

République Algérienne Démocratique et Populaire Ministère de
L'Enseignement Supérieur et de la Recherche Scientifique
Université Abbés Laghrour Khenchela
Faculté des Sciences et de la Technologie
Département de Mathématiques et Informatique



Mémoire pour obtenir le diplôme de

Master en Informatique

Spécialité : Génie logiciel et systèmes distribués.

**Une approche d'optimisation par essaim des
chauves-souris pour la sélection d'attributs dans
la classification de données**

Présenté Par :

Ahlem **LEBOUAZDA**

Jury

Dr MAAROUK Toufik	(MCB)	Univ.Abbés Laghrour Khenchela	Examineur
Dr Hichem HOUASSI	(MCB)	Univ.Abbés Laghrour Khenchela	Rapporteur
Dr MERAH El kamel	(MBA)	Univ.Abbés Laghrour Khenchela	Président

2017/2018

Dédicace

«Les batailles de la vie ne sont pas gagnées par les plus forts, ni par les plus rapides, mais par ceux qui n'abandonnent jamais».

La fouine



Remerciements

J'aimerai, en premier lieu, remercier mon Dieu tout puissant qui m'a donné la volonté et la force afin de réaliser ce travail.

Je voudrai remercier grandement, Dr Hichem HOUASSI, qui m'a dirigée. Il a toujours été disponible, à l'écoute de mes questions, et il s'est toujours intéressé à l'avancement de mes travaux. Pour tout cela merci.

J'exprime toute ma reconnaissance aux membres de jury de m'avoir fait l'honneur de participer à mon jury. Mes remerciements

Merci à qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité, à vous, mes parents, ma sœur et mes frères.

Je n'oublie pas de remercier tous mes collègues et amis sans exception.

Résumé

La sélection des attributs est un processus de recherche qui joue un rôle assez important dans la fouille de donnée , elle permet de sélectionner un sous-ensemble pertinents des attributs parmi un ensemble de départ large, en utilisant divers critères et différentes méthodes pour faciliter les taches de la fouille de donnée et surtout la tâche de classification supervisée, dont le but d'améliorer la performance du classificateur .

La sélection d'un sous ensemble d'attributs reste un problème d'optimisation difficile et lourd, il est souhaitable d'avoir des méthodes de sélection approchées (méta-heuristique) rapides et efficaces pour une solution active.

Nous focalisons sur les méta-heuristiques issue de l'intelligence en essaims BA (Binary Algorithm) pour la classification supervisée de donnée.

Dans ce travail, nous avons proposé un algorithme de sélection de sous-ensemble d'attributs pertinents dénoté BBA(Binary Bats Algorithm), les expérimentations ont été effectuées en utilisant une base de donnée IRIS, ces résultats obtenus sont concurrentiel et montre l'efficacité de notre approche proposée par rapport aux autres.

Mots clés: Sélection d'attributs, Méta-heuristiques, Binary Bats(BA), Binary Bats Algorithm (BBA) .

Abstract

Attribute selection is a search process that plays a fairly important role in data mining; it allows you to select a relevant subset of the attributes from a large starting set. Using various criteria and different methods to facilitate the tasks of data mining and especially the task of supervised classification, whose purpose is to improve the performance of the classifier.

The selection of a subset of attributes remains a difficult and cumbersome optimization problems, it is desirable to have quick and efficient methods of selection (meta-heuristics) for an active solution.

We focus on the meta-heuristics derived from intelligence swarms BA (Binary Algorithm) for supervised data classification.

In this work, we have proposed a subset selection algorithm of relevant attributes denoted BBA (Binary Bats Algorithm), the experiments were carried out using an IRIS database, these results are competitive and show the effectiveness of our proposed approach compared to others.

Key words: Attribute Selection, Meta-heuristics, Binary Bats (BA) , Binary Bats Algorithm (BBA).

Table de matières

Dédicace	v
Remerciements	v
Résumé	v
Abstract	v
I.1. Introduction	6
I.2. Extraction de connaissances à partir des données.....	6
I.2.1 Définition	6
I.2.2. <i>Etapas du processus ECD</i>	7
I.2.2.1. La collecte de données.....	7
I.2.2.2. Le prétraitement de données.....	7
I.2.2.3. Fouille de données (Datamining)	7
I.2.2.4. Post traitement (Evaluation et présentation).....	7
I.3. La fouille de données	8
I.3.1 Données, informations et savoir dans la fouille de données.....	9
I.3.2 Les données utilisées dans la fouille de données	10
I.3.3 Les taches de la fouille de données.....	11
I.3.3.1 Association	11
I.3.3.2 Description	12
I.3.3.3 La classification automatique supervisée	12
I.3.3.4 Clustering :	12
I.3.3.5 Prédiction.....	12
I.3.4 Les méthodes de la fouille de données.....	12
I.3.4.1. Les méthodes de visualisation et description	13
I.3.4.2. Les méthodes de structuration et de classification	13
I.3.4.3. Les méthodes d'explication et de prédiction	13
I.3.5 <i>Domaines d'application</i>	13
I.3.5.1. Dans le secteur bancaire	13
I.3.5.2. Dans l'assurance.....	14
I.3.5.3. Dans la télécommunication	14

I.3.5.4. Dans le secteur de commerce	14
I.4. Conclusion.....	14
II.1. Introduction	16
II.2. Classification non-supervisée.....	16
II.3. Classification supervisée	17
II.3.1 Objectif de la classification supervisée	17
II.3.2 Définition formelle.....	17
II.3.3 Les méthodes de la classification supervisée	18
I.3.3.1. Les arbres de décision	18
II.3.3.2. Les Algorithmes génétique.....	19
II.3.3.3. Les réseaux de neurones	21
II.3.4 Les phases de la classification supervisé	22
II.3.4.1. La phase d'apprentissage.....	22
II.3.4.2. La phase de classement	23
II.4. Conclusion.....	24
III.1. Introduction	26
III.2. Sélection d'attribut	26
III.2.1. Le problème de la sélection d'attributs	26
III.2. Définition.....	27
III.3. Pertinence d'un attribut.....	27
III.4. Processus de sélection d'attributs	28
III.4.1. Génération des sous-ensembles d'attributs	29
III.4.2. Evaluation des sous-ensembles	29
III.5. Catégorisation des méthodes de sélection d'attributs	30
III.5.1. Catégorisation basée sur le critère d'évaluation	30
III.5.1.1. Approche de Filtre	30
III.5.1.2. Approche du Wrapper	31
III.5.1.3. Approche Hybride	32
III.5. Catégorisation basé sur la stratégie de recherche.....	33
III.6. Méthodes de la sélection d'attributs.....	35
III.6.1. Les méthodes classique.....	35
a) FOCUS	36

b) Automatic Branch & Bound (ABB)	36
III.6.3. Méthodes heuristiques	38
a) Sequential Forward Selection (SFS)	38
b) Relief	39
III.6. Méthodes Aléatoires	40
a) Las Vegas Filter et Las Vegas Wrapper	40
b) Les algorithmes génétiques pour la sélection d'attributs.....	42
III.7. Méthodes de sélection d'attributs basées sur l'intelligence en essais....	42
III.7.1. L'algorithme des essaims de lucioles (Firfly algorithm).....	43
III.7.2. L'algorithme d'optimisation par les colonies d'abeilles (BCO).....	43
III.7.3. L'algorithme d'optimisation de l'exploration bactérienne (BFO).....	44
III.8. Conclusion.....	45
IV.1. Introduction	47
IV.2. Conception de l'approche BBA proposée.....	47
IV.2.1. Codage des particules	47
IV.2.2. Évaluation des particules	48
IV.2. L'Algorithme BBA	51
IV.3. Expérimentations.....	53
IV.3.1. Matériels et logiciels utilisés	53
IV.3.2. Jeux de données de test.....	54
IV.3.3. Critères de comparaison	55
IV.3.4. Réglage des paramètres.....	55
IV.3.5. Résultats expérimentaux	56
IV.5. Conclusion.....	57
Conclusion Générale.....	55
Annexes.....	56
Bibliographies.....	66

Liste des tableaux

Tableau I. 1. *Modèle de tableau de données* 10

Tableau III. 1. Catégorisation des méthodes de sélection des attributs pour la classification.
..... 35

Tableau IV. 1. Description de la base de données IRIS 54

Liste des figures

Figure I. 1. <i>Processus général d'extraction de connaissances</i>	8
Figure II. 1. <i>Schéma général de la classification.</i>	18
Figure II. 2. <i>Arbre de décision.</i>	19
Figure II. 3. <i>Algorithme génétique</i>	20
Figure II. 4. <i>Réseaux Neurones</i>	22
Figure III. 1 <i>Etapas de sélection d'attributs [34].</i>	29
Figure III. 2. <i>Modèle général de l'approche Filtre [41]</i>	31
Figure III. 3. <i>Modèle général de l'approche Wrapper [41].</i>	32
Figure III. 4. <i>Catégorisation des méthodes de sélection d'attributs [41]</i>	33
Figure IV. 1. <i>Convergence le meilleur moyen G_{best}</i>	56

Liste des algorithmes

<i>Algorithme III. 1. L'algorithme FOCUS</i>	36
<i>Algorithme III. 2. Algorithm Automatic Branch and Bound (ABB)</i>	37
<i>Algorithme III. 3. L'algorithme SFS</i>	38
<i>Algorithme III. 4. Algorithme Relief</i>	40
<i>Algorithme III. 5. Algorithme LVF</i>	41
<i>Algorithme III. 6. Algorithme BCO</i>	44

Liste des acronymes

DM : Data mining.

BA: Bats algorithm.

BBA: Binary bats algorithm.

FS: Feature sélection.

Gbest : Meilleure position connue par toutes les particules de l'essaim.

IRIS : Base de données.

DM : Data mining.

Introduction générale

La fouille de données est un processus d'extraction de connaissance dont l'objectif est la découverte de modèles de données valides, nouveaux, utiles et compréhensibles dans de larges bases de données , il existe différentes tâches de la fouille de données , inclusivement la classification supervisée, association, clustering.

En se focalisant sur la classification supervisée comme une tâche très importante ou son objectif principalement de définir des règles permettant de classer des objets dans des classes.

La sélection d'attributs ou de caractéristiques (Attribute , Feature Selection ou FS) est l'une des étapes les plus importantes dans le traitement de données, c'est une procédure permettant de choisir un sous-ensemble minimum de caractéristiques à partir de l'ensemble original , de sorte que l'espace de caractéristiques soit réduit de façon optimale selon certains critères d'évaluation. La sélection de caractéristiques peut être considérée comme un problème d'optimisation avec de nombreux critères concurrents, qui peuvent être, par exemple : réduire au minimum le nombre de caractéristiques sélectionnées, minimiser l'erreur de classification.

Le problème de la sélection des caractéristiques est, depuis longtemps, un sujet de recherche actif. Les premières approches de sélection de caractéristiques s'appuient sur des méthodes classiques et des algorithmes de recherche tels que la programmation dynamique. Récemment, la tendance de la recherche dans le domaine de la sélection de caractéristiques est orientée vers les méthodes bio-inspirées, en particulier celles issues de l'intelligence en essaim. Ces méthodes ont suscité un grand intérêt dans les deux dernières décennies, et de nombreux algorithmes d'optimisation basés sur l'inspiration biologique ont gagné une énorme popularité.

Les méta-heuristiques sont formellement définie comme une génération itérative processus qui guide une heuristique subordonnée en combinant des concepts intelligemment différents pour explorer et exploiter l'espace de recherche, elle fournissent une solution de qualité en temps raisonnable a des problèmes difficiles par rapport aux méthodes classiques.

De plus, plusieurs travaux ont abordé le même problème avec un large éventail de techniques inspirées par la nature.

Dans le cadre de ce mémoire, nous nous intéressons au problème de la sélection d'attributs pour la classification, et plus précisément, à l'utilisation d'approches d'intelligence en essaim pour effectuer la sélection de caractéristiques dans un contexte de classification.

A cet effet, notre thème de ce mémoire se base sur la proposition d'une nouvelle approche intéressante de méta-heuristique d'optimisation appelé « Binary Bats Algorithm » (BBA), qui a été démontré pour surpasser certains biens connus d'inspiration naturelle techniques d'optimisation.

Cette approche pointe d'une part à réduire la taille de l'ensemble d'attributs utilisés avec une convergence rapide vers la solution optimale globale et d'autre part à améliorer la tâche de classification (augmenter le taux de classification).

Démarche de travail

Après cette section introductive, ce mémoire est organisé de la manière suivante :

Chapitre 1 : Dans ce chapitre on va présenter la fouille de données comme étant une tâche principale dans le processus de l'extraction de connaissances à partir des données (ECD), ses tâches, ses méthodes, ainsi ses domaines d'application

Chapitre 2 : Nous définissons la classification supervisée comme une tâche essentielle dans le processus d'extraction de connaissances en déterminant son propre objectif, ses méthodes, ses domaines d'application, ainsi ses phases.

Chapitre 3 : Nous présenterons un état de l'art sur le problème d'optimisation étudié dans ce travail qui est « la sélection d'attributs » et le processus général de la sélection, ainsi que les différentes méthodes proposées dans la littérature pour résoudre ce problème, en se concentrant sur la classe des méta-heuristiques connues dans la littérature avec une présentation des principales méta-heuristiques surtout les variétés des algorithmes d'optimisation basés sur l'intelligence en essaims.

Chapitre 4 : L'objectif final de ce mémoire est de proposer une approche de sélection d'attributs basée sur une hybridation entre des méthodes bio-inspirées.

Ce chapitre est consacré à la résolution du problème de sélection d'attributs par une forte variante de l'algorithme de chauve-souris appelée algorithme de chauve-souris binaire (BBA) pour construire un modèle hybride pour la classification des données avec une implémentation de BBA.

Enfin, en conclure.

Chapitre I

Extraction des connaissances et fouille de données

I.1. Introduction

Les données brutes, malgré leur quantité qui augmente d'une façon exponentielle, n'ont presque aucune valeur, ce qui est le plus important en fait c'est les connaissances pour lesquelles nous sommes tous assoiffés et qui sont obtenus par la compréhension de ces données, mais plus on a de données plus ce processus devient difficile. De nos jours, les changements de notre environnement sont dénotés par des capteurs qui sont devenus de plus en plus nombreux. Par conséquent, la compréhension de ces données est très importante. Et comme il est dit par Piatestky-Shapiro, « (...) as long as the world keeps producing data of all kinds (...) at an ever increasing rate, the demand for data mining will continue to grow » . D'où la fouille de données devient une nécessité [1].

La fouille de données c'est un concept jeune qui apparaît en 1989 aux Etats-Unis sous un premier nom de KDD (Knowledge Discovery in Databases) en français ECD (Extraction de Connaissances à partir des Données) [2].

I.2. Extraction de connaissances à partir des données

I.2.1 Définition

L'extraction de connaissances à partir de bases de données est un processus qui consiste à passer de données brutes à des connaissances. Fayyad définit ce concept comme « un processus non trivial qui construit un modèle valide, nouveau, potentiellement utile et au final compréhensible, à partir de données. » [15].

L'ECD vise à transformer des données (volumineuse, multiforme, stockées sous différents formats sur des supports pouvant être distribuées) en connaissance.

L'ECD est vu comme une ingénierie pour extraire des connaissances a partir de données [15].

L'ECD se déroule en un ensemble d'étapes complexes.

I.2.2. Etapes du processus ECD

I.2.2.1. La collecte de données

Collecter des informations à partir de plusieurs ressources et les organiser.

I.2.2.2. Le prétraitement de données

Un prétraitement est effectué à la fois sur les individus et sur les variables, consiste à nettoyer les données, les mettre en forme, traiter les données manquantes, échantillonner les individus, sélectionner et construire des variables, on obtient ainsi un ensemble de données cibles. Cette phase a une place importante au sein du processus d'ECD car c'est elle qui va déterminer la qualité des modèles construits lors de la phase de fouille de données.

Elle peut prendre jusqu'à 60% du temps dédié au processus d'ECD.

I.2.2.3. Fouille de données (Datamining)

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.

L'algorithme sélectionné est alors appliqué aux données cibles dans le but de rechercher les structures sous-jacentes des données et de créer des modèles explicatifs ou prédictifs.

I.2.2.4. Post traitement (Evaluation et présentation)

Cette phase est constituée de l'évaluation qui mesure l'intérêt des modèles extraits, pour aboutir de nouvelles connaissances, et de les présenter comme résultats à l'utilisateur.

Le processus (ECD) présent est itératif et plusieurs retours en arrière dans les différentes phases sont nécessaires pour affiner les résultats.

La finalité de l'ECD est de pouvoir traiter des données brutes et volumineuses, et à partir de ces données, d'établir des connaissances directement utilisables par un expert ou un non expert du domaine étudié [16] [18].

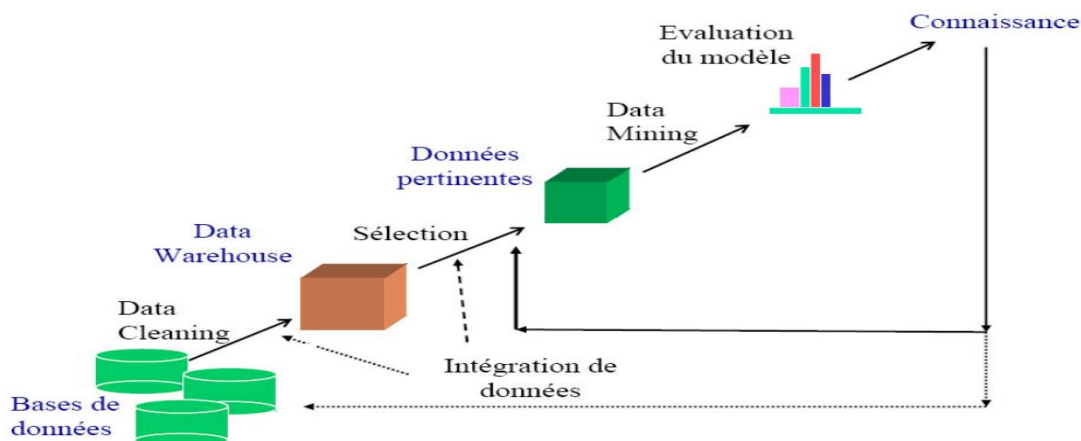


Figure I. 1. *Processus général d'extraction de connaissances.*

I.3. La fouille de données

Le Data mining ou fouille de données est l'une des étapes du processus ECD (voir Figure I.1).

Cette étape regroupe l'ensemble des méthodes et techniques destinées à l'exploration des bases de données de façon automatique, ou semi-automatique, dans l'optique de détecter des règles, des associations, des tendances nouvelles et imprévisibles, des structures particulières restituant l'essentiel de l'information utile [5].

Le terme « Data mining » désigne l'analyse de données depuis différentes perspectives et le fait de transformer ces données en informations utiles. L'extraction des connaissances à partir des données est un processus non trivial d'identification des modèles valides, nouveaux, potentiellement utiles et au final compréhensibles, à partir de données [6] :

Valide : vérifiée par des experts du domaine et correcte dans le futur.

Nouveau : Ce qui est recherché est non prévisible, inconnu.

Utile : utilisé pour prendre des décisions.

Compréhensible : significatif et facile à comprendre

L'exploration de données est un nouveau domaine en croissance rapide. Il tire des idées et des ressources de plusieurs disciplines, y compris l'apprentissage automatique, les statistiques, recherche dans les bases de données, informatique de haute performance et commerce [8].

Le data mining est l'ensemble des méthodes scientifiques destinées à l'exploration et l'analyse de (souvent) grandes bases de données informatiques en vue de détecter dans ces données des profils-type, des comportements récurrents, des règles, des liens, des tendances inconnues (non fixées a priori), des structures particulières restituant de façon concise l'essentiel de l'information utile pour l'aide à la décision [9].

I.3.1 Données, informations et savoir dans la fouille de données

✓ Données

Les données sont des faits, des nombres, ou des textes pouvant être traités par un ordinateur [7]. Une donnée est caractérisée par un ensemble de champs, de caractères, ou encore d'attributs. Un attribut peut être de nature qualitative ou quantitative en fonction de l'ensemble des valeurs qu'il peut prendre :

Attribut qualitatif : Sa valeur est d'un type défini en extension (une couleur, une marque de voiture, . . .).

Attribut quantitatif : L'attribut est de nature quantitative : un entier, un réel, . . . etc.

La valeur d'un attribut est censée représenter une certaine mesure d'une quantité :

Attribut nominal : Les valeurs sont des symboles (des noms) , elles utilisent seuls les tests d'égalité.

Attribut ordinal : Il impose une notion d'ordre, mais il n'est pas possible de calculer directement des distances entre des valeurs ordinales, ainsi les opérateurs de soustraction

Attribut de type intervalle : Les intervalles impliquent une notion d'ordre, et les valeurs sont mesurées dans des unités spécifiques et fixées.

Attribut de type rapport (ratio) : Toutes les opérations mathématiques sont autorisées sur les attributs de ce type.

✓ **Informations**

Les patterns, associations et relations entre toutes ces données permettent d'obtenir des informations [7].

✓ **Savoir**

Les informations peuvent être converties en savoir à propos de patterns historiques ou des tendances futures [7].

I.3.2 Les données utilisées dans la fouille de données

N'importe quel ensemble de données non structuré n'est pas analysable par les méthodes de DM sont appelé tableaux de données. Alors il faudra extraire d'une situation complexe de données, une situation analysable exprimable en termes de tableau de données.

Un tableau de données consiste à deux ensembles d'objets : les lignes correspondent aux individus (ou entités), les colonnes correspondent aux variables, les périodes peuvent être associés aux lignes ou aux colonnes selon les cas.

		Variables				
		X1	X2	X3	Xj	Xj
Individus	1					
	2					
	3					
	i				Xij	

Tableau I. 1. *Modèle de tableau de données*

I : ensemble des individus ;

X : ensemble des variables ;

X_{ij} : désigne l'élément curant du tableau ;

✓ Les variables

Selon la définition de More « toute caractéristique d'une personne ou d'une chose qui peut être exprimé par un nombre est appelée variable.... ».

On distingue deux types de variables : les variables quantitatives (comme les tailles, le pourcentage, le taux, etc.) et les variables qualitative comme on parle de modalités de réponses à une question d'un questionnaire (exemple : « Etes-vous satisfait du service x ? » les modalités seront « oui », « non », « je ne sais pas ».

✓ Les individus ou entités

L'individu est une personne ou une chose qui appartient à une population de référence. Si au lieu d'étudier toute la population, on n'en examine qu'une partie appelée échantillon.

I.3.3 Les taches de la fouille de données

Le data mining est à la mode mais il n'est pas toujours facile pur le non-spécialiste de se faire une idée claire de ce qu'est le data mining.

On dénombre cinq taches du Data mining : [10] [11]

I.3.3.1 Association

La recherche de règles d'association est la tâche la plus intéressante du data mining. C'est également celle qui est la plus répandue dans le monde des affaires, notamment en marketing pour l'analyse du panier de consommation. La recherche de règles d'association cherche à découvrir les règles de quantification ou de relation entre deux ou plusieurs attributs. Les règles d'association sont de la forme « Si antécédent, puis conséquente », avec une mesure confiance associée à la règle. La recherche de règles d'associations dans une grande base de données permet de découvrir des règles cachées utiles pour la prise de décision.

I.3.3.2 Description

Parfois, les chercheurs et les analystes essaient simplement de trouver des façons de décrire des tendances cachées dans les données. Les descriptions des modèles et des tendances servent à expliquer ou vérifier un fait.

I.3.3.3 La classification automatique supervisée

La classification automatique supervisée consiste à examiner les caractéristiques d'un objet nouvellement présenté afin de l'affecter à une classe d'un ensemble prédéfini. Le modèle généré permet, en outre, de prédire ou estimer la valeur manquante ou erronée en utilisant le modèle de classification comme référence [12].

I.3.3.4 Clustering :

La classification automatique non supervisée : C'est une tâche qui vise à identifier des ensembles d'éléments qui partagent certaines similarités. Elle se distingue de la classification automatique supervisée par le fait qu'elle ne se base pas sur des classes prédéfinies [11].

Le principe général de la classification automatique supervisée, également appelée clustering, repose sur le regroupement des objets de telle manière que ceux qui appartiennent à la même classe soient fortement similaires entre eux et fortement dissimilaires avec les objets qui appartiennent aux autres classes. Cette tâche précède, généralement, la phase de clustering.

I.3.3.5 Prédiction

Consiste à estimer une valeur future. En général, les valeurs connues sont historisées. On cherche à prédire la valeur future d'un champ. Cette tâche est proche des précédentes. Les méthodes de classification et d'estimation peuvent être utilisées en prédiction [13].

I.3.4 Les méthodes de la fouille de données

Le data mining, ou la fouille de données dans son sens restreint, est un cœur du processus ECD. Cette phase fait appel à multiples méthodes issues de la statistique, de l'apprentissage automatique, de la reconnaissance de formes ou de visualisation, et des outils issus de différents domaines de l'informatique ou l'intelligence artificielle, en vue de découvrir des connaissances utiles.

Chapitre I _____ Extraction de connaissances et fouille de données

Les méthodes de la fouille de données permettent de découvrir ce que contiennent les données utilisées. [14][15].

Trois grandes familles de méthodes se distinguent :

I.3.4.1. Les méthodes de visualisation et description

L'objectif de ces méthodes est de permettre à l'analyste d'avoir une compréhension synthétique de l'ensemble de ses données. Il s'agit donc principalement d'outils de la synthèse d'information.

Cette synthèse peut s'exprimer par les indicateurs statistiques.

I.3.4.2. Les méthodes de structuration et de classification

En ECD, nous avons affaire à une profusion de données. Décrire ces données s'avère parfois difficile à cause de cette volumétrie.

L'utilisateur cherche souvent à identifier des groupes d'objets semblables au sens d'une métrique donnée. Ces groupes peuvent par exemple correspondre à une réalité ou à des concepts particuliers.

I.3.4.3. Les méthodes d'explication et de prédiction

Ces méthodes ont pour objectif de rechercher à partir des données disponibles un modèle explicatif ou prédictif entre, d'une part, un attribut particulier à prédire et, d'autre part, des attributs prédictifs dans le cas où un tel modèle serait produit et qu'il s'avérerait valide, il pourrait alors être utilisé à des fins de prédictions.

I.3.5 Domaines d'application

La technologie de data mining a une grande importance économique grâce aux possibilités qu'elle offre pour optimiser la gestion des ressources (humaines et matérielles). Parmi les domaines d'application actuels du data mining : [1]

I.3.5.1. Dans le secteur bancaire

Le secteur bancaire est à la tête de tous les autres domaines industriels grâce à son utilisation des techniques du data mining dans ses grandes bases de données clients. Bien que les banques ont employées des outils d'analyse statistiques avec un peu de succès pendant plusieurs années,

les modèles précédemment invisibles des comportements des clients deviennent maintenant plus clair à l'aide des nouveaux outils du data mining. Prédire la réaction des clients aux changements des taux d'intérêt.

I.3.5.2. Dans l'assurance

Les compagnies d'assurance peuvent bénéficier des méthodes du data mining, qui aident les entreprises à réduire les coûts, augmenter les profits, de conserver les clients actuels, d'acquérir de nouveaux clients, et développer de nouveaux produits.

I.3.5.3. Dans la télécommunication

A nos jours, toute activité de télécommunication a utilisé une la technique de data mining pour l'analyse des achats de services de télécommunications, prédiction de modèles d'appels téléphoniques, la gestion des ressources et de trafic réseau, ainsi l'automatisation de la gestion du réseau et de la maintenance en utilisant l'intelligence artificielle pour diagnostiquer et réparer les problèmes de transmission du réseau.

I.3.5.4. Dans le secteur de commerce

Les techniques du data mining ont été très utiles pour le CRM (Customer Relationship Marketing) en développant des modèles pour la prédiction de la propension du client à acheter, l'évaluation des risques pour chaque transaction... etc.

I.4. Conclusion

Le data mining est l'extraction d'informations prédictives cachés dans de grandes base de données. C'est une technologie nouvelle et puissante qui donne la possibilité aux entreprises de se concentrer sur les informations les plus importantes dans leurs données. Les outils du data mining peuvent prédire les futurs tendances et actions, permettant de prendre les bonnes décisions. C'est ce qui rend le data mining la technologie la plus importantes

Chapitre II

La classification supervisée

II.1. Introduction

La classification supervisée est une tâche de l'étape de fouille de données dans le processus d'extraction de connaissances à partir des bases de données. Elle est largement appliquée dans la vie courante. En effet, il existe une multitude de problèmes qui entrent dans ce cadre, parmi lesquels on trouve la reconnaissance des caractères manuscrits, la reconnaissance des paroles, la catégorisation des textes, la détection des spams, l'aide au diagnostic médical, la bioinformatique...etc. [19]

A la différence de la classification non supervisée où les groupes d'objets sont découverts à posteriori, les techniques de classification supervisée s'appliquent lorsqu'on veut rattacher un nouvel objet (observation) à une classe qui est choisie parmi un ensemble de classes connues. Cette étape suit, généralement, l'étape de clustering. La plupart des algorithmes de classification tentent de trouver un modèle (une fonction mathématique) qui explique le lien entre les données d'entrée et les classes de sortie. Un ensemble d'apprentissage (ensemble de classes) est donc utilisé par l'algorithme. Cette méthode de raisonnement est appelée inductive car on induit la connaissance (le modèle) à partir des données d'entrée (objets à classifier) et des sorties (leurs classes). Grâce à ce modèle, on peut alors prédire les classes de nouvelles données. Le modèle est bon s'il permet de bien prédire [20].

II.2. Classification non-supervisée

Cette classification est aussi appelée "classification automatique", "clustering" ou encore "regroupement". Dans ce type de classification on est amené à identifier les populations d'un ensemble de données. On suppose qu'on dispose d'un ensemble d'objets que l'on note par $X = \{x_1, x_2, \dots, x_n\}$ caractérisé par un ensemble de descripteurs D ,

L'objectif du clustering est de trouver les groupes auxquels appartiennent chaque objet x qu'on note par $C = \{C_1, C_2, \dots, C_n\}$. Ce qui revient à déterminer une fonction notée Y_s qui associe à chaque élément de X un ou plusieurs éléments de C .

Il faut pouvoir affecter une nouvelle observation à une classe. Les observations disponibles ne sont pas initialement identifiées comme appartenant à telle ou telle population.

Parmi les méthodes non-supervisées les plus utilisées, citons deux types d'approches : les centres mobiles (k-means) et la classification hiérarchique.

II.3. Classification supervisée

La classification est la tâche la plus connue de la fouille de données et qui semble être une obligation humaine, afin de comprendre notre vie quotidienne, nous sommes constamment classifiés, catégorisés et évalués.

Cette tâche supervisée consiste à étudier les caractéristiques d'un nouvel objet pour lui attribuer une classe prédéfinie, elle est caractérisée par une définition de classes bien précise et un ensemble d'exemples classés auparavant.

II.3.1 Objectif de la classification supervisée

L'objectif est de créer un modèle qui peut être appliqué aux données non classifiées dans le but de les classifiées [21].

II.3.2 Définition formelle

La classification supervisée est un processus composé d'une phase d'apprentissage suivie d'une phase de classement [19].

Dans la classification supervisée, on dispose d'un ensemble X de N exemples, des couples (donnée, étiquette), chaque donnée $x_i \in D$ est caractérisée par P attributs et par sa classe $y_i \in Y$.

Dans un problème de classification supervisée, la classe prend sa valeur parmi un ensemble Y fini.

Le problème consiste alors, en s'appuyant sur l'ensemble d'exemples $X = (x_i, y_i)$, $i \in \{1.. N\}$, à prédire la classe de toute nouvelle donnée $x \in D$ [22].

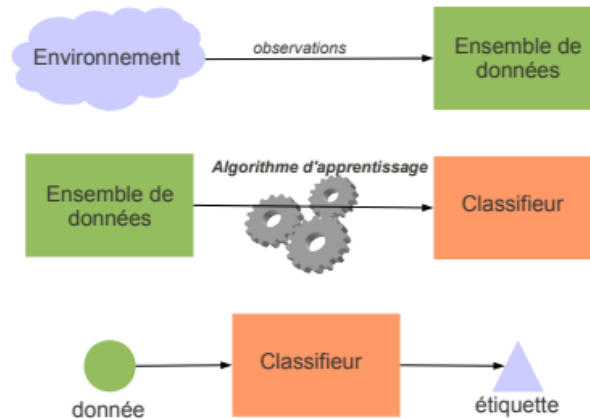


Figure II. 1. Schéma général de la classification.

II.3.3 Les méthodes de la classification supervisée

Les systèmes d'apprentissage permettant d'obtenir une telle procédure peuvent être basés sur des hypothèses probabilistes (classifieur naïf de Bayésien), sur des notions de proximité (plus proches voisins) ou sur des recherches dans des espaces d'hypothèses (arbres de décisions) [23].

I.3.3.1. Les arbres de décision

Les arbres de décisions sont des outils d'aide à la décision qui permettent selon des variables discriminantes de répartir une population d'individus en groupes homogènes en fonction d'un objectif connu. Les arbres de décision sont des outils puissants et populaires pour la classification et la prédiction.

Un arbre de décision permet à partir des données connues sur le problème de donner des prédictions par réduction niveau par niveau du domaine des solutions.

Chaque nœud interne d'un arbre de décision permet de répartir les éléments à classifier de façon homogène entre ses différents fils en portant sur une variable discriminante de ces éléments. Les branches qui représentent les liaisons entre un nœud et ses fils sont les valeurs discriminantes de la variable du nœud. Et en fin, les feuilles d'un arbre de décision représentent les résultats de la prédiction des données à classifier (voir la figure 2) [1]

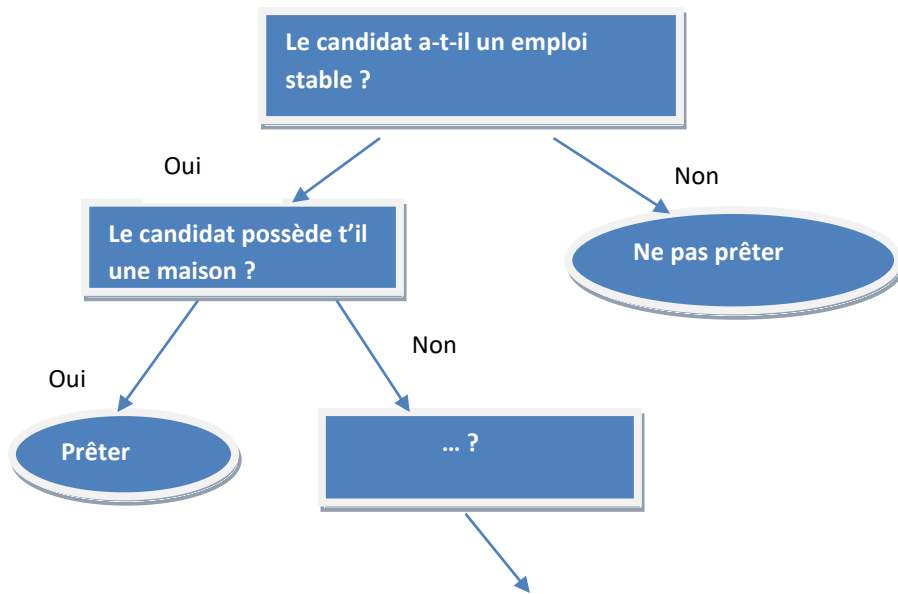


Figure II. 2. *Arbre de décision.*

Avantages

- ✓ Les arbres de décision sont capables de produire des règles compréhensibles.
- ✓ Les arbres de décision effectuent la classification sans exiger beaucoup de calculs.
- ✓ Les arbres de décision sont en mesure de manipuler à la fois les variables continues et catégorielles.

Inconvénients

- ✓ Manque de performance dans le cas de plusieurs classes; les arbres deviennent très complexes et ne sont pas nécessairement optimaux.
- ✓ Demande beaucoup de temps de calcul lors de la construction (le choix du meilleur partitionnement) et l'élagage (la comparaison de sous-arbres).
- ✓ Moins bonnes performances concernant les prédictions portant sur des valeurs numériques [1].

II.3.3.2. Les Algorithmes génétique

Un algorithme génétique se constitue d'une catégorie de programmes dont le principe est la reproduction des mécanismes de la sélection naturelle pour résoudre un problème donné (Figure 2). L'optimisation des problèmes combinatoires et surtout les problèmes dits NP-complets (dont le temps de calcul croit de façon non polynomiale

avec la complexité du problème) est l'objectif principale des algorithmes génétiques, ils sont particulièrement adaptés à ce type de problèmes.

Ces algorithmes constituent parfois une alternative intéressante aux réseaux de neurones mais sont le plus souvent complémentaires [1].

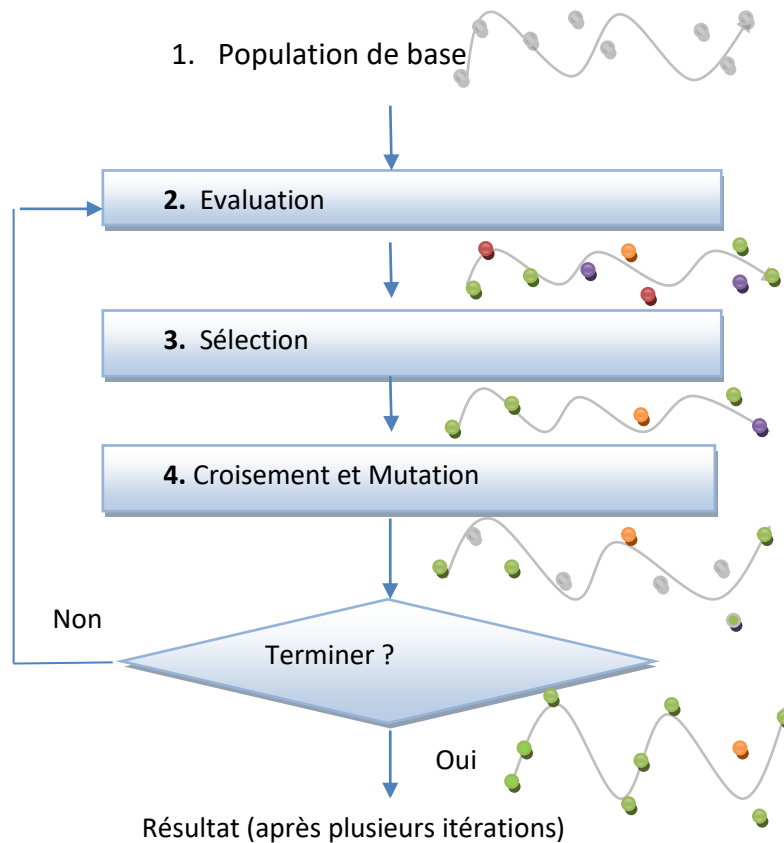


Figure II. 3. *Algorithme génétique.*

Avantages

- ✓ Ils utilisent l'évaluation de la fonction objective sans prendre en compte sa nature ce qui lui donne plus de souplesse et un large domaine d'application.
- ✓ Ils sont dotés de parallélisme car ils travaillent sur plusieurs points en même temps il s'agit des individus de la population.
- ✓ L'utilisation de règles de transition probabilistes de croisement et de mutation permet dans certains cas d'éviter des optimums locaux et d'aller vers un optimum global.

Inconvénients

- ✓ Temps de calcul très élevé car ils nécessitent de nombreux calculs particulièrement au niveau de la fonction d'évaluation.
- ✓ Difficiles à mettre en œuvre à cause :
 - ✚ Des paramètres parfois difficiles à déterminer comme la taille de la population ou le taux de mutation. Ce qui implique la nécessité de plusieurs essais car le succès de l'évolution en dépend, ce qui limite encore l'efficacité de l'algorithme.
 - ✚ Du choix de la fonction d'évaluation qui est critique, elle doit prendre en compte les bons paramètres du problème. Elle doit donc être choisie avec soin.

- ✓ Il est impossible d'être sûr que la solution obtenue après un nombre fini de générations soit la meilleure, on peut seulement être sûr que l'on s'est approché de la solution optimale [1].

II.3.3.3. Les réseaux de neurones

Un réseau de neurones est un modèle de calcul dont le fonctionnement vise à simuler le fonctionnement des neurones biologiques, il est constitué d'un grand nombre d'unités (neurones) ayant chacune une petite mémoire locale et interconnectées par des canaux de communication qui transportent des données numériques.

Ces unités peuvent uniquement agir sur leurs données locales et sur les entrées qu'elles reçoivent par leurs connections. Les réseaux de neurones sont capables de prédire de nouvelles observations (sur des variables spécifiques) à partir d'autres observations (soit les mêmes ou d'autres variables) après avoir exécuté un processus d'apprentissage sur des données existantes.

La phase d'apprentissage d'un réseau de neurones est un processus itératif permettant de régler les poids du réseau pour optimiser la prédiction des échantillons de données sur lesquelles l'apprentissage a été fait. Après la phase d'apprentissage le réseau de neurones devient capable de généraliser.

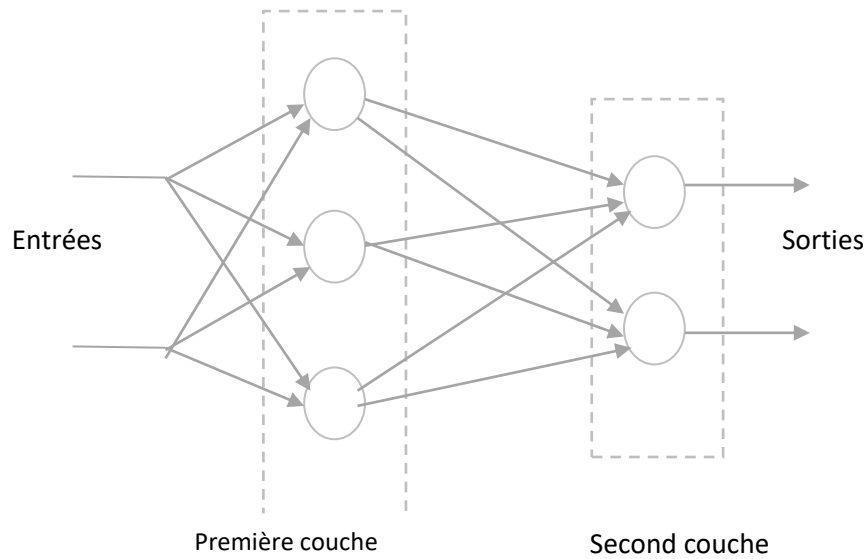


Figure II. 4. Réseaux Neurones.

Avantages

- ✓ Les réseaux de neurones sont théoriquement capables d'approximer n'importe quelle fonction continue et ainsi le chercheur n'a pas besoin d'avoir aucune hypothèse du modèle sous-jacent.

Inconvénients

- ✓ Généralement les réseaux de neurones ne sont pas souvent utilisés dans les tâches du datamining parce qu'ils produisent des modèles souvent incompréhensibles et demande un long temps d'apprentissage [19].

II.3.4 Les phases de la classification supervisée

La classification supervisée est un processus composé d'une phase d'apprentissage suivie d'une phase de classement :

II.3.4.1. La phase d'apprentissage

La phase d'apprentissage consiste à construire un modèle (ou classifieur) qui approxime au mieux la fonction à partir d'un ensemble d'exemples sélectionnés de manière aléatoire dans l'ensemble d'apprentissage.

Dans le cas où l'on a plusieurs classes, on parle d'apprentissage multi-classes.

Si le système de classification ne peut traiter qu'une seule classe, le mode opératoire dans le cas de plusieurs classes, consistera à faire autant d'apprentissages qu'il y a de classes où les instances positives sont celles de la classe apprise et les instances négatives sont les exemples des autres classement [19].

II.3.4.2. La phase de classement

Dans la phase de classement le modèle appris F est utilisé pour affecter une classe à chaque nouvel exemple. Il s'agit de déterminer $Y_k = f(Ok)$. Pour valider le modèle appris, un ensemble test contenant des exemples dont on connaît la classe Y_i est utilisé.

Le modèle commet une erreur lorsque Y_k est différent de $f(Ok)$. Le modèle appris F approxime au mieux la fonction $f(OK)$ lorsque le nombre d'erreurs calculé tend vers zéro. Ce qui revient à déterminer le taux d'erreur du modèle (taux d'erreur = 1 – taux de précision) qui exprime le pourcentage du nombre d'exemples mal classés par rapport au nombre total d'exemples.

Si le modèle est validé sur l'ensemble test alors il est utilisé pour classer les exemples dont on ne connaît pas la classe [19].

II.4. Conclusion

Nous avons constaté à travers cette étude des algorithmes de classification supervisée qu'il en existe plusieurs, certains sont issus des recherches sur l'apprentissage (machine learning) comme les arbres de décision et l'algorithme k-NN, d'autres de l'intelligence artificielle comme les réseaux de neurones. Ces algorithmes diffèrent principalement par :

- La capacité de traiter n'importe quel type de données (comme k-NN, les arbres de décision).
- Le temps de traitement (l'algorithme k-NN par exemple présente un temps de traitement relativement coûteux dû au calcul des voisins).
- La sensibilité aux bruits (les arbres de décision sont très sensibles aux bruits par rapport aux autres algorithmes).

La classification supervisée peut mesurer l'importance de chaque attribut pour classer de nouveaux objets

Chapitre III

Prétraitement et Sélection des Attributs

III.1. Introduction

La sélection d'attributs est une thématique de recherche assez active depuis plusieurs décennies et en cours de développement dans divers application. Elle constitue une étape importante dans le prétraitement des données de grande dimension acheminées vers la classification supervisée ou non supervisée. En effet l'apparition des grandes bases de données dans le domaine de l'apprentissage et les systèmes de fouille de données "Data Mining" a exigé une réduction de dimension, avant d'entamer la tâche de classification des données.

Dans ce chapitre, nous allons présenter le problème de sélection d'attributs ainsi que les différentes méthodes proposées dans la littérature pour résoudre ce problème.

III.2. Sélection d'attributs

III.2.1. Le problème de la sélection d'attributs

La sélection d'attributs joue un rôle très important en classification lorsqu'un grand nombre P de variables sont disponibles, certaines pouvant être peu significatives, corrélées ou non pertinentes [27].

Elle consiste à sélectionner un sous-ensemble de variables. La sélection permet également de faciliter l'étape d'apprentissage et de réduire la complexité des algorithmes ainsi que les temps de calcul [28].

La sélection non supervisée de caractéristiques reste un problème difficile et lourd en temps de calcul. Lorsqu'il s'agit de traiter des bases de données de grande taille, qu'il n'est pas possible d'étiqueter manuellement, il est souhaitable d'avoir des méthodes de sélection rapides et efficaces.

III.2. Définition

La sélection d'attributs est un procédé permettant de choisir un sous-ensemble optimal d'attributs pertinents, à partir d'un ensemble original des attributs, selon un certain critère de performance. En fait, le choix d'un ensemble optimal de descripteurs, ne signifie pas obligatoirement la sélection d'un ensemble composé seulement des attributs jugés pertinents et utiles. Il peut y contenir des attributs non pertinents, mais qui ont de meilleures performances, prendre avec d'autres attributs [29].

Le problème de sélection d'attributs peut être défini par: Soit $F = \{f_1, f_2 \dots f_n\}$ un ensemble d'attributs de taille N ou N représente le nombre total d'attributs étudiées.

Soit E_v une fonction qui permet d'évaluer un sous-ensemble d'attributs. Nous supposons que la plus grande valeur de E_v soit obtenue pour le meilleur sous-ensemble d'attributs. [37]

L'objectif de la sélection est de trouver un sous-ensemble F' ($F' \subseteq F$) de taille N' ($N' \leq N$) tel que :

$$E_v(F') = \max_{Z \subseteq F} E_v(Z) \quad (1)$$

Où $|Z| = N'$ et N' est, soit un nombre prédéfini par l'utilisateur.

La sélection d'attributs est un processus qui consiste à chercher dans l'ensemble des variables explicatives disponibles un ensemble optimal des caractéristiques les plus importantes à un système donné [26].

III.3. Pertinence d'un attribut

La performance d'un algorithme d'apprentissage dépend fortement des attributs utilisés dans la tâche d'apprentissage. La présence d'un attribut redondant ou non pertinent peut réduire cette performance.

Dans la littérature, il existe plusieurs définitions de la pertinence d'un attribut, la plus connue est celle de [37]. Selon cette définition, un attribut est classé comme étant très pertinent, peu pertinent et non pertinent :

Très pertinent : Un attribut F_i est dit très pertinent si son absence entraîne une détérioration significative de la performance du système de classification utilisé.

Peu pertinent : Un attribut F_i est dit peu pertinent s'il n'est pas "très pertinent" et s'il existe un sous-ensemble V tel que la performance de $V \cup \{F_i\}$ soit significativement meilleure que la performance de V .

Non pertinent : Les attributs qui ne sont ni "peu pertinents" ni "très pertinents" représentent les attributs non pertinents. Ces attributs seront en général supprimés de l'ensemble d'attributs de départ [30].

III.4. Processus de sélection d'attributs

La sélection d'attributs est un domaine de recherche actif et en cours de développement [31] [32].

Selon [33] une procédure de sélection d'attributs est généralement composée de quatre étapes illustrées par la Figure III.1.

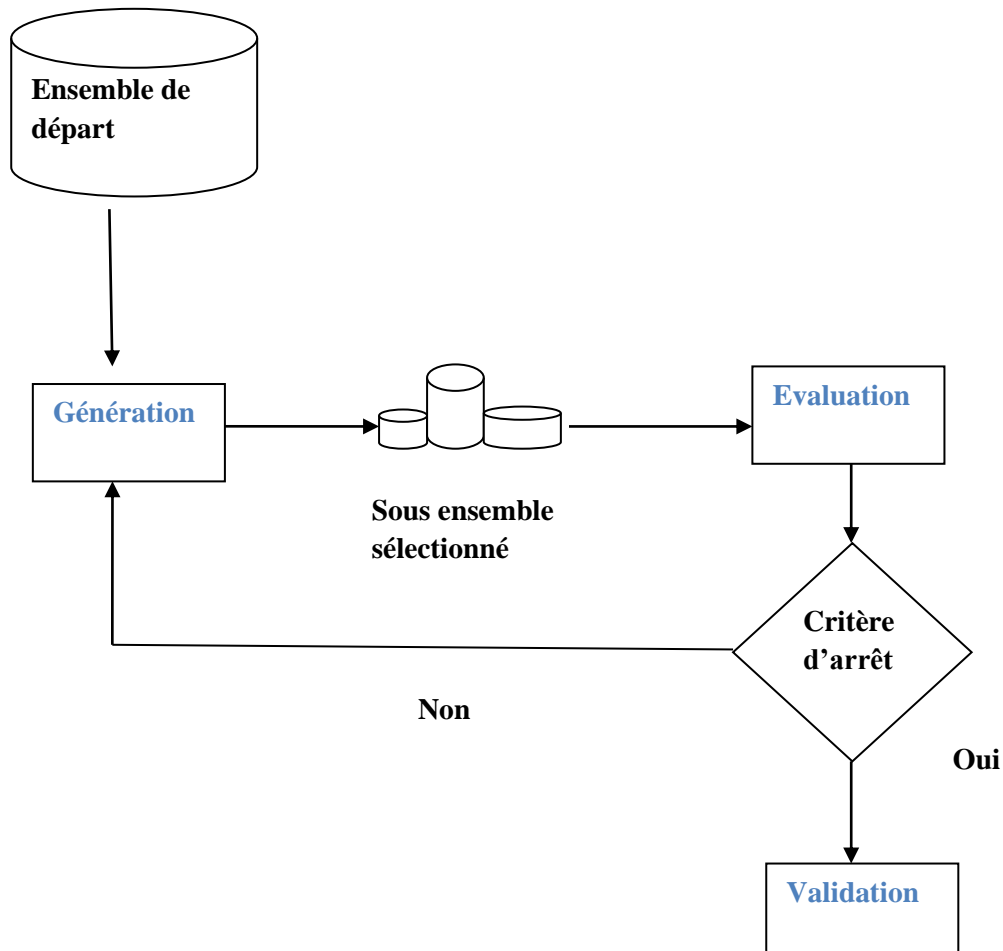


Figure III. 1 *Etapes de sélection d'attributs [34].*

III.4.1. Génération des sous-ensembles d'attributs

Dans le cadre de la sélection d'attributs, la procédure de génération désigne la façon de générer l'ensemble d'attributs candidat à examiner parlent aussi de procédure de recherche. Le principe général consiste à générer successivement des sous-ensembles d'attributs à évaluer.

III.4.2. Evaluation des sous-ensembles

L'objectif associé à la fonction d'évaluation est de mesurer la capacité d'un attribut, ou d'un ensemble d'attributs, à discriminer les classes de la partition impliquée par l'attribut endogène. L'optimalité d'un sous-ensemble est relative à la fonction d'évaluation utilisée.

Les auteurs de [9] considèrent que ces fonctions peuvent être regroupées en cinq catégories qui sont les suivantes : information, distance, consistance, précision.

III.4.3. Critère d'arrêt

Certains critères doivent être définis pour arrêter le processus de recherche sur les sous-ensembles d'attributs [34].

Le critère d'arrêt peut être un temps de calcul fixé, un nombre d'itérations fixé, ou encore le fait que les sous-ensembles candidats deviennent trop homogènes, etc. [35]

III.4. Validation

Cette étape consiste à vérifier si le sous-ensemble est valide ou non, en d'autre terme si l'objectif souhaité est atteint ou non. Une manière de faire la validation des résultats est de mesurer directement ces derniers en utilisant des connaissances a priori sur les données.

Par exemple, si nous employons le taux d'erreur de classification comme un indicateur de performance pour le traitement, pour un sous-ensemble d'attributs sélectionné, nous pouvons simplement suivre l'expérience "avant et après" pour comparer le taux d'erreur de classificateur sur l'ensemble complet d'attributs et sur le sous-ensemble sélectionné [35].

III.5. Catégorisation des méthodes de sélection d'attributs

Dans la littérature, différentes méthodes ont été développées et utilisées pour la sélection d'attributs, Les auteurs proposent plusieurs catégorisations de ces méthodes selon les stratégies de recherche et les fonctions d'évaluations utilisées.

III.5.1. Catégorisation basée sur le critère d'évaluation

Plusieurs auteurs préfèrent séparer les méthodes de sélection de caractéristiques en trois approches (Filtre, Wrapper et hybride) [45] [40].

III.5.1.1. Approche de Filtre

Cette approche est réalisée comme un prétraitement, elle exploite les propriétés intrinsèques des attributs utilisées, sans référence à une quelconque application. Elle se base sur la performance de la fonction d'évaluation calculée directement sur l'ensemble d'apprentissage comme : la distance, l'information, la dépendance, et la cohérence [56].

La **Figure III. 2** présente le modèle de l'approche filtre pour la sélection d'attributs.

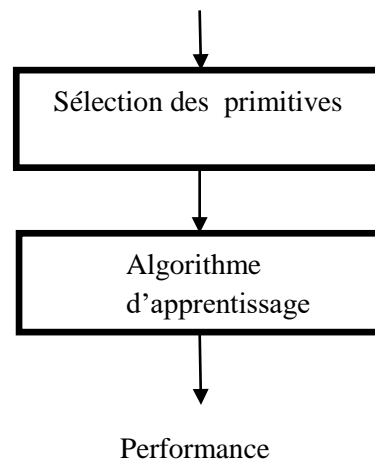


Figure III. 2. *Modèle général de l'approche Filtre [41].*

III.5.1.2. Approche du Wrapper

L'approche Wrapper utilise l'algorithme d'apprentissage comme une fonction d'évaluation, elle définit donc la pertinence des attributs par l'intermédiaire d'une prédiction de la performance du système final. L'approche filtre est plus rapide que l'approche Wrapper en terme de génération de résultats. Cependant, cette dernière a l'avantage de fournir généralement des résultats plus pertinents pour la classification. [57]. La **Figure III.3** présente le modèle général de l'approches Wrapper pour la sélection des attributs.



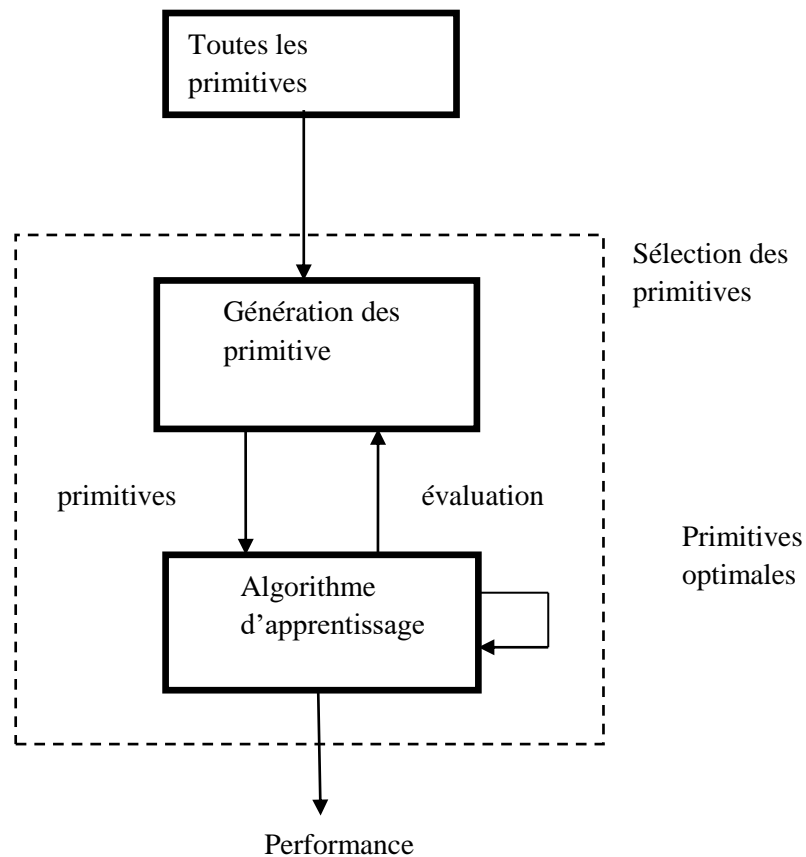


Figure III. 3. *Modèle général de l'approche Wrapper [41].*

III.5.1.3. Approche Hybride

Les algorithmes hybrides ont été plus récemment proposés par rapport aux précédents (filtre et wrapper). Dans ces algorithmes, d'abord, une mesure de « goodness » de sous ensembles d'attributs, basée sur les attributs des données, est utilisée pour choisir les meilleurs sous-ensembles pour une cardinalité donnée. Par la suite, la validation croisée est exploitée pour décider du meilleur sous-ensemble final entre les différentes cardinalités. Ces algorithmes se concentrent principalement sur la combinaison des algorithmes Filtre et Wrapper pour obtenir les meilleures performances possibles, avec un algorithme d'apprentissage particulier et une complexité temporelle comparable à celle des algorithmes de type filtre [57].

III.5. Catégorisation basé sur la stratégie de recherche

Une autre catégorisation des méthodes de sélection des attributs est celle basée sur la stratégie de recherche. Liu et Yu [40] proposent de séparer les méthodes en fonction de la stratégie utilisées (complète, heuristique ou aléatoire).

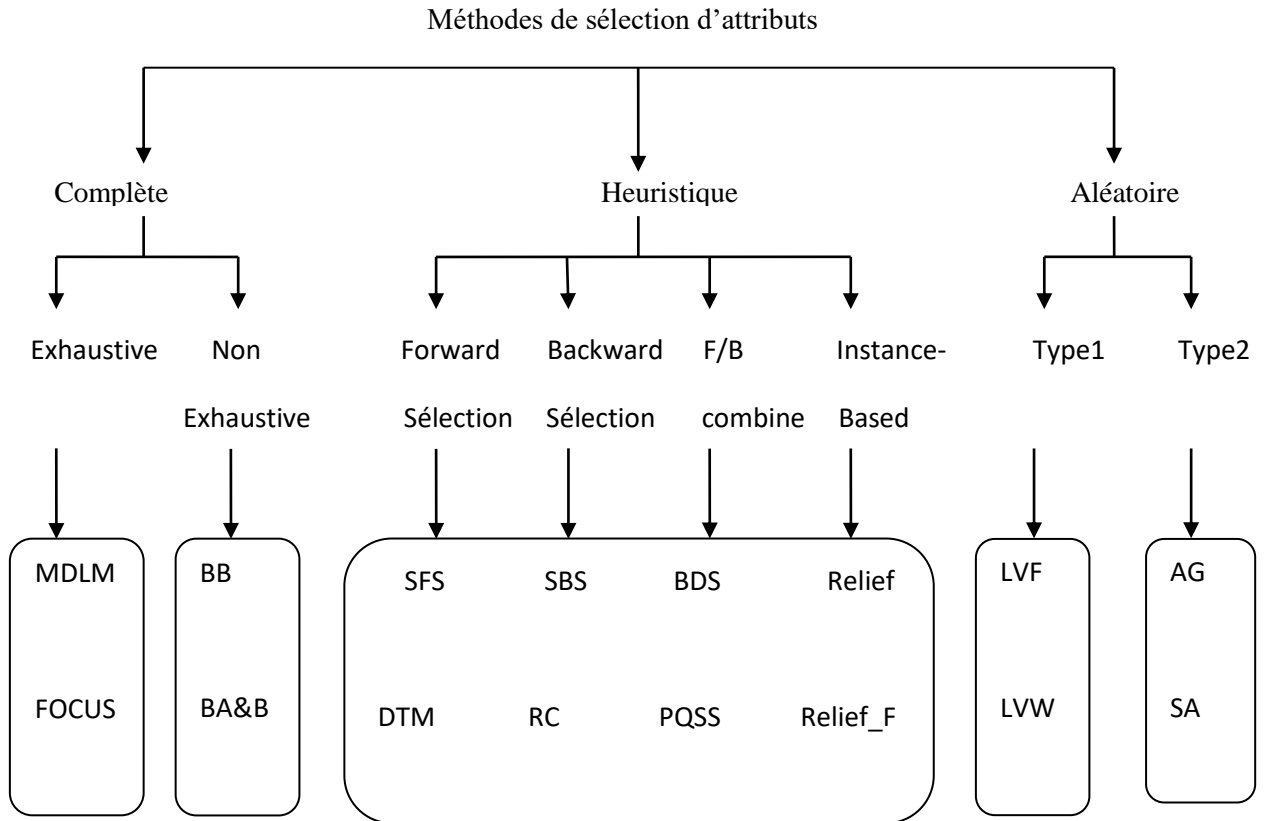


Figure III. 4. *Catégorisation des méthodes de sélection d'attributs [41]*

La Figure III.4 présente une classification des méthodes de sélection des attributs basée sur les trois types de procédures de génération (La procédure de génération). Les procédures complètes sont subdivisées en «exhaustive» et «non exhaustive», les procédures heuristiques sont subdivisées en «sélection forward», «sélection backward», «forward /backward combinés», et les catégories «d'instance-based». De même, les procédures de génération aléatoires sont regroupés en «type I» et «type II».

Les auteurs de [57] ont proposé deux dimensions afin de catégoriser les méthodes de sélection des attributs pour la classification: la stratégie de recherche (complète, séquentielle et aléatoire) et le critère d'évaluation (Filtre, Wrapper et Hybride).

Plusieurs méthodes de sélection de caractéristiques sont collectées dans le Tableau III.

1 selon cette catégorisation :

			La stratégie de recherche		
			Complète	Séquentielle	Aléatoire
Critère D'éva- luation	Filtre	Distance	B&B BFF	Relief ReliefF SFS Segen's	
		information	MDLM	DTM Koller's	
		Dépendance	Bobrowski's	POE+ACC	
		Cohérence	FOCUS ABB MIFFES_1Schli- Mmer's	Set Cover	LVF

	Wrapper	L'exactitude Des prévision	BS AB&B	SBS-SLASH PQSS RCQueiros' Xing's	SA RGS LVW RMHC AG RVE
--	----------------	---	------------------------	---	---

Tableau III. 1 . *Catégorisation des méthodes de sélection des attributs pour la classification [40].*

III.6. Méthodes de la sélection d'attributs

III.6.1. Les méthodes classique

Un grand nombre de méthodes classiques de sélection d'attributs sont décrites dans la littérature, nous avons choisi d'en présenter certaines, parmi les plus connues, en les catégorisant selon la technique utilisée pour la procédure de génération de sous ensembles (génération des sous ensemble).

III.6.2. Méthodes complètes

Ces méthodes examinent toutes les combinaisons d'attributs possibles. Elles ont une complexité de calcul très élevée (espace de recherche de l'ordre de $O(2^N)$ pour N attributs) mais elles garantissent de trouver le sous ensemble optimal d'attributs. A titre d'exemples de ces méthodes, nous allons présenter : FOCUS [39] et Automatic Branch & Bound [44].

a) FOCUS

C'est une méthode simple [44] qui utilise la mesure de l'incohérence pour évaluer les sous-ensembles d'attributs. Elle considère toutes les combinaisons d'attributs à partir d'un sous-ensemble vide, et l'évaluation des sous-ensembles avec un attribut, des sous-ensembles avec deux attributs, et ainsi de suite. La réalisation de cet algorithme est donnée dans l'**algorithme III. 1**.

Entrées : F l'ensemble complet d'attributs, INCONSIST () : mesure de l'incohérence.

S ← □

Pour i ← 1 à |F| **Faire**

Pour chaque sous-ensemble S de taille I **Faire**

Si INCONSIST(S) = 0

Retourner S

Fin pour

Fin pour

Algorithme III. 1. L'algorithme FOCUS [45].

Cet algorithme doit générer $\sum_{i=1}^m \binom{N}{i} = \sum_{i=1}^M \frac{N!}{i!(N-i)!}$ sous-ensembles afin de trouver le sous-ensemble de m attributs (dans un ensemble de N attributs) qui satisfait la mesure de l'incohérence. Il est clair que le nombre de sous-ensembles d'attributs évalués est trop grand, même en cas de valeurs modérées de m.

b) Automatic Branch & Bound (ABB)

ABB (Automatic Branch & Bound) c'est une procédure de recherche BB optimale automatisée qui utilise la stratégie backward. C'est une variante ayant sa limite comme le taux d'incohérence des données lorsque l'ensemble complet d'attributs est utilisé (**Algorithme III.2**). Elle commence par l'ensemble complet d'attributs et supprime un attribut, à chaque fois, dans un algorithme de recherche en largeur d'abord (breadthfirst search) (jusqu'à ce qu'aucun attribut ne puisse être enlevé alors que le critère de l'incohérence est respecté [56]).

Entrées : $S(X)$: un échantillon S décrit par X , $|X| = n$, J : mesure d'évaluation (monotonique)

Procédure ABB ($S(X)$: échantillon ; var L' : liste des ensembles)

Pour chaque x dans X **Faire**

en queue ($Q, X \setminus \{x\}$) // supprimer un attribut à chaque fois.

Fin pour

Tant que non vide (Q) **Faire**

$X' :=$ de queue (Q) // X' est légitime si elle n'est pas un sous-ensemble d'un état coupé

Si légitime (X') et $J(S(X')) \geq J_0$ **Faire**

$L' :=$ append (L', X')

ABB($S(X'), L'$)

Fin si

Fin tant que

Fin

Début

$Q :=$ // file d'attente d'états en instance

$L' := [X]$ // Liste des solutions

$J_0 := J(S(X))$ // la valeur minimale autorisée de J

ABB ($S(X), L'$) // appelle initial à ABB

$k :=$ la plus petite taille d'un sous-ensemble dans L'

$L :=$ ensemble d'éléments de L' de taille k

Fin

Sortie : L : toutes les solutions équivalentes trouvées.

Algorithme III. 2. *Algorithm Automatic Branch and Bound (ABB) [47].*

III.6.3. Méthodes heuristiques

La recherche s'effectue de sorte qu'il ne soit pas nécessaire d'évaluer l'ensemble de tous les sous-ensembles d'attributs possibles. Ainsi, une plus grande vitesse est atteinte, étant donné que l'espace de recherche est plus petit que dans le cas des méthodes complètes. Les méthodes heuristiques ne garantissent pas de trouver le sous-ensemble optimal, nous allons, à titre d'exemple présenter les méthodes Sequential Forward Selection (SFS), Sequential Backward Selection (SBS) et Relief [48].

a) Sequential Forward Selection (SFS)

Sequential Forward Selection est l'algorithme de recherche gloutonne (greedy) le plus simple, il a été proposé en 1963 par Marill [49].

A partir de l'ensemble vide, ajouter successivement le meilleur attribut X^+ qui est trouvé par l'augmentation de la fonction objective $J(Y_k + X^+)$ lorsqu'il est combiné avec les attributs Y_k qui ont déjà été sélectionnés.

1. Commencer avec un ensemble vide $Y_0 = \{ \}$
2. Sélectionner le meilleur attribut $X^+ = \operatorname{argmax} [J(Y_k + X)]; x \supset Y_k$
3. Mettre à jour $Y_{k+1} = Y_k + X^+; k = k + 1$
4. Aller à 2

Algorithme III. 3. L'algorithme SFS [50].

SFS donne les meilleures performances lorsque le sous-ensemble optimal a un petit nombre d'attributs. Lorsqu'on est proche de l'ensemble vide, un grand nombre d'états peuvent être potentiellement évalués. Quand on est proche de l'ensemble complet, la région examinée par SFS est étroite car la plupart des attributs ont déjà été sélectionnés. L'espace de recherche est dessiné comme une ellipse afin d'insister sur le fait qu'il y a moins d'états vers l'ensemble vide ou plein. Le principal inconvénient de SFS est qu'il est incapable de supprimer des attributs qui deviennent impertinents après l'ajout d'autres attributs.

b) Relief

Relief [59] est un algorithme basé sur les poids d'attributs. Cet algorithme est inspiré de l'apprentissage basé sur les exemples (instance-based learning). En tenant compte des données d'apprentissage S , de la taille de l'échantillon m , et d'un seuil de pertinence τ ($0 \leq \tau \leq 1$), Relief détecte les attributs qui sont statistiquement pertinents au concept cible. Nous supposons que le type de chaque attribut est soit nominal (y compris booléen) ou numérique (entier ou réel). La différence entre les valeurs d'attributs de deux instances X et Y est définie par la fonction suivante :

$$\text{Diff}(x_v, y_v) = \begin{cases} 0 & \text{Si } x_v \text{ et } y_v \text{ sont identiques} \\ 1 & \text{si } x_v \text{ et } y_v \text{ sont différents} \end{cases} \quad (1)$$

Dans le cas où x_v et y_v sont numériques.

Dans le cas où x_v et y_v sont nominales

$$\text{Diff}(x_v, y_v) = (x_v, y_v) / n\mu_v \quad (2)$$

Où $n\mu_v$ est une unité de normalisation pour normaliser les valeurs de diff dans l'intervalle $[0, 1]$.

L'algorithme choisit aléatoirement une instance $I \in S$ et trouve ses deux voisins les plus proches. Le premier est l'instance la plus proche de la même classe que I (near hit) et le deuxième est l'instance la plus proche d'une classe différente (near miss). L'idée sous jacente est que l'attribut est le plus pertinent dans I s'il est le plus capable de séparer I de son near miss, et le moins capable de la séparer de son near hit. Le résultat est une version pondérée de l'ensemble d'attributs d'origine. L'algorithme pour un problème à deux classes est décrit dans l'Algorithme III.4.

Entrées :

p : pourcentage d'échantillonnage

D : mesure de distance

$S(X)$: un échantillon S est décrit par X

$|X| = n, m := p|S|$

Initialiser le tableau $w []$ à zéro

Faire m fois

I : = Instance aléatoire (S)

INH : = Near-Hit (I, S)

Inm : = Near-Miss (I, S)

Pour chaque $i[1..n]$ Faire

$W[i] := w[i] + d_i(I, Inm)/m - d_i(I, Inh)/m$

Fin pour**Fin**

Sortie : w – tableau des poids d'attributs.

Algorithme III. 4. *Algorithme Relief [58].*

III.6. Méthodes Aléatoires

Ces méthodes n'ont pas une manière spécifique pour générer le sous-ensemble d'attributs à analyser, mais utilisent des méthodes aléatoires. Ainsi, une recherche probabiliste est effectuée dans l'espace des caractéristiques. Les résultats obtenus en utilisant ces types de méthodes dépendent du nombre des itérations, sans garantir que le sous-ensemble optimal soit atteint. Les méthodes LVF et les algorithmes génétiques appartiennent à cette catégorie.

a) Las Vegas Filter et Las Vegas Wrapper

LVF (Las Vegas Filter) est un algorithme qui génère plusieurs fois des sous-ensembles aléatoires et calcule la consistance de l'échantillon:

Une incohérence dans X' et S est définie par deux instances dans S , qui sont égales lorsque l'on considère seulement les attributs de X' et qui appartiennent à des classes différentes. Le but est de trouver le sous-ensemble minimal d'attributs conduisant à des incohérences nulles [60].

LVF est décrit dans l'algorithme 5, il est particulièrement efficace pour des ensembles de données ayant des attributs redondants [47]. Cette méthode peut trouver le sous-ensemble optimal même pour les ensembles de données bruitées si le niveau de bruit est spécifié au début [61].

On peut dire que le principal avantage de LVF est qu'il réduit rapidement le nombre d'attributs dans les premières étapes avec une certaine confiance [47]. Son inconvénient est qu'il peut prendre, dans certains problèmes, plus de temps pour trouver le sous-ensemble optimal, par rapport aux algorithmes heuristiques, car il ne peut pas profiter des connaissances antérieures [52].

Entrées :

max : le nombre maximum d'itérations,
 J : mesure d'évaluation
 S(X) : un échantillon S décrite par X , $|X| = n$
 L : = [] // L enregistre les meilleurs ensembles
 Best := X // Initialiser la meilleure solution
 J0 := J(S(X)) // valeur minimum autorisée de J

Répéter max fois

X' := sous ensemble aléatoire (Best)

Si $J(S(X')) \geq J0$ **Faire**

Si $|X'| < |Best|$ **Faire**

Best := X'

L := [X'] // L est réinitialisée

Sinon

Si $|X'| = |Best|$ **Faire**

L := ajouter (L, X')

Fin si

Fin si

Fin si

Fin

Sortie :

L : toutes les solutions équivalentes trouvées

Algorithme III. 5. *Algorithme LVF [53].*

b) Les algorithmes génétiques pour la sélection d'attributs

Les algorithmes génétiques sont des techniques de recherche adaptative basées sur les principes de la sélection naturelle en biologie. Ils sont considérés parmi les méthodes de sélection classiques des attributs et sont largement appliqués dans ce domaine [53] [54].

Pour la sélection des attributs, une solution est généralement codée sous forme d'une chaîne binaire de longueur fixe qui représente un sous-ensemble des attributs. La valeur de chaque position dans la chaîne représente la présence ou l'absence d'un attribut particulière.

L'algorithme de sélection est un processus itératif où chaque génération est produite par l'application d'opérateurs génétiques, tels que la mutation et le croisement sur les membres de la génération actuelle. La mutation modifie certaines valeurs (ajout ou suppression d'attributs) dans un échantillon aléatoire. Le croisement combine les différents attributs d'une paire de sous-ensembles dans un nouveau sous-ensemble.

L'application des opérateurs génétiques sur une population d'individus est déterminée par leur fitness (quel sous-ensemble des attributs est le meilleur par rapport aux autres en utilisant une stratégie d'évaluation). Le meilleur attribut d'un sous ensemble a plus de chances d'être sélectionnée pour former un nouveau sous ensemble par le biais du croisement ou de la mutation [50].

III.7. Méthodes de sélection d'attributs basées sur l'intelligence en essaims

Parmi les nombreuses méthodes qui sont proposées pour la sélection d'attributs, les algorithmes d'optimisation basés sur la notion de population tels que la méthode d'optimisation par les essaims particulaires (PSO) ou les chauves souris (BA) ont attiré beaucoup d'attention. Cette section présente les principales approches de l'intelligence en essaims qui sont appliquées dans le domaine de la sélection des attributs.

III.7.1. L'algorithme des essais de lucioles (Firefly algorithm)

L'algorithme d'optimisation basé sur les essais de lucioles (insectes lumineux) ou Firefly algorithmes (FA ou FFA) développé par Xin-She Yang à l'Université de Cambridge en 2007 [51], utilise les trois règles idéalisées suivantes:

1. Toutes les lucioles sont unisexes, cela signifie qu'une luciole est attirée par une autre, indépendamment de son sexe.
2. L'attractivité et la luminosité sont proportionnelles entre elles, de sorte que, pour deux lucioles clignotantes, la moins lumineuse se déplacera vers la plus lumineuse. L'attractivité et la luminosité diminuent à l'augmentation de leur distance. S'il n'y a pas une luciole plus brillante que l'autre, elles vont se placer de manière aléatoire.
3. La luminosité d'une luciole est déterminée par le point de vue de la fonction objective.
Pour un problème de maximisation, la luminosité est simplement proportionnelle à la valeur de la fonction objective.

D'autres formes de la luminosité peuvent être définies d'une manière identique à la fonction de fitness dans les algorithmes génétiques.

III.7.2. L'algorithme d'optimisation par les colonies d'abeilles (BCO)

L'algorithme Bee Colony Optimization ou BCO est un algorithme basé sur la population [52]. Une population d'abeilles artificielles cherche la solution optimale.

Les abeilles artificielles représentent des agents en collaboration, qui résolvent des problèmes complexes d'optimisation combinatoire. Chaque abeille artificielle génère une solution au problème. L'algorithme se compose de deux phases alternées: passage forward et passage backward. Dans chaque passage forward, chaque abeille artificielle explore l'espace de recherche.

Elle s'applique à un nombre prédéfini de mouvements, qui construisent et/ou améliorent la solution, créant ainsi une nouvelle solution.

Après avoir obtenu de nouvelles solutions partielles, les abeilles vont de nouveau au nid et commencent la deuxième phase :

Le passage dit backward. Dans le passage backward, toutes les abeilles artificielles partagent des informations sur leurs solutions [53].

Initialisation

Pour toutes les I itérations **Faire**

Pour tous les m stages **Faire**

Pour toutes les B abeilles **Faire**

Passage forward: choisir une solution partielle

Fin Pour

Pour toutes les B abeilles **Faire**

passage backward: l'échange d'informations sur les solutions partielles avec les abeilles dans le nid

Fin Pour

Fin Pour

Si la meilleure solution obtenue dans l'itération est la meilleure globale,

Mettre à jour la solution la plus connue.

Fin Pour.

Algorithme III. 6. *Algorithme BCO.*

III.7.3. L'algorithme d'optimisation de l'exploration bactérienne (BFO)

L'algorithme d'optimisation basé sur l'exploration bactérienne (Bacterial Foraging Optimization ou BFO) proposé en 2002 par Passino [54] est basé sur le comportement alimentaire des bactéries d'*Escherichia coli* (*E. coli*) présentes dans l'intestin humain [56]. La théorie de l'exploration pour la recherche de nourriture est basée sur l'hypothèse que les bactéries obtiennent les constituants énergétiques maximaux dans un temps supposé être petit. Le BFO de base se compose de trois

principaux mécanismes : la chimiotaxie, la reproduction et l'élimination-dispersion [54] [55].

III.8. Conclusion

La sélection d'attributs est un domaine de recherche très actif, incluant un grand nombre d'algorithmes appliqués dans plusieurs domaines. Ce chapitre présente le problème de la sélection d'attributs, ainsi nous avons décrit le processus de sélection des attributs avec ces étapes. Ensuite une brève description pour les méthodes de la sélection de d'attributs en se concentrant, vers sa fin, sur les méthodes de l'intelligence en essaims.

Chapitre IV

Une approche BBA pour la sélection des attributs

IV.1. Introduction

Une fois une recherche exhaustive de l'ensemble optimal d'attributs dans un espace de grande dimension peut être impraticable, plusieurs méta-heuristique tentent de modéliser la sélection des attributs comme un problème d'optimisation NP-difficile, ainsi des hybridations entre les métas heuristiques sont proposées dans la littérature pour résoudre le problème de sélection d'attributs dans la classification de données

Plusieurs algorithmes heuristiques dérivés du comportement des systèmes biologiques et / ou physiques dans la nature ont été proposé comme des méthodes puissantes pour des optimisations globales.

L'auteur de [62] a proposé une nouvelle méta-heuristique méthode pour l'optimisation continue, c'est à savoir l'algorithme de chauve-souris (BA), qui est basé sur la capacité fascinante de la chauve-souris à trouver leur proie et discriminer les différents types d'insectes même dans l'obscurité complète. Une telle approche a démontré pour surpasser certains biens connus d'inspiration naturelle techniques d'optimisation.

Dans ce chapitre nous détaillons une forte variante de l'algorithme de chauve-souris appelée algorithme de chauve-souris binaire (BBA) pour construire un modèle pour la sélection des attributs dans la classification des données et nous présentons les résultats de tests.

IV.2. Conception de l'approche BBA proposée

Binary bats algorithme est conceptuellement similaire à l'algorithme général de Bat, la différence réside dans l'espace de recherche. En général Bat algorithme s'exécute dans l'espace de recherche continu, alors que BBA s'exécute dans l'espace binaire.

IV.2.1. Codage des particules

Pour chaque chauve-souris une position X_i , elle représenté par un ensemble binaires coordonnées qui indiquent si un attribut appartiendra à l'ensemble final des attributs ou non (pertinent ou non pertinent), les bits vaut 1 aux attributs sélectionnés et les bits vaut 0 aux attributs non sélectionnés.

L'essaim c'est l'ensemble de particules.

Exemple

Soit la position de particule « i » représentée comme suit :

$$X_i = [1, 0, 1, 1, 0, 0, 1, 1, 0]$$

Les éléments correspondants aux positions 1, 3, 4, 7, 8 sont les attributs sélectionnés, alors que 2, 5, 6, 9 non sélectionnés.

La taille du vecteur correspond au nombre d'attributs dans la base de données

IV.2.2. Évaluation des particules

Comme la qualité de la solution est liée au nombre de chauves-souris, nous devons évaluer chacun d'entre eux en formant un classificateur avec les attributs sélectionnés codés par la position de la chauve-souris et également pour classer un ensemble d'évaluation.

Ainsi, nous avons besoin d'un jeûne et classificateur robuste, puisque nous avons une instance pour chaque batte. En tant que tel, nous avons choisi d'utiliser la forêt Optimum-Path (OPF) classificateur disponible dans « *weka* » sous le nom (KNN).

IV.2.3. Modélisation du problème de sélection des attributs par BAA

Afin de modéliser cette approche, les auteurs de [62] ont idéalisé certaines règles, comme suit:

- Toutes les chauves-souris utilisent l'écholocation pour détecter la distance, et elles aussi "connaître" la différence entre nourriture / proie et barrière de fond d'une manière magique.
- A Une chauve-souris b_i vole au hasard avec la vitesse V_i à la position X_i avec une fréquence fixe Q_{min} , longueur d'onde variable λ et loudness A_0 pour rechercher une proie. Ils peuvent automatiquement ajuster la longueur d'onde (ou la fréquence) de leur émission impulsions et ajuster le taux d'émission d'impulsion $r \in [0, 1]$, en fonction de la proximité de leur cible.
- Bien que le volume puisse varier de plusieurs façons, les auteurs de [62] supposons que le volume varie d'un grand (positif) A_0 à une valeur constante A_{min} .

Le problème de sélection des attributs fait partie des problèmes d'optimisation discret / binaire, il est représenté dans l'algorithme BAA comme suit :

- L'idée principale est de générer un swarm de N particules, chaque chauve souris i est représentée par sa position X_i , sa vitesse V_i et fréquence Q_i .
- Une fonction d'évaluation (fonction fitness) est calculée pour chaque chauve souris.
- Mettre à jour la meilleure solution G_{best} pour toute la population des chauves souris.
- Calculer loudness A , pulse rate r et la fonction de fitness pour chaque particule.
- Mettre à jour la fréquence, vitesse et la position pour chaque particule.
- Le processus de recherche continue à chercher la meilleure solution selon la fonction fitness, jusqu'à ce que le critère d'arrêt soit vérifié, si ce dernier est vérifié, on retient la meilleure solution.
- Dans chaque itération, la même procédure sera répétée.
- Lorsque le critère d'arrêt est vérifié, la meilleure solution (G_{best}) est retournée.

La Figure IV.1. présente le fonctionnement général de l'algorithme BAA :

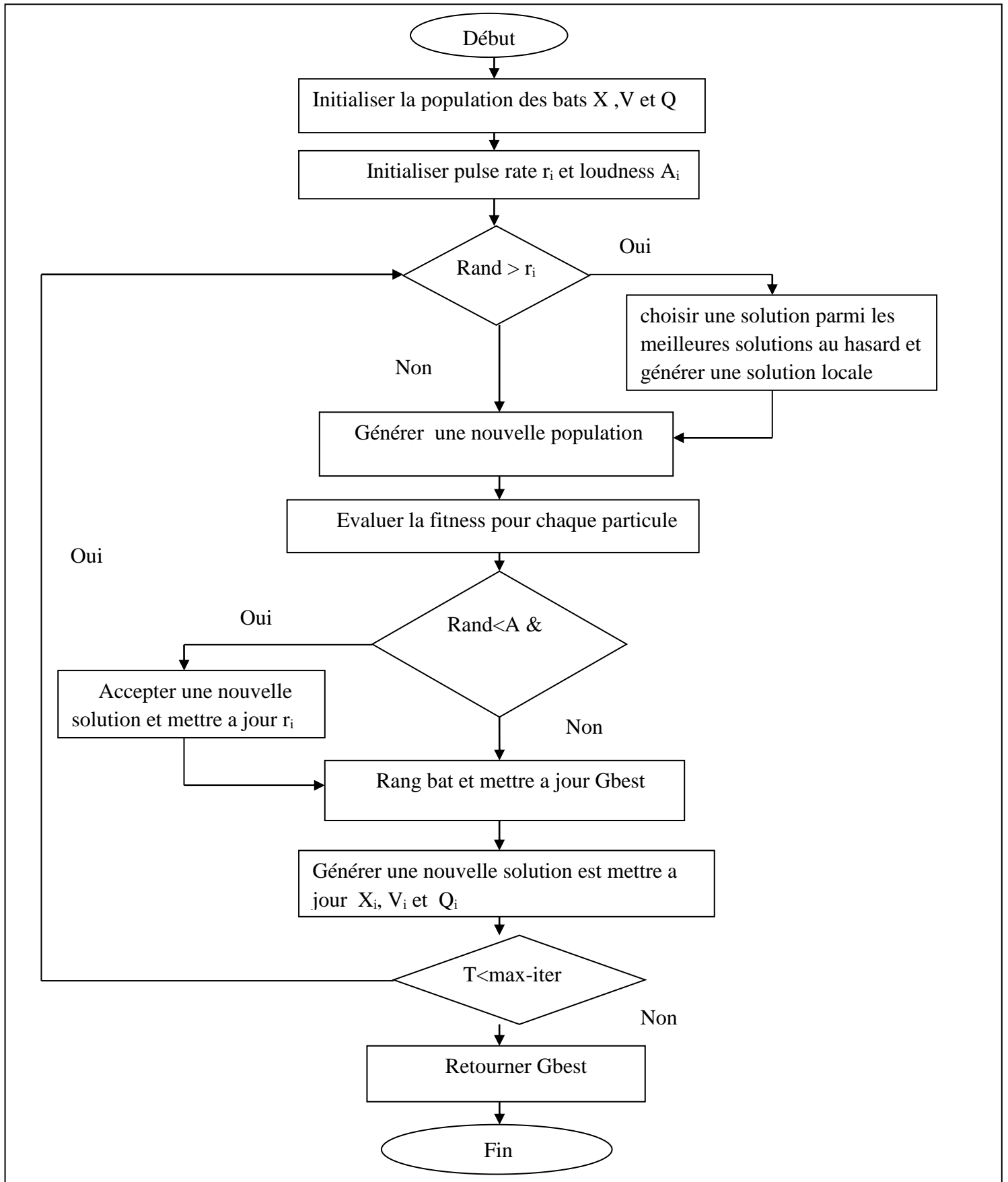


Figure IV.1. Schéma fonctionnel de l'algorithme

IV.2. L 'Algorithme BBA

L'algorithme de BBA s'exécute par étapes comme suit :

Etape 1 : Initialiser la population initiale :

La position initiale X_i , la vitesse V_i et la fréquence Q_i sont initialisés pour chaque chauve-souris b_i .

Etape 2 : Evaluation des particules

Evaluer la valeur de la fonction de fitness pour toutes les chauves souris .

Etape 3 : Trouvé le meilleur actuelle (best)

$$[f_{\min}, I] = \min(\text{Fitness}) \quad (1)$$

F_{\min} : le résultat retourné par la fonction de fitness, avec I est l'emplacement de F_{\min}

Etape 4: Mettre a jour la vitesse et la fréquence

Le mouvement des chauves souris données en mettant à jour leur vitesse et leur position en utilisant les équations 2, 3,5 comme suite :

$$V_i(t+1) = V_i(t) + (X_i(t) - G_{\text{best}}) F_i \quad (2)$$

G_{best} : représente le meilleur obtenue « so far »

$$X_i^{h_i(t+1)} = \begin{cases} (x_i^h(t) - 1) \text{rand} \leq f(v_i^h(t+1)) \\ x_i^h(t) \text{rand} > f(v_i^h(t+1)) \end{cases} \quad (3)$$

Rand : un nombre généré aléatoirement dans l'intervalle $[0,1]$.

x_i^h : la valeur de la variable h de décision pour la chauve souris i au pas de temps « t ».

x_i^h : la position, v_i^h : la vitesse pour la chauve souris « i » dans l'itération t de h ieme dimension $(x_i^h(t) - 1)$: représente le complément de $x_i^h(t)$

Où :

$$F(v_i^h(t)) = \left| \frac{2}{\pi} \arctan\left(\frac{\pi}{2} v_i^h(t)\right) \right| \quad (4)$$

Et :

$$F_i = F_i + (F_{\min} - F_{\max}) * \text{rand} \quad (5)$$

- Le résultat de l'équation (1) est utilisé pour contrôler le rythme et la portée du mouvement de la chauve souris.
- Ces équations dans une certaine mesure peuvent garantir la capacité d'exploitation de la BBA

Cependant pour mieux effectuer l'augmentation, une opération de marche aléatoire a également été utilisée comme suit :

$$X_{\text{new}} = X_{\text{old}} \quad (6)$$

X_{old} : une solution sélectionnée aléatoirement parmi la meilleure solution actuelle.

- Pour chaque itération de l'algorithme de volume et le taux d'impulsion A_i et démission r_i sont mis à jour comme suit :

$$A_i (t+1) = \alpha A_i(t) \quad (7)$$

$$r_i (t+1) = r_i (0) (1 - e^{-\beta}) \quad (8)$$

Avec α et β sont des constantes.

- Calculer la fonction de transfert selon l'équation (4)
- Evaluer une nouvelle solution:

If ($F_{\text{new}} \leq \text{fitness} (i)$) && ($\text{rand} < A$)

$\text{Sol} (i , :) = s (i , :)$;

$\text{Fitness} (i) = F_{\text{new}}$

End

Etape 5: mettre à jour le meilleur actuel

```

If  $F_{\text{new}} \leq f_{\text{min}}$ 
    Best = s ( i ; :)
     $F_{\text{min}} = F_{\text{new}}$ 
End

```

Etape 6 : Répéter l'étape 4 et 5 jusqu'à ce que le critère d'arrêt soit vérifié.

IV.3. Expérimentations

IV.3.1. Matériels et logiciels utilisés

Notre approche a été implémentée et exécutée en utilisant :

- Un ordinateur de 4GO de RAM et un processeur Intel core (TM) 2.3 GHz.
- MATLAB.
- Logiciel WEKA.

WEKA

(Waikato environment for knowledge analysis) en français « environnement Waikato pour l'analyse de connaissances ». Weka est un ensemble d'algorithmes d'apprentissage automatique pour les tâches d'exploration de données. Les algorithmes peuvent être appliqués directement à un ensemble de données ou appelés à partir de votre propre code Java. Weka contient des outils pour le pré-traitement des données, la classification, la régression, le clustering, les règles d'association et la visualisation. Il est également bien adapté au développement de nouveaux schémas d'apprentissage machine.

« Weka » est un logiciel open source publié sous licence publique générale GNU.

L'espace de travail « Weka » contient une collection d'outils de visualisation et d'algorithmes pour l'analyse des données et la modélisation prédictive allié à une interface graphique pour un accès facile de ses fonctionnalités.

Il comprend plusieurs versions, la plus récente entièrement basée sur Java (Weka 3), pour laquelle le développement a débuté en 1997, est désormais utilisée dans beaucoup de domaines d'application différents, en particulier pour l'éducation et la recherche.

Avantage de WEKA

- Il est libre et gratuit, distribué selon les termes de la licence publique générale GNU.
- Il est facile à utiliser par un novice en raison de l'interface graphique qu'il contient
- Il contient une collection complète de préprocesseurs de données et de techniques de modélisation.

IV.3.2. Jeux de données de test

Nous avons utilisé des données provenant de la base de données IRIS ; Le jeu de données Iris connu aussi sous le nom de Iris de Fisher, Les iris de Fisher" sont des données proposées en 1933 par le statisticien Ronald Aylmer Fisher comme données de référence pour l'analyse discriminante et la classification. Les données correspondent à 3 espèces de fleurs (Iris setosa, Iris virginica, Iris versicolor).

Quatre attributs ont été mesurés: la longueur et la largeur des sépales et des pétales en centimètres. Sur la base de la combinaison de ces quatre variables, Fisher a élaboré un modèle d'analyse discriminante linéaire permettant de distinguer les espèces les unes des autres.

Le détail de jeux de données de la base IRIS et données dans le tableau suivant :

Nb espèces	Nb d'échantillons	Nb d'attributs	Description
3	50	4	<p>Une espèce (Iris setosa, Iris virginica ou Iris versicolor) correspond à 50 échantillons</p> <p>Les quatre attributs (la longueur et la largeur des sépales et des pétales) sont mesurés par chaque échantillon</p>

Tableau IV. 1. Description de la base de données IRIS.

IV.3.3. Paramètres

Nous avons choisi pour nos jeux de test de notre approche Les paramètres suivants :

- le nombre des itérations T
- le nombre de particules nop.
- le nombre de variable nov.
- le taux d'impulsion A
- le taux d'émission r.

IV.3.4. Réglage des paramètres

Dans cette approche on s'intéresse à la taille de population, le nombre d'itérations et le nombre d'exécutions qui représente le nombre de fois que l'algorithme est exécuté.

Nous présentons ci-dessous les différents paramètres appliqués aux jeux de données :

- T= 500.
- Nop=50.
- Nov=90.
- A=25.
- r=1.

IV.3.5. Résultats expérimentaux

✚ Etude du nombre des itérations sur la meilleur moyen G_{best} .

Test :

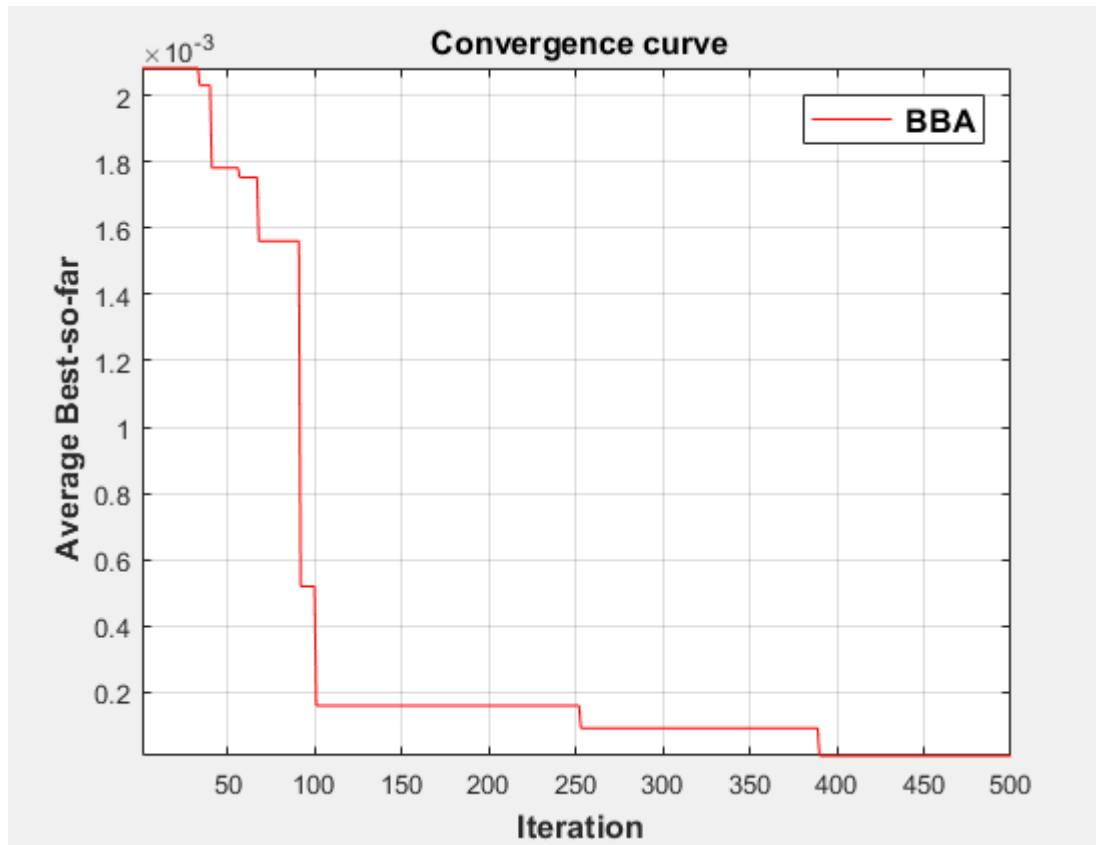


Figure IV. 5. Convergence le meilleur moyen G_{best}

IV.5. Conclusion

Dans ce chapitre nous avons utilisé la nouvelle approche binaire du BA pour la sélection d'attributs dans la classification des données, ensuite la comparer avec d'autres approches de la littérature telles que l'optimisation par essaim de particules (PSO), Harmony Recherche, Algorithme de Recherche Gravitationnelle et Algorithme de Firefly.

Les résultats de comparaison montrent l'efficacité de l'approche BBA par rapports aux autres approches, elle trouve des bons résultats dans un temps réduit (elle converge vers la solution optimale rapidement), on a remarqué une réduction importante de la taille de l'ensemble des attributs sélectionnés qui donne généralement des taux de classification meilleure que l'utilisation de l'ensemble totale des attributs des bases de tests.

Conclusion Générale

1. Conclusion et perspectives

Les paramètres les plus importants qui peuvent agir sur la performance de la tâche de la classification supervisée de données sont le nombre et la qualité des attributs extraits à partir des entités.

La sélection des attributs consiste à choisir parmi un grand ensemble d'attributs un sous-ensemble pertinent pour un problème étudié, elle réduit le nombre des attributs non pertinents, redondants ou bruités. En effet, la sélection d'attributs fait partie des problèmes d'optimisation NP-difficile qui peut être résolu par les méta-heuristiques.

Dans notre travail nous avons proposé une nouvelle approche BBA pour la résolution du problème de sélection d'attributs basée sur la méta-heuristique proposée dans la littérature BA.

Le but de notre approche est d'améliorer la performance de la méthode BA pour avoir une autre très variante afin de mieux explorer l'espace de recherche. Les expérimentations sont réalisées sur une base de données IRIS.

2. Comme perspectives

Il est certain que la solution proposée n'est pas la meilleure d'une manière absolue, plusieurs améliorations sont envisageables et plusieurs points peuvent être étudiés dans le cadre de travaux futurs, tel que :

- ✚ Concevoir et développer une plateforme de résolution de problèmes d'optimisation de différentes complexités dans laquelle nous intégrons notre méthode et d'autres méthodes existantes pour fournir un moyen permettant de faciliter la mise en œuvre de différentes méta-heuristiques pour la résolution des problèmes d'optimisation et en particulier la sélection d'attribut

Annexes

✚ La classe BBA

```
function [best,fmin,cg_curve]=BBA(n, A, r, d, Max_iter, CostFunction)

addpath([pwd filesep 'matlab2weka']);
javaaddpath('D:\matlab_ws\Weka-3-6\weka.jar');
javaaddpath([pwd filesep 'matlab2weka' filesep 'matlab2weka.jar']);
Qmin=0;
Qmax=2;
N_iter=0;
Q=zeros(n,1);
v=zeros(n,d);
Sol=zeros(n,d);
cg_curve=zeros(1,Max_iter);
for i=1:n
    for j=1:d
        if rand<=0.5
            Sol(i,j)=0;
        else
            Sol(i,j)=1;
        end
    end
end
for i=1:n
    Fitness(i)=CostFunction(Sol(i,:));
end
[fmin,I]=min(Fitness);
best=Sol(I,:);
```

```

while (N_iter<Max_iter)
    % Loop over all bats/solutions
    N_iter=N_iter+1;
    cg_curve(N_iter)=fmin;
    for i=1:n
        for j=1:d
            Q(i)= Omin+(Omin-Omax)*rand;
            v(i,j) = v(i,j) + (Sol(i,j) - best(j) ) * Q(i);
            V_shaped_transfer_function = abs((2/pi)*atan((pi/2)* v(i,j)));

            if rand<V_shaped_transfer_function
                Sol(i,j)=~Sol(i,j);
            else
                Sol(i,j)=Sol(i,j);
            end

            if rand>r
                Sol(i,j)=best(j);
            end
        end

        Fnew = CostFunction(Sol(i,:));

        if (Fnew<=Fitness(i)) && (rand<A)
            Sol(i,:)=Sol(i,:);
            Fitness(i)=Fnew;
        end

        if Fnew<=fmin
            best=Sol(i,:);
            fmin=Fnew;
        end
    end

    disp(['Number of evaluations: ',num2str(N_iter)]);
    disp([' fmin=',num2str(fmin)]);
end

```

✚ La classe main

```
clear all;
close all;
clc;

CostFunction=@(x) MyCost(x);

Max_iteration=30;
noP=30;
noV=4;
A=.25;
r=.1;

%BPSO with s-shaped family of transfer functions
[gBest, gBestScore ,ConvergenceCurve]=BBA(noP, A, r, noV, Max_iteration, CostFunction);

plot(ConvergenceCurve,'DisplayName','BBA','Color','r');
hold on
title('\fontsize{12}\bf Convergence curve');
xlabel('\fontsize{12}\bf Iteration');ylabel('\fontsize{12}\bf Average Best-so-far');
legend('\fontsize{12}\bf BBA');
grid on
axis tight
save results
```

La classe Matlab2weka

```
function wekaOBJ = matlab2weka(name, featureNames, data, targetIndex)
    if(~wekaPathCheck),wekaOBJ = []; return,end
    if(nargin < 4)
        targetIndex = numel(featureNames);

    end

    import weka.core.*;
    vec = FastVector();
    if(iscell(data))
        for i=1:numel(featureNames)
            if(ischar(data{1,i}))
                attvals = unique(data(:,i));
                values = FastVector();
                for j=1:numel(attvals)
                    values.addElement(attvals{j});
                end
                vec.addElement(Attribute(featureNames{i},values));
            else
                vec.addElement(Attribute(featureNames{i}));
            end
        end
    else
        for i=1:numel(featureNames)
            vec.addElement(Attribute(featureNames{i}));
        end
    end
    wekaOBJ = Instances(name,vec,size(data,1));
    if(iscell(data))
        for i=1:size(data,1)
            inst = Instance(numel(featureNames));
            for j=0:numel(featureNames)-1
                inst.setDataset(wekaOBJ);
                inst.setValue(j,data{i,j+1});
            end
            wekaOBJ.add(inst);
        end
    else
        for i=1:size(data,1)
            wekaOBJ.add(Instance(1,data(i,:)));
        end
    end
    wekaOBJ.setClassIndex(targetIndex-1);
end
```

La classe MyCost

```
function [o] = MyCost(x)

load fisheriris;
rand('twister',0);
perm = randperm(150);
meas = meas(perm,:);
species = species(perm,:);

featureNames1 = {'sepalength','sepalwidth','petallength','petalwidth','class'};
data = [num2cell(meas),species];
train1 = data(1:120,:);
test1 = data(121:end,:);

v = (x(1, 1) * 1000) + (x(1, 2) * 100) + (x(1, 3) * 10) + x(1, 4);
cols = 0;
if v == 1
    featureNames = {'petalwidth','class'};
    train1 = data(1:120, 4);
    test1 = data(121:end,4);
end
if v == 10
    featureNames = {'petallength','class'};
    train1 = data(1:120, 3);
    test1 = data(121:end,3);
end
if v == 11
    featureNames = {'petallength','petalwidth','class'};
    train = data(1:120, [3 4]);
    test = data(121:end,[3 4]);
end
if v == 100
    featureNames = {'sepalwidth','class'};
    train1 = data(1:120, 1);
    test1 = data(121:end,1);
end
if v == 101
    featureNames = {'sepalwidth','petalwidth','class'};
    train1 = data(1:120, [1 3]);
    test1 = data(121:end,[1 3]);
end
if v == 111
    featureNames = {'sepalwidth','petallength','petalwidth','class'};
    train1 = data(1:120, [2 3 4]);
    test1 = data(121:end,[2 3 4]);
end
if v == 1000
    featureNames = {'sepalength','class'};
    train1 = data(1:120, 1);
    test1 = data(121:end,1);
end
if v == 1001
    featureNames = {'sepalength','petalwidth','class'};
    train1 = data(1:120, [1 4]);
    test1 = data(121:end,[1 4]);
end
if v == 1010
    featureNames = {'sepalength','petallength','class'};
    train1 = data(1:120, [1 3]);
    test1 = data(121:end,[1 3]);
end
end
```

```

classindex = 5;
train1 = matlab2weka('iris-train',featureNames1,train1,classindex);
test1 = matlab2weka('iris-test',featureNames1,test1);
nb = trainWekaClassifier(train1,'bayes.NaiveBayes');
predicted = wekaClassify(test1,nb);
actual = test1.attributeToDoubleArray(classindex-1);
errorRate = sum(actual ~= predicted)/30;
o = errorRate;
-end

```

✚ La classe train wekaClassifier

```

function wekaClassifier = trainWekaClassifier(wekaData,type,options)
    if(~wekaPathCheck),wekaClassifier = []; return,end
    wekaClassifier = javaObject(['weka.classifiers.',type]);
    if(nargin == 3 && ~isempty(options))
        wekaClassifier.setOptions(options);
    end
    wekaClassifier.buildClassifier(wekaData);
end

```

✚ La classe weka2matlab

```
function [mdata, featureNames, targetNDX, stringVals, relationName] =...
                                                weka2matlab(wekaOBJ, mode)
if(~wekaPathCheck), mdata = []; return, end
if(nargin < 2)
    mode = [];
end

if(not(isjava(wekaOBJ)))
    fprintf('Requires a java weka object as input. ');
    return;
end

mdata = zeros(wekaOBJ.numInstances, wekaOBJ.numAttributes);
for i=0:wekaOBJ.numInstances-1
    mdata(i+1,:) = (wekaOBJ.instance(i).toDoubleArray)';
end

targetNDX = wekaOBJ.classIndex + 1;

featureNames = cell(1, wekaOBJ.numAttributes);
stringVals = cell(1, wekaOBJ.numAttributes);
for i=0:wekaOBJ.numAttributes-1
    featureNames{1, i+1} = char(wekaOBJ.attribute(i).name);

    attribute = wekaOBJ.attribute(i);
    nvals = attribute.numValues;
    vals = cell(nvals, 1);
    for j=0:nvals-1
        vals{j+1, 1} = char(attribute.value(j));
    end
    stringVals{1, i+1} = vals;
end
end
```

```

end

relationName = char(wekaOBJ.relationName);

if(iscell(mode))
    celldata = num2cell(mdata);
    for i=1:numel(stringVals)
        vals = stringVals{1,i};
        if(not(isempty(vals)))
            celldata(:,i) = vals(mdata(:,i)+1)';
        end
    end
    mdata = celldata;
end
end
end

```

✚ La classe wekaClassify

```

function [predictedClass, classProbs] = wekaClassify(testData,classifier)
    if(~wekaPathCheck),classProbs = []; return,end
    for t=0:testData.numInstances -1
        classProbs(t+1,:) = (classifier.distributionForInstance(testData.instance(t)))';
    end
    [prob,predictedClass] = max(classProbs,[],2);
    predictedClass = predictedClass - 1;
end

```

✚ La classe wekaNBexample

```
load fisheriris;
rand('twister',0);
perm = randperm(150);
meas = meas(perm,:);
species = species(perm,:);

featureNames = {'sepalength','sepalwidth','petalength','petalength', 'class'};
data = [num2cell(meas),species];

train = data(1:120 , [1 1 3 4 5]);
test = data(121:end, [1 1 3 4 5]);

classindex = 5;
train = matlab2weka('iris-train',featureNames,train,classindex);
test = matlab2weka('iris-test',featureNames,test);

%nb = trainWekaClassifier(train,'bayes.NaiveBayes');
%predicted = wekaClassify(test,nb);
%actual = test.attributeToDoubleArray(classindex-1);
%errorRate = sum(actual ~= predicted)/30
```

✚ La classe wekaPathCheck

```
function bool = wekaPathCheck()
    bool = true;
    w = strfind(javaclasspath('-all'),'weka.jar');
    if isempty([w{:}])
        bool = false;
        fprintf('\nPlease add weka.jar to the matlab java class path.\n');
        help wekaPathCheck;
    end
end
```

Bibliographies

- [1] Chami, Jazia (2009). « Une plate forme orientée agent pour le data » .
Mémoire magister : *Sciences et Technologies de l'Information et de la Communication* (STIC).Batna : Université HADJ LAKHDAR, 108 p.
- [2] Javier, Bravo(2017). « Une Science ouverte dans une République numérique » : *Guide stratégique*, Marseille : Open Edition Press. 23p.
- [3] Marius, Fieschi (2006). “Intelligent Database Systems Research Lab School of Computing Science”: Data mining. Canada .28p.
- [4] Kergosien,E(2013)« fouilledonnéeFondements ».Repéré«http://www.ekergosien.net/images/CoursECD_Fondements_Kergosien.pdf ».
- [5] Ben Abde Allah, W. (2017). « Ordonnancement des conteneurs ». Ingénieur en Informatique : *Système et réseaux soucieux*. Tunis : Université de Tunis, 70 p.
- [6] Talbi, E-G. « Fouille de données ». Repéré à « <http://www.lifl.fr/~talbi/Cours-Data-Mining.pdf>».
- [7] Bastien, L. (2018). “Data mining” .Paris. «<https://www.lebigdata.fr/data-mining-definition-exemples>».
- [8] Markus, H. (2004).” Data mining technique” .313 -355 p.
- [9] Stéphane, T. (2014). “Data mining”. Repéré à «<http://blogperso.univ-rennes1.fr/stephane.tuffery/public/Tuffery-MasterRennes2013-2014-DataMining-Presentation.pdf>».
- [10] Maria ,M. « La fouille de données ». Repéré à «http://mma.perso.eisti.fr/HTML_DM/Cours/Cours1/datamining.pdf».
- [11] H.Liu et L.Yu. « Toward integrating feature selection algorithms for classification and clustering ». In : *IEEE Transaction On Knowledge* 17 (2005), p. 491–502.
- [12] David, H, . Heikki , M , . Padhraic , S(2001). “Principles of Data Mining”.
The Mit Press :546 p.
- [13] François , D., Rémi , G (2000) . « Apprentissage a partir d'exemple » .Repéré à «<http://www.grappa.univ-lille3.fr/polys/apprentissage/index.html>».

- [14] Alouane , B (2007) . « Recherche de partitions floues optimales par segmentation floue pour la fouille de données quantitatives » . Mémoire de Magister : *Système Informatique et Génies des logiciels* . Boumerdes : M'hamed BOUGARA , 202p.
- [15] BenAmina , M . (2007). « Extraction automatique des connaissance a partir des données » .Mémoire de Magister : *Informatique et Automatique* . Oran : Université d'Oran ,
- [16] Bernard , E. (2008). « Introduction à la Fouille de Données » : Data Mining . Marseille . 1-8 p.
- [17] Usama , F., Gregory , PS., Padhraic , S (1996). «From Data Mining to Knowledge Discovery in Databases» AI Magazine , 17(3) , 38- 54.
- [18] René, L., Gilles .V (1998). « Le data mining, Eyrolles » . 344p.
- [19] Engelbert, M . , Njiwoua , p.(2004) . « Treillis de concepts et classification supervisée ». Lens IUT de Lens – Université d'Artois Rue de l'université SP 16, 62307 Lens cedex , 1-41p.
- [20] Ben Abde Allah , W .(2017) . « Ordonnancement des conteneurs » . Ingénieur en Informatique : *Système et réseaux soucieux* . Tunis : Université de Tunis , 70 p.
- [21] « Classification supervisée Aperçu de quelques méthodes avec le logiciel R » .Repéré à «<http://www.math.univ-angers.fr/~labatte/enseignement%20UFR/master%20MIM/classificationsupervisee.pdf>»
- [22] Nello , .C , Bernhard,s (2002) . “The New Generation of Learning Machines” ,23(2).1-12p.
- [23] Mouhamadou , LS ., Fodé , C., Samba , N., Yahya , S., Mohamed, A (2014). « Approche de sélection d'attributs pour la classification basée sur l'algorithme RFE-SVM ». ARIMA Journal, 17 , 197-219 p.
- [24] EL FERCHICHI, Sabra (2013). « Sélection et extraction d'attributs pour les problèmes de classification ». Thèse de Doctorat : *Automatique , Génie Informatique , Traitement de signal et Images*. Tunis : Université Lille1 des Sciences et Technologies Université de Tunis EL MANR Ecole National de l'Ingénieurs de Tunis, p136.
- [25] CAMPEDEL, Marine., MOULINES, Eric. (2004). « Méthodologie de sélection de caractéristiques pour la classification d'images satellitaires ». Mémoire de Master : *Traitement du Signal et des Images*. Paris : Ecole Nationale Supérieure des Télécommunications, p16.

- [26] Semani , S . , Frélicot , C . , Courtellemont , P. « Une approche de type filtrage pour la sélection de variables ». Application `a la reconnaissance automatique de poissons . 1-4 .
- [27] I.Inza et al. « Filter versus wrapper gene selection approaches in DNA microarray domains ».In : *Artificial Intelligence in Medicine* 31 (2004), p. 91–103.
- [28] J.Robinson, S.Sinton et Y.R.Samii. « Particle swarm, genetic algorithm, and their hybrids :optimization of a profiled corrugated horn antenna ». In : *Proceedings of the IEEE InternationalSymposium in Antennas and Propagation Society* (2002), p. 314–317
- [29] Yousr , N . , Kaabi , M (2015). « La Sélection des variables » [cour] . Repéré à « <http://tct.org.tn/download/libre/1.pdf> ».
- [30] CHOUAIB , Hassan (2011) . « Sélection de caractéristiques: méthodes et applications ». Thèse : *Informatique . Paris : Université Paris Descartes* , p138.
- [31] KALAKECH , M (2011) . « Sélection semi-supervisée d’attributs ». Thèse : *Automatique et Informatique Industrielle . Lille : Université Lille1* , p185.
- [32] CL.Blake et CJ.Merz. « UCI repository of machine learning databases ». In : *CA : University of California ,Irvine, Department of Information and Computer Science* (1998), p. 203–232.
- [33] Dash , M., Liu , H(1997) . « Intelligent Data Analysis .Feature selection for classification » . vol (1) . p131 - 156 .
- [34] El Akadi , A (2012) . « Contribution a la sélection des variables pertinente en classification supervisée » . Thèse : Informatique et Télécommunications . Rabat – Maroc : Université Mohammed-V Agdal , p122.
- [35] Meshoul .S (2006) . « Approches bio-inspirées pour la reconnaissance de formes » . Magister : Informatique . constantine : universuté de Mountouri , p116.
- [36] Wei-Chang , Yeh ., Yuan-Ming , Yeh ., Cheng-Wei Chiu ., Yuk Ying Chung et Al (2014). “A Wrapper-Based Combined Recursive Orthogonal Array and Support Vector Machine for Classification and feature Selection” , vol . (8) , p11- 24.
- [37] I.Inza et al. « Filter versus wrapper gene selection approaches in DNA microarray domains ». In : *Artificial Intelligence in Medicine* 31 (2004), p. 91–103.
- [38] Almuallim H., Dietterich T.G(1991). “Learning with many irrelevant features”. In: *Proc. of The Ninth National Conference on Artificial Intelligence (AAAI- 91)*, Anaheim, California pp 547–552.
- [39] Liu H., Yu L(2005) . “Toward Integrating Feature Selection Algorithms for Classification and Clustering”, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No 4, pp 491–502.

- [40] Benahmed N 2002 (2002). « Optimisation de réseaux de neurones pour la reconnaissance de chiffres manuscrits isolés : Sélection et pondération des primitives par algorithmes génétiques », Mémoire de Maîtrise, Université du Québec, Canada.
- [41] Liu H., Yu L(2005). “Toward Integrating Feature Selection Algorithms for Classification and Clustering” , *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No 4, pp 491–502.
- [42] Almuallim H., Dietterich T.G (1991). “Learning with many irrelevant features”. In: *Proc. of The Ninth National Conference on Artificial Intelligence (AAAI- 91)*, Anaheim, California pp 547–552.
- [43] Liu H., Motoda H (1998). Dash M. “A Monotonic Measure for Optimal Feature Selection”. In *Proc. of the European Conference on Machine Learning*, Springer Verlag. pp 101–106.
- [44] Cakmakov D.(2002, Bennani, Y (2002). “Feature selection for pattern recognition”, Skopje, Informa.
- [45] Liu H., Motoda H., Dash M(1998). “A Monotonic Measure for Optimal Feature Selection”. In *Proc. of the European Conference on Machine Learning*, Springer Verlag. pp 109–115.
- [46] Belanche L.A., Gonzalez F.F(2011). “Review and Evaluation of Feature Selection Algorithms in Synthetic Problems” , Arxiv preprint arXiv:1101.2320.
- [47] Kira K., Rendell L(1992). “A practical approach to feature selection”, in: *Proc. Of the ninth international workshop on machine learning Morgan Kaufmann Publishers Inc.* San Francisco, CA, USA , pp.249-256 .
- [48] Grandidier F(2003). « Un nouvel algorithme de sélection de caractéristiques Application à la lecture automatique de l’écriture manuscrite », Thèse de doctorat, Université du Québec, Canada .
- [49] Ladha L., Deepa T(2011). “Feature Selection Methods And Algorithms” ,*International Journal on Computer Science and Engineering (IJCSE)*, Vol. 3 No. 5, pp 1787- 1797.
- [50] Dorigo M., Gambardella L.M (1997). « Ant Colony System: a cooperative learning approach to the traveling salesman problem”. *Evolutionary Computation*, IEEE Transactions on, Vol. 1, issue 1, pp 53-66.
- [51] Dash M. and Liu H(1997). “Feature Selection Methods for Classification” , *Intelligent Data Analysis*, Vol. 1, pp 131–156.

- [52] Zhang P, Verma B, Kumar K.(2003), “Neural vs. Statistical classifier in conjunction with genetic algorithm feature selection in digital mammography”. In Proc. *Congress on Evolutionary Computation (CEC-2003)*, Vol 2, pp 1206 – 1213, 8-12 .
- [53] L.C.Molina, L. Belanche et A. Nebot. « Evaluating feature selection algorithms ». In *CCIA-LNCS* (2002), p. 216–227.
- [54] Hall M.A (1999) . “Correlation-based Feature Selection for Machine Learning”. PhD Thesis University of Waikato, Australia.
- [55] Vanitha S., Dr. T(2014) . “Padma, A Survey on Swarm Intelligence Algorithms”. In *International Journal of Computer Science and Mobile Computing*, Vol. 3Issue.5, pp. 994-998.
- [56] Teodorovic D., Dell’Orco M (2005). “Bee colony optimization - a cooperative learning approach to complex transportation problems”. In: *Advanced ORand AI Methods in Transportation*. Proc. the 10th EWGT Meeting, pp. 51– 60. Poznan, Pologne, 13-16
- [57] Teodorovic D (2009). “Bee Colony Optimization (BCO)”, In: Chee P. L., Lakhmi C. J., et Satchidananda D. (Eds.) *Innovations in Swarm Intelligence, Studies in Computational Intelligence*, Vol. 248, pp. 39-60
- [58] Passino K.M.(2002), “Biomimicry of bacterial foraging for distributed optimization and control”. *IEEE Control System Magazine*, Vol. 22, Issue 3, pp 52–67.
- [59] Pattnaik S.S., Bakwad K.M., Devi S., Panigrahi B.K., Das S (2011). “Parallel Bacterial Foraging Optimization” In: Panigrahi B.K., Shi Y., et Lim M.-H. (Eds.): “*Handbook of Swarm Intelligence, Adaptation, Learning, and Optimization*”, Vol. 8, Springer-Verlag Berlin Heidelberg, pp. 487–502
- [60] MENGHOUR , K (2015) . « Une Approche Bio- Inspirées pour la Sélection des Attributs » . Thèse : Sciences et Technologies de l’Information et de la Communication . Annaba : UNIVERSITE BADJI MOKHTAR , p147 .
- [61] Dangauthier P, Spalanzani A et Bessière P(2004) . « Méthodes statistiques et algorithmes génétiques pour la sélection de connaissances préalable », Actes du congrès francophone de Reconnaissance des Formes et Intelligence Artificielle
- [62] X.-S. Yang., “Bat algorithm for multi-objective optimisation,” *International Journal of Bio-Inspired Computation*, vol. 3, no. 5, pp. 267–274, 2011.

