUIARD1AQ..i&docid=smrmUOvtnUUjkM&w=490&h=232&q=particules%20d%E2%80%98un%20essaim&ved=2ahUKEwib9OiG-a_vAhUVRhoKHRWZAa4QMygvegUIARD1AQ

[26] https://pageperso.univ-lr.fr, Arnaud Revel (revel.arnaud@gmail.com) cour « Apprentissage Semi-Supervisé » Date de consultation: 13/03/2021

[28] https://www.math.ens.fr/cours-apprentissage/Obozinski/Cours1.pdf

[29] www.math.univ-toulouse.fr, cours «apprentissage Supervise», Date de consultation: 11/03/2021

[30] www.math.univ-toulouse.fr/~besse/Wikistat/

[31] http://cedric.cnam.fr/~saporta/DM.pdf

[38] NOCA (2012) IHFD. Available at: https://www.noca.ie/irish-hip-fracture-database (consulter: 16 Décembre 2016).

[39] www.animaldiversity.org. Consulté le 29 avril 2021

[40] www.catalogueoflife.org. Consulté le 29 avril 2021

[45] 1993. « Glow Worm Lampyris noctiluca » (En ligne). Consulté le 29 avril 2021 à http://www.bracknellforest.gov.uk/council/departments/leisure/countryside/bap/glow_worm.htm

[48] https://www.python.org/doc/essays/blurb/

[49] https://www.kaggle.com/uciml/pima-indians-diabetes-database

[57] https://jupyter.org/