MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA

FACULTE DES SCIENCES ET DE LA TECHNOLOGIE

Département du Génie Électrique et de l'Informatique

Mémoire de fin d'étude

Présenté pour l'obtention du Diplôme de Master en Informatique

Option : Ingénierie des Systèmes et Technologies du Web

THEME

Méthode de Sélection d'Attributs pour la Classification basée sur la Métaheuristique des Lucioles

Proposé et dirigé par,

M. Sahmadi Brahim

Préparé et présenté par,

Melle. Nora SISSANI

Melle. Rekia RAHMANI

AnnéeUniversitaire2015-2016

Résumé

Le travail de recherche présenté dans ce mémoire se situe dans le domaine de datamining pour la classification et plus précisément, celui de la sélection d'attributs et porte sur la contribution à la création d'une nouvelle méthode de sélection afin de réduire la taille des données d'apprentissage en choisissant les attributs pertinents et non redondants avant la réalisation de l'apprentissage supervisé pour la classification.

L'objectif de la sélection d'attributs est d'améliorer la qualité de l'apprentissage supervisé du classifieur utilisé dans le but de créer des modèles de prédiction de haute performance permettant de prédire correctement la classe de nouveaux exemples.

Nous nous sommes intéressés en particulier à l'étude et la mise en œuvre de la nouvelle méthode de sélection d'attributs FA-SVM basé sur l'algorithme de luciole qui est une méta heuristique récente bio-inspiré issue de l'intelligence en essaim. En plus cette méthode utilise le classifieur de type SVM pour la classification.

MOTS-CLÉS : Datamining, algorithme de luciole, Classification, Apprentissage supervisé, Sélection d'attributs, Machines à Vecteurs de Supports (SVM)

Summary

The work presented in this document can be placed within the data mining for classification field and, more specifically, attribute or feature selection witch is one of the important and frequently used techniques in data pre-processing for data mining.

The main objective of this selection is to reduce the number of features by removing irrelevant, redundant, or noisy data, and improving mining performance such as predictive accuracy and result comprehensibility in order to create prediction model with height performance.

In our work, after a presentation of classical and bio-inspired attribute selection methods, we are interested in the bio-inspired methods in particular those of swarm intelligence for feature selection.

In this context, we propose FA-SVM as a new attribute selection method based on firefly algorithm witch is a new meta heuristic bio-inspired from swarm intelligence and this method use the Support Vector Machine as a classifier.

The experiments were performed and the results showed that, we can reduce the features number and improve the correct classification rate.

Keywords: Attribute or feature selection, bio-inspired attribute selection methods, swarm intelligence. Firefly algorithm data mining

ملخص

يندرج البحث العلمي المقدم في هذه المذكرة في مجال التنقيب عن البيانات، وبشكل أكثر تحديدا التصنيف الاوتوماتيكي. فمن اجل الحد من الحجم الكبير للبيانات يجب استعمال مرحلة اختيار الخصائص قبل مرحلة التصنيف والتي من خلالها يتم حذف الخصائص المكررة والغير المفيدة في عملية التصنيف واختيار أحسن مجموعة خصائص مميزة ذات أقل عدد ممكن والتي تمكننا من رفع نسبة التصنيف و تحسين نوعية المصنف التعليمي المستعمل من أجل خلق نماذج تنبؤيه عالية الدقة التي تسمح بالتنبؤ بالمجموعة الصحيحة التي تنتمي اليها الأمثلة جديدة.

مساهمتنا في هذا المجال تندرج في دراسة وتنفيذ طريقة جديدة لاختيار السمات FA-SVM تفيدنا في مرحلة اختيار الخصائص قبل مرحلة التصنيف و التي تعتمد على خوارزمية اليراعات المضيئة المستوحاة من الطبيعة الحيوية المستنبطة من نتيجة الذكاء الجماعي لسرب من اليراعات المضيئة؛ بالإضافة إلى ذلك تستخدم هذه الطريقة متجهات آلات دعم التمييز (SVM) لتحليل البيانات من أجل تصنيفها والتي تعد من أشهر طرق التصنيف الأتوماتيكي التي تعتمد على إيجاد منحني أو مستوي فائق يفصل العينات المدخلة عن بعضها البعض.

مكنتنا التجارب المنجزة أثناء هذا البحث من التوصل الى نتائج أحسن مقارنة بالأبحاث السابقة في هذا المجال والتي تؤكد أهمية استعمال هذه الطريقة وفعاليتها.

كلمات مفتاحية:

استخراج البيانات، خوارزمية اليراعات المضيئة، التصنيف، التعليم تحت اشراف، اختيار السمات، متجهات آلات دعم التمييز.

Remerciement

Nous tenons tout d'abord à remercier Dieu tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce Modeste travail.

Nous adressons nos remerciements les plus sincères, à

Mr : SAHMADI Brahim

Le promoteur de ce projet. Sa grande connaissance dans le domaine, ainsi que son expérience, ont joué un rôle important dans la conception de ce travail.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leurs propositions.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Dédicace

À les plus belles créatures que Dieu a créée sur terre,,,

À mes source de tendresse, de patience et de générosité,,,

À ma mère et mon père !

À tous mes frères et sœurs, ainsi que leurs enfants.

À toute la famille SISSANI.

À ma chère binôme RAHMANI REKIA *et sa famille.*

À tous mes amis et collègues.

À tous les étudiants de la promotion 2015/2016

Option : Ingénierie des systèmes et technologie du web.



Noura

Dédicace

Je dédie ce modeste travail à :

- *Mes très chers parents qui m'ont toujours soutenu et aidé dans mes choix, et pour leurs vifs encouragements durant ces années d'études.*
- *Mes très chères sœurs et leurs familles et mes frères.*
- *Mon binôme Noura et sa famille.*
- *Tous mes amis de la promotion 2015/2016 section ISTW.*
- *Toutes mes copines et collègues de travail.*
- *Tous ceux qui me sont chers.*



Rehia

Sommaire

Résumé
Sommaire
Liste des figures
Liste des Tableaux
Liste des Acronymes
Introduction Générale …… I

Chapitre I: Sélection D'Attributs

1. Introduction ……..…….....22
2. L'apprentissage automatique ……...……...……..…… 22
   2.1 La fouille de données ……...……..…… 22
   2.2 La classification …… 24
3. Sélection d'attributs ……..…… 25
   3.1 Définition ……..……26
   3.2 Objectifs de la Sélection d'attributs ……..……26
   3.3 Définition pour la sélection d'attributs en classification …… 27
   3.4 Applications de la sélection d'attributs ……..……...27
4. Processus de la sélection d'attributs ……..…… 28
   4.1 Première étape : La procédure de génération……..…..……28
      4.1.1 La direction de recherche ……..……..……29
      4.1.2 Stratégies de recherche……..…….. 30
   4.2. Deuxième étape : Critère et fonctions d'évaluation……..……31
      4.2.1 Critère d'évaluation ……..……...31
      4.2.2La fonction d'évaluation ……..……31
   4.3. Troisième étape : Le critère d'arrêt ……33
   4.4Quatrième étape : Procédure de validation ……..……33
5. Approche de sélection d'attributs ……..…… 36
   5.1 Approche par Filtrage ……..……..…… 36
   5.2 Approche enveloppante…… 37

5.3 Approche intégrée ……....38

6. Méthodes classiques de sélection d'attributs …… 39

6.1 Méthodes utilisés par l'approche Filtre …..…… 39

6.1.1 Focus ……...……..…….39

6.1.2 Branch and Bound ……..……..……. 40

6.1.3 Automatic Branch & Bound ……..…… 40

6.1.4 LVF (Las Vigas Filter) ...……41

6.2 Méthodes utilisés par l'approche enveloppante ……...41

6.2.1 SFS (Sequential Forward Selection) ……..41

6.2.2 SBS (Sequential Backward Selection) ……...…41

6.2.3 Les Métaheuristiques pour la sélection d'attributs …… 42

7. Métaheuristiques de Sélection d'attributs ……..……...……42

7.1. Intensification et diversification ……...…… 43

7.2 Classification des métaheuristiques ……...…… 43

7.2.1 Les métaheuristiques à solution unique ……...……44

7.2.2 Les métaheuristiques à population …….…….. 45

8 Formalisation de notre Problématique ……...…… 47

9. Conclusion …… 48

Chapitre II : L'Algorithme De Luciole & Les Séparateurs à Vaste Marge

1. Introduction ……...…… 50

2. Algorithme des Lucioles pour la sélection d'attribut …….…….. 50

2.1 Inspiration …… 50

2.2 Principe de l'Algorithme ……..……50

2.3 Paramètres de l'Algorithme …….…… 51

2.4 L'Algorithme de FireFly……...……52

2.5 L'Algorithme de luciole binaire……...…… 53

2.6 Cause de l'efficacité de FA ……..……53

2.7 Les applications de l'algorithme FireFly ……. 53

3 .Méthode des Séparateurs à Vastes Marges (SVM) ……..…… 54

3.1 Définition de la méthode SVM ……..……..... 55

3.2Bref historique des SVMs …….…… 55

3.3 Principe de fonctionnement …….…… 55

3.4 Notions de base sur SVM ….....…….... 56

3.5 Maximisation de la marge ……..…….....57

3.6 SVM à deux classes ….....……..… 58

3.6.1 Cas des données linéairement séparable ……...…… 58

3.6.2 Cas des données non linéairement séparable ……...…… 58

3.6.3 C-SVM : SVM à marge douce et le paramètre C ……...……..……59

3.6.4 Fonction noyau RBF et le paramètre σ……...……..…… 60

3.7 SVM multi-classes ……...…….... 61

3.7.1 Un contre tous …….. 62

3.7.2 Un contre un …… 63

3.8 Les domaines d'applications des SVMs ……...…… 63

3.9 Avantages et inconvénients ……..……64

4. Conclusion …… 64

Chapitre III : La Méthode Proposée Pour La Sélection D'Attributs & La Classification

1. Introduction ……...……66

2. Méthode proposée …….…… 66

2.1 Méthode FA-SVM proposée…….…… 67

2.2 Conception générale …….……..... 67

2.2.1 Type d'approche ……....68

2.2.2 Direction et stratégie de recherche ……...… 68

2.2.3 Fonction d'évaluation …… 69

2.2.4 Critère d'arrêt ……69

3. Organigramme générale ……..…….....70

4. Conception détaillée …….…… 71

4.1 Phase d'initialisation ……..…….....71
4.2 Phase d'amélioration …….…….....73

4.3 Architecture de la méthode FA-SVM …….. 75

5. Implémentation de la méthode FA-SVM …….....76

5.1 La spécification du système et analyse des besoins ……...……..… 76

5.2 Conception ……..... 79

5.2.1 Conception technique ……....79
5.2.2 Conception Graphique …….…… 84
5.3. Réalisation ……..……...…… 85

5.3.1 Environnement de développement ……....85
5.3.2 Présentation de l'application ……87

6. Conclusion …… 89

Chapitre VI : RESULTATS ET EVALUATION

1. Introduction …….. 91

2. Matériel et logiciels utilisées ……..... 91

3. Description des bases de Données de l'UCI ……..... 92

4. Prétraitement des données …….……... 93

5. Choix expérimentaux …… 94

5.1 Paramètres de l'algorithme de lucioles ……..94

5.2 Paramètres de SVM …….....94

6. Évaluation des performances de classification …… 95

7. Résultat de Test ……96

7.1 Test de la base de données Heart stat ……...……...……96

7.2 Classification Sans Sélection d'attributs d'un ensemble des bases de données …… 98

7.3 Représentation de TCC et le nombre d'attributs sélectionnés en fonction de nombre d'itérations ……..……... 99

7.4 Interprétation des résultats …… 108

7.5 Comparaison avec les résultats d'autres méthodes ……..108

8. Discussion …….…… 109

9. Conclusion …… 110

Conclusion générale ……..... 111

Bibliographie et Webographie …… 113

Liste des figures

Chapitre I

Figure 1	L'ECD à travers quatre phases …….……	22
Figure 2	Schéma des différentes techniques issues de la fouille de données…...	23
Figure 3	Schéma général de la classification ……...……	24
Figure 4	Processus de sélection d'attributs ……..……..……..	28
Figure 5	Schéma de validation croisé pour l'évaluation d'un processus de sélection ……	34
Figure 6	Schéma de validation croisé d'ordre k…….....……	35
Figure 7	Principe de l'approche par filtrage……..…….……	37
Figure 8	Principe de l'approche enveloppante …..……	38
Figure 9	Comparaison de l'approche enveloppante et Filtre …...……...…….	39
Figure 10	Classification des métaheuristiques ……	44

Chapitre II

Figure 1	Mouvement des Firefly …….……	52
Figure 2	Taxonomie de l'application de Firefly …….……	54
Figure 3	Nuage de points à 2 dimensions avec un hyperplan optimal ...…..…...	57
Figure 4	Hyperplan avec faible marge ...……..……....	57
Figure 5	Meilleur Hyperplan séparateur …..……..……	57
Figure 6	Cas linéairement separable ..……..……....	58
Figure 7	Cas non-linéairement séparable. ……..……....	58
Figure 8	Projection des données dans un espace de dimension supérieure ….....	58

Figure 9 Données dans le cas non séparables ε variable d'écart à la marge objective ɣ qui est un réel strictement positif …..……. 59

Figure 10 Nuage de point à 4 classes avec l'approche Un contre Tous ……... 62

Figure 11 Nuage de point à 4 classes avec l'approche Un contre Un ……...… 63

Chapitre III

Figure 1 L'organigramme général de la méthode FA-SVM ……..…… 70

Figure 2 Representation binaire……..……..…… 72

Figure 3 Architecture générale de la méthode FA-SVM …….…..…… 75

Figure 4 Procédure de la prédiction de la méthode FA-SVM ...…….…… 76

Figure 5 Diagramme de cas d'utilisation1 ...……..…… 77

Figure 6 Diagramme de cas d'utilisation 2 ..……..…… 78

Figure 7 Diagramme de classe ...……...…… 80

Figure 8 Chargement des données d'apprentissage, paramétrage de SVM et l'AF 81

Figure 9 Classification sans selection …...…… 82

Figure 10 Classification avec selection …...…… 83

Figure 11 La prediction ……. 84

Figure 12 Interface de l'IDE Netbeans ……..…… 86

Figure 13 L'interface de l'application -La forme Principale …… 87

Figure14 L'interface de l'application-Forme du réglage des paramètres ……. 88

Figure15 L'interface de l'application-Forme de la prédiction ……..…… 88

Chapitre IV

Figure 1 Graphe de la moyenne de taux de classification en fonction de nombre d'itérations ……..……....…… 100

Figure 2 Graphe de la moyenne du nombre d'attributs sélectionnés en fonction de nombre d'itérations ……...…… 100

Figure 3 Graphe de la moyenne de taux de classification en fonction de nombre d'itérations ….……..…… 101

Figure 4 Graphe de la moyenne du nombre d'attributs sélectionnés en fonction de nombre d'itérations ……..…….. 101

Figure 5 Graphe de la moyenne de taux de classification en fonction de nombre d'itérations …….…….…. 102

Figure 6 Graphe de la moyenne du nombre d'attributs sélectionnés en fonction de nombre d'itérations ...……. 102

Figure 7 Graphe de la moyenne de taux de classification en fonction de nombre d'itérations ……..… 103

Figure 8 Graphe de la moyenne du nombre d'attributs sélectionnés en fonction de nombre d'itérations ……...…… 103

Figure9 Graphe de la moyenne de taux de classification en fonction de nombre d'itérations ……....… 104

Figure10 Graphe de la moyenne du nombre d'attributs sélectionnés en fonction de nombre d'itérations ……...…… 104

Figure11 Graphe de la moyenne de taux de classification en fonction de nombre d'itérations ……...… 105

Figure12 Graphe de la moyenne du nombre d'attributs sélectionnés en fonction de nombre d'itérations …….…….. 105

Figure 13 Graphe de la moyenne de taux de classification en fonction de nombre d'itérations ……...… 106

Figure 14 Graphe de la moyenne du nombre d'attributs sélectionnés en fonction de nombre d'itérations ……...…… 106

Figure 15 Graphe de la moyenne de taux de classification en fonction de nombre d'itérations …… 107

Figure 16 Graphe de la moyenne du nombre d'attributs sélectionnés en fonction de nombre d'itérations ……..……. 107

Liste des Tableaux

Chapitre IV

Tableau 1	Caractéristiques des bases utilisées pour le test ……..…..……	93
Tableau 2	Paramètres de l'algorithme de lucioles……..……..	94
Tableau 3	Paramètres du classifieur SVM…….……...…….……	94
Tableau 4	Classification sans sélection de la base de données Heart-stat …..	96
Tableau 5	Classification avec sélection de la base de données Heart-stat …..	97
Tableau 6	Classification avec et sans sélection d'un ensemble des bases de données …….	98
Tableau 7	Classification avec et sans sélection de la base de données Colon-Cancer …….…..…..……	100
Tableau 8	Comparaison entre GA+SVM, MA+ SVM, SLS+SVM et FA-SVM ……	109

Liste des Acronymes

ABB	**A**utomatic **B**ranch and **B**ound
AFSA	**A**lgorithm **A**rtificial **F**ish **S**warm (Algorithme des essaims de poissons artificiels)
AG	**A**lgorithmes **G**énétiques
AM	**A**lgorithme **M**imétiques
BB	**B**ranch and **B**ound
BDS	**B**i-**D**irectional **S**earch
BE	**B**ackward **E**lemination
CPU	**C**entral **P**rocessing **U**nit (unité centrale de traitement, UCT)
DM	**D**ata **M**ining
ECD	**E**xtraction de **C**onnaissances à partir de **D**onnées
EP	**P**rogrammation **É**volutionnaire
FA	**F**irfly **A**lgorithm
FA-SVM	**F**irfly **A**lgorithm- **S**éparateurs à **V**astes **M**arges
FS	**F**orward **S**election
GO	**G**iga **O**ctet
H	**H**yperplan
HPC	**H**igh-**P**erformance **C**omputer
IA	**I**ntelligence **A**rtificielle
JVM	**J**ava **V**irtual **M**achine
KDD	**K**nowledge **D**iscovery in **D**atabase

LVF	**L**as **V**igas **F**iltre
NBAS	**N**om**B**re des **A**ttributs **S**électionnés
P.O.O	**P**rogrammation **O**rientée **O**bjet
PCA	**A**nalyse des **C**omposants **P**rincipaux
PG	**P**rogrammation **G**énétique
PSO	**P**article **S**warm **O**ptimization (optimisation par les essaims de particules)
RAM	**R**andom **A**ccess **M**emory (memoirs vive)
RBF	**R**adial **B**asis **F**unction
RO	**R**echerche **O**pérationnelle
SBS	**S**equential **B**ackward **S**election
SDM	**S**ystème **D**ata**M**aing
SE	**S**tratégies d'**E**volution
SFS	**S**equential **F**orward **S**election
SLS	**S**tochastic **L**ocal **S**earch
SVM	**S**éparateurs à **V**astes **M**arges
TCC	**T**aux de **C**lassification **C**orrecte
UCI	**U**niversité de **C**alifornie à **I**rvine
UML	**U**nified **M**odeling **L**anguage (langage de modélisation unifié)
ABB	**A**utomatic **B**ranch and **B**ound
AFSA	**A**lgorithm **A**rtificial **F**ish **S**warm (Algorithme des essaims de poissons artificiels)

Introduction Générale

Depuis des années, le rêve des chercheurs dans le domaine de l'intelligence artificielle était de concevoir des machines capables d'apprendre. Pour atteindre cet objectif, les systèmes dit à apprentissage automatique, sont souvent utilisés dans le contexte du datamining, pour comprendre et reproduire l'habileté humaine, d'apprendre de ses expériences passées et de l'adapter dans les systèmes artificiels. L'apprentissage automatique (Machine Learning en anglais), champ d'étude de l'intelligence artificielle, concerne la conception, l'analyse, le développement et l'implémentation de méthodes permettant à une machine d'évoluer, dans la résolution des problèmes.

L'apprentissage automatique et exactement l'apprentissage supervisé utilisée par les systèmes de classification vis à extraire une règle générale ou une procédure de classification à partir d'un ensemble des données d'apprentissage. Ces données qui sont représentées par de nombreux attributs, beaucoup sont non-significatifs, redondants ou inutiles, ce qui crée du bruit pour les méthodes de classification et diminue la qualité de l'apprentissage et par conséquent crée des modèles de prédiction inexactes.

Pour remédier à ce problème et assurer une efficace utilisation des techniques de datamining, une phase de prétraitement de données est indispensable. La sélection d'attributs est l'une des techniques utilisée comme une phase préliminaire aux algorithmes d'apprentissage supervisé pour la classification. C'est un sujet de recherche très actif depuis une dizaine d'années sur lequel travaillent activement les spécialistes en Datamining et en optimisation.

Le problème de sélection d'attributs consiste donc à choisir parmi un ensemble d'attributs de très grande taille, un sous-ensemble optimal plus petit d'attributs qui a les propriétés suivantes : il doit être composé d'attributs pertinents et doit chercher à éviter les attributs redondants pour réduire la dimension de l'espace de travail. De plus, cet ensemble doit permettre de satisfaire au mieux l'objectif fixé, à savoir la précision de l'apprentissage.

Dans ce contexte, l'objectif de la sélection d'attributs est d'améliorer la qualité de l'apprentissage supervisé du classifieur utilisé dans le but de créer des modèles de prédiction de haute performance permettant de prédire correctement la classe de nouveaux exemples.

En plus, L'évolution de l'informatique et des technologies de stockage connait une explosion de volumes des données. Grâce aux performances accrues des ordinateurs et aux méthodes utilisées par les Système de DM ,Il est maintenant possible d'analyser de grandes quantités de données de dimension élevée et d'en extraire des mines d'informations.

Cependant, Les premières approches de sélection d'attributs s'appuient sur des algorithmes de recherche et des méthodes classiques de sélection d'attributs qui semblent moins adaptées à l'augmentation de la dimensionnalité qui est le problème majeur face aux Système de DM pour la classification et représente un nouvel défi pour ces méthodes. Pour combler ce besoin, la tendance de la recherche est orientée actuellement vers des méthodes bio-inspirées appelées des métasheuristiques, en particulier celles issues de l'intelligence en essaim.

L'algorithme de luciole (Firefly Algorithm) est une métaheuristique très récente issue de l'intelligence en essaim inspiré par le comportement de clignotement de lucioles.

Nous proposons à travers ce mémoire une application de cet algorithme dans le domaine de datamining et plus particulièrement il s'agit de la proposition de la méthode FA-SVM comme une méthode de sélection d'attributs pour la classification supervisé.

L'idée fondamentale de cette méthode est d'utiliser une métaheuristique basée sur l'algorithme des essaims de lucioles (Firfly Algorithm) au début du processus de recherche pour la sélection des attributs pertinents et éliminer les attributs redondants et d'exploiter pleinement la puissance de l'apprentissage du classifieur SVM à noyau RBF qui a démontré son capacité à obtenir d'excellentes performances pour des tâches de classification.

La validation de notre méthode a été faite en utilisant la validation croisée à 10. Afin d'évaluer notre méthode, nous avons réalisé une série d'expérimentations sur des bases de données réelles. Les résultats de ces expérimentations sont prometteurs et confirment l'intérêt d'utiliser cette méthode dans la sélection d'attributs.

Notre mémoire comprend quatre chapitres structurés comme suit :

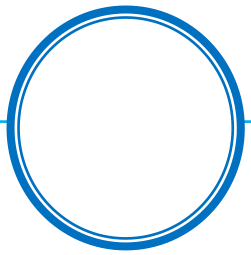
Dans le premier chapitre, nous présenterons le domaine et le processus de la sélection d'attributs, ensuite les différentes approches, les méthodes classiques et bio-inspirées de la sélection d'attributs seront exposé.

Le deuxième chapitre sera consacré à l'étude de l'algorithme de luciole et les Séparateurs à Vaste Marge(SVM).

Le troisième chapitre présentera la méthode proposée FA-SVM qui sera étudiée et développée au cours de ce mémoire.

Dans le quatrième chapitre, nous présenterons et discuterons les différents résultats obtenus.

Nous terminerons notre mémoire par une conclusion générale qui contient les perspectives ouvertes par ce travail de recherche.



CHAPITRE I

Sélection D'Attributs

1 Introduction

La croissance extrêmement rapide des données collectées dans les bases de données, et la nécessité d'une réactivité efficace de la part des décideurs face à ces informations nouvelles, ont stimulé cette dernière décennie, le développement rapide de l'Extraction des Connaissances à partir des Données. L'ECD[1] est le processus [2] non trivial, interactif, itératif qui permet d'identifier des modèles valides, nouveaux, potentiellement utiles et compréhensibles à partir de bases de données massives [33].

2 L'apprentissage automatique :

2.1 La fouille de données :

Contrairement aux idées reçues, l'ECD ne se limite pas exclusivement à la fouille de données qui en constitue toutefois la partie visible [10]. L'ECD se compose en effet de quatre phases [26] : acquisition et stockage des données (Data Warehousing), Pré-traitement de données (Pré-Processing), fouille de données (Data Mining) et post-traitement (Post-Processing). Dans le cadre de ce mémoire, nous nous intéressons à l'étape 2, mais avant d'aborder la sélection d'attributs, nous donnerons un aperçu sur le datamining, L'apprentissage et la classification supervisée.

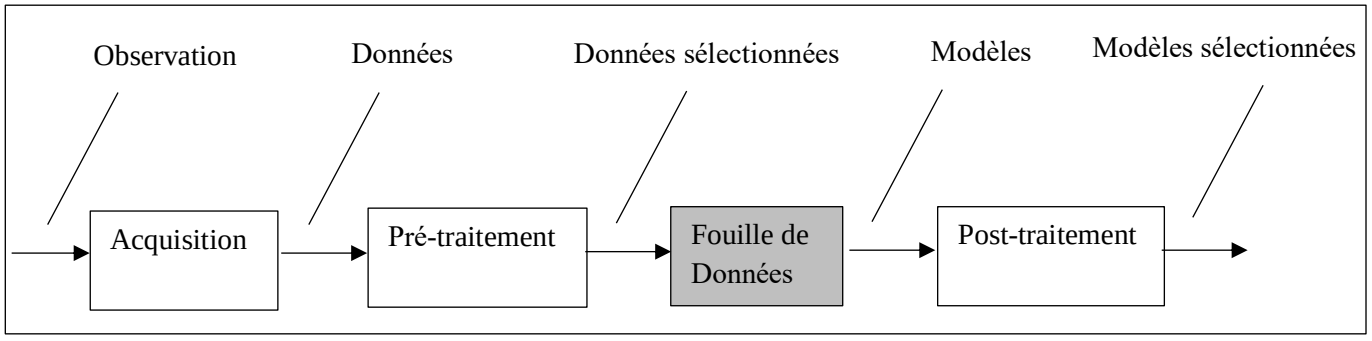


Figure I.1: *L'ECD à travers quatre phases*

[1] Extraction de connaissances à partir de données (ECD), et Knowledge Discovery in Database (KDD) en anglais.

[2] Un ensemble d'étapes et d'actions dont la finalité est l'extraction des tendances et des corrélations au sein des données

La fouille de données [3] est un domaine très vaste qui se situe à l'intersection de différents domaines tels que l'informatique, l'intelligence artificielle, et les statistiques[9].La fouille de données a été définit selon le Groupe Gartner comme« Le processus de découverte de nouvelles corrélations, modèles et tendances en analysant une grande quantité de données, en utilisant les technologies de reconnaissance des formes ainsi que d'autres techniques statistiques et mathématiques»[19]. Elle se propose d'utiliser un ensemble d'algorithmes et de méthodes pour construire des modèles à partir des données, et d'en extraire le maximum des connaissances. La fouille de données regroupe différentes techniques qui sont représentées sur le schéma de la Figure I.2 suivante :

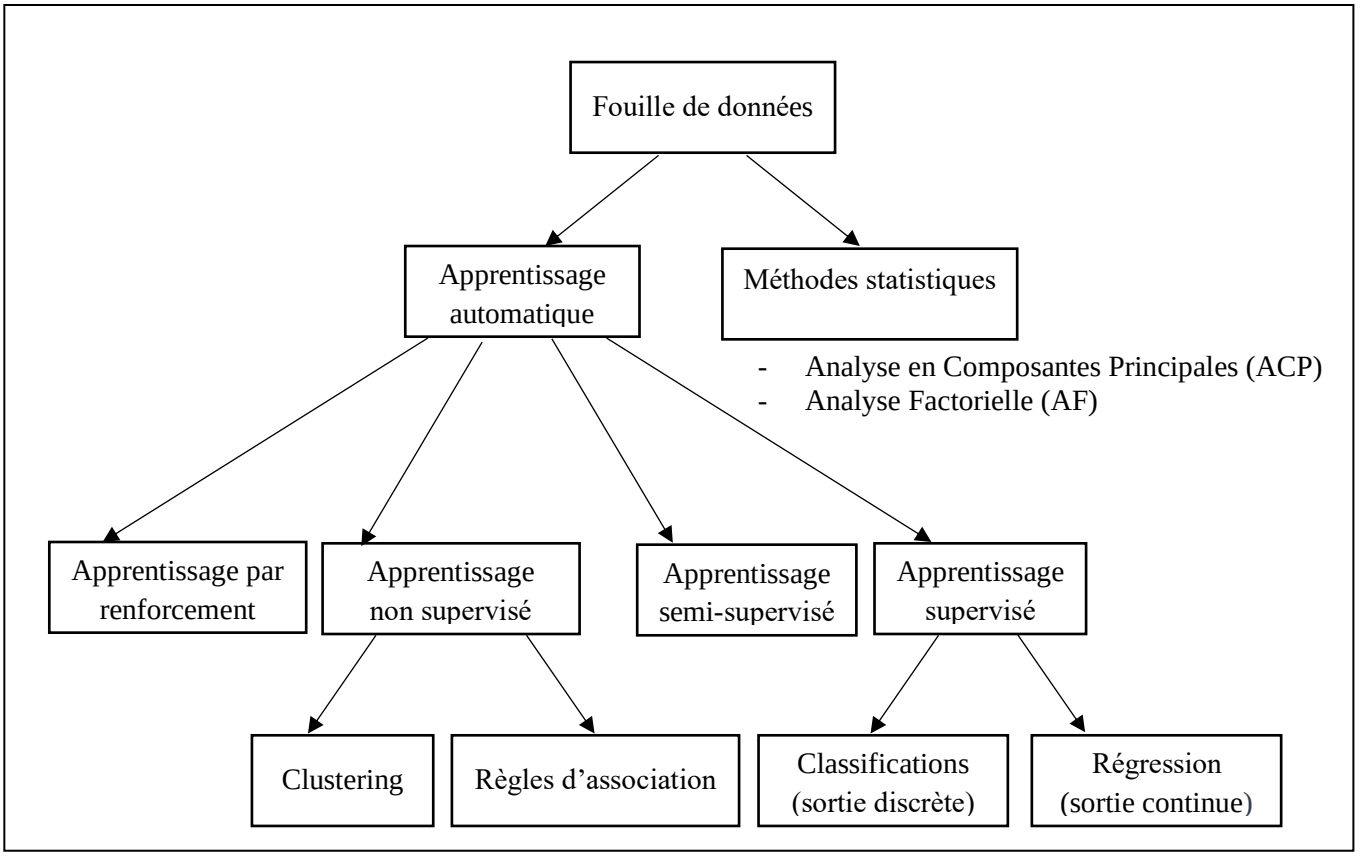


Figure I.2 : *Schéma des différentes techniques issues de la fouille de données : [11]*

[3] Data Mining ou DM, forage de données, prospection de données et l'exploration de données.

2.2 La classification :

L'apprentissage automatique et spécialement l'apprentissage supervisé est l'une des techniques de la fouille de données qui consiste à extraire une règle générale ou une procédure de classification à partir d'un ensemble d'exemples [4] d'apprentissage.

Le terme classification peut désigner deux classes de méthodes distinctes : la classification supervisée et la classification non-supervisée[5]. La classification non supervisée a pour but de constituer des groupes d'exemples, (ou des groupes d'instances) en fonction des données observées, sans connaissance a priori. En revanche, La classification supervisée utilise la connaissance a priori, sur l'appartenance d'un exemple à une classe pour construire un système de reconnaissance de ces classes.

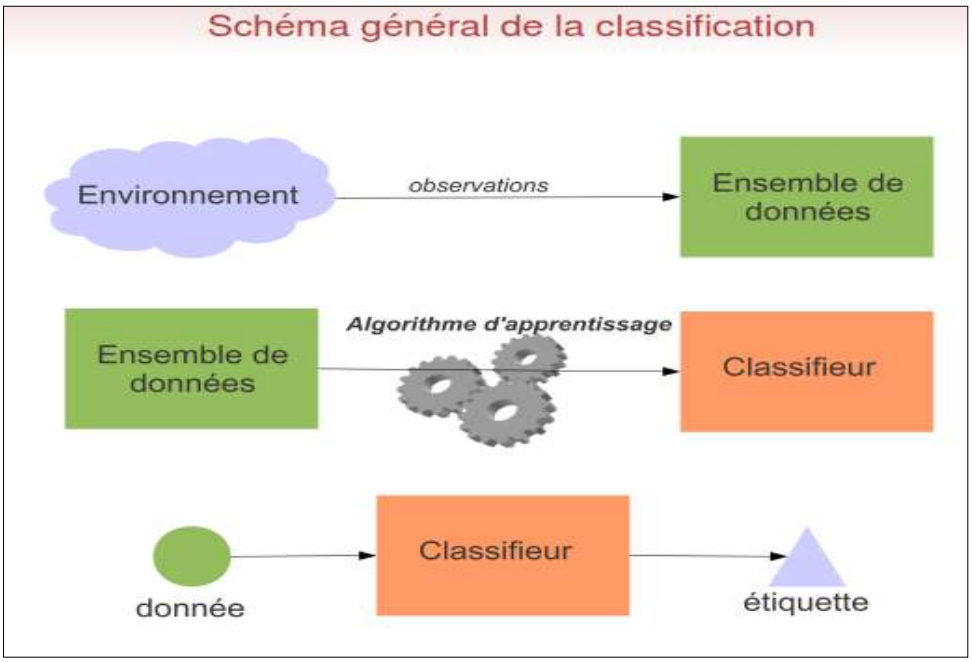


Figure **I.3 :** *Schéma général de la classification* [20]

Dans ce mémoire, nous nous intéressons à la classification à partir des exemples (la classification supervisée). L'objectif de cette tâche est d'apprendre à l'aide des exemples, une procédure de classification qui permet de prédire l'appartenance d'un nouvel exemple à une classe. Les méthodes utilisées pour la classification supervisée sont

[4] Est une donnée pour laquelle on dispose de sa classe. On utilise donc un ensemble d'exemples d'apprentissage classés pour prédire la classe de nouvelles données ; c'est une tâche « d'apprentissage à partir d'exemples », ou « apprentissage supervisé ». [9]

[5] Automatic classification et Clustering en anglais.

nombreuses, citons : la méthode des Séparateurs à Vastes Marges (SVM), les Réseaux de Neurones,… etc. Ces outils peuvent prédire les futurs tendances et actions, permettant de prendre les bonnes décisions. C'est ce qui rend le datamining la technologie la plus importante. Nous présenterons en détail dans le chapitre II la méthode de classification SVM.

En outre, un système de classification inclut deux étapes [12] :

- Une étape d'apprentissage qui peut être vue comme une phase de développement, aboutissant à la mise en œuvre d'une stratégie de classification. Tout ce qui est appris par le classifieur [6] est représenté sous la forme des règles de classification.
- Une étape de test au cours de laquelle les performances du système de classification sont évaluées, et dans laquelle les données de test vont être utilisées pour estimer la précision des règles de classification générées pendant la première phase. Si la précision du modèle est considérée comme acceptable, la règle pourra être appliquée à des nouvelles données.

3 Sélection d'attributs :

Dans un système de classification, Plusieurs paramètres peuvent influer sur ses performances. Notamment la qualité des attributs qui pose des problèmes majeurs dans les applications. Il est possible que certains de ces attributs correspondent au bruit ou qu'ils soient peu informatifs, redondants ou même inutiles au problème de la classification.

La sélection d'attributs consiste à choisir un sous-ensemble d'attributs intéressants pour le problème étudié. Cette problématique peut concerner différentes tâches d'apprentissage ou de fouille de données, dans les domaines du traitement d'images et de l'analyse de données en bio-informatique. Dans tous ces domaines, les applications nécessitent de traiter des données décrites par un très grand nombre d'attributs.

[6] Est un algorithme qui, à partir d'un ensemble d'exemples, produit une prédiction de la classe de toute donnée. [9]

De ce fait, il est devenu indispensable de proposer des méthodes efficaces pour sélectionner les attributs pertinents. Un grand nombre d'algorithmes de sélection d'attributs est disponible dans la littérature. On distingue deux catégories :

Les algorithmes de classement des attributs (Feature ranking) et les algorithmes de recherche de sous-ensembles (Subset selection).

- La première catégorie d'algorithmes consiste à ordonner l'ensemble d'attributs de départ selon un critère d'évaluation, et à sélectionner ensuite les attributs les plus pertinents vis-à-vis du critère utilisé.
- La deuxième catégorie recherche le sous-ensemble d'attributs le plus pertinent selon un certain critère de sélection, et c'est la catégorie d'algorithmes qui nous intéresse. Il s'agit plus particulièrement de la sélection d'attributs réalisée pour la classification supervisée.

3.1 Définition :

La sélection d'attributs, de caractéristiques[7], est un problème difficile qui a été étudié depuis les années 70. La sélection d'attributs est un processus de recherche ou une technique utilisée pour choisir les caractéristiques, les variables ou les mesures les plus intéressantes, pertinentes ou informatives d'un système donné, dans le but de réaliser la tâche pour laquelle il a été conçu.[14]

3.2 Objectifs de la Sélection d'attributs :

La sélection d'attributs pour la classification supervisée vise à atteindre les objectifs principaux suivants :

- L'amélioration de la qualité de l'apprentissage supervisé du classifieur utilisé dans le but de créer des modèles de prédiction de haute performance permettant de prédire correctement la classe de nouveaux exemples.
- Sélection d'un sous-ensemble plus petit d'attributs pertinents et élimination des attributs redondants ainsi que les attributs qui sont source de bruit.

[7] Attribute or feature selection en anglais

- Grâce à la réduction de la dimensionnalité via la sélection d'attributs, non seulement le temps d'apprentissage et d'exécution sera réduit mais aussi l'apprentissage deviendra moins coûteux.

3.3 Définitions pour la sélection d'attributs en classification :

Dans l'apprentissage automatique, La notion de la redondance d'attributs se comprend intuitivement et elle est généralement exprimée en termes de corrélation entre les attributs. On peut dire que deux attributs sont redondants (entre eux) si leurs valeurs sont complètement corrélées.

La notion de pertinence n'a pas de définition formelle qui est acceptée par tous, malgré les efforts déployés par les différents auteurs, Parce que le concept de la pertinence est associé à une tâche particulière. La définition suivante donnée par Almullin et Diettrich en 1991 considère que tous les attributs sont des booléens.

Définition1 (attribut pertinent) : Un attribut A_i est dit pertinent pour un concept C si A_i apparait dans chaque formule booléenne qui représente C. Il est dit non pertinent sinon. [26]

Définition2 (attribut pertinent) : Un attribut Ai est pertinent, s'il existe deux instances (X1, Y1) et (X2, Y2) appartenant à deux classes différentes ($y1 \neq y2$) telles que : la valeur de l'attribut Ai est différente pour les deux instances et les valeurs des autres attributs sont identiques. [26]

LAETITIA JORDAN dans [26] donne la définition suivante à un sous ensemble optimal :

Définition3 (Sous ensemble optimal) : Soit un inducteur I, et un ensemble de données d'entrainement D avec les attributs A1...An. Un sous ensemble d'attribut optimal, A_{opt} est le sous ensemble qui maximise la qualité de classifieur, mesuré par le pourcentage de prédiction correcte sur un ensemble de test.

Définition4 (bruit) : on dit que les données sont bruitées lorsque certains attributs ont une valeur inconnue ou invalide.

3.4 Applications de la sélection d'attributs :

Comme beaucoup de systèmes, dans une vaste variété de domaines, traitent de grandes quantités de données, la sélection d'attributs a trouvé une très large applicabilité. Parmi les principaux domaines d'application nous pouvons citer la fouille de données de

manière générale et la catégorisation de textes en particulier, la reconnaissance de formes telle que l'écriture et les images ainsi que le domaine de la bio-informatique.

4 Processus de la sélection d'attributs :

Les différentes méthodes proposées dans la littérature pour la sélection d'attributs utilisent une procédure de sélection généralement composée de quatre étapes de base suivante [31] :

- **Une procédure de génération :** pour générer le prochain sous-ensemble d'attributs.
- **Un critère d'évaluation :** pour évaluer le sous-ensemble généré par la procédure de génération.
- **Un critère d'arrêt :** pour décider quand arrêter la procédure de recherche.
- **Une procédure de validation :** pour vérifier si le sous-ensemble est valide ou pas.

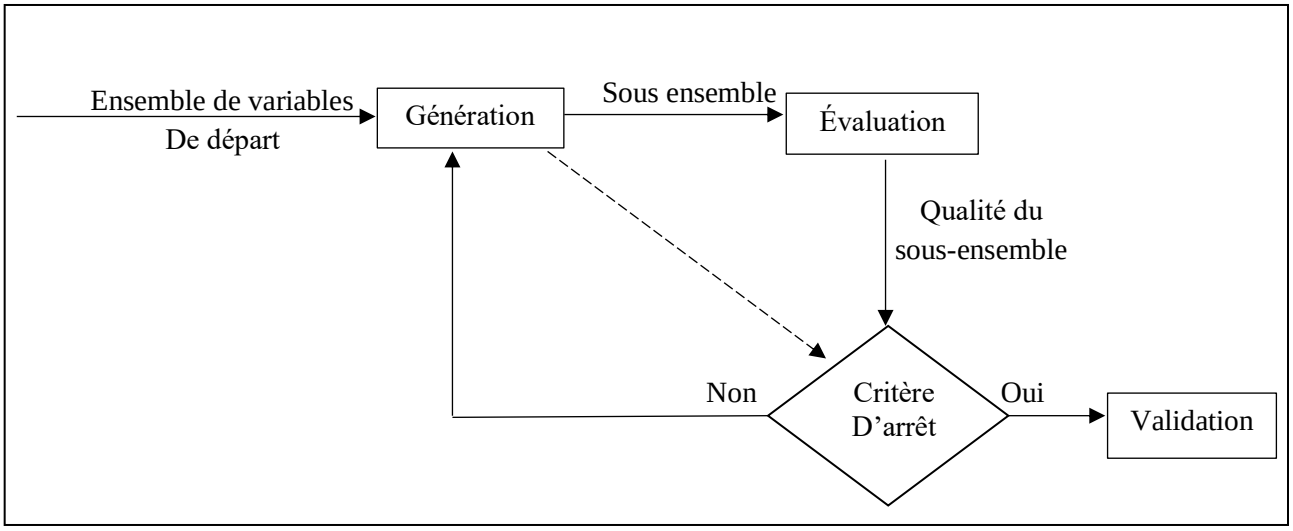


Figure I.4 : *Processus de sélection d'attributs* [31]

4.1 Première étape : La procédure de génération

La procédure de génération est une procédure de recherche permettant d'explorer l'espace de recherche pour construire les différentes combinaisons des attributs. La génération de sous-ensemble est essentiellement un processus de recherche qui à chaque étape détermine un sous-ensemble candidat dans l'espace de recherche pour l'évaluation. [32]

La nature de ce processus est déterminée par deux choix fondamentaux :

- **La direction de recherche** (le point (ou les points) de départ de la recherche).
- **La stratégie de recherche** (recherche complète, heuristique et aléatoire).

4.1.1 La direction de recherche :

Les directions de recherche peuvent être de trois types :

a) **Ascendante : « forward selection (FS) » :** Cette stratégie part d'un ensemble vide. Les attributs sont ajoutés un à un. À chaque itération, l'attribut optimal suivant un certain critère est ajouté. Le processus s'arrête soit quand il n'y a plus d'attributs à ajouter, soit quand un certain critère est satisfait. Une fois qu'un attribut a été ajouté, la FS ne peut le retirer.

b) **Descendante « backward elemination(BE) »** : Cette stratégie part de l'ensemble initial d'attributs. A chaque itération, un attribut est enlevé de l'ensemble. Cet attribut est tel que sa suppression donne le meilleur sous-ensemble selon un critère particulier. Une fois l'attribut supprimé, il est impossible de le réintégrer.

c) **Les méthodes bidirectionnelles(Stepwise) :**Ces méthodes permettent de pallier le problème de l'irrévocabilité de la suppression ou de l'ajout d'un attribut, problème présent dans les deux autres directions de recherche. En effet, l'importance d'une variable peut se modifier ultérieurement. Ces méthodes autorisent l'ajout et la suppression d'un attribut de l'ensemble des attributs à n'importe quelle étape de la recherche autre que la première ou la dernière.

En résumé, L'ensemble considéré comme point de départ par [18] peut être :

a) l'ensemble de tous les attributs disponibles.

b) l'ensemble vide.

c) un ensemble d'attributs tirés de manière aléatoire.

d) une population de sous-ensembles d'attributs lorsqu'on travaillera avec des méthodes à base de population.

4.1.2 Stratégies de recherche :

a) Génération complète : Les approches regroupées dans cette catégorie effectuent une recherche complète du sous-ensemble optimal par rapport à la fonction d'évaluation choisie. L'objectif est de trouver le sous-ensemble optimal par rapport à la fonction d'évaluation, qui est garantie et peut être réalisé en utilisant des technique tel que : Branch and Bound (BB), Best First Search (BFF), Automatic Branch and Bound (ABB), Focus. [31]

b) Génération séquentielle ou heuristique : Cette catégorie regroupe les algorithmes itératifs pour lesquels chaque itération permet de sélectionner ou rejeter un ou plusieurs attributs. Les algorithmes avec une génération séquentielle sont simples à implémenter et rapides dans la production des résultats**.** Parmi les techniques utilisées, on trouve: Sequential Forward Selection (SFS), Sequential Backward Selection (SBS), sequential backward selection-SLASH (SBS- SLASH), bi-directional search (BDS), Relief [32]

c) Génération aléatoire : Par rapport aux deux premières catégories, l'utilisation de la génération aléatoire dans la procédure de sélection d'attributs est la plus récente. La procédure commence avec un sous ensemble sélectionné aléatoirement et procède de deux façons différentes :

- L'une consiste à continuer la génération des sous-ensembles avec la recherche séquentielle.
- l'autre consiste à générer le sous ensemble suivant d'une manière complètement aléatoire.

Ces méthodes recherchent des sous-ensembles en effectuant un maximum d'itérations. Le choix de valeurs appropriées aux paramètres pour la procédure de recherche est une étape très importante afin d'obtenir des résultats satisfaisants. Parmi les techniques utilisées, on peut citer les algorithmes génétiques (AG), le recuit simulé. [14]

4.2 Deuxième étape : Critère et fonctions d'évaluation

4.2.1 Critère d'évaluation :

On distingue deux grandes approches pour la sélection d'attributs suivant que l'on va évaluer les attributs individuellement ou évaluer des sous-ensembles d'attributs.

a) **L'évaluation individuelle :** L'évaluation individuelle consiste à évaluer chaque attribut indépendamment des autres à partir des informations fournies par les données en lui attribuant un poids (score) selon son degré de pertinence. Ce score permet d'établir une liste classée des attributs.

b) **L'évaluation d'un sous-ensemble :** Beaucoup de méthodes de sélection s'appuient sur l'évaluation des sous-ensembles pour gérer à la fois la redondance et la pertinence. La génération de sous-ensembles produit des sous-ensembles candidats suivant une stratégie de recherche. Chaque sous-ensemble d'attributs candidat est évalué par une certaine mesure d'évaluation et comparé avec le meilleur sous-ensemble d'attributs obtenu précédemment par rapport à cette mesure. Si le sous-ensemble courant est meilleur, il remplace le meilleur sous-ensemble d'attributs mémorisé. Le processus de génération et d'évaluation d'un sous-ensemble est répété jusqu'à ce qu'un critère d'arrêt soit satisfait**.** [18]

Avantage : Les mesures d'évaluation utilisées par cette approche prennent en compte l'existence et l'effet d'attributs redondants en plus qu'un sous-ensemble d'attributs sélectionné par cette approche se rapproche d'un sous-ensemble optimal.

4.2.2 La fonction d'évaluation :

La fonction d'évaluation est utilisée pour mesurer la pertinence des attributs en les appréciant de manière individuelle ou sous-ensembles d'attributs. Différentes fonctions d'évaluation ont été proposées pour évaluer un attribut ou un sous-ensemble d'attributs dans un contexte de sélection. Ces mesures d'évaluation sont divisées selon Dash et Liu [31] en cinq catégories alors que Liu et Yu [32] les regroupent en deux catégories : des mesures dépendantes et indépendantes.

4.2.2.1 Les mesures indépendantes :

a) Les mesures d'information : Les mesures d'information déterminent le gain d'information pour un attribut considéré, le gain d'information apporté par un attribut étant estimé à partir des probabilités a posteriori. Un attribut X est préféré à un attribut Y si le gain d'information apporté par l'attribut X est plus grand que celui apporté par l'attribut Y.

b) Les mesures de dépendance : Les mesures de dépendance peuvent être divisées en deux catégories :

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

c) Les mesures de consistance : Les mesures de consistance cherchent à évaluer si l'attribut (ou le sous-ensemble d'attributs) étudié contient les informations nécessaires à la discrimination des classes.

d) Les mesures de distance : Les mesures de distance sont aussi nommées mesures de séparabilité, divergence ou de discrimination. Un attribut ou un sous-ensemble d'attributs est sélectionné s'il permet une meilleure séparabilité et cohérence des classes. En effet, le but est de :

- maximiser la dispersion interclassée (séparabilité), afin que les points représentatifs des différentes classes forment dans l'espace d'attributs des nuages les plus séparés possibles les uns des autres.
- minimiser la dispersion intra-classe (cohérence), afin que les nuages de points représentatifs de chaque classe soient les plus compacts possible.

4.2.2.2 Les mesures dépendantes :

Cette catégorie de critères est celle utilisée par toutes les méthodes enveloppe qui utilisent le classifieur comme une fonction d'évaluation.

- **Les mesures de précisions** : Le classifieur choisit, parmi tous les sous-ensembles de variables, celui qui a la meilleure précision prédictive.
- **Les mesures d'erreur de classification :** Le sous-ensemble d'attributs le plus discriminant est celui pour lequel le taux d'erreur de classification est le plus faible.

4.3 Troisième étape : Le critère d'arrêt :

Le nombre optimal d'attributs n'étant pas connu a priori, il sera fixé grâce à un critère d'arrêt du processus de sélection. L'utilisation d'une règle pour contrôler la procédure de sélection permet d'arrêter la recherche lorsqu'aucun nouvel attribut n'est suffisamment informatif. C'est un choix souvent défini en fonction de la procédure de recherche et/ou du critère d'évaluation. Les critères d'arrêt les plus fréquents sont :

- **Basés sur l'algorithme de génération :** on peut par exemple décider d'arrêter la recherche en fixant un seuil sur le nombre d'attributs à sélectionner ou sur le nombre d'itérations.
- **Basés sur l'évaluation :** dans ce cas, on arrête la recherche en fixant un seuil soit sur la fonction d'évaluation. soit on peut ainsi décider d'arrêter la procédure de sélection dès que ce taux diminue ou alors dès qu'il atteint un certain seuil.

4.4 Quatrième étape : Procédure de validation :

Une méthode de sélection d'attributs (en pratique) doit être validée. Dash et Liu [31] proposent d'ajouter une quatrième composante à un algorithme de sélection d'attributs : une procédure de validation. Elle essaie de tester la validité des sous-ensembles des attributs sélectionnées avec la réalisation de différents tests, et la comparaison des résultats obtenus avec les résultats précédents, ou avec des résultats d'autres méthodes de sélection de caractéristiques en utilisant des données artificielles, réelles, ou les deux. Parmi les techniques utilisées : la validation croisée.

Le modèle de validation pour la sélection et la classification d'attributs est présenté dans la Figure I.5, qui montre la partition de données en un ensemble d'échantillons d'apprentissage et un ensemble d'échantillons de test. Cette méthode peut être utilisée pour évaluer n'importe quel type de méthode de sélection, soit de filtrage soit d'enveloppe ou soit d'approche intégrée.

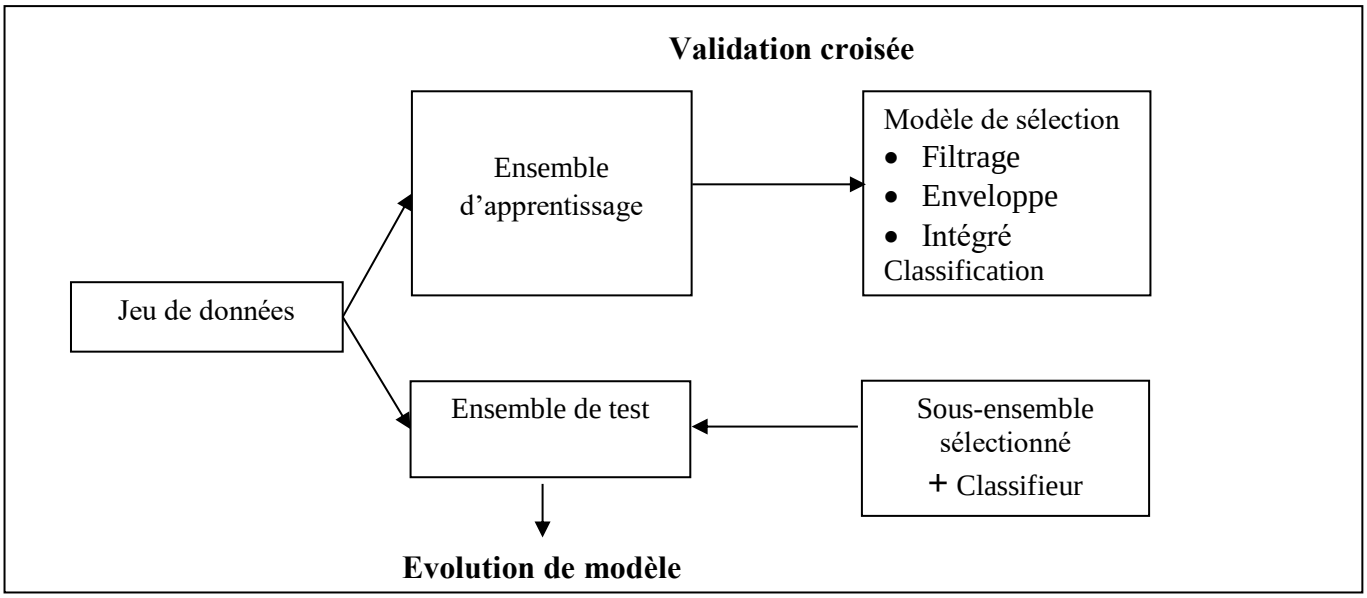


Figure I.5 *: Schéma de validation croisé pour l'évaluation d'un processus de sélection*

a) Validation croisée : K-Fold : L'algorithme de validation croisé k-Fold consiste à :

- déviser les exemples du jeu de données initial en k- sous-ensembles de taille égale numérotés de 1 à k.
- Ecarter ensuite le 1er bloc qui nous servira d'ensembles de test, et utiliser les k-1 autre blocs afin de constituer l'ensemble d'apprentissage.
- Puis, suivre le protocole classique de construction du modèle de prédiction sur l'ensemble d'apprentissage, en validant ses performances sur l'ensemble de test. On obtient donc une estimation des performances du prédicteur calculé sur les blocs 2 à k et dont on a validé les performances sur le 1er bloc.
- Recommencer ensuite cette opération en réservant chaque bloc successivement comme bloc de test, et on calcule la moyenne et l'écart-type des résultats sur les k expériences.

De manière générale, on choisit k=10 si le jeu de données est suffisamment grand.

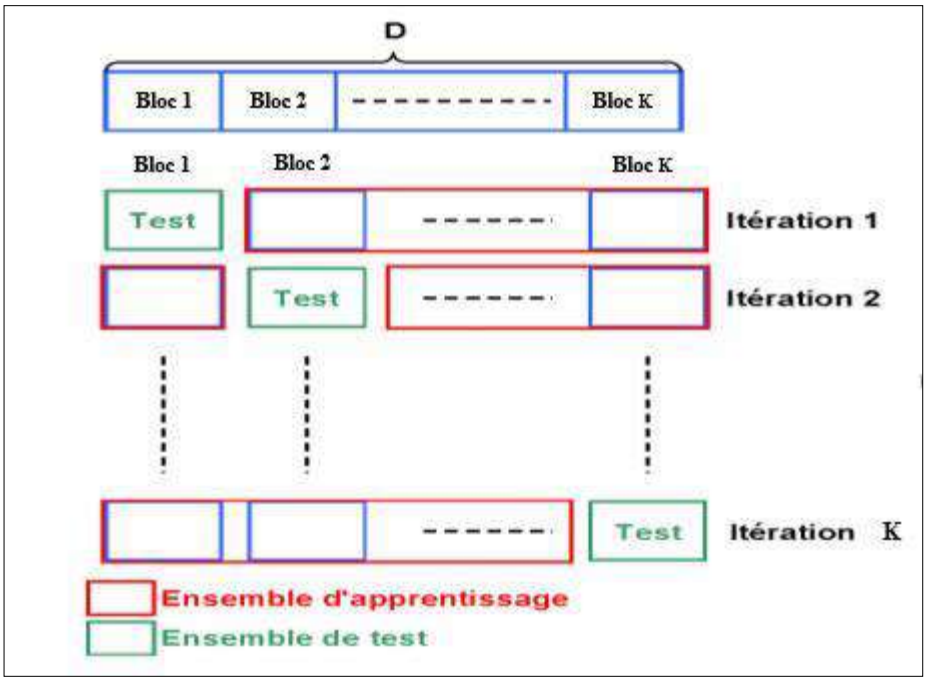


Figure I.6 : *Schéma de validation croisé d'ordre k*

b) Validation croisée : Leave One Out : Cette méthode est dérivée de la méthode de validation croisée k-Fold en prenant k= n. n étant le nombre d'exemples. A chaque itération, on va donc faire l'apprentissage sur tous les exemples, moins un, et tester sur un seul exemple, afin de vérifier s'il est correct. [11]

Cette méthode est plus couteuse, en terme de temps de calcul, que la validation croisée k-fold, car elle nécessite beaucoup plus de tests .Cependant, elle sera intéressante pour des jeux de données sur lesquels on ne dispose que de peu d'exemples. Ou alors si l'on a besoin d'une estimation plus précise des performances du modèle de prédiction.

c) Validation croisée : Repeated Random Sub sampling : Cette méthode est également dérivée de la méthode de validation croisée k-Fold. L'idée est également de découper aléatoirement la population, classiquement en 2/3 pour l'ensemble d'apprentissage et 1/3 pour l'ensemble de test qui est appelé 3-fold dans la littérature. [11]

d) Bootstrapping : C'est une technique très souvent utilisée dans le cadre de jeux de données contenant peu d'exemples. Lorsque l'on utilise le principe de Bootstrapping, on sélectionne aléatoirement les exemples dans le jeu de données initial afin de constituer les ensembles d'apprentissage et de test, en prenant garde à ne pas prendre deux fois les mêmes exemples. [11]

5 Approche de sélection d'attributs :

Ces approches se distinguent par la manière dont la sélection d'attributs interagit (ou non) avec le mécanisme de classification. Nous distinguons trois approches.

5.1 Approche par filtrage (Filtre) :

a) Principe :

- Dans cette approche la sélection des attributs est une étape indépendante de la construction du classifieur. C'est une étape de prétraitement des données avant la classification.
- 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 qui sont utilisé pour estimer le pouvoir individuel de prédiction de la classe des attributs.
- Pour donner un score à un sous-ensemble, une première solution est de donner un score à chaque variable indépendamment des autres et de faire la somme de ces scores. Cette approche est nommée **Feature Ranking.** L'autre solution est d'évaluer un sous-ensemble dans sa globalité, cette approche est nommée **subset ranking**. La figure suivante I.7 : représente son principe.

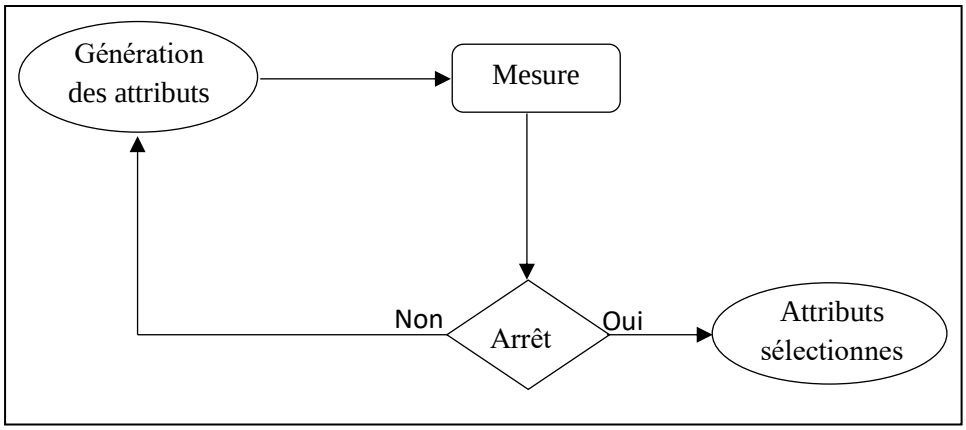


Figure I.7 : *Principe de l'approche par filtrage*

b) Avantages & Inconvénients : Cette approche présente les avantages suivants :

1. L'ensemble d'attributs sélectionnés peut être utilisé pour différentes méthodes de classification, car l'approche est indépendante de l'algorithme d'apprentissage.

2. La détermination du sous-ensemble d'attributs est beaucoup plus rapide que dans le cas de l'approche enveloppante, car le temps de calcul d'une mesure de distance, de consistance ou de dépendance est moins que le temps de calcul du taux de classification.

3. Vu le faible temps de calcul et la simplicité des mesures utilisées, cette approche peut travailler avec des bases de données assez larges.

En contrepartie, on peut noter l'inconvénient suivant :

1. L'utilisation de l'approche de filtrage ne garantit pas qu'on explore au maximum le biais de la méthode de classification.

5.2 Approche enveloppante (Wrapper) :

a) Principe :

- L'approche enveloppante utilise l'algorithme d'apprentissage comme une fonction d'évaluation.
- Elle définit donc la pertinence d'un sous-ensemble d'attributs par l'intermédiaire d'une prédiction de la performance du système final.

- L'approche filtre est plus rapide que l'approche enveloppante en terme de génération de résultats. Cependant, cette dernière à l'avantage de fournir généralement des résultats plus pertinents pour la classification.

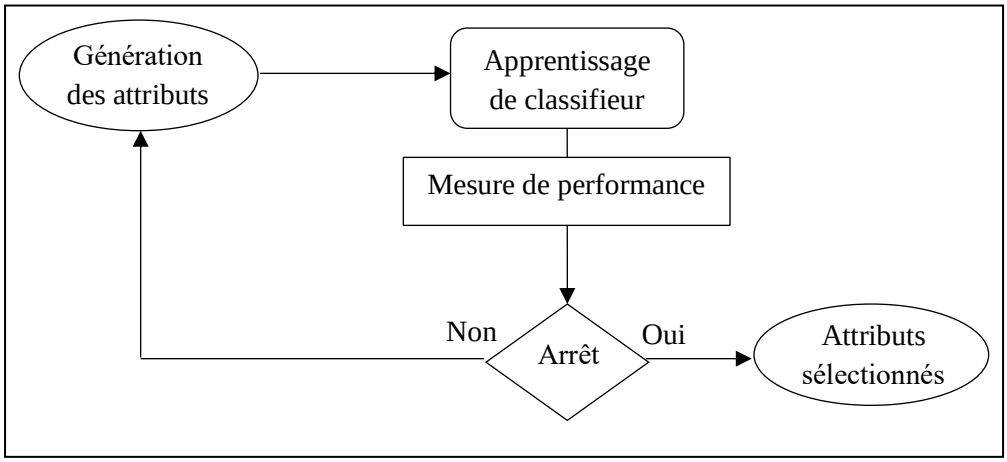


Figure I.8 : *Principe de l'approche enveloppante*

b) Avantages & Inconvénients :

L'avantage principal de cette approche est qu'elle permet d'explorer au maximum tout le biais de l'algorithme d'induction et de fournir généralement des résultats plus pertinents pour la classification.

En contrepartie, elle présente les inconvénients suivants :

1. L'ensemble des attributs sélectionnés est spécifique à l'algorithme d'induction utilisé. Donc nous ne pouvons pas garantir que l'ensemble trouvé donne des bons résultats avec d'autres d'algorithmes.

2. Le calcul de taux de classification est très gourmand côté temps de calcul.

5.3 Approche intégrée (Embedded) :

Cette approche est proche des méthodes d'enveloppe, car elles combinent le processus d'exploration avec un algorithme d'apprentissage. La différence avec les méthodes

enveloppes est que le classifieur sert non seulement à évaluer un sous ensemble candidats mais aussi à guider le mécanisme de sélection.

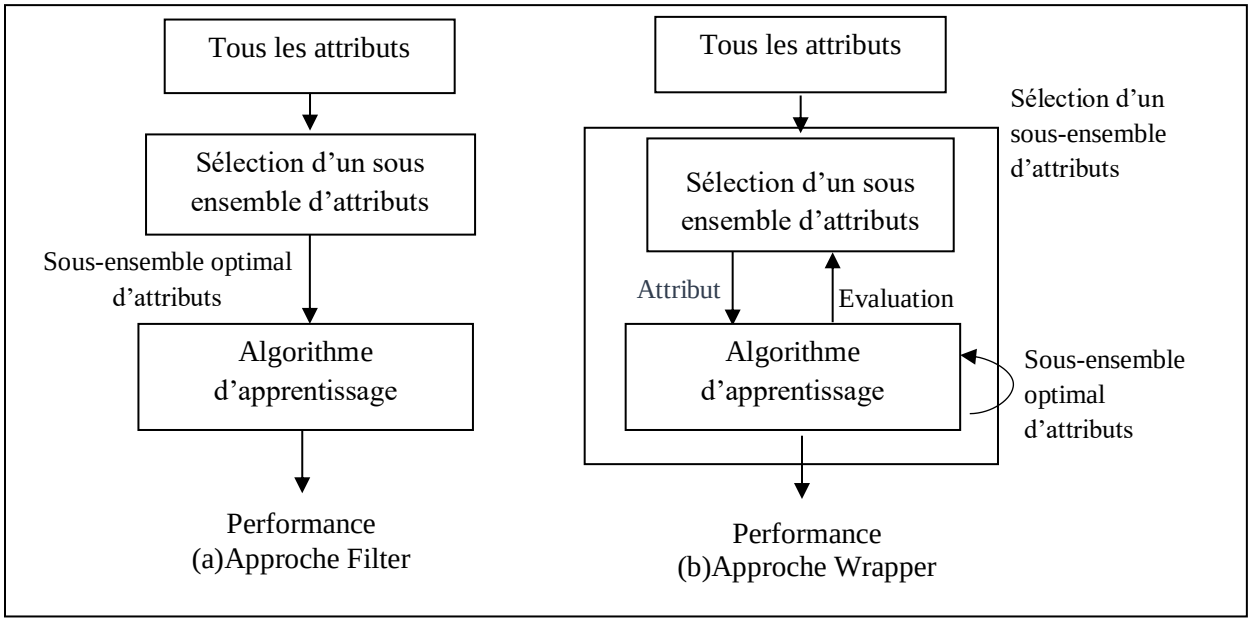


Figure I.9 : *Comparaison entre l'approche enveloppante et Filtre.*

6 Méthodes classiques de sélection d'attributs :

6.1 Méthodes utilisés par l'approche Filtre :

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.

6.1.1 Focus :

a) Principe : FOCUS est une méthode de type Stepwise Forward Sélection qui est présenté par Almuallim et Dietterich en 1991 et implique un haut degré de recherche à travers l'espace [29]. Elle 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. [14]

b) Algorithme : L'algorithme de cette méthode est originalement défini pour des données booléennes non bruitées, recherche la combinaison minimale d'attributs qui

discrimine parfaitement les classes, c'est ce que l'on appelle l'ensemble minimum de variables[8] . Il est clair que le nombre de sous-ensembles d'attributs évalués est trop grand.

6.1.2 Branch and Bound (BB) :

a) Principe : est une méthode développé par Narendra et Fukunaga en 1977 qui commence avec l'ensemble initial des variables (backward elimination) et cherche ensuite la variable à retirer en maximisant la fonction d'évaluation. Elles sont basées sur l'hypothèse que la fonction d'évaluation adoptée remplie la condition de monotonie.

b) Définition (Fonction Monotone) : Une fonction possède la propriété de monotonie si elle considère qu'un sous-ensemble de variables n'est pas meilleur qu'un ensemble plus grand contenant ce sous-ensemble. [29]

L'utilisation de l'hypothèse de monotonicité dans la recherche du sous-ensemble optimal signifie que l'ajout de nouveaux attributs à un sous-ensemble d'attributs ne peut jamais diminuer la fonction d'évaluation.[14]

c) Algorithme : L'algorithme commence avec l'ensemble total d'attributs et enlève un attribut à la fois. L'algorithme garantit d'arriver à la solution optimale sans parcourir tous les sous-ensembles de variables Il est inefficace pour réduire l'espace de recherche.

6.1.3 Automatic Branch & Bound (ABB) :

a) Principe : ABB (Automatic Branch & Bound) 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.

b) Algorithme : Elle commence par l'ensemble complet d'attributs et supprime un attribut, à chaque fois, dans un algorithme de recherche en largeur d'abord (breadth- first search) jusqu'à ce qu'aucun attribut ne puisse être enlevé alors que le critère de l'incohérence est respecté.

[8] Min-Feature Bias en anglais.

6.1.4 LVF (Las Vigas Filter) :

a) Principe : C'est un algorithme probabiliste, pour la sélection d'attributs. LVF fait des choix probabilistes de sous-ensembles dans la recherche de l'ensemble optimal. Il est rapide pour faire décroitre le nombre d'attributs [30] .Il est particulièrement efficace pour des ensembles de données ayant des attributs redondants . Cette méthode peut trouver le sous-ensemble optimal même pour les ensembles de données bruitées si le niveau de bruit est spécifié au début.

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. 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. [14]

6.2 Méthodes utilisés par l'approche enveloppante :

6.2.1 SFS (Sequential Forward Selection) :

SFS est pour la recherche en avant. Il consiste à ajouter les attributs un à un à partir d'un ensemble vide d'attributs, en prenant à chaque étape celui qui permet de maximiser un critère J. [30], 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. Le principal inconvénient de SFS est qu'il est incapable de supprimer des attributs qui deviennent impertinents après l'ajout d'autres attributs.[14]

6.2.2 SBS (Sequential Backward Selection) :

SBS est pour la recherche en arrière. Cet algorithme part de l'ensemble complet d'attributs, et les supprime un à un, à chaque étape. Il supprime celui dont l'absence permet de maximiser un critère J. L'algorithme SBS est performant et conduit vite à des optima locaux. Mais les premières itérations de SBS sont très couteuses.

6.2.3 Les Métaheuristiques pour la sélection d'attributs :

L'approche enveloppante utilise les métaheuristiques comme des techniques pour la sélection d'attributs. La plupart des métaheuristiques existantes ont été utilisées dans le processus de recherche de sous ensemble d'attributs.

7 Métaheuristiques de Sélection d'attributs :

L'optimisation est un aspect fondamental de l'ingénierie et de la résolution de problèmes. Nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation. Par conséquent, nombreuses méthodes de résolution ont été développées en recherche opérationnelle (RO) et en intelligence artificielle (IA). Ces méthodes peuvent être classées sommairement en deux grandes catégories [15] :

- **Les méthodes exactes** (complètes) qui garantissent la complétude de la résolution.
- **Les méthodes approchées** (incomplètes) qui perdent la complétude pour gagner en efficacité. Depuis une dizaine d'années, des progrès importants ont été réalisés avec l'apparition d'une nouvelle génération de méthodes approchées puissantes et générales, souvent appelées *métaheuristiques.*

Avant d'aborder les métaheuristiques de sélection d'attributs, nous donnerons quelques notions de base.

Définition5 (Optimisation combinatoire) : Un problème d'optimisation combinatoire est soit un problème de minimisation, soit un problème de maximisation d'une fonction donnée, et est composé par un ensemble d'instances. [18] Il existe de nombreux problèmes d'optimisation pour lesquels les variables indépendantes sont limitées à un ensemble de valeurs discrètes. Ces problèmes sont appelés problèmes d'optimisation combinatoire ou d'optimisation discrète.

Définition6 (Heuristique) : Une heuristique[9] est une méthode, conçue pour un problème d'optimisation donné, qui produit une solution non nécessairement optimale lorsqu'on lui fournit une instance de ce problème. [18]

[9] Le mot heuristique vient du grec ancien eurisko ('je trouve') et qualifie tous qui sert à la découverte, à l'invention et à la recherche. [11].

Définition7 (Métaheuristique) : Une métaheuristique[10] est définie comme un processus itératif qui guide une heuristique subordonnée en combinant intelligemment différents concepts pour explorer et exploiter l'espace de recherche, des stratégies d'apprentissage sont utilisées pour structurer des renseignements pour trouver des solutions efficaces presque optimales.[18]

7.1 Intensification et diversification : [16]

Le principe d'intensification et de diversification est un point critique pour toute métaheuristique, il consiste à trouver un compromis entre les deux tendances duales suivantes :

- Il s'agit d'une part d'intensifier l'effort de recherche vers les zones les plus prometteuses de l'espace de solutions.
- Il s'agit d'autre part de diversifier l'effort de recherche de façon à être capable de découvrir de nouvelles zones contenant de meilleures combinaisons.

7.2 Classification des métaheuristiques :

Une manière de classifier les métaheuristiques est de distinguer celles qui travaillent avec une population de solutions de celles qui ne manipulent qu'une seule solution à la fois. La figure I.10 suivante donnera un panorama des méthodes les plus utilisées.

[10] Une métaheuristique est définie de manière similaire à une heuristique, mais à un niveau d'abstraction plus élevé. Elle est constituée d'un ensemble de concepts fondamentaux qui permettent d'aider à la conception de méthodes.

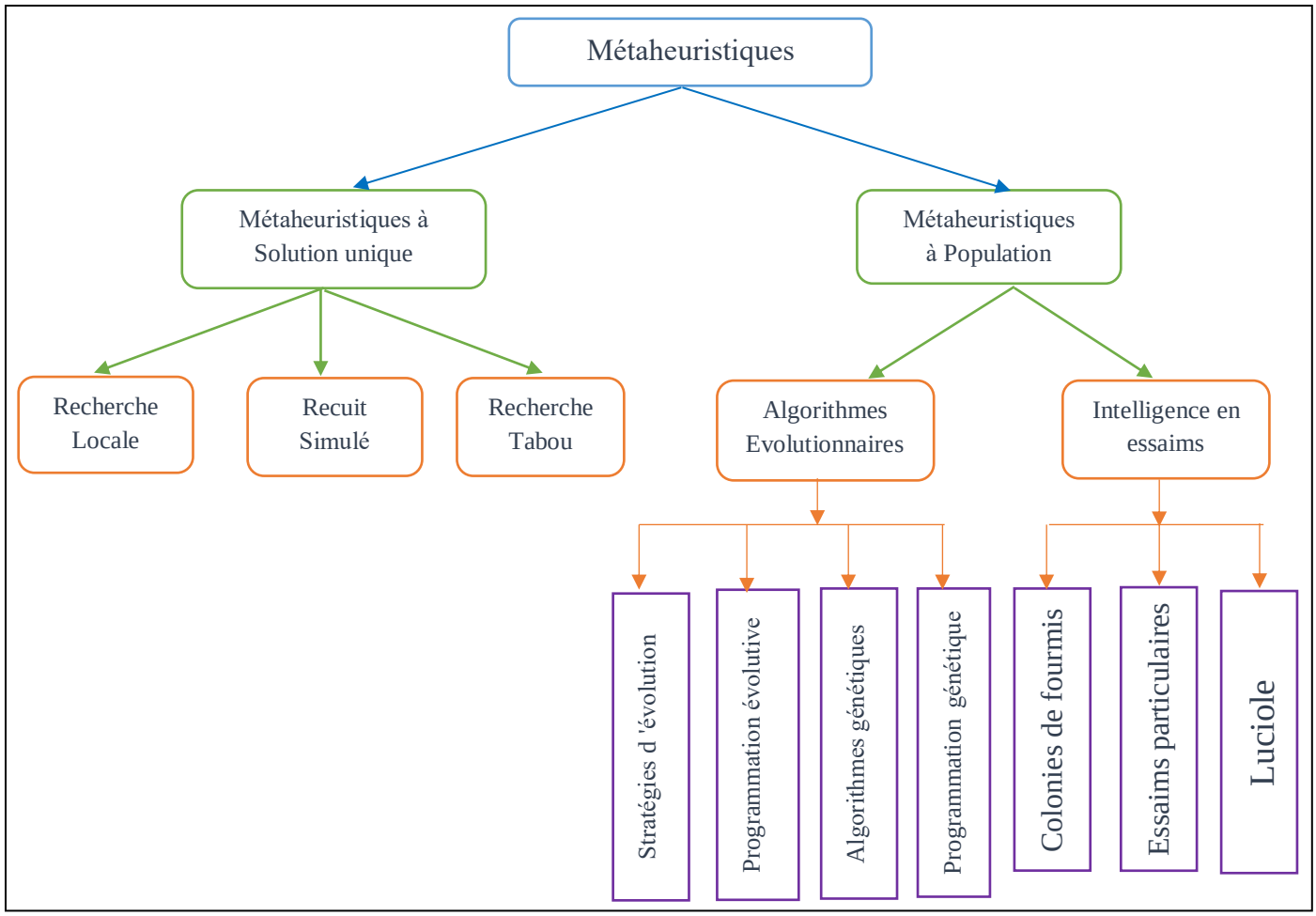


Figure I.10 : *Classification des métaheuristiques*

Dans le cadre de notre travail, nous présentons l'intelligence en essaims (branche droite de la Figure I.10) et nous nous intéressons particulièrement à l'algorithme de luciole qui est un paradigme récent de l'intelligence informatique.

7.2.1 Les métaheuristiques à solution unique :

Les méthodes qui tentent itérativement d'améliorer une solution sont appelées méthodes de recherche locale ou méthodes de trajectoire. La méthode Tabou, le Recuit Simulé et la Recherche à Voisinages Variables sont des exemples typiques de méthodes de trajectoire. Ces méthodes construisent une trajectoire dans l'espace des solutions en tentant de se diriger vers des solutions optimales. [16]

7.2.2 Les métaheuristiques à population :

Cette catégorie regroupe les méthodes qui travaillent avec une population de solutions. La première classe d'approches évolutionnaires contient les méthodes classiques telles que les algorithmes génétiques, les stratégies d'évolution, la programmation évolutive ainsi que la programmation génétique. La deuxième classe inclut les méthodes évolutionnaires récentes issues de l'intelligence en essaims (swarm intelligence). Nous nous intéressons plus particulièrement de l'intelligence en essaims.

7.2.2.1 La première classe : Approches évolutionnaire classique :

Les méthodes évolutionnaires représentent une classe de métaheuristique ayant comme principe de faire évoluer une population de base constituée de solutions au problème à résoudre.

Un algorithme évolutif typique est composé de trois éléments essentiels [15] :

- **une population** constituée de plusieurs individus représentant des solutions potentielles (configurations) du problème donné.
- **un mécanisme d'évaluation** de l'adaptation de chaque individu de la population à l'égard de son environnement extérieur.
- **un mécanisme d'évolution** composé d'opérateurs permettant d'éliminer certains individus et de produire de nouveaux individus à partir des individus sélectionnés.

Différents types d'algorithmes évolutionnaires ont été développés au début des années soixante :

- **Stratégies d'Evolution (SE) :** Cette famille de méthodes est chronologiquement la première parmi tous les algorithmes évolutionnaires. Les (SE) ont été développés par deux ingénieurs Rechenberg et Schwefel au cours de leurs travaux de recherches sur des problèmes numériques pour l'optimisation paramétrique.
- **Programmation Évolutionnaire(EP):**Un an après l'apparition des stratégies d'évolution ,cette famille voit le jour et a été développée aux années 60 par L. Fogel dans le cadre de nouveaux concept d'apprentissage en intelligence artificielle , pour la création d'automates à états finis, puis utilisé plus tard par D.B. Fogel aux années 91.

- **Algorithmes Génétiques (AG) :** Ce sont les algorithmes évolutionnaires les plus connus et les plus répondus les (AG), développées par J. Holland en 75 comme outils de modélisation de l'adaptation et qui travaillent dans un espace de chaînes de bits.
- **Programmation Génétique (PG) :** C'est la dernière famille d'algorithmes évolutionnaires à avoir vu le jour et a été développée par J. Koza en 90 dans le but d'atteindre un des rêves des programmeurs : faire écrire un programme automatiquement. La représentation est basée sur des arbres représentant les programmes.

7.2.2.2 La deuxième classe : Approches évolutionnaire récentes issues de l'intelligence en essaims :

Dans ces dernières années, l'intelligence en essaims avait gagné une grande popularité. Il offre une grande diversité de comportements et de morphologies, Cette diversité est une mine d'inspiration fascinante pour les systèmes informatiques.

a) Définition (l'intelligence en essaims) : Toute tentative de concevoir des algorithmes ou des dispositifs de résolution des problèmes, distribuées, inspirés des comportements collectifs des insectes sociaux ou d'autres sociétés animales. [27]

b) Notion de bases sur L'intelligence en essaims :

- L'intelligence en essaims étudie le comportement collectif des systèmes composés de nombreux individus (fourmis, abeilles, termites, oiseaux ou poissons) qui interagissent localement entre eux et avec leur environnement.
- Un algorithme d'intelligence en essaims est fondé sur le comportement des essaims qui existent dans la nature (par exemple, des essaims d'insectes ou d'oiseaux).
- Les chercheurs en informatique ont développé des systèmes basés sur les essaims, suite à la réussite et l'efficacité observée de différents essaims pour résoudre des problèmes difficiles dans la nature.
- L'essaim complet présente un comportement intelligent, en fournissant des solutions efficaces pour les problèmes complexes tels que la recherche de plus court chemin. L'intelligence en essaims est basée sur l'étude du comportement collectif dans des systèmes auto-organisés décentralisées.

c) Catégories de L'intelligence en essaims :

L'optimisation par colonies de fourmis et par les essaims de particules sont les premiers algorithmes de l'intelligence en essaims. D 'autre méthodes plus récente commencent néanmoins à percer, comme les algorithmes inspirés des abeilles, l'algorithme des essaims de poissons artificiels (Artificial Fish Swarm Algorithm ou AFSA), L'algorithme des chauves-souris (Bat algorithm), l'algorithme d'optimisation basé sur l'exploration bactérienne et l'algorithme des essaims de lucioles (Firfly algorithm) qui sera présenté en détail dans le prochain chapitre.

8 Formalisation de notre Problématique :

Notre problématique est divisée en deux parties :

a) Le problème de sélection d'attributs : est un problème d'optimisation combinatoire mono-objectif de maximisation (S, f) où

- S est l'ensemble des attributs.
- Un sous-ensemble X de S représente l'ensemble de solutions.
- f représente la fonction à maximiser, c'est la fonction objective qui assigne à chaque solution $s \in X$ le nombre réel (ou entier) $f(s)$.

Notre objectif est de de trouver une solution $s^* \in X$ optimisant la valeur de la fonction objectif f. Une telle solution s^* s'appelle une *solution optimale.*

b) Le problème de classification : rentre dans le cadre de l'apprentissage statistique supervisé. Ce problème est également dénommé problème de décision en statistiques, ou de reconnaissance des formes. [12].

Notre objectif est de classer les données d'apprentissage en plusieurs classes, donc il faut prévoir la classe ***y*** d'un vecteur N-dimensionnel **X***(N est la dimension du problème=le nombre d'attributs)***,** en se basant sur les mesures des variables qui l'expliquent avec pour seule information celle contenue dans un exemple d'apprentissage **E**. Il s'agit d'une discrimination Multi-classe. Nous supposons dans notre cas que les données sont des couples $(\mathbf{x_i}, \mathbf{y_i})_{\mathbf{1 \le i \le l}} \in \mathbf{X} * \mathbf{Y}$ où X (ensemble de **L** exemples) désigne l'espace des variables explicatives souvent pris dans R^N ,Y={-1 ,+1 ,…..} .Chaque donnée(exemple) $\mathbf{x_i} \in \mathbf{X}$ est caractérisée par **N** attributs et par sa classe $\mathbf{yi} \in \mathbf{Y}$.

Dans un problème de classification supervisée, la classe prend sa valeur parmi un ensemble **Y** fini. L'appartenance d'une observation (exemple) xi à une classe ou à une autre est matérialisée ici par la valeur de son étiquette $\mathbf{y_i}$. L'échantillon d'apprentissage E est souvent dénoté par [9] :

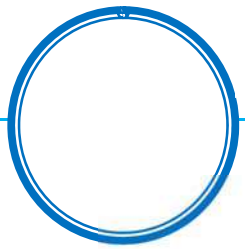
$$\mathbf{E= \{(x_1, y1), (x_2, y2) \ldots. (Xl, yl)\}} \in (\boldsymbol{X} * \boldsymbol{Y})^{\boldsymbol{l}}$$

Le problème consiste alors, à prédire la classe $\mathbf{y_i}$ de toute nouvelle donnée. En termes de vocabulaire, on utilise le mot « étiquette » comme synonyme de « classe ». Et on dit classification binaire quand le nombre de classes **Y** est égal à 2.

9 Conclusion :

La sélection d'attributs ou de caractéristiques est considérée comme un problème d'optimisation qui a intéressé beaucoup de chercheurs. Les métaheuristiques à population se sont révélées d'être des méthodes puissantes pour résoudre des problèmes d'optimisation réputés difficiles.

Dans ce chapitre nous avons présenté les concepts de base et les méthodes les plus utilisées dans le domaine de la sélection d'attributs. Ensuite, nous avons présenté les différentes approches évolutionnaires classiques et récente qui sont largement appliquées pour ce problème.



CHAPITRE II

L'Algorithme De Luciole & Les Séparateurs à Vaste Marge

1 Introduction

Les techniques d'optimisation bio-inspirées ont obtenu une grande attention au cours des dernières années en raison de sa robustesse, simplicité et efficacité pour résoudre des problèmes d'optimisation complexes.

Actuellement, un ensemble de métaheuristiques bio-inspirées basées sur le comportement naturel des essaims, des abeilles, des oiseaux, et des fourmis avaient émergé comme une alternative pour surmonter les difficultés présentées par les méthodes classiques dans l'optimisation.

L'algorithme d'optimisation de luciole (FireFly Algorithm ou FA ou FFA) est une métaheuristique très récente inspiré du comportement de clignotement de lucioles. Dans cet algorithme, les solutions générées aléatoirement seront considérés comme des lucioles, et la luminosité est attribuée en fonction de leur performance sur la fonction objective.

2 Algorithme des lucioles pour la sélection d'attributs

2.1 Inspiration :

Les lucioles (en anglais Fireflies) sont de petits coléoptères ailés capables de produire une lumière clignotante froide pour une attraction mutuelle. Les femelles peuvent imiter les signaux lumineux d'autres espèces afin d'attirer des mâles qu'elles capturent et dévorent. Les lucioles ont un mécanisme de type condensateur, qui se décharge lentement jusqu'à ce que certain seuil est atteint, ils libèrent l'énergie sous forme de lumière. Le phénomène se répète de façon cyclique [8].

2.2 Principe de l'Algorithme :

L'algorithme d'optimisation basé sur les essaims de lucioles appelé aussi FireFly Algorithm est le plus récent métaheuristique qui a été développé par l'auteur Xin-She Yang dans [1] [2] [3] [4] [5] [6] à l'université de Cambridge en 2007 (Yang 2008, Yang 2009).

Il est inspiré par l'atténuation de la lumière sur la distance et basé sur le comportement idéale des caractéristiques de clignotement des lucioles et l'attraction mutuelle mais il considère toutes les lucioles comme unisexes il utilise les trois règles idéalisées suivantes cité dans [1] [4] [3] [5] :

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 à optimiser. 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.

2.3 Paramètres de l'Algorithme :

a) La luminosité I : la variation de la luminosité I avec la distance r est définit par

$$\boldsymbol{I = I_0\, e^{-\gamma r 2}} \qquad \text{(II.1)}$$

Où I_0 est la luminosité initiale des lucioles.

b) L'attractivité β, et le coefficient d'absorption γ : [7]

Comme l'attractivité est proportionnelle à la luminosité des lucioles adjacentes alors la variation de l'attractivité β avec la distance r est définit par

$$\boldsymbol{\beta = \beta_0\, e^{-\gamma r 2}} \qquad \text{(II.2)}$$

Où β_0 est l'attractivité à r = 0.

c) La distance r et la dimension d : [3] [8]

La distance r_{ij} entre 2 lucioles est déterminée par la formule suivante :

$$\boldsymbol{r_{ij} = \left\| X_i - X_j \right\| = \sqrt{\sum_{k=1}^{d} \left(X_k^i - X_k^j\right)^2}} \qquad \text{(II.3)}$$

Où x_i^k est le k ième composant des coordonnées la de i ème luciole et d est le nombre de dimensions.

d) Calcul des mouvements : [2]

Le mouvement d'une luciole X_i à une autre luciole X_j plus attirante est calculée via la formule suivante :

$$\boldsymbol{X_i = X_i + \beta_0 e^{-\gamma r_{ij}^2}\left(X_j - X_i\right) + \alpha\left(rand - \frac{1}{2}\right)} \qquad \text{(II.4)}$$

Où le premier et le deuxième paramètre sont liés à l'attractivité et le troisième paramètre est une randomisation. α est le paramètre aléatoire et peut être constant. " rand " est un générateur de nombre aléatoire uniformément distribuée dans [0, 1].

e) Le nombre de Fireflies et le nombre d'itérations :

La figure II.1 montre les positions initiales de 25 fireflies (droite) et la position finale après 20 itérations.

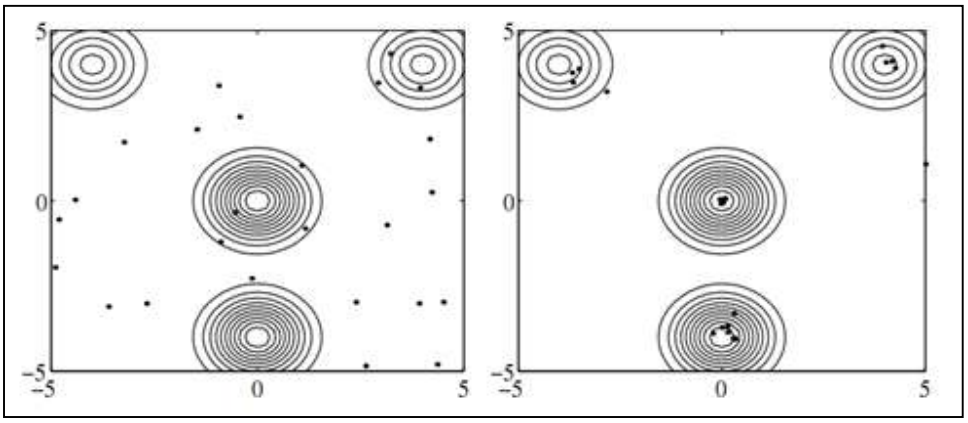


Figure II.1 : *positions de 25 fireflies.* [4]

2.4 L'Algorithme de FireFly : [8]

PROCEDURE FIREFLY (NBR_ITER : NOMBRE D'ITERATIONS)

Début
Générer une population initiale de lucioles x_i (i =1..., n) ;
Déterminer les intensités de lumière I_i à x_i via f (x_i) ;
Tant que (t<=nbr_iter)
Faire
pour i = 1 à n //toutes les lucioles
pour j=1 à n //toutes les lucioles
si (I_j> I_i) **alors** - calculer l'attractivité β_{ij} selon la distance r_{ij}
- déplacer luciole x_i vers x_j avec l'attractivité β_{ij}
sinon déplacer x_i aléatoirement
fin si ;
évaluer la nouvelle solution ;
mettre à jour l'intensité I_i ;
vérifier si luciole x_i est la meilleur ;
fin j ;
fin i ;
Trouver la meilleure luciole en fonction d'objective ;
t++ ;
Fait ;
Fin.

2.5 L'Algorithme de luciole binaire :

A la recherche des solutions optimales, les lucioles font des mouvements et par conséquent changent de positions. Ces nouvelles positions sont calculées en utilisant la formule (II.4) pour chaque dimension de luciole et le résultat est un nombre réel. La conversion des positions vers le binaire consiste à suivre les étapes suivantes :

- Générer un nombre *Rand* aléatoire entre 0 et 1.
- Utiliser la fonction Sigmoïd pour calculer le nombre T (x_i(k)) en utilisant la position réel via la formule suivante [17] :

$$\boldsymbol{T(x_i(k)) = \frac{1}{1+e^{-x_i(k)}}} \quad \text{(II.5)}$$

- Calculer la nouvelle position x_i(k) en binaire de la manière suivante [17] :

$$\boldsymbol{x_i(k) = \begin{cases} x^*(k), Rand < T(x_i(k)) \\ 0, sinon \end{cases}} \quad \text{(II.6)}$$

Où x_i(k) est le k ième composant des coordonnées de la i ème luciole et x* est la solution optimale.

2.6 Cause de l'efficacité de FA :

L'application de l'algorithme de firefly a donné des bons résultats presque dans tous les domaines où il a été appliqué, la question qui se pose pourquoi FA est efficace ? Il y a plusieurs causes pour son réussite. Par l'analyse faite dans [5] des caractéristiques du FA standard/classique. On peut conclure les points suivants :

- Selon l'attractivité des lucioles (l'attractivité locale est plus forte que l'attractivité à long distance).
- FA peut automatiquement subdiviser sa population en sous-groupes et par conséquent, le FA peut résoudre efficacement des grands problèmes d'optimisation.

2.7 Les applications de l'algorithme FireFly : [3]

L'algorithme FA a été appliqué pour résoudre plusieurs problèmes d'optimisation, de classification, et d'engineering dans la pratique. La taxonomie de développement de l'application de FA est illustrée dans la Figure II.2.

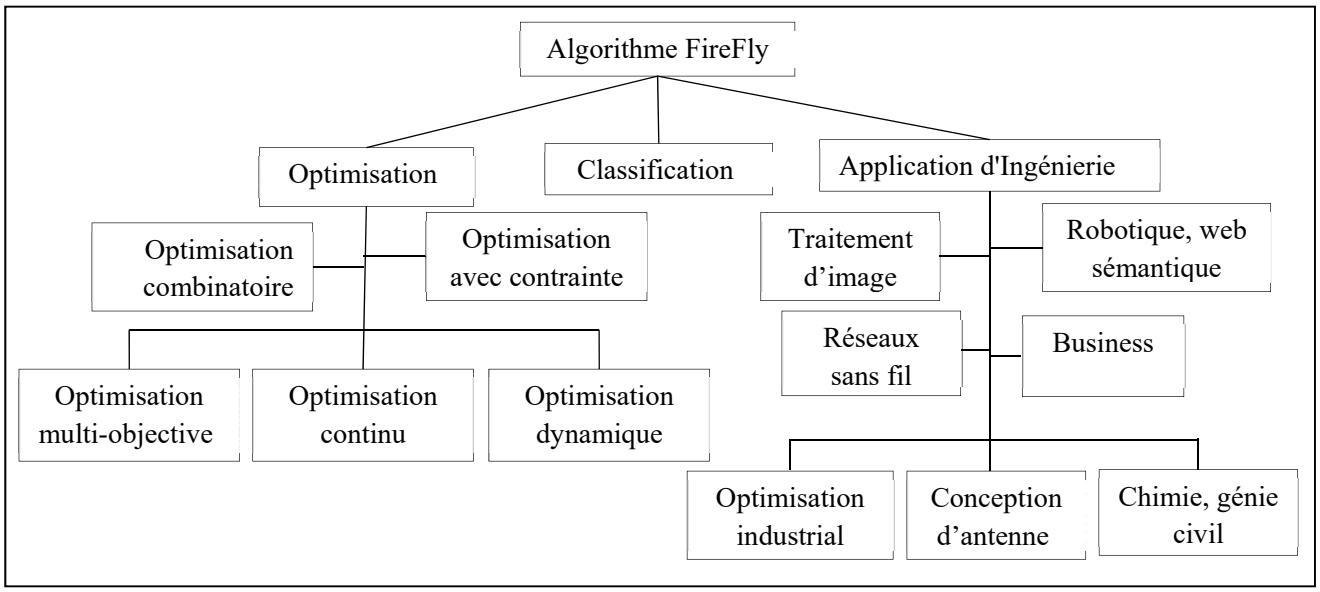


Figure II.2 : *Taxonomie de l'application de Firefly [3]*

A partir de cette figure on peut dire que FA a été appliqué pour résoudre les classes des problèmes d'optimisation suivantes : continu, combinatoire, avec contrainte, multi-objective et optimisation dynamique. En plus il a été utilisé pour résoudre des problèmes de classification : apprentissage automatique, datamining et réseaux de neurones. Finalement FA est utilisé dans la plupart des branches de l'engineering.

3 Les Séparateurs à Vaste Marge pour la classification supervisée :

L'objectif des méthodes de classification est d'identifier les classes auxquelles appartiennent des objets à partir de certains paramètres descriptifs. Elles s'appliquent à un grand nombre d'activités humaines et conviennent en particulier au problème de la prise de décision automatisée. La procédure de classification sera extraite automatiquement à partir d'un ensemble d'exemples. Un exemple consiste en la description d'un cas avec la classification correspondante.

Un système d'apprentissage doit alors, à partir de cet ensemble d'exemples, extraire une procédure de classification, il s'agit en effet d'extraire une règle générale à partir des données observées. La procédure générée devra classifier correctement les exemples de l'échantillon et avoir un bon pouvoir prédictif pour classifier correctement de nouvelles descriptions. Les méthodes utilisées pour la classification sont nombreuses, citons : la méthode des Séparateurs à Vastes Marges (SVM), les Réseaux de Neurones, etc.

Nous présentons dans la suite de ce chapitre la méthode de classification SVM.

3.1 Définition de la méthode SVM :

Les Séparateurs à Vaste Marge appelés aussi Les machines à vecteurs de supports (SVMs) ont été introduits dès 1992 et peuvent être utilisés pour résoudre des problèmes de discrimination, définir à quelle classe appartient un individu, et des problèmes de régression, c'est-à-dire prédire la valeur numérique d'une variable. Les SVM ont originellement été développées pour la reconnaissance de la forme ou la classification. Dans ces domaines, elles se sont rapidement imposées.

Le succès de cette méthode est justifié par les solides bases théoriques qui la soutiennent. Elles permettent d'aborder des problèmes très divers dont la classification. SVM est une méthode particulièrement bien adaptée pour traiter des données de très haute dimension.

3.2 Bref historique des SVMs :

- En 1970 Les bases théoriques des SVM sont fondées par Vapnik et ses collègues avec le développement de la Théorie de Vapnik-Chervonenkis, et par Valiant et la théorie de l'apprentissage PCA. Il s'agit de deux notions fondamentales : la notion de marge maximale et la notion de fonction noyau.
- Dès 1970 La notion d'hyper plan à marge maximale fut explorée par Vladimir Vapnik et A.lerne.
- En 1909, le théorème de Mercer traite les fonctions noyaux, leur application à l'apprentissage artificiel fut montrée en 1967 par Aizermann, Bravermann et Rozoener.
- En 1992 les idées précédentes sont rassemblées par Boser, Guyon et Vapnik dans un article qui est l'article fondateur des SVM.
- En 1995, Corte et Vapnik introduisent un classificateur à marge souple et l'algorithme fut étendu au cas de régression dans la même année par Vapnik.

3.3 Principe de fonctionnement :

Le principe générale des SVM est la construction d'une fonction f appelée fonction de décision qui a un vecteur d'entrée x fait correspondre une valeur y,

$$\boldsymbol{y = f(x)} \qquad \text{(II.7)}$$

Où x représente l'exemple à classer et y la classe à laquelle correspond l'exemple en entrée. Comme tout problème de classification, cette fonction fait appel à un jeu de données d'apprentissage pour apprendre les paramètres du modèle.

Les SVMs sont des classifieur qui reposent sur deux idées clés : la notion de marge maximale et la notion de fonction noyau.

- La première idée clé est la notion de marge maximale. On cherche l'hyperplan qui sépare les exemples positifs des exemples négatifs, en garantissant que la distance entre la frontière de séparation et les échantillons les plus proches (marge) soit maximale, ces derniers sont appelés vecteurs supports. et comme on cherche à maximiser cette marge, on parlera de séparateurs à vaste marge.
- La deuxième idée clé est la notion de fonction noyau (kernel). Il s'agit de transformer l'espace de représentation des données d'entrées en un espace de plus grande dimension appelée espace de caractéristiques, dans lequel il est probable qu'il existe une séparatrice linéaire. Afin de pouvoir traiter des cas où les données ne sont pas linéairement séparables.

3.4 Notions de base sur SVM :

Définition1 (Hyperplan ou Hyperplan valide) : Un hyperplan est un classificateur qui va séparer les exemples positifs des exemples négatifs dans l'ensemble des exemples d'apprentissage.

Définition2 (Hyperplan optimal) : Un hyperplan optimale est un hyperplan valide qui maximise la marge et passe ≪ au milieu ≫ des points des deux classes d'exemples.

Définition3 (marge) : une marge est la distance minimale entre l'hyperplan et les exemples d'apprentissage et exactement la distance entre l'hyperplan et les vecteurs de support.

Définition4 (vecteurs de support) : Les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support. Ils représentent les vecteurs discriminant qui sont utilisés pour classer un nouveau cas.

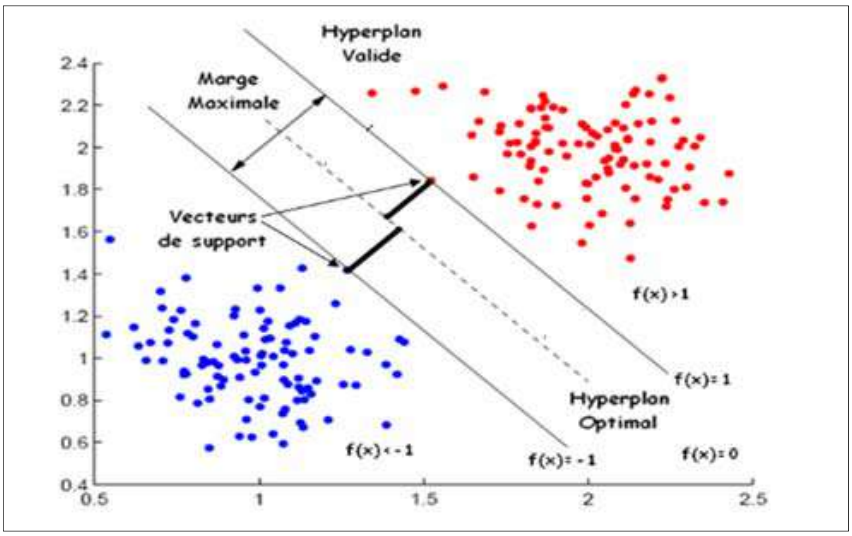


Figure II.3 : *Nuage de points à 2 dimensions avec un hyperplan optimal*

Il existe une infinité d'hyperplans capable de séparer parfaitement les deux classes d'exemples. Le principe des SVM est de choisir celui qui va maximiser la marge.

3.5 Maximisation de la marge :

La maximisation de la marge permet d'assurer la bonne classification des nouveaux exemples par un classifieur qui se comporte le mieux vis-à-vis des données d'apprentissage.

Dans le schéma (figureII.4 et figure II.5), la partie droite nous montre qu'avec un hyperplan optimal, un nouvel exemple reste bien classé alors qu'il tombe dans la marge. On constate sur la partie gauche qu'avec une plus petite marge, l'exemple se voit mal classé.

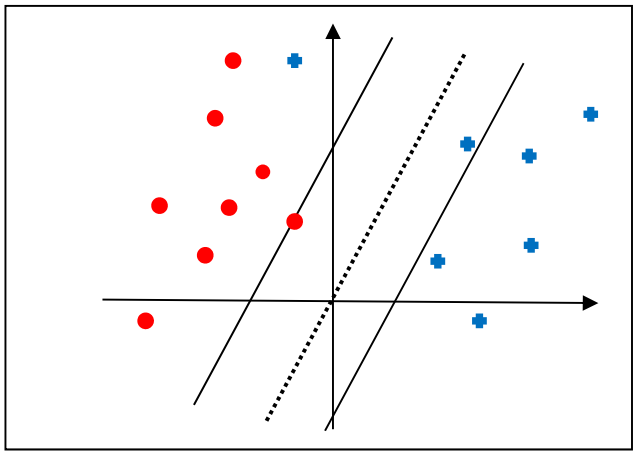
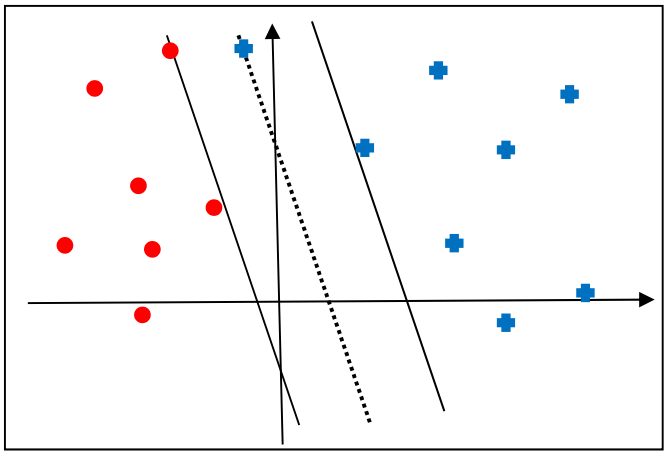


Figure II.4 : *Hyperplan avec faible marge*



FigureII.5 : *Meilleur Hyperplan séparateur*

3.6 SVM à deux classes :

La classification binaire est l'un des problèmes de discrimination les plus simple. Selon la distribution spatiale des individus, il y a deux cas de classifications binaire, nous expliquons chacun d'eux par le biais d'exemples simple et intuitifs. Parmi les modèles des SVM, on constate les cas linéairement séparables et les cas non linéairement séparables.

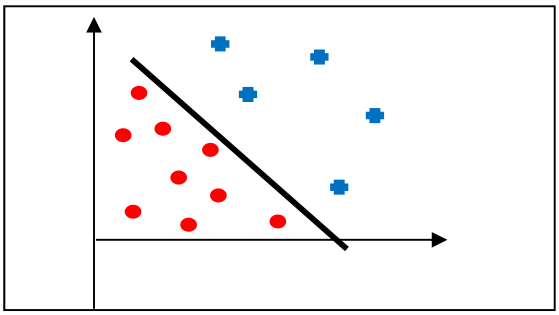


Figure II.6 : Cas *linéairement séparable*

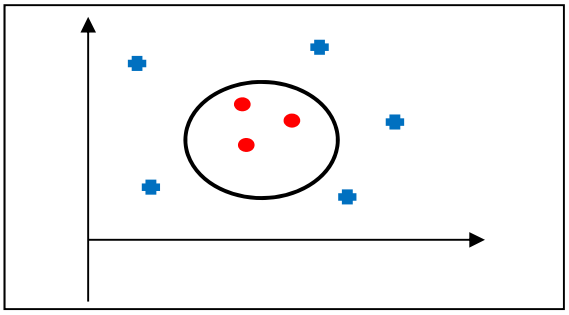


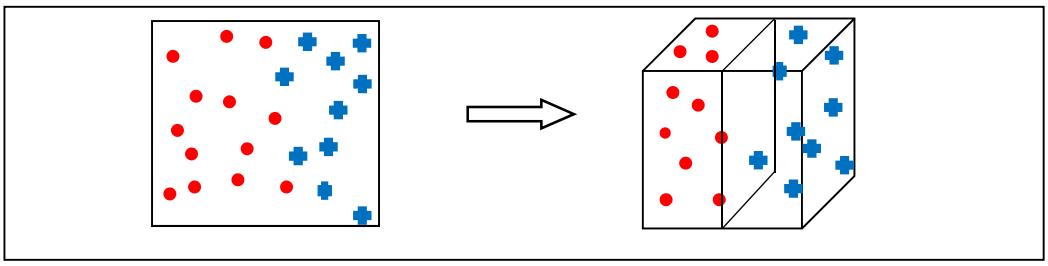
Figure II.7 : *Cas non linéairement séparable*

3.6.1 Cas des données linéairement séparable :

Ce sont les plus simples des SVM car ils permettent de trouver facilement le classificateur linéaire de marge maximale. Il s'agit de trouver un hyperplan (H) séparant les données d'apprentissage en deux populations, de telle manière que les exemples appartenant à la même classe se trouve d'un même côté de l'hyperplan.

3.6.2 Cas des données non linéairement séparable :

Il est clair que les frontières de décision linéaires sont très limitées pour résoudre tous les problèmes de classification. Pour que le classifieur SVM trouve des frontières de décision non linéaires l'idée est l'extension des SVMs au cas non-linéaire qui est assez simple et



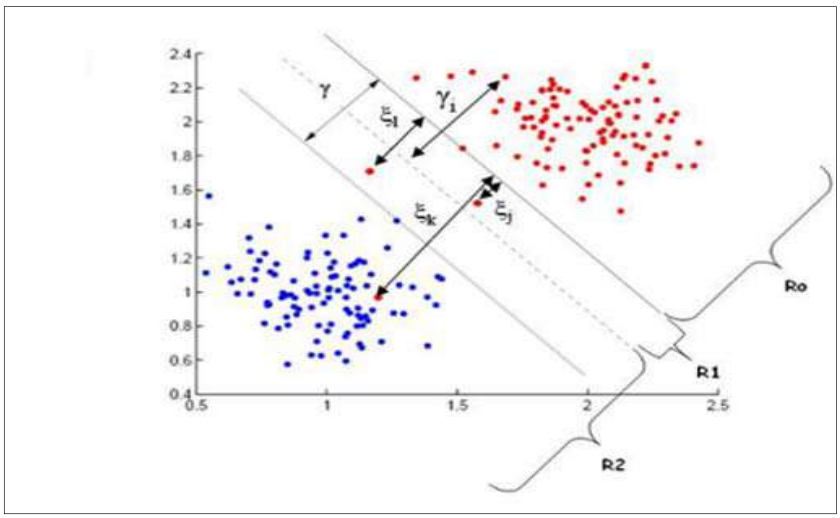
FigureII.8 : *Projection des données dans un espace de dimension supérieure*

repose sur la projection des données dans un espace de dimension supérieure dans lequel le problème devient linéaire. Cette transformation non linéaire est réalisée via une fonction noyau. Ceci est représenté par le schéma suivant :

3.6.3 C-SVM : SVM à marge douce et le paramètre C :

Dans des problèmes réels, il est rare de tomber sur des données linéairement séparables. Des outliers (points mal classifiés) existent toujours, quel que soit l'hyperplan considéré. Nous avons besoin donc d'un modèle plus robuste, qui soit capable de tolérer les outliers, quand les données sont plus ou moins bruitées. L'idée est d'introduire un terme d'erreur qui permet parfois à un exemple de passer à l'autre côté de l'hyperplan. Pour cela on introduit des variables "lâches" $\zeta 1 \ldots \zeta n >= 0$.

Un $\zeta i > 0$ correspond à une erreur dans la zone marge. Une erreur de classification (ie, un point passé à l'autre côté de l'hyperplan) se produit si $\zeta i > 1$. Ainsi, $\sum \zeta i$ est un majorant du nombre d'erreurs sur l'ensemble d'apprentissage. [23]



FigureII.9: *Données dans le cas non séparables* ε *variable d'écart à la marge objective* γ *qui est un réel strictement positif.*

$\zeta_i = \max(0, \gamma-\gamma_i)$, $\zeta_i > \gamma \Leftrightarrow$ (x_i, y_i) est mal classé

Soit x^+_j une observation de la classe positive :

$$\begin{cases} \text{Si } x_j^+ \in R0 \Leftrightarrow \zeta j = 0 \\ \text{Si } x_j^+ \in R_1 \Leftrightarrow 0 < \zeta j < \gamma \\ \text{Si } x_j^+ \in R2 \Leftrightarrow \zeta j >= \gamma \end{cases} \quad \text{(II.8)}$$

Paramètre C : La constante **C** est un hyper paramètre qui doit être choisit en fonction de la connaissance du bruit sur les données et qui représente le compromis entre la maximisation de la marge et la minimisation de l'erreur sur les données d'apprentissage.

- Plus C est grand plus en pénalise les erreurs, plus les variables de relâchement ζ_i prennent de faibles valeurs, et plus la marge de tolérance aux erreurs est petite.
- Plus C est petit, plus la marge de tolérance aux erreurs est grande, il y aura plus des exemples qui tombent dans la zone entre les deux hyperplans séparateurs.
- Dans le cas extrême où C tend vers 0, la marge devient si grande que tous les points tombent dans la zone de marge.

3.6.4 Fonction noyau RBF et le paramètre σ :

a. Définition : Le noyau est une fonction k qui associe à tout couple d'observations *(x_i ; x_j)* une mesure de leur « influence réciproque » calculée à travers leur corrélation ou leur distance. Des exemples typiques de noyaux sont [24] :

- *Noyau polynomial d'ordre p :*

$$K(x_i, x_j) = (\alpha + \langle x_i, x_j \rangle)^p \quad \text{(II.9)}$$

- *Noyau linéaire :* $K(x_i, x_j) = \langle x_i . x_j \rangle$ (correspondant aux SVM linéaires).
- *Le noyau radial de Laplace* (Laplace RBF) :

$$K(x ; x') = \exp(-\partial||x-x'||). \quad \text{(II.10)}$$

- *Le noyau tangente hyperbolique* (sygmoïd) :

$$K(x ; x') = \tanh(\alpha + \beta\langle x, x' \rangle). \quad \text{(II.11)}$$

- *Noyau gaussien de largeur de bande ∂ (gaussien* RBF) *:*

$$K(x_i,x_j) = \exp\left(-\frac{||xi-xj|^2}{2\partial}\right). \quad \text{(II.12)}$$

Dans notre travail on a utilisé un noyau gaussien de largeur de bande ∂ (RBF).

b. Noyau RBF : Les noyaux Radial Basis Function sont de la forme :

$$\boldsymbol{K(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{2\partial}\right)} \qquad \text{(II.13)}$$

c. Avantage du noyau RBF :

Voici trois raisons pour choisir ce noyau :

- Premièrement, ce noyau en transposant les attributs dans un espace de plus grande dimension, permet de trouver une frontière de décision non linéaire dans l'espace de départ.
- Deuxièmement, le noyau RBF a un seul paramètre ($\boldsymbol{\partial}$) intrinsèque, on peut donc jouer uniquement sur deux paramètres $\boldsymbol{\partial}$ et C pour régler l'apprentissage.
- Enfin, le noyau RBF présente moins de difficultés numériques que certains noyaux.

d. Le paramètre ∂ pour la fonction noyau RBF :

- Le paramètre σ permet de régler la largeur de la gaussienne. Sa valeur influence directement le résultat de classification.
- En prenant un ∂ grand, la similarité d'un exemple par rapport à ceux qui l'entourent sera assez élevée et conduit à une frontière de séparation peu régulière, les zones occupées par chaque classe sont discontinues.
- Un σ petit conduit à une séparation plus lisse et régulière, les zones de label séparées par la frontière sont connexes.

3.7 SVM multi-classes :

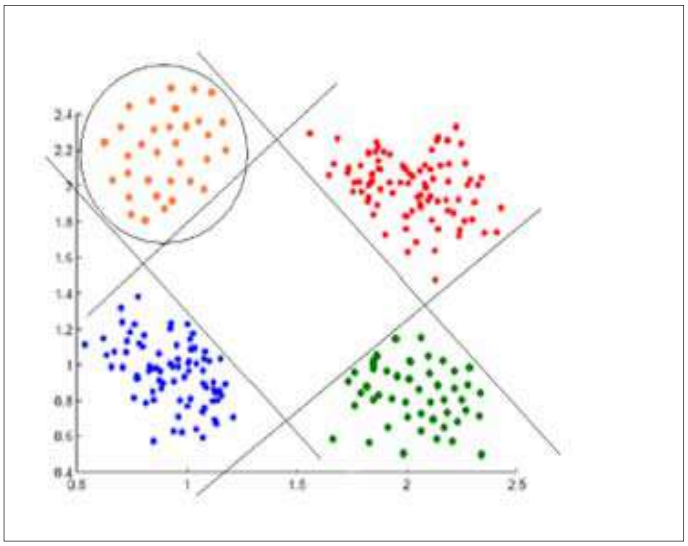
A l'origine les SVMs sont ce qu'on a appelés des classifieurs binaires et le principe du SVM expliqué dans la partie précédente se résume dans la résolution des problèmes de classification binaire.

D'ailleurs, en réalité, les problèmes rencontrés sont de type multi-classes, d'où l'importance d'étendre le principe du SVM aux problèmes de plus de deux classes. L'extension des SVM au cas multi-classes est actuellement un domaine de recherche très ouvert et fait l'objet de plusieurs travaux en cours. Nous allons par suite expliquer brièvement quelques méthodes parmi les plus utilisées.

3.7.1 Un contre tous :

Cette méthode est appelée en anglais **One-Versus-All (1-vs-A)** et consiste à déterminer pour chaque classe un hyperplan séparant celle-ci de toutes les autres classes, et pour M classes, on doit déterminer M fonctions de décision par M classifieurs SVM. Tous les exemples appartenant à la classe considérée sont étiquetés positivement (+1) et tous les exemples n'appartenant pas à la classe sont étiquetés négativement (-1). Il faut donc pour reconnaitre la classe d'appartenance d'un exemple, le soumettre à tous les classifieurs. Et par conséquent on a pour chaque exemple de test, M valeurs de sortie $f_i(x)$ avec (i=1, ... M) sont disponibles. Ainsi, une donnée est affectée à la classe qui correspond à la valeur maximale des fonctions de décision. Cette méthode suppose donc la construction de M classifieurs et M comparaisons pour la décision**.**

Exemple. *Pour M=4classes ; on aura 4 fonctions de décision.*



FigureII.10 : *Nuage de point à 4 classes avec l'approche Un contre Tous.*

3.7.2 Un contre un :

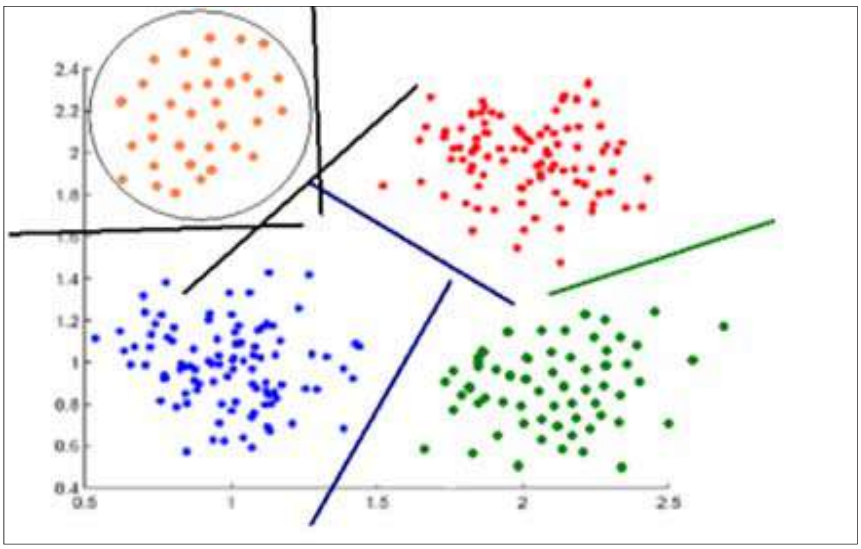
Cette méthode est appelée en anglais **One-Versus-One (1-vs-1)** et consiste à concevoir des classifieurs spécialisés dans la comparaison classe à classe (1-vs-1), on a donc M*(M-1)/2 fonctions de décisions qui sont apprises par M*(M-1)/2 classifieurs pour un problème à M classes et chacune d'entre elles effectue un vote pour l'affectation d'un nouveau exemple. Il faut donc pour reconnaitre la classe d'appartenance d'un exemple, le soumettre à tous les classifieurs 1-vs-1, chacun vote pour sa classe préférée, et la classe qui obtient le maximum de votes devient ensuite la classe majoritaire après le vote et remporte la bonne décision. Cette méthode possède un gros inconvénient : sa complexité augmente rapidement avec M puisqu'elle nécessite M*(M-1)/2 comparaisons.

Exemple :

Pour **M=2** classes ⟶ 2(2-1)/2 donc **une** fonction de décision.

Pour **M=3** classes ⟶ 3(3-1)/2 donc **3** fonctions de décision.

Pour **M=4** classes ⟶ 4(4-1)/2 donc **6** fonctions de décision.



FigureII.11 *: Nuage de point à 4 classes avec l'approche Un contre Un*

3.8 Les domaines d'applications des SVMs :

SVM est une méthode de classification qui montre de bonnes performances dans la résolution des problèmes variés. Cette méthode a montré son efficacité dans de nombreux domaines d'applications tels que le traitement d'image, la parole, la bioinformatique,

traitement de la langue et ce même sur des ensembles de données de très grandes dimensions.

3.9 Avantages et inconvénients :

a. Avantage :

- Les SVM possèdent des fondements mathématiques solides.
- Un grand taux de classification et de généralisation par rapport aux méthodes classiques.
- Elle nécessite moins d'effort pour designer l'architecture adéquate (petit nombre de paramètre à régler ou à estimer).
- Les exemples de test sont comparés juste avec les supports vecteur et non pas avec tous les exemples d'apprentissage.
- Décision rapide.

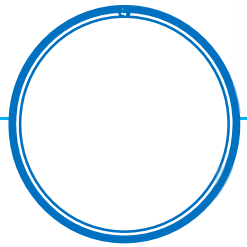
b. Inconvénients :

- Grande quantité d'exemples en entrées implique un calcul matriciel important.
- Temps de calcul élevé lors d'une régularisation des paramètres de la fonction noyau.

4 Conclusion

Nous avons exposé dans ce chapitre la métaheuristique récente basée sur les essaims de lucioles, son principe de fonctionnement, l'algorithme de luciole et ses paramètres, son efficacité et ses domaines d'applications sachant qu'il a donné des bons résultats presque dans tous les domaines où il a été appliqué.

Ensuite on a présenté la méthode d'apprentissage introduite par Vladimir Vapnik les Machines à vecteurs de supports. Nous avons expliqués les différents aspects de cette méthode de classification, le cas de deux classes et celui de multi-classes en présentant les théories qui sont à la base des SVM à savoir les fonctions noyaux et la notion de marge maximale.



CHAPITRE III

La Méthode Proposée Pour La Sélection D'Attributs & La Classification

1 Introduction:

La sélection d'attributs est une technique permettant de choisir un sous ensemble d'attributs pertinents pour la construction des modèles d'apprentissage robustes. Elle fournit une meilleure compréhension des données par la sélection des attributs importants au sein des données.

L'utilisation des méthodes inspirées de la biologie a attiré l'attention de la communauté de recherche depuis longtemps. De nos jours, les chercheurs essaient de proposer des améliorations importantes qui permettent l'apparition d'une nouvelle génération de méthodes puissantes et générales. Les métaheuristiques semblent bien appropriées pour traiter le problème de la sélection d'attributs qui est considéré comme un problème d'optimisation combinatoire NP-difficile.

Dans ce chapitre, nous proposons une méthode bio-inspiré de sélection d'attributs basés sur deux choix fondamentaux qui sont : (voir chapitre II).

- Une métaheuristique à population «Algorithme des lucioles ».
- La méthode de classification « SVM»

2 Méthode FA-SVM proposée:

Dans notre méthode nous considérons le problème de la sélection d'attributs comme un problème d'optimisation combinatoire mono objectif de maximisation (S, f) où *S* est l'ensemble des attributs et *f* la fonction à maximiser (fonction objective), c'est-à-dire qu'on cherche un sous-ensemble d'attributs afin de construire un classifieur ou un modèle de classification avec la performance maximum prédisant la classe de chaque nouvelle entrée.

Plus exactement, nous formulons ce problème de maximisation (S, f), tel que :

- L'espace de recherche S est défini par les sous-ensembles d'attributs sélectionnés possibles.
- La fonction objective $f : S \rightarrow R$ est tq quel que soit $s \epsilon S$, f(s) représente le taux de classification d'un classifieur construit avec le sous-ensemble d'attributs s.

Après qu'un problème d'optimisation est formulé correctement la tâche la plus importante est de trouver les solutions optimales en utilisant des procédures et des techniques appropriés.

Notre objectif est de proposer une méthode pour éliminer un grand nombre des attributs non pertinents et surtout des attributs redondants pour réduire la dimension de l'espace de travail et par conséquent améliorer la qualité de classification et l'apprentissage utilisé dans le but de créer un modèle de prédiction de haute performance.

Pour atteindre cet objectif la méthode FA-SVM sera proposée dans ce chapitre pour la sélection d'attributs et la classification. Elle sera représentée Figure III.1 et utilise l'approche enveloppante entre l'algorithme de lucioles (FA) et le classifieur (SVM) et elle a été implémentée en utilisant la version binaire de l'algorithme de firefly (FA).

2.1 Principe de la méthode FA-SVM proposée :

L'idée fondamentale de cette méthode est d'utiliser la métaheuristique de lucioles (FA) au début du processus de recherche pour la sélection d'un sous-ensemble optimale d'attributs pertinents et d'exploiter pleinement la puissance de l'apprentissage du classifieur SVM à noyau RBF qui a démontré son capacité à obtenir d'excellentes performances pour des tâches de classification, et utiliser ce classifieur pour évaluer les sous-ensembles obtenus par FA. Pour effectuer une sélection d'attributs cette méthode nécessite de nombreuses itérations. Et comme l'algorithme d'apprentissage est un processus itératif, le coût de calcul de l'ensemble du processus est très haut, puisque l'algorithme d'apprentissage est utilisé à chaque itération pour évaluer la fitness de chaque luciole.

2.2 Conception générale:

Pour la conception de notre méthode FA-SVM nous devons choisir les quatre éléments qui caractérisent les algorithmes de sélection d'attributs :

- Le type d'approche ;
- La direction et stratégie de recherche.
- Le critère et la fonction d'évaluation
- Le critère d'arrêt.

2.2.1 Type d'approche:

La méthode FA-SVM proposée utilise l'approche enveloppante parmi les trois approches existantes (voir chapitre I). Dans laquelle l'algorithme d'apprentissage fait partie intégrante du processus de sélection et sert de fonction d'évaluation. Cette méthode permet au mécanisme de sélection d'interagir avec le classifieur SVM pour trouver un sous-ensemble d'attributs qui sera optimal pour ce modèle d'apprentissage. L'intérêt d'utiliser une approche enveloppe est qu'elle fournit généralement des résultats plus pertinents pour la classification.

2.2.2 Direction et stratégie de recherche :

La deuxième décision à prendre est donc le point de départ de la recherche, il peut être de quatre sortes :

- l'ensemble de tous les attributs disponibles,
- l'ensemble vide,
- un ensemble d'attributs tirés de manière aléatoire.
- une population de sous-ensembles d'attributs lorsqu'on travaillera avec des méthodes à base de population.

Le point de départ de la recherche dans la méthode FA-SVM est une population de sous-ensembles d'attributs.

Ensuite, il faut choisir une méthode de génération à appliquer pour générer les sous-ensembles d'attributs (la génération complète ou la génération heuristique ou la génération aléatoire).

Nous avons utilisé la méthode la plus récente c'est *la génération aléatoire*. Il s'agit de constituer une population initiale servant de départ à l'algorithme. Grâce au comportement aléatoire de notre procédure d'initialisation, les solutions dans la population initiale sont tout à fait différentes. Ce point est très important pour cet algorithme parce qu'une population homogène a peu de chance de trouver une solution optimal.

2.2.3 Fonction d'évaluation:

Il existe deux grandes approches pour l'évaluation de la sélection d'attributs : suivant que l'on va évaluer les attributs individuellement ou évaluer des sous-ensembles d'attributs.

Le classifieur SVM utilisé par la méthode FA-SVM sert à évaluer le sous-ensemble d'attributs sélectionnés par une luciole, sachant que beaucoup de méthodes de sélection s'appuient sur l'évaluation des sous-ensembles pour gérer à la fois la redondance et la pertinence. Et comme notre but est de mesurer la capacité de prédiction de cette sous ensemble, Il semble intéressant d'utiliser la précision ou le taux de classification comme une fonction d'évaluation et qui sont les fonctions utilisées par toutes les méthodes enveloppe.

2.2.4 Critère d'arrêt:

Les critères d'arrêt existants sont soit basés sur l'algorithme de génération ou bien basés sur l'évaluation. La méthode FA-SVM proposée utilise les deux critères d'arrêt suivants :

- **Le premier critère basé sur l'algorithme de génération** qui consiste d'arrêter la recherche quand le nombre des itérations est égal au nombre maximal des itérations fixé précédemment comme un paramètre de FA.
- **Le deuxième critère basé sur l'évaluation :** dans ce cas la procédure de sélection sera terminée quand le taux de classification atteint 100%.

3 Organigramme général:

L'organigramme général de notre algorithme de sélection d'attributs est donné ci-après :

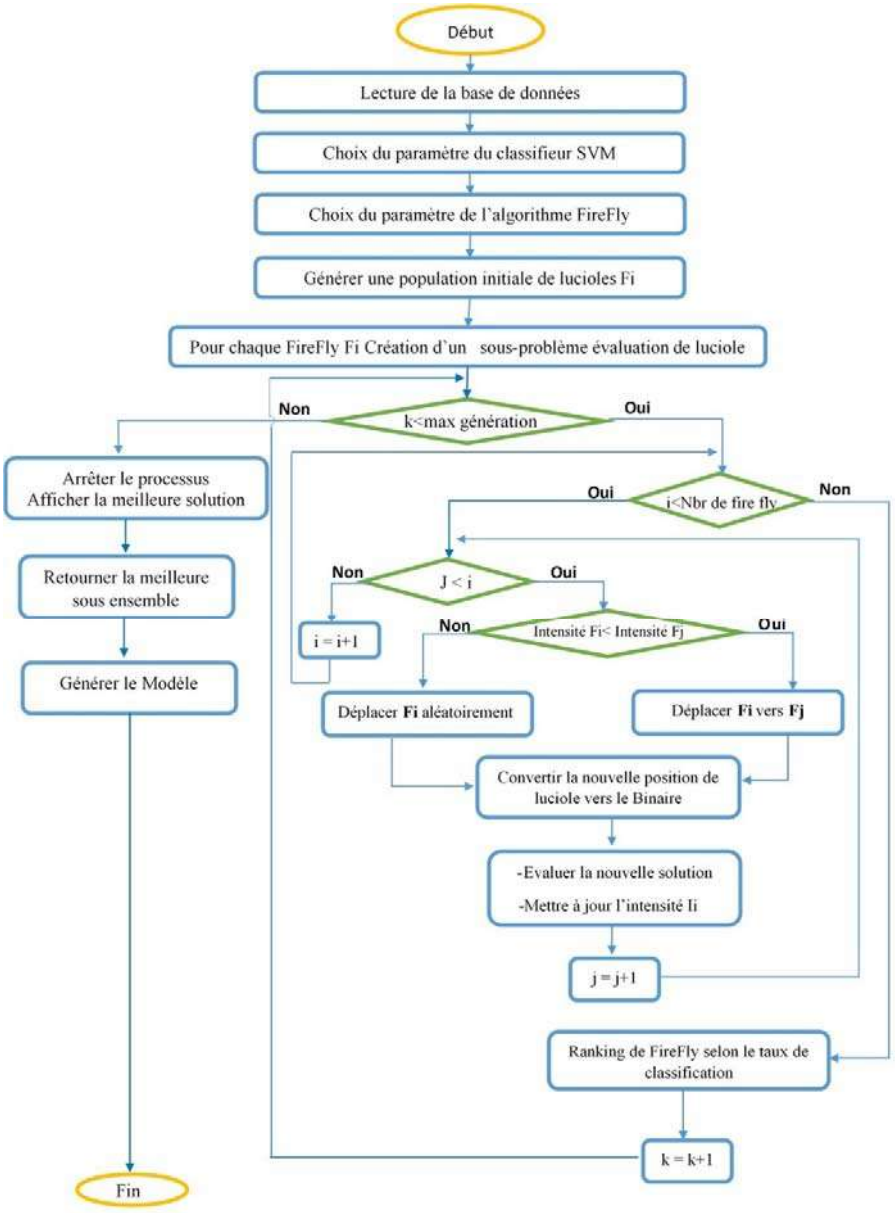


Figure III.1 : *L'organigramme général de la méthode FA-SVM*

4 Conception détaillée :

Nous avons proposé une démarche de sélection d'attributs suivant un processus à deux phases :

- Dans la première phase que nous appelons : phase d'initialisation, nous proposons d'appliquer l'algorithme de base FA-SVM afin d'obtenir un taux de classification initiale.
- A la seconde phase, que nous appelons phase d'amélioration, nous allons exécuter le programme sur plusieurs itérations afin d'améliorer le taux de classification retournée à la première phase et obtenir un taux de classification maximal.
- Les principales étapes de la méthode FA-SVM proposée sont les suivantes :

4.1 Phase d'initialisation :

Étape 1 : Initialisation et Choix des paramètres

- Initialiser les paramètres de FA :
  - L'attractivité $\beta 0$.
  - Le coefficient d'absorption γ.
  - La taille de la population des lucioles N.
- Déterminer les paramètres d'apprentissage de SVM
  - Initialiser les paramètres de SVM :
  - Déterminer le type de classification C-SVM
  - Déterminer les paramètres de la fonction noyau RBF.
- Déterminer le maximum d'itérations autorisées.

Dans cette étape, la population est initialisée avec N lucioles (solutions) générés aléatoirement.

Etape 2 : La sélection d'attributs et la représentation des solutions

Un aspect important de l'utilisation des métaheuristiques pour résoudre des problèmes est la façon dont sont codées toutes les solutions. Les algorithmes utilisant une métaheuristique établissent une analogie entre l'ensemble de solutions d'un problème et l'ensemble d'individus d'une population naturelle.

Représentation Bio-Inspiré	**Problème réelle**
Population de Lucioles	Ensemble de solutions
Position de luciole	Sous-Ensemble d'attributs sélectionnés proposé Comme solution (valeurs binaires 0 et 1)
Intensité de luciole	Taux de classification

Pour représenter le sous ensemble d'attributs sélectionnés, nous avons choisis d'appliquer ***la version binaire*** de l'algorithme FireFly dans laquelle la position de chaque FireFly dans l'espace de recherche multidimensionnel est représentée sous forme d'un tableau binaire N-dimensionnel qui est généré aléatoirement : les valeurs de bits 1 et 0 représentent un attribut sélectionné et non sélectionné, respectivement.

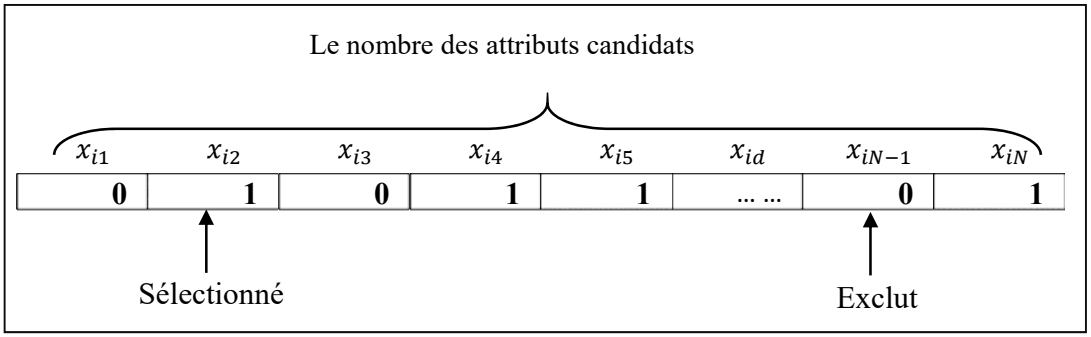


Figure III.2 : *Représentation binaire*

Étape 3 : Utilisation de la fonction objective (fitness), fonction d'évaluation et une méthode de validation :

Dans cette étape, il faut utiliser la fonction d'évaluation choisit pour évaluer les sous-ensembles générés par la procédure de génération. La tâche la plus importante est d'évaluer une luciole (ensemble d'attributs ou la solution) par une fonction objective ; le choix de cette fonction objective est très important dans l'utilisation de l'algorithme FA et dans toute les métaheuristiques pour résoudre des problèmes d'optimisation, c'est la clé de la réussite de l'évaluation, il détermine ce qu'une luciole devrait optimiser.

Cette étape consiste en une méthode enveloppe où FA explore des sous-ensembles candidats, et pour voir si un sous ensemble d'attributs obtenu est le meilleur en termes de taux de classification, chaque candidat est évalué grâce à un classifieur

SVM, qui fait un apprentissage afin de définir la pertinence des attributs par la prédiction du taux de classification du système final.

Ce classifieur utilise la procédure de la validation croisée. Nous avons choisi la validation croisée à 10 Folds puisqu'elle a été utilisée avec succès et a prouvé son efficacité dans plusieurs domaines. Le résultat est retenu comme une fonction d'évaluation des sous-ensembles d'attributs. Cette mesure est calculée par :

$$\mathbf{TCC = Total\ correct\ /\ L} \qquad \text{(III.1)}$$

Où *Total correct* est le nombre d'exemples correctement classifiés par le classifieur.
L est le nombre total d'exemples.
Le taux de classification indique si le sous-ensemble candidat permet une bonne discrimination des classes**.** Le taux de classification (TCC) est donc la fonction objective (fitness) à maximiser retenu dans notre méthode FA-SVM.

Étape 4 : Mise à jour de l'intensité et la position de chaque luciole.

Dans un espace de recherche multidimensionnelle, la position de chaque luciole pour chaque dimension est à 0 ou à 1 et sa modification est liée à l'intensité.

La population de lucioles fait des mouvements où chaque luciole est attiré par une autre ayant une intensité supérieur et également un taux de classification meilleur c-à-dire que la population fait des mouvement et converge vers le sous ensemble optimal ayant un taux de classification maximale. A chaque mouvement des lucioles, les nouvelles positions sont obtenu en réel et seront convertie en binaire, grâce à des méthodes de conversion indiquées dans le chapitre II (2.5).

Étape 5 : Ranking de lucioles selon le taux de classification et mémorisation de la meilleure solution (position).

4.2 Phase d'amélioration :

Étape 1 : Les itérations

Afin d'améliorer le taux de classification retournée à la première phase et obtenir un taux de classification maximal, le programme doit être exécuté sur plusieurs itérations. Si un critère d'arrêt est rencontré (le nombre d'itérations égal au nombre

maximal d'itérations ou bien le taux de classification égal à 100%) le processus s'arrête et le meilleur sous ensemble d'attributs est retourné.

Étape 2 : Sélection des paramètres de SVM

Dans la littérature, la recherche de paramètres adaptés est appelée *sélection de modèle*. Dans notre méthode FA-SVM nous avons essayé d'ajuster deux paramètres pour le classifieur SVM , le premier relatif au noyau RBF (Radial Basis Fonction) appelé gamma permet de choisir un niveau de localité entre les données, et le deuxième dit de régularisation, habituellement noté C. en utilisant la validation croisée de type Grid-Search, pour laquelle une recherche systématique est mise en œuvre en discrétisant l'espace des paramètres à l'aide d'un pas fixe.

4.3 Architecture de la méthode FA-SVM :

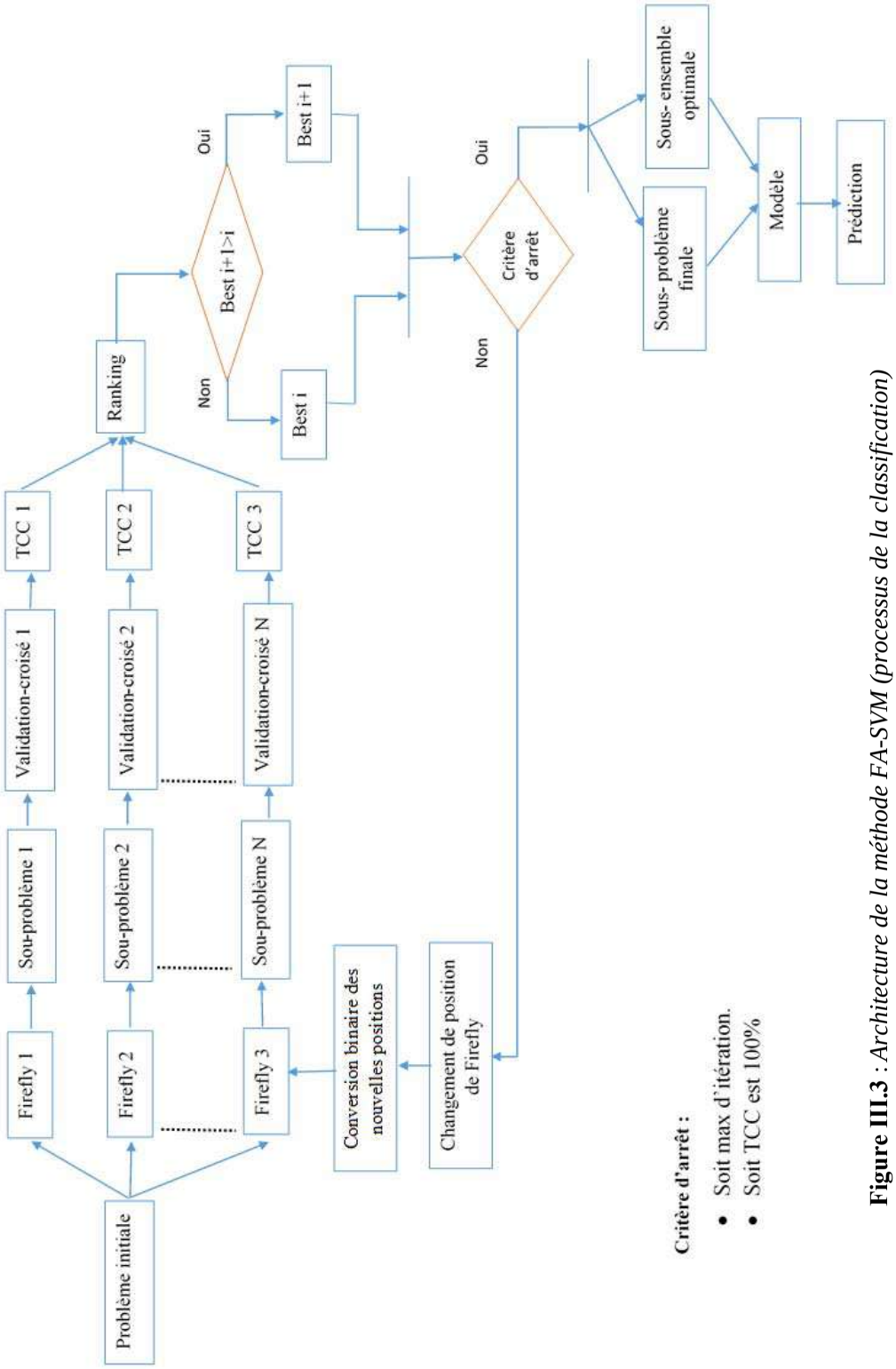


Figure III.3 : *Architecture de la méthode FA-SVM (processus de la classification)*

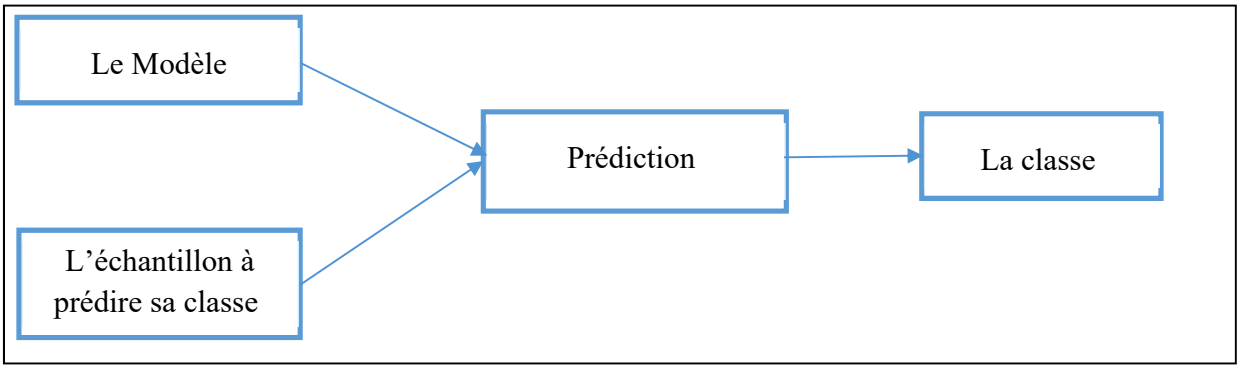


Figure III.4 : *Architecture de la méthode FA-SVM (processus de la prédiction)*

5 Implémentation de la méthode FA-SVM :

Pour implémenter la méthode FA-SVM proposée, nous avons fait une modélisation d'un système de datamining pour la classification et la prédiction. Pour cela, Nous avons choisi UML comme un langage de modélisation en utilisant ses diagrammes et JAVA comme un langage de programmation en utilisant l'IDE Netbeans.

Nous présentons dans cette partie les spécifications du système et l'analyse des besoins et les diagrammes UML.

5.1 La spécification du système et analyse des besoins :

Le but de cette phase est de définir ce que le système doit faire. Ce système doit permettre à l'utilisateur d'effectuer les tâches fondamentales suivantes :

- La classification des échantillons d'apprentissages à partir des bases de données volumineuses à fin de créer un modèle.
- La prédiction des nouveaux échantillons via ce modèle.

Pour atteindre cet objectif nous avons représenté les grandes fonctionnalités de notre système par les deux diagrammes des cas d'utilisation suivant sachant que notre système est avec un seul utilisateur.

Diagramme des cas d'utilisation1 : **« Génération du modèle de classification »**

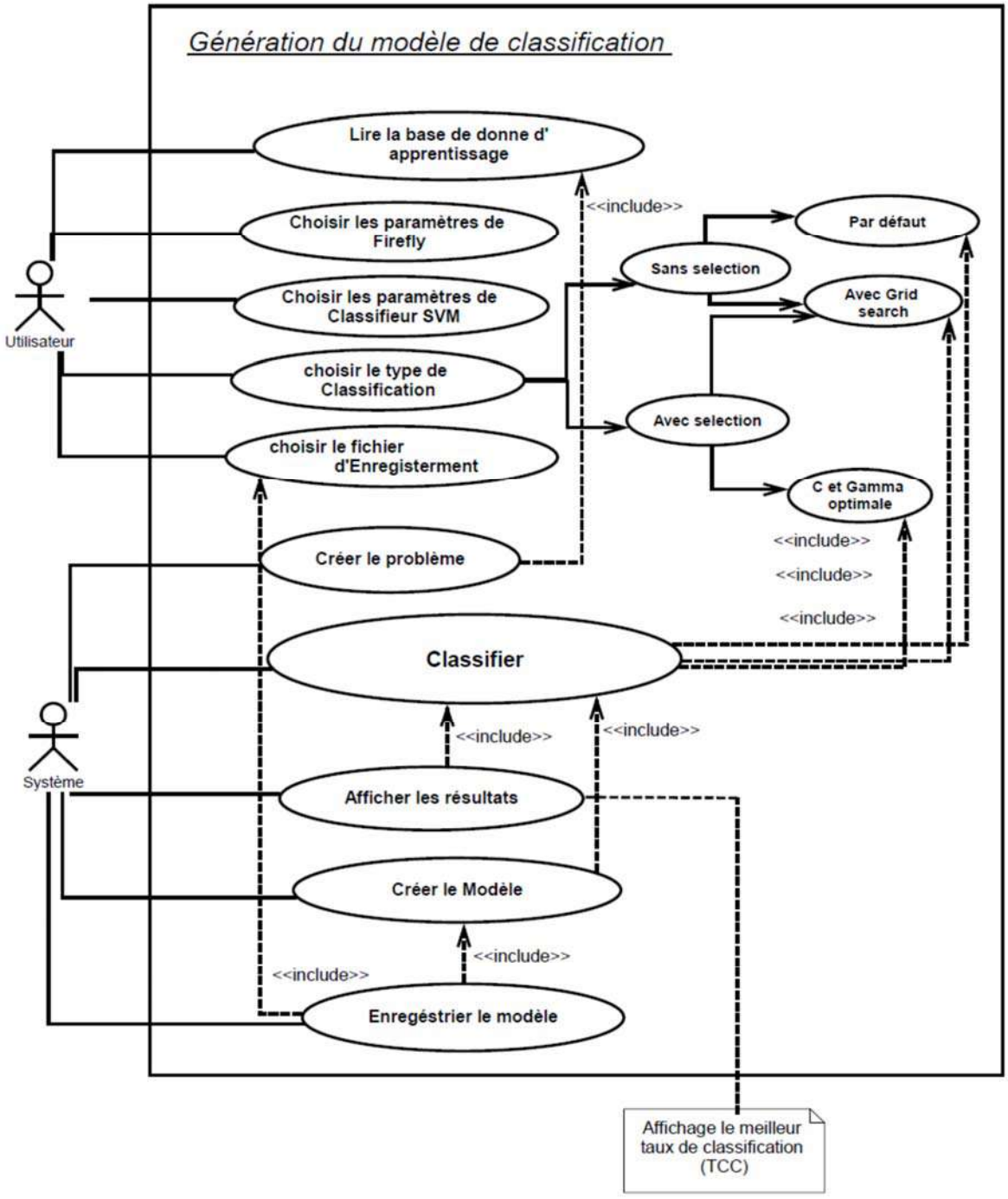


Figure III.5 : *Diagramme des cas d'utilisation1*

Scénario nominal

1. Choix de la base de données d'apprentissage.
2. Création du problème.
3. Choix des paramètres du FA
4. Choix des paramètres du classifieur SVM
5. Choix du type de sélection.
6. La classification.
7. Affichage des résultats.
8. Création et enregistrement du modèle.

Diagramme des cas d'utilisation2 : **« Système de Prédiction »**

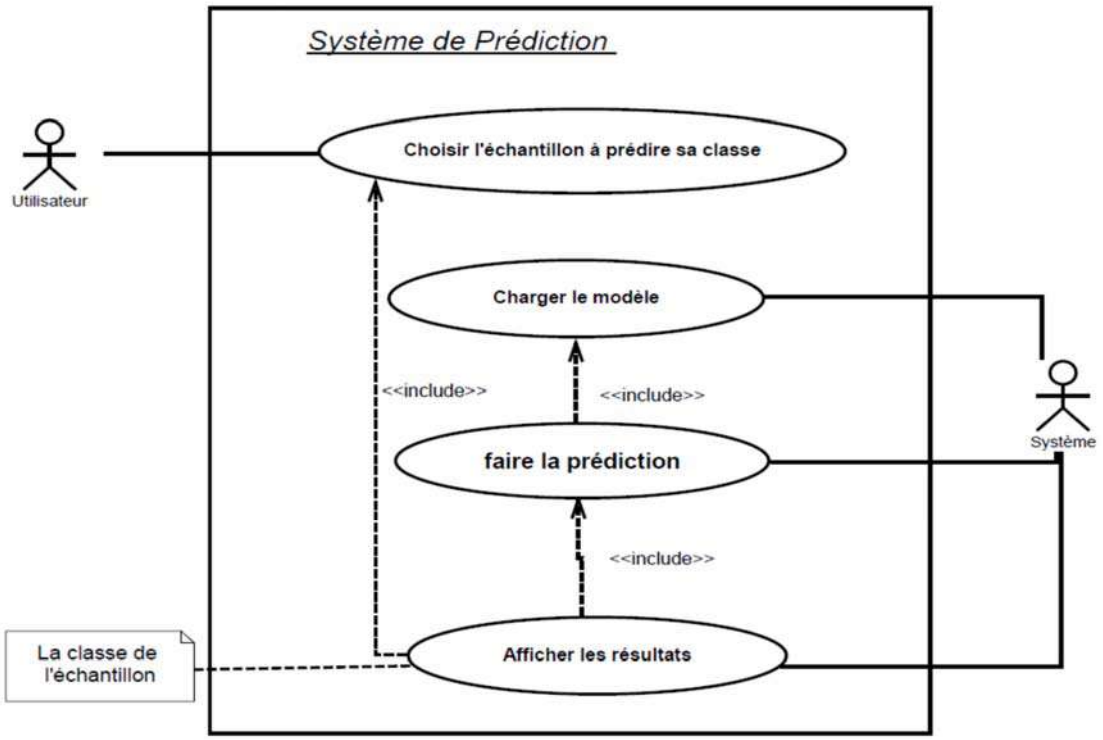


Figure III.6 : *Diagramme de cas d'utilisation 2*

Scénario nominal:

1. Choix de l'échantillon à prédire sa classe.
2. Chargement du modèle.
3. Prédiction.
4. Affichage des résultats.

Lors du processus de classification, les phases d'apprentissage peuvent prendre des temps de calcul importants lorsque le volume de données traitées est grand, par conséquent plus la machine utilisée pour l'exécution de l'application est performante (mémoire/CPU) plus les temps d'exécution seront minimaux et vice versa. Les performances matérielles se verront réellement solliciter lors de l'exécution du processus de prédiction sur des bases de données réelles, généralement très volumineuses.

5.2 Conception:

Après avoir défini les spécifications du système, la phase de conception vient pour mieux l'éclaircir. Dans une première partie, nous entamerons la conception technique en décrivant les vues statiques et dynamiques du système en utilisant les diagrammes UML appropriés. Ensuite, nous présentons la conception graphique de notre projet.

5.2.1 Conception technique :

Dans cette partie, nous entamons la conception technique de l'application. Nous commençons par la détermination du diagramme de classe, puis le diagramme d'activité.

- **Description de la vue statique (diagramme de classe) :** Le diagramme de classes est considéré comme le plus important de la modélisation orienté objet. Il s'agit d'une vue statique du fait qu'on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classe permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier. Le diagramme de classe retenu à la fin de la conception est le suivant :

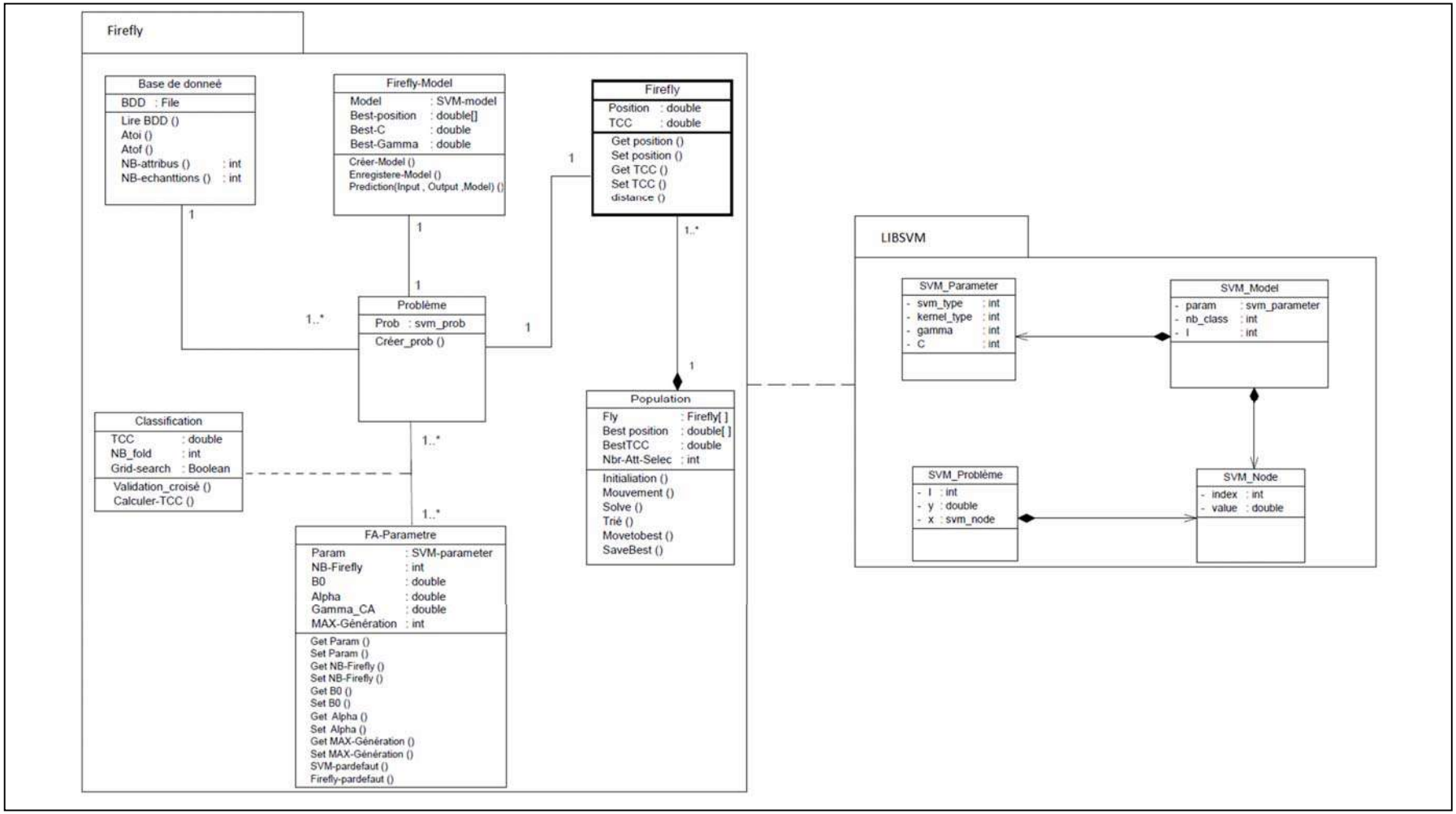


Figure III.7 : *Diagramme de classe.*

- **Description de la vue dynamique (diagramme d'activités) :** La description de la vue dynamique de notre application est réalisée à travers les différents diagrammes d'activité. En effet, un diagramme d'activité décrit l'enchainement des activités d'un système.

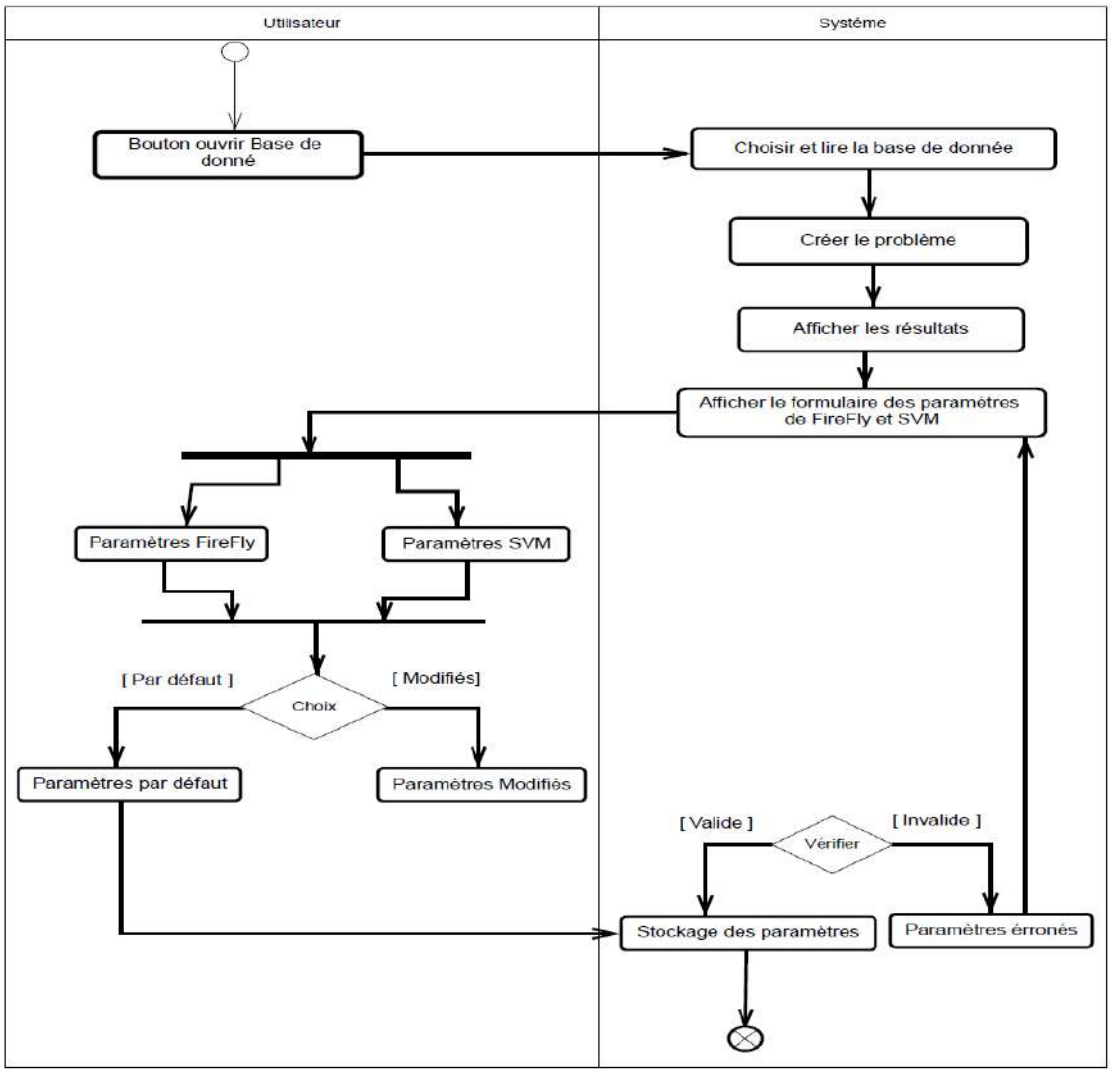


Figure III.8: *Chargement des données d'apprentissage, paramétrage de SVM et l'algorithme de FireFly*

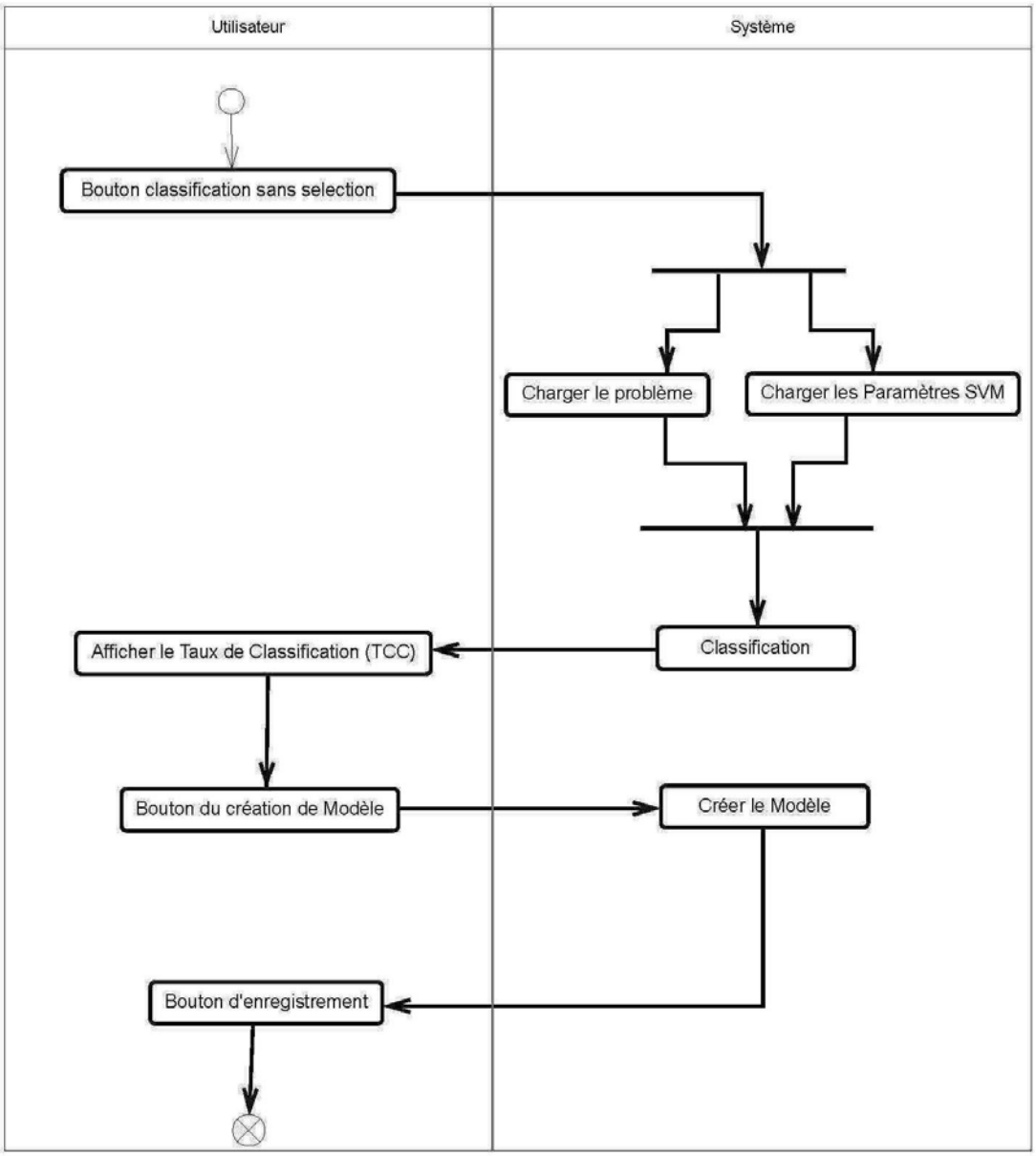


Figure III.9 : *Classification sans sélection*

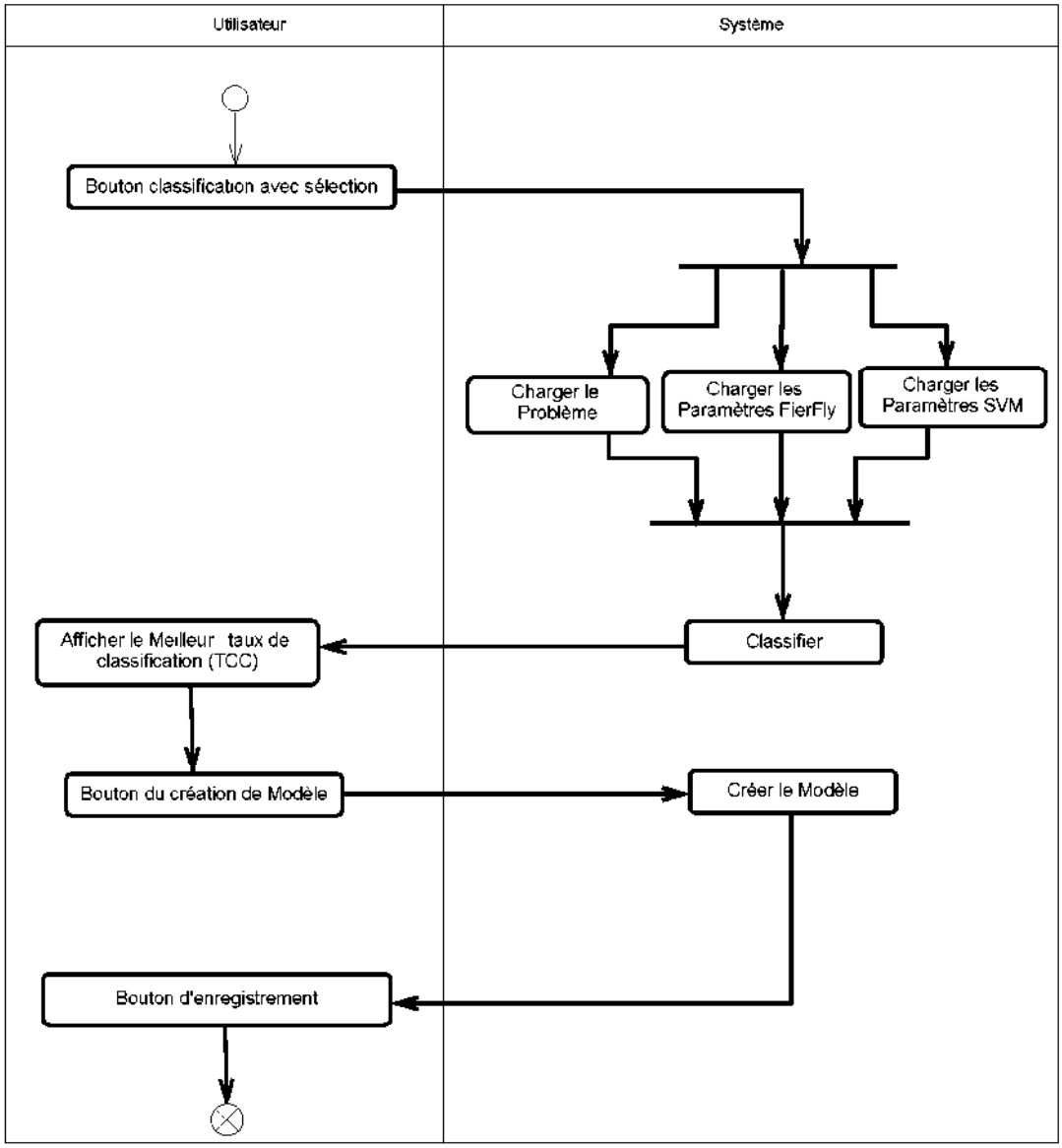


Figure III.10: *Classification avec sélection*

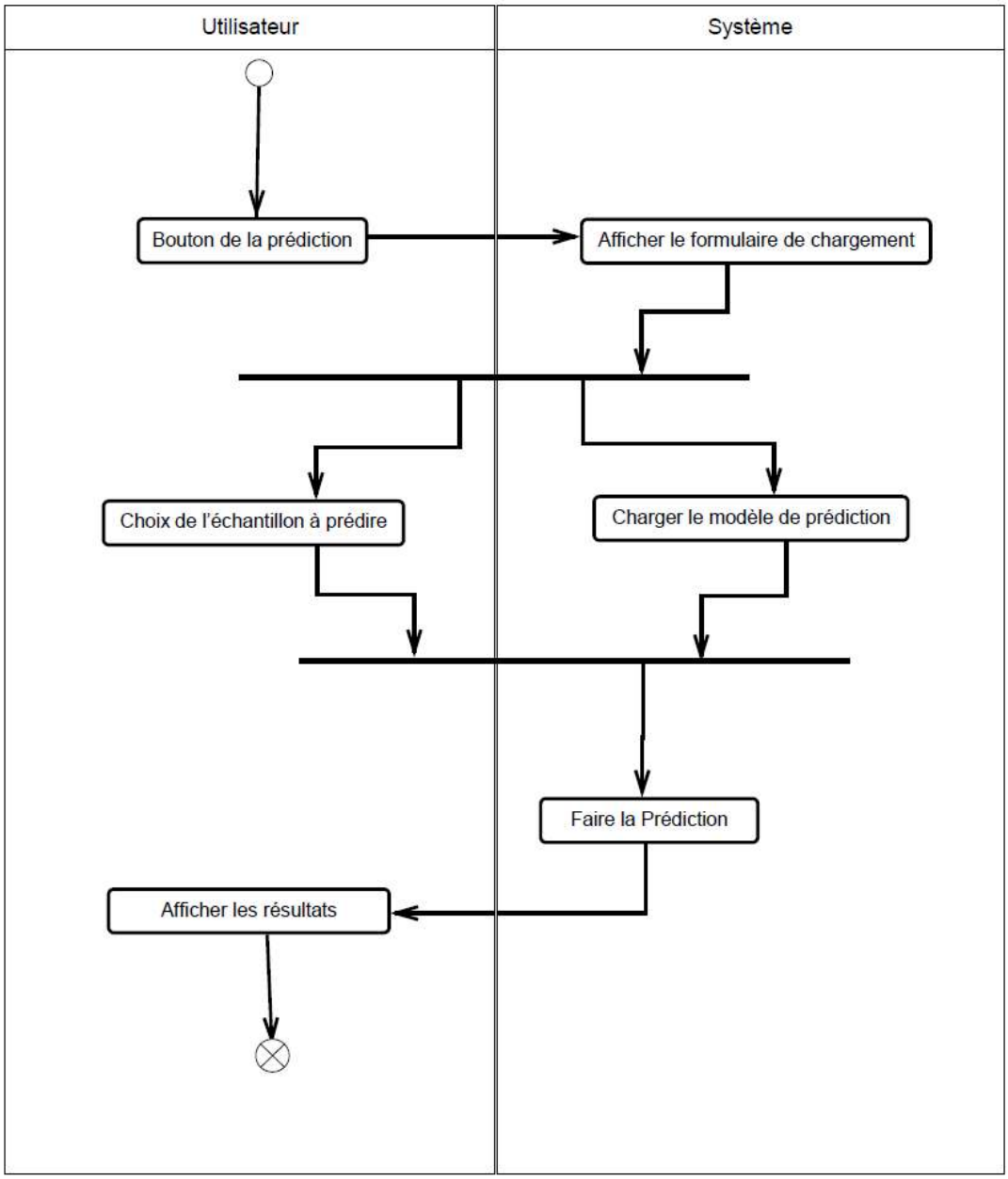


Figure III.11: *La prédiction*

5.3 Réalisation:

L'application à réaliser s'inscrit dans le cadre de la modélisation d'un système de datamining pour la classification et la prédiction. Dans cette partie nous décrivons d'abord l'environnement de développement ensuite nous présentons l'interface de l'application conçu précédemment.

5.3.1 Environnement de développement :

- **Langage de programmation JAVA :** afin d'assurer que notre programme fournira les mêmes résultats s'il sera exécuté dans des environnements de test différents et spécialement des HPC fonctionnant avec des systèmes d'exploitation différents. Nous avons choisi de réaliser notre application avec le langage Java.

Java est un langage de P.O.O qui contribue à la fiabilité des logiciels et elle facilite la réutilisation de code existant grâce à sa portabilité en utilisant la JVM.

- **Environnement de développement Netbeans :** NetBeans est un projet open source ayant un succès et une base d'utilisateur très large, une communauté en croissance constante, et près 100 partenaires mondiaux et des centaines de milliers d'utilisateur à travers le monde. Sun Microsystems a fondé le projet open source NetBeans en Juin 2000 et continue d'être le sponsor principal du projet.

Nous avons réalisé l'application en utilisant L'IDE NetBeans version 8.0.2 [21].C'est un environnement de développement et un outil pour les programmeurs pour écrire, compiler, déboguer et déployer des programmes. Il est écrit en Java mais peut supporter n'importe quel langage de programmation. Il y a également un grand nombre de modules pour étendre L'IDE NetBeans. L'IDE NetBeans est un produit gratuit, sans aucune restriction quant à son usage. Le site : https://netbeans.org offre toute les informations nécessaires sur cette IDE.

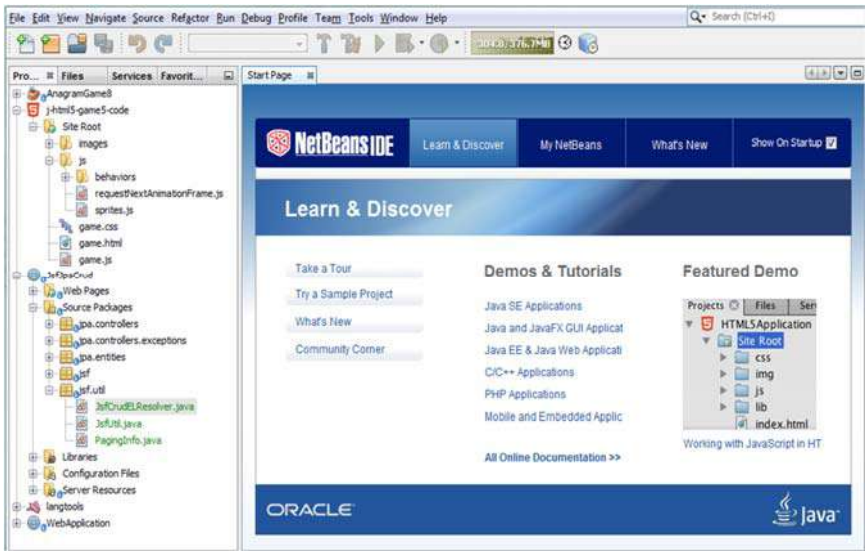


Figure III.12 : *L'interface de l'IDE Netbeans*

- **La bibliothèque Libsvm :** Dans le but de standardiser et orienter les futurs travaux de recherches sur les Machines à Support de Vecteurs (Support Vector Machines), la librairie LIBSVM [28] a été élaborée par Chih-Chung Chang et Chih-Jen Lin, deux chercheurs du département informatique de l'université de Taipei à Taiwan « National Taiwan University ». C'est une librairie dédiée aux Machines à Support de Vecteurs (Support Vector Machines) et vastement exploitée dans le domaine de l'apprentissage automatique (learning machines), principalement en supervisé. Cette librairie disponible en langage C++ et en JAVA englobe un ensemble de classes contenant des méthodes implémentant des algorithmes de fouille de données, particulièrement utilisée pour mettre en œuvre des classifications et des régressions.

5.3.2 Présentation de l'application :

- **Les Interfaces de l'application :**

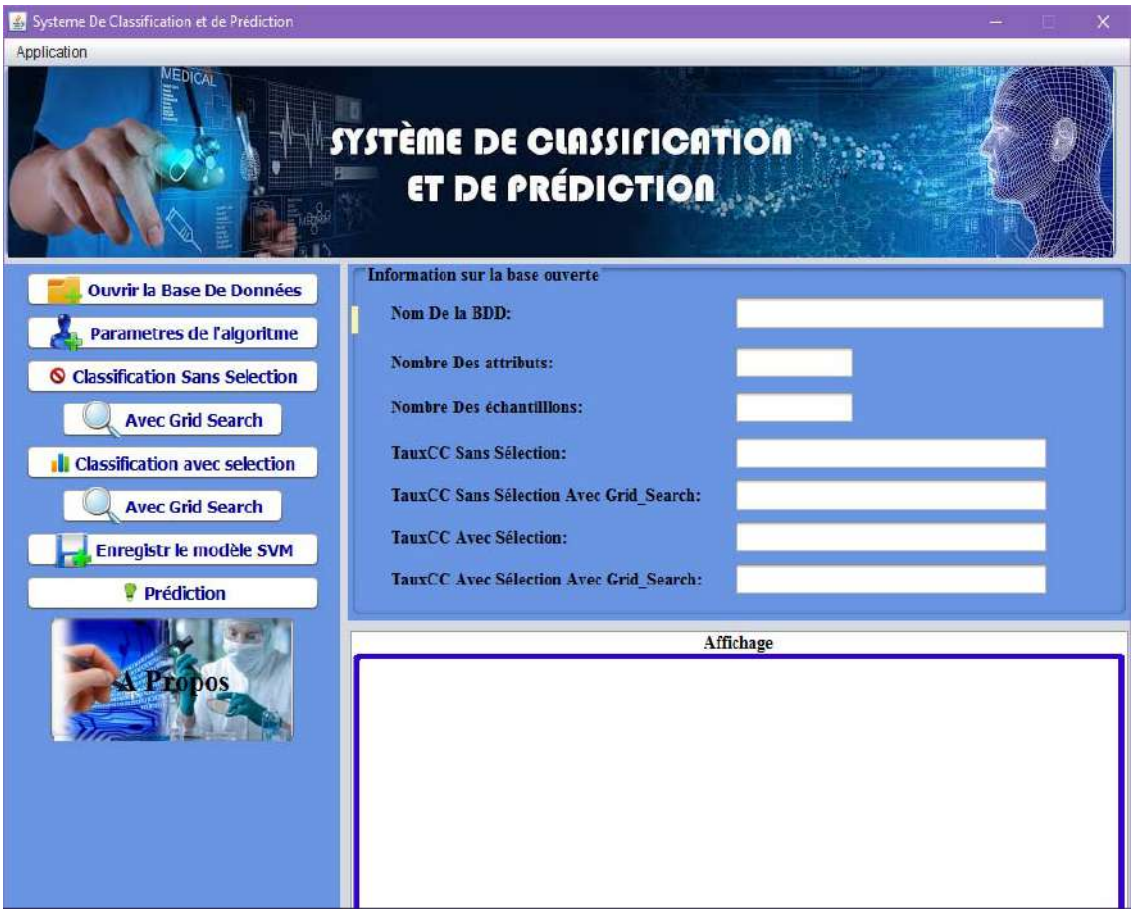


Figure III.13 : *L'interface de l'application -La forme Principale-*

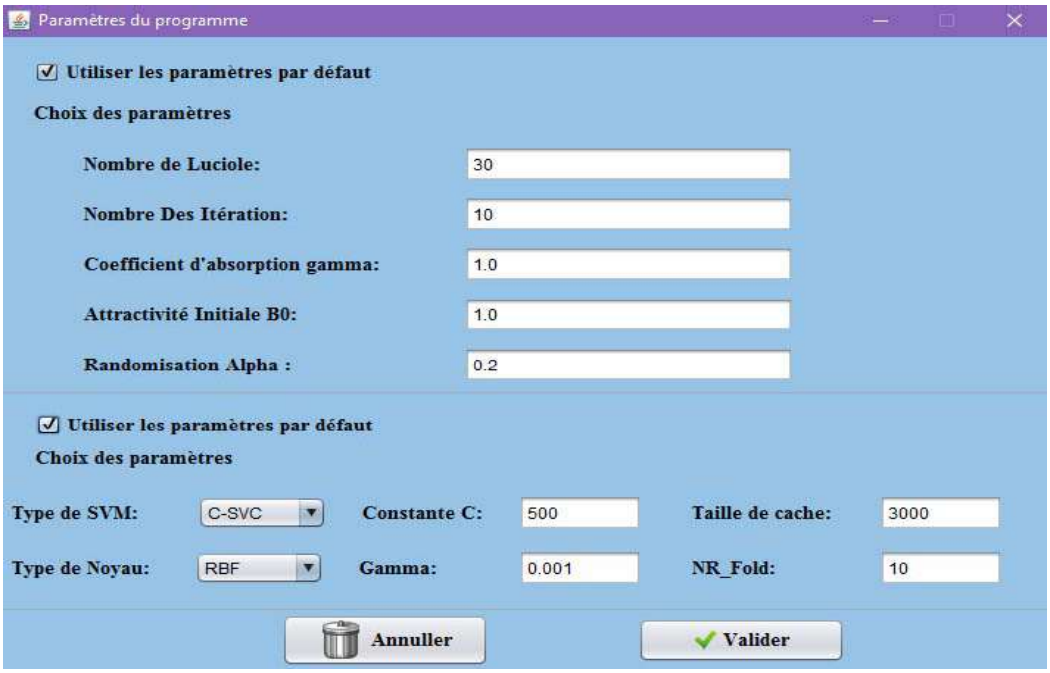


Figure III.14 : *L'interface de l'application-Forme du réglage des paramètres-*

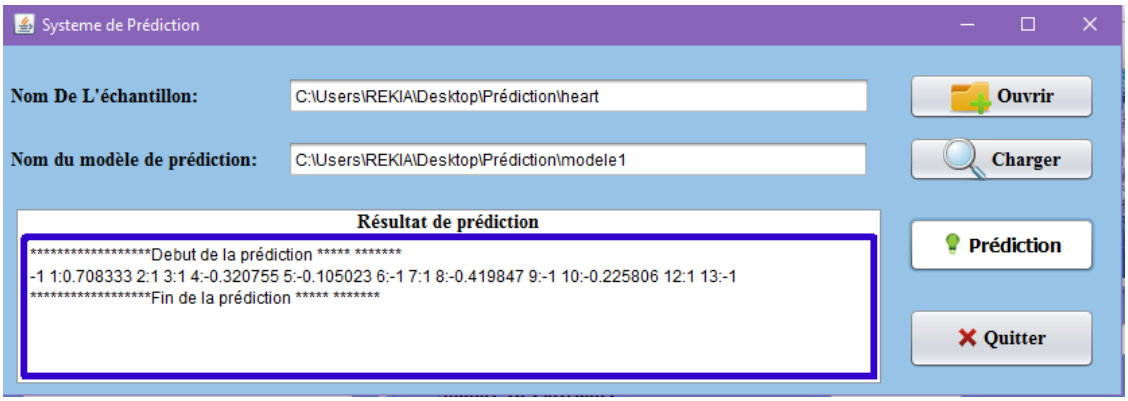


Figure III.15 : *L'interface de l'application-Forme de la prédiction-*

6 Conclusion

La méthode de sélection FA-SVM a été proposée pour améliorer les performances d'un système de dataming pour la classification.

Dans ce chapitre après avoir présentée une conception générale nous avons donné une description détaillée de chaque étape de cette méthode, ensuite l'implémentation a été faite en créant un système de datamining pour la classification, en utilisant le langage JAVA et d'autres outils, sachant que nous interagissant avec l'utilisateur à travers une interface graphique pour guider le processus de sélection vers le bon chemin.

Nous consacrons le chapitre suivant aux tests et évaluation des résultats obtenus de la méthode FA-SVM proposée.

CHAPITRE IV

Résultats et Évaluation

1 Introduction:

Nous présentons dans ce chapitre, l'expérimentation de la méthode FA-SVM, proposée dans le chapitre précèdent, par leur application dans le domaine de la sélection d'attributs et du datamining. La première partie de notre travail consiste à faire une évaluation initiale sans sélection d'attributs des performances de classification de notre modèle SVM en termes de taux de classification appliqués sur 9 bases de données. Dans la deuxième partie et afin d'améliorer les performances de la classification, une procédure de sélection d'attributs pertinents a été effectuée en utilisant la méthode enveloppe FA-SVM.

2 Matériel et logiciels utilisées:

Le matériel utilisé pour développer la méthode de sélection d'attributs FA-SVM est :

- **Un ordinateur portable** avec les caractéristiques suivantes : 4 processeurs Intel(R) Core(TM) CPU i7-4500U, Mémoire RAM de 8 GO sous le système d'exploitation Windows 10.
- **Le superordinateur (HPC)** [1]: Nous avons utilisé pour les tests le superordinateur du centre de calcul de l'université de Médéa. C'est un cluster composé de 32 nœuds de calcul, chaque nœud est un serveur IBM X3550 M3 doté de deux processeurs Intel Xeon X5675 de 6 cœurs de 3.06 GHz et de 12 Mo de Cache. Le cluster fonctionne avec le système d'exploitation Linux Redhat 6.0 Enterprise for HPC.

Pour y lancer des commandes depuis Windows nous avons utilisé le logiciel Putty et pour transférer les fichiers vers HPC le logiciel Filezella a été utilisé.

Afin de pouvoir évaluer les résultats de l'approche proposées FA-SVM, nous avons utilisé le système de classification et prédiction conçu et réalisé précédemment pour cette tâche, et qui a l'avantage de pouvoir être exécuté sur différents type de HPC et avec différents Systèmes d'exploitation.

[1] High-Performance Computer

3 Description des bases de Données de l'UCI :

Nous avons utilisé des données provenant de l'UCI Machine Learning Repository [13] qui contient une collection de bases. Nous avons utilisé ces jeux de données publics, facilement accessibles et qui sont utilisés dans de nombreux travaux concernant la classification des données. Ces jeux constituent en quelque sorte des jeux tests qui permettent de comparer les méthodes proposées depuis quelques années. Nous donnons ci-dessous les caractéristiques des jeux de données que nous avons choisis pour la validation de notre méthode de sélection d'attributs FA-SVM. Le test a été faite sur des bases de données de dimensions différentes :

- Les petites bases :

Base1 : Breast-cancer.

Base 2 : Heart

Base 3 : Sonar

Base 4 : Vehicule

Base 5 : Diabetes

Base 6 : Wine.

- Les grandes bases :

Base 7 : Colon-cancer

Base 8 : Leu-Glob-cancer

Base 9 : Ducke-Breast-cancer

Tableau IV.1 : *Caractéristiques des bases utilisées pour le test.*

N°	Base de données	Nombre d'Attributs	Nombre D'échantillons	Nombre de Classes
1	Diabètes	8	768	2
2	Breast_Cancer	10	683	2
3	Heart_stat	13	270	2
4	Sonar	60	208	2
5	Vehicle	18	846	4
6	Wine	13	528	11
7	Colon_Cancer	2000	20	2
8	Duke_breast-cancer	7129	44	2
9	Leu_glob_cancer	7129	72	2

4 Prétraitement des données:

Le processus de prédiction avec les SVMs nécessite la normalisation des données brutes, afin d'éviter que les valeurs à grandeurs extrêmes n'altèrent la qualité de l'analyse. De plus, les fichiers de données d'apprentissage et de test doivent être normalisés sur le même intervalle et de la même manière. La bibliothèque LIBSVM propose une fonctionnalité pour la normalisation des données, elle peut être utilisée en ligne de commande via la commande svm-scale.

Dans notre travail on a fait une normalisation linéaire des valeurs des attributs sur l'intervalle [-1 ; +1], en utilisant la formule suivante :

$$x' = \left(\frac{x - Min_a}{Max_a - Min_a}\right) * 2 - 1 \qquad \text{(IV.1)}$$

Où x représente la valeur d'origine, x' désigne la valeur normalisée, Max_a est la limite supérieure des valeurs de l'attribut a, et Min_a est la limite inférieure des valeurs de l'attribut a.

5 Choix expérimentaux:

La mise en œuvre d'une classification par les Méthodes à Vecteurs de Support SVM requiert la détermination préalable des valeurs adéquates des hyper-paramètres inhérents à la méthode. La validation croisée demeure la méthode de l'état de l'art dédiée à cette tâche. L'inconvénient majeur de cette dernière, réside dans les charges de calculs qui ne permettent pas de l'exploiter pour des applications dont le volume de données est très grand.

5.1 Paramètres de l'algorithme de lucioles :

Les paramètres de l'algorithme de lucioles ont été déterminés expérimentalement. Ils sont présentés dans le Tableaux IV.2.

Tableau IV.2 : *Paramètres de l'algorithme de lucioles.*

Paramètres	Valeur
Nombre de firefly	30
Nombre des itérations	10, 20,30, 40, 50, 60, 70, 80, 90,100
Beta0	1.0
Alpha	0.2
Gamma (Coefficient absorption)	1.0

5.2 Paramètres de SVM:

Tableau IV.3 : *Paramètres du classifieur SVM.*

Paramètre	Valeur
svm_type	C-SVM=0
kernel_type	RBF=2
gamma	1/Nombre d'attributs
C	1
Nr-fold	10
cross_validation	1

6 Évaluation des performances de classification :

Nous utilisons une validation croisée à 10 folds dans tous les tests faites, à fin de garantir d'un côté, que tous les échantillons participent en tant qu'ensemble d'apprentissage et comme ensemble de test et d'un autre côté, on assure la répartition homogène des classes entre les dix blocs.

Afin de vérifier l'efficacité de FA-SVM proposée, le taux de classification correct TCC a été utilisé pour toutes les bases de données. Le TCC est définit comme suit :

TCC = (nombre des décisions correct /nombre total des exemples)*100 (IV.2)

Suite à la nature non-déterministe de la méthode proposée, plusieurs exécutions (20) ont été considérées pour chaque base de données. La valeur minimale, la valeur maximale, et la moyenne du taux de précision de la classification pour chaque base sont rapportés.

7 Résultats de Test:

7.1 Test de la base de données Heart-stat :

Tableau IV.4 : *Classification sans sélection de la base de données Heart-stat.*

N° Exécution	Classification Sans Sélection			
	SVM *Defaut*	SVM *Optim*		
	TCC	C *Optim*	g *Optim*	TCC
1	80,740	16	0,001	84,070
2	82,590	4	0,007	85,550
3	81,480	4	0,007	84,440
4	80,740	4	0,007	84,810
5	81,110	4	0,030	84,440
6	81,110	16	0,001	84,070
7	80,740	4	0,007	84,810
8	80,000	1024	0,001	84,440
9	81,110	16	0,001	85,180
10	81,480	16	0,001	84,810
11	80,000	4	0,001	84,070
12	80,000	4	0,007	84,440
13	81,110	16	0,007	84,440
14	81,480	16	0,001	84,810
15	81,480	4	0,007	84,810
16	80,740	4	0,001	84,070
17	80,370	16	0,001	84,440
18	80,740	64	0,001	84,440
19	78,140	16	0,001	84,440
20	80,370	1	0,001	84,070

Taux -Class (Sans selection)			
Min	**Max**	**Moy**	**Ecart type**
84.07	85.55	84.53	0.40

Tableau IV.5 : *Classification avec sélection de la base de données Heart-stat*

N° Exécution	Classification Avec Sélection					
	FA+SVM ***Optim***				**FA+SVM** ***Grid-Search***	
	TCC	**NBAS**	**C** ***Optim***	**g** ***Optim***	**TCC**	**NBAS**
1	86.29	5	16	0.5	.8777	7
2	85.55	7	256	0.001	86.66	4
3	85.92	8	16	0.125	87.03	7
4	85.18	9	16	0.125	87.03	3
5	85.92	10	4096	0.0019	87.03	8
6	85.92	8	1024	0.03125	68.66	10
7	85.55	9	16384	0.0078	87.77	7
8	85.92	10	4	0.125	86.29	9
9	85.92	9	16	0.001	86.66	3
10	85.55	9	16384	0.001	87.03	9
11	86,66	7	64	0.03125	87.03	9
12	85,55	8	4	0.5	87.77	9
13	85,18	6	4096	0.007	86.66	3
14	86,29	7	4096	0.007	87.40	7
15	85,18	9	1	0.125	87.03	6
16	85,55	10	4	0.007	86.66	8
17	85,18	10	1024	0.007	87.03	8
18	85,18	8	4	0.5	86.66	9
19	84,81	10	16384	0.03125	86.66	7
20	85,18	9	1024	0.03125	87.40	6

Taux -Class (Avec selection)			
Min	**Max**	**Moy**	**Ecart type**
86.66	87.77	86.11	4.12

7.2 Classification Avec et sans Sélection d'attributs d'un ensemble des bases de données :

Le Tableau IV.6 présente pour chaque jeu de données le taux de classification Minimal et maximal, la moyenne et l'écart type de ce taux.

Tableau IV.6 : *Classification avec et sans sélection d'un ensemble des bases de données.*

Base de Données	**Classification Sans Sélection**				**Classification Avec Sélection FA**					
	Min%	**Max%**	**Moy%**	**Ecart Type**	**Min%**	**Max%**	**Moy%**	**Ecart Type**	**Moy NBAS**	**NB Attr Orig.**
Diabètes	76.82	78.26	77.44	0.39	78.39	78.91	78.65	0.13	4.35	8
Breast Cancer	96,78	97,22	97,05	0,13	97,36	97,51	97,47	0,07	5,80	10
Sonar	93,26	96,63	94,47	0,90	93,26	96,63	94,47	0,90	19.90	60
Véhicule	74,70	77,90	76,29	0,75	86.05	87.35	86.67	0.34	12.85	18
Wine	97,19	99,44	98,29	0,56	99.43	100	99.74	0.29	8.30	13
Colon_Cancer	60,00	80,00	68,25	6,13	100	100	100	0,00	434,05	2000
D_breast-cancer	47,73	59,09	54,09	3,73	93,18	95,45	94,66	1,11	1558,3	7129
Leu_glob_cancer	70,83	73,61	72,92	1,06	98.61	98.61	98.61	0,00	1124.85	7129

7.3 Représentation de TCC et le nombre d'attributs sélectionnés en fonction de nombre d'itérations :

Afin de sélectionner le meilleur sous ensemble d'attributs, nous allons étudier l'évolution des performances de notre méthode FA-SVM testé sur 9 bases de données, en calculant le TCC en fonction du nombre d'attributs sélectionnés et en fonction du nombre d'itérations (pour 10 à 100 itérations).

Les résultats obtenus sont données par le tableau et les figures suivantes :

❖ **Test sur la base de données Colon_Cancer :**

Tableau IV.7 : *Classification avec et sans sélection de la base de données Colon-Cancer*

N° Exe	Colon_cancer																				TCC Sans selection
	Taux-Class Avec selection																				
	Iter: 10		Iter: 20		Iter: 30		Iter: 40		Iter: 50		Iter: 60		Iter: 70		Iter: 80		Iter: 90		Iter: 100		
	TCC	# Att	TCC	# Att	TCC	# Att	TCC	# Att	TCC	# Att	TCC	# Att	TCC	# Att	TCC	# Att	TCC	# Att	TCC	# Att	
1	100,00	721	100,00	687	100,00	670	100,00	595	100,00	595	100,00	543	100,00	537	100,00	537	100,00	503	100,00	471	70,00
2	100,00	779	100,00	668	100,00	588	100,00	570	100,00	520	100,00	474	100,00	474	100,00	474	100,00	446	100,00	446	75,00
3	100,00	728	100,00	686	100,00	686	100,00	632	100,00	591	100,00	546	100,00	523	100,00	523	100,00	523	100,00	511	80,00
4	100,00	752	100,00	659	100,00	659	100,00	578	100,00	520	100,00	490	100,00	480	100,00	459	100,00	447	100,00	445	65,00
5	100,00	686	100,00	614	100,00	524	100,00	524	100,00	507	100,00	473	100,00	466	100,00	429	100,00	406	100,00	395	75,00
6	100,00	684	100,00	602	100,00	511	100,00	511	100,00	487	100,00	487	100,00	453	100,00	442	100,00	442	100,00	413	75,00
7	100,00	791	100,00	658	100,00	591	100,00	591	100,00	519	100,00	501	100,00	483	100,00	445	100,00	434	100,00	431	65,00
8	100,00	667	100,00	622	100,00	580	100,00	580	100,00	546	100,00	477	100,00	425	100,00	422	100,00	395	100,00	388	65,00
9	100,00	659	100,00	624	100,00	624	100,00	590	100,00	571	100,00	571	100,00	529	100,00	517	100,00	517	100,00	478	75,00
10	100,00	657	100,00	597	100,00	575	100,00	559	100,00	559	100,00	484	100,00	480	100,00	480	100,00	467	100,00	461	65,00
11	100,00	757	100,00	722	100,00	640	100,00	639	100,00	564	100,00	544	100,00	544	100,00	511	100,00	488	100,00	442	60,00
12	100,00	773	100,00	743	100,00	638	100,00	548	100,00	508	100,00	441	100,00	423	100,00	415	100,00	410	100,00	410	75,00
13	100,00	767	100,00	619	100,00	510	100,00	510	100,00	509	100,00	502	100,00	483	100,00	467	100,00	446	100,00	429	70,00
14	100,00	811	100,00	616	100,00	557	100,00	557	100,00	557	100,00	534	100,00	517	100,00	444	100,00	442	100,00	412	70,00
15	100,00	797	100,00	702	100,00	646	100,00	621	100,00	550	100,00	517	100,00	517	100,00	489	100,00	466	100,00	438	65,00
16	100,00	719	100,00	638	100,00	598	100,00	598	100,00	534	100,00	492	100,00	468	100,00	468	100,00	440	100,00	435	60,00
17	100,00	728	100,00	701	100,00	631	100,00	590	100,00	512	100,00	509	100,00	486	100,00	448	100,00	420	100,00	413	60,00
18	100,00	745	100,00	680	100,00	629	100,00	596	100,00	581	100,00	581	100,00	532	100,00	502	100,00	493	100,00	483	60,00
19	100,00	873	100,00	601	100,00	548	100,00	427	100,00	392	100,00	373	100,00	373	100,00	373	100,00	372	100,00	344	70,00
20	100,00	745	100,00	632	100,00	554	100,00	552	100,00	512	100,00	496	100,00	474	100,00	452	100,00	441	100,00	436	65,00

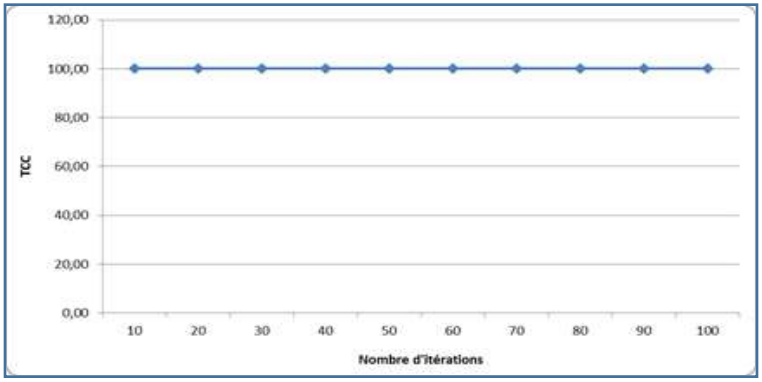


Figure IV.1 : *Graphe de la moyenne de taux de classification en fonction de nombre d'itérations*

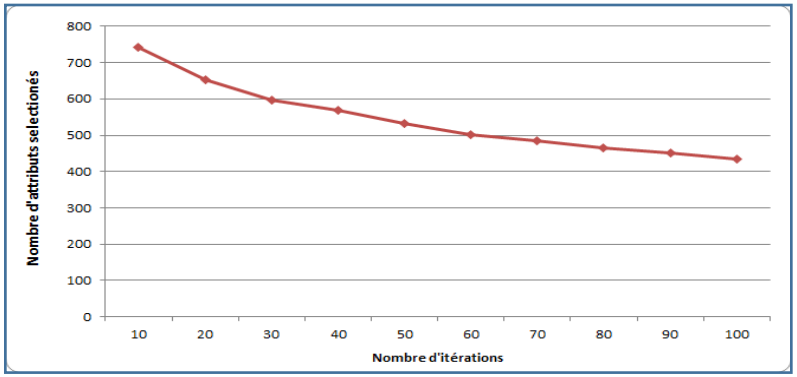


Figure IV.2 : *Graphede la moyenne du nombre d'attributs sélectionnés en fonction de nombre d'itérations*

❖ La base Breast_Cancer:

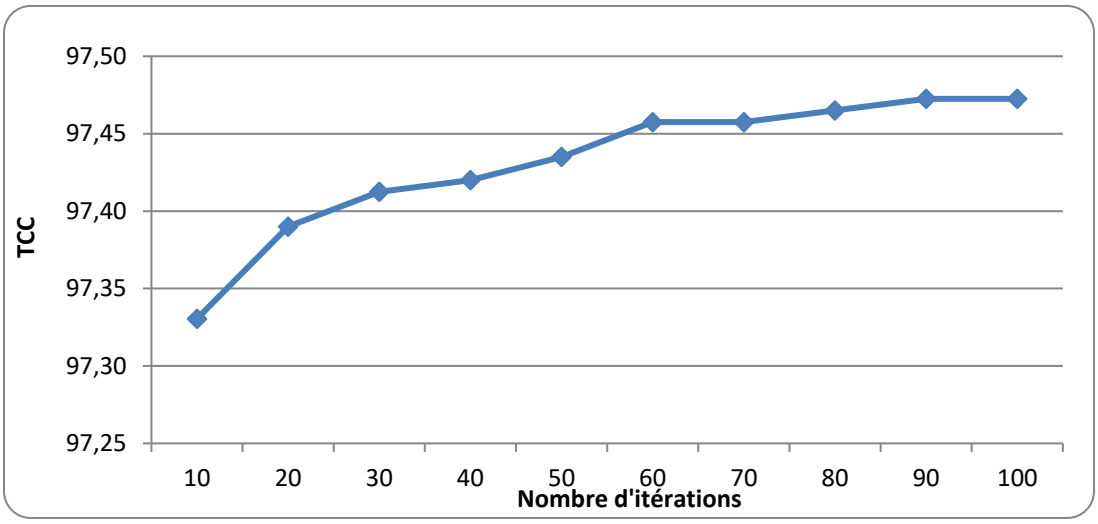


Figure IV. 3 : *Graphe de la moyenne TCC en fonction de nombre d'itérations*

En analysant le graphe de la Figure IV. 3 nous remarquons l'amélioration du taux de classification avec l'augmentation des nombres d'itérations.

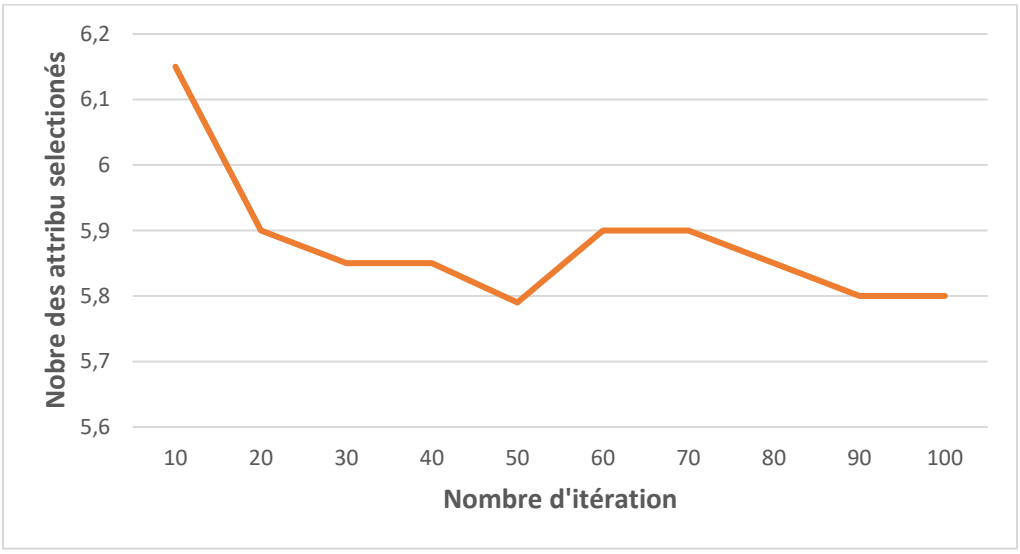


Figure IV. 4 : *Graphe de la moyenne du nombre d'attributs sélectionnés en fonction de nombre d'itérations*

En analysant le graphe de la Figure IV. 4 nous remarquons la diminution du nombre d'attributs sélectionnées avec l'augmentation des nombres d'itérations.

❖ La base Duke_Breast_Cancer:

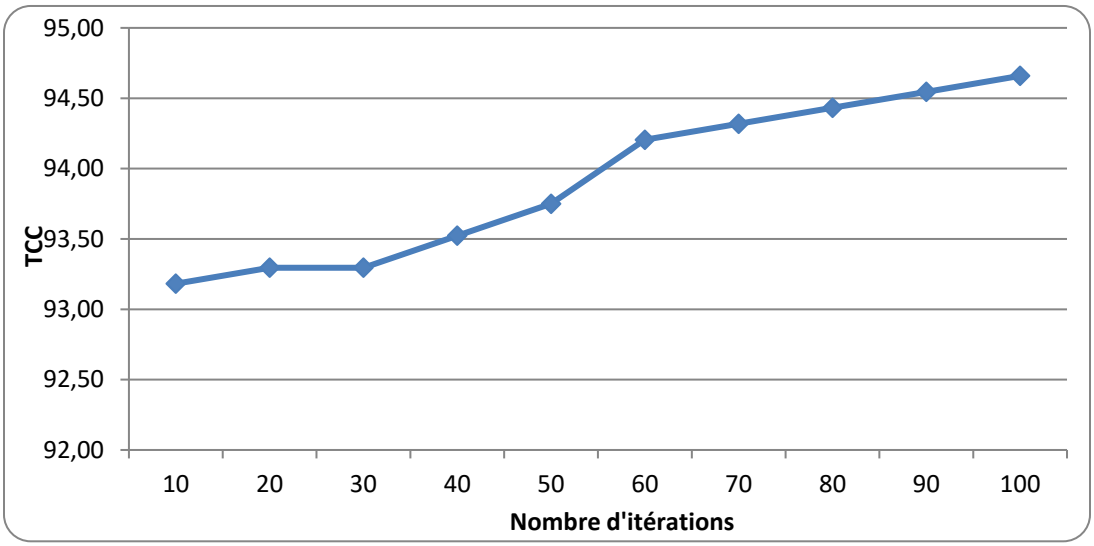


Figure IV. 5 : *Graphe de la moyenne TCC en fonction de nombre d'itérations*

En analysant le graphe de la Figure IV. 5 nous remarquons l'amélioration du taux de classification avec l'augmentation des nombres d'itérations.

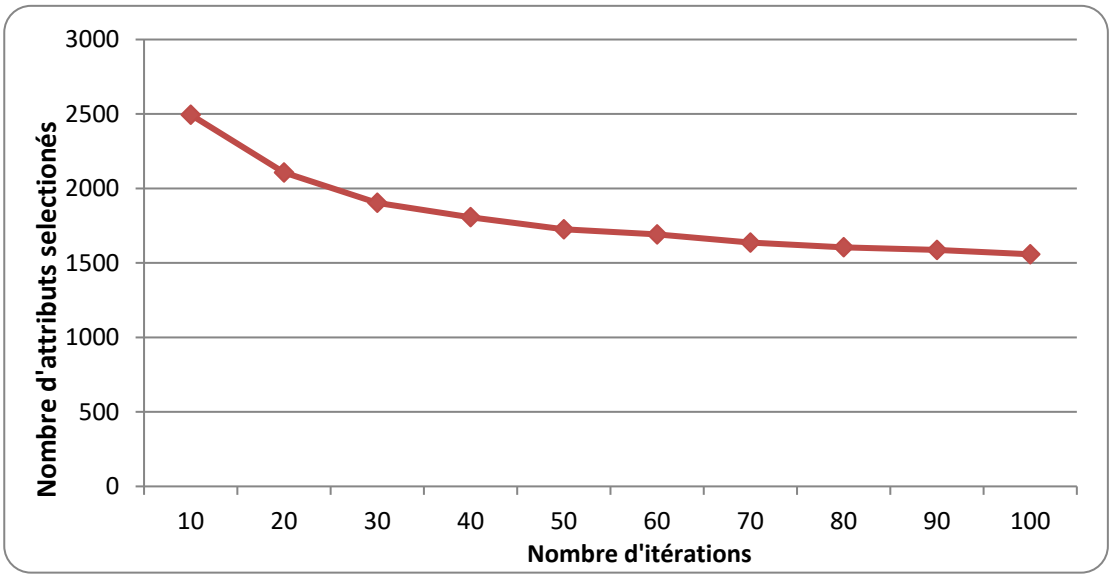


Figure IV. 6 : *Graphe de la moyenne du nombre d'attributs sélectionnés en fonction de nombre d'itérations*

En analysant le graphe de la Figure IV. 6 nous remarquons la diminution du nombre d'attributs sélectionnées avec l'augmentation des nombres d'itérations.

❖ La base Wine:

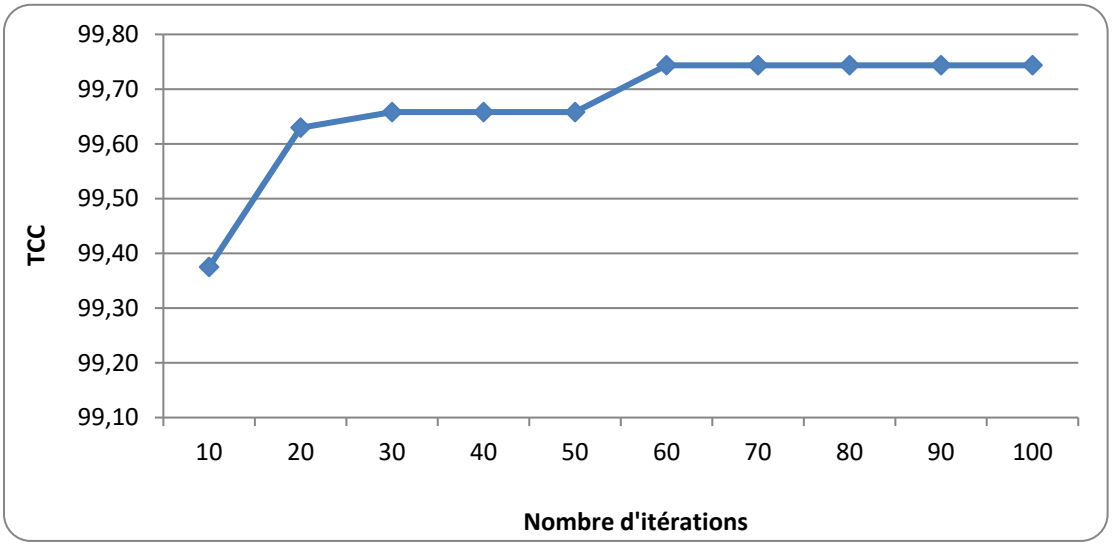


Figure IV. 7 : *Graphe de la moyenne TCC en fonction de nombre d'itérations*

En analysant le graphe de la Figure IV. 7 nous remarquons l'amélioration du taux de classification avec l'augmentation des nombres d'itérations.

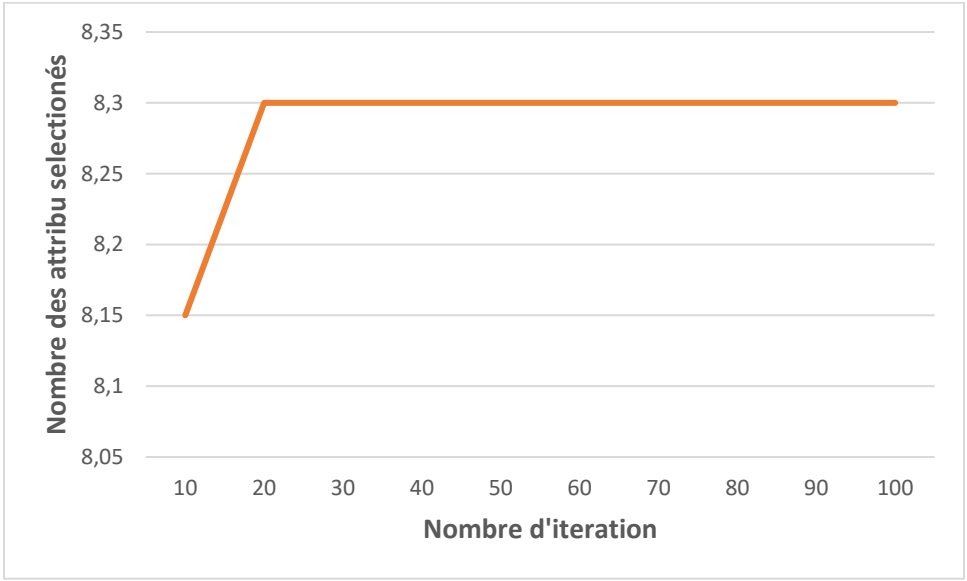


Figure IV. 8 : *Graphe de la moyenne du nombre d'attributs sélectionnés en fonction de nombre d'itérations*

En analysant le graphe de la Figure IV.8 nous remarquons la stabilité du nombre d'attributs sélectionnées avec l'augmentation des nombres d'itérations.

❖ La base Véhicule :

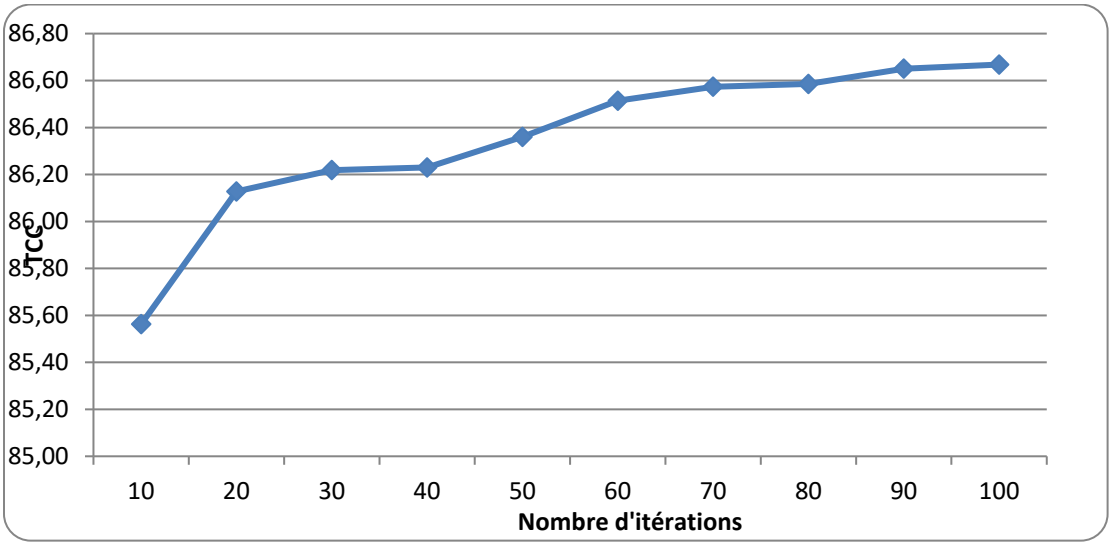


Figure IV. 9 : *Graphe de la moyenne TCC en fonction de nombre d'itérations*

En analysant le graphe de la Figure IV. 9 nous remarquons l'amélioration du taux de classification avec l'augmentation des nombres d'itérations.

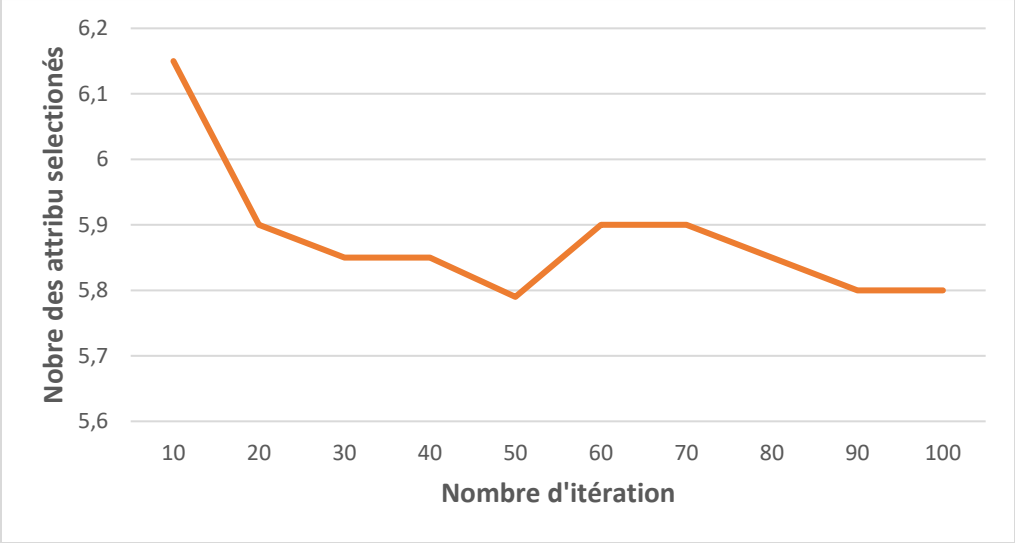


Figure IV. 10 : *Graphe de la moyenne du nombre d'attributs sélectionnés en fonction de nombre d'itérations*

En analysant le graphe de la Figure IV. 10 nous remarquons la diminution du nombre d'attributs sélectionnées avec l'augmentation des nombres d'itérations.

❖ La base Diabète:

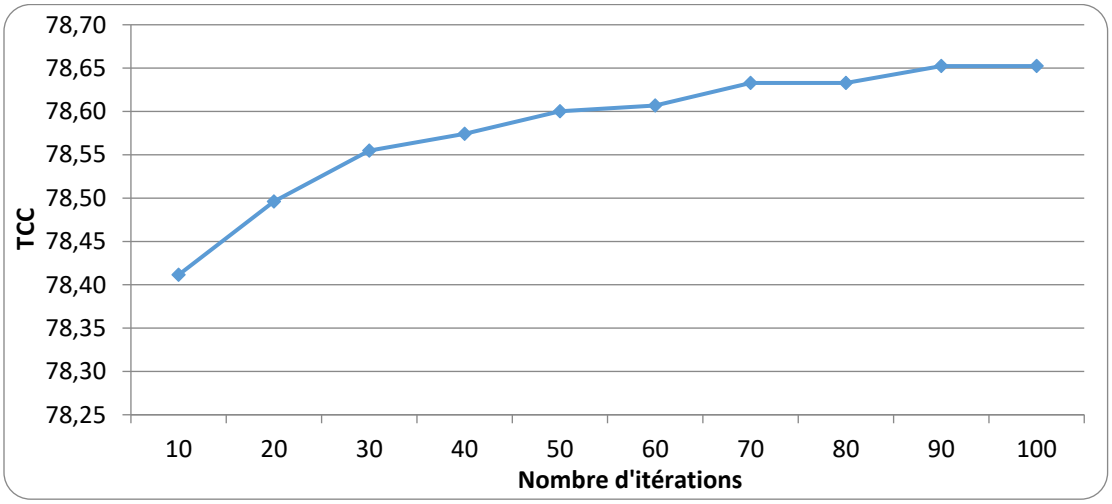


Figure IV. 11 : *Graphe de la moyenne de 1TCC en fonction de nombre d'itérations*

En analysant le graphe de la Figure IV. 11 nous remarquons l'amélioration du taux de classification avec l'augmentation des nombres d'itérations.

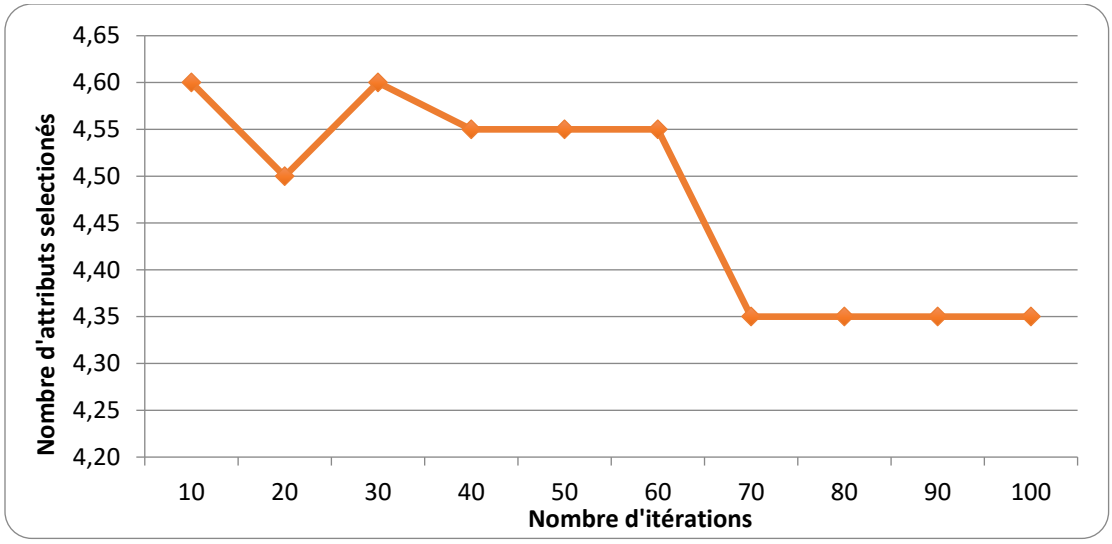


Figure IV. 12 : *Graphe de la moyenne du nombre d'attributs sélectionnés en fonction de nombre d'itérations*

En analysant le graphe de la Figure IV. 12 nous remarquons la diminution du nombre d'attributs sélectionnées avec l'augmentation des nombres d'itérations.

❖ La base Sonar:

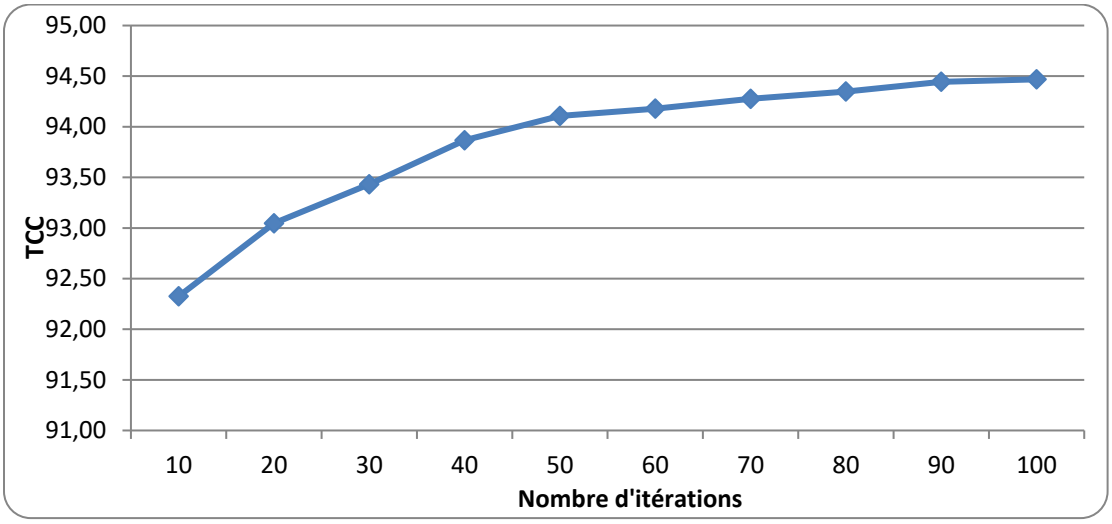


Figure IV. 13 : *Graphe de la moyenne de 1TCC en fonction de nombre d'itérations*

En analysant le graphe de la Figure IV. 13 nous remarquons l'amélioration du taux de classification avec l'augmentation des nombres d'itérations.

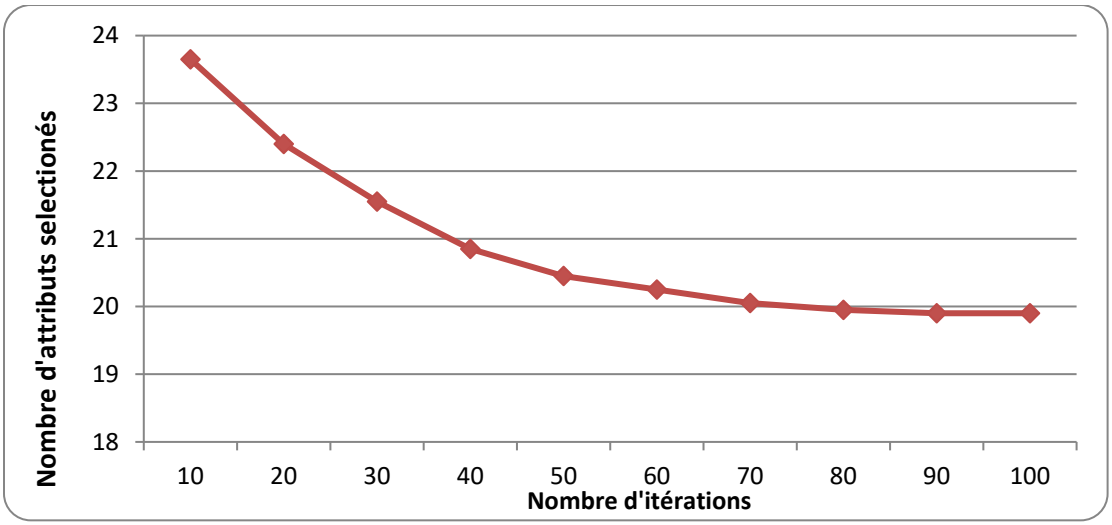


Figure IV. 14 : *Graphe de la moyenne du nombre d'attributs sélectionnés en fonction de nombre d'itérations*

En analysant le graphe de la Figure IV. 13 nous remarquons la diminution du nombre d'attributs sélectionnées avec l'augmentation des nombres d'itérations.

❖ **La base Leukemia_glob_cancer :**

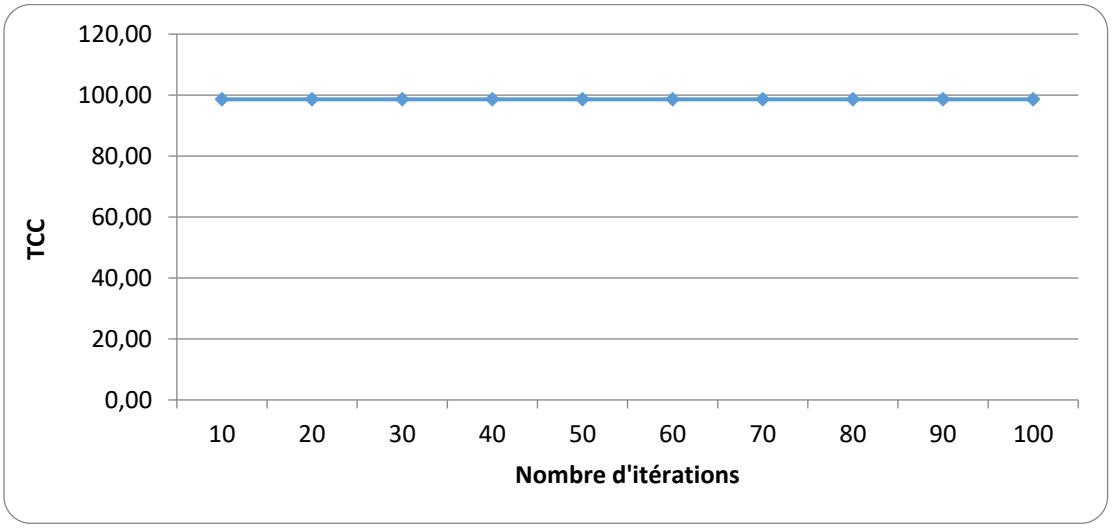


Figure IV. 15 : *Graphe de la moyenne de taux de classification en fonction de nombre d'itérations*

En analysant le graphe de la Figure IV. 15 nous remarquons l'amélioration du taux de classification jusqu'à 100% avec l'augmentation des nombres d'itérations.

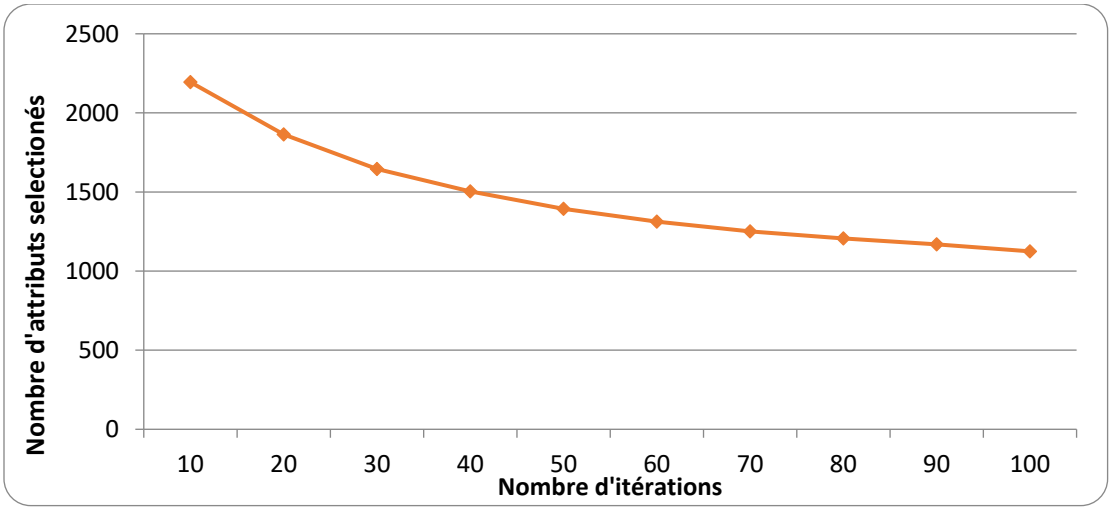


Figure IV. 16 : *Graphe de la moyenne du nombre d'attributs selectionés en fonction de nombre d'itérations*

En analysant le graphe de la Figure IV. 16 nous remarquons la diminution du nombre d'attributs sélectionnées avec l'augmentation des nombres d'itérations.

Les résultats des expérimentations sont regroupés dans des tableaux qui permettent d'évaluer le comportement général de notre méthode FA-SVM testées avec 9 jeux de données et sur 100 itérations .Ces résultats sont présentés de manière graphique dans les Figures 1 à 16.

L'étude de ces résultats nous permet de tirer un certain nombre de remarques qui sont les suivantes :

- ✓ Amélioration de TCC avec toutes les bases de données choisis.
- ✓ Le TCC augmente avec l'augmentation du nombre des itérations avec toutes les bases de données choisis.
- ✓ Réduction du nombre d'attributs sélectionnés sur toutes les bases de données choisis.

7.4 Interprétation des résultats :

Une amélioration dans le TCC obtenu en sélectionnant un sous ensemble d'attributs par rapport à la totalité des attributs est justifiée par la présence des attributs peu informatifs ou inutiles au problème de la classification, leurs présence dans l'ensemble initiale contient un bruit donc leurs élimination améliore la classification.

7.5 Comparaison avec les résultats d'autres méthodes :

Dans le but d'évaluer l'efficacité de la méthode proposée FA-SVM, une comparaison des résultats expérimentaux obtenus par notre méthode avec les résultats des travaux cité dans [34], [35] est présentée dans le Tableau IV.8 suivant :

Tableau IV.8 : *Comparaison entre GA+SVM, MA+SVM, SLS+SVM et FA-SVM.*

	FA-SVM	GA-SVM	MA-SVM	SLS-SVM
	TCC Moy (%)	TCC Moy (%)	TCC Moy (%)	TCC Moy (%)
Breast_cancer	97,47	97.20	97.44	97.37
Heart_stat	86.11	84.11	84.56	84.56
Sonar	94.47	83.64	88.57	83.73
Véhicule	86.67	71.61	82.24	72.29
Diabetes	78.65	77.53	77.71	77.87

Le TCC obtenu par la méthode FA-SVM proposée est supérieur en comparaison avec d'autres approches (Algorithme génétique, Algorithme Mimétiques et le SLS) avec les jeux de données suivantes : Breast-cancer, Heart_stat, Sonar, Véhicule et Diabetes.

8 Discussion:

Les tests que nous avons effectués nous ont permis de sélectionner un modèle adapté pour chaque base de données et d'améliorer sa performance en sélectionnant un sous ensemble d'attributs pertinent. Les Taux de bonne classification obtenus atteignent 100% ,99.74%, 97.47%, 94.47%, 94.66%,86.67 respectivement sur les bases Colon-Cancer, Wine, Breast-Cancer, Sonar, Duke-Breast-Cancer et Véhicule.

Le nombre d'attributs après sélection est réduit considérablement dans toutes les bases de test en améliorant en même temps le taux de classification .Le nombre d'attributs a été réduit

- jusqu'à 1558.30 (Moyenne de 20 exécutions) sur 7129 d'attributs d'origine pour la base Duke-Breast-Cancer.
- jusqu'à 434.05 sur 2000 pour la base Colon-Cancer.
- jusqu'à 5.80 et 4.35 sur 10 pour les bases Breast-Cancer et Diabète respectivement.
- jusqu'à 19.90 sur 60 pour la base Sonar.

9 Conclusion

Les résultats obtenus indiquent que la méthode FA-SVM est compétitive par rapport aux autres métaheuristiques (Algorithme génétique, Algorithme Mimétiques et le SLS) pour la sélection d'attributs, et les expérimentations nous ont montré que cette méthode est efficace. Nous avons évalué notre méthode suivant plusieurs axes :

- Notre méthode permet de traiter les données volumineuses et bruitées ; et par conséquent permet de réduire conséquemment la taille de l'espace de représentation ;
- Elle permet relativement souvent d'améliorer la qualité d'apprentissage ou tout au moins de la garder constante et de garantir la stabilité du modèle.
- L'utilisabilité de notre méthode semble bonne : le paramétrage de l'algorithme est facile, intelligible, intuitif et peut permettre l'intervention de l'utilisateur ou d'un expert du domaine, la présentation des résultats est explicite et intelligible.

Conclusion Générale

Les métaheuristiques forment un ensemble de méthodes utilisées en recherche opérationnelle et en intelligence artificielle pour résoudre des problèmes d'optimisation réputés difficiles..En utilisant une version binaire de l'algorithme de luciole qui est une métaheuristique bio-inspirée récene. Nous avons essayé dans ce memoire d'appliquer cette algorithme dans la sélection d'attributs avant la classification supervisé .

Afin d'aboutir à notre objectif,nous avons à travers une recherche bibliographique vue les principales notions de la selection d'attributs et les différentes méthodes classiques et les métaheuristiques appliquées dans selection d'attributs. Ensuite, Une étude plus approfondis a été faite sur l'algorithme de luciole et la méthode de clasification SVM. Finalement, nous avons proposé la méthode FA-SVM de type envloppe pour la sélection d'attributs en utilisant l'algorithme de luciole combiné avec le classificaur SVM en donnant les détails de sa conception et de sa implémentation.

La méthode FA-SVM proposé constitue une méthode simple, cependant puissante et présente les points forts suivantes :

1. L'utilisation d'une métaheuristique récente issue de l'intelligence en essaims qui a donné des bons résultats presque dans tous les domaines où elle était appliquée. (Algorithme de luciole).
2. L'utilisation d'un classifieur de haute performance grâce à ses fondements mathématiques à savoir la maximisation de la marge et l'utilisation de la fonction noyaux. (Le classifieur SVM).
3. Grâce à la programmation orienté objet en utilisant le langage java, l'implémentation de la méthode proposée peut être exécuté sur différents calculateurs fonctionnants avec différents systèmes d'exploitations
4. L'utilisation d'un mécanisme pour trouver les meilleures valeurs possibles des paramètres de FA-SVM. Dans notre travail, les paramètres du classifieur SVM utilisée sont réglés en utilisant une grille de recherche (Grid-Search).
5. L'utilisation de plusieurs bases de données publiques de petites, moyenne et de grandes tailles pour l'évaluation de la méthode de sélection d'attributs FA-SVM proposée a prouvé que notre méthode permet de traiter les données volumineuses et bruitées ; et par conséquent permet de réduire considérablement la taille de l'espace de

représentation en éliminant le bruit et la redondance. Aussi elle permet relativement souvent d'améliorer la qualité d'apprentissage ou tout au moins de la garder constante et de garantir la stabilité du modèle.

Comme perspectives de recherche et suite de ce travail, il serait souhaitable de travailler sur la réduction du temps d'exécution et de calcul en proposant une implémentation parallèle de la méthode FA-SVM proposée. Il est aussi possible d'utiliser l'algorithme de luciole avec d'autres algorithmes de classification tels que les réseaux de neurones, le plus proche voisin...

En effet, suite à notre étude et au constat de la rareté des travaux incluant les essaims de lucioles pour la sélection d'attributs, nous proposons l'utilisation et, éventuellement, l'hybridation de l'algorithme de luciole avec d'autres méthodes d'intelligence artificielle.

Bibliographie et Webographie

[1] Xin-She Yang ‹‹ Firefly Algorithm, Stochastic Test Functions and Design Optimization ›› , Int. J. Bio-Inspired Computation, Vol. 2, No. 2, pp.78–84(2010).

[2] Xin-She Yang ‹‹Firefly Algorithms for Multimodal Optimization ›› Department of Engineering-University of Cambridge,-Trumpington Street, Cambridge CB2 1PZ, UK.

[3] Xin-She Yang, Iztok Fister, Iztok Fister Jr et Janez Brest ‹‹ A comprehensive review of firefly algorithms ›› University of Maribor, Faculty of electrical engineering and computer science, Smetanova 17, 2000 Maribor, Slovenia. School of Science and Technology, Middlesex University, London NW4 4BT, UK.

[4] Xin-She Yang, ‹‹Nature Inspired Metaheuristic Algorithms Second Edition ›› University of combridge UK (2010).

[5] Xin-She Yang and Xing shi He, ‹‹Firefly Algorithm: Recent Advances and Applications ››, Int. J. Swarm Intelligence, Vol. 1, No. 1, pp. 36–50. DOI: 10.1504/IJSI.2013.055801 (2013).

[6] Xin-She Yang ‹‹Firefly Algorithm Lèvy Flights and Global Optimization ›› Research and Development in Intelligent Systems XXVI (EdsM. Bramer, R. Ellis, M. Petridis), Springer London, pp. 209-218 (2010).

[7] Broderick Crawford, Ricardo Soto, Miguel Olivares-Suárez and Fernando Paredes ‹‹ A Binary Firefly Algorithm for the Set Covering Problem ››.

[8] Latifa DEKHICI, Khaled BELKADI ‹‹Métaheuristique des Lucioles Discrète pour les Flow Shop Hybrides ›› Laboratoire LAMOSI - Département d'informatique Université U.S.T.O.M.B. -31000 Oran –ALGERIE

[9] Philippe Preux ‹‹Fouille de données Notes de cours ›› Université de Lille 3- 26 mai 2011.Disponibe sur le lien : http://www.grappa.univ-lille3.fr/~ppreux/fouille.pdf

[10] Miller, Harvey. J et Jiawei, Han. ‹‹ Geographic data mining and knowledge discovery: an overview››. New York : Taylor & Francis, 2001.

[11] Vincent Gardeux ‹‹ Conception d'heuristiques d'optimisation pour les problèmes de grande dimension : application à l'analyse de données de puces à ADN ›› Thèse de doctorat en informatique -Université ParisEst, 2011. Français. <NNT : 2011PEST1022>. <tel-00676449>

[12] Mouhamadou Lamine SAMB, Fodé CAMARA, Samba NDIAYE Yahya SLIMANI, MohamedAmir ESSEGHIR‹‹Approche de sélection d'attributs pour la classification basée sur l'algorithme RFE-SVM ›› ARIMA Journal, vol. 17 (2014), pp. 197-219.

[13] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/ Consultéele 15 février 2016.

[14] MENGHOUR Kamilia ‹‹ Approches Bio-inspirées pour la Sélection d'Attributs ›› Thèse de doctorat en STIC -UNIVERSITE BADJI MOKHTAR de ANNABA (2014-2015).

[15] Jin-Kao Hao, Philippe Galinier, Michel Habib ‹‹ Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes›› Revue d'Intelligence Artificielle Vol : No. 1999.

[16] Dr. LEMOUARI ALI ‹‹ Introduction aux Métaheuristique ›› Spécialités Système SIAD et IA -université de Jijel (2014).

[17] Surafel Luleseged Tilahun,Jean Medard T Ngnotchouye ‹‹ Abstract FireFlyy Algorithm for optimization problems with non-continuous variables: A Review and Analysis ›› University of KwaZulu-Natal article de 25/02/2016.

[18] José Crispin Hernandez ‹‹Hybrid metaheuristics algorithms for gene selection and classification of microarray data ›› Computer Science. University d'Angers, 2008. French.

[19] The Gartner Group , www.gartner.com

[20] Cécile Caponi ‹‹Introduction à la fouille de données ››- Université Aix-Marseille.

[21] https://netbeans.org

[22] Martin Sewell. ‹‹ History of Support Vector Machines ››. Disponible en ligne sur : http://www.svms.org/history.html.

[23] Buxian zhang ‹‹ Apprentissage à base noyau ›› Ecole Telecom Paris.

[24] Stéphan Clémençon ‹‹ Classification supervisée et SVM ›› (2011)

[25] Millonas M., “Swarms, Phase Transitions, and Collective Intelligence”. In: Langton, C.G. (ed.): “Artificial Life III”, Vol. 17, pp. 417–445. Addison- Wesley Publishing Company, Reading. 1994

[26] LAETITIA JORDAN ‹‹ Métaheuristique pour l’extraction de connaissance : application à la génomique ›› Thèse de doctorat en informatique -UNIVERSITE de Lille1(2003).

[27] E. Bonabeau, M. Dorigo, G. Theraulaz, “Swarm Intelligence: From Natural to Artificial Systems”. New York: Oxford University Press.

[28] https://www.csie.ntu.edu.tw/~cjlin/libsvm/ consulté le 01/27/2016

[29] Chapitre de Sélection de variable disponible sur le site theses.univ-lyon2.fr/documents/lyon2/2004/legrand_g/.../legrand_g_chapitre02.pdf Consulté le 01/03/2016

[30] ALLAOUI Abdya ‹‹ Application des techniques des métaheuristiques pour l’optimisation de la tâche de la classification de la fouille de données ›› Thèse de Magister en informatique -UNIVERSITE d’Oran (2012).

[31] Dash M. and Liu H. “Feature Selection Methods for Classification”, Intelligent Data Analysis, Vol. 1, pp 131–156. 1997

[32] Liu H., Yu L. “Toward Integrating Feature Selection Algorithms for Classification and Clustering”, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No 4, pp 491–502. 2005.

[33] Ali El Akadi ‹‹ contribution à la sélection de variables pertinentes en classification supervisée : Application à la sélection des gènes pour les puces à ADN et des caractéristiques faciales ›› Thèse de doctorat en informatique -UNIVERSITE de Mohammed V_MAROC (2012).

[34] Messaouda Nekkaa, Dalila Boughaci ‹‹ A memetic algorithm with support vector machine for feature›› IREGULAR RESEARCH PAPER Publié le : 7 Fevrier 2015

[35] Messaouda Nekkaa, Dalila Boughaci‹‹ sélection and classification Hybrid Harmony Search Combined with Stochastic Local Search for Feature Selection ››IREGULAR RESEARCH PAPER Publié le : 26 Juin 2015