

N°d'ordre

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE MINISTERE DE
L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE DJILLALI LIABES SIDI BEL
ABBES
FACULTE DES SCIENCES EXACTES
DEPARTEMENT D'INFORMATIQUE

THESE DE DOCTORAT EN SCIENCE

SPÉCIALITÉ : INFORMATIQUE

Présentée et soutenue par :

BIDI NORIA

Approches Méta-Heuristiques pour les Tâches de Classification

Soutenue le -- --

Devant le jury Composé de :

Dr. TOUMOUH ADIL	Maître de conférences rang A	U.SBA	(Président)
Pr. BELALEM GHANEM	Professeur	U.ORAN	(Examineur)
Dr. BERRABAH DJAMEL	Maître de conférences rang A	U.SBA	(Examineur)
Dr. DEBBAT FATIMA	Maître de conférences rang A	U.Mascara	(Examinatrice)
Dr. SALEM MOHAMMED	Maître de conférences rang A	U.Mascara	(Examineur)
Pr. ELBERRICHI ZAKARIA	Professeur	U.SBA	(Directeur de thèse)

Année Universitaire : 2017-2018

Résumé

Dans l'exploration de données, de nombreuses techniques sont utilisées pour extraire des informations utiles. L'une de ces techniques est la classification, la classification de volume massif de données est une tâche complexe en raison entre autres la présence d'attributs redondants et non pertinents. Pour obtenir les meilleurs modèles de classification, la sélection d'attributs est devenue une étape essentielle du prétraitement des données.

Dans cette thèse, nous présentons deux nouveaux algorithmes adaptatifs appelés FS-PeSOA (feature selection penguins search optimization algorithm) et FS-SLOA (feature selection seven spot ladybird optimization algorithm) qui sont des méthodes de sélection méta-heuristiques. Ils seront combinés avec différents classificateurs pour trouver les meilleures attributs, qui atteignent la plus grande précision dans la classification. Ensuite, nous étudions et comparons l'hybridation de plusieurs méthodes de filtres et de méta-heuristiques, y compris la sélection d'attributs basée sur les corrélations (CFS), l'analyse de composantes principales (PCA) et les statistiques X2 (CHI) comme méthodes de filtres et l'algorithme génétique (GA), FS-PeSOA et FS-SLOA comme méthodes méta-heuristiques. Pour l'évaluation de chaque approche, nous explorons l'utilisation de la validation croisée 5 fois, de la validation croisée 10 fois et la division simple de données (90% pour les données d'apprentissage et 10% pour les données de test).

Enfin, nous présentons l'algorithme SLOA pour découvrir les règles de classification, cet algorithme suit l'approche du Michigan. Nos approches proposées ont été expérimentées sur des ensembles de données de référence (Wisconsin Breast Cancer, Diabète Pima, Mammographie Mass, Dermatologie, Colon Tumor et Cancer de la Prostate). Les résultats expérimentaux prouvent que les précisions de classification sont puissantes pour les différents ensembles de données.

Mots-clés : Exploration de Données, Classification Supervisé, Sélection des Attributs, les Méta-heuristiques, KNN, SVM, NB.

ملخص

في التنقيب عن البيانات ، تستخدم العديد من التقنيات لاستخراج معلومات مفيدة. واحدة من هذه التقنيات هي التصنيف. إن تصنيف الحجم الهائل للبيانات مهمة معقدة نظرا لوجود سمات زائدة وغير ذات صلة. للحصول على أفضل نماذج التصنيف، أصبح اختيار ميزة خطوة أساسية في المعالجة المسبقة للبيانات.

في هذه الأطروحة، نقدم أولا خوارزميتين لاستدلال تسميان (FS-PeSOA) و (FS-SLOA)، وسوف تكون مقترنة مع مصنفاة مختلفة للعثور على أفضل الميزات و تحقيق أعلى دقة في التصنيف. بعد ذلك، نقوم بالتحقيق والمقارنة بين تهجين عدة طرق التصفية لاختيار الميزات و خوارزميات الاستدلال، بما في ذلك الارتباطات القائمة على اختيار ميزة (CFS) ، وتحليل المكون الرئيسي (CFS) والإحصاءات (X2) (CHI) كجزء من طرق التصفية والخوارزمية الجينية (GA)، (FS-PeSOA) و (FS-SLOA)، كجزء من الخوارزميات الاستدلالية. ولتقييم كل طريقة، نستخدم التحقق من صحة ه أضعاف، والتحقق من ١٠ أضعاف والتقسيم البسيط للبيانات.

وأخيرا، نقدم خوارزمية (SLOA) لاكتشاف قواعد التصنيف، وهذه الخوارزمية تتبع نهج ميشيفان. وقد تم تجربة الخوارزميات المقترحة على مجموعات البيانات القياسية المعروفة (ويسكونسن سرطان الثدي، بيما السكري، ماس الثدييات، الأمراض الجلدية، ورم الكولون ومجموعات بيانات سرطان البروستاتا). النتائج التجريبية تثبت أن دقة التصنيف قوية لمجموعات البيانات المختلفة.

الكلمات المفتاحية : استخراج المعرفة ، التنقيب عن البيانات ، التصنيف ، سرطان الثدي، السكري، الأمراض الجلدية، ورم الكولون و سرطان البروستاتا.

Abstract

In data mining, many techniques are used to extract useful information. One of these techniques is the classification, the classification of the massive volume of data is a complex task due to the presence of redundant and irrelevant features. To obtain the best classification models, feature selection became an essential data pre-processing step.

In this thesis, we first present two new adaptive algorithms called FS-PeSOA (Feature Selection Penguins Search Optimization Algorithm) and FS-SLOA (Feature Selection Seven Spot Ladybird Optimization Algorithm) which are meta-heuristics feature selection methods, they will be combined with different classifiers to find the best subset features, which achieve the highest accuracy in classification. Then, we investigate and compare the hybridization of several filters and meta-heuristics methods, including correlations based feature selection (CFS), principal component analysis (PCA) and the X2 statistics (CHI) as part of filters methods and the genetic algorithm (GA), FS-PeSOA and FS-SLOA as part for meta-heuristics methods. For the evaluation of each approach, we explore the use of 5-fold cross validation, 10-fold cross validation and simple split data (90% for train data and 10% for test data).

Finally, we present SLOA algorithm to discover classification rules, this algorithm follows Michigan's approach. Our proposed approaches have been experimented on well known benchmark datasets (Wisconsin Breast Cancer, Pima Diabetes, Mammographic Mass, Dermatology, Colon Tumor and Prostate Cancer data sets). Experimental results prove that the classification accuracies are powerful for different data sets.

Keywords: Data Mining, Classification, Feature Selection, Meta-heuristics, Penguins Search Optimization Algorithm, Seven Spot Ladybird Optimization Algorithm, KNN, SVM, NB.

Remerciements

Je remercie Dieu, le Tout Puissant.

En second lieu, je tiens à remercier mon directeur de thèse, **Pr. Zakaria Elberrichi**, pour son encadrement continu, son suivi, ses orientations, et ses efforts pour l'aboutissement de ce travail. Je le remercie également pour la confiance qu'il m'a toujours témoignée et ses précieux conseils. Ma gratitude pour sa patience, sa qualité humaine, son comportement envers moi, son hospitalité, et son soutien tout au long de ces cinq dernières années. J'adresse mes remerciements à Messieurs Dr. Toumouh Adil Pr. Belalem Ghanem, Dr. Berrabah Djamel, Dr. Debbat Fatima et Dr. Salem Mohammed de m'avoir fait l'honneur d'examiner ce travail de thèse et d'avoir accepté de faire partie des membres du jury.

J'exprime mes remerciements aux personnes qui sont les plus proches et les plus chères. À mes parents pour lesquels je suis éternellement reconnaissant pour m'avoir constamment soutenu, pour avoir cru en moi pour mener à bien ce travail, pour leurs prières et leurs encouragements. Merci infiniment !

Un remerciement particulier à mon mari Mouedden Abdelkader de m'avoir soutenu ces derniers mois, à mes frères Mohammed, Lakhal, Sadik et Benali pour leurs conseils, orientations et encouragements.

J'adresse aussi mes remerciements les plus sincères à ma nièce Assmaa Hansali, à ma belle sœur Djabare Naziha à mon amie Zoubida Siahmed pour leur aides pour l'achèvement de mon travail et aussi pour leur encouragements.

Et enfin, merci à tous ceux qui ont participé à l'aboutissement de ce travail.

Je dédie ce travail à mes parents,

À mes frères et mes sœurs,

À mon mari,

À mes enfants.

Tables des matières

Table des figures	6
Liste des tableaux	8
Introduction générale	6
1 Notions fondamentales	7
Introduction	7
1.1 Extraction de connaissances à partir de données (ECD)	8
1.1.1 Définition	8
1.1.2 Processus ECD	8
1.1.3 La fouille de données (Data Mining)	10
1.2 La classification	16
1.2.1 Principe de la classification	16
1.2.2 Classification supervisée	18
1.2.3 Fonctionnement de la classification supervisée	18
1.2.4 Les méthodes de la classification supervisée	19
1.3 La sélection d'attributs	24
1.3.1 Définitions	25
1.3.2 Le processus de la sélection d'attributs	26
1.3.3 Les techniques de sélection d'attributs:	33
Conclusion	35

2 États de l'art	36
Introduction	36
2.1 Les méta-heuristiques	37
2.1.1 Définition	37
2.1.2 Les enjeux des méta-heuristiques	39
2.1.3 Le fonctionnement de méta-heuristique	40
2.1.4 La classification des méta-heuristiques	41
2.2 Les méta-heuristiques pour la sélection d'attributs	50
2.3 Les méta-heuristiques pour la classification	62
Conclusion	69
3 Approches proposées pour la sélection d'attributs et la classification	70
Introduction	71
3.1 Les approches proposées pour la sélection d'attributs	71
3.1.1 L'approche FS-PeSOA	71
3.1.2 L'approche FS-SLOA	78
3.1.3 Les approches hybrides	85
3.2 L'approche Proposée pour la classification	88
3.2.1 Classificateur basée sur SLOA	88
Conclusion	93
4 Résultats et expérimentations	94
Introduction	94
4.1 Critères d'évaluations	94
4.2 Données de l'UCI	95
4.3 Le matériel utilisé	97
4.4 Les approches proposées pour la sélection d'attributs	97
4.4.1 L'approche FS-PeSOA	97
4.4.2 L'approche FS-SLOA	102
4.4.3 Les approches hybrides	107
4.5 L'approches proposée pour la classification	114

4.5.1 Résultats et discussion	114
Conclusion	116
Conclusion générale	118

Table des figures

1.1	Le processus d'extraction des connaissances à partir de données	10
1.2	Le processus de la sélection d'attributs	27
3.1	Les pingouins	72
3.2	L'organigramme générale de l'algorithme FS-PeSOA	76
3.3	Les coccinelles à sept points	78
3.4	La représentation d'une solution	81
3.5	Les méthodes métaheuristiques/filtres	87
3.6	Les méthodes filtres/métaheuristiques	88

Liste des tableaux

2.1 Synthèse de méthode de la sélection d'attributs	61
3.1 Les paramètres de FS-PeSOA	75
3.2 Les paramètres de FS-SLOA	83
4.1 Les caractéristiques de l'ensemble de données utilisés	96
4.2 Les paramètres de FS-PeSOA	98
4.3 Les paramètres de GA	98
4.4 Les accuracies obtenus pour différent datasets	99
4.5 Les f-measures obtenus pour différent datasets	99
4.6 Une comparaison des résultats obtenus par notre méthode FS-PeSOA et d'autres méthodes de la littérature dans différents datasets	100
4.7 Les paramètres de FS-SLOA	102
4.8 Les paramètres de l'algorithme génétique	102
4.9 Les accuracies obtenues pour différents datasets	103
4.10 Les f-measures obtenues pour différents datasets	104
4.11 Une comparaison des résultats obtenus par notre méthode FS-SLOA et d'autres méthodes de la littérature dans différents datasets	105
4.12 L'accuracy et le nombre d'attributs obtenus pour dermatologie dataset . .	110
4.13 L'accuracy et le nombre d'attributs obtenus pour mammographic dataset .	111
4.14 L'accuracy et le nombre d'attributs obtenus pour WDBC dataset	112
4.15 L'accuracy et le nombre d'attributs obtenus pour pima diabetes dataset .	113

4.16 Les Accuracies obtenues pour différents datasets	114
4.17 Une comparaison des résultats obtenus par notre méthode SLOA pour la classification et d'autres méthodes de la littérature dans différents datasets	115

Introduction générale

Ces dernières années, le volume de données, de toutes sortes, croît d'une manière extrêmement rapide. La nécessité d'extraction des informations et de connaissances utiles à partir de ce volume génère un nouveau champ de recherche qui a pour objectif l'analyse de ces grandes quantités de données afin de trouver des modèles intelligibles à partir de ces données. L'extraction de connaissance à partir de données est un domaine interdisciplinaire qui se trouve au confluent de la statistique, de l'intelligence artificielle notamment de l'apprentissage automatique, de l'analyse des données et des interfaces de communication homme-machine.

L'extraction de connaissance à partir de données est un processus qui se compose de plusieurs étapes, la fouille de données est la plus importante étape de ce processus. Elle tire ces techniques et tâches de plusieurs autres domaines. Dans notre travail on s'intéresse particulièrement à la tâche de la classification supervisée.

Motivation

Face à la numérisation croissante de nos activités, l'hétérogénéité des supports de stockage, la croissance en puissance des machines capables de supporter de gros volumes de données avec un coût moins chères, nous permet de stocker toutes les données possibles liées au problème en question et l'exécution du processus intensif d'exploration, les données collectées et stockées dans des bases de données sont énormes et se trouvent partout dans les entreprises, les établissements de recherche, etc.

Surtout que les humains ne sont pas les seuls producteurs de données ; les machines aussi y contribuent avec leurs cartes sim, leurs capteurs, etc.

Cette masse importante de données de type satellitaires, simulations scientifiques, multidimensionnelles, génomiques (micro-arrays, . . .) a motivé le développement de l'extraction de connaissances à partir de données (ECD). Ce développement a généré une demande pressante d'outils permettant de transformer les données en connaissances. Ces données qui sont inexploitable par les méthodes d'analyse classiques et qui ont besoin de traitement en temps réel pour prendre des décisions stratégiques efficaces, améliorer la productivité, exploiter le vécu (données historiques) pour prédire le futur et faire émerger des régularités ou des motifs qui permettent d'ajouter de l'intelligence aux archives de données afin d'être capable par la suite de décider.

Les chercheurs s'intéressent aux nombreux travaux depuis bien longtemps notamment à partir des domaines tels que les statistiques, la reconnaissance de formes et l'intelligence artificielle. De nouvelles méthodes d'extraction automatique de connaissances, de nouvelles approches, de nouveaux savoirs et de nouvelles manières de penser et de travailler sont apparues.

Dans le processus d'extraction de connaissance à partir de données, la fouille de données est la plus importante étape qui se base sur plusieurs tâches telles que la classification, les règles d'association et de regroupement, etc. La classification est un sujet de recherche très actif car plusieurs problèmes peuvent être exprimés sous la forme de problèmes de classification. Les chercheurs tentent de trouver d'autres modèles pour mener à bien plusieurs applications importantes dans notre vie comme les applications de diagnostic médicale, de la détection des fraudes et de détection biométrique, etc. Leurs motivations sont d'une part de tester de nouveaux algorithmes sur le problème de la classification et de connaître leurs apports, et d'autre part, de proposer de nouvelles sources d'inspiration.

Problématique

La tâche la plus populaire de la fouille de données est la classification car plusieurs problèmes peuvent être exprimés sous forme de problèmes de classification. En effet, il faut pouvoir associer à chaque élément sa classe correspondante. Ce type de problème se résout généralement après une étape primordiale qui est la sélection d'attributs, cette étape consiste à choisir parmi un ensemble d'attributs de grande taille un sous ensemble d'attributs intéressants ou attributs représentatifs qui permettront de bien identifier les différentes classes.

Lors de la conception d'un classificateur, on utilise un ensemble d'exemples et de contre exemples. A l'aide de ces exemples, le classificateur va mettre à jour ses paramètres afin de pouvoir reconnaître au mieux les classes d'exemples qu'on lui fournit ainsi de généraliser et de classer d'autres exemples que ceux fournis pour l'apprentissage. Lorsque l'on fait de l'apprentissage on parle de la classification supervisée. Il existe de nombreux classificateurs mais tous ont un point commun, ils ont besoin de s'adapter au problème auquel ils sont confrontés. En effet, il faut qu'ils puissent déterminer quelles sont les classes des différents sous ensembles d'attributs et comment les différencier.

Dans le domaine médical, de nombreuses méthodes de classification et de sélection d'attributs ont été utilisées, mais il n'y a pas vraiment de méthode qui semble être meilleure que les autres. Il y a de nombreuses méthodes qui se sont révélées très efficaces sur certains problèmes et non pour d'autres. Donc il est nécessaire de tester de nouveaux algorithmes et de proposer de nouvelles sources d'inspiration sur le problème de la classification.

La problématique de cette thèse est de voir quel est l'apport des méthodes métaheuristiques dans l'amélioration de la tâche de la classification supervisée des données et leurs application à la sélection du modèle, plus particulièrement la sélection d'un sous ensemble d'attributs pertinents.

Contribution

Au cours de ce travail, nous proposons des méthodes méta-heuristiques pour l'amélioration des tâches de la fouille de données plus précisément la classification supervisée et la sélection des sous ensembles d'attributs des données.

Le choix de ces outils se justifie par le fait que les méthodes exactes qui montrent leur efficacité sur des problèmes particuliers ou lors de considération de critères spécifiques, nous pensons qu'elles ne permettent pas de traiter les problèmes où une recherche exacte à un coût exponentiel en temps de calcul et en espace mémoire. En effet, cela fait partie des problèmes NP-difficiles qui peuvent être résolus par les méta-heuristiques.

Les méta-heuristiques sont des algorithmes d'optimisation applicables à une grande variété de problèmes. Elles sont généralement des méthodes stochastiques itératives inspirées de la nature dans le but de résoudre au mieux des problèmes d'optimisation pour lesquels il n'existe pas des méthodes classiques plus efficaces. L'application des méta-heuristiques en classification a connu une évolution particulière ces dernières années grâce aux avancées technologiques.

Dans cette thèse, notre contribution se décompose en deux parties. La première partie se concentre sur la sélection de sous-ensembles d'attributs. En premier lieu, nous proposons deux méthodes méta-heuristiques pour la sélection de sous ensembles d'attributs. L'une appelée FS-PeSOA (Feature Selection Penguins Search Optimization Algorithm) qui se base sur le comportement de chasse chez les pingouins et l'autre appelée FS-SLOA (Feature Selection Seven Spot Ladybird Optimization Algorithm) qui se base sur le comportement de fourragement chez les coccinelles à sept points. En deuxième lieu, nous proposons deux approches hybrides (filtre/méta-heuristique) et (méta-heuristique/filtre) en utilisant des méthodes filtres (CFS (Correlation based Feature Subset), chisquare et PCA (Principal Component Analysis)) et des méthodes méta-heuristiques (Fs-PeSOA, Fs-GA et Fs-SLOA). Nous avons choisi d'évaluer les performances en termes de précision du classificateur (accuracy) construit à l'aide des bases de données obtenus du

centre d'apprentissage automatique de système intelligent de L'Université de Californie à Irvine (UCI). Dans la deuxième partie, nous nous intéressons à la classification par une approche méta-heuristique SLOA. Ainsi, cette thèse comporte une comparaison générale et un état de l'art pour chacun de ces points particulièrement développé. Ces travaux ont été validés par la publication d'un ensemble d'articles.

Organisation de la thèse

Cette thèse est composée de quatre chapitres dont nous présentons une brève description dans les paragraphes suivants :

Dans le premier chapitre, nous commençons par présenter brièvement le processus d'extraction de connaissances à partir de données (ECD) ainsi que ses principales composantes. Ensuite, nous détaillons la fouille de données, ses techniques pour l'extraction de connaissances, ses différents types d'utilisation et ses tâches en se concentrant sur la classification supervisée et la sélection d'attributs, ses étapes, ainsi que leurs différentes méthodes.

Dans le deuxième chapitre, nous présenterons les méta-heuristiques, ses enjeux, ainsi que la classification de ces différentes approches. Ensuite nous citerons les différentes méthodes proposées dans la littérature qui sont implémentées pour la sélection des attributs et la classification.

Le troisième chapitre sera consacré à la résolution de problème de sélection des attributs et le problème de la classification par de nouvelles méthodes méta-heuristiques hybrides.

Dans Le quatrième chapitre, nous exposerons les résultats expérimentaux obtenus à partir de l'implémentation de différentes méthodes, ainsi que des résultats expérimentaux comparés aux autres travaux de l'état de l'art.

Enfin, on termine par une conclusion générale et quelques perspectives.

Publications internationales :

Bidi N., Elberrichi Z. (2017). Using Penguins Search Optimization Algorithm for Best Features Selection for Biomedical Data Classification. International Journal of Organizational and Collective Intelligence. Volume 7. Issue 4.

Bidi N., Elberrichi Z. (2018). Best Features Selection For Biomedical Data Classification Using Seven Spot Ladybird Optimization Algorithm. International Journal of Applied Metaheuristic Computing (IJAMC). Volume 9, Issue 3, Article 4.

Communications internationales :

Bidi, N. et Elberrichi, Z. (2016) Feature selection for text classification using genetic algorithms. In : 8th International Conference on Modelling, Identification and Control (ICMIC), pp. 806–810. IEEE Press, Algiers .

Bidi N. et Fizazi H.(2013). A Combined Appmach Using Meta-heuristic and Mathematical Operators for Images Segmentation. In 11th International Symposium on Programming and Systems (ISPS) Page(s) :122 - 127. IEEE Press, Algiers .

Publication en cours d'évaluation :

Bidi N., Elberrichi Z., “Combining Filter Methods and Metaheuristics for Efficient Feature Selection : A Biomedical Classification Perspective” International Journal of Natural Computing Research (IJNCR), submitted June 2017.

Bidi N. et Elberrichi Z.“ Combining Metaheuristics Methods and Filter for Feature Selection”. International Journal of Metaheuristics (IJMHeur), submitted february 2018.

Communications internationales en cours d'évaluation :

Bidi N., Elberrichi Z., “Seven Spot Ladybird Optimization Algorithm : An adapted and efficient metaheuristic for medical data classification”. in International Symposium on Combinatorial Optimization 2018. (ISCO2018), submitted january 2018.

Notions fondamentales

Introduction

L'homme et les machines n'ont pas cessé de créer des données et de les stocker. Pour mieux utiliser ces vastes bases de données dans tous les domaines, nous avons besoin d'outils permettant de transformer les données en connaissances ou de découvrir des connaissances à partir de ces bases de données. Ce défi a engendré le développement d'un nouveau axe de recherche : l'extraction de connaissances à partir des bases de données (ECD) ou *knowledge discovery from data bases* (KDD) en Anglais.

Le présent chapitre est divisé en trois parties, la première partie est consacrée à l'extraction de connaissances à partir des bases de données (ECD), nous donnons un aperçu général sur le processus ECD (définition, étapes. . .) et nous insistons sur une étape importante qui est la fouille de données (définition, tâche, techniques utilisées. . .). Dans la deuxième partie, nous focalisons sur une tâche de cette étape de fouille de données, qui est la tâche de la classification supervisée (définition, fonctionnement, méthodes utilisées...). La troisième partie sera consacrée à une étape de prétraitement qui joue un rôle important dans la fouille de données qui est la sélection d'attributs (définition, processus. . .). Enfin nous terminons par une conclusion.

1.1 Extraction de connaissances à partir de données (ECD)

1.1.1 Définition

l'extraction de connaissance à partir de données est un domaine très vaste. Il est défini selon Fayyaad [1] comme un processus non trivial, interactif et itératif qui permet d'identifier des modèles valides, nouveaux, potentiellement utiles et compréhensibles à partir de bases de données massives. Le terme processus signifie que le ECD se compose de plusieurs étapes, le terme non trivial réfère au fait que, l'extraction des connaissances n'est pas aussi simple que le calcul d'une moyenne ou de la variance d'un ensemble de nombres [2]. Le terme interactif signifie que l'homme est au cœur du processus de l'ECD et le terme itératif réfère au fait que, l'ECD n'est pas un processus linéaire où chaque étape est appliquée une seule fois, il offre la possibilité de retourner à n'importe quelle étape pour aboutir à la fin avec le modèle de connaissance recherché. L'objectif d'ECD est de trouver des modèles intelligibles à partir de données, ces modèles doivent être compréhensibles, utiles et ayant un sens pour le preneur de décision.

1.1.2 Processus ECD

Le processus d'extraction des connaissances à partir de données (ECD) présenté dans [Figure 1.1](#) se décompose en plusieurs étapes. Ces étapes peuvent être regroupées en cinq phases majeures selon [3] :

1.1.2.1 La phase de compréhension du domaine étudié

Appelé aussi (intégration et collecte de données), cette étape consiste à regrouper un ensemble de données relatives au système cible de l'étude. Ces données qui sont le plus souvent hétérogènes doivent permettre de décrire au mieux le problème traité. Cette première phase se concentre sur la collecte des données nécessaires pour mener à bien un processus ECD. Lors de cette phase, l'utilisateur ne sait pas encore si les données qu'il a réunies seront toutes adaptées à son problème ni si ces données seront suffisantes.

1.1.2.2 La phase de prétraitement

Au cours de cette phase, trois étapes sont effectuées (sélection, nettoyage et transformation). La sélection d'attributs est une étape de prétraitement qui consiste à sélectionner, dans l'entrepôt, les données qui seront retenues pour construire le modèle. Le nettoyage de données consiste à gérer la qualité des données et plus particulièrement les imprécisions et les incertitudes qu'elles peuvent contenir. La transformation consiste à préparer les données brutes et les convertir en données appropriées. Elle structure les données dans un formalisme attendu par les algorithmes qui seront appliqués par la suite. Cette phase a une place importante car elle permet d'améliorer l'efficacité du processus et de donner une plus grande lisibilité du problème à étudier.

1.1.2.3 La phase de fouille de données

La fouille de données est le cœur du processus ECD. Cette phase intègre le choix de la méthode d'apprentissage qui va être employée et son paramétrage. Ces choix doivent tenir compte des contraintes liées au domaine étudié ainsi que des connaissances que les experts du domaine peuvent nous fournir.

1.1.2.4 La phase de post-traitement

Dans cette phase les résultats obtenus par la phase précédente doivent être évalués par les différents critères d'évaluation pour mesurer la qualité des modèles construits.

1.1.2.5 La phase d'interprétation et d'exploitation des résultats

L'interprétation concerne l'analyse des résultats qui sont sous forme de modèles ou de règles permet d'obtenir des connaissances par un expert du domaine assisté par le développeur. Celui-ci pourra décider d'arrêter le processus d'ECD, ou au contraire de modifier les opérations effectuées lors des étapes précédentes et de relancer le processus. Pour l'exploitation, les résultats qui sont sous forme de modèles ou de règles permettent d'obtenir des connaissances. Ce sont ces connaissances qui seront fournies à l'utilisateur.

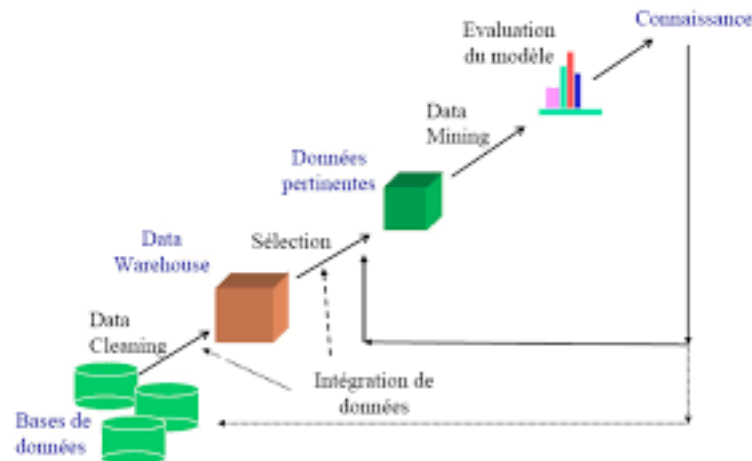


Figure 1.1 : Le processus d'extraction des connaissances à partir de données

1.1.3 La fouille de données (Data Mining)

Le terme fouille de données (Data Mining) est utilisé pour décrire l'une des étapes du processus d'ECD (Knowledge Discovery in Databases) mais pour de nombreux chercheurs, ce terme est utilisé comme un synonyme de l'ECD. Pour cette raison, nous citons plusieurs définitions de la fouille de données avec quelques références :

Selon le Groupe Gartner [4], le Data Mining appelé aussi fouille de données est le processus de découverte de nouvelles corrélations, modèles et tendances en analysant une grande quantité de données, en utilisant les technologies de reconnaissance des formes ainsi que d'autres techniques statistiques et mathématiques.

Le Data Mining est un domaine interdisciplinaire utilisant en même temps des techniques d'apprentissage automatique, de reconnaissance des formes, des statistiques, des bases de données et de visualisation pour déterminer les manières d'extraction des informations à partir de très grandes bases de données [3].

La découverte d'informations originales, auparavant inconnues et potentiellement utiles, à partir des données [5]. L'exploration et l'analyse, par des moyens automatiques ou semi automatiques, d'un large volume de données afin de découvrir des tendances ou des règles [6].

1.1.3.1 La Catégorisation des systèmes de la fouille de données

Les systèmes de fouille de données ont différentes vues, donc ils peuvent être catégorisés selon plusieurs critères tel que le type de données à manipuler (spatiales, séries temporelles, textuelles, etc), les modèles de base de données à fouiller (base de données relationnelles, base de données orientées objets, les data warehouses, etc), les techniques utilisées (les réseaux neurones, les algorithmes génétiques, les statistiques, la visualisation, etc) et les types de connaissances à découvrir.

1.1.3.2 Les méthodes de la fouille de données

Selon [7], les méthodes de la fouille de données peuvent être classées en deux grandes familles : méthodes prédictives et méthodes descriptives.

Les méthodes descriptives : Les méthodes descriptives sont des méthodes non supervisées qui permettent de décrire de manière concise les données en caractérisant les propriétés intéressantes et mettent l'accent sur la compréhension et l'interprétation de ces dernières. En d'autres termes, c'est un processus dans lequel l'apprenant reçoit des exemples d'apprentissage qui ne comprenant que des données d'entrées dont le but de produire des modèles expliquant les données sous étude en trouvant les tendances et les corrélations qui résument les relations entre ces données.

Les méthodes prédictives : Les méthodes prédictives sont des méthodes supervisées qui utilisent la variable cible contrairement aux méthodes non supervisées. Comme leur nom l'indique, ces méthodes permettent d'exploiter les données disponibles dans la base de données avec des résultats connus pour créer un modèle permettant de prédire les valeurs des autres données. En d'autres termes, c'est un processus dans lequel l'apprenant reçoit des exemples d'apprentissage comprenant à la fois des données d'entrée et de sortie dont le but est de prédire correctement la valeur future des variables étudiées.

1.1.3.3 Les tâches de fouille de données

En entreprise, il existe plusieurs difficultés ou besoins de type intellectuels, économiques ou même commerciaux qui peuvent être exprimés comme des tâches de fouille de données. On va citer ces tâches en leur donnant des descriptions générales afin de lever toute ambiguïté sur ces tâches qui paraît similaires :

- Classification
- Estimation
- Prédiction
- Groupement par similitudes
- Segmentation (ou clusterisation)
- Description

La classification : Selon Berry « La classification consiste à examiner des caractéristiques d'un élément nouvellement présenté afin de l'affecter à une classe d'un ensemble prédéfini. » [8]. Donc, la classification consiste à étudier les caractéristiques d'un nouvel élément pour lui attribuer une classe prédéfinie. Les éléments à classifiés sont généralement des enregistrements. La classe est un champ particulier à valeurs discrètes. La tâche de classification consiste à mettre à jour les enregistrements d'une base de données en déterminant un champ de classe. Les classes résultantes de la classification regroupent les objets ayant des caractéristiques similaires et séparent les objets ayant des caractéristiques différentes. L'objectif est de créer un modèle qui peut être appliqué aux données non classifiées dans le but de les classifiées [9].

L'estimation : L'estimation ressemble beaucoup à la classification à part que la variable de sortie est numérique plutôt que catégorique comme dans le cas de la classification. L'estimation permet d'obtenir une variable de sortie de type continue en utilisant plusieurs fonctions combinant les données en entrée. Cette tâche consiste à compléter une valeur manquante dans un champ particulier en fonction d'autres champs de

l'enregistrement. La relation entre ce champ particulier et les autres champs de l'enregistrement vont fournir un modèle d'estimation. Ce modèle sera appliqué pour estimer d'autres cas.

L'estimation est souvent utilisée dans différents domaines de recherche et de commerce. Par exemple en marketing, pour proposer des offres aux meilleurs clients potentiels. Les techniques les plus appropriées à l'estimation sont : le réseau de neurones et la régression linéaire simple ou multiple.

La prédiction : La prédiction ressemble à la classification et à l'estimation, à part que, dans la prédiction les enregistrements sont classés suivant des critères (ou des valeurs) prédites (estimées) et dans une échelle temporelle différente. Les techniques les plus appropriées à la prédiction sont : le raisonnement basé sur la mémoire, les arbres de décision et les réseaux de neurones.

Le regroupement par similitudes : Le regroupement par similitudes consiste à déterminer et grouper les éléments qui vont naturellement ensemble afin de définir des règles d'association. Les règles d'associations sont de la forme "Si antécédent, alors conséquent". La technique la plus appropriée au regroupement par similitudes est l'analyse du panier de la ménagère.

La segmentation : La segmentation a plusieurs synonymes tels que le partitionnement, la catégorisation et la classification non supervisée. Cette tâche se base sur un apprentissage non supervisée, elle ne dispose d'aucune information au préalable que la description des variables. Pour la segmentation, il n'y a pas de classe à expliquer ou de valeur à prédire définie a priori, il s'agit de regrouper par données ressemblantes les données non étiquetées dont on dispose, donc créer un cluster.

Un cluster est une collection d'enregistrements similaires l'un à l'autre, et différents à ceux existants sur les autres clusters. La différence entre la segmentation et les différentes tâches est que la tâche de segmentation ne classe pas, n'estime pas, ne prévoit

pas la valeur d'une variable sortantes. Au lieu de cela, les algorithmes de clustering visent à segmenter la totalité de données en des sous groupes relativement homogènes. Ils maximisent l'homogénéité à l'intérieur de chaque groupe et la minimisent entre ces derniers [10].

Les algorithmes de clustering visent à segmenter la totalité de données en des sous groupes relativement homogènes. La technique la plus appropriée à la clusterisation est l'analyse des clusters

La description : La description est la première tâche pour différents outils de Data Mining. Elle permet de décrire les données d'une base complexe en expliquant les relations existantes dans ces données pour essayer de découvrir et de comprendre les individus, les produits et les processus qui sont à l'origine de ces données. Une bonne explication des données engendre une meilleure exploitation de ces dernières.

1.1.3.4 Les techniques de la fouille de données

Il existe de nombreuses techniques qui sont utilisées pour les tâches de fouille de données, chacune de ces techniques possède sa propre utilité et rendement. Ces techniques proviennent de diverses disciplines telles que l'informatique, l'intelligence artificielle, les statistiques, etc.

Nous citons à titre d'exemple les techniques les plus utilisées.

- Les réseaux de neurones.
- Les arbres de décisions.
- Le raisonnement basé sur la mémoire.
- Les réseaux Bayésiens.
- Les règles d'association.
- La méthode du plus proche voisin.

1.1.3.5 Les domaines d'application

Aujourd'hui, la fouille de données est utilisée dans un important nombre de domaines d'activités grâce aux possibilités qu'elle offre afin d'optimiser la gestion des ressources (humaines et matérielles). Nous citons ci dessous quelques domaines et quelques applications de la fouille de données dans ces domaines :

La fouille de données dans le secteur bancaire et financière :

- L'attribution de crédit bancaire.
- Détection d'usage frauduleux de cartes bancaires.
- Découverte de relations cachées entre les indicateurs financiers.
- Etc.

La fouille de données dans la bio-informatique et la biotechnologie :

- La prédiction les structures de différentes protéines.
- la détermination de la complexité des structures de plusieurs médicaments.
- Le diagnostic médical.
- Prédiction de la réponse d'un patient à un traitement.
- Étude des corrélations entre le dosage dans un traitement et l'apparition d'effets secondaires.
- Etc.

La fouille de données dans le marketing, vente et distribution :

- Détection d'associations de comportements d'achat.
- Découverte de caractéristiques de clientèle.
- Prédiction de probabilité de réponse aux campagnes de mailing.
- Etc.

La fouille de données dans le secteur des assurances et santé :

- Découverte d'associations des demandes de remboursements.
- Identification des nouvelles polices d'assurances pour conserver les clients actuels et acquérir de nouveaux clients.
- Détection d'association de comportements pour la découverte de clients à risque.
- Etc.

La fouille de données dans la télécommunication :

- Analyse des achats de services de télécommunications.
- Prédiction de modèles d'appels téléphoniques.
- Gestion des ressources et de trafic réseau.
- Etc.

1.2 La classification

1.2.1 Principe de la classification

La classification est une tâche de fouille de données qui intervient dans plusieurs domaines tels que la reconnaissance des formes, la reconnaissance de gènes, les statistiques, l'intelligence artificielle, l'aide multicritère à la décision, l'attribution de crédit bancaire, la prédiction des sites archéologiques, le diagnostic médical et la détection de fraudes fiscales, etc. Donc plusieurs définitions ont été proposées par les spécialistes de ces domaines :

Définition 1.2.1 Selon Mari [11] "Effectuer une classification, c'est mettre en évidence des relations entre des objets, et leurs paramètres".

Définition 1.2.2 selon Henriot [12] "La classification consiste à affecter des objets, des candidats et des actions potentielles à des catégories ou des classes prédéfinies".

Définition 1.2.3 *"Le processus de classification cherche à mettre en évidence les dépendances implicites qui existent entre les objets, les classes entre elles, les classes et les instances. La classification recouvre les processus de reconnaissance de la classe d'un objet, et l'insertion éventuelle d'une classe dans une hiérarchie. Ce mode de raisonnement permet de reconnaître un objet en identifiant ses caractéristiques, relativement à la hiérarchie étudiée. La classification fait intervenir un processus de décision d'appartenance"* [13].

Une classe est créée à partir d'un ensemble d'objets (attributs, caractéristiques ou critères) qui sont semblables entre eux et qui sont dissemblables à ceux d'autres classes. Les classes sont construites de façon à maximiser les similarités des objets qui sont dans la même classe et à minimiser les similarités de ces objets avec ceux des autres classes.

Un « classificateur » est un algorithme qui, à partir d'un ensemble d'exemples, produit une prédiction de la classe de toute donnée [14].

Définition 1.2.4 *"La classification est l'action de regrouper en différentes catégories des objets ayant certains points communs ou faisant partie d'un même concept, sans avoir connaissance de la forme ni de la nature des classes au préalable, on parle alors de problème d'apprentissage non supervisé ou de classification automatique, ou l'action d'affecter des objets à des classes prédéfinies, on parle dans ce cas d'apprentissage supervisé ou de problème d'affectation"* [15].

On déduit de cette dernière définition que la classification peut désigner deux approches distinctes : la classification supervisée et la classification non-supervisée.

Dans la classification non supervisée, on trouve un ensemble de problématiques où l'espace de classes n'est pas spécifié à l'avance, on ne tient compte que de la similarité des objets sans se préoccuper de leur classe d'appartenance respective. Il s'agit d'identifier et de construire un système de classes sur la base d'observations dans l'espace des caractéristiques.

Les méthodes de la classification non supervisée ont pour but de constituer des groupes d'attributs en fonction des données observées, sans connaissance a priori. En revanche dans la classification supervisée, on tient compte de la classe d'appartenance de chacun des objets et les classes sont spécifiées à l'avance. Les méthodes supervisées utilisent la connaissance a priori sur l'appartenance d'un d'attribut à une classe pour construire un système de reconnaissance de ces classes.

1.2.2 Classification supervisée

La classification supervisée est l'utilisation d'un ensemble d'exemples pour prédire la classe de nouvelles données. Elle consiste à inférer, à partir d'un échantillon de données classées, une procédure de classification qui permet de prédire l'appartenance d'un nouvel exemple à une classe. Cette procédure de classification est obtenue à partir des recherches qui se font sur un espace de modèles basés sur des hypothèses probabilistes, des notions de proximité, des recherches de structure, etc.

1.2.3 Fonctionnement de la classification supervisée

Le fonctionnement de la classification supervisée se décompose en deux points :

- La phase d'apprentissage (modèle d'apprentissage) c'est tout ce qui est appris par l'algorithme de classification est représenté sous la forme des règles de classification. Ces règles permettent d'associer les objets aux classes de références connues. L'algorithme de classification apprend à partir du jeu d'apprentissage et construit le modèle.

Pour l'apprentissage on distingue deux raisonnements : Le premier est le raisonnement inductif, il part du particulier vers le général. Il consiste à considérer un ensemble maximal de règles de classification, puis il réduit cet ensemble de règles à un sous-ensemble qui le résume au mieux. Le deuxième raisonnement est déductif, du général au particulier, il consiste aussi à construire les règles une à une jusqu'à obtenir une bonne description de l'ensemble d'apprentissage.

- La phase de la classification proprement dite, dans laquelle les données de tests vont être utilisées pour estimer la précision des règles de classification générées pendant la première phase. Si la précision du modèle est considérée comme acceptable, la règle pourra être appliquée à de nouvelles données. Le modèle d'apprentissage construit est employé pour classer de nouveaux objets.

1.2.4 Les méthodes de la classification supervisée

1.2.4.1 Classifieur bayésien naïf

Le classificateur bayésien naïf est une méthode qui trouve son fondement théorique dans le théorème de Bayes [16]. Cette technique qui permet de calculer les probabilités conditionnelles fournit une approche simple, avec une sémantique claire de représenter, d'utiliser, et d'apprendre des connaissances probabilistes.

On obtient cette probabilité par la multiplication de la probabilité d'obtenir cette classe (parmi les échantillons d'apprentissage) avec la probabilité d'obtenir chaque caractéristique pour cette classe, selon la valeur moyenne et la variance préalablement calculée. La classe qui obtient la plus grande probabilité en fonction des caractéristiques de l'objet présenté est choisie en tant que classe prédite pour l'objet.

Ce classificateur permet une très bonne résolution des problèmes basés sur les liens. Leur fonctionnement est basé sur la théorie des graphes. Un réseau Bayésien est un graphe orienté dans lequel les nœuds représentent les variables et dans lequel les arêtes symbolisent les dépendances entre les variables. Il mesure la probabilité d'apparition d'un événement connaissant le résultat observé sur d'autres variables.

1.2.4.2 Réseaux de Neurones

Les réseaux de neurones sont des techniques d'intelligence artificielle qui sont apparues dans les années 1950 [17] ou le fonctionnement est inspiré du fonctionnement des neurones biologique. Les réseaux de neurones sont généralement optimisés par des mé-

thodes d'apprentissage de type statistique grâce à leur capacité de classification et de généralisation.

Un réseau de neurones est constitué d'un graphe pondéré orienté dont les nœuds symbolisent les neurones. Chaque neurone fait une somme pondérée de ses entrées (ou synapses) et retourne une valeur en fonction de sa fonction d'activation (fonction signe, fonction sigmoïde, ...). Cette valeur peut être utilisée soit comme une des entrées d'une nouvelle couche de neurones, soit comme un résultat qu'il appartient à l'utilisateur d'interpréter (classe, résultat d'un calcul, etc). Les liens synaptiques sont les connexions entre les neurones, ces liens propagent l'activité de neurones avec une pondération caractéristique de la connexion (poids synaptique).

L'interconnexion entre les neurones permet un calcul global complexe qui en classification se traduit par des frontières de décision aux formes complexes. La phase d'entraînement du réseau consiste à régler les poids synaptiques grâce à l'ensemble d'apprentissage.

Un réseau de neurones présente l'avantage d'être un classificateur très précis (si bien paramétré). Il offre la possibilité de faire le parallélisme et de résister aux pannes. L'inconvénient du réseau de neurones est la complexité de détermination de l'architecture du réseau et de leur paramètres. Il existe plusieurs types de réseaux de neurones, la plus courante est le perceptron multicouche.

1.2.4.3 Séparateurs à Vaste Marge (SVM)

Le SVM est introduit par [18], son principe de base est de trouver l'hyperplan optimal (frontière de décision) qui sépare deux classes dans l'espace de description. Cet espace est celui qui maximise la distance entre les deux classes (la marge).

Le SVM utilise des fonctions de noyau qui, dans un espace augmenté, permettent une séparation optimale des points en différentes catégories. SVM repose sur deux notions :

la notion de marge maximale et la notion de fonction noyau. Pour la marge maximale, on cherche l'hyperplan qui sépare les exemples positifs et négatifs, en garantissant que la distance entre la frontière de séparation et les échantillons les plus proches (marge) soit maximale.

Afin de pouvoir traiter des cas où les données ne sont pas linéairement séparables, la deuxième idée clé de SVM est de transformer l'espace de représentation des données d'entrées en un espace de plus grande dimension appelée espace de caractéristiques (vraisemblablement de dimension infinie), dans lequel il est probable qu'il existe une séparatrice linéaire. Ceci est réalisé grâce à une fonction noyau, qui doit respecter certaines conditions, et qui a l'avantage de ne pas nécessiter la connaissance explicite de la transformation à appliquer pour le changement d'espace.

Les fonctions noyau permettent de transformer un produit scalaire dans un espace de grande dimension, ce qui est coûteux, en une simple évaluation ponctuelle d'une fonction.

Le SVM présente l'avantage d'offrir des décisions rapides et de bonne qualité même pour un traitement de données à grandes dimensions. L'un des inconvénients de SVM est qu'il utilise une classification binaire d'où la nécessité d'utiliser l'approche un-contre-un, en un temps de calcul élevé.

1.2.4.4 K-Plus Proche Voisin (K-PPV)

Le K-plus proche voisin ou K-Nearest Neighbor (K-NN) est une méthode basée sur la notion de proximité (voisinage) entre les variables et sur le raisonnement à partir des cas similaires pour prendre une décision. C'est l'échantillon d'apprentissage, associé à une fonction de distance et une fonction de choix de la classe en fonction des classes des voisins les plus proches, qui constitue le modèle.

Pour prédire la classe d'un échantillon, l'algorithme cherche les k plus proches voisins

de ce nouveau cas et prédit la réponse la plus fréquente de ces k plus proches voisins [19]. Le principe de décision consiste tout simplement donc à calculer la distance de cet échantillon à tous les échantillons fournis.

Un objet est classifié selon un vote majoritaire par ses voisins, l'objet obtient la classe qui est la plus commune chez ses K plus proche voisins dans l'espace des caractéristiques. k doit donc un entier positif, généralement petit. On choisit souvent un k impair pour éviter l'égalité dans le vote.

La distance utilisée pour le calcul de la proximité des voisins est le plus souvent la distance euclidienne. Les exemples d'apprentissage sont des vecteurs dans un espace multidimensionnel. La phase d'apprentissage consiste simplement à conserver ces données dans un format qui permettra le calcul efficace des distances et la recherche des voisins. Les performances de la méthode dépendent du nombre de voisins (le nombre k), du choix de la distance (la fonction de similarité), et du mode de combinaison des réponses des voisins.

1.2.4.5 Les arbres de décision

Un arbre de décision [20], [21] est la représentation graphique d'une procédure de classification. Il permet de modéliser simplement, graphiquement et rapidement un phénomène mesuré plus ou moins complexe. Les arbres de décisions représentent graphiquement un ensemble de règles facilement interprétables. Bien sûr, souvent la taille de l'arbre (plusieurs centaines de nœuds) empêche de saisir la procédure de classification dans son ensemble mais en partant de la description d'un élément, sa classification est toujours compréhensible.

L'induction avec des arbres de décision est l'une des formes d'algorithme d'apprentissage les plus simples et pourtant les plus efficaces [22]. Dans l'apprentissage des arbres de décisions, on attribue la classe majoritaire sauf dans le cas où l'on utilise des fonctions de coût ou de risque. Les méthodes vont différer par les choix effectués pour ces

différents opérateurs, c'est-à-dire sur le choix d'un test (par exemple, utilisation du gain et de la fonction entropie) et le critère d'arrêt (quand on arrête la croissance de l'arbre, soit quand on décide si un nœud est terminal). Avec un tel algorithme, on peut calculer un arbre de décision dont l'erreur apparente est faible, voire nulle.

Un arbre de décision parfait est un arbre de décision tel que tous les exemples de l'ensemble d'apprentissage soient correctement classifiés. Les arbres de décisions complexes peuvent avoir des taux d'erreur très élevés à cause de sur-apprentissage qui peut se produire lorsque l'ensemble d'apprentissage contient des données bruitées ou il ne contient pas certains exemples importants, ou encore lorsque les exemples sont trop spécifiques. L'élagage est l'une des solutions pour réduire ces taux d'erreurs en simplifiant l'arbre par suppression de quelques branches.

1.2.4.6 Bagging

La méthode du Bagging a été introduite par [23]. Le mot Bagging est la contraction des mots Bootstrap et Aggregating. Le principe du bagging est d'entraîner un algorithme d'apprentissage sur plusieurs bases d'apprentissage obtenues par tirage avec remise (ou bootstrap) de m exemples d'apprentissage dans l'échantillon d'apprentissage S . Pour chaque tirage b , une hypothèse h_b est obtenue. L'hypothèse finale est basée sur les moyennes des hypothèses obtenues. Son avantage est qu'on améliore la performance des classificateurs instables en calculant la moyenne de leurs réponses.

Ainsi, si les hypothèses h_b calculées pour chaque tirage b ont une variance importante, alors l'hypothèse finale aura une variance réduite. Un classificateur est dit instable si un petit changement dans les données d'apprentissage provoque un gros changement dans le comportement du classificateur. Le but du bagging est d'atténuer l'instabilité inhérente à certains classificateurs.

1.2.4.7 Boosting

Le boosting est une méthode d'agrégation de classificateurs faibles pour obtenir un classificateur performant [24]. Son principe consiste à assigner un poids égal à chaque exemple d'apprentissage ($1/n$) (appelé pondération) où n est le nombre d'échantillons. Ce poids indique la difficulté de prédire la classe de cet exemple.

L'algorithme s'exécute T fois construisant T classificateurs sur les exemples d'apprentissage pondérés. Chaque fois qu'on produit un classificateur on change les poids des nouveaux exemples utilisés pour le classificateur suivant, on augmente le poids de ceux dont la classe est mal prédite et on diminue le poids des autres.

Ensuite, on calcule l'erreur (e) du modèle sur l'ensemble pondéré. Si e est égale à zéro ou e est supérieure à 0.5 on termine la construction du classificateur. L'idée est de forcer le nouveau classificateur à diminuer l'erreur attendue. Le classificateur final se forme en utilisant un schéma de vote.

1.3 La sélection d'attributs

La sélection d'attributs (en anglais feature subset selection ou FSS) est un sujet de recherche très actif dans différents domaines où on trouve des applications qui nécessitent de traiter des données décrites par un très grand nombre d'attributs tel que l'apprentissage artificiel, du traitement d'images, la fouille de données, etc.

Dans le domaine de la fouille de données et plus précisément pour la tâche de la classification supervisée, la sélection d'attributs est réalisée dans le but d'améliorer la qualité de la solution, la visibilité et l'interprétation du résultat, de réduire les temps d'apprentissage et d'exécution, en conséquence l'apprentissage est moins coûteux pour éviter le sur-apprentissage et de diminuer la complexité des modèles tout en conservant leur efficacité afin de mener à bien la tâche pour laquelle il a été conçu.

Dans [25], l'auteur affirme que le pouvoir de classification vient souvent d'un choix réfléchi et judicieux des attributs fondés sur une bonne connaissance du domaine, plutôt que de la conception spécifique de l'algorithme de séparation. Il est donc intéressant d'extraire ou de sélectionner les attributs importants à partir des exemples. Nous donnons un ensemble de définitions permettant de présenter la sélection d'attributs et les notions nécessaires à la compréhension de cette tâche.

1.3.1 Définitions

Le problème de la sélection d'attributs est défini comme suit : étant donné un ensemble d'attributs, choisir un sous ensemble de taille m qui conduit à la plus petite erreur de classification [26].

La sélection consiste à déterminer les attributs les plus pertinents pour la tâche de classification, et l'extraction consiste à construire de nouveaux attributs par combinaison optimale des attributs initiaux [27].

La sélection de variables est un procédé permettant de choisir un sous ensemble optimal de variables pertinentes, à partir d'un ensemble original des variables, selon un certain critère de performance [28].

Selon Kohavi [29] Soit un inducteur I , et un ensemble de données d'entraînement D avec les attributs A_1, A_2, \dots, A_N . Un sous ensemble d'attributs optimal, A_{opt} est le sous ensemble qui maximise la qualité du classificateur induit $Cl = I(D)$, mesurée par le pourcentage de prédiction correcte sur un ensemble test. Un sous ensemble d'attributs optimal n'est pas nécessairement unique puisqu'il est possible d'avoir la même exactitude de classification en utilisant différents sous ensembles de données.

Il existe plusieurs définitions de la pertinence d'un attribut dans la littérature. Celles-ci dépendent de la nature des données, de l'existence de bruit dans les données et des données dupliquées. La définition suivante donnée par Almuallin et Diettrich en 1991 considère que tous les attributs sont des booléens [30].

Définition 1.3.1 *Un attribut A_i est dit pertinent pour un concept C si A_i apparaît dans chaque formule booléenne qui représente C . Il est dit non pertinent sinon.*

Définition 1.3.2 *Un attribut A_i est pertinent, s'il existe deux instances $(X_1, Y_1)(X_2, Y_2)$ appartenant à deux classes différentes (y_1, y_2) telles que : la valeur de l'attribut A_i est différente pour les deux instances et les valeurs des autres attributs sont identiques.*

Dash et Liu [31] définissent un attribut non-pertinent comme celui n'affectant pas la structure fondamentale des données et un attribut redondant comme celui n'apportant rien de nouveau pour décrire la structure fondamentale des données.

Gennari, Langley et Fisher [32] considèrent des données bruitées et pouvant être dupliquées. Ils définissent les attributs pertinents comme ceux dont les valeurs changent systématiquement en fonction de l'appartenance de la donnée à telle ou telle catégorie.

1.3.2 Le processus de la sélection d'attributs

Selon [33], le processus de la sélection d'attributs comporte en général quatre étapes basiques (voir [Figure 1.2](#)) :

- La procédure de génération de sous-ensemble candidat potentiel.
- Le choix de stratégie d'évaluation des sous-ensembles d'attributs.
- Le critère d'arrêt de la recherche.
- La procédure de validation des sous ensembles.

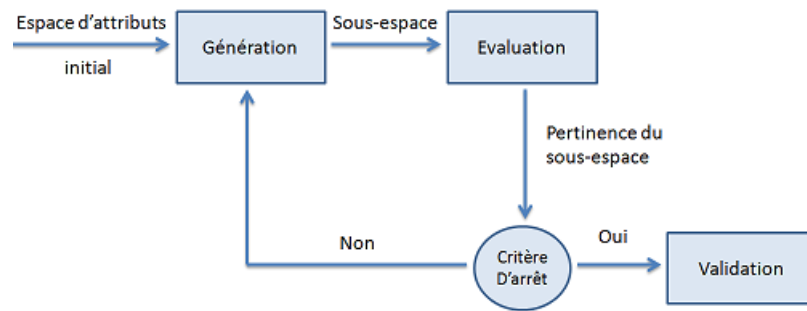


Figure 1.2 : Le processus de la sélection d'attributs

Nous détaillons ci-dessous les étapes importantes de ce processus.

1.3.2.1 La procédure de génération

La procédure de génération permet de générer un sous-ensemble d'attributs pour chaque itération, ce sous-ensemble d'attributs va être évalué lors de l'étape d'évaluation. Pour former un ensemble de départ dans cette procédure de génération, il existe différents cas qui sont fortement liés à la direction de recherche suivit ultérieurement.

L'ensemble considéré comme point de départ peut être soit l'ensemble de tous les attributs disponibles où les attributs qui sont itérativement ajoutés (Forward selection), soit l'ensemble vide où les attributs qui sont itérativement retirés (Backward selection), soit un ensemble d'attributs tirés de manière aléatoire où un nouveau sous-ensemble d'attributs est créé de manière aléatoire à chaque itération (Random generation).

Après l'initialisation, les stratégies de recherche utilisées pour explorer l'espace des attributs peuvent classer les méthodes de génération en trois grandes approches, la génération complète, la génération aléatoire et la génération séquentielle [33].

La génération complète : La procédure de génération complète garantie le résultat optimal par rapport au critère d'évaluation utilisé. Une recherche exhaustive est effectuée pour trouver l'ensemble optimal d'attributs sur tout l'espace des solutions possibles.

Différentes procédures de recherche heuristiques et fonction peuvent être utilisées afin de réduire l'espace de recherche sans prendre des risques de perdre les résultats optimaux : Branch and Bound [34], Beam Search [35].

La stratégie de recherche complète est divisée en «exhaustive» et «non exhaustive». La catégorie exhaustive est une méthode qui peut évaluer tous les sous ensembles où elle peut faire la recherche 'Breadth first' pour arrêter la recherche dès qu'un sous ensemble optimal est trouvé. Dans la catégorie non exhaustive, on trouve différentes techniques de recherches comme : Branch and Bound , Best First et Beam search.

La génération séquentielle : La recherche séquentielle ne garantit pas le résultat optimal, elle permet d'ajouter ou de supprimer un ou plusieurs attributs à chaque itération de cette procédure, on considère à nouveau tout l'ensemble des attributs restants pour l'étape de la sélection.

La génération des sous-ensembles est typiquement incrémentable (diminution/ augmentation). Il existe plusieurs stratégies de recherches heuristiques. Nous nous intéressons à deux approches de génération séquentielle :

Stratégie de recherche séquentielle ascendante (Forward Selection) : Cette approche part d'un ensemble vide et à chaque étape de génération de sous ensemble considère tous les attributs qui ne sont pas encore sélectionnés pour retenir le meilleur entre eux, et l'ajouter à l'ensemble courant.

Stratégie de recherche descendante (Backward Selection) : Cette approche inverse part de l'ensemble total des attributs et à chaque itération elle permet de supprimer un ou plusieurs attributs. Leur principal avantage est qu'elle est de complexité quadratique et qu'elles peuvent être appliquées efficacement à des données de taille modérée. Les algorithmes avec la recherche séquentielle sont connus par leurs simplicités de mise en œuvre et leurs rapidités.

La génération aléatoire : Elle commence par un sous ensemble choisi au hasard. Il y a deux chemins : L'un suit la recherche séquentielle (Random Start, Méthode de Descente et Recuit Simulé [35]). L'autre, chaque ensemble d'attributs est généré de manière complètement aléatoire (le sous ensemble courant n'est pas issu d'une augmentation ou diminution d'attributs du sous ensemble précédent) à titre d'exemple l'algorithme Las-Vigas [36].

Pour toutes les approches, l'utilisation de l'aléatoire aide pour éviter des optima locaux dans l'espace de recherche et le sous ensemble optimal sélectionné dépend de ressources disponibles et de choix des paramètres. les stratégies de génération aléatoire sont divisées en deux types :

- La génération d'un sous ensemble reste constante (la même pour tous les sous ensembles).
- La génération d'un sous ensemble se change lors de l'exécution de programme.

1.3.2.2 La fonction d'évaluation

La fonction d'évaluation permet d'évaluer chaque nouveau sous ensemble généré par des critères d'évaluation. Le choix d'un sous-ensemble optimal, résultat de la procédure de recherche, est relatif à la fonction d'évaluation utilisée. Un meilleur sous ensemble est déterminé par certains critères (peut être optimal pour un critère donné et pas pour un autre). Les critères d'évaluation sont regroupés en deux grandes classes : les critères indépendants et les critères dépendants.

Les critères indépendants : Afin de sélectionner certains attributs ou sous-ensemble d'attributs, la fonction d'évaluation utilisée est basée sur les caractéristiques de l'ensemble des données, indépendamment de tout algorithme de classification. Les critères indépendants les plus populaires sont : les mesures d'information, les mesures de consistance, les mesures de dépendance et les mesures de distance.

Les mesures d'information : Les mesures d'information déterminent typiquement le gain d'information de l'attribut x . Le gain d'information de x est défini comme la différence entre l'incertitude antérieure et postérieure prévue en utilisant cet attribut. L'attribut x est préféré à l'attribut y si le gain de x est plus grand que celui de y (la mesure d'entropie de Shannon).

Les mesures de consistance : Dans la sélection du sous ensemble d'attributs, les mesures de consistance ont une fortes relations avec la classe d'information. Ces mesures cherchent à évaluer si le sous-ensemble d'attributs contient les informations nécessaires à la séparation des classes en essayant de trouver le nombre minimal d'attributs qui séparent les classes comme la consistance de l'ensemble complet d'attributs. L'inconsistance est définie comme deux instances ayant les mêmes valeurs d'attributs mais avec différentes étiquettes de classes.

Les mesures de dépendance : C'est la mesure de corrélation qui mesure la capacité de prévoir la valeur d'une variable à partir des valeurs des autres. La corrélation entre l'attribut x et la classe C est trouvée par la mesure de dépendance classique. Si la corrélation de x avec C est élevée que celle de y avec C , l'attribut x est préféré que l'attribut y . Les fonctions d'évaluation basées sur les mesures de dépendances peuvent être divisées en deux catégories :

- la première est une mesure de corrélation qui quantifie la dépendance des attributs les uns par rapport aux autres.
- La deuxième catégorie est une mesure de dépendance qui caractérise la corrélation entre un attribut ou un sous-ensemble d'attributs et une classe [33].

Les mesures de distance : Dans les mesures de distance (mesure de séparabilité ou divergence ou discrimination), un attribut ou un sous-ensemble d'attributs est sélectionné s'il permet une meilleure séparabilité et cohérence des classes. Pour deux classes de problèmes, un attribut x est préféré à un autre attribut y si x induit la plus grande différence dans la distribution de probabilités conditionnelles des deux classes que y .

En effet, le but est de : maximiser la dispersion inter-classes (séparabilité), afin que les points représentatifs des différentes classes forment dans l'espace d'attributs des nuages les plus séparés possibles les uns des autres. Minimiser la dispersion intra-classe (cohérence), afin que les nuages de points représentatifs de chaque classe soient les plus compacts possible. Les exemples de cette mesure sont : la distance euclidienne, la distance de Mahalanobis, la distance de Bhattacharya et le critère de Fisher. Ces méthodes sont rapides, plus générales et moins coûteuses en temps de calcul.

Les critères de dépendances : Afin de sélectionner certains attributs ou sous-ensemble d'attributs, la fonction d'évaluation dépend l'algorithme de classification utilisée. Dans la classification, la précision est largement utilisée comme une mesure primaire, elle est peut être utilisée comme un critère de dépendance pour la sélection d'attributs.

Les mesures de taux d'erreurs des classificateurs : Dans cette mesure, l'attribut ou les sous-ensembles d'attributs considérés sont évalués en fonction de la qualité de la classification obtenue en utilisant ces attributs. Le sous-ensemble d'attributs le plus discriminant est celui pour lequel le taux d'erreur de classification est le plus faible [37]. Pour la prédiction des classes des instances non connues, le niveau de précision est très élevé bien que très coûteux en calcul.

1.3.2.3 Critère d'arrêt

Le critère d'arrêt consiste à utiliser une règle pour contrôler la sélection de variables qui permettra d'arrêter la recherche dans le cas où aucune variable n'est plus suffisamment informative. Le critère d'arrêt peut être un temps de calcul fixé, un nombre d'itérations fixé, l'absence de gain de performance par rapport aux solutions déjà trouvées ou encore le fait que les sous ensembles candidats deviennent trop homogènes. C'est un choix souvent défini en fonction de la procédure de recherche [38] et/ou du critère d'évaluation [39]. Les critères d'arrêts les plus fréquents sont :

Basés sur l’algorithme de génération : On peut décider d’arrêter la recherche en fixant un seuil sur le nombre d’attributs à sélectionner ou sur le nombre d’itérations. Cependant, dans de nombreuses applications, le nombre d’attributs à sélectionner est très difficile à fixer au préalable. De même, un critère fondé sur un nombre maximal d’itérations peut s’avérer brutal et arrêter trop tôt ou trop tard la sélection [38].

Basés sur l’évaluation : Dans ce cas, on arrête la recherche en fixant un seuil soit sur la fonction d’évaluation, soit entre deux itérations consécutives. Le choix du critère d’arrêt est ainsi un choix délicat qui reste un problème non résolu dans des nombreux algorithmes de sélection.

1.3.2.4 La procédure de validation

La validation permet de tester la validité du sous-ensemble d’attributs sélectionnés pendant la procédure de sélection d’attributs en effectuant plusieurs tests sur des exemples de données générées artificiellement et/ou sur des données réelles. En effet, l’ensemble des données est généralement divisé en deux sous-ensembles distincts (le sous-ensemble d’apprentissage, le sous-ensemble de test). Selon la répartition des données, il existe différentes approches de validation :

La méthode Holdout : les données sont divisées en deux sous-ensembles : le sous-ensemble d’apprentissage et le sous-ensemble de test dans des proportions $1/2$, $1/2$ pour chacun de ses deux sous-ensembles ou $2/3$ pour l’ensemble d’apprentissage et $1/3$ pour l’ensemble de test.

La méthode K-validation croisée : L’ensemble des données est partitionné en K parties de tailles à peu près égales. Nous réalisons ainsi K fois la procédure de validation et à chaque fois une des parties constitue l’ensemble test et les $K-1$ parties restantes sont réunies pour former l’ensemble d’apprentissage.

La méthode de resubstitution : L’ensemble d’apprentissage est utilisé comme ensemble de test.

1.3.3 Les techniques de sélection d'attributs :

Dans la littérature, les techniques de sélection d'attributs sont catégorisées selon différentes manières : les stratégies de recherches, les critères d'évaluation et la manière ou le processus de sélection d'attributs est combiné avec le processus d'apprentissage du modèle. Dans la dernière catégorisation, les techniques de sélection d'attributs peuvent être organisées en quatre classes selon [41]. La première est l'approche enveloppe, la seconde est l'approche embarquée, la troisième est l'approche de filtre et la dernière est l'approche hybride.

1.3.3.1 Les méthodes enveloppes

Les méthodes enveloppes (wrapper methods) effectuent une recherche dans l'espace des sous-ensembles de variables, guidées par le résultat du modèle, donc ils incorporent l'algorithme de classification dans la procédure de sélection d'attributs utilise le taux d'erreur de classification comme critère d'évaluation. Ces méthodes permettent d'obtenir de bonnes performances, elles ont souvent de meilleurs résultats que les autres méthodes[41].

Cependant, l'utilisation de telles méthodes nécessite pour chaque sous espace d'attributs candidats d'effectuer la classification, ce qui peut devenir coûteux en temps de calcul surtout lorsque la dimension de l'espace d'entrée est grande. Ces méthodes sont très dépendantes de l'algorithme de classification utilisé comme critère d'évaluation. Ce dernier, s'il est mal adapté, il pourrait contribuer à une sélection de mauvais attributs. Ces méthodes se reposent sur le choix d'un seuil pour le critère de pertinence choisi ou d'un nombre d'attributs à choisir, le choix de ces paramètres n'est pas facile à réaliser[43].

1.3.3.2 Les méthodes embarquées

Les méthodes embarquées (Embedded methods) utilisent l'information interne du modèle de classification. Ces méthodes sont donc proches des méthodes enveloppes, du fait qu'elles combinent le processus d'exploration avec un algorithme d'apprentissage sans

l'étape de validation, pour maximiser la qualité de l'ajustement et minimiser le nombre d'attributs [41].

La différence entre les méthodes embarquées et les méthodes enveloppes est que le classificateur sert non seulement à évaluer un sous-ensemble candidats mais aussi à guider le mécanisme de sélection. Selon [41], ces méthodes seraient bien plus avantageuses en terme de temps de calcul que les méthodes de type enveloppes et seraient robustes face au problème de sur-apprentissage.

1.3.3.3 Les méthodes filtres

Les méthodes filtres (filter methods) opèrent directement sur les données, elles ne tiennent que des informations présentes dans les données. Elles sont indépendantes de tout algorithme de classification et utilisent une fonction d'évaluation basée sur les caractéristiques de l'ensemble des données afin de sélectionner certains attributs ou sous-ensemble d'attributs.

Ces méthodes ont l'avantage d'être rapides et indépendantes du modèle de classification, mais au prix de résultats inférieurs, Cependant, comme elles sont indépendantes de l'étape de classification, elles ne permettent pas de garantir que le meilleur taux de classification soit obtenu dans l'espace retenu et il y a un risque de Sur-apprentissage.

1.3.3.4 Les méthodes hybride

Les méthodes hybrides (hybrid methods) sont récemment proposés pour de large base de données [42]. Ces algorithmes se concentrent principalement sur la combinaison des algorithmes filtres et enveloppes pour obtenir les meilleures performances possibles, avec un algorithme d'apprentissage particulier et une complexité temporelle comparable à celle des algorithmes de type filtres. Ces algorithmes utilisent des mesures de sous-ensembles d'attributs basées sur les caractéristiques des données pour choisir les meilleurs sous-ensembles pour une cardinalité donnée, ensuite elles exploitent une validation croisée pour choisir le meilleur sous-ensemble final entre les différentes cardinalités.

Conclusion

Ce chapitre a pour objectif d'introduire l'extraction de connaissances à partir de données (ECD), de citer le processus d'extraction de connaissances en insistant sur une étape importante qui est la fouille de données, ses tâches et plus précisément la classification supervisée qui est une tâche centrale de l'étape de fouille de données dans le processus d'extraction de connaissances à partir de données et de présenter la sélection d'attributs.

La sélection d'attributs est l'une des techniques importantes et fréquentes appliquée à la tâche de la classification dans la fouille de données, elle assure une réduction de la dimension de données en améliorant la précision de la prédiction et de la compréhension du classificateur. La recherche d'un sous ensemble d'attributs optimal à partir d'un ensemble d'attributs de dimension élevée est un problème d'optimisation NP-difficile, de sorte que différentes stratégies de méta-heuristiques sont appliquées pour trouver le meilleur sous-ensemble d'attributs possibles.

Le chapitre suivant est consacré aux méthodes dites les méta-heuristiques.

États de l'art

Introduction

La classification est devenue un domaine hautement actif dans la recherche. Pour cette tâche, des techniques sophistiquées ont été développées et appliquées dans les domaines aussi bien commerciaux que scientifiques. Mais le besoin existe toujours pour compléter ces méthodes par de nouvelles qui seraient capables de traiter les données qui deviennent toujours plus grandes, plus complexes, ayant une structure non linéaire, et par conséquent mal expliquées par les outils traditionnels, ces méthodes sont les méta-heuristiques.

Le présent chapitre est divisé en trois parties, la première partie est consacré aux méthodes méta-heuristiques. Nous présenterons ses enjeux, ainsi que la classification de ces différentes approches. Dans la deuxième partie, nous nous focalisons sur l'utilisation des méta-heuristiques pour la sélection des attributs où nous citons les différentes méthodes proposées dans la littérature. La troisième partie sera consacrée à la classification par les méta-heuristiques en citant les différentes méthodes implémentées dans la littérature. Enfin nous terminons par une conclusion.

2.1 Les méta-heuristiques

2.1.1 Définition

Dans la résolution de problèmes d'optimisation, deux types de méthodes sont souvent mises en opposition : les méthodes exactes et les méthodes approchées. Les méthodes exactes fournissent une solution optimale au problème traité et ont généralement le défaut d'être très coûteuses en temps d'exécution pour des problèmes difficiles. Au contraire, les méthodes approchées visent à générer des solutions de haute qualité en un temps de calcul raisonnable, mais il n'existe aucune garantie de trouver la solution optimale.[44]

Les méthodes heuristiques sont une sous-catégories de ces méthodes approchées. Ils peuvent encore diviser en deux sous-classes : les heuristiques spécifiques à un problème donné, et les méta-heuristiques qui regroupent des méthodes plus génériques.

Les méta-heuristiques sont des techniques apparues dans les années 1980 pour compléter les approches existantes et être en mesure d'analyser la quantité phénoménale de donnée disponible. Elles se comportent comme des algorithmes de recherche, tentant d'apprendre les caractéristiques d'un problème à fin d'en trouver une approximation de la meilleure solution d'une manière proche des algorithmes d'approximations. Les méta-heuristiques ont montré leur efficacité dans de vastes domaines d'application en résolvant de nombreux problèmes d'optimisation.

Plusieurs définitions ont été proposées pour expliquer la notion de méta-heuristique, nous citons parmi elles :

Définition 2.1.1 *“A meta-heuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solutions.”* [45]

Définition 2.1.2 *“A meta-heuristic is an iterative master process that guides and modi-*

fies the operations of subordinate heuristics to efficiently produce high-quality solutions. It may manipulate a complete (or incomplete) single solution or a collection of solutions at each iteration. The subordinate heuristics may be high (or low) level procedures, or a simple local search, or just a constructive method.” [46]

Définition 2.1.3 *“Meta-heuristics are typically high level strategies which guide an underlying more problem specific heuristic, to increase their performance. The main goal is to avoid the disadvantages of iterative improvement and, in particular, multiple descents by allowing the local search to escape from local optima. This is achieved by either allowing worsening moves or generating new starting solutions for the local search in a more “intelligent” way than just providing random initial solutions. Many of the methods can be interpreted as introducing a bias such that high quality solutions are produced quickly. This bias can be of various forms and can be cast as descent bias (based on the objective function), memory bias (based on previously made decisions) or experience bias (based on prior performance). Many of the meta-heuristic approaches rely on probabilistic decisions made during the search. But the main difference to pure random search is that in meta-heuristic algorithms randomness is not used blindly but in an intelligent, biased form.” [47]*

Dans [48], les auteurs résument les propriétés attachées à la notion de méta-heuristique :

- Les méta-heuristiques sont des stratégies qui “guident” le processus de recherche.
- L’objectif est d’explorer efficacement l’espace de recherche afin de trouver une solution proche de l’optimal.
- Les techniques mises en œuvre dans les méta-heuristiques vont de la simple recherche locale à des procédures complexes d’apprentissage.
- Les méta-heuristiques sont des algorithmes de résolution incertains et souvent non déterministes.
- Elles peuvent intégrer des mécanismes évitant d’être piégées dans une zone de l’espace de recherche.

- Les méta-heuristiques sont décrites suivant un niveau d'abstraction indépendant du problème spécifique à traiter.
- Les méta-heuristiques peuvent encapsuler les informations spécifiques au problème sous la forme de sous-heuristiques contrôlées à un niveau supérieur.
- Les méta-heuristiques les plus modernes introduisent des mécanismes pour adapter et guider la recherche dynamiquement.

L'intérêt croissant apporté aux méta-heuristiques est tout à fait justifié par le développement des machines avec des capacités calculatoires énormes, ce qui a permis de concevoir des méta-heuristiques de plus en plus efficace lors de la résolution de plusieurs problèmes à caractère NP-difficile pour les quels on ne connaît pas de méthode classique pratique.

2.1.2 Les enjeux des méta-heuristiques

Les enjeux des méta-heuristiques groupés par Meignan [49] peuvent être présentés dans les points suivante :

2.1.2.1 La performance

Tout comme les heuristiques, l'objectif principal d'une méta-heuristique est de fournir, une fois instanciée, une méthode de résolution performante en termes de qualité de résultat et de temps de résolution. L'exécution distribuée des méta-heuristiques est une solution intéressante pour améliorer les performances de ces méthodes. Cette problématique est largement encouragée avec l'apparition récente des architectures de processeur multi-core permettant l'exécution simultanée de plusieurs processus sur un même processeur d'ordinateur.

2.1.2.2 La robustesse

Une fois instanciée, une méta-heuristique doit être capable de résoudre aussi efficacement que possible différentes instances d'un même problème. Il s'agit non seulement de pouvoir traiter de manière efficace des instances de différentes tailles, mais de traiter

aussi des instances dont la configuration est différente, tout cela en évitant à l'utilisateur d'intervenir sur les paramètres de la méthode de résolution. Différentes versions dites adaptatives ou auto-adaptatives ont été conçues pour assurer cette robustesse, principalement en modifiant dynamiquement les paramètres de la méthode.

2.1.2.3 La simplicité de mise en œuvre

L'utilisation d'une méta-heuristique doit permettre de simplifier le développement par rapport à la mise en œuvre d'une heuristique spécifique au problème traité. La facilité de compréhension est aussi un élément essentiel des méta-heuristiques. Ainsi, la métaphore joue un rôle important dans les méta-heuristiques et peut expliquer le succès de certaines d'entre elles.

2.1.2.4 La flexibilité

Par définition une méta-heuristique est un schéma générique de résolution pouvant s'appliquer à différents problèmes d'optimisation. La flexibilité se définit par la capacité d'une méta-heuristique de traiter différents problèmes en conservant ses performances.

2.1.3 Le fonctionnement de méta-heuristique

Les méta-heuristiques fonctionnent selon un comportement itératif, en alternant des phases d'intensification, de diversification et la notion de mémoire. L'état de départ est souvent choisi aléatoirement, l'algorithme se déroulant ensuite jusqu'à ce qu'un critère d'arrêt soit atteint.

2.1.3.1 La diversification

La diversification ou exploration désigne les processus visant à récolter de nouvelles informations sur le problème à optimiser. La stratégie de diversification la plus simple consiste à redémarrer périodiquement le processus de recherche à partir d'une solution générée aléatoirement ou choisie judicieusement dans une région non encore visitée de l'ensemble des solutions admissibles.

2.1.3.2 L'intensification

Pour sa part, l'intensification ou exploitation utilise l'information déjà récoltée pour explorer, en détail, les zones jugées prometteuses dans l'espace de recherche. Sa mise en œuvre réside, le plus souvent, dans l'élargissement temporaire du voisinage de la solution courante. Du point de vue des méta-heuristiques, il s'agit dans les exemples extrêmes d'une recherche locale déterministe.

2.1.3.3 La mémoire

La mémoire est le support de l'apprentissage, qui permet à l'algorithme de ne tenir compte que des zones où l'optimum global est susceptible de se trouver et de garder en mémoire les résultats passés, évitant ainsi les optimums locaux. Citons des méthodes ayant une mémoire à long terme qui conserve une trace de mouvements récemment effectués, des solutions visitées ou en générale des décisions prises et une mémoire à court terme qui est une accumulation de paramètres de synthèse sur la recherche.

Dans le cas le plus simple, elles se limitent à considérer l'état de la recherche à une itération donnée, pour déterminer la prochaine itération, on parlera donc de méthode sans mémoire. C'est le cas de la plupart des méthodes de recherche locale. Toutes les méta-heuristiques s'appuient sur l'équilibre entre l'intensification et la diversification de la recherche. Ces deux dynamiques de recherche ne sont donc pas contradictoires, mais complémentaires. Sinon, on assistera à une convergence trop rapide vers des minima locaux par manque de diversification ou à une exploration trop longue par manque d'intensification.

2.1.4 La classification des méta-heuristiques

Les méta-heuristiques ont montré leur efficacité dans de vastes domaines d'application. Elles sont généralement conçues au départ pour des problèmes discrets mais peuvent s'adapter aux autres types de problèmes. Il existe plusieurs manières de classer les méta-heuristiques selon l'utilisation de l'historique de la recherche, l'inspiration de la

nature ou non et selon leur principe de fonctionnement durant la recherche de la solution. On choisit cette dernière classification où on peut distinguer deux catégories :

2.1.4.1 Les approches trajectoires

Les approches trajectoires appelées aussi méthodes itératives à solution unique. Elles sont toutes basées sur un algorithme de recherche de voisinage qui commence avec une solution initiale, puis s'améliore pas à pas en choisissant une nouvelle solution dans son voisinage [50]. Ces algorithmes partent d'une solution initiale (obtenue de façon exacte, ou par tirage aléatoire) et s'en éloignent progressivement, pour réaliser une trajectoire. Dans cette catégorie, se rangent : la méthode de descente, le recuit simulé, la méthode Tabou et la recherche par voisinage variable.

La méthode de descente : Le principe de la méthode de descente [51] (dite aussi basic local search) consiste à partir d'une solution s et à choisir une solution s' dans un voisinage de s , telle que s'améliore la recherche (généralement telle que $f(s') < f(s)$). On peut décider soit d'examiner toutes les solutions du voisinage et prendre la meilleure de toutes (ou prendre la première trouvée), soit d'examiner un sous-ensemble du voisinage. La méthode de Descente peut être décrite comme suit :

Algorithm 1 Méthode de Descente

- 1: **procédure** DESCENTE SIMPLE (SOLUTION INITIALE S)
 - 2: **Répéter :**
 - 3: Choisir s' dans $N(s)$ **Si** $f(s') < f(s)$ **alors** $s \leftarrow s'$
 - 4: **Jusqu'à ce que** $f(s') = f(s), \forall s' \in S$
-

L'ensemble S définit l'ensemble des points pouvant être visités durant la recherche. La structure de voisinage N donne les règles de déplacement dans l'espace de recherche. La fonction f induit une topologie sur l'espace de recherche. Une variante consiste à parcourir $N(s)$ et à choisir la première solution s' rencontrée telle que $f(s') = f(s)$.

La méthode du recuit simulé : Le recuit simulé (simulated annealing) est souvent présenté comme la plus ancienne des méta-heuristiques, en tout cas, la première à mettre spécifiquement en oeuvre une stratégie d'évitement des minima locaux [52]. Elle s'inspire d'une procédure utilisée depuis longtemps par les métallurgistes. Pour obtenir un alliage sans défaut, les métallurgistes chauffent d'abord à blanc leur morceau de métal, avant de laisser l'alliage se refroidir très lentement (technique du recuit).

Pour simuler cette évolution d'un système physique vers son équilibre thermodynamique à une température T , la méthode du recuit simulé exploite l'algorithme de Métropolis. Cet algorithme commence par s_0 , et continue jusqu'à k_{max} (maximum d'étapes) ou jusqu'à un état ayant une énergie e_{max} , l'appel voisin (s) engendre un état aléatoire voisin d'un état s , l'appel aléatoire renvoie une valeur aléatoire dans l'intervalle $[0,1]$. L'appel $temp(r)$ renvoie la température à utiliser selon la fraction r du temps total déjà dépensé.

Le recuit simulé peut être décrit comme suit :

Algorithm 2 Recuit Simulé

- 1: $s \leftarrow s_0; e \leftarrow E(s); k \leftarrow 0;$
 - 2: Tantque $k < k_{max}$ ét $e < e_{max}$ $s \leftarrow voisin(s);$
 - 3: $e_n \leftarrow E(s_n);$
 - 4: Si $e_n < p(e_n - e, temp(k/k_{max}))$ alors $s \leftarrow s_n;$
 - 5: $e \leftarrow e_n;$
 - 6: $k \leftarrow k + 1;$
 - 7: Retourner s
-

La méthode Tabou : La méthode Tabou est une technique de recherche dont les principes ont été proposés pour la première fois par Fred Glover [53] dans les années 80, et elle est devenue très classique en optimisation combinatoire. Elle se distingue par des méthodes de recherche locale simples, par le recours à un historique des solutions visitées, de façon à rendre la recherche un peu moins « aveugle ». Il devient donc possible de s'extraire d'un minimum local, mais, pour éviter d'y retomber périodiquement, certaines solutions sont bannies, elles sont rendues « taboues » [53].

À l'inverse du recuit simulé qui génère de manière aléatoire une seule solution voisine à chaque itération, Tabou examine un échantillonnage de solutions de $N(s)$ et retient la meilleure s' même si $f(s') > f(s)$. La recherche Tabou ne s'arrête donc pas au premier optimum trouvé. Le danger serait alors de revenir à s immédiatement, puisque s est meilleure que s' . Pour éviter de tourner ainsi en rond, on crée une liste T qui mémorise les dernières solutions visitées et qui interdit tout déplacement vers une solution de cette liste. Cette liste T est appelée liste Tabou. La recherche Tabou peut être décrite comme suit :

Algorithm 3 Recherche Tabou

- 1: $s \leftarrow s_0; T \leftarrow \emptyset; s^* \leftarrow s;$
 - 2: **Tant que** (arrêt n'est pas satisfait)
 - 3: Déterminer une solution s' qui minimise $f(s')$ dans $NT(s)$
 - 4: **Si** $f(s') < f(s^*)$ **alors** $s^* \leftarrow s';$
 - 5: $s \leftarrow s';$
 - 6: mise-à-jour $T;$
 - 7: **Fin tq**
-

La méthode GRASP : La méthode GRASP (pour Greedy Randomized Adaptive Search Procedure) est une méta-heuristique simple, développée à la fin des années 90 par Feo et Resende [54]. Elle est adaptée aux problèmes dont les solutions se construisent pas à pas, comme le problème d'ordonnancement de tâches, dont la solution consistera en une certaine suite d'opérations à ordonner sur des machines[54].

La méthode GRASP est une méthode itérative qui se déroule en deux phases :

- La phase de construction continue jusqu'à ce qu'une solution complète soit obtenue.
- La phase d'amélioration : à partir de la solution trouvée par la phase précédente, une méthode de Descente, Recuit Simulé ou Recherche Tabou est appliquée pour améliorer la solution.

La méthode GRASP peut être décrite comme suit :

Algorithm 4 La méthode GRASP

- 1: $f^* \leftarrow \infty$ (une valeur de meilleure solution rencontrée)
 - 2: **Tant que** (critère d'arrêt n'est pas vérifié)
 - 3: $s \leftarrow \emptyset$;
 - 4: Tant que (la solution courante n'est pas complète)
 - évaluer l'ensemble des éléments pouvant être rajoutés à s
 - déterminer la liste RCL
 - choisir un meilleur élément dans RCL et l'ajouter à S .
 - 5: **Fin tq**
 - 6: Appliquer une méthode de Descente, Recuit Simulé ou de Recherche Tabou à
 - 7: $s' \leftarrow s$;
 - 8: **Si** $f(s') < f(s^*)$ **alors** $s^* \leftarrow s'$;
 - 9: $f(s^*) \leftarrow f(s')$;
 - 10: **Fin tq**
-

2.1.4.2 Les approches à population

Les approches à population ou évolutionnaires consistent à travailler avec un ensemble de solutions simultanément, que l'on fait évoluer graduellement. Les méthodes évolutives manipulent un groupe de solutions admissibles à chacune des étapes du processus de recherche. L'utilisation de plusieurs solutions simultanément permet naturellement d'améliorer l'exploration de l'espace des configurations. Dans cette seconde catégorie, se rangent : les algorithmes génétiques, les algorithmes par colonies de fourmi, l'optimisation par essaim particulier, les algorithmes à estimation de distribution, le path relinking (ou chemin de liaison).

Les algorithmes génétiques : Avec les algorithmes évolutionnaires, nous passons à une autre catégorie de méta-heuristiques, celles des méthodes dites évolutionnaires, qui manipulent un ensemble de plusieurs solutions simultanément. La méta-heuristique la

plus connue dans cette branche est celle reposant sur un algorithme génétique, inspiré du concept de sélection naturelle élaboré par Darwin [55].

Dans la théorie de l'évolution et de la génétique, la connaissance acquise par chaque espèce est codée dans les chromosomes de ses membres. Lors des reproductions sexuelles, les contenus des chromosomes sont mélangés, modifiés et transmis aux descendants par un certain nombre d'opérateurs génétiques : la mutation, qui se traduit par l'inversion d'une faible partie du matériel génétique, et le croisement qui échange certaines parties des chromosomes des parents.

Nous parlerons donc d'individus, pour parler de solutions (complètes, ou partielles). L'ensemble des individus forme une population, que nous ferons évoluer pendant une certaine succession d'itérations appelées générations, jusqu'à ce qu'un critère d'arrêt soit vérifié. Pour passer d'une génération à une autre, nous soumettrons la population à des opérateurs de sélection. Les opérateurs de variation, eux, permettront de transformer la population, de façon à favoriser l'émergence de meilleurs individus. Ces opérateurs sont à définir en fonction du problème considéré.

On définit également la fonction de performance (fitness), qui traduit la capacité d'adaptation des individus au problème considéré. Cette fonction, que l'algorithme génétique cherche à maximiser, est étroitement dépendante de la fonction objectif.

Algorithme Génétique peut être décrit comme suit :

Algorithm 5 L'algorithme génétique

- 1: Générer aléatoirement une population d'individus.
 - 2: Pour passer d'une génération k à une génération $k+1$, on applique les trois opérations pour tous les éléments de la population k :
 - sélectionner les couples de parents p_1, p_2 en fonction de leur adaptation.
 - appliquer l'opérateur de croisement avec une probabilité p_c et générer les couples d'enfants $c_1 \text{ etc } c_2$, certains éléments sont choisis en fonction de leur adaptation, on applique l'opérateur de mutation avec une probabilité p_m et on génère les individus p'
 - évaluer $c_1 \text{ etc } c_2, p'$ pour l'insérer dans la nouvelle population.
 - 3: **Le critère d'arrêt** : Soit le nombre de générations est fixé à l'avance ou un temps limité pour trouver une solution où il n'y a pas d'évolution.
-

Les algorithmes de colonies de fourmi : Les fourmis sont des insectes sociaux qui présentent un comportement de groupe intelligent et complexe. Le comportement collectif des insectes apparaît à partir d'une interaction simple et indirecte des membres d'un groupe appelé les colonies.

Une colonie de fourmis communique indirectement via des modifications dynamiques des pistes de phéromones et construisent une solution à un problème donné. La phéromone (un produit chimique volatil produite par les glandes situées dans les abdomens des fourmis) est utilisée pour des alertes de sécurité et pour guider les autres fourmis vers la source de nourriture. Le comportement coopératif de recherche de nourriture était un intérêt particulier pour les chercheurs qui ont constaté que la colonie est en mesure de trouver le plus court chemin entre le nid et une source de nourriture même si les fourmis individuelles n'ont pas une vision globale du chemin. Comme les algorithmes génétiques, les algorithmes de colonies de fourmi font évoluer une population d'agents, selon un modèle stochastique.

Cet algorithme est encore inspiré de la nature et de son organisation, et a été mis au point par Dorigo [56] au début des années 90. Son principe repose sur le comportement particulier des fourmis qui, lorsqu'elles quittent leur fourmilière pour explorer leur environnement à la recherche de nourriture, finissent par élaborer des chemins qui s'avèrent fréquemment être les plus courts pour aller de la fourmilière à une source de nourriture intéressante.

Chaque fourmi laisse en effet derrière elle une traînée de phéromone à l'attention de ses congénères ; les fourmis choisissent alors une plus grande probabilité les chemins contenant les plus fortes concentrations de phéromones, il se forme ainsi ces « auto-routes » à fourmis, qui sillonnent le paysage. Ce mode de communication particulier, qui fait intervenir des modifications dans l'environnement, est appelé stigmergie. Algorithme de colonies de fourmi peut être décrit comme suit :

Algorithm 6 L'algorithme de Colonies de Fourmi

- 1: Initialiser la trace de la phéromone τ à 0 pour toute décision.
 - 2: **Tant que** (Critère d'arrêt non atteinte) **Faire**
 - 3: **Pour** Chaque fourmi $k \in k$ **Faire**
 - 4: Construire une solution en tenant compte de la visibilité et la trace
 - 5: Mettre à jour la trace τ_d ainsi que la solution trouvée
 - 6: **Fin pour**
 - 7: **Fin Tq**
-

L'évolution différentielle : L'évolution différentielle [57] est une méthode pour l'optimisation mathématique des fonctions multidimensionnelles. Appeler aussi approche de mesure du poids de chaque attribut. Cette approche permet de chercher à optimiser un sous-ensemble, on cherche à appliquer des fonctions qui donnent un degré de pertinence à tous les attributs durant l'apprentissage, une valeur de poids élevé pour les terminaux pertinents, et un degré proche de zéro pour les attributs non pertinents.

L'idée principale est l'utilisation d'un opérateur de recombinaison ternaire pour la créa-

tion de nouvelles générations. Le nouvel individu est créé en ajoutant la différence entre deux individus à un troisième individu. L'évolution différentielle utilise des informations extraites de la population courante afin de déterminer l'évolution future de la recherche. Il s'inspire en partie de la boucle évolutionnaire car il peut se décomposer en plusieurs phases proches des techniques évolutionnaires : l'initialisation, l'évaluation, la mutation, le croisement et la sélection.

Algorithme de L'évolution différentielle peut être décrit comme suit :

Algorithm 7 L'algorithme de L'évolution différentielle

- 1: Initialisation de la population de μ individus
 - 2: Evaluation des μ individus
 - 3: **Tant que** le critère d'arrêt n'est pas satisfait faire
 - 4: Sélection de ρ individus en vue de la phase de reproduction
 - 5: Croisement des ρ individus sélectionnés Mutation des λ enfants obtenus
 - 6: Evaluation des λ enfants obtenus
 - 7: Sélection pour le remplacement
 - 8: **Fin**
-

L'optimisation par essaim particulier : L'optimisation par essaim particulier (OEP), ou Particle Swarm Optimization (PSO) en anglais, est un algorithme évolutionnaire qui utilise une population de solutions candidates pour développer une solution optimale au problème. Cet algorithme a été proposé par Russel Eberhart (ingénieur en électricité) et James Kennedy (socio-psychologue) en 1995 [58]. Il s'inspire à l'origine du monde du vivant, plus précisément du comportement social des animaux évoluant en essaim, tels que les bancs de poissons et les vols groupés d'oiseaux.

L'algorithme PSO déplace un essaim de particules (les solutions candidates) dans l'espace de recherche en modifiant leur position et leur vitesse selon des règles simples. Le déplacement de chaque particule est influencé par sa vitesse, la meilleure position qui a été retenue et la meilleure position connue par toutes les particules de l'essaim.

Les positions des particules améliorées localement permettent à l'essaim de se déplacer

globalement vers de bonnes solutions. L'algorithme de l'optimisation par essaim particulier peut être décrit comme suit :

Algorithm 8 L'algorithme d'optimisation par essaim particulier

- 1: Initialiser l'essaim (positions et vitesses des particules)
 - 2: Evaluer les particules
 - 3: **Répéter**
 - 4: **Pour** chaque particule de l'essaim **faire**
 - 5: Mettre à jour la vitesse de la particule
 - 6: Mettre à jour la position de la particule
 - 7: Mettre à jour la meilleure position connue de la particule
 - 8: Mettre à jour la meilleure position connue de l'essaim
 - 9: **Fin pour**
 - 10: **Jusqu'à** critère d'arrêt vérifié
 - 11: Renvoyer meilleure position connue de l'essaim
-

2.2 Les méta-heuristiques pour la sélection d'attributs

Le principe de la sélection des attributs consiste à choisir parmi un ensemble d'attributs de grande taille, un sous ensemble d'attributs importants en évaluant chaque attribut pour lui assigner un score de pertinence qui permet de classer les attributs.

Les attributs les mieux classés (les plus pertinents) seront sélectionnés pour la phase du traitement. Le problème de la sélection d'attributs fait parti des problèmes NP_difficiles puisque pour N attributs, nous avons $2^N - 1$ sous ensembles possibles donc il est considéré comme un problème d'optimisation combinatoire. C'est pourquoi, les méta-heuristiques semblent bien appropriées pour traiter ce problème.

Il existe des méthodes différentes qui sont implémentées pour la sélection d'attributs par les méthodes méta-heuristiques parmi elles nous citons :

Dans le travail de [59], une approche de sélection d'attributs basés sur l'algorithme génétique est présentée. Cette approche utilise les réseaux de neurones pour la classification. Les résultats présentés dans cet article indiquent que les algorithmes génétiques constituent une approche attrayante pour résoudre le problème de sélection de sous ensemble d'attributs pour des différents tâches telles que le diagnostic médical et d'autres applications intéressantes incluent l'exploration automatisée de données et la découverte de connaissances à partir de données. Des expériences ont été menées sur 11 datasets de l'UCI. Une étude comparative de la performance de l'approche proposée avec celle des méthodes classiques pour la sélection de sous ensemble d'attributs est effectuée.

Les auteurs [60] ont utilisé la recherche par dispersion (Scatter Search) pour la sélection de sous ensemble d'attributs. Cette méthode se diffère de l'algorithme génétique par la taille de l'ensemble de solutions évaluées et la façon de combiner les solutions existantes pour fournir de nouvelles solutions. Cette méthode est combinée avec trois classificateurs (IB1, Naive Bayes et C4.5). Les expérimentations sont réalisées sur douze datasets de l'UCI. Les résultats obtenus sont comparables à celles obtenues dans la littérature.

Dans le travail [61], les auteurs proposent une méthode hybride avec un algorithme génétique et une méthode basée sur la séparabilité de classes pour la sélection de sous ensemble d'attributs. Cet article utilise la méthode filtre de séparabilité de classes pour initialiser les algorithmes génétique. L'objectif de cette étude est de déterminer si la méthode hybride présente des avantages par rapport aux algorithmes génétique simples et aux différents algorithmes de sélection des attributs. Les expériences décrites dans cet article utilisent le classificateur Naive Bayes pour des ensembles de données de domaine public et artificiel (9 datasets). Elles démontrent que, dans la plupart des cas, l'algorithme hybride proposé trouve de meilleure résultats en termes de précision de classification et de temps de calcul.

L'auteur de [62] a proposé une nouvelle version de l'algorithme Ant Miner appelé CAnt Miner. CAnt Miner est une approche hybride de l'Algorithme de Colonies de Fourmis et les Arbres de Décisions C4.5. Dans cette approche Wrapper, une nouvelle fonction heuristique basée sur la compatibilité entre les attributs lors de la construction d'une règle a été proposée. En plus l'auteur introduit également une stratégie pour arrêter dynamiquement l'ajout de termes dans la partie antérieure de la règle au lieu d'utiliser un seuil spécifié par l'utilisateur. Les expérimentations sont réalisées sur des bases de données de l'UCI. Les résultats de cette approche sont comparés à ceux obtenus par : l'Algorithme Génétique, la recherche par dispersion et C4.5. Les résultats obtenus sont compétitifs.

Dans le travail de Candelieri [63], l'auteur explore et compare l'hybridation de plusieurs méta-heuristiques, y compris l'algorithme génétique, la recherche tabou et optimisation par colonies de fourmis pour implémenter un modèle de sélection d'attributs. Ces hybridations utilisent le classificateur SVM à la fois en tant que classificateur unique et ensembliste.

Dans ce framework, SVM est utilisé pour l'apprentissage du modèle de classification. Dans le cas d'un classificateur unique, la méta-heuristique est utilisée en premier lieu pour rechercher la fonction noyau performante (linéaire, Polynôme ou Radial) et ses paramètres associés ; en deuxième lieu pour optimiser les n fonctions noyau ainsi que leurs paramètres et coefficients associés. Dans le cas d'un classificateur ensembliste, la méta-heuristique est utilisée pour trouver le meilleur m SVM classificateurs et leurs poids associés pour la combinaison. Le framework a été testé sur huit datasets dont plusieurs étaient médicaux. Pour l'évaluation, une validation croisée de 10folds a été utilisé.

Les résultats montrent que les trois modèles étaient très compétitifs. L'Algorithme Génétique était généralement plus rapide et plus efficace que la Recherche Tabou. En ce qui concerne l'optimisation par colonies de fourmis, les résultats ont été comparables aux autres applications.

Dans le travail de [64], Les paramètres des classificateurs et les caractéristiques des données sont choisis d'une manière stochastique et évoluée indépendamment en utilisant la recherche de dispersion (scatter search). Les paramètres étaient pour un ensemble de trois classificateurs ; SVM, BP-ANNs et DT(C4.5). Les instances de données avec les valeurs manquantes ont été supprimées de tous les classificateurs à l'exception du DT. Chaque classificateur a été exécuté trois fois et un vote majoritaire a été obtenu pour combiner les différentes exécutions. Des expériences ont été menées sur 18 datasets de UCI et ont été comparées à quatre études similaires dont certaines impliquent l'utilisation d'ensembles d'un plus grand nombre des classificateurs. L'approche proposée a atteint la plus haute précision de classification. En outre, en comparaison avec les résultats des classificateurs individuels, La performance moyenne de l'ensemble est meilleure que les classificateurs individuels.

les auteurs de [65] introduisent un modèle à quatre étapes pour la classification des données médicales. Ce modèle utilise des techniques de prétraitements et de regroupements de données pour améliorer la précision de la classification. Dans la première étape qui concerne la sélection de sous ensemble d'attributs en utilisant une régression par étapes.

Les sous ensembles d'attributs sélectionnés sont ensuite pondérés par la méthode de gradient. La prochaine étape concerne le regroupement de raisonnement à base des cas (CBR) des données d'entrée. Ensuite, la fonction d'appartenance triangulaires floues (the fuzzy Triangle membership function) est appliquée pour discrétiser les données et l'arbre de décision ID3 est appliqué pour créer un arbre de classification pour chaque groupe. La dernière étape consiste à extraire les termes flous utilisés dans l'arbre de décision en utilisant l'algorithme génétique pour améliorer encore la précision de la classification.

Le modèle a été appliqué à deux datasets de UCI ; WDBC et liver disorder. Des comparaisons par rapport à plusieurs méthodes de classification y compris k-NN, Naive Bayes, SVM, Arbre de Décision Floue et à d'autres études similaires ont été effectuées. Les résultats montrent que la précision moyenne atteinte était de 99,5% et 85% pour le WDBC

et liver disorder, respectivement.

L'article ne montre pas clairement comment le codage de l'algorithme génétique est fait pour permettre un nombre différent de termes flous pour chaque attribut.

L'étude de [66] démontre l'avantage d'utiliser des méthodes le calcul évolutionnaire pour guider le paramètre et le choix des représentants de classe. Cette étude évalue l'effet sur la précision de la classification en ajoutant du bruit aux attributs, en ajoutant des variables bruyantes supplémentaires et en ajoutant toutes les variables à deux éléments. Un simple classificateur de distance minimum (instance based) était appliqué à quatre datasets de l'UCI : New Thyroid, HungarianHeart, Heart-Statlog et Lenses.

Dans le travail de [67], une méthode de PSO discret est appliquée comme méthode embarqué de sélection de sous-ensemble d'attributs pour un problème de classification binaire.

La régression linéaire a été choisie comme algorithme d'apprentissage. L'attribut est ajouté au sous ensemble d'attributs collectés si cela améliore la précision de la classification. La complexité computationnelle est gérée en limitant le nombre d'attributs considérés dans chaque itération. Les paramètres PSO sont basés sur des recherches empiriques et théoriques antérieures.

Des expériences ont été effectuées sur dix datasets de l'UCI datasets montrent que PSO a produit une précision identique ou presque identique à une recherche exhaustive accompagnée. Les résultats montrent en outre que la précision de PSO était compétitive par rapport à la génération de sous-ensemble aléatoire.

D'autres comparaisons ont été effectuées avec d'autres méthodes de sélection de sous ensemble d'attributs en utilisant le même algorithme d'apprentissage et les mêmes datasets. Ces méthodes comprennent la recherche tabus et la recherche par dispersion (scattersearch).

Les résultats montrent la supériorité de l'algorithme proposé en termes de précision de classification et de coût du calcul.

La méta-heuristique ACO a été utilisée par [68] en combinaison avec un classificateur à base de règles floues (Fuzzy rule based classifier). Ce classificateur effectue l'apprentissage de modèle tandis que l'ACO est utilisé pour la sélection des sous ensembles d'attributs. Cette sélection d'attributs basé ACO est appelée (AFS). AFS se compose de deux colonies : la première est attribué à la détermination du nombre de d'attributs à sélectionner et la seconde, sélectionne les attributs elles-mêmes.

Le choix de la cardinalité d'attributs est basé sur le critère discriminant Fisher. La fonction de fitness est basée sur la minimisation du nombre d'attributs sélectionnés et la précision de classification du modèle obtenu à l'aide du classificateur basé sur les de règles floues. Les expériences ont été effectuées sur cinq datasets de l'UCI. Les résultats de ces expériences montrent une amélioration significative par rapport aux d'autres études citer dans la littérature.

Dans le travail de [69], quatre méthodes différentes appelées PSO, l'engin de particules Quantum-Behave, SVM (LSSVM) et la stratégie de jeu active, sont utilisés pour l'apprentissage du classificateur SVM. L'évaluation et la comparaison de ces méthodes montrent que l'approche SVM-PSO donne une meilleure précision sur un ensemble de données du WDBC.

Dans le travail de [70], on se base sur un classificateur SVM, trois approches, dont l'AG, l'optimisation des colonies de fourmis (ACO) et l'optimisation des essaims de particules (PSO), ont été utilisées pour sélectionner les attributs les plus importantes dans le Wisconsin Diagnostic Breast Cancer(WDBC) dataset. Dans leur recherche, les résultats pour PSO-SVM, GA-SVM et ACO-SVM sont respectivement 97.3%, 97.19% et 95.38%. Ces résultats ont réduit considérablement le temps de calcul d'apprentissage et ont montré que le GA-SVM était meilleur que les autres.

Dans le travail de [71], une comparaison de diverses techniques d'extraction d'attributs comme F-Score, Algorithme génétique, K-means, ReliefF et SVM-RFE (Support Vector Machine-Recursive Feature Elimination) est présentée.

Les meilleures attributs sélectionnées par ces techniques sont utilisées par le classificateur SVM pour classer l'ensemble de données WDBC. Dans cette recherche, la performance de SVM est mieux améliorée en utilisant SVM-RFE qui est une approche intégrée. Cette dernière permet d'éliminer récursivement les attributs non importants.

Dans le travail de [72], les auteurs ont proposé un modèle hybride qui utilise l'algorithme génétique et le réseau de propagation du retour (GA-BPN). Dans ce modèle GA permet d'initialiser et d'optimiser les poids de connexion de BPN. Ce modèle hybride a été identifié en utilisant deux méthodes : arbre de décision et méthode GA-CFS. Cette méthode de sélection d'attributs a amélioré la précision de la classification de GA-BPN (77,70%) pour le Pima Indian Diabetes dataset.

Dans le travail de [73], un AG a été utilisé comme méthode de sélection d'attributs et le J48graft Decision Tree (J48graft DT) a été utilisé pour la classification et la prédiction d'attributs sélectionnés. La méthode proposée a été utilisée pour la classification de la maladie du diabète. Il minimise le coût de calcul, le temps de calcul et maximise le ROC. Les résultats expérimentaux ont obtenu une précision de classification de 74,78%.

Dans le travail de [74], une programmation génétique avec KNN(GP-KNN) et la programmation génétique avec SVM (GP-SVM) sont introduites. Cette contribution consiste à utiliser GP pour la sélection d'attributs, puis les attributs sélectionnés sont utilisés comme entrée aux classificateurs KNN et SVM pour la classification du diabète. L'évaluation et la comparaison de ces méthodes montrent que l'approche GP-KNN offre une meilleure précision (78,5%).

Dans le travail de [75], un algorithme hybride appelé (GA-SVM) a été implémenté, cet algorithme combine l'algorithme génétique avec SVM. Le modèle proposé est utilisé pour optimiser les valeurs des paramètres de SVM et pour obtenir le sous-ensemble optimal d'attributs. Ces attributs sélectionnés sont utilisés pour la classification. Des expériences ont été effectuées sur cinq datasets de l'UCI.

Les résultats expérimentaux montrent que le GA-SVM peut avoir de meilleures performances en précision de classification par rapport aux performances du classificateur SVM seul.

Dans le travail de [76], un filtre et une wrapper méthodes de sélection d'attributs ont été proposés. Dans l'approche filtre, les auteurs utilisent l'optimisation de l'essaim des particules (PSO) avec la fonction de sélection de d'attributs (CFS) comme fonction de fitness et dans l'approche wrapper, ils utilisent comme fonction de fitness la précision de cinq classificateurs différents : l'arbre de décision, Naïve Bayes, Bayesian, fonction de base radiale et k-plus proche voisin. Les sous-ensembles d'attributs sélectionnés sont utilisés comme entrée à ces cinq classificateurs. Des expériences ont été effectuées sur trois datasets de UCI : Breast Cancer, Heart Statlog et Dermatology dataset. Les résultats montrent que la précision de classification accrue pour tous les types de données utilisés et une meilleure précision à été obtenus par PSO-Bays (99,45%).

Dans le travail de [77], les auteurs proposent deux techniques : le score (SNR) et l'optimisation des essaims de particules (PSO). Dans la première étape, ils utilisent k-means pour regrouper l'ensemble de données et dans chaque cluster, Chaque gène a été classé par son score SNR. Dans la deuxième étape, ces gènes sont utilisés comme entrées pour le PSO. Les sous-ensembles d'attributs sélectionnés sont utilisés comme entrée à trois classificateurs : un SVM, le k-NN et le réseau de neurones probabilistes (PNN). Les résultats expérimentaux montrent que l'algorithme proposé est plus efficace et donne de meilleurs résultats que les autres travaux cités dans la littérature. Pour l'ensemble de données du colon, la précision des différents classificateurs est k-NN (99,44%), SVM (97,50%) et PNN (96,00%).

Dans le travail de [78], un nouveau framework hybride (NHF) a été proposé en combinant le gain d'information (GI), le F-score, l'AG, le PSO et le SVM. Le projet comporte trois étapes. Dans la première étape, GI est utilisé pour construire une liste de classement des caractéristiques.

Dans la deuxième étape, PSO effectue la tâche de sélection de fonctionnalité combinant SVM, le F-score est considéré comme une partie de la fonction objective de PSO. Les sous-ensembles des attributs sont filtrés selon la liste de classement à partir de la première étape, puis les résultats sont fournis à l'initialisation de GA. Dans la troisième étape, GA initialise l'individu de la population à partir des résultats de la deuxième étape. Leur méthode a été comparée avec PSO, GA, ACO et le recuit simulé (SA) sur cinq ensembles de données de référence : la leucémie, les cancers du poumon, du côlon, du sein et du cerveau.

Dans le travail de [79], les auteurs proposent une nouvelle méthode qui utilise l'optimisation des essaims de particules pour la sélection d'attributs et arbre de décision pour la classification. Les auteurs comparent les performances de cette méthode avec d'autres méthodes de classification de référence bien connues (SVM, carte auto-organisée, réseau neuronal de rétro-propagation, arbre de décision C4.5, Naive Bayes, arbre de décision CART et système de reconnaissance immunitaire artificiel) et effectuent des expériences sur onze datasets. Par conséquent, l'approche proposée a obtenu une précision plus élevée par rapport à toutes les autres méthodes.

Le travail présenté dans [80] utilise un algorithme mimétique basé sur les couvertures de Markov. Il est le seul travail trouvé dans la littérature qui utilise un algorithme mimétique pour faire la sélection de gènes. Ce modèle emploie la technique de bootstrap pour valider ses résultats. La comparaison démontre que ces résultats sont dominants pour le taux de classification et pour le nombre de gènes sélectionnés.

Les auteurs de [81] présentent deux modèles de sélection : une optimisation par essaims particulier et un algorithme génétique. Tous les deux s'appuient sur un classificateur SVM. Pour les jeux de données de la leucémie, du sein, du poumon et de l'ovaire. Ces deux modèles obtiennent des taux de classification supérieurs avec des nombres de gènes sélectionnés plus grands.

Dans le travail de [82], ils ont proposé un modèle hybride qui a intégré l'ensemble brut avec l'optimisation par essaim particulier (PSO) pour la classification des données médicales. Dans cette étude, pour trouver l'ensemble minimal d'attributs, les auteurs utilisent la réduction relative d'attribut basé sur PSO et l'algorithme de réduction rapide basé sur PSO. Les résultats obtenus ont été utilisés comme entrée au différents classificateur. La meilleure précision était de 88% avec le classificateur Naïve Bayes pour le dataset SPECTF.

Dans le travail de [83], les auteurs ont proposé un modèle hybride pour la sélection des attributs. Leur hybridation se base sur la combinaison de l'algorithme de Fire fly avec roughset pour la sélection des attributs et la logique floue à l'intervalle de type 2 pour la classification de heart disease dataset. Ce modèle a été comparée avec BPSO (binary particle swarm optimization), roughset et avec d'autre classificateurs tel que SVM, bayes naïves et KNN. Cette comparaison montre que ce modèle est meilleur que d'autres techniques en termes de précision, de temps de traitement et de vitesse de convergence avec un nombre d'attribut minimale. Mais le temps d'apprentissage du classificateur est élevé.

Dans le travail de [84], les auteurs ont proposé un modèle hybride pour trouver le meilleur sous-ensemble des attributs à l'aide de fuzzy rough set, l'ACO et PSO. Cette étude utilise la combinaison de fuzzy rough set avec KNN pour la classification. Différents tests comme le test Ztest, T-test, sign et test Wilcoxon ont été utilisés pour comparer les performances. Parmi ceux-ci, le test de Wilcoxon était meilleur que tous les autres tests. Des expériences ont été effectuées sur différents médicale datasets de UCI.

Le travail de [85] vise à développer un nouvel algorithme hybride pour la classification des données du diabète. Le nouvel algorithme est nommée (MPSO-NN), l'optimisation des essaims de particules est modifié et combiné avec le réseau de neurones multicouches Perceptron avec apprentissage par rétropropagation pour classer les patients diabétiques.

Une méthode de sélection d'attributs appelée Genetic-Relative Reduct Algorithm a été effectuée. Les auteurs n'expliquent pas comment cette méthode de réduction est appliquée. Les résultats obtenus montrent que La précision a été augmentée de 81,8% pour le PIMA diabète et de 84,2% pour US diabète. Les valeurs obtenues pour d'autres métriques supportent également l'algorithme MPSO-NN proposé dans la classification des maladies diabétiques.

Selon le travail de [86], plusieurs taxonomies ont été utilisé pour catégoriser les techniques de la sélection d'attributs. Pour classer les méthodes sélectionnées présentées ci dessus, nous choisissons la taxonomie suivante (voir [Tableau 2.1](#)) :

- La méthode de sélection fait référence à la manière dont les différentes méthodes interagissent avec l'ensemble complet des attributs afin d'effectuer la sélection. Les quatre catégories sont : Filtre , Wrapper, Embedded et hybride.
- Le type d'évaluation fait référence à la manière dont les attributs sont éliminés (Ranker et Subset)
- L'approche d'apprentissage consiste à déterminer si la sélection est basée sur la présence ou absence d'informations de classe (Supervisé, Semi-supervisé et Non-supervisé)
- La dimensionnalité de classe fait référence au nombre de classes traitées (Binaire, Multiple binaire et Multi-classes).

Tableau 2.1 : Synthèse de méthode de la sélection d'attributs

	La méthode de sélection				Le type d'évaluation		L'approche d'apprentissage			La dimension classe		
	Filtre	Wrapper	Embedded	hybride	Ranker	Subset	Supervisé	Semi-sup	non sup	Binaire	Multi-bin	Multi
[59]		X			X		X			X		X
[60]				X	X		X			X		X
[61]	X			X	X		X					X
[62]				X	X		X			X		X
[63]		X				X	X			X		X
[64]		X				X	X					X
[65]				X		X	X			X		
[66]							X					X
[67]		X				X	X			X		
[68]		X				X	X					X
[69]		X			X		X			X		
[70]		X				X	X			X		
[71]		X			X		X			X		
[72]				X	X		X			X		
[73]				X		X	X			X		
[74]				X		X	X			X		
[75]		X				X	X					X
[76]	X	X			X		X					X
[77]	X	X			X		X			X		
[78]				X	X		X					X
[79]		X				X	X					X
[80]				X	X		X					X
[81]				X		X	X					X
[82]				X		X	X					X
[83]				X	X		X			X		
[84]				X		X	X					X
[85]				X		X	X			X		

2.3 Les méta-heuristiques pour la classification

Afin de satisfaire les objectifs spécifiques de la classification, des méthodes plus spécifiques ont été proposées. Ces méthodes permettent d'améliorer la tâche de la classification en améliorant la précision et la compréhension du classificateur. Les méthodes méta-heuristiques qui nécessitent peu ou pas de connaissances générales du problème constituent des techniques intéressantes pour l'apprentissage des classificateurs, en raison de leur bonne performance et de leurs faibles besoins en matière de calcul.[45]

Dans la littérature, il existe différentes méthodes méta-heuristiques pour la classification parmi elles nous citons :

Dans le travail de [87], les auteurs ont proposé un algorithme combiné d'algorithme génétique (GA) et de réseau neurone artificiel pour la classification du WDBC dataset. L'opérateur de croisement et de mutation ont été modifié. Pour montrer que cette approche était supérieure à d'autres algorithmes, différentes mesures de performance ont été utilisées comme la précision, la sensibilité, la spécificité, la courbe de ROC, la matrice de confusion et l'AUC. Dans cet article, les données sont divisées en données d'apprentissage et de données de test avec quatre panneaux différents et une comparaison avec d'autres travaux a été effectuée. Ce modèle a la précision la plus élevée de 98,24%, 99,63%, 100% pour 50%-50%, 60%-40% et 70% de données d'apprentissage et 30% de données de test respectivement.

Dans le travail de [88], les auteurs proposent une classification en utilisant un algorithme inspiré de chauves-souris avec la variable de vol de prélèvement chaotique (Chaotic Levy Flights). Chaotic Levy Flights permet d'enrichir le comportement de recherche, de générer efficacement de nouvelles solutions, d'augmenter la diversité des solutions, d'éviter que l'algorithme de chauves-souris soit piégé dans un optimum local et d'augmenter les chances de trouver une solution globale optimale. L'algorithme proposé est appliqué sur Pima Indians Diabetes dataset.

Les résultats expérimentaux prouvent la supériorité de l'algorithme proposé par apport à l'algorithme de chauves-souris traditionnel ainsi que les différents classificateurs qui ont été implémentés sur le même ensemble de données et dans le même environnement.

Dans le travail de [89], une stratégie en deux phases est présentée comme suit : la première phase utilise une hybridation de l'algorithme génétique de Michigan et la programmation génétique pour produire des règles de classification sous la forme de : $IF X_1 \text{ et } X_2 \text{ et } \dots X_n \text{ THEN class} = Y$. l'AG est appliqué aux données numériques alors que le GP est appliqué aux données nominaux.

La deuxième étape consiste à appliquer Pittsburgh GA pour trouver une combinaison optimale des règles résultantes avec l'opérateur OR. La population est divisée en plusieurs sous-populations évoluant simultanément et correspondant à différents nombres de règles dans une seule solution (ensemble de règles).

Les résultats de l'application de l'algorithme ont été très encourageants et elles dépassent d'autres méthodes de classification comme l'algorithme DT C4.5 et les réseaux neuronaux. Cependant, le système est coûteux en termes de calcul et adaptable à la classification hors ligne.

Dans le travail [90], les auteurs proposent un modèle méta-heuristique hybride pour la classification des données médicales appelé GAColonies. Ce modèle se présente sous la forme d'une liste de décision. Le nom de l'hybride GAColonies, découle du fait que chaque individu de la population de GA est une liste de décision produite par une colonie de fourmis artificielle ACO.

Cette approche se compose de deux phases. Dans la première phase, le modèle de classification sous la forme d'une liste de décision est construit à l'aide d'ACO. Plusieurs modèles sont construits en utilisant différents sous-ensembles de données d'apprentissage. Dans la deuxième phase, ces modèles sont utilisés comme population initiale pour les AG qui optimisent les modèles existants au moyen de l'évolution naturelle. L'utilisation d'une solution initiale valide dans la phase des AG accélère la recherche et améliore la classification en termes de précision et de taille.

Cette approche est différente de l'approche ACO-GA présentée dans le travail de [91] où les auteurs combinent les arbres de décision (C4.5) avec les AG. L'hybridation proposée dans ce dernier travail traite des ensembles de règles plutôt que des listes de règles. Les expérimentations, qui se sont portées sur neuf datasets de UCI, montrent l'efficacité de l'approche proposée.

Dans le travail de [92], les auteurs montrent que la sélection de la bonne combinaison de méthodes de prétraitement a un impact considérable sur la classification. Les opérations de prétraitement considérées incluent la discrétisation des attributs numériques, la sélection du sous-ensemble d'attributs et la gestion des valeurs manquantes. La classification est effectuée par un algorithme d'optimisation des colonies de fourmis (ACO).

Les résultats expérimentaux sur 25 dataset médicales montrent qu'une amélioration significative de la précision est obtenue, cependant les auteurs ne précisent pas comment l'algorithme de classification fonctionne.

Dans le travail de [93], une approche de détection du cancer du sein à partir d'images de rayons X a été proposée où l'ensemble de données se caractérise par un déséquilibre élevé. Chaque patient est représenté par un ensemble de points de données. Le critère d'évaluation était de minimiser l'AUC par patient plutôt que par point de données afin de minimiser le sur-apprentissage.

Les auteurs utilisent quatre types de classificateurs : AdaBoost, SVM à base de classe (CB-SVM), SVM à base de patient (PB-SVM) et GA. Dans CB-SVM, l'intention était d'équilibrer les classes positives et négatives. Les patients avec moins d'instances positives étaient plus difficiles à identifier que ceux ayant des cas plus positifs.

Pour résoudre ce problème, PB-SVM a été conçu de telle sorte que la somme des poids des échantillons positifs était égale à la somme des poids échantillons négatifs. De plus, pour chaque patient 'i', la somme des poids d'échantillons positifs est égale à celle du patient 'j'. Une légère amélioration a été obtenue sur le CB-SVM.

Dans le travail de [94], Les auteurs introduisent une nouvelle méthode de diagnostic automatisé basée sur la structure organisationnelle générique des tissus de cellules cancéreuses à partir d'images BMP biopsies à basse résolution.

Deux phases sont mises en œuvre : la première consiste à construire le graphique cellulaire, cette étape transforme l'image BMP en un graphique monochrome ; où les nœuds correspondent aux cellules ou aux clusters de cellules en fonction de la résolution utilisée. Les bords sont attribués selon une probabilité basée sur la distance Euclidienne entre les nœuds. La deuxième phase est la coloration graphique en utilisant un nombre de couleurs, de sorte que les nœuds situés dans la même plage de distance euclidienne obtiennent la même couleur.

Les algorithmes Mimétique (AM) sont utilisés pour optimiser la coloration graphique. Dans ce travail, les AM sont composés d'une recherche heuristique, algorithme de coloration séquentielle et un algorithme génétique avec un opérateur de mutation modifié. La sortie du programme était le nombre de couleurs utilisées pour l'image échantillon. Les auteurs ne précisent pas comment le diagnostic cancéreux des cellules peut être tiré de cette information. Cependant, ils fournissent plus de formalisme sur la densité et les caractéristiques organisationnelles des cellules qui aident dans le processus de diagnostic.

Une hybridation de règle d'induction, recuit simulée et recherche Tabu pour la tâche de classification est présenté dans [95]. Les attributs continus ont été discrétisés. Les règles de classification sont créées de manière incrémentale pour une meilleure lisibilité et une meilleure valeur prédictive. La probabilité d'une addition d'un terme à une règle de classification dépend de l'entropie de la valeur de l'attribut par opposition à l'entropie dans les arbres de décision qui est calculé pour les attributs dans leur ensemble.

Les délais d'attente de la recherche de Tabu visent à réduire la probabilité qu'une valeur d'attribut particulière soit sélectionnée deux fois, augmentant ainsi la diversité de la recherche. La qualité d'une règle calculée comme le produit de la sensibilité et de la spécificité sert comme paramètre d'énergie pour le recuit simulée.

Une première température élevée en recuit simulée permet de recevoir des solutions de qualité inférieure au début pour une meilleure exploration de l'espace de recherche. Au fur et à mesure que la recherche se déroule et que la température se refroidit, seules les règles de haute qualité seront acceptées dans la liste des règles finales, ce qui intensifiera la recherche dans des zones prometteuses.

L'algorithme de classification proposé a été comparé à Ant Miner et à l'algorithme d'induction de règles CN2 sur quatre datasets médicales de UCI. En termes de précision prédictive, Ant Miner était en tête. Cependant, en termes de nombre de règles et de termes par règle, recuit simulée. Tabu Miner a réalisé de bons résultats qui dépassent CN2 et qui est très comparable à celui de Ant Miner.

Dans [96] les auteurs ont proposé une approche basée sur l'algorithme génétique pour découvrir des règles prédictives, complètes et compréhensibles. L'approche proposée est testée sur quatre bases de données d'UCI. Les expérimentations montrent que l'algorithme génétique proposé dans cet article est adapté à l'extraction de règles de classification et que les règles découvertes ont des performances de classification plus élevées pour l'ensemble de données Iris (98,85%). En termes de comparaison, les résultats obtenus par l'approche proposée sont meilleurs pour les différents datasets par rapport aux algorithmes C4.5 et DTGA.

Dans [97], les auteurs ont appliqué l'algorithme génétique pour découvrir des règles de prédiction pour la classification de dataset (Wisconsin Breast Cancer). Les résultats obtenus ont une précision de classification de (97,70%). Ces résultats sont meilleurs par rapport aux autres systèmes existants cités dans des travaux connexes.

Dans [98], les auteurs introduisent deux approches Particle Swarm Classifier (PSCClass) et Constructive Particle Swarm Classifier (cPSCClass). Ces approches sont basées sur PSO pour résoudre les problèmes de classification. Elles sont évaluées avec sept bases de données, leurs performances ont été comparées à celles d'autres algorithmes d'essaims ainsi qu'à certaines méthodes bien connues de la littérature, telles que k-NN, Bayes naïves

et un réseau de neurones MLP. Les algorithmes PSCClass et cPSCClass ont montré des résultats similaires entre eux pour toutes les bases de données évaluées, avec l'avantage d'avoir une architecture construite dynamiquement (nombre de particules dans l'essaim).

Dans [99], un algorithme PSO pour découvrir les règles de classification a été proposé. Cet algorithme adopte une méthode de codage réel, qui peut être appliquée à la fois aux attributs catégoriques et continus, comme démontré dans l'expérience sur les datasets Zoo (attributs catégoriques) et Wine (attributs continus). Les expériences montrent qu'il existe un petit nombre de conditions par règle et quelques règles par ensemble de règles, et montrent également que les règles ont de bonnes performances de précision prédictive sur des données continues.

Dans [100], les auteurs ont proposé un nouvel algorithme de découvrir des règles de classification basé sur l'algorithme PSO. L'approche proposée a été comparé avec les deux algorithmes de classification Ant-Miner et GA sur six ensembles de données. Les résultats expérimentaux montrent que l'algorithme proposé atteint une plus grande précision avec une liste de règles beaucoup plus petite que les autres algorithmes de classification.

Dans [101], différentes PSO approches ont été proposées pour les tâches de Data Mining. Dans la première phase, trois algorithmes différents d'exploration de données ont été implémentés et comparés empiriquement ses résultats avec un autre algorithme évolutif (GA) et avec J48. La deuxième phase a été consacrée à l'amélioration de l'une des PSO variantes en termes de type d'attribut et de complexité temporelle. Les résultats obtenus indiquent que les algorithmes d'exploration de données de PSO sont compétitifs, non seulement avec d'autres techniques évolutives, mais aussi avec l'algorithme J48, et peuvent être appliqués avec succès.

Dans [102], deux approches du PSO binaire sont appliquées aux problèmes de classification : l'une appelée PSO de Pittsburgh (PPSO) où chaque particule représente une

ou plusieurs règles de prédiction et l'autre appelée PSO Michigan (MPSO) où une seule particule est utilisée pour résoudre le problème de classification. La classification est effectuée en fonction de la règle des plus proches voisins (k-NN). Les deux classificateurs obtiennent de meilleures performances que de nombreuses techniques de classification. Le classificateur MPSO du Michigan présente des avantages évidents par rapport au PPSO en termes de taux de réussite et de vitesse.

Dans [103], les auteurs contribuent dans deux perspectives ; d'une part en fournissant un état de l'art de l'utilisation de ACO pour découvrir des règles de classification et d'autre part en fournissant une analyse des performances des approches de classification bio-inspirées en sélectionnant Ant-Miners (Ant_Miner, cAnt_Miner, cAnt_Miner2 et cAnt_MinerPB) pour découvrir les règles de classification en termes de précision, termes par règle, nombre de règles, temps d'exécution et taille de modèle découverts par l'algorithme d'exploration de règles correspondant. Un plus grand nombre de jeux de données (18 jeux de données publics disponibles sur le référentiel UCI) sont utilisés pour les expériences.

Dans [104], un GA et un ACO sont étudiés, dans le contexte de l'extraction de règles. Les auteurs tentent d'explorer ces algorithmes inspirés par la nature, leur fonctionnement et leurs applications dans le contexte de l'extraction de règles et de l'extraction d'exceptions. Ce modèle prend en compte les exceptions présentes dans l'ensemble de données. Les auteurs concluent que ces algorithmes sont assez efficaces, simples et capables de découvrir un modèle de classification comparable à un modèle conçu à l'aide des techniques traditionnelles.

Une amélioration d'un algorithme hybrides (PSO / ACO) pour découvrir des règles de classification a été proposée dans [105]. La nouvelle version de l'algorithme PSO / ACO appelée (PSO / ACO2), traite aussi directement des valeurs d'attributs continues et nominaux. Les auteurs évaluent (PSO / ACO2) en 27 dataset.

Les résultats montrent que PSO / ACO2 est au moins compétitif avec PART (algorithme

standard de l'industrie) en termes de précision, et que PSO / ACO2 produit des ensembles de règles significativement plus simples (plus petites) et montre également que la version PSO réduite pour les attributs continus fournit une légère augmentation de la précision par rapport à l'évolution différentielle des variantes.

Conclusion

Ce chapitre a pour objectif d'introduire les méta-heuristiques, de citer leurs enjeux et de présenter la classification de ses différentes méthodes. Avant de voir l'apport d'utilisation des méta-heuristiques pour résoudre les problèmes de la classification des données médicales, les caractéristiques des ensembles de données médicales, y compris le bruit, l'incomplétude et l'existence d'attributs multiples et éventuellement non pertinentes, doivent être abordées par l'application des méthodes méta-heuristique puisque la sélection d'attributs a un effet profond sur la performance de classificateur.

Les résultats de ces travaux sont prometteurs pour progresser de manière essentielle dans ce domaine. Le chapitre suivant est consacré aux méthodes hybrides basé sur les méta-heuristiques.

Approches proposées pour la sélection d'attributs et la classification

Introduction

Les méta-heuristiques inspirées de la nature ont démontré leur succès dans l'application et la résolution de plusieurs problèmes et applications. Dans notre travail, on utilise ces méta-heuristiques pour la classification qui est devenue un domaine hautement actif dans la recherche. La classification est une tâche complexe qui nécessite plusieurs étapes du prétraitement des données, l'une de ces étapes est la sélection des attributs.

Le présent chapitre est divisé en deux parties, la première partie est consacrée aux méta-heuristiques pour la sélection d'attributs où nous présentons deux nouveaux algorithmes adaptatifs. Le premier algorithme est nommé FS-PeSOA (Feature selection-Penguin Search Optimization Algorithm) et le deuxième algorithme est nommé FS-SLOA (Feature Selection-Seven Spot Ladybird Optimization Algorithm). Ensuite nous exposant des approches hybrides. L'aspect hybride réside dans la séquence d'une méthode de filtrage et d'une méthode méta-heuristique et la séquence d'une méthode méta-heuristique et d'une méthode de filtrage. Pour les méthodes de filtrage, on utilise la sélection d'attribut basée sur les corrélations(CFS), l'analyse en composantes principales(PCA) et X2 (CHI). Pour les méthodes méta-heuristiques on utilise l'algorithme génétique (AG), FS-SLOA et FS-PeSOA. Dans la deuxième partie, on s'intéresse à l'uti-

lisation d'une méthode méta-heuristique SLOA (Seven Spot Ladybird Optimization Algorithm) pour la classification, plus précisément pour découvrir des règles de classification. Enfin nous terminons par une conclusion.

3.1 Les approches proposées pour la sélection d'attributs

3.1.1 L'approche FS-PeSOA

L'approche FS-PeSOA (Feature selection-Penguin Search Optimization Algorithm) est une méthode de sélection des attributs basée sur une méthode méta-heuristique appelée PeSOA (Penguin Search Optimization Algorithm). PeSOA est développée par Youcef Gheraibia et Abdelouahab Moussaoui en 2013 pour résoudre les problèmes d'optimisation, elle est basée sur le comportement de chasse des pingouins. La stratégie de chasse des pingouins est un travail collaboratif de l'effort et le temps où les pingouins bénéficient de leurs plongées en optimisant l'énergie globale dans le processus de la chasse collective et de la nutrition [106].

3.1.1.1 Le comportement de chasse des pingouins

Le pingouin (voir [Figure 3.1](#)) est un oiseau de mer de la famille des alcidés. Il dispose des ailes qui lui permettent de nager. Les pingouins peuvent plonger à plus de 520 m pour la recherche de nourriture. Les pingouins se nourrissent de poissons et de calmars, ils doivent chasser en groupe et synchroniser leurs plongées pour optimiser la recherche de nourriture [107]. Les pingouins communiquent les uns avec les autres avec des vocalisations. Ces vocalisations permettent l'identification unique de chaque pingouin et la reconnaissance des pingouins par d'autre [108]. La quantité de la nourriture nécessaire pour un pingouin est variable selon les espèces, l'âge, la variété et la disponibilité alimentaire mesurée dans chaque région.



Figure 3.1 : Les pingouins

3.1.1.2 l'Algorithme de PeSOA

Les pingouins sont divisés en groupes et chaque groupe commence la recherche de nourriture dans une région spécifique. L'état de chaque pingouin est représenté par sa position et sa réserve d'oxygène. Après un certain nombre de plongées, le pingouin retourne à la surface et partage avec ses filiales du groupe la position et la quantité des aliments trouvés [106].

Soit $G = G_1, G_2 \dots G_K$ l'ensemble de groupes disjoints de pingouins répartis de façon aléatoire dans l'ensemble de l'espace de solution. Chaque groupe G_i contient d_j pingouins où chaque pingouin j de G_i est placé à une solution $x_j^i(t)$.

Mise à jour la solution PeSOA : Le pingouin j nage vers une nouvelle position à l'instant $t+1$ dans l'espace de recherche par l'Équation 3.1 :

$$x_j^i(t+1) = x_j^i(t) + 0_j^i(t)rand() | x_{localbest}^i - x_j^i(t) | \quad (3.1)$$

Où $x_j^i(t+1)$ représente la nouvelle solution, $x_j^i(t)$ représente la solution précédente, $0_j^i(t)$ représente la réserve d'oxygène, $rand()$ est un nombre aléatoire et $x_{localbest}^i$ représente la meilleur solution locale.

Les pingouins explorent le long du chemin guidé par la meilleur solution locale qui a trouvé la plupart des aliments dans la dernière plongée. La nage du pingouin est accélérée par la réserve d'oxygène qui reflète son état de santé déterminée par des plongées précédentes.

Mise à jour de la réserve d'oxygène : Après chaque plongée, la réserve d'oxygène du pingouin est mise à jour par l'Équation 3.2 :

$$O_j^i(t+1) = O_j^i(t) + (f(x_j^i(t+1)) - f(x_j^i(t))) * |x_j^i(t+1) - x_j^i(t)| \quad (3.2)$$

Où f est la fonction objectif. La réserve d'oxygène dépend à la fois de gain de la source de nourriture et de la durée de la natation de pingouin. Si le gain énergétique est positif, plus le pingouin reste sous l'eau, plus les quantités de nourriture qu'il attrape et il devient ainsi plus sain. Par conséquent, la réserve d'oxygène est mise à jour en fonction de l'amélioration de la fonction objectif.

La réserve d'oxygène augmente si la nouvelle solution est meilleure que la précédente, et la réserve d'oxygène diminue dans l'autre cas. Le pingouin effectue des plongées successives jusqu'à ce que l'oxygène soit épuisé, le pingouin va migrer vers un autre groupe en raison de la pénurie de nourriture dans son groupe.

Mise à jour de l'abondance alimentaire : Le degré d'abondance alimentaire associé à un groupe indique la teneur en énergie des proies capturées par tous les membres dans ce groupe. A la lumière du butinage des pingouins, le degré d'abondance de la nourriture peut être estimé par la quantité de poisson mangé (QEF), qui est calculée par l'Équation 3.3 :

$$QEF^i(t+1) = QEF^i(t) + \sum_{j=1}^{d_i} (O_j^i(t+1) - O_j^i(t)) \quad (3.3)$$

Le QEF d'un groupe représente l'attrait des pingouins de ce groupe. Une grande valeur de QEF signifie que la région offre suffisamment de nourriture pour tout le groupe et même sollicite les pingouins de migrer d'autres groupes.

La communication inter-groupe : Lorsque la réserve d'oxygène du pingouin est épuisée, il doit revenir à la surface pour respirer de l'air. Le pingouin communique avec les autres par la vocalisation avec le signal de QEF du groupe. Comme l'information se propage entre les groupes, le pingouin sur la surface a l'occasion d'apprendre le de-

gré d'abondance alimentaire de divers groupes. Il fait alors une décision de migrer à un autre endroit où le groupe de destination a un degré d'abondance élevée.

Mise à jour de la probabilité : Le pingouin peut migrer pour rejoindre un autre groupe en raison de la pénurie alimentaire dans le groupe qu'il appartient à l'origine. Le pingouin met à jour son appartenance à un groupe par référence à une fonction d'appartenance relative au degré d'abondance de divers groupes. Le pingouin rejoint un groupe avec une probabilité proportionnelle à la QEF du groupe correspondant, augmentant la probabilité de succès de la recherche de nourriture dans la prochaine plongée. D'autre part, la région explorée par un groupe est abandonnée si tous les membres de ce groupe ont migré vers d'autres groupes. La valeur de la fonction d'appartenance de se joindre au groupe j est une probabilité donnée par l'Équation 3.4 :

$$P^i(t+1) = QEF^j(t) / \sum_{j=1}^{nb_gr} (QEF^j(t)) \quad (3.4)$$

Le processus de collaboration se répète jusqu'à ce que le nombre d'itération soient atteints.

3.1.1.3 PeSOA pour la sélection des attributs

Les principales étapes à suivre dans l'application de l'algorithme Fs-PeSOA sont la sélection d'une représentation de solution appropriée (codage), une fonction d'évaluation adéquate (fitness function) et l'amélioration de cette solution.

Dans la première étape, nous définissons une solution comme un vecteur numérique, nous utilisons un vecteur de (0 et 1) avec 1 signifie que l'attribut est sélectionnée, et 0 sinon. Nous générons un nombre de pingouins en tant que population initiale. Dans la deuxième étape, nous évaluons la qualité de chaque solution candidate avec trois classificateurs dans chaque cas : NB, KNN, SVM. Dans la troisième étape, la position du pingouin qui représente une solution est mise à jour jusqu'à ce que sa valeur de fonction objective soit améliorée ou que les réserves d'oxygène soient épuisées.

Paramètres de FS-PeSOA : les différents paramètres utilisés dans l'algorithme FS-PeSOA sont donnés dans le [Tableau 3.1](#) :

Paramètre	Notation
Nb_gr	Nombre de groupe
Nb_peng	Nombre des pingouins (solutions) par groupe
Nb_feat	Nombre d'attributs désirés (la taille de chaque pingouin)
Nb_it	Nombre d'itérations
Max_it	Nombre d'itérations maximales
O	La réserve d'oxygène pour chaque pingouin
X_new	La nouvelle solution (position)
X_Last	La dernière solution
X_localBest	La meilleure solution de groupe
X_globalBest	La meilleure solution de la population
Rand()	Un nombre aléatoire

Tableau 3.1 : Les paramètres de FS-PeSOA

L'organigramme et le Pseudo code FS-PeSOA L'organigramme générale de notre algorithme de sélection d'attributs ([Figure 3.2](#)) et le pseudo code (algorithme 9) sont donnés ci-après :

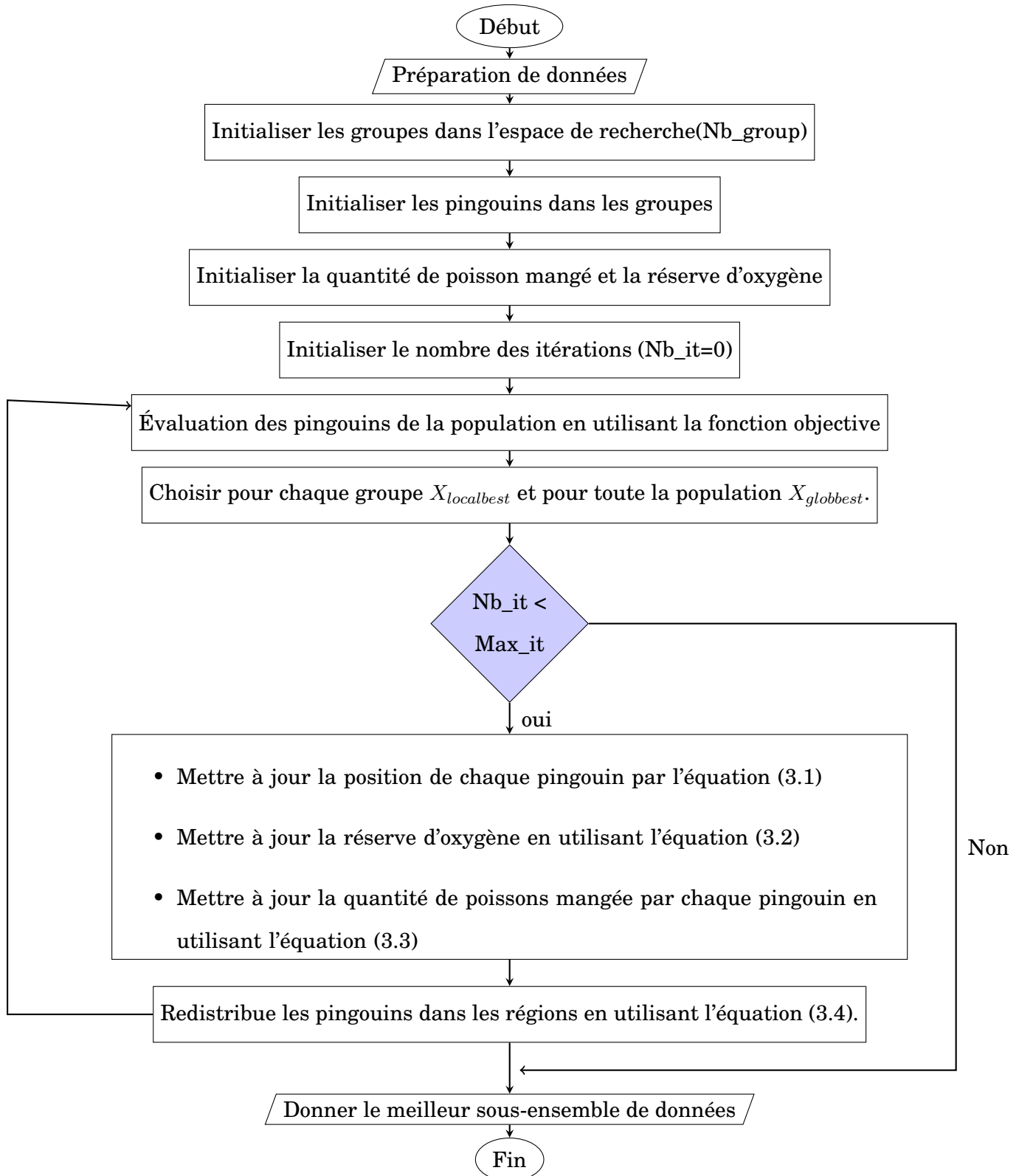


Figure 3.2 : L'organigramme générale de l'algorithme FS-PeSOA

Algorithm 9 Le Pseudo code de FS-PeSOA

- 1: Entrée : les données (tous les attributs), Nb_gr, Nb_peng, Max_it ;
 - 2: Sortie : sous ensemble données sélectionnées.
 - 3: Début
 - 4: Lire les données
 - 5: Préparation des données (supprime les valeurs manquantes et divise les données en un ensemble d'apprentissage et un ensemble de test)
 - 6: Générer un nombre (Nb_gr) de groupes dans l'espace de solution
 - 7: Générer une population de solutions aléatoire de solutions (Nb_peng) par groupes ;
 - 8: Initialisation de nombre maximal d'itération (max_ite)
 - 9: Calculer le meilleur individu de chaque groupe $X_{localbest}$
 - 10: Calculer le meilleur individu de la population $X_{globbest}$
 - 11: Tant que Nombre d'itération < max_ite faire
 - 12: Pour chaque groupe i faire
 - 13: Pour chaque pingouin j dans ce groupe faire
 - 14: Initialiser la réserve d'oxygène pour chaque pingouin
 - 15: Tant que (la réserve d'oxygène n'est pas épuisée et la position du pingouin n'est pas améliorée) faire
 - 16: Mettre à jour la position du pingouin en utilisant l'équation (3.1) ;
 - 17: Mettre à jour la meilleure solution locale $X_{localbest}$ si $X_{new} > X_{localbest}$;
 - 18: Mettre à jour les réserves d'oxygène en utilisant l'équation (3.2)
 - 19: Fin tant que, Fin pour
 - 20: Mettre à jour les quantités de poisson mangées pour ce groupe par l'équation (3.3)
 - 21: Fin pour
 - 22: Mettre à jour la meilleure solution globale $X_{globbest}$
 - 23: Mettre à jour les valeurs des fonctions d'appartenance pour chaque groupe par équation (3.4)
 - 24: Redistribuer les pingouins à des groupes en fonction de la fonction d'appartenance
 - 25: Abandonner le groupe s'il n'a pas de membres
 - 26: Fin tant que, Fin.
-

3.1.2 L'approche FS-SLOA

L'approche FS-SLOA (Feature Selection-Seven Spot Ladybird Optimization Algorithm) est une méthode de sélection des attributs par une approche méta-heuristique appelée SLOA (Seven Spot Ladybird Optimization Algorithm) développée par Peng Wang, Zhouquan Zhu, and Shuai Huang en 2013 [109] pour résoudre des problèmes d'optimisation numérique.

Cet algorithme s'inspire du comportement de fourragement chez les coccinelles à sept points.

Les études sur cette insecte (voir [Figure 3.3](#)) ont montré qu'elles sont plus sociales que nous le croyons qu'ils soient, car elles utilisent différents types de phéromones à différentes étapes de son cycle de vie, tels que les œufs, les larves, etc. Leurs comportements pour la recherche de nourriture ont été largement étudiés. Certains chercheurs ont classé les niveaux environnementaux des coccinelles à sept points en proie (prey), les pièces (patches) et les habitats (habitats).



Figure 3.3 : Les coccinelles à sept points

3.1.2.1 Algorithme d'optimisation SLOA

Le processus de fourragement s'organise autour de sept étapes :

1. Division de patches : Supposons que l'espace de recherche (environnement) est un espace de dimension N .
2. Initialisation de la population : Supposons que chaque coccinelle à sept points est traitée comme une solution. $X_{ij}(i = x_1, x_2, \dots, x_N)$.

3. Calcul de fitness : Pour chaque coccinelle calculer leur fitness dans N espace de dimension.
4. Choix de la meilleure solution : La valeur de fitness courante de chaque coccinelle est comparée à la valeur de sa meilleure position précédente. Si la valeur courante est meilleure que la précédente, la valeur de x est égale à la valeur courante, et la position est égale à la position courante.

La meilleure valeur de fitness de toutes les coccinelles dans un patch a été comparée à la valeur de fitness de leur meilleure position précédente (x_{lbest}). Si la valeur courante est meilleure que la précédente, la valeur x_{lbest} est égale à la valeur courante, et la position est égale à la position courante.

La meilleure valeur de fitness de toutes les coccinelles dans la population a été comparée à la valeur de fitness de leur meilleure position précédente (x_{gbest}). Si la valeur courante est meilleure que la précédente, la valeur (x_{gbest}) est égale à la valeur courante, et la position est égale à la position courante.

5. Dispersion : Dans SLOA, si une position ne s'améliore pas dans un nombre prédéterminé de cycles, une nouvelle position est produite dans le patch où il existe x_{gbest} , en remplaçant la position abandonnée.

La nouvelle position est produite près de x_{lbest} pour partager les informations de la meilleure coccinelle dans tout l'environnement. La valeur du nombre prédéterminé de cycles est un paramètre de contrôle important dans le SLOA.

Si la position est abandonnée, la coccinelle découvre une nouvelle position comme suit (voir [Équation 3.5](#)) :

$$X_{(i,j)} = X_{gbest} + rand() * w \quad (3.5)$$

Où w est un constant et rand() est un nombre aléatoire entre [0, 1].

6. mise à jour de solution : La position d'une coccinelle est mise à jour en fonction de son mouvement précédent. Si une coccinelle a fait une vaste recherche, la position

de la coccinelle est modifiée comme suit (voir [Équation 3.6](#), [Équation 3.7](#)) :

$$V_i(t) = c * r_1 * (X_{lbest}(t) - X_i(t)) + \epsilon_1 \quad (3.6)$$

$$X_i(t + 1) = X_i(t) + V_i(t) \quad V_i(t) < V_{max} \quad (3.7)$$

Après des recherches intensives, une coccinelle passe à une recherche extensives. La position est mise à jour selon les équations suivantes (voir [Équation 3.8](#), [Équation 3.9](#)) :

$$V_i(t) = c * r_2 * (X_{gbest}(t) - X_i(t)) + \epsilon_2 \quad (3.8)$$

$$X_i(t + 1) = X_i(t) + V_i(t) \quad V_i(t) > V_{max} \quad (3.9)$$

Dans [Équation 3.6](#) et [Équation 3.8](#), r_1 et r_2 deux nombres aléatoires réparties uniformément entre 0 et 1 et la constante positive c est utilisée pour le réglage de l'étape de recherche et de la direction de recherche dans chaque itération.

En [Équation 3.7](#) et [Équation 3.9](#), les vitesses des coccinelles dans chaque dimension est limitée à la vitesse maximale, ce qui détermine la précision de la recherche des coccinelles dans un espace de solution. Si V_{max} est trop élevée, les coccinelles vont probablement voler au-dessus de la solution optimale. Toutefois, si V_{max} le est trop faible, les coccinelles vont tomber dans l'espace de recherche local et n'auront aucune méthode pour poursuivre la recherche globale. En règle générale, V_{max} est fixé comme suit (voir [Équation 3.10](#)) :

$$V_{max} = 2 * (ub - lb) \quad (3.10)$$

Où ub et lb sont les limites supérieure et inférieure de chaque patch, respectivement. Les paramètres ϵ_1 et ϵ_2 sont généralement définis comme petits nombres aléatoires.

7. condition d'arrêt : Si le nombre d'itération maximal est atteint, le SLOA est terminé, sinon on retourne à 3.

3.1.2.2 SLOA pour la sélection d'attributs

Codage : La clé la plus importante dans l'application de tout algorithme méta-heuristique est la sélection d'une représentation appropriée de la solution qui a appelé le codage.

Dans notre approche FS-SLOA, une solution est généralement codée sous forme d'une chaîne binaire de longueur fixe qui représente un sous-ensemble d'attributs (voir [Figure 3.4](#)). La valeur de chaque position dans la chaîne représente la présence ou l'absence d'un attribut particulier. la valeur 1 signifie que l'attribut correspondante est sélectionnée tandis que 0 signifie qu'il n'est pas sélectionnée. Chaque position est un sous-ensemble d' attributs.

0	1	1	1	0	1	1	0	1	1
---	---	---	---	---	---	---	---	---	---

Figure 3.4 : La représentation d'une solution

Population initiale l'algorithme génère une population initiale de solutions distribuées de façon aléatoire. Cette étape consiste généralement à représenter chaque ensemble d'attributs du dataset par un vecteur.

Afin de commencer la recherche avec un ensemble de solutions initiales diversifiées, nous générons une population initiale de coccinelles réparti dans différents patches. Chaque patch est placé dans une région séparée avec une distance minimale à toute autre, cette stratégie de génération (Algorithme 10) est proposée à l'origine dans la recherche par dispersion [5].

Algorithm 10 L'algorithme d'initialisation

- 1: Entrée : Espace de solution, N (nombre de patches), MaxDist (distance inter-patch minimale).
 - 2: Sortie : N région centre dans l'espace de solution
 - 3: choisir aléatoirement le centre du premier patch, noté C_0
 - 4: $i = 1$
 - 5: Tant que $i < N$ fais
 - 6: Choisir un centre C_i au hasard pour le groupe suivant
 - 7: $j = 0$
 - 8: Tant que $j < i$ fais
 - 9: si Distance $(C_i, C_j) > \text{MaxDist}$ alors
 - 10: $j = j + 1$
 - 11: sinon allez à l'étape 6
 - 12: fin si
 - 13: fin tout en
 - 14: $i = i + 1$
 - 15: fin tandis que
 - 16: Fin.
-

La fonction d'évaluation : Pour évaluer la qualité de chaque solution, trois classificateurs ont été utilisés séparément : NB, KNN et SVM.

Amélioration de solution :

- Améliorer la position de chaque coccinelle en utilisant l'équation [Équation 3.7](#), [Équation 3.9](#).
- Mettre à jour la meilleure solution de chaque patch.
- Mettre à jour la meilleure solution de la population.

Paramètres de FS-SLOA pour la sélection d'attributs : Dans cette partie, nous définissons les paramètres utilisés pour l'algorithme FS-SLOA pour la sélection d'attri-

buts (voir [Tableau 3.2](#)).

Paramètre	Notation
Nb_Coc	Nombre de coccinelle
Nb_patch	Nombre de patch
Nb_it	Nombre d'itération fixé au début
Max_it	Nombre d'itération maximale
X(t+1)	La nouvelle solution (position)
X_gbest	La meilleure solution de la population
X_lbest	La meilleure solution locale (patch)
X(t)	La dernière solution
r_1 et r_2	Un nombre aléatoire entre [0,1]
C, w	Constants positif
ϵ_1 et ϵ_2	Petits nombres aléatoires

Tableau 3.2 : Les paramètres de FS-SLOA

Le Pseudo code FS-SLOA : Le pseudo code de l'algorithme de sélection d'attributs FS-SLOA est donné ci-après :

Algorithm 11 Le Pseudo code de FS-SLOA

- 1: Entrée : les données (tous les attributs), Nb_patch, Nb_coc, Max_it ;
 - 2: Sortie : sous ensemble données sélectionnés.
 - 3: Début
 - 4: Lire les données
 - 5: Préparation des données (divise les données en un ensemble d'apprentissage et un ensemble de tests)
 - 6: Générer un nombre (Nb_patch) de patches dans l'espace de solution
 - 7: Générer un nombre (Nb_coc) des coccinelles pour chaque groupe
 - 8: Générer une population de solutions P dans les patchs
 - 9: Initialisation du nombre maximal d'itération (Max_ite)
 - 10: Calculer le meilleur individu de chaque patch X_lbest
 - 11: Calculer le meilleur individu de la population X_gbest
 - 12: Tant que Nombre d'itération < Max_ite faire
 - 13: Pour chaque patch i faire
 - 14: Pour chaque coccinelle j dans ce patch faire
 - 15: Améliorer la position de coccinelle x_{ji} avec l'équation 3.7 et l'équation 3.9
 - 16: Choisissez la meilleure solution de x_{ji}
 - 17: Si (fitness (x_{ji}) > fitness (X_lbest))
 - 18: X_lbest = x_{ji}
 - 19: Fin tant que
 - 20: Fin pour
 - 21: Fin pour
 - 22: Calculer meilleur individu pour chaque patch X_glbest
 - 23: Calculer meilleur individu de la population X_gbest
 - 24: Si (fitness (X_lbest) > fitness (X_gbest))
 - 25: X_gbest = X_lbest
 - 26: Fin si
 - 27: Fin tant que
 - 28: Fin.
-

3.1.3 Les approches hybrides

Les approches hybrides sont des méthodes de résolution qui ont montré des avenues très prometteuses au cours de ces dernières années. Ces algorithmes hybrides ont permis d'obtenir des résultats très intéressants dans une grande variété de problèmes.

Plusieurs recherches ont introduit l'hybridation de méta-heuristiques avec d'autres méthodes. C'est ainsi l'objectif pour notre travail est d'étudier l'apport de l'hybridation entre les méthodes méta-heuristiques et les méthodes filtres dans le domaine de la sélection des attributs. Ces méthodes ont des particularités bien différentes qui peuvent être associées pour produire de meilleurs résultats. L'aspect hybride réside dans la séquence d'une méthode de filtre et d'une méthode méta-heuristique et la séquence d'une méthode méta-heuristique et d'une méthode de filtre.

Nous choisissons trois méthodes de filtre, à savoir la sélection de caractéristique basée sur la corrélation (CFS), l'analyse en composantes principales (PCA) et les statistiques X² (CHI), et pour les méthodes méta-heuristiques, nous avons opté pour l'algorithme génétique (AG), l'approche FS-PeSOA (Feature Selection-Penguin Search Optimization Algorithm) et l'approche FS-SLOA (Feature Selection-Seven Spot Ladybird Optimization Algorithm). Ces trois méthodes sont des méthodes wrapper. Chaque méthode est mise en œuvre individuellement afin d'utiliser les avantages d'une méthode pour compléter les faiblesses d'une autre, De cette manière, on exploite pleinement la puissance de différentes méthodes. Les sous-ensembles des attributs obtenus ont ensuite été utilisés comme entrées pour différents classificateurs, y compris Multilayer Perceptron, Bagging (SVM) et Boosting.

3.1.3.1 Méthodes de filtrage

Dans notre recherche, nous avons choisi trois méthodes de filtrage :

- Le filtre CFS (Correlation Feature Selection) [110] est un algorithme qui évalue deux aspects d'un sous-ensemble d'attributs : sa capacité à prédire la classe et la corrélation entre les attributs du sous-ensemble.

- La PCA est une méthode d'analyse de données multivariées. cette technique statistique linéaire projective non supervisée de réduction de dimensionnalité introduite par Pearson [111] utilisée pour convertir un ensemble d'entités original en un autre ensemble de caractéristiques de dimension inférieure.
- Chi-Square (X_2) est une méthode basée sur la théorie de statistique proposée par le statisticien Karl Pearson en 1900 [111] [112], il évalue les attributs en estimant l'association entre l'attribut et une classe donnée.

3.1.3.2 Méthodes métaheuristiques

La sélection des attributs est connue sous le nom de problème NP-difficile, différentes stratégies de méta-heuristiques sont appliquées pour trouver le meilleur sous-ensemble d'attributs possible. Dans nos approches, les méta-heuristiques étudiées sont :

- L'algorithme génétique est un algorithme heuristique majeur qui imite la théorie de l'évolution de Darwin [55].
- FS-SLOA est une méthode de sélection d'attributs méta-heuristiques développée par Noria Bidi et Zakaria Elberrichi (Bidi et Elberrichi, 2017) [113].
- FS- PESOA est une méthode de sélection d'attributs méta-heuristiques développée par Noria Bidi et Zakaria Elberrichi (Bidi et Elberrichi, 2017) [114].

3.1.3.3 Méthode métaheuristique/filtre

Dans ces méthodes, nous commençons par le chargement du dataset. Puis l'application d'une seule méthode méta_heuristique. Ensuite le sous-ensemble d'attribut sélectionné sera utilisé comme entrée pour la méthode filtre. Finalement, le meilleur sous ensemble global est retenu comme le meilleur sous ensemble d'attributs de l'algorithme.(voir [Figure 3.5](#)). Ces méthodes hybrides sont :

- Algorithme FS-PESOA/CFS
- Algorithme FS-PESOA/PCA

- Algorithme FS-PESOA/ (X_2)
- Algorithme FS-SLOA/CFS
- Algorithme FS-SLOA/PCA
- Algorithme FS-SLOA/ (X_2)
- Algorithme génétique/CFS
- Algorithme génétique/PCA
- Algorithme génétique/ (X_2)

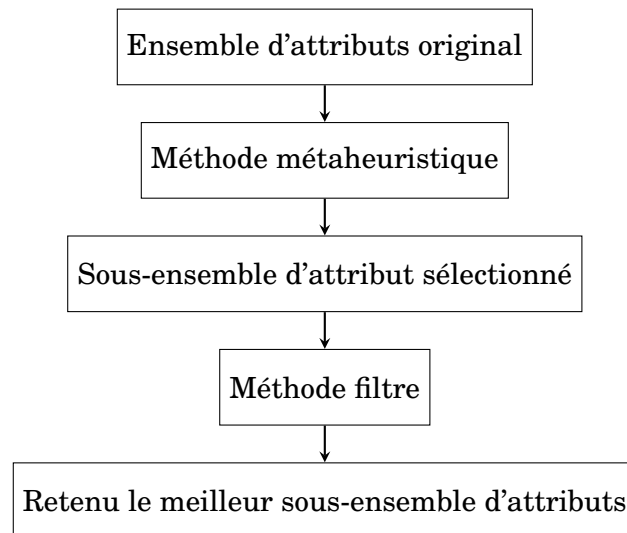


Figure 3.5 : Les méthodes métaheuristiques/filtres

3.1.3.4 Méthodes filtres/métaheuristiques

Dans ces méthodes, nous commençons par le chargement du dataset. Puis l'application d'une seule méthode filtre. Ensuite le sous-ensemble d'attribut sélectionné sera utilisé comme entrée pour la méthode méta_heuristique. Finalement, le meilleur sous ensemble global est retenu comme le meilleur sous ensemble d'attributs de l'algorithme.(voir [Figure 3.6](#)). Ces méthodes hybrides sont :

- Algorithme CFS/FS-PESOA

- Algorithme PCA/FS-PESOA
- Algorithme (X_2) /FS-PESOA
- Algorithme CFS/FS-SLOA/CFS
- Algorithme PCA/FS-SLOA/PCA
- Algorithme FS-SLOA/ (X_2)
- Algorithme CFS/Algorithme génétique
- Algorithme PCA/Algorithme génétique
- Algorithme (X_2) /Algorithme génétique

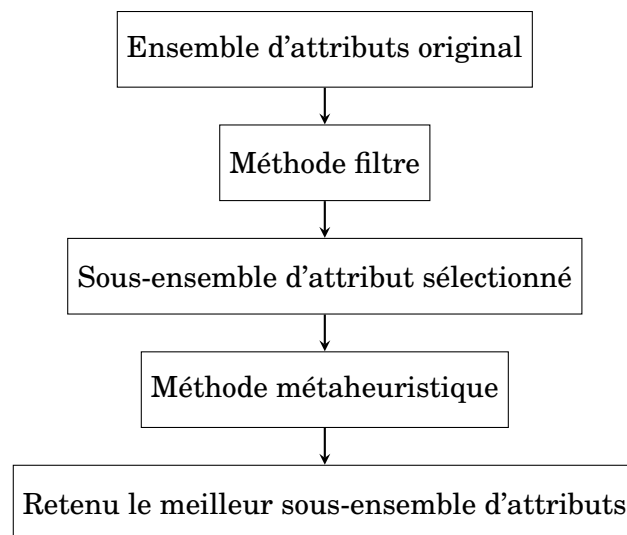


Figure 3.6 : Les méthodes filtres/métaheuristiques

3.2 L'approche Proposée pour la classification

3.2.1 Classificateur basée sur SLOA

La classification consiste à classer des données, dans différentes catégories connues à l'avance. Le problème consiste à découvrir des règles de classement qui permettent de ranger une donnée dans telle ou telle catégorie. Il sera ensuite possible d'utiliser ces

règles pour classer les données futures.

Les règles de classification utilisent des représentations de type : Si conditions, alors conclusion. La représentation de connaissance sous forme de règles a l'avantage d'être intuitivement compréhensible pour l'utilisateur, car l'objectif général de l'exploration de données est de découvrir des connaissances qui ne sont pas seulement précis, mais aussi compréhensible.

Pour découvrir des règles de classification, nous avons deux approches [115] : l'approche Michigan et l'approche Pittsburgh. L'approche du Michigan, initialement appliquée aux classificateurs systèmes, qui évalue une règle individuelle. L'approche de Pittsburgh évalue l'ensemble de règles entiers et non une règle individuelle.

Dans notre travail, nous proposons une adaptation de l'algorithme SLOA pour la classification. cette approche méta-heuristique qui est utilisée pour découvrir des règles de classification basée sur l'approche du Michigan, où chaque solution code une seule règle de prédiction, et les interactions entre ces règles ne sont pas prises en compte. Ces solutions sont plus simples et syntaxiquement plus courtes, ce qui tend à réduire le temps. Donc, pour découvrir les règles qui décrivent nos données, l'algorithme doit être exécuté plusieurs fois, si nous voulons prédire des classes différentes.

3.2.1.1 Représentation des règles

Les règles sont exprimées sous forme : Si condition Alors conclusion. La partie Si de la règle est appelée antécédent de la règle, ou la condition préalable. La partie Alors de la règle est appelée conséquence de la règle, ou la prédiction de la classe. Pour représenter cette partie, nous associons tous les individus de la population avec la même classe prédite, qui n'est jamais modifiée pendant l'exécution de l'algorithme.

Dans la partie antécédente, les conditions sont une conjonction de tests logiques simples décrivant les propriétés des instances à satisfaire. Chaque attribut contient un ensemble de valeurs de domaine distinctes. Pour décrire les valeurs des attributs, nous utilisons

le quartile, qui divise la gamme de valeurs en quatre parties ayant chacune 25% des valeurs. Q1 est le 25%, Q2 est le point de 50% et Q3 est le point de 75%.

L'espace de recherche est défini par l'ensemble de données. Si l'ensemble de données a N attributs ; l'espace de recherche est N-dimensionnel. De plus, chaque dimension est définie comme la valeur de chaque attribut. La valeur de l'attribut est identifiée par (1) si elle est inférieure au premier quartile, (2) si elle est inférieure à la médiane, (3) si elle est inférieure au deuxième quartile et (4) si elle est supérieure à le troisième quartile. Pour les cas dans lesquels l'attribut n'apparaît pas dans une règle, nous utilisons le (0).

3.2.1.2 Population initiale

La population est initialisée en générant au hasard des coccinelles. Nous avons pris trois ensembles de données du UCI. Ces ensembles de données contiennent des données continues.

3.2.1.3 Fonction d'évaluation

Pour évaluer la qualité de chaque solution candidate (règle) dans la population définit la fonction de fitness comme suivante (voir [Équation 3.11](#)) :

$$Se = vp + vn / (vp + fn + vn + fn) \quad (3.11)$$

- Vrai positif (vp) : l'instance respecte la règle et la classe réelle qui correspond à la classe donnée.
- Faux positif (fp) : l'instance respecte la règle et la classe réelle ne correspond pas à la classe donnée.
- Vrai négatif (vn) : l'instance ne respecte pas la règle et la classe réelle qui ne correspond pas à la classe donnée.
- Faux négatif (fn) : l'instance ne respecte pas la règle et la classe réelle qui correspond à la classe donnée.

3.2.1.4 Amélioration :

- Améliorer la position de chaque solution.
- Mettre à jour la meilleure solution de chaque patch.
- Mettre à jour la meilleure solution de la population.

3.2.1.5 Le Pseudo code SLOA pour la classification

Le pseudo code de l'algorithme SLOA pour la classification est donné ci-après :

Algorithm 12 le Pseudo code SLOA pour la classification

- 1: Entrée : les données, $R = \emptyset$, Nb_patch, Nb_coc, Max_it;
- 2: Sortie : ensemble de règle R.
- 3: Début
- 4: Lire les données
- 5: Préparation des données (supprime les valeurs manquantes et divise les données en un ensemble d'apprentissage et un ensemble de tests)
- 6: Générer un nombre (Nb_patch) de patches dans l'espace de solution
- 7: Générer un nombre (Nb_coc) des coccinelles pour chaque groupe
- 8: Générer une population de solutions P dans les patches
- 9: Initialisation de nombre maximal d'itération (Max_ite)
- 10: Calculer meilleur individu de chaque patch X_lbest
- 11: Calculer meilleur individu de la population X_gbest
- 12: Tant que Nombre d'itération < Max_ite faire
- 13: Pour chaque patch i faire
- 14: Pour chaque coccinelle j dans ce patch faire
- 15: Améliorer la position de coccinelle x_{ji} avec l'équation 3.7 et l'équation 3.9
- 16: Choisissez le meilleure solution de xji
- 17: Si (fitness (x_{ji}) > fitness (X_lbest))
- 18: X_lbest = x_{ji}
- 19: Fin tant que
- 20: Fin pour
- 21: Fin pour
- 22: Calculer meilleur individu pour chaque patch X_glbest
- 23: Calculer meilleur individu de la population X_gbest
- 24: Si (fitness (X_lbest) > fitness (X_gbest))
- 25: X_gbest = X_lbest
- 26: Définir $R = R \cup$ meilleure solution ;
- 27: Élaguer l'ensemble de règles en éliminant la règle redondante
- 28: Décoder les règles les plus intéressantes.
- 29: Fin si
- 30: Fin tant que
- 31: Fin.

Conclusion

Ce chapitre a pour objectif d'exposer des approches que nous proposons d'appliquer pour la sélection d'attribut et la classification. Pour la sélection d'attribut, nous avons proposé l'approche FS-PeSOA, l'approche FS-SLOA et des approches hybrides. Pour la classification, nous avons proposé la méta-heuristique SLOA comme classificateur. Enfin nous terminons par la conclusion.

Résultats et expérimentations

Introduction

Le présent chapitre expose les résultats des expérimentations des approches présentées dans le chapitre précédent. Il est divisé en deux parties, la première partie est consacré à exposer les résultats des expérimentations des méthodes de la sélection d'attributs (FS-PeSOA, FS-SLOA et les approches hybrides). La seconde partie expose les résultats des expérimentations de la méthode méta-heuristique pour la classification. Enfin nous terminons par une conclusion.

4.1 Critères d'évaluations

L'efficacité des approches proposées est mesurée par deux métriques d'évaluation bien connues et très utilisées qui sont l'exactitude (Accuracy) et la F-mesure.

- accuracy est la proportion du nombre de tuples positifs et de tuples négatifs obtenus par les algorithmes de classification dans le nombre total d'occurrences, comme indiqué dans l'Équation 4.1.

$$Accuracy = ((TP + TN)) / ((TP + TN + FP + FN)) \quad (4.1)$$

TP, TN, FP et FN désignent les nombres respectivement vrais positifs, vrais négatifs, les faux positifs et faux négatifs, respectivement.

- La F-mesure est la moyenne harmonique de la précision et du rappel, comme le montre l'Équation 4.2.

$$F1 - measure = 2 \frac{Precision Recall}{Precision + Recall} \quad (4.2)$$

Precision est le nombre de vrais positifs divisé par le nombre de vrais positifs et de faux positifs, comme indiqué dans l'Équation 4.3.

$$Precision = TP / (TP + FP) \quad (4.3)$$

Recall est le nombre de vrais positifs divisé par le nombre de vrais positifs et le nombre de faux négatifs, comme indiqué dans l'Équation 4.4.

$$Recall = TP / (TP + FN) \quad (4.4)$$

Nous utilisons le processus de validation croisée pour évaluer les performances. L'algorithme de validation croisée à k blocs (k-fold cross validation) consiste à découper l'ensemble initial d'exemples D en k blocs. On répète alors k phases d'apprentissage-évaluation, où une hypothèse h est obtenue par apprentissage sur (k-1) blocs de données et testée sur le bloc restant. L'estimateur de l'erreur est obtenu comme la moyenne des k erreurs empiriques ainsi obtenues.

4.2 Données de l'UCI

L'Université de Californie à Irvine (UCI) offre à la communauté de fouille de données une collection de bases de données. L'archive a été créée comme une archive ftp en 1987 par David Aha et autres étudiants des cycles supérieurs à l'UCI [116]. Nous donnons ci-dessous les caractéristiques des jeux de données que nous avons choisis pour la validation de notre algorithme.

Six ensembles de données de référence (Wisconsin Breast cancer, Pima Diabetes, Mammographic Mass, and Dermatology data sets) ont été obtenus à partir du référentiel d'apprentissage automatique (UCI) de l'UCI, comme indiqué dans le Tableau 4.1.

Datasets	Attributs	Instances	Classe	Valeur manquante
Wisconsin Breast cancer	32	569	2	No
Pima Indian Diabetes	8	768	2	Yes
Mammographic Mass	5	961	2	Yes
Dermatology	33	366	6	Yes
Colon tumor	2000	62	2	No
Prostate cancer	12600	21	2	No

Tableau 4.1 : Les caractéristiques de l'ensemble de données utilisés

- Wisconsin Diagnostics Breast Cancer (WDBC), il a été créé par William H. Wolberg à l'Université du Wisconsin. Cet ensemble de données comprend un total de 569 observations avec des cas bénins et malins, soit respectivement 357 et 212 observations. Pour chaque observation, il y a 30 variables en vedette. Ces caractéristiques sont calculées à partir d'images numériques de fines aiguilles (FNA) de masses mammaires.
- Le Pima Indian Diabetes contient 768 patterns avec 8 caractéristiques appartenant à deux classes (Not Diabetic, Diabetic). 268 échantillons ont été identifiés comme positifs et les 500 autres échantillons ont été identifiés comme négatifs. Les deux classes sont réparties de manière non-uniforme avec 34,9% des instances positives et 65,1% des instances négatives.
- Le masse mammographique, cette base de données contient une évaluation BI-RADS, l'âge du patient et trois attributs BI-RADS : masse, marge de masse, masse volumique) et repose sur une mammographie numérique recueillie à l'Institut de radiologie de l'Université d'Erlangen. Nuremberg entre 2003 et 2006 [49]. Il se compose de 961 enregistrements où 516 sont bénignes et 445 sont des masses malignes.
- Dermatology, cette base de données contient 34 attributs, dont sont linéaires et

l'un d'entre eux est nominal. Chaque échantillon de données dermatologiques est classé en six catégories : le psoriasis, la dermatite séboréique, le lichen plan, le pityriasis rosé, la dermatite chronique et le pityriasis rubra pilaris.

- Tumeur du côlon contient des échantillons d'expressions géniques qui ont été analysés avec un réseau d'oligonucléotides Affymetrix. Il contient 62 échantillons prélevés sur des patients atteints d'un cancer du côlon avec 2000 niveaux d'expression génique. Parmi eux, 40 biopsies de tumeurs proviennent de tumeurs et 22 biopsies (normales) proviennent de parties saines des colons des mêmes patients étiquetés comme négatifs et positifs respectivement.
- Le cancer de la prostate, l'ensemble de données sur le cancer de la prostate comprend 136 expériences de micropuces avec 12600 niveaux d'expression génique. Deux classes doivent être différenciées : tumeur avec 77 (52 + 25) échantillons et non tumorale avec 59 (50 + 9) échantillons.

4.3 Le matériel utilisé

Le matériel utilisé est un ordinateur portable avec processeur Intel® Core™ i3-3120 CPU @ 2.50GHz, 2.0GHz avec 4,00 GO de RAM, sous le système d'exploitation Windows.

4.4 Les approches proposées pour la sélection d'attributs

4.4.1 L'approche FS-PeSOA

4.4.1.1 Paramètres de FS-PeSOA

FS-PeSOA possède plusieurs paramètres, les valeurs de ces paramètres sont obtenues après plusieurs expériences sur chaque variable afin de trouver les meilleures valeurs. le [Tableau 4.2](#) présente les différents paramètres avec leur valeurs.

Paramètre	Valeur
Nb_peng	50
Nb_gr	10
Nb_feat	50% du nombre d'attributs maximale
Nb_it	100
Rand()	un nombre aléatoire

Tableau 4.2 : Les paramètres de FS-PeSOA

Afin de montrer l'amélioration de notre approche, nous avons comparé la performance de l'algorithme Fs-PeSOA avec les résultats des trois classificateurs de base, KNN, Naive Bayes et SVM qui sont utilisés dans l'ensemble de données sans sélection de caractéristiques et avec les résultats obtenus par l'algorithme génétique utilisé comme méthode de sélection des caractéristiques combinée avec ces trois classificateurs de base.

4.4.1.2 Paramètres de GA pour la sélection d'attributs

L'algorithme génétique a les paramètres de réglage présentés dans le [Tableau 4.3](#).

Paramètre	Valeur
Nombre d'attributs	50% du nombre d'attributs maximale
Taille de la population	5
Maximum nombre de générations	100
Sélection Tournoi (Taille du tournoi)	0.25
Taux de mutation	0.3
Taux de croisement	0.7

Tableau 4.3 : Les paramètres de GA

4.4.1.3 Résultats des expérimentations

Dans toutes les expériences, l'ensemble de données est divisé en deux sous-ensembles disjoints, à savoir l'ensemble d'apprentissage et de test en utilisant le schéma de validation croisée 10-fold. Le [Tableau 4.4](#) et le [Tableau 4.5](#) résument les résultats obtenus. A titre de comparaison, les précisions de classification obtenues avec la méthode proposée et d'autres approches existantes de la littérature sont présentées dans le [Tableau 4.6](#).

	SVM	SVM	SVM	KNN	KNN	KNN	NB	NB	NB FS-
		Genetic	FS-		Genetic	FS-		Genetic	PeSOA
			PeSOA			PeSOA			
WDBC	97.77%	97.71%	99.82%	92.91%	97.92%	98.24%	95.78%	96.22%	98.90%
Diabet	69.53%	77.03%	76.44%	67.05%	73.68%	83.67%	77.69%	76.47%	76.15%
Mammog	80.00%	79.99%	81.44%	79.71%	82.37%	81.44%	78.35%	80.15%	65.97%
Dermat	96.00%	96.70%	99.39%	92.98%	94.88%	98.48%	95.41%	98.33%	98.87%
Colon	81.42%	97.72%	100%	85.71%	89.23%	99.65%	67.14%	70.23%	71.43%
Prostate	87.13%	88.12%	100%	84.25%	85.66%	99.23%	66.32%	77.45%	98.88%

Tableau 4.4 : Les accuracies obtenus pour différent datasets

	SVM	SVM	SVM	KNN	KNN	KNN	NB	NB	NB FS-
		Genetic	FS-		Genetic	FS-		Genetic	PeSOA
			PeSOA			PeSOA			
WDBC	95.47%	96.12%	98.49%	91.87%	96.88%	97.15%	94.77%	95.28%	96.44%
Diabet	68.56%	76.83%	76.98%	76.75%	72.88%	79.92%	75.99%	85.32%	77.31%
Mammog	79.52%	77.43%	81.48%	78.65%	81.45%	80.97%	77.45%	79.10%	80.00%
Derma	94.04%	96.01%	98.96%	92.98%	93.88%	97.96%	93.84%	97.33%	98.61%
Colon	81.32%	97.42%	99.90%	84.41%	89.02%	99.51%	66.04%	70.01%	70.52%
Prostate C	87.21%	87.02%	99.89%	84.05%	84.78%	98.83%	65.33%	76.28%	97.98%

Tableau 4.5 : Les f-measures obtenus pour différent datasets

Dataset	Auteur	Année	Méthode	Accuracy
WDBC	Kaur et al	2016	SVM-RFE	97%
	Méthode proposée	2017	FS-PeSOA-SVM	99.95%
			FS-PeSOA-KNN	98.25%
			FS-PeSOA-NB	98.35%
Diabet	Choubey, D.K et al	2015	GA-J48graft DT	74.78%.
	Méthode proposée	2017	FS-PeSOA-SVM	77.86%
			FS-PeSOA-KNN	83.68%
			FS-PeSOA-NB	78.38%
Mammog	Kumar R.G.et al	2014	GA-SVM	79.00%
	Approches proposées	2017	FS-PeSOA-SVM	82.51%
			FS-PeSOA-KNN	82.47%
			FS-PeSOA-NB	81.44%
Derm	Harb et al	2014	PSO-Bay	99.45%
			PSO-NBay	99.18%
	Approches proposées	2017	FS-PeSOA-SVM	99.49%
			FS-PeSOA-KNN	98.68%
FS-PeSOA-NB			98.78%	
Colon Tumor	Sahu et al	2012	PSO-KNN	99.44%
	Approches proposées	2017	FS-PeSOA-SVM	100.00%
			FS-PeSOA-KNN	99.65%
			FS-PeSOA-NB	71.43%
Prostate Cancer	Chen et al	2014	PSO-DT	94.10%
	Approches proposées	2017	FS-PeSOA-SVM	100.00%
			FS-PeSOA-KNN	99.23%
			FS-PeSOA-NB	98.88%

Tableau 4.6 : Une comparaison des résultats obtenus par notre méthode FS-PeSOA et d'autres méthodes de la littérature dans différents datasets

4.4.1.4 Discussion

D'après le [Tableau 4.4](#) nous remarquons que les résultats de la méthode de sélection d'attribut de type wrapper basée sur l'algorithme FS-PeSOA combinée avec les différents classificateurs SVM , KNN et Naive Bayes respectivement sont meilleurs que les résultats obtenus avec la méthode de sélection d'attribut basée sur l'algorithme génétique combinée avec les différents classificateurs SVM , KNN et Naive Bayes respectivement pour six ensembles de données (Wisconsin Breast Cancer, Diabète Pima, Masse mammographique, Dermatologie, Colon Tumor et Prostate Cancer).

Ces différents résultats obtenus pour ces deux méthodes méta-heuristiques cités aux paragraphes précédents sont meilleurs que les résultats obtenus avec les différents classificateurs (SVM , KNN et Naive Bayes) sans sélection d'attribut.

Les résultats atteignent une précision de classification de 100% pour les ensembles de données sur les tumeurs du côlon et le cancer de la prostate avec un classificateur FS-PeSOA-SVM.

Pour le [Tableau 4.5](#), presque les mêmes remarques pour le [Tableau 4.4](#) où nous utilisons l'Accuracy comme critère d'évaluation mais avec des résultats moins bons pour les différents datasets et les différents classificateurs . Les résultats atteignent une précision de classification de 99.90% pour les ensembles de données sur les tumeurs du côlon avec un classificateur FS-PeSOA-SVM.

D'après la comparaison des résultats obtenue par notre méthode FS-PeSOA et d'autres méthodes de la littérature dans les différents ensembles de données qui se présente dans le [Tableau 4.6](#), nous remarquons que les résultats de notre méthode sont meilleurs par rapport aux autres méthodes. Les résultats obtenus garantissent l'efficacité des paramètres extraits.

4.4.2 L'approche FS-SLOA

4.4.2.1 Paramètres de FS-SLOA

Dans cette partie, nous définissons les valeurs des paramètres utilisés pour notre algorithme FS-SLOA. le [Tableau 4.7](#) présente les différents paramètres avec leur valeurs.

Paramètre	Valeur
Nb_Coc	50
Nb_patch	10
Max_it	100
r_1 et r_2	Un nombre aléatoire entre [0,1]
C, w	1, 1
ϵ_1 et ϵ_2	Petits nombres aléatoires

Tableau 4.7 : Les paramètres de FS-SLOA

4.4.2.2 Paramètres de l'algorithme génétique pour la sélection d'attributs

L'algorithme génétique a les paramètres de réglage présentés dans le [Tableau 4.8](#).

Paramètre	Valeur
Nombre d'attributs	50% du nombre d'attributs maximale
Taille de la population	5
Maximum nombre de générations	100
Sélection Tournoi (Taille du tournoi)	0.25
Taux de mutation	0.3
Taux de croisement	0.7

Tableau 4.8 : Les paramètres de l'algorithme génétique

4.4.2.3 Résultats des expérimentations et discussion

Les différentes expérimentations sont portées sur les quatre datasets (Wisconsin Breast cancer, Pima Diabetes, Mammographic Mass, Dermatology).

Pour évaluer l'efficacité de l'algorithme Fs-SLOA sur quatre jeux de données de référence (Wisconsin Breast cancer, Pima Diabetes, Masse mammographique et Dermatologie), nous avons mené trois expériences. Dans ces expériences, l'ensemble de données est divisé en deux sous-ensembles disjoints, à savoir un ensemble d'apprentissage et un ensemble de tests en utilisant le schéma de validation croisée 10 folds. Tout d'abord, les trois classificateurs de base, KNN, Naive Bayes et SVM, sont utilisés dans tout le jeu de données sans sélection d'attribut.

Dans la deuxième expérience, l'algorithme génétique est utilisé comme une méthode de sélection d'attribut combinée avec ces trois classificateurs de base.

Troisièmement, le FS-SLOA proposé est appliqué, le [Tableau 4.9](#) et le [Tableau 4.10](#) résumement les résultats obtenus. A titre de comparaison, les précisions de classification obtenues avec la méthode proposée et d'autres approches existantes de la littérature sont présentées dans le [Tableau 4.11](#).

	SVM	SVM	SVM	KNN	KNN	KNN	NB	NB	NB
		Genetic	SLOA		Genetic	SLOA		Genetic	SLOA
W B C	96.01%	97.00%	99.95%	92.91%	97.92%	98.25%	95.78%	96.22%	98.35%
Diabet	69.53%	77.03%	77.86%	07.05%	73.68%	83.68%	77.69%	86.47%	78.38%
Mammog	80.00%	78.00%	82.51%	79.71%	82.37%	82.47%	78.35%	80.15%	81.44%
Derm	96.00%	96.70%	99.49%	92.98%	94.88%	98.68%	95.41%	98.33%	98.78%

Tableau 4.9 : Les accuracies obtenues pour différents datasets

	SVM	SVM	SVM	KNN	KNN	KNN	NB	NB	NB
		Genetic	SLOA		Genetic	SLOA		Genetic	SLOA
WDBC	95.47%	96.12%	98.98%	91.87%	96.88%	97.15%	94.77%	95.28%	96.44%
P Diabt	68.56%	76.83%	76.98%	76.75%	72.88%	75.90%	75.89%	75.32%	75.31%
Mammog	79.52%	77.43%	81.48%	78.65%	81.45%	80.97%	77.45%	79.10%	80.00%
Derm	94.04%	96.01%	98.96%	92.98%	93.88%	97.96%			

Tableau 4.10 : Les f-measures obtenues pour différents datasets

Dataset	Auteur	Année	Méthode	Accuracy
WDBC	Sweilam N.H. et al	2010	SVM-PSO	93.06%
	Prasad Y. et al	2010	GA-SVM	97.19%
	Afzan A. et al	2012	GA-BPNN	83.36%
	Kaur S. et al	2016	SVM-RFE	97%
	Méthode proposée	2017	FS-SLOA-SVM	99.95%
			FS-SLOA-KNN	98.25%
FS-SLOA-NB			98.35%	
Pima Diabetes	Aslam et al	2010	GP-KNN	78.5%
	Karegowda et al	2011	GA-BPN	77.70%
	Méthode proposée	2017	FS-SLOA-SVM	77.86%
			FS-SLOA-KNN	75.92%
			FS-SLOA-NB	78.38%
Mammogram	Kumar G. et al	2014	GA-SVM	79.00%
	Méthode proposée	2017	FS-SLOA-SVM	82.51%
			FS-SLOA-KNN	82.47%
			FS-SLOA-NB	81.44%
Dermatology	Hany M. Harb et al	2014	PSO-DT	97.27%
			PSO-RBF	89.62%
	Méthode proposée	2017	FS-SLOA-SVM	99.49%
			FS-SLOA-KNN	98.68%
			FS-SLOA-NB	98.78%

Tableau 4.11 : Une comparaison des résultats obtenus par notre méthode FS-SLOA et d'autres méthodes de la littérature dans différents datasets

4.4.2.4 Discussion

Nous pouvons observer d'après le [Tableau 4.9](#) que la méthode proposée de sélection d'attribut basée sur l'algorithme FS-SIOA combinée avec les différents classificateurs SVM, KNN et Naive Bayes respectivement, a donné de meilleurs résultats que ceux obtenus avec que les différents classificateurs sans sélection d'attribut et ceux issus de l'algorithme génétique pour quatre ensembles de données (Wisconsin Breast Cancer, Diabète Pima, Masse mammographique, Dermatologie).

Les résultats atteignent une précision de classification de 99.95% pour les ensembles de données sur le dataset WDBC avec un classificateur FS-PeSOA-SVM.

Dans le [Tableau 4.10](#), nous remarquons que les résultats de l'algorithme FS-PeSOA combinée avec les différents classificateurs sont meilleurs que les résultats obtenus avec la méthode de sélection d'attribut basée sur l'algorithme génétique combinée avec les différents classificateurs pour les quatre ensembles de données. Les différents résultats obtenus sont meilleurs que les résultats obtenus avec les différents classificateurs (SVM, KNN et Naive Bayes) sans sélection d'attribut.

Les résultats atteignent un taux de classification de 98.98% pour le dataset WDBC avec un classificateur FS-PeSOA-SVM. Les résultats obtenus en utilisant l'Accuracy comme critère d'évaluation sont meilleurs que celles obtenus avec F-measure.

Le [Tableau 4.11](#) représente une comparaison de nos approches (FS-SLOA-SVM, FS-SLOA-KNN et FS-SLOA-NB) avec quelque méthode de l'état de l'art pour la classification de quatre ensembles de données.

Pour l'ensemble WDBC, nos trois approches ont donné les trois meilleurs taux de précision (99.95%,98.25%,99.35%) suivies par l'approche de Prasad (97.30%,97.19% et 95.38%).

Pour l'ensemble Diabet, nos approches obtenus le premier et le troisième taux de préci-

sion (77.86% et 78.38%) tandis que l'approche de Aslam a obtenu un taux de (78.50%). Pour l'ensemble Dermatologie l'approche FS-SLOA-SVM a obtenu un taux de (99.49%), l'approche FS-SLOA-NB avec le quatrième taux de (98.78%) devancées par les approches basé sur PSO de Harb. nous pouvons dire que nos approches ont obtenus des résultats encourageant et meilleurs que la majorité des approches de l'état étudié.

4.4.3 Les approches hybrides

Pour ces approches, nous avons choisi d'évaluer les performances en terme de précision du classificateur (accuracy) pour les quatre base de données (Wisconsin Breast Cancer, Pima Diabetes, Mammographic Mass, Dermatology).

On a utilise trois stratégies de validation la validation croisée (5 – *foldcross* validation), la validation croisée (10 – *fold* cross validation) et simple split data (90% pour apprentissage et 10% pour le test) pour chaque dataset, l'évaluation des performances en classification a été réalisée en utilisant les classificateurs (Bagging, AdaBoost et Multilayer_perceptron) sans sélection d'attributs, en deuxième lieu, avec sélection d'attributs par les méthodes filtre, meta-heuristique, hybride(filter suivi de meta-heuristique et meta-heuristique suivi de filter).

4.4.3.1 Résultats et Discussion

Le [Tableau 4.12](#) présente les résultats d'un simple classificateur Multilayer-perceptron et de deux classificateurs ensemblistes Bagging (DecisionTree) et AdaBoost en termes de précision de classification et nombre d'attributs sélectionné pour Dermatology dataset.

Le [Tableau 4.13](#) présente les résultats pour Mammographic Le [Tableau 4.14](#) présente les résultats pour WDBC. Le [Tableau 4.15](#) présente les résultats pour Pima Diabetes.

Le CFS obtient des meilleures résultats que celles obtenues avec les autres filtres pour les trois stratégies pour (avec Multilayer-perceptron, Dermatology et Mammographic

avec AdaBoost) et des résultats moins bonne mais avec un nombre d'attributs moins pour Mammographic avec Bagging.

Pour Dermatology, les performances obtenues avec le CFS sont meilleures que celles obtenues par les autres filtres pour les stratégies 5-fold ,10-fold et pour la troisième stratégie les performances sont identiques pour les trois filtres avec Bagging ; et ils sont meilleures que celles obtenues avec les autres filtres pour la troisième stratégie et ils sont identiques que celles obtenues avec les deux autres stratégies pour Multilayer-perceptron.

Pour Mammographic, Pour Multilayer-perceptron, les performances de CFS sont meilleures avec un nombre d'attributs moins que celles obtenues avec les autres filtres pour 10-fold et ils sont moins que celles obtenues avec les autres filtres pour 5-fold et la troisième stratégie. Pour WDBC, le filtre CFS obtient des meilleures résultats avec un nombre d'attributs moins que celles obtenues avec les autres filtres pour 10-fold et 5-fold stratégie mais moins bonnes pour la troisième stratégie avec le Bagging. Pour AdaBoost, en utilisant la troisième stratégie les performances sont meilleurs à celles obtenues avec les autres filtres et avec un nombre d'attributs moins. Mais moins bonne pour 5-fold et 10-fold stratégies.

Pour le Pima Diabetes, les performances obtenues avec le CFS sont meilleures avec un nombre d'attributs moins que celles obtenues avec les autres filtres par 10-fold pour Bagging ; pour AdaBoost, les performances sont moins bonne et avec un nombre d'attributs moins que celles obtenues avec les autres pour 10-fold et ils sont identiques avec 5-fold et la troisième stratégie pour ces deux classificateurs. Pour Multilayer-perceptron, les performances obtenues sont moins que celles obtenues avec les autres filtres pour la troisième stratégie. Mais pour 5-fold et 10-fold stratégie les performances sont meilleures que celles obtenues avec les autres filtres.

Pour les différents dataset, les performances qui sont obtenues en utilisant les méthodes

méta-heuristiques (FS-PeSOA, GA et FS-SLOA) sont meilleures que celles obtenues avec les filtres pour les trois stratégies. Pour la méthode hybride (filtre-méta (CFS)), ces résultats sont meilleurs que celles obtenues avec les méthodes méta-heuristiques cependant pour les deux autres méthodes hybrides (filtre-méta (CHI et PCA)) les résultats sont identiques avec les méthodes méta-heuristiques. Pour la méthode hybride (Méta-filtre) ces résultats sont meilleures que celles obtenues avec les méthodes Filtre-méta avec un nombre d'attributs moins.

Dans ces expérimentations, nous avons comparé l'utilisation de validations croisées 5-fold, 10-fold et de la division aléatoire (90% pour apprentissage et 10% pour teste). D'après les résultats obtenus on remarque que la division aléatoire donne des résultats mieux que les résultats de 10-fold et la stratégie de 10-fold donne des résultats mieux que 5-fold pour dermatologie dataset avec tous les classificateurs, pour Mammographic Mass avec Bagging et Multilayer-perceptron, pour Wisconsin Breast Cancer avec Bagging, AdaBoost et pour Pima Diabetes Multilayer-perceptron.

On peut remarquer depuis les quatre tableaux que les meilleurs taux classification avec un nombre d'attributs moins sont obtenus par l'hybridation de méta-heuristique suivi de filtre (FS-SLOA_PCA) avec (81.19% et 3 attributs) pour Mammographic et avec (97.99% et 14 attributs) pour Dermatology et par (FS-SLOA_CFS) avec (98.92%, 8 attributs) pour WDCB et avec (84.15% et 5 attributs) pour Pima Diabetes.

Tableau 4.12 : L'accuracy et le nombre d'attributs obtenus pour dermatologie dataset

	Bagging(DecisionTree)			Ada Boost			Multilayer-perceptron						
	Acc(5-f)	Acc(10-f)	Attr	Acc(5-f)	Acc(10-f)	Attr	Acc(5-f)	Acc(10-f)	Attr				
File	CFS	92.60%	93.40%	93.60%	19	95.40%	96.20%	97.60%	19	96.20%	96.40%	97.40%	19
	Chisquare	92.30%	93.10%	93.60%	34	94.00%	96.40%	97.40%	34	96.20%	96.40%	97.30%	34
	PCA	92.30%	93.10%	93.60%	34	94.00%	96.40%	97.40%	34	96.20%	96.40%	97.30%	34
Meta_H	FS-PeSOA	97.10%	97.39%	97.59%	17	97.42%	97.48%	97.91%	17	97.00%	97.60%	97.95%	17
	Genetic	93.10%	94.10%	94.12%	17	93.10%	93.10%	94.48%	17	93.01%	93.45%	97.48%	17
	FS-SLOA	97.30%	97.44%	97.45%	17	97.55%	97.81%	97.83%	17	97.41%	97.68%	97.83%	17
F_M(CFS)	CFS_FS-PeSOA	97.38%	97.46%	97.61%	15	97.28%	97.88%	97.92%	15	97.45%	97.68%	97.86%	15
	CFS_Genetic	95.13%	95.10%	95.23%	15	93.04%	93.53%	94.49%	15	94.00%	94.02%	95.11%	15
	CFS_FS-SLOA	97.51%	97.63%	97.33%	15	97.46%	97.58%	97.87%	15	97.34%	97.87%	97.95%	15
M_F(Cfs)	FS-PeSOA_CFS	97.23%	97.38%	97.47%	13	97.11%	97.81%	97.90%	14	97.60%	97.89%	97.98%	13
	Genetic_CFS	95.00%	95.01%	95.21%	13	94.01%	94.02%	94.31%	14	94.45%	94.56%	95.04%	13
	FS-SLOA_CFS	96.48%	97.49%	97.79%	13	97.49%	97.56%	97.79%	14	97.02%	97.49%	97.97%	13
F_M(Chi)	Chi_FS-PeSOA	97.10%	97.39%	97.59%	17	97.42%	97.48%	97.91%	17	97.00%	97.60%	97.95%	17
	Chi_Genetic	93.10%	94.10%	94.12%	17	93.10%	93.10%	94.48%	17	93.01%	93.45%	97.48%	17
	Chi_FS-SLOA	97.30%	97.44%	97.45%	17	97.55%	97.81%	97.83%	17	97.41%	97.68%	97.83%	17
M_F(Chi)	FS-PeSOA_Chi	94.12%	96.66%	96.56%	14	97.32%	97.55%	97.81%	14	97.72%	97.58%	97.95%	14
	Genetic_Chi	94.47%	94.55%	94.77%	17	93.24%	93.83%	93.48%	14	94.81%	97.85%	97.87%	14
	FS-SLOA_Chi	97.12%	97.23%	97.40%	14	97.04%	97.18%	97.28%	14	97.24%	97.58%	97.87%	14
F_M(PCA)	PCA_FS-PeSOA	97.10%	97.39%	97.59%	17	97.42%	97.48%	97.91%	17	97.00%	97.60%	97.95%	17
	PCA_Genetic	93.10%	94.10%	94.12%	17	93.10%	93.10%	94.48%	17	93.01%	93.45%	97.48%	17
	PCA_FS-SLOA	97.30%	97.44%	97.45%	17	97.55%	97.81%	97.83%	17	97.41%	97.68%	97.83%	17
M_F(PCA)	FS-PeSOA_PCA	96.59%	96.69%	96.77%	14	97.35%	97.78%	97.92%	14	97.10%	97.75%	97.91%	14
	Genetic_PCA	94.17%	94.74%	94.73%	14	93.14%	93.77%	94.37%	14	93.92%	93.75%	94.80%	14
	FS-SLOA_PCA	96.25%	96.59%	96.70%	14	97.45%	97.58%	97.69%	14	97.19%	97.79%	97.99%	14

Tableau 4.13 : L'accuracy et le nombre d'attributs obtenus pour mammographic dataset

	Bagging(Decision Tree)			Ada Boost			Multilayer-perceptron						
	Acc(5-f)	Acc(10-f)	Acc(split)	attr	Acc(5-f)	Acc(10-f)	Acc(split)	attr	Acc(5-f)	Acc(10-f)	Acc(split)	attr	
Fibre	CFS	82.40%	82.60%	81.50%	3	82.90%	82.90%	81.30%	3	81.10%	81.60%	82.20%	3
	Chisquare	82.70%	83.40%	82.20%	5	82.70%	75.10%	79.20%	5	81.70%	81.00%	82.40%	5
	PCA	82.70%	83.40%	82.20%	5	82.70%	75.10%	79.20%	5	81.70%	81.00%	82.40%	5
Meta_H	FS-PeSOA	83.10%	83.04%	83.12%	3	81.25%	81.34%	81.44%	3	80.00%	80.45%	80.97%	3
	Genetic	79.00%	79.05%	79.80%	3	77.28%	77.37%	77.39%	3	75.07%	75.34%	76.45%	3
	FS-SLOA	83.11%	83.51%	83.45%	3	83.01%	83.02%	83.07%	3	81.04%	80.68%	80.90%	3
F_M(CFS)	CFS_FS-PeSOA	81.44%	81.85%	81.98%	2	81.25%	81.44%	81.56%	2	79.51%	79.45%	79.97%	3
	CFS_Genetic	75.63%	75.84%	75.90%	2	79.52%	77.61%	78.39%	2	75.94%	75.92%	76.25%	2
	CFS_FS-SLOA	81.95%	81.69%	81.89%	2	82.01%	82.09%	82.18%	2	81.26%	81.58%	81.79%	3
M_F(Cfs)	FS-PeSOA_CFS	83.91%	84.00%	84.07%	3	83.13%	83.34%	83.45%	3	97.31%	79.25%	79.85%	3
	Genetic_CFS	77.01%	77.12%	77.37%	3	76.12%	76.24%	76.36%	2	75.37%	75.34%	76.45%	2
	FS-SLOA_CFS	83.16%	83.38%	83.45%	3	84.03%	84.44%	84.51%	3	81.27%	81.43%	81.88%	3
F_M(Chi)	Chi_FS-PeSOA	83.10%	83.04%	83.12%	3	81.25%	81.34%	81.44%	3	80.00%	80.45%	80.97%	3
	Chi_Genetic	79.00%	79.05%	79.80%	3	77.28%	77.37%	77.39%	3	75.07%	75.34%	76.45%	3
	Chi_FS-SLOA	83.11%	83.51%	83.45%	3	83.01%	83.02%	83.07%	3	81.04%	80.68%	80.90%	3
M_F(Chi)	FS-PeSOA_Chi	83.15%	83.44%	83.52%	3	83.01%	83.33%	83.41%	3	79.98%	80.97%	81.07%	3
	Genetic_Chi	78.10%	78.13%	78.20%	3	75.32%	76.81%	77.03%	3	74.67%	74.36%	75.85%	3
	FS-SLOA_Chi	83.02%	83.32%	83.33%	3	83.02%	83.44%	83.56%	3	80.95%	80.91%	82.07%	3
F_M(PCA)	PCA_FS-PeSOA	83.10%	83.04%	83.12%	3	81.25%	81.34%	81.44%	3	80.00%	80.45%	80.97%	3
	PCA_Genetic	79.00%	79.05%	79.80%	3	77.28%	77.37%	77.39%	3	75.07%	75.34%	76.45%	3
	PCA_FS-SLOA	83.11%	83.51%	83.45%	3	83.01%	83.02%	83.07%	3	81.04%	80.68%	80.90%	3
M_F(PCA)	FS-PeSOA_PCA	83.14%	83.35%	83.54%	3	83.01%	83.22%	83.31%	3	80.98%	02.98%	81.97%	3
	Genetic_PCA	78.02%	78.24%	78.22%	3	75.82%	76.07%	76.79%	3	75.54%	75.34%	76.25%	3
	FS-SLOA_PCA	83.14%	83.43%	83.54%	3	83.23%	83.23%	83.23%	3	80.56%	82.49%	82.61%	3

Tableau 4.14 : L'accuracy et le nombre d'attributs obtenus pour WDBC dataset

	Bagging(DecisionTree)			Ada Boost			Multilayer-perceptron					
	Acc(5-f)	Acc(10-f)	Acc(split)	attr	Acc(5-f)	Acc(10-f)	Acc(split)	attr	Acc(5-f)	Acc(10-f)	Acc(split)	attr
Fitre	CFS	95.40%	95.40%	16	94.00%	94.20%	98.40%	16	97.20%	97.00%	98.40%	16
	Chisquare	94.30%	94.90%	32	95.20%	95.80%	98.30%	32	95.90%	95.90%	96.80%	32
	PCA	94.30%	94.90%	32	95.20%	95.80%	98.30%	32	95.90%	95.90%	96.80%	32
Meta_H	FS-PeSOA	95.84%	96.44%	16	96.01%	96.02%	97.11%	16	96.00%	96.60%	96.95%	16
	Genetic	90.01%	90.23%	16	91.95%	92.01%	92.12%	16	90.70%	91.33%	91.45%	16
	FS-SLOA	97.05%	97.75%	16	97.66%	97.95%	98.48%	16	96.14%	96.68%	97.81%	16
F_M(CFS)	CFS_FS-PeSOA	94.94%	94.854%	10	93.99%	94.03%	95.32%	10	96.54%	96.68%	97.86%	10
	CFS_Genetic	90.19%	90.12%	10	87.59%	87.99%	88.02%	10	93.00%	93.43%	93.95%	10
	CFS_FS-SLOA	96.07%	96.19%	10	95.51%	95.65%	96.95%	10	96.34%	96.78%	97.98%	10
M_F(Cfs)	FS-PeSOA_CFS	94.20%	94.24%	8	94.45%	95.05%	95.43%	8	96.60%	96.89%	97.98%	8
	Genetic_CFS	89.01%	89.12%	8	88.01%	89.11%	89.22%	8	94.45%	94.56%	94.97%	8
	FS-SLOA_CFS	95.75%	95.75%	8	94.44%	94.59%	94.78%	8	96.22%	96.49%	98.92%	8
F_M(Chi)	Chi_FS-PeSOA	95.84%	96.44%	16	96.01%	96.02%	97.11%	16	96.00%	96.60%	96.95%	16
	Chi_Genetic	90.01%	90.23%	16	91.95%	92.01%	92.12%	16	90.70%	91.33%	91.45%	16
	Chi_FS-SLOA	97.05%	97.75%	16	97.66%	97.95%	98.48%	16	96.14%	96.68%	97.81%	16
M_F(Chi)	FS-PeSOA_Chi	95.40%	95.41%	11	96.07%	96.40%	97.32%	11	96.27%	96.58%	97.95%	11
	Genetic_Chi	92.01%	92.23%	11	92.95%	93.01%	93.12%	11	92.70%	93.33%	93.45%	11
	FS-SLOA_Chi	97.12%	97.82%	11	97.56%	98.70%	98.78%	11	96.24%	96.58%	97.97%	11
F_M(PCA)	PCA_FS-PeSOA	95.84%	96.44%	16	96.01%	96.02%	97.11%	16	96.00%	96.60%	96.95%	16
	PCA_Genetic	90.01%	90.23%	16	91.95%	92.01%	92.12%	16	90.70%	91.33%	91.45%	16
	PCA_FS-SLOA	97.05%	97.75%	16	97.66%	97.95%	98.48%	16	96.14%	96.68%	97.81%	16
M_F(PCA)	FS-PeSOA_PCA	94.27%	94.42%	11	95.26%	96.02%	97.13%	11	96.10%	96.75%	97.91%	11
	Genetic_PCA	92.00%	92.05%	11	92.89%	92.96%	92.91%	11	92.67%	92.99%	93.44%	11
	FS-SLOA_PCA	96.03%	96.66%	11	96.47%	96.75%	98.05%	11	96.19%	96.89%	97.99%	11

Tableau 4.15 : L'accuracy et le nombre d'attributs obtenus pour pima diabetes dataset

	Bagging(Decision Tree)			Ada Boost			Multilayer-perceptron						
	Acc(5-f)	Acc(10-f)	Acc(split)	attr	Acc(5-f)	Acc(10-f)	Acc(split)	attr	Acc(5-f)	Acc(10-f)	Acc(split)	att	
Fibre	CFS	72.60%	71.70%	77.20%	4	74.10%	73.80%	75.60%	8	75.40%	75.30%	76.00%	4
	Chisquare	72.60%	71.30%	77.20%	8	74.10%	73.80%	75.60%	8	75.00%	80.10%	80.70%	8
	PCA	72.60%	71.30%	77.20%	8	74.10%	73.80%	75.60%	8	75.00%	80.10%	80.70%	8
Meta_H	FS-PeSOA	72.87%	73.87%	77.72%	4	71.56%	71.67%	74.67%	4	74.89%	74.89%	79.55%	4
	Genetic	71.21%	72.23%	73.32%	4	69.66%	69.68%	70.69%	4	70.32%	70.74%	72.74%	4
F_M(CFS)	FS-SLOA	72.68%	73.96%	77.66%	4	70.52%	70.68%	75.40%	4	74.56%	75.78%	80.72%	4
	CFS_FS-PeSOA	71.57%	71.56%	77.36%	6	72.92%	72.12%	74.32%	3	70.94%	70.60%	77.34%	3
M_F(Cfs)	CFS_Genetic	69.56%	69.14%	70.05%	3	66.48%	66.08%	70.15%	3	70.00%	70.02%	70.25%	3
	CFS_FS-SLOA	72.99%	72.99%	77.99%	3	72.46%	72.47%	74.52%	3	70.69%	70.68%	77.45%	3
	FS-PeSOA_CFS	71.45%	71.73%	77.37%	6	71.23%	71.32%	74.32%	6	71.00%	73.32%	79.20%	6
F_M(Chi)	Genetic_CFS	67.78%	76.61%	70.11%	6	66.08%	66.08%	71.08%	6	70.02%	70.92%	72.03%	6
	FS-SLOA_CFS	72.92%	72.80%	77.75%	6	70.04%	70.04%	74.04%	6	72.42%	73.98%	79.48%	6
	Chi_FS-PeSOA	72.87%	73.87%	77.72%	4	71.56%	71.67%	74.67%	4	74.89%	74.89%	79.55%	4
M_F(Chi)	Chi_Genetic	71.21%	72.23%	73.32%	4	69.66%	69.68%	70.69%	4	70.32%	70.74%	72.74%	4
	Chi_FS-SLOA	72.68%	73.96%	77.66%	4	70.52%	70.68%	75.40%	4	74.56%	75.78%	80.72%	4
	FS-PeSOA_Chi	71.56%	71.60%	77.25%	3	72.68%	72.67%	75.76%	3	75.21%	75.32%	80.52%	3
M_F(Chi)	Genetic_Chi	69.10%	69.03%	70.62%	3	69.69%	69.68%	70.92%	3	70.02%	70.06%	71.92%	3
	FS-SLOA_Chi	73.72%	73.66%	77.52%	3	73.68%	73.67%	75.86%	3	75.49%	75.94%	80.99%	3
	PCA_FS-PeSOA	72.87%	73.87%	77.72%	4	71.56%	71.67%	74.67%	4	74.89%	74.89%	79.55%	4
F_M(PCA)	PCA_Genetic	71.21%	72.23%	73.32%	4	69.66%	69.68%	70.69%	4	70.32%	70.74%	72.74%	4
	PCA_FS-SLOA	72.68%	73.96%	77.66%	4	70.52%	70.68%	75.40%	4	74.56%	75.78%	80.72%	4
	FS-PeSOA_PCA	72.61%	72.52%	77.40%	3	73.60%	73.66%	75.66%	3	74.22%	74.21%	80.15%	3
M_F(PCA)	Genetic_PCA	68.62%	68.51%	96.11%	3	68.76%	68.68%	71.68%	3	70.72%	70.72%	70.11%	3
	FS-SLOA_PCA	72.87%	72.86%	77.68%	3	73.93%	73.92%	75.22%	3	75.89%	75.88%	81.19%	3

4.5 L'approches proposée pour la classification

4.5.1 Résultats et discussion

Afin d'ajuster les paramètres de l'algorithme SLOA pour la classification ; plusieurs expériences ont été réalisées de manière empirique. Les meilleures valeurs de paramètres sont : 50 coccinelles à sept , 5 patchs, 30 itérations, nombres aléatoires entre $[0, 1]$, $W = 1$, $c = 2$.

Pour évaluer l'efficacité de notre algorithme, nous divisons l'ensemble de données en deux sous-ensembles disjoints, à savoir un ensemble d'apprentissage et un ensemble de test utilisant le schéma de validation croisée 10 folds. Le [Tableau 4.16](#) présente l'évaluation des trois classificateurs de base, KNN, Naive Bayes et SVM et les résultats de l'algorithme proposé. A titre de comparaison, les précisions de classification obtenues avec la méthode proposée et d'autres approches existantes de la littérature sont présentées dans le [Tableau 4.17](#).

	SVM	KNN	NB	Méthode SLOA
W D B C	96.01%	92.91%	95.87%	98.05%
Diabet	69.53%	77.05%	77.69%	77.73%
Mammog	80.00%	79.71%	78.35%	79.42%

Tableau 4.16 : Les Accuracies obtenues pour différents datasets

Dataset	Auteur	Année	Méthode	Accuracy
WDBC	Z. Wang et al	2007	PSO	97.95%
			Ant_Miner	96.04%
	Sousa T et al	2004	GPSO	93.4%
	Méthode proposée	2018	SLOA	98.05%
Pima Diabetes	Cervantes et al	2005	PSO	74.78%.
	E.Sreedevi	2017	HGA	72.214%
	Zulfiqar et al	2017	Ant_Miner	69.15%
			cAnt_Miner	68.61%
			cAnt_Miner2	68.62%
			cAnt_MinerPB	73.30%
Méthode proposée	2018	SLOA	77.33%	
Mammogram	Méthode proposée	2018	SLOA	79.421%
Mass				

Tableau 4.17 : Une comparaison des résultats obtenus par notre méthode SLOA pour la classification et d'autres méthodes de la littérature dans différents datasets

L'approche proposée est plus performante que les autres méthodes d'apprentissage machine telles que les classificateurs KNN, Naive Bayes et SVM pour différents jeux de données. Les résultats atteignent une précision de classification de 98,05% pour le cancer du sein du Wisconsin, sans avoir recours à la sélection des attributs.

Pour le [Tableau 4.17](#) nous remarquons que le résultat pour le WDCB a obtenu une précision de 98.05% avec notre approche et de 97% pour l'approche de Wang. La classification du Pima diabetes dataset, a donné une précision de 77.73% tandis que l'approche proposée par Cervantes a donné une précision de 74.78% avec le PSO. Pour le Mammographic mass dataset ; nous n'avons pas trouvé une approche dans la littérature pour la comparer à la notre.

Conclusion

Ce chapitre a pour objectif d'exposer des approches que nous proposons d'appliquer pour la sélection d'attribut et la classification. Pour la sélection d'attribut, nous avons proposé l'approche FS-PeSOA, l'approche FS-SLOA et des approches hybrides. Pour la classification, nous avons proposé la méta-heuristique SLOA comme méthode de classification. Enfin nous terminons par une conclusion.

Conclusion générale

L'extraction des connaissances à partir de données est un axe de recherche très actif notamment en raison de leurs applications dans des domaines où plusieurs problèmes peuvent être exprimés sous la forme de problèmes de classification. Dans ce processus, la classification est la plus importante étape.

Dans notre thèse, nous avons proposé différentes approches basées sur les méta-heuristiques afin d'améliorer des tâches de la fouille de données, en particulier la sélection d'attributs et la classification supervisée.

Dans la première partie, on s'intéresse à la sélection de sous-ensembles d'attributs. En premier lieu, nous proposons deux méthodes méta-heuristiques pour la sélection de sous-ensembles d'attributs FS-PeSOA et FS-SLOA et en deuxième lieu nous détaillons deux autres approches hybrides (filtre/méta-heuristique) et (méta-heuristique/filtre) où on compare l'hybridation de plusieurs méthodes de filtrage et de méta-heuristique, y compris la sélection d'attribut basée sur les corrélations (CFS), l'analyse en composantes principales (PCA), les statistiques X2 (CHI) et l'algorithme FS-SLOA, FS-PeSOA dans le cadre des méthodes méta-heuristiques.

L'aspect hybride réside dans la séquence d'une méthode de filtrage et d'une méthode méta-heuristique ou la séquence d'une méthode méta-heuristique et d'une méthode de filtrage. La comparaison a été réalisée par l'utilisation de la validation croisée pour la classification avec différentes techniques d'apprentissage.

Dans la deuxième partie nous nous intéressons à la classification par une approche méta-heuristique SLOA. Les résultats expérimentaux montrent que cette approche est compétitive mais il n'existe pas une méthode universelle pour toutes les datasets et pour tous les classificateurs.

En perspective, plusieurs voies de recherches prometteuses peuvent être suivies pour la continuité de ce travail nous pouvons utiliser d'autres paramètres et valeurs heuristiques qui peuvent améliorer les résultats. Une autre alternative est de considérer d'autres algorithmes de classification ou d'autres méta-heuristiques,...

Bibliographie

- [1] Fayyad U. M., Piatetsky-Shapiro G. et Smyth P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11), pp. 27-34.
- [2] Bachtobji M. A. (2005). *Informatique Appliquée à la Gestion, Mémoire de Mastère Université de Tunis, Institut Supérieur de Gestion.*
- [3] Cabena P., Hadjinian P. , Stadler R., Verhees J. et Zanasi A.(1998). *Discovering Data Mining : From Concept to Implementation*, Prentice Hall, Upper Saddle River, NJ.
- [4] Piatsky-shapiro G. (2005). Data mining and knowledge discovery 1996 to 2005 : overcoming the hype and moving from «university» to «business» and «analytics», *Data mining and Knowledge Discovery*, 15(1), 99-105.
- [5] Frawley W., Piatetsky-Shapiro G, Gregory et Christopher M. (1992). *Knowledge Discovery in Databases : An Overview.*
- [6] Michael J.A. et Linoff G. S. (1997). *Data Mining :Techniques appliquées au marketing, à la vente et aux services clients.*
- [7] Usama M., Fayyad U.M., Gregory P.S. et Padhraic S. (1996). From data mining to knowledge discovery : An overview. In *Advances in Knowledge Discovery and Data Mining*, pages 1–34.

-
- [8] Berry M.J. (1997). Immigration, acculturation and adaptation. *Applied Psychology : An International Review* 46, 5-34.
- [9] BERRY M.J. et LINOFF G.S. (2000). *Mastering Data Mining : The Art and Science of Customer Relationship Management*.
- [10] Larose D.T. (2005). *Discovering Knowledge In Data : An Introduction to DataMining*, Central Connecticut State University, Published by John Wiley and Sons, Inc., Hoboken, New Jersey.
- [11] Mari J. et Napoli A. (1996). Aspects de la classification. Rapport technique 2909, INRIA.
- [12] Henriët L. (2000). *Système d'évaluation et de classification multicritères pour l'aide à la décision, construction de modèles et procédures d'affectation*. Thèse de doctorat en science. Université Paris Dauphine.
- [13] Bogner K. (2003). *Aspects théoriques de la classification à base de treillis*. Université Debrecen : Institut de mathématiques et informatique.
- [14] Preux p.(2001). *Fouille de données notes de cours, université de lille*.
- [15] Michie D., Spiegelhalter D. et C.C. (1994). *Machine learning, neural and statistical classification*. New York : Ellis Horwood.
- [16] Bayes T. (1773). *An Essay towards solving a Problem in the Doctrine of Chances*. *Philosophical Transactions of the Royal Society of London*.
- [17] Rosenblatt F. (1958). The Perceptron : a probabilistic model for information storage and organization in the brain. *Psychological Review* 65 : 386-408.
- [18] Vapnik W. (1995). *The Natural of statistical learning Theory*, Springer Wiley.
- [19] YANG Y.(1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*, vol. 1, n 1/2,p. 69–90, Kluwer Academic Publishers.

- [20] Quinlan J.R. (1986). Induction of decision trees. *Machine Learning*.
- [21] Quinlan J.R. (1993). *C4.5 : Programs for Machine Learning*. Morgan Kaufmann, San Diego.
- [22] Russell S.J. et Norvig P. (2003). *Artificial Intelligence : A Modern Approach*. Pearson Education.
- [23] Beiman L. (1996). Bagging predictors. *Machine Learning*, 24 :123–140.
- [24] Schapire R.E. (1990). The strength of weak learning. *Machine Learning*, 5(2) :197–227.
- [25] Kanal L. et Chandrasekaran B.(1972) On linguistic, statistical and mixed models for pattern recognition, *Frontiers of Pattern Recognition*, New York, Academic Press, pp. 163-192.
- [26] Jain A.K., Duin R.P.W. et Mao J.(2000). Statistical Pattern Recognition : A Review. *IEEE Trans. Pattern Analysis and Machine Learning*, vol. 22, no. 1.
- [27] Rakotomalala R.(1997). *Graphes d'induction*, Thèse de doctorat de l'Université Claude Bernard, Lyon, 1997.
- [28] Feraud, R.B.M., Clérot F. et Lemaire, V.(1998). The orange customer analysis platform. In *Proceedings of the 10th International Conference on Data Mining, Lecture Notes in computer Science*, pages 584-594. Springer.
- [29] Kohavi R. et John G.H.(1997). Wrappers for feature subset selection. *Artif. Intell.*, pages 273–324.
- [30] Almuallim H. et Dietterich T.G. (1991). Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 547–552.

- [31] Dash M. et Liu H. (1998). Hybrid search of feature subsets. In H.Y. Lee and H. Motoda, editors, Proceedings of the Fifth Pacific Rim International Conference on AI :PRICAI'98, pages 238–249, Singapore, 22–27.
- [32] Gennari J. H. et Fisher P. (1989). Models of incremental concept formation. *Artificial Intelligence*, p11-61.
- [33] Dash M. ET Liu H. (1997). Feature selection for classification. *Intelligent Data Analysis*, 1 :131–156.
- [34] Narendra P.M. et Fukunaga K. A. (1977). branch and bound algorithm for feature subset selection. *IEEE Trans. on Computer*, p 917–922.
- [35] Doak J. (1992). An evaluation of feature selection methods and their application to computer security. Technical report, Davis CA : University of California, Department of Computer Science.
- [36] Brassard G. et Bratley P. (1996). *Fundamentals of Algorithms*. Prentice Hall, New Jersey.
- [37] Diday E., Lemaire J., Pouget J. et Testu F. (1982). *Elements d'analyse de données*. Dunod.
- [38] Semani D. (2004). Une méthode supervisée de sélection et de discrimination avec rejet. Application au projet Aquathèque. PhD thesis, Université de La Rochelle.
- [39] Della Pietra S., Della Pietra V. et Lafferty J. (1997). Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4) :380–393.
- [40] Goswami S., and Chakrabarti A. (2014). Feature Selection : A Practitioner View. *I.J. Information Technology and Computer Science*, 11, 66-77.
- [41] Lal T. N., Chapelle O., Weston J. et Elisseeff A. (2004). *Embedded Methods*. Springer.

- [42] Das S.(2001). Filters, wrappers and a boosting-based hybrid for feature selection. In Proceedings of the Eighteenth International Conference on Machine Learning.
- [43] Guyon I., Weston J., Barnhill S. et Vapnik V.(2002). Gene selection for cancer classification using support vector machines,” MACHLEARN : Machine Learning,vol. 46. no. 1-3, pp. 389-422.
- [44] Palpant M. (2005) Recherche exacte et approchée en optimisation combinatoire : schémas d’intégration et applications. Thèse de Doctorat, Université d’Avignon.
- [45] Osman I.H. et Laporte G. (1996). Metaheuristics : a bibliography. Annals of Operations Research 63, 513-623.
- [46] Voß S., Martello S., Osman I.H. et Roucairol C. (1999). Meta-Heuristics-Advances and Trends in Local Search Paradigms for Optimization. Kluwer Academic Publishers, Dordrecht, The Netherlands.
- [47] Stützle T. (1999). Local Search Algorithms for Combinatorial Problems – Analysis, Algorithms and New Applications. DISKI – Dissertationen zur Künstlichen Intelligenz. Infix, Sankt Augustin, Germany.
- [48] Blum C. et Roli A. (2003). Metaheuristics in combinatorial optimization : Overview and conceptual comparison. ACM Computing Surveys, 35(3) :268–308.
- [49] Meignan D., Créput J.C. et Koukam A. (2008). A Coalition-based Metaheuristic for the Vehicle Routing Problem. In Proc. of IEEE World Congress on Evolutionary Computation.
- [50] Bachelet V. (1999). Métaheuristiques parallèles hybrides :Application au QAP. PhD thesis, USTL LIFL France.

- [51] Harish C. (1978). Admissible invariant distributions on reductive p-adic groups. In Proceedings of the 1977 Annual Seminar of the Canadian Mathematical Congress(W. Rossmann, Ed.), Queen's Papers in Pure and Applied Math., Vol. 48, pp. 281-347.
- [52] Kirkpatrick S., Gelatt C.D.Jr. et Vecchi M.P. (1983). Optimization by simulated annealing," Science, vol.220, no. 4598, pp. 671-680.
- [53] Glover F. (1998). A Template for Scatter Search and Path Relinking. In : Hao J.-K., Lutton E., Ronald E., Schoenauer M. et Snyers D. (eds.) AE 1997. LNCS, vol. 1363, p. 13. Springer, Heidelberg.
- [54] Feo T.A. et Resende M.G.C. (1995). Greedy randomized adaptive search procedures. J. of Global Optimization, 6 :109–133.
- [55] Gen M., et Cheng R. (1997). Genetic Algorithms & Engineering Design. John Wiley & Sons, Inc., Hoboken.
- [56] Dorigo M., Maniezzo V. et Colorni A. (1991). Positive feedback as a search strategy. Technical report 91-016, politecnico di Milano.
- [57] Storn R. et Price K. (1997). Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, p 341–359.
- [58] Kennedy J. et Eberhart R.(1995). Particle swarm optimization, proceedings of ieee international conference on neural networks. 1942–1948.
- [59] Yang J. et Honavar V. (1997). Feature subset selection using a genetic algorithm. IEEE Intelligent Systems, Vol. 13, No. 2, pp. 44-49.
- [60] Garcia F., Garcia M.T., Melian B.B., Jose A., Perez M. et Marcos J. (2004). Solving feature subset selection problem by a Parallel Scatter Search. European Journal of Operational Research 169.

- [61] Cantu-Paz E., Newsam S. et Kamath C. (2004). Feature Selection in Scientific Applications. Proceedings of the 10th ACM International conference on Knowledge Discovery and Data Mining (KDD), pp. 788–793.
- [62] Shahzad W. (2010). Classification and Associative Classification Rule Discovery Using Ant Colony Optimization. these, Pakistan.
- [63] Candelieri A. (2011). A hyper-solution framework for classification problems via metaheuristic approaches. 4OR-Q J Oper Res. Volume 9, Issue 4, pp425–428. Springer-Verlag.
- [64] Chen S., Lin S. et Chou S. (2011). Enhancing the Classification Accuracy by Scatter Search-Based ensemble Approach. Applied Soft Computing 11, 1021–1028.
- [65] Fan C., Chang P. et Hsieh J.L. (2011). A Hybrid Model Combining Case-based Reasoning and Fuzzy Decision Tree for Medical Data Classification. Applied Soft Computing 11, 632–644.
- [66] Luukka P. et Lampinen, J. (2011). Differential Evolutionary Classifier in Noisy Settings with Interactive Variables. Applied Soft Computing 1, 891–899.
- [67] Unler A. et Murat A. (2010). Discrete Optimization : A discrete particle swarm optimization method for feature selection in binary classification problems. European Journal of Operational Research 206(3), 528–539.
- [68] Vieira S.M., Sousa J. et Runkler T.A. (2009). Multi-Criteria And Feature Selection Using Fuzzy Classifiers. In : Swarm Intelligence for Multi-objective Problems in Data Mining : Studies in Computational Intelligence, vol. 242, pp. 19–36.
- [69] Sweilam N.H., Tharwat A.A. et Abdel Moniem N.K. (2010). Support vector machine for diagnosis cancer disease : A comparative study. Egyptian Informatics Journal., 11(2), 81–92.

- [70] Prasad Y., Biswas K. et Jain C. (2010). Svm classifier based feature selection using ga, aco and pso for sirnadesign. In Proceedings of the first international conference on advances in swarm intelligence. pp. 307–314.
- [71] Kaur S. et Kalra S. (2016). Feature Extraction Techniques Using Support Vector Machines In Disease Prediction. In Proceedings of the 4th International Conference on Science, Technology and Management (ICSTM-16) , India International Centre, New Delhi.
- [72] Karegowda A. G., Manjunath A. S. et Jayaram M. A. (2011). Application Of Genetic Algorithm Optimized Neural Network Connection Weights For Medical Diagnosis Of Pima Indians Diabetes. International Journal on Soft Computing, 2(2), 15–23.
- [73] Choubey D. K. et Sanchita P. (2015). GA_J48graft DT : A Hybrid Intelligent System for Diabetes Disease Diagnosis. International Journal of Bio-Science and Bio-Technology, 7(5), 135-150.
- [74] Aslam M. A., Zhechen Z. et Asoke K.N. (2013). Feature generation using genetic programming with comparative partner selection for diabetes classification. Expert Systems With Applications, Vol., 40(13), 5402–5412.
- [75] Kumar, G., Ramachandra G. A., et Nagamani K. (2014). An efficient feature selection system to integratingsvm with genetic algorithm for large medical datasets. International Journal of Advanced Research in Computer Science and Software Engineering, 4(2), 272–277.
- [76] Harb H.M. et Desouky A.S. (2014). Feature Selection on Classification of Medical Datasets based on Particle Swarm Optimization. International Journal of Computer Applications, 104(5).
- [77] Sahu B. et Mishra D. (2012). A Novel Feature Selection Algorithm using Particle Swarm Optimization for Cancer Microarray Data. Procedia Engineering, 38, 27-31.

- [78] Zhao W., Wang G., Wang H. B., Chen H. L., Dong H. et Zha Z. D. (2011). A novel framework for gene selection. *Int. J. Adv. Comput. Technol.*, 3, 184–191.
- [79] Chen K.H., Wang K. J., Tsai M.L., Wang K.M., Adrian A.M., Cheng, W.C. et Chang K.S. (2014). Gene selection for cancer identification : a decision tree model empowered by Particle Swarm Optimization algorithm. *BMC bioinformatics*, 15(1), 49.
- [80] Zhu Z. et Ong Y.S. (2007). Memetic Algorithms for Feature Selection on Microarray Data, *Proceedings of the 4th international symposium on Neural Networks : Advances in Neural Networks*.
- [81] Alba E. et al (2007). Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms. *IEEE C Evol Computat.* 9 :284–290.
- [82] Inbarani H., Azar A.T. et Jothi G. (2014). Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis. *Computer Methods and Programs in Biomedicine*, Vol. 113, No. 1, pp. 175-185.
- [83] Long N.C., Meesad P. et Unger H. (2015). A highly accurate firefly based algorithm for heart disease prediction. *Expert Systems with Applications*, Vol. 42, No. 21, pp. 221–8231.
- [84] Panda M. et Abraham A. (2014). Hybrid evolutionary algorithms for classification data mining. *Neural Computation and Application*, Vol. 26, No. 3, pp. 507– 523.
- [85] Ateeq K. et Ganapathy G. (2017). The novel hybrid Modified Particle Swarm Optimization – Neural Network (MPSO-NN) Algorithm for classifying the Diabetes. *International Journal of Computational Intelligence Research* ISSN 0973-1873 Volume 13, Number 4 (2017), pp. 595-614.

- [86] Goswami S., Kumar A. , Chakrabarti A. et Chakraborty B. , A feature cluster taxonomy based feature selection technique, *Expert Systems with Applications : An International Journal*, v.79 n.C, p.76-89.
- [87] Bhardwaj A. et Tiwari A. (2015). Breast cancer diagnosis using Genetically Optimized Neural Network model. *Expert Systems with Applications*, Vol.42, No.10, pp.4611–4620.
- [88] Soliman O.S. et Elhamd E.A. (2015). A Chaotic Levy Flights Bat Algorithm for Diagnosing Diabetes Mellitus'. *International Journal of Computer Applications by IJCA Journal*.
- [89] Tan K., Yu Q., Heng C. et Lee T. (2003). Evolutionary Computing for Knowledge Discovery in Medical Diagnosis. *Artificial Intelligence in Medicine* 27(2), 129–154.
- [90] AlMuhaideb S. et Menai M.E.B. (2014). A new hybrid metaheuristic for medical data classification. *International Journal of Metaheuristics*, Vol.3, No.1, pp. 59–80.
- [91] Sarkar B.K., Sana S.S. et Chaudhuri K. (2012). A genetic algorithm-based rule extraction system. *Applied Soft Computing*. vol.12, pp. 238-254.
- [92] Almuhaideb S. et Menai M.E.B. (2016). *Frontiers of Computer Science*. Volume 10, Issue 6, pp 1082–1102.
- [93] Lo H.Y., Chang C.M., Chiang T.H., et al. (2008) Learning to Improve Area-Under-FROC for Imbalanced Medical Data Classification Using an Ensemble Method. In : *ACM SIGKDD Explorations Newsletter*, vol. 10(2), ACM, New York.
- [94] Bhattacharyya D., Pal A.J. et Kim T. (2011) Cell-graph coloring for cancerous tissue modeling and classification. *Multimedia Tools and Applications*.

- [95] Chorbev I., Mihajlov D. et Jolevski I. (2009). Web Based Medical Expert System with Self Training Heuristic Rule Induction Algorithm. In : Proceedings of the First International Conference on Advances in Databases, Knowledge, and Data Application (DBKDA), pp. 143–148. IEEE Computer Society, Washington.
- [96] Al-Maqaleh B.M. et Shahbazkia H. (2012). A Genetic Algorithm for Discovering Classification Rules in Data Mining. *International Journal of Computer Applications (0975 – 8887)* Volume 41– No.18.
- [97] Bethapudi P. , Reddy E.S et Sitamahalakshmi T. (2015). A Computational Approach for Better Classification of Breast Cancer using Genetic Algorithm. *International Journal of Advanced Trends in Computer Science and Engineering*.
- [98] Szabo A. et De Castro L. N. (2013). A Constructive Data Classification Version of the Particle Swarm Optimization Algorithm. *Mathematical Problems in Engineering* .Volume 2013.
- [99] Liu Y., Qin Z., Shi Z., and Chen J. (2004). Rule discovery with particle swarm optimization. In *Content Computing*, pages 291–296. Springer.
- [100] Wang Z., Sun X. and Zhang D. (2007), A PSO-Based Classification Rule Mining Algorithm, In proceedings of the 3rd International Conference on Intelligent Computing : Advanced Intelligent Computing Theories and Applications.
- [101] Sousa T, Silva A, Neves A. (2004). Particle Swarm based Data Mining Algorithms for classification tasks, *Parallel Computing*, Volume 30, Issues 5-6, 767-783.
- [102] Cervantes A., Galván I. et Isasi P. (2005). A Comparison between the Pittsburgh and Michigan Approaches for the Binary PSO Algorithm. in Pro-

- ceedings of IEEE Congress on Evolutionary Computation (CEC'05), pp. 290–305.
- [103] Ali Z. et Shahzad W. (2017). Comparative Analysis and Survey of Ant Colony Optimization based Rule Miners. *International Journal of Advanced Computer Science and Applications*.
- [104] Pathak A. et Vashistha J. (2015). Classification Rule and Exception Mining Using Nature Inspired Algorithms. *International Journal of Computer Science and Information Technologies*, Vol. 6 (3), 3023-3030.
- [105] Holden N. and Freitas A.A. (2008). A hybrid PSO/ACO algorithm for discovering classification rules in data mining. *Journal of Artificial Evolution and Applications - Particle Swarms : The Second Decade*, Volume 2008.
- [106] Gheraibia Y. et Moussaoui A. (2013). Penguins search optimization algorithm (pesoa). In *Recent Trends in Applied Artificial Intelligence*, pp.222–231, Springer.
- [107] Takahashi A., Sato K., Nishikawa J., Watanuki Y. et Naito Y. (2004). Synchronous diving behavior of Adelie penguins. *Journal of Ethology*, Vol.22,5–11.
- [108] Gaylord S. (1976). *Penguins : Past and Present, Here and There*. Yale University Press.
- [109] Wang P., Zhu Z. et Huang S. (2013). Seven-spot ladybird optimization : a novel and efficient metaheuristic algorithm for numerical optimization. *The Scientific World Journal*. Volume 2013, 11 pages. Article ID 378515.
- [110] Hall M.A. (1998). Correlation-based feature selection for machine learning. PhD thesis, Department of Computer Science, University of Waikato, Hamilton, New Zealand.

- [111] Pearson K. (1901). Principal components analysis. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, vol.6, no.2, p. 559.
- [112] Pearson K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. Philos. Mag. 50, 157.
- [113] Bidi N., Elberrichi Z. (2017). Using Penguins Search Optimization Algorithm for Best Features Selection for Biomedical Data Classification. International Journal of Organizational and Collective Intelligence. Volume 7. Issue 4.
- [114] Bidi N., Elberrichi Z. (2018). Best Features Selection For Biomedical Data Classification Using Seven Spot Ladybird Optimization Algorithm. International Journal of Applied Metaheuristic Computing (IJAMC). Volume 9, Issue 3, Article 4.
- [115] Saroj K., Kumar D. et Kanika D. (2011). A Genetic Algorithm with Entropy Based Probabilistic Initialization and Memory for Automated Rule Mining. In Proceedings of First International Conference on Computer Science and Information Technology, CCSIT, Bangalore, India, Springer, pp. 604–613.
- [116] Blake C.L. et Merz C.J. (1998). UCI repository of machine learning databases. Department of Information and Computer Sciences, University of California, Irvine. <http://www.ics.uci.edu/mllearn/MLRepository.html>.