

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE ABOU-BEKR BELKAID – TLEMCCEN



THÈSE

Présentée à :

FACULTE DES SCIENCES – DEPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

DOCTORAT

Spécialité: *Informatique*

Par :

BOUBLENZA Amina
(Ep. YUCEF)

Sur le thème

Coopération entre classifieurs hétérogènes pour la reconnaissance des données médicales

Soutenue publiquement le 22 février 2017 à Tlemcen devant le jury composé de :

| | | | |
|----------------------------------|------------|-------------------------|--------------------|
| Pr. CHIKH Azzedine | Professeur | Université de Tlemcen | Président |
| Pr. CHOUARFIA Abdellah | Professeur | Université d'Oran -USTO | Examineur |
| Dr. MERAD Lotfi | MCA | EPST de Tlemcen | Examineur |
| Dr. BENAMAR Abdelkrim | MCA | Université de Tlemcen | Examineur |
| Dr. ABDERRAHIM Mohammed El-Amine | MCA | Université de Tlemcen | Examineur |
| Dr. BENZAOUZ Mortada | MCB | Université de Tlemcen | Invité |
| Mr. CHIKH Mohammed Amine | Professeur | Université de Tlemcen | Directeur de thèse |

Résumé

Durant ces dernières années, le recours de la médecine moderne à l'intelligence artificielle ne cesse d'augmenter. L'avènement de cette technique a permis d'aboutir à des outils de plus en plus efficaces qui appuient efficacement le diagnostic médical et la prise de décision. Une des applications les plus émergentes est la détection fiable et automatisée des pathologies. Cela a fait l'objet de recherches algorithmiques très actives dans ce que l'on appelle la classification des données médicales.

Les études théoriques et pratiques ont démontré que la classification ensembliste émet des résultats très prometteurs. Par ailleurs, la construction d'un ensemble de classifieurs performants consiste à sélectionner individuellement les membres de l'ensemble en se basant sur des critères prédéfinis. En outre, le choix des attributs précédant la classification est d'une importance cruciale dans le déroulement du processus.

L'objectif de cette thèse est d'étudier et d'analyser le compromis entre le niveau de précision individuelle de chaque classifieur et la diversité mutuelle entre chaque paire de classifieurs hétérogènes. L'accent est porté également sur la sélection d'attributs et son impact sur la performance du résultat de classification finale. Dans ce contexte, nous proposons une nouvelle implémentation qui a pour objectif la détection de différentes pathologies. La solution proposée est validée expérimentalement sur des bases de données biomédicales réelles. Les résultats sont comparés à l'état de l'art et démontrent l'efficacité de l'implémentation proposée.

Mots-clés: Apprentissage automatique, classification supervisée, sélection d'attributs, diversité, ensemble de classifieurs, taux d'erreur.

Abstract

Over recent years, the use of artificial intelligence in modern medicine becomes increasingly important. This new technology has allowed the development of reliable and automated tools which significantly assists medical diagnosis and decision making. Multiple fields, such as the detection of diseases, have been subject to very active algorithmic researches on the topic of medical data classification.

Theoretical and empirical studies have demonstrated that ensemble of classifiers produces accurate and very promising results. Otherwise, building a efficient set of classifiers consists in selecting the individual members, based on predefined criteria. Moreover, choosing attributes prior to the classification is of major importance in the performance of the general process.

The work presented in this thesis aims to study and analyze the compromise between the level of the individual accuracy of each classifier and the mutual diversity between every heterogenous classifier pair. We particularly focused on the selection of attributes and its impact on the performance of the final classification result. In this context, we propose a new approach which aims to improve the recognition of several diseases. The proposed solution is validated over real biomedical data basis. The results are compared to the state of the art, and demonstrate the efficiency of the proposed approach.

Key words: machine learning, supervised classification, attribute selection, diversity, ensemble or set of classifiers, Q-statistic, classification error.

ملخص

في السنوات الأخيرة، تزايد استخدام الذكاء الاصطناعي في الطب الحديث بشكل ملفت للانتباه، وأخذ أبعاداً محورية في برمجة وسائل فعالة في التشخيص والفحوصات واتخاذ القرارات الطبية في مجالات متعددة كالكشف عن مختلف الأمراض. وقد قام الاخصائيون ببحوث مكثفة في مجال تصنيف البيانات الطبية بطريقة تلقائية. وأظهرت الدراسات النظرية والعملية أنّ الاعتماد على مجموعة من المصنفات تعطي نتائج دقيقة وواحدة جداً. من ناحية اخرى فان بناء مجموعة عضوية فعالة يتمثل في اختيار كل عضو بشكل فردي من بين مجموعة عامة من المصنفات بناء على معايير محددة مسبقاً. وبالإضافة إلى ذلك فإن اختيار السمات، قبل مباشرة عملية التصنيف له أهمية حاسمة في تحقيق النتائج المنتظرة

يهدف العمل المقدم في هذه الأطروحة إلى دراسة الموازنة ما بين الدقة الفردية لكل مصنف والمميزات الناتجة عن استعمال التنوع في مجموعات المصنفات. وفي سياق آخر، فقد اجرينا دراسة شاملة ومفصلة عن عملية انتقاء السمات. وما ينتج عنها في ما يخص فعالية التصنيف النهائية

في البداية تم التركيز على مناقشة واختبار طرق اختيار السمات الأكثر استعمالاً في الكشف عن دقات القلب الغير منتظمة. ويتبع هذا الجزء بوصف مختلف مخططات ضم المصنفات فضلاً عن أهمية خطوة انتقاء المصنفات قبل الشروع في عملية الدمج. ونقدم في الجزء الأخير طريقة جديدة تعتمد على مبدأ تنوع التصنيف مع معاينتها في قاعدة بيانات بيوطبية حقيقية. ونهدف من خلال هذا الاقتراح تحسين فعالية الكشف عن الأمراض بصفة عامة. حيث يتم استخدام مصنفات غير متجانسة لبناء المجموعة، بالاستفادة من مختلف القرارات الناتجة عنها، حتى لو كان بعضها خاطئ، مما يؤدي الى بناء المجموعة الامثل. يتم تحديد هذه المجموعة وفقاً للأداء الفردي لكل عضو أثناء مرحلة التعلم ووفقاً للتنوع البيئي من خلال تقييم التبعية الاحصائية. يذكر ان ضم مجموعة من المصنفات المتنوعة يتم عن طريق الاستعانة بقواعد ثابتة كالمعدل والتصويت بالأغلبية، وغيرها. وننوه الى ان نتائج العمل المقترح فورنت بعدة طرق موجودة في مختلف الابحاث العلمية، مما مكننا من إثبات فعالية استعمال التنوع بين مجموعة المصنفات في الطريقة المقترحة.

Remerciements

Au-delà de la formalité d'usage, c'est avec un grand plaisir que je remercie en premier lieu mon directeur de thèse, Monsieur CHIKH Mohammed Al Amine, pour la confiance qu'il m'a accordée en acceptant d'encadrer ce travail doctoral. Je souhaite également lui exprimer ma gratitude pour tous ses conseils, ses orientations et la qualité des commentaires et suggestions dont il m'a fait part. J'ai été extrêmement sensible à ses qualités humaines d'écoute et de compréhension tout au long de ce travail.

J'adresse toute ma gratitude et mes remerciements à M. Bernard ESPINASSE Professeur des Universités en informatique à l'Ecole Polytechnique Universitaire de Marseille, France, pour son accueil chaleureux tout au long de mon séjour au sein du laboratoire LSIS (Laboratoire des Sciences de l'Information et des Systèmes). Je me souviendrai toujours de son aide et ses conseils ainsi que celle de son équipe.

Je remercie vivement M. CHIKH Azzedine pour m'avoir fait l'honneur de présider mon jury de thèse. Mes vifs remerciements vont également à M. CHOUARFIA, M. MERAD, M. BENAMAR et M. ABDERRAHIM pour avoir bien voulu de prendre le temps d'examiner cette thèse. Je remercie également M. BENAZZOUZ pour avoir accepté notre invitation.

Je souhaiterais exprimer ma gratitude à Pr. Breksi Reguig Fethi, directeur du laboratoire GBM Tlemcen et Pr Bessaid Abdelhafid responsable de l'équipe Imagerie Médicale. Je remercie aussi les membres du laboratoire GBM qui sont des amis avant d'être des collègues. A Ismahene, Amel, Ibtissem, Mohammed, Amine et Omar, Redouane et Souad. Une pensée particulière pour l'équipe CREDOM qui ont pu mouvementé notre fameux sous-sol : Sarah, Khalida, Asma Amine, Mustapha, Ismahène et Imane Nedjar.

Je tiens à exprimer ma gratitude à tous les enseignants du département d'informatique à qui je dois mon respect et ma reconnaissance ;

A l'issue de la rédaction de cette recherche, j'ai la certitude que la thèse n'est pas uniquement une expérience scientifique, car, si l'épreuve est individuelle, ses implications sont sociales, académiques, familiales, et tout simplement humaines. je n'aurais jamais pu

réaliser ce travail ni supporter la pression et le stress sans le soutien d'un grand nombre de personnes dont la générosité, la bonne humeur et leur sympathie m'ont permis de progresser dans mes recherches.

Mes sincères remerciements à mes voisins « les sous-soliens » pour nos nombreuses pauses café partagées avec mes meilleures amies Wafaa Abderrahim, Amina Benosman, Sarah Bouchikhi, Houaria Badir, Nesma Settouti et Soumia Kherbouche. Et surtout un grand Merci pour la petite cérémonie de mon départ ^^ . Un clin d'œil à mon amie d'enfance Dr. Wafaa Abdelli.

C'est avec un énorme plaisir que je remercie mon cher mari, mon âme-sœur et mon meilleur ami, Hakim, pour ses nombreuses corrections, suggestions et encouragements. Je n'aurais jamais pu terminer ce travail sans sa compréhension, ses sacrifices et toute son aide afin de me laisser consacrer un maximum de temps pour ma thèse. Je remercie également toute ma belle famille et en particulier mes beaux parents, mes belles-sœurs Amina et Ikram, mes beaux-frères Mohammed, Abdou, Yacine et Brahim ainsi que mon petit neveu pour leur soutien et encouragements.

Du fond du cœur je remercie infiniment toute ma famille, ma véritable source de bonheur. Je remercie mes sœurs qui n'ont jamais cessé de me soutenir et de m'apporter leur aide, Lamia, pour les nombreux services qu'elle me rend au quotidien, Latifa pour sa générosité et ses délicieuses pâtisseries, Nassima pour ses leçons de moral, Ikram, ma confidente, pour nos nombreuses discussions nocturnes et Nesrine ma sœur jumelle avec qui j'ai passé toute mon enfance. Je remercie aussi toute ma famille, mes beaux-frères, mes neveux et particulièrement ma grand-mère, Henia, pour toutes ses prières intarissables. Je ne remercierai jamais assez les meilleurs et plus tendre parents au monde « Papati & Mamati », pour leur amour et dévouement. Je les remercie aussi pour leur confiance qui me pousse toujours à me surpasser et donner le meilleur de moi-même. Mon souhait dans la vie c'est qu'ils soient toujours fiers de moi.

Mes chaleureux remerciements pour mon rayon de soleil, ma petite princesse, Dina, qui me couvre de joie et tendresse. Son sourire, sa bonne humeur et ses calins me remontent toujours le moral.

Je dédie cette thèse de doctorat à :
Mes parents et mes sœurs ;
Mes neveux et nièces ;
Ma belle famille ;
Ma grand-mère ;
Mon mari ;
Et Dina.

*« Les défis sont ce qui rend la vie intéressante
les surmonter est ce qui lui donne du sens. »*
Joshua Marine

Table des matières

| | |
|--------------------------|------|
| Résumé | ii |
| Abstract..... | iii |
| ملخص..... | iv |
| Remerciements | v |
| Table des matières | viii |
| Liste des tableaux | xii |
| Liste des figures..... | xiii |

| | |
|-----------------------------------|----------|
| Introduction générale..... | 1 |
|-----------------------------------|----------|

| | |
|---|----------|
| Chapitre 1. Sélection des attributs dans le domaine biomédical | 4 |
|---|----------|

| | |
|---|----|
| 1.1. Introduction | 5 |
| 1.2. Différence entre sélection et extraction des variables | 6 |
| 1.3. La sélection d'attributs | 7 |
| 1.3.1. Motivation | 7 |
| 1.3.2. Notions de pertinence d'un attribut | 8 |
| 1.3.3. Notions de redondance d'un attribut | 8 |
| 1.4. Procédure de sélection d'attributs | 9 |
| 1.4.1. La procédure de génération | 9 |
| 1.4.2. La fonction d'évaluation | 11 |
| 1.4.2.1. Approche wrapper | 11 |
| 1.4.2.2. Approche filter | 13 |
| 1.4.2.3. Approche embedded | 15 |
| 1.4.3. Le critère d'arrêt | 16 |
| 1.4.4. La validation | 16 |
| 1.5. Techniques de l'apprentissage automatique | 17 |
| 1.5.1. Catégories de l'apprentissage automatique | 17 |
| 1.5.2. Problème de classification | 19 |
| 1.5.3. Validation et évaluation des classifieurs | 21 |
| 1.6. Etat de l'art de la sélection d'attributs dans le domaine biomédical | 23 |
| 1.7. Conclusion | 25 |

| | |
|--|-----------|
| Chapitre 2. Etude des méthodes de sélection d'attributs | 27 |
| 2.1. Introduction | 28 |
| 2.2. Matériels et méthodes | 29 |
| 2.2.1. Présentation de l'ensemble de données | 29 |
| 2.2.2. Classifieurs utilisés | 31 |
| 2.2.3. Evaluation de la base de donnée originale..... | 31 |
| 2.2.4. Prétraitement de la base | 32 |
| 2.2.5. Utilisation de la base dans la littérature..... | 34 |
| 2.3. Sélection des attributs avec l'approche Filter | 37 |
| 2.3.1. Approche univarié | 37 |
| 2.3.1.1. Méthode Gain d'information | 37 |
| 2.3.1.2. Méthode GainRatio | 38 |
| 2.3.1.3. Méthode Khi2..... | 39 |
| 2.3.1.4. Méthode ReliefF..... | 40 |
| 2.3.1.5. Méthode Incertitude symétrique..... | 41 |
| 2.3.1.6. Méthode SVM | 41 |
| 2.3.1.7. Méthode OneR..... | 42 |
| 2.3.2. Approche multivarié | 42 |
| 2.3.2.1. Méthode CFS..... | 43 |
| a. Algorithme BestFirst..... | 43 |
| b. Algorithme GeneticSearch..... | 44 |
| c. Algorithme GreedyStepwise | 44 |
| 2.3.2.2. Méthode de consistance | 44 |
| d. Algorithme de recherche aléatoire..... | 45 |
| 2.4. Résultats obtenus | 45 |
| 2.4.1. Filtre univarié | 46 |
| 2.4.1.1. Méthode InfoGain..... | 46 |
| 2.4.1.2. Méthode GainRatio | 46 |
| 2.4.1.3. Méthode Chi2 | 47 |
| 2.4.1.4. Méthode ReliefF..... | 48 |
| 2.4.1.5. Méthode Incertitude symétrique..... | 49 |
| 2.4.1.6. Méthode SVM | 49 |
| 2.4.1.7. Méthode OneR..... | 50 |
| 2.4.2. Filtre multivarié | 53 |
| 2.4.2.1. Méthode CFS..... | 53 |
| 2.4.2.2. Méthode Consistance | 53 |
| 2.5. Discussion des résultats | 54 |

| | | |
|------|------------------|----|
| 2.6. | Conclusion | 58 |
|------|------------------|----|

Chapitre 3. Vers une classification ensembliste 59

| | | |
|----------|---|----|
| 3.1. | Introduction | 60 |
| 3.2. | Terminologies et taxonomies | 62 |
| 3.2.1. | Classifieur de base | 62 |
| 3.2.2. | Définitions et motivations | 63 |
| 3.2.3. | Structures des classifieurs ensembliste..... | 65 |
| 3.3. | Etat de l’art des approches ensemblistes | 72 |
| 3.3.1. | Ensemble de classifieurs homogènes | 72 |
| 3.3.2. | Ensemble de classifieurs hétérogènes | 74 |
| 3.4. | Critères de sélection des classifieurs de base | 76 |
| 3.4.1. | Algorithmes de recherches | 77 |
| 3.4.2. | Le rôle de la diversité pour la sélection des classifieurs..... | 78 |
| 3.4.3. | Quantification de la diversité..... | 79 |
| 3.4.3.1. | Mesures "pairwise" | 79 |
| 3.4.3.2. | Mesures "non-pairwise" | 82 |
| 3.5. | Conclusion..... | 82 |

Chapitre 4. Nouvelle approche ensembliste pour la détection des arythmies cardiaques 83

| | | |
|--------|--|----|
| 4.1. | Introduction | 84 |
| 4.2. | Matériels et méthodes..... | 85 |
| 4.3. | Fusion des classifieurs par règles fixes | 87 |
| 4.4. | Sélection des classifieurs de base | 89 |
| 4.4.1. | Sélection dynamique des classifieurs | 89 |
| 4.4.2. | Recherche <i>SingleBest</i> | 90 |
| 4.4.3. | Recherche <i>N-Best</i> | 90 |
| 4.4.4. | Recherche vers l’avant..... | 90 |
| 4.4.5. | Discussion des résultats | 91 |
| 4.5. | Méthode proposée pour la sélection des classifieurs de base..... | 92 |
| 4.5.1. | Description détaillée | 92 |
| 4.5.2. | Résultats et discussion | 94 |
| 4.6. | Etude comparative avec la littérature | 95 |
| 4.7. | Classification ensembliste des données biopuces..... | 98 |
| 4.7.1. | Présentation des bases de données considérées..... | 98 |
| 4.7.2. | Présentation générale des données biopuces | 99 |

| | | |
|--------|--|-----|
| 4.7.3. | Prétraitement des données | 101 |
| 4.7.4. | Résultats de la combinaison par règles fixes | 103 |
| 4.7.5. | Résultats obtenus en utilisant notre approche..... | 104 |
| 4.7.6. | Comparaison avec la littérature : données bipuces..... | 106 |
| 4.8. | Conclusion | 109 |

| | |
|---|------------|
| Conclusion générale et perspectives..... | 110 |
|---|------------|

| | |
|---|------------|
| Références bibliographiques..... | 113 |
|---|------------|

Liste des tableaux

| | |
|---|-----|
| Tableau 1-1 Techniques de sélection d'attributs [21] | 15 |
| Tableau 1-2 Matrice de confusion | 22 |
| Tableau 2-1 Attributs de la base « Arrhythmia » | 30 |
| Tableau 2-2 Abréviations des classifieurs utilisés | 31 |
| Tableau 2-3 Résultats obtenus sans prétraitement de la base de données | 32 |
| Tableau 2-4 Résultats obtenus avec ré-échantillonnage des données | 32 |
| Tableau 2-5 Travaux réalisés en utilisant la base de données « Arrhythmia » | 35 |
| Tableau 2-6 Score des meilleurs attributs en utilisant la méthode <i>InfoGain</i> | 38 |
| Tableau 2-7 Scores des meilleurs attributs en utilisant la méthode <i>GainRatio</i> | 39 |
| Tableau 2-8 Score des meilleurs attributs en utilisant la méthode <i>Khi2</i> | 40 |
| Tableau 2-9 Score des meilleurs attributs en utilisant la méthode <i>ReliefF</i> | 41 |
| Tableau 2-10 Score des meilleurs attributs en utilisant la méthode <i>SymmU</i> | 41 |
| Tableau 2-11 Rang meilleurs attributs en utilisant la méthode <i>SVM</i> | 42 |
| Tableau 2-12 Score des meilleurs attributs en utilisant le filtre <i>OneR</i> | 42 |
| Tableau 2-13 Erreur obtenue avec la méthode <i>InfoGain</i> | 46 |
| Tableau 2-14 Erreur obtenue avec la méthode <i>GainRatio</i> | 47 |
| Tableau 2-15 Erreur obtenue avec la méthode <i>Chi2</i> | 48 |
| Tableau 2-16 Erreur obtenue avec la méthode <i>ReliefF</i> | 48 |
| Tableau 2-17 Erreur obtenue avec la méthode <i>SymmU</i> | 49 |
| Tableau 2-18 Erreur obtenue avec la méthode <i>SVM</i> | 50 |
| Tableau 2-19 Erreur obtenue avec la méthode <i>OneR</i> | 50 |
| Tableau 2-20 Résultats obtenus en utilisant <i>CFS</i> | 53 |
| Tableau 2-21 Résultats obtenus en utilisant <i>ConsistencySubsetEval</i> | 53 |
| Tableau 2-22 Tableau de contingence 2x2 pour le test <i>McNemar</i> | 57 |
| Tableau 2-23 Comparaison statistique entre les meilleurs filtres..... | 58 |
| Tableau 3-1 Relation entre les sorties de deux classifieurs | 79 |
| Tableau 4-1 Abréviations des classifieurs utilisés | 85 |
| Tableau 4-2 Performances des classifieurs de base..... | 86 |
| Tableau 4-3 Erreurs obtenus en fusionnant les classifieurs par règles fixes | 89 |
| Tableau 4-4 L'erreur de la fusion en utilisant la recherche en avant | 91 |
| Tableau 4-5 Pseudo-code de l'algorithme proposé | 94 |
| Tableau 4-6 Comparaison avec les résultats de la littérature | 96 |
| Tableau 4-7 Résultats de la fusion par classifieurs homogènes | 97 |
| Tableau 4-8 Nombre d'attributs pour les différents ensembles de données..... | 102 |
| Tableau 4-9 Taux d'erreur des données biopuces en utilisant le filtre <i>CFS</i> | 102 |
| Tableau 4-10 fusion de tous les classifieurs de base par les règles fixes : application aux données biopuces..... | 103 |
| Tableau 4-11 Méthode proposée : application aux données biopuces | 104 |
| Tableau 4-12 Etat de l'art sur les données biopuces..... | 106 |

Liste des figures

| | | |
|------------|---|-----|
| Figure 1.1 | Principes d'extraction et de sélection des caractéristiques [6]..... | 6 |
| Figure 1.2 | Procédure de sélection d'attributs [16] | 9 |
| Figure 1.3 | Procédure de l'approche "wrapper" | 12 |
| Figure 1.4 | Procédure de l'approche "filter" | 13 |
| Figure 1.5 | Principe de la classification [24] | 20 |
| Figure 2.1 | Comparaison de l'erreur « avant » et « après » ré-échantillonnage..... | 33 |
| Figure 2.2 | Comparaison des classifieurs en appliquant les différentes méthodes..... | 52 |
| Figure 2.3 | Moyenne d'erreur en variant le nombre d'attributs | 55 |
| Figure 2.4 | Comparaison entre filtre avec l'approche multivarié..... | 56 |
| Figure 2.5 | Comparaison de l'erreur entre CFS+BF et SVM..... | 56 |
| Figure 3.1 | Processus général de fusion des données | 61 |
| Figure 3.2 | Illustration de l'espace partitionnement de la meilleure hypothèse avec un ensemble de classifieurs | 64 |
| Figure 3.3 | Panorama global des méthodes de combinaison parallèle des classifieurs selon les travaux de Zouari et al. [24]..... | 68 |
| Figure 3.4 | Panorama global des méthodes de combinaison des classifieurs selon Matteo et Giorgio [95]..... | 69 |
| Figure 3.5 | Modèles de consensus (unanimité, majorité et pluralité)..... | 71 |
| Figure 3.6 | Illustration schématique des trois méthodes d'ensemble homogènes populaires | 74 |
| Figure 4.1 | Erreur obtenue en utilisant la recherche <i>N-Best</i> | 90 |
| Figure 4.2 | Schéma global de l'approche proposée..... | 93 |
| Figure 4.3 | Résultats d'évaluation de l'algorithme de recherche proposé..... | 94 |
| Figure 4.4 | Comparaison du taux d'erreur entre la méthode proposée et la recherche <i>N-Best</i> | 95 |
| Figure 4.5 | Etapes d'une analyse par puces à ADN | 99 |
| Figure 4.6 | Processus d'acquisition de l'image | 101 |
| Figure 4.7 | Comparaison du taux d'erreur pour la détection des types de leucémies | 105 |
| Figure 4.8 | Comparaison du taux d'erreur pour la détection du cancer de prostate..... | 105 |
| Figure 4.9 | Comparaison du taux d'erreur pour la détection du cancer du colon | 106 |

Introduction générale

Le recours aux méthodes dites intelligentes pour le développement d'outils d'aide au diagnostic médical, ne cesse de progresser durant ces dernières années. En effet, la collecte d'informations, à partir des rapports médicaux et des bilans de différentes maladies, ont contribué à l'avènement des dossiers médicaux électroniques, et a ainsi conduit à la réalisation de nombreuses bases de données regroupant d'importantes quantités de données.

L'application des techniques de l'intelligence artificielle sur ces bases a permis une exploitation fiable du contenu d'information. La réussite de ces techniques dans différents domaines tels que la recherche sur le web ou encore la reconnaissance de la parole et de l'écriture, était une source de motivation pour étendre le cas d'usage au domaine médical. Ces méthodes sont connues dans les milieux scientifiques sous le terme « apprentissage artificiel » ou encore « apprentissage automatique » [1].

L'apprentissage automatique (*Machine Learning* en anglais) désigne le concept selon lequel une machine arrive à déduire le raisonnement automatique des règles à partir d'un ensemble d'exemples. L'objectif est de résoudre des problèmes relativement complexes, provenant du monde réel, la machine doit apprendre à produire la sortie désirée lorsque l'on introduit un groupe particulier de données. Cette dernière définition fait appel aux techniques de classification qui consistent à déterminer, à partir des données disponibles pour un problème posé, un modèle de classification automatique de nouvelles données jusqu'alors inconnues [2].

Un système capable d'apprendre directement à reconnaître différents types de maladies serait d'autant plus attrayant qu'il permettrait à la fois de réduire le temps du diagnostic, en évitant le passage par plusieurs experts, mais aussi de détecter et d'identifier, en particulier les maladies à risque, ce qui permet de déclencher, assez rapidement, le traitement approprié.

Selon l'encyclopédie de l'apprentissage automatique [3], il n'existe pas un classifieur « parfait », qui donne la meilleure réponse possible à n'importe quel problème. Ces dernières années, les recherches dans le domaine de l'apprentissage automatique ont prouvé que l'utilisation d'un ensemble de classifieurs permet d'améliorer les performances de classification de manière significative par rapport à l'utilisation d'un classifieur unique. L'objectif de combiner plusieurs classifieurs étant de gérer les forces et les faiblesses de chaque membre, ce qui conduit à la meilleure décision possible.

En effet, un ensemble de classifieurs permet de réduire le taux d'erreur et augmente la probabilité que la décision soit correcte. D'autre part, un choix judicieux des membres de l'ensemble de classifieurs permet de minimiser davantage le taux d'erreur. La justification formelle de ce résultat intuitif repose sur la notion de diversité inter-classifieurs.

L'objectif des travaux de recherche présenté dans cette thèse concerne l'exploitation d'outils issus de l'apprentissage automatique pour la détection et la classification des données médicales afin d'en tirer l'information la plus correcte possible. Nous assimilons le diagnostic médical d'une maladie à un problème d'apprentissage automatique et de classification afin d'émettre un avis pertinent avec un risque minime d'erreur. Ainsi, cette thèse propose une application de l'apprentissage automatique à un problème concret et d'actualité : l'aide au diagnostic médical pour la détection des pathologies, telles que les arythmies cardiaques.

Nous abordons aussi la notion de pertinence d'attributs et de leurs ensembles ainsi que le rôle capital qu'ils jouent dans la détermination de la performance des classifieurs. En effet, l'identification des attributs pertinents est une étape essentielle afin d'appliquer efficacement les algorithmes d'apprentissage artificiel. Une étude détaillée de l'état de l'art est présentée afin de faire le point sur l'avancement actuel et sur les différentes solutions proposées.

Le présent mémoire est organisé de la manière suivante :

dans le chapitre I, nous présentons les méthodes de sélection d'attributs. Nous abordons tout d'abord l'intérêt de la sélection d'attributs. Ensuite, nous détaillons les différentes

étapes et catégories de sélection d'attribut. À la fin, nous abordons le principe de classification et les mesures de performance des classifieurs.

Dans le chapitre II, nous établissons une étude expérimentale entre plusieurs catégories de méthodes de sélection d'attributs de type « filter ».

Dans le chapitre III, nous introduisons les méthodes de fusion au niveau des données, des attributs, et des décisions. Nous nous intéressons particulièrement à la fusion de décision (combinaison des classifieurs). Les différentes architectures de combinaison d'un ensemble de classifieurs ainsi que les critères de sélection sont présentés et discutés.

Dans le chapitre IV, nous présentons une nouvelle méthode basée sur le principe de diversité de classification tout en tirant la meilleure performance individuelle de chaque classifieur. Cette approche est appliquée sur plusieurs bases de données biomédicales dans l'objectif d'améliorer la reconnaissance des pathologies. Ainsi, des classifieurs hétérogènes sont utilisés pour construire un ensemble d'apprentissage en se basant sur leurs différentes décisions, éventuellement erronées, afin de choisir l'ensemble des classifieurs le plus optimal. Les résultats de l'implémentation proposée sont comparés à la littérature. Ce travail a fait l'objet d'une publication en revue internationale [4].

Chapitre I. SELECTION DES ATTRIBUTS DANS LE DOMAINE BIOMEDICAL

| | | |
|--------|--|----|
| 1.1. | Introduction | 5 |
| 1.2. | Différence entre sélection et extraction des variables | 6 |
| 1.3. | La sélection d'attributs | 7 |
| 1.3.1. | Motivation | 7 |
| 1.3.2. | Notions de pertinence d'un attribut | 8 |
| 1.3.3. | Notions de redondance d'un attribut | 8 |
| 1.4. | Procédure de sélection d'attributs | 9 |
| 1.4.1. | La procédure de génération | 9 |
| 1.4.2. | La fonction d'évaluation | 11 |
| 1.4.3. | Le critère d'arrêt | 16 |
| 1.4.4. | La validation | 16 |
| 1.5. | Techniques de l'apprentissage automatique | 17 |
| 1.5.1. | Catégories de l'apprentissage automatique | 17 |
| 1.5.2. | Problème de classification | 19 |
| 1.5.3. | Validation et évaluation des classifieurs | 21 |
| 1.6. | Etat de l'art de la sélection d'attributs dans le domaine biomédical | 23 |
| 1.7. | Conclusion | 25 |

1.1. INTRODUCTION

Les chercheurs en intelligence artificielle visent toujours à programmer des machines capables d'effectuer des tâches qui requièrent de l'intelligence. Cependant, programmer des machines capables de s'adapter à toutes les situations et éventuellement d'évoluer en fonction de nouvelles contraintes est difficile [1]. L'enjeu est de contourner cette difficulté en dotant la machine de capacités d'apprentissage lui permettant de tirer profit de son expérience. C'est pourquoi parallèlement aux recherches sur le raisonnement automatique se sont développées des recherches sur l'apprentissage par les machines en anglais « *machine Learning* ». Le principal objectif de ses recherches est la résolution automatique des problèmes complexes par la prise de décision sur la base d'observations de ces problèmes [1].

L'utilisation de l'apprentissage automatique pour les applications biomédicales connaît une augmentation considérable [3]. Ce regain d'intérêt a plusieurs causes. D'une part, l'application réussie des techniques d'apprentissage automatique dans différents domaines tels que la recherche sur le web, la reconnaissance de la parole et de l'écriture, la modélisation spatiale, etc. D'autre part, le développement le plus récent est l'avènement des dossiers médicaux électroniques. Par ailleurs, L'industrie s'est orienté vers le développement des techniques permettant la production de grandes quantités de données qui sont difficiles à exploiter [3].

Nous abordons ce chapitre par introduire les méthodes de réduction des données, en particulier la réduction du vecteur d'entrée, à savoir, le nombre d'attributs. Il existe principalement deux méthodes différentes : Sélection et extraction des variables. Dans ce chapitre, nous nous intéressons particulièrement aux méthodes de sélection des variables. Au début, les principales motivations pour ce choix sont décrites, suivies par la définition des notions de redondance et de pertinence d'un attribut. Par la suite, nous détaillons les différentes étapes et types de sélection d'attributs avant de présenter le principe de classification. Nous terminons ce chapitre par un état de l'art dans le domaine biomédical.

1.2. DIFFÉRENCE ENTRE SÉLECTION ET EXTRACTION DES VARIABLES

L'augmentation considérable de la quantité des données générées a conduit les chercheurs à employer des méthodes de réduction de la dimension des attributs d'entrée. Ces méthodes permettent de rendre l'ensemble des données plus représentatif du problème. Elles permettent généralement de réduire, non seulement, l'espace de stockage nécessaire pour ces données, mais aussi le temps d'apprentissage et d'exploitation des modèles de classification. Il existe deux approches de réduction de la dimension, à savoir: l'extraction d'attributs et la sélection d'attributs [5] [6].

Les méthodes d'extraction d'attributs consistent à transformer l'ensemble d'attributs de départ en un nouvel ensemble, généralement plus petit, tout en conservant autant que possible la structure originale des données. Parmi ces méthodes, nous citons principalement l'Analyse en Composantes Principales (ACP) et l'Analyse Factorielle Discriminante (AFD). Ces méthodes permettent de conserver les informations des attributs et ne tolèrent aucune perte de données. Cependant, elles souffrent d'inconvénients majeurs, qui sont principalement un temps de calcul élevé et l'absence d'une explicité des attributs extraits.

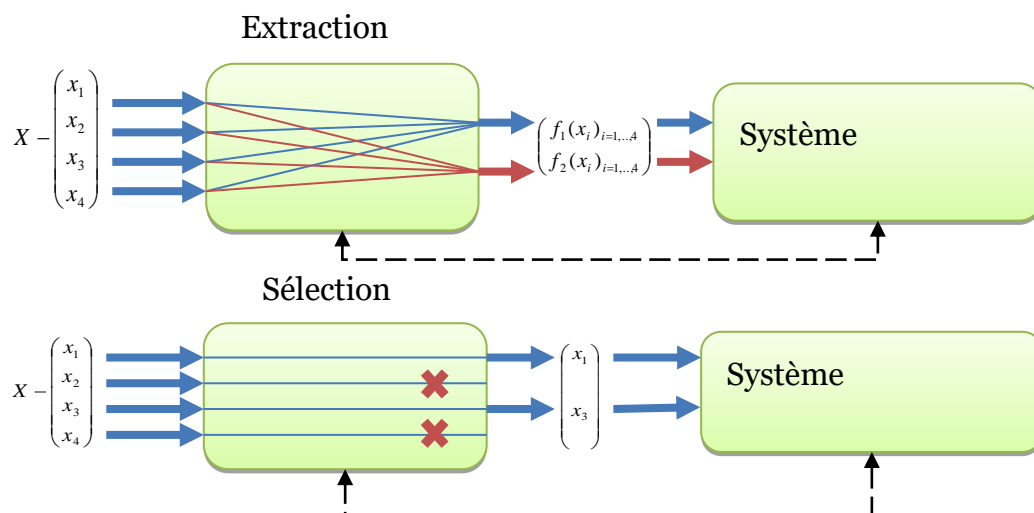


Figure I.1 Principes d'extraction et de sélection des caractéristiques [6]

Les méthodes de sélection d'attributs quant à elles consistent à choisir un sous-ensemble pertinent d'attributs à partir de l'ensemble original selon un critère de performance. Ces méthodes permettent une caractérisation plus rapide des données. La sélection d'attributs ne modifie pas la représentation originale des données. En effet, les attributs sélectionnés gardent leur sémantique de départ et peuvent alors être interprétés plus facilement par l'utilisateur [6] [7]. La Figure I.1 illustre les principes des méthodes d'extraction et de sélection des attributs.

1.3. LA SELECTION D'ATTRIBUTS

1.3.1. MOTIVATION

Le progrès technologique a permis d'augmenter sensiblement les capacités de mesure et a ainsi augmenté la taille des bases de données. Le traitement de grosses bases de données dans un temps relativement raisonnable est devenu un challenge majeur. Ceci a rendu la tâche de prise de décision plus difficile et plus complexe. La sélection d'attributs est une thématique de recherche qui a suscité un intérêt dans plusieurs domaines tels que l'apprentissage artificiel, la fouille de données, l'analyse de données en bio-informatique, ou encore le traitement d'images [8], [9]. En ce qui concerne les données médicales, la sélection d'attributs cherche à identifier les symptômes pertinents des maladies ciblées et vise à améliorer les performances de classification (en éliminant les attributs de faible influence).

La sélection d'attributs consiste à choisir parmi un ensemble d'attributs de grande taille un sous-ensemble d'attributs pertinents pour une application donnée [10]. Les principales motivations derrière ce choix sont :

- l'amélioration de la classification en éliminant les attributs ayant des informations inutiles ;
- l'aboutissement à une meilleure généralisation des données en évitant le sur-apprentissage ;
- l'interprétabilité des résultats en conservant la sémantique des attributs de départ ;

- la réduction du temps d'apprentissage et d'exécution ce qui résulte en un apprentissage moins coûteux en termes de complexité calculatoire.

En présence de centaines d'attributs, voire de milliers, il existe une forte probabilité pour que des attributs soient corrélés et expriment des informations similaires, que l'on considère « redondantes ». Par ailleurs, les attributs qui fournissent le plus d'information pour la classification seront considérés comme étant « pertinents ». Un ensemble d'attributs pertinents permet de satisfaire au mieux l'objectif fixé, à savoir, un apprentissage à la fois précis et rapide.

1.3.2. NOTIONS DE PERTINENCE D'UN ATTRIBUT

Les attributs dont les valeurs varient systématiquement avec les valeurs de classe sont considérés comme étant pertinents [11]. On distingue deux types d'attribut : les attributs fortement pertinents et les attributs faiblement pertinents. Les attributs fortement pertinents sont indispensables et devraient figurer dans tout sous-ensemble optimal sélectionné, car leur absence conduit à une erreur de reconnaissance de la fonction cible. D'autre part, la faible pertinence indique que l'attribut en question n'est pas toujours important, mais il peut devenir nécessaire pour un sous-ensemble optimal dans certaines conditions.

1.3.3. NOTIONS DE REDONDANCE D'UN ATTRIBUT

La notion de redondance d'attributs est généralement exprimée en termes de corrélation entre attributs. Deux attributs sont redondants (entre eux) si leurs valeurs sont complètement corrélées. Cette définition ne se généralise pas directement pour un sous-ensemble d'attributs. Les auteurs dans [12] présentent une définition formelle de la redondance qui permet de concevoir une approche d'identification et d'élimination des attributs redondants. Cette formalisation repose sur le principe de couverture de Markov (Markov blanket) [7], [12] où l'on examine la possibilité d'identifier les attributs inter corrélés. Ce type d'approche a été utilisé dans des applications biologiques [13].

1.4. PROCÉDURE DE SÉLECTION D'ATTRIBUTS

Les algorithmes de sélection d'attributs peuvent être divisés en deux catégories: algorithmes de classement des attributs (Feature ranking) et algorithmes de recherche de sous-ensembles d'attributs (Subset selection) [14]. Selon Dash [15], une procédure de sélection d'attributs est généralement composée de quatre étapes illustrées par la Figure I.2:

- étape 1: la procédure de génération (elle s'applique uniquement aux algorithmes de recherche de sous-ensembles).
- étape 2: la fonction d'évaluation.
- étape 3: le critère d'arrêt.
- étape 4: la procédure de validation.

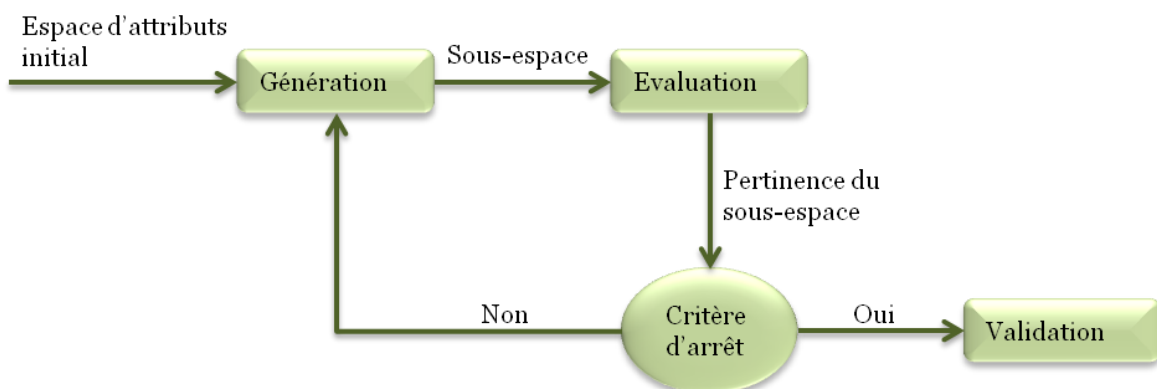


Figure I.2 Procédure de sélection d'attributs [15]

1.4.1. ÉTAPE1 : LA PROCÉDURE DE GÉNÉRATION

La procédure de génération permet, à chaque itération, de générer un sous-ensemble d'attributs qui sera évalué lors de la seconde étape de la procédure de sélection. Cette procédure peut être initiée avec un ensemble vide d'attributs ou avec l'ensemble de tous les attributs ou encore avec un sous-ensemble d'attributs choisi aléatoirement. Dans les deux premiers cas, les attributs sont itérativement ajoutés (Forward selection) ou retirés (Backward selection). Dans le troisième cas, on ajoute/retire des attributs (tout comme

dans les deux premiers cas), tandis que le nouveau sous-ensemble d'attributs est créé de manière aléatoire à chaque itération (Random generation).

En effet, pour un ensemble d'attributs initial de dimension d , le nombre total de sous-ensembles candidats qui peuvent être générés par la procédure de génération est 2^d . Ce nombre est d'autant plus élevé la dimension d est élevée. En conséquence, trois grandes approches de génération d'attributs ont été proposées dans la littérature: la génération complète, la génération aléatoire et la génération séquentielle [15].

La génération complète : Dans cette situation, une recherche complète du sous-ensemble d'attributs optimal (selon la fonction d'évaluation utilisée) est effectuée. Une recherche exhaustive est absolument complète alors qu'une recherche non exhaustive peut être complète. Ceci est possible en utilisant différentes fonctions heuristiques afin de réduire l'espace de recherche sans pour autant compromettre les chances de trouver le sous-ensemble optimal. Il s'agit d'utiliser un processus de backtracking (retour sur trace) permettant de revenir en arrière si la sélection s'engage dans une mauvaise direction de génération.

La génération aléatoire : Les procédures de génération aléatoire parcourent aléatoirement toutes les possibilités de sous-ensembles candidats, permettant ainsi de ne pas arrêter la recherche lorsque la fonction d'évaluation d'un sous-ensemble atteint un optimum local. Cependant, les 2^d sous-ensembles candidats ne sont pas tous évalués, contrairement aux procédures de génération complètes. Un nombre maximal d'itérations sont imposées afin que le temps de calcul reste relativement raisonnable. Les algorithmes génétiques (AG) [16], [17] sont les méthodes de génération aléatoire les plus couramment utilisées. Contrairement aux méthodes de génération complète dont la complexité est exponentielle, la complexité de calcul des méthodes basées sur une génération aléatoire est quadratique [18].

La génération séquentielle : Le principe est d'ajouter ou de supprimer un ou plusieurs attributs au fil des itérations. On distingue alors deux approches de génération séquentielles:

- Recherche vers l'avant : l'approche Forward ou Ascendante part d'un ensemble vide d'attributs auquel, à chaque itération on ajoute un ou plusieurs attributs.
- Recherche vers l'arrière : l'approche Backward ou Descendante est l'approche inverse, qui part de l'ensemble total des attributs. Chaque itération permet de supprimer un ou plusieurs attributs.

Les algorithmes utilisant ces approches de génération sont réputés pour leur simplicité de mise en œuvre et leur rapidité. Cependant, comme ils n'explorent pas tous les sous-ensembles possibles d'attributs et ne permettant pas de retour en arrière durant la recherche, ils sont donc considérés comme étant sous optimaux. Néanmoins, quelques algorithmes permettent d'ajouter et/ou de retirer itérativement des attributs. C'est notamment le cas de l'algorithme *SFFS* (*Sequential Forward Floating Selection*) qui consiste à appliquer après chaque étape *Forward* des étapes *Backward* tant que le sous-espace d'attributs correspondant améliore la fonction d'évaluation. Il existe un autre algorithme appelé *SBFS* (*Sequential Backward Floating Selection*) qui applique le même principe, à la différence que les deux étapes sont inversées.

1.4.2. ÉTAPE 2 : LA FONCTION D'EVALUATION

Pour sélectionner un sous-ensemble d'attributs, on distingue les trois types d'approches différentes « *wrapper* », « *filter* » et « *embedded* » [10] :

1.4.2.1. APPROCHE WRAPPER

Les approches de type *wrapper*, aussi appelées « approches enveloppantes », sont caractérisées par l'utilisation des mesures d'erreur de classification comme critère d'évaluation [11]. En effet, dans ces mesures, l'attribut ou les sous-ensembles d'attributs considérés sont évalués en fonction de la qualité de la classification obtenue en utilisant ces attributs. Ainsi, le sous-ensemble d'attributs le plus discriminant est celui qui présente le taux d'erreur de classification le plus faible .

Ces approches incorporent l'algorithme de classification dans la procédure de recherche et de sélection d'attributs ce qui permet d'obtenir théoriquement des bons résultats. La procédure de cette approche est illustrée par la Figure I.3.

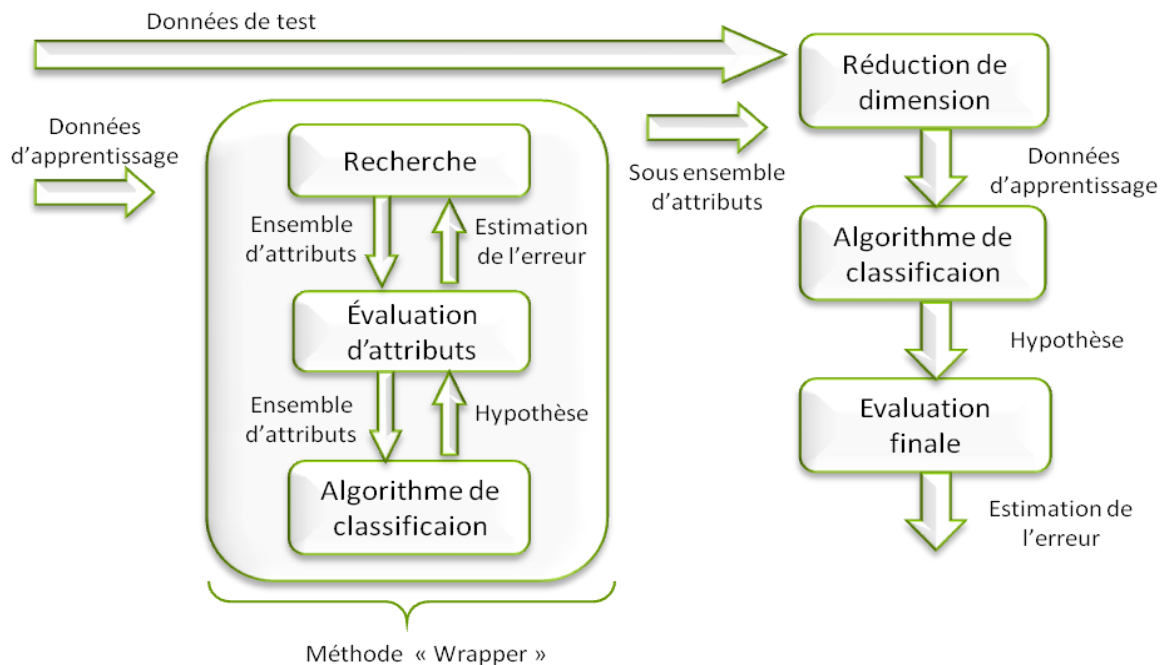


Figure I.3 Procédure de l'approche «wrapper»

Cependant, il existe deux inconvénients principaux qui limitent ces méthodes. Il s'agit de la complexité et la durée nécessaire pour la sélection. En effet, l'utilisation de telles méthodes nécessite pour chaque sous-espace d'attributs candidats d'effectuer la classification, ce qui peut devenir coûteux en temps de calcul, d'autant plus lorsque la dimension de l'espace d'entrée est grande.

D'autre part, ces méthodes sont très dépendantes de l'algorithme de classification utilisé comme critère d'évaluation. Ce dernier, s'il est mal adapté, pourrait contribuer à une sélection de mauvais attributs.

1.4.2.2. APPROCHE FILTER

Comparées à l'approche wrapper, l'approche *filter* (ou filtre) rapides, plus générales et moins coûteuses en temps de calcul, ce qui leur permet d'opérer plus facilement avec des bases de données de dimensions relativement très grandes. Cependant, leur indépendance de l'étape de classification ne permet pas de garantir que le meilleur taux de classification soit obtenu dans l'espace retenu. La procédure de cette approche est illustrée par la Figure I.4.

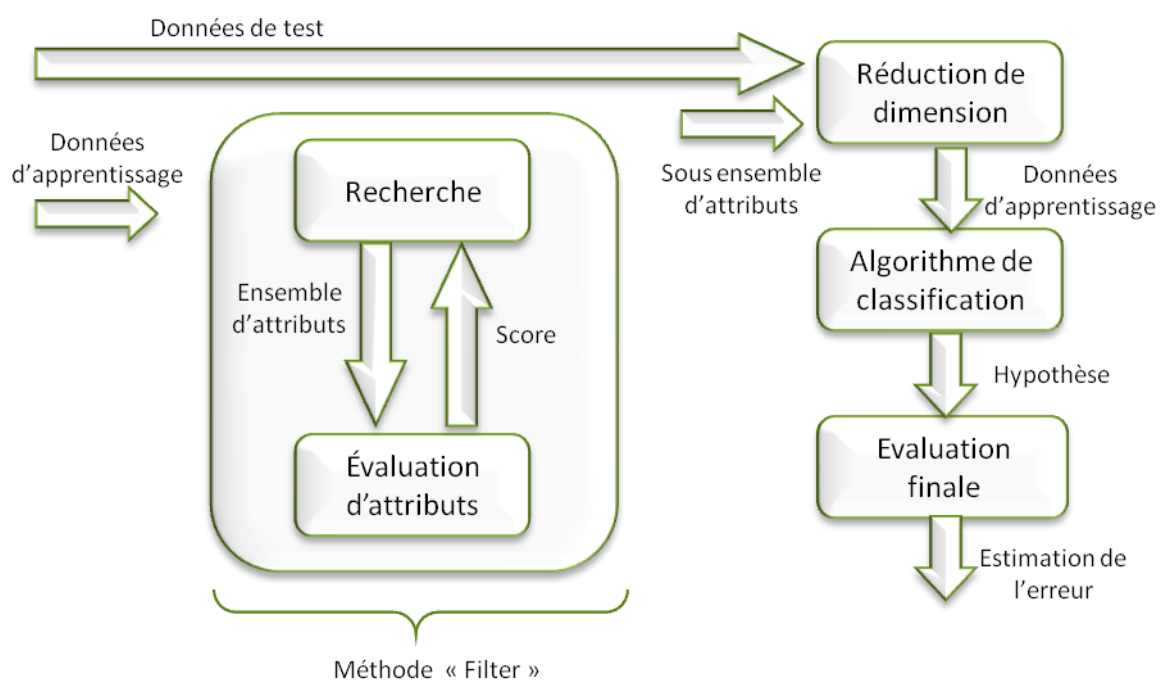


Figure I.4 Procédure de l'approche «filter»

Les approches de type *filter* utilisent une mesure d'évaluation basée sur les caractéristiques de l'ensemble des données, indépendamment de tout algorithme de classification. L'objectif étant de sélectionner certains attributs ou sous-ensemble d'attributs.

Il est à noter que lorsque l'on mesure la pertinence des attributs en les appréciant de manière individuelle, il s'agit d'un *filtre univarié* qui 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. Alors que lorsque l'on mesure la pertinence d'un sous ensemble d'attributs, il s'agit d'un *filtre multivarié* qui recherche le sous-ensemble

d'attributs le plus pertinent selon un certain critère de sélection. Ces algorithmes doivent alors trouver le meilleur sous-ensemble d'attributs parmi tous les sous-ensembles candidats en utilisant l'une des différentes méthodes de génération présentées ci-dessus.

Différentes fonctions d'évaluation ont été proposées pour évaluer un attribut ou un sous-ensemble d'attributs dans un contexte de sélection de type « filter ». Elles peuvent être classées en quatre approches distinctes [15] :

- Les **mesures d'information** quantifient le gain d'information apporté par un attribut étant estimé à partir des probabilités à posteriori. Un attribut f_1 est préféré à un attribut f_2 si le gain d'information apporté par l'attribut f_1 est plus grand que celui apporté par l'attribut f_2 .
- Les **mesures de consistance** ont une relation étroite avec la classe d'information dans la sélection du sous-ensemble d'attributs. Elles permettent de trouver le nombre minimal d'attributs qui séparent les classes comme consistance de l'ensemble complet d'attributs. En effet, ce type de mesure cherche à évaluer si l'attribut (ou le sous-ensemble d'attributs) étudié contient les informations nécessaires à la discrimination des classes [19].
- Les **mesures de dépendance** calculent la corrélation entre deux attributs. Autrement dit, c'est le fait de savoir à quel degré la valeur d'un attribut peut être prédite par la valeur d'un autre attribut. Plus l'attribut représente au mieux la classe, meilleure est son évaluation. Une petite variation permet de mesurer la dépendance entre plusieurs attributs, et de ce fait avoir le degré de redondance.
- Enfin, les **mesures de distance**, aussi appelées mesures de séparabilité, de divergence ou encore de discrimination. Dans cette mesure, un attribut ou un sous-ensemble d'attributs est sélectionné s'il permet une meilleure séparabilité et cohérence des classes.

Nous pouvons distinguer une autre mesure de type filter qui ressemble étroitement à l'approche embedded. Ce filtre se base sur un algorithme de classification afin sélectionner

les attributs pertinents (sans prendre en condiration les mesures de performances). Ainsi le score de chaque attribut est calculé selon son rôle lors de la construction du modèle de classification.

1.4.2.3. APPROCHE EMBEDDED

Dans l'approche « embedded » ou « intégrée », le processus de sélection d'attributs est effectué conjointement avec le processus de classification, contrairement aux approches décrites ci-dessus. En effet l'approche « filter » n'intègre pas l'apprentissage dans la sélection des attributs et l'approche « wrapper » utilise un algorithme d'apprentissage pour mesurer la qualité des sous-ensembles de fonctionnalités sans intégrer les connaissances sur la structure spécifique de la fonction de classification ou de régression. Dans l'approche intégrée, le classifieur sert non seulement à évaluer un sous-ensemble candidat mais aussi à guider le mécanisme de sélection.

Le Tableau I-1 résumé les avantages et les inconvénients de chaque approche. Notant que les approches de type « filter » sont plus largement utilisées principalement pour leur moindre coût de calcul ainsi que pour leur indépendance de tout algorithme de classification.

Tableau I-1 Techniques de sélection d'attributs [20]

| Approche | Avantages | Inconvénient |
|--|---|--|
| Filter | Univariée | |
| | Rapide | Ignore la dépendance entre les attributs |
| | Adaptable | |
| | Indépendant du classifieur | Ignore l'interaction avec le classifieur |
| | Multivariée | |
| | Les modèles intègrent la dépendance entre attributs | Plus lent que les techniques univariées |
| Indépendant du classifieur | Moins adaptable que les techniques univariées | |
| Meilleur calcul de complexité par rapport aux méthodes wrapper | Ignore l'interaction avec le classifieur | |

| | | |
|-----------------|--|--|
| Wrapper | Interagit avec le classifieur | Risque de sur-apprentissage |
| | Dépendance entre attributs | Sélection dépendante du classifieur Calcul intensif |
| Embedded | Interaction avec le classifieur | |
| | Meilleur calcul de complexité par rapport aux méthodes wrapper | Sélection dépendante du classifieur |
| | Dépendance entre attributs | |

1.4.3. ÉTAPE 3 : LE CRITÈRE D'ARRÊT

Le nombre optimal d'attributs n'étant pas connu a priori, il est fixé grâce à un critère d'arrêt du processus de sélection. Le choix du critère d'arrêt dépend généralement de la procédure de recherche [21] et/ou du critère d'évaluation [22]. Les critères d'arrêt les plus fréquents sont les suivants:

- Un certain seuil est atteint (comme par exemple en définissant le nombre d'attributs ou le nombre d'itérations) ;
- Addition (ou suppression) de n'importe quel attribut ne produit pas un meilleur sous-ensemble ;
- Un sous-ensemble sélectionné est suffisamment bon (un sous-ensemble peut être suffisamment bon si, par exemple, son taux d'erreur de classification est plus faible que le taux d'erreur admissible pour une tâche donnée) ;
- La recherche est accomplie.

1.4.4. ÉTAPE 4 : LA VALIDATION

Cette étape ne fait pas partie de la procédure de sélection d'attributs. Cependant, elle permet de tester la validité du sous-ensemble d'attributs sélectionnés en effectuant plusieurs tests sur des exemples de données réelles et/ou générées artificiellement.

La façon la plus simple pour valider les attributs est de mesurer directement ces derniers en utilisant des connaissances a priori sur les données. Si l'on connaît les attributs pertinents à l'avance, il est possible de comparer cet ensemble d'attributs connu avec les attributs sélectionnés (les connaissances sur les attributs non pertinents ou redondants peuvent également être utiles). Habituellement, dans les applications du monde réel, il n'existe pas de connaissances a priori. Par conséquent, des méthodes indirectes sont considérées en surveillant le changement des performances par rapport aux changements des attributs. Par exemple, si le taux d'erreur de classification est employé comme un indicateur de performance pour le traitement, pour un sous-ensemble d'attributs sélectionné, il est possible de simplement suivre l'expérience « avant et après » pour comparer le taux d'erreur de classifieur sur l'ensemble complet d'attributs et sur le sous-ensemble sélectionné.

1.5. TECHNIQUES DE L'APPRENTISSAGE AUTOMATIQUE

La sélection d'attributs est une technique permettant de choisir les caractéristiques, les variables ou les mesures les plus intéressantes, pertinentes et informantes, à un système donné, pour la réalisation de la tâche pour laquelle il a été conçu. Cette phase est généralement un module important d'un système complexe. Les domaines d'application des techniques de sélection de caractéristiques sont variés tels que la modélisation, l'apprentissage automatique (Machine Learning) et l'analyse exploratoire de données (Data Mining). Dans ce mémoire, nous nous intéressons plus particulièrement à la sélection d'attributs pour l'apprentissage automatique et la classification supervisée des données médicales.

1.5.1. CATEGORIES DE L'APPRENTISSAGE AUTOMATIQUE

L'apprentissage est une caractéristique propre aux êtres humains. Ainsi en exécutant des tâches similaires c'est possible d'acquérir la capacité d'améliorer les performances. Ce principe peut être appliqué à des machines et ainsi améliorer leur capacité d'apprentissage, on parle du «Machine learning» ou en français apprentissage automatique [2], [1].

L'apprentissage automatique peut être classé principalement en quatre catégories: i) l'apprentissage supervisé, ii) l'apprentissage non supervisé, iii) l'apprentissage semi-supervisé et iv) l'apprentissage par renforcement [2].

L'apprentissage supervisé nécessite un formateur, qui fournit les instances d'entraînement d'entrée-sortie, autrement dit, un expert doit préalablement étiqueter des exemples. Dans ce cas, les classes sont prédéterminées et les exemples sont connus, le système apprend à classer et adapte ses paramètres selon un modèle de classement (algorithme de classification ou classifieur). Le processus se passe en deux phases : lors de la première phase (hors ligne, dite d'apprentissage), il s'agit de déterminer un modèle des données étiquetées. La seconde phase (en ligne, dite de test) consiste à prédire l'étiquette d'une nouvelle donnée, connaissant le modèle préalablement appris. Parfois il est préférable d'associer une donnée non pas à une classe unique, mais à une probabilité d'appartenance à chacune des classes prédéterminées. Il s'agit d'un **apprentissage supervisé probabiliste**.

En l'absence de formateurs, la sortie désirée pour une instance d'entrée donnée n'est pas connue. Dans ce cas, le système ne dispose que d'exemples, sans les étiquettes, et le nombre de classes ainsi que leur nature n'ont pas été prédéterminées. Par conséquent, l'algorithme doit adapter ses paramètres de manière autonome. Ce type d'apprentissage est appelé **apprentissage non supervisé**.

Le troisième type d'apprentissage vise à faire apparaître la distribution sous-jacente des exemples dans leur espace de description. Il est mis en œuvre quand des données ou étiquettes manquent. Le modèle doit utiliser des exemples non étiquetés pouvant néanmoins renseigner. Il s'agit de **l'apprentissage semi-supervisé**.

Dans **l'apprentissage par renforcement**, l'algorithme ne connaît pas explicitement les instances d'entrée-sortie, mais il apprend un comportement étant donné une observation. L'action de l'algorithme sur l'environnement produit une valeur de retour qui guide l'algorithme d'apprentissage.

Il existe d'autres catégories d'apprentissage tels que **l'apprentissage partiellement supervisé** dans lequel un modèle énonce qu'une donnée n'appartient pas à une certaine classe, sans préciser dans quelle classe elle appartient ; ou encore **l'apprentissage par transfert** qui peut être vu comme la capacité d'un système à reconnaître et à appliquer des connaissances et des compétences, apprises à partir des tâches antérieures, sur de nouvelles tâches ou domaines partageant des similitudes.

1.5.2. PROBLEME DE CLASSIFICATION

Dans un cas typique d'apprentissage supervisé, les données sont représentées par une table d'exemples ou d'instances. Chaque instance est décrite par un nombre fixe d'attributs dont un label qui dénote la classe à laquelle il appartient. Il existe deux types d'attributs : nominal (la valeur est membre d'un ensemble ordonné) ou numérique (la valeur est réellement un nombre).

Dans l'apprentissage supervisé, lorsque l'ensemble des valeurs de sortie est fini, on parle d'un problème de classification, qui revient à attribuer une étiquette à chaque entrée. La fonction de prédiction est alors appelée un classifieur.

La classification permet donc de structurer et d'organiser un ensemble d'observations en plusieurs groupes homogènes dans le but de faciliter l'analyse des données, et ainsi tirer des conclusions fiables pour effectuer des prédictions efficaces. Un classifieur peut être défini comme étant un système de reconnaissance qui travaille dans un certain espace de caractéristiques. Il utilise une certaine base pour ajuster ses paramètres, prend sa décision à partir d'une certaine règle et fournit en sortie un certain type de réponse [23]. Ce principe est illustré dans la Figure I.5.

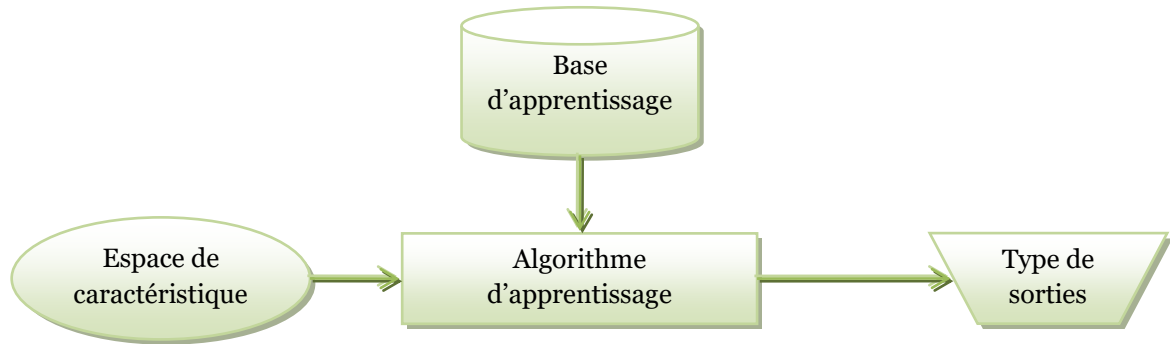


Figure I.5 Principe de la classification [23]

Selon Xu et al [24], il existe trois types de sortie (classe, rang et mesure) :

- **Type classe** : Les sorties sont données comme un label de classe unique représentant la classification de niveau abstrait qui peut être considéré comme un expert pointant vers une seule catégorie en tant que classe gagnante. Les sorties sous forme de classe ignorent les informations sur les chances des autres classes n'ayant pas été sélectionnées comme décision finale.
- **Type rang** : le classifieur attribue un classement à chaque classe avec le rang le plus élevé étant le premier choix. La valeur de rang ne peut pas être utilisée dans l'isolement parce que le rang le plus élevé ne signifie pas nécessairement un niveau de confiance élevé dans la classification.
- **Type mesure** : beaucoup de méthodes de classification génèrent leurs décisions sur la base de certaines mesures obtenues par une fonction discriminante d'un classifieur ou certaines mesures de distance entre les données dans l'espace d'entrée. Ces mesures représentent des degrés de soutien pour chaque classe et peuvent être normalisées afin d'estimer la probabilité ou la fonction d'appartenance floue d'un échantillon particulier appartenant à des classes différentes.

Chaque type de sortie (classe, rang ou mesure) correspond à un niveau d'information différent fourni par le classifieur. La sortie de type classe est la plus simple mais la moins riche en information. La sortie de type rang reflète l'ordre de préférence des propositions

fournies par le classifieur. La sortie de type mesure est quant à elle la plus riche en information puisqu'elle reflète le niveau de confiance du classifieur dans ses propositions. Notant que les sorties de type classe sont les plus universelles. En effet, lorsque les sorties sont de type rang ou de type mesure, on peut évidemment les transformer en type classe. Ceci consiste à tenir compte uniquement de la première solution de la liste proposée par chaque classifieur.

1.5.3. VALIDATION ET EVALUATION DES CLASSIFIEURS

La classification automatique vise à assigner des objets à des catégories ou classes. Le principe général des systèmes de classification se déroule en deux phases : la phase d'apprentissage est une phase de développement aboutissant à la mise en œuvre d'une stratégie de classification et la phase de test au cours de laquelle les performances du système de classification sont évaluées (dans cette phase, les labels de classes des données ne sont pas connus par le système) [25]. En général, un système n'est prêt pour une utilisation réelle qu'après une succession d'étapes d'apprentissage et de test, permettant de mettre en place une stratégie de classification efficace.

Selon la répartition des données entre les deux sous-ensembles (apprentissage et test), il existe différentes approches de validation [19]. Nous citons:

- **La méthode Holdout** : les données sont divisées en deux sous-ensembles : le sous-ensemble d'apprentissage et le sous-ensemble de test dans des proportions $1/2$, $1/2$ pour chacun de ses deux sous-ensembles ou $2/3$ pour l'ensemble d'apprentissage et $1/3$ pour l'ensemble de test.
- **La méthode de resubstitution** : l'ensemble d'apprentissage est utilisé comme ensemble de test.
- **La méthode k-validation croisée** : l'ensemble des données est partitionné en k parties de tailles à peu près égales. Nous réalisons ainsi k fois la procédure de validation et à chaque fois une des parties constitue l'ensemble test et les $k-1$ parties restantes sont réunies pour former l'ensemble d'apprentissage.

- **La méthode Leaving-one-out** est un cas particulier de la méthode k-validation croisée où l'ensemble des données est divisé en n parties. La base de test contient alors à chaque fois une seule donnée. Une fois que les données sont divisées en un ensemble d'apprentissage et un ensemble de test, l'erreur de classification est mesurée sur l'ensemble de test en utilisant les prototypes de classes de l'ensemble d'apprentissage.

Afin de mesurer de qualité d'un classifieur, il est important d'estimer les erreurs de classification qu'il peut engendrer, autrement dit, la probabilité que la classe prédite pour une donnée quelconque soit incorrecte.

L'**erreur de classification**, erreur de généralisation ou taux d'échec (abrégée par E ou e) d'un classifieur est la probabilité que ce classifieur ne prédise pas correctement la classe d'une donnée. L'**erreur apparente** ou erreur d'apprentissage (abrégée par E_{app}) est mesurée avec les exemples utilisés pour la construction du classifieur : c'est la proportion d'exemples dont la classe est mal prédite.

Il existe aussi d'autres mesures très connues dans la littérature qui permettent d'évaluer les performances d'un classifieur. Ces mesure sont généralement calculés à partir de la matrice de confusion. Le Tableau I-2 représente une matrice de confusion pour une classification binaire.

Tableau I-2 Matrice de confusion

| | <i>Classe prédite positives</i> | <i>Classe prédite négatives</i> |
|-------------------------|---------------------------------|---------------------------------|
| <i>Classe positives</i> | VP | FN |
| <i>Classe négatives</i> | FP | VN |

Une matrice de confusion contient des informations sur les résultats réels et prévus donnés par un classificateur.

Vrai positif (VP) : le nombre de bonnes prédictions positives ;

Vrai négatif (VN) : le nombre de bonnes prédictions négatives ;

Faux positifs (FP) : le nombre d'erreurs sur les prévisions positives ;

faux négatif (FN) : le nombre d'erreurs sur les prédictions négatives.

A partir de la matrice de confusion, on peut calculer le taux d'exactitude, le taux de spécificité et le taux de sensibilité :

- 1) **L'exactitude ou la précision** : c'est le taux de bonne classification.

$$Pr = 1 - E = \frac{VP + VN}{VP + FP + VN + FN} \quad (1)$$

- 2) **Sensibilité du test** : C'est la proportion de tests positifs (vrai positif) chez les sujets malades. Moins les faux négatifs sont nombreux, plus le test est sensible.

$$Se = \frac{VP}{VP + FN} \quad (2)$$

- 3) **Spécificité du test**: C'est la proportion de tests négatifs chez les sujets non malades. Moins les faux positifs sont nombreux, plus le test est spécifique.

$$Sp = \frac{VN}{FN + VN} \quad (3)$$

1.6. ETAT DE L'ART DE LA SELECTION D'ATTRIBUTS DANS LE DOMAINE BIOMEDICAL

Les bases de données biomédicales sont souvent représentées par un grand nombre de caractéristiques de la maladie et un nombre relativement faible de dossiers des patients. Ces caractéristiques (attributs) ne sont pas toutes pertinentes et peuvent être source de bruit. Plusieurs travaux de recherche ont été réalisés pour remédier à cette malédiction de dimension. Ces travaux peuvent être divisés en plusieurs axes de recherches indépendants :

- **Traitement des données médicales** : Afin de démontrer l'utilité de la sélection des attributs dans le domaine biomédical, plusieurs recherches ont été testées sur les différents ensembles de données médicales disponibles sur le net, comme par exemple le répertoire de « UCI Machine Learning ». Dans [26], les auteurs

proposent une technique qui recherche une division stratégique de l'espace des caractéristiques dans le but d'identifier les meilleurs sous-ensembles de caractéristiques pour chaque instance. Cette technique est basée sur l'approche Wrapper, où un algorithme de classification est utilisé comme fonction d'évaluation pour différencier entre plusieurs sous-ensembles d'attributs. Dans [27], des règles d'association et de corrélation des caractéristiques ont été utilisées afin de réduire la dimensionnalité dans le domaine médical. Dans [28] les auteurs se sont basé sur des méthodes d'hybridation des *Particules Swarm Optimization* (PSO) et la théorie des ensembles.

- **La recherche et analyse de documents médicaux** : Les documents textuels sont généralement représentés comme une matrice attribut-document. Les attributs peuvent être des mots simples à partir du document de texte ou paires plus complexes extraites afin d'enrichir la représentation matricielle [29]. Dans [30] une approche wrapper basée sur la recherche d'information sémantique est proposée. La similarité sémantique consiste à calculer la similarité entre les termes conceptuellement similaires, mais lexicalement dissemblables. D'autres applications dans le domaine biologique ont été revues dans [31].
- **L'imagerie médicale** : La sélection d'attributs dans l'imagerie est aussi très importante [32]. Dans [33] les auteurs ont passé en revue les différentes méthodes qui existent dans la littérature pour la détection du cancer du sein.
- **L'analyse et la prédiction des séquences qui codent les protéines** : sachant que de nombreuses caractéristiques peuvent être extraites d'une séquence et la plupart des dépendances se produisent entre des positions adjacentes. Pour faire face à la grande quantité d'attributs possibles, et la quantité souvent limitée d'échantillons, les auteurs de [34] ont présenté le modèle interpolé Markov. Ils ont utilisé l'interpolation entre les différents ordres du modèle de Markov en se basant sur des échantillons de petite taille et une méthode de filtrage (χ^2) dans l'objectif de sélectionner les caractéristiques pertinentes. Saeys et al. [35], ont combiné différentes mesures de codage de prédiction potentiel, et ont ensuite utilisé une

approche de filtre multivariée pour la couverture de Markov et ainsi ne conserver que les attributs les plus pertinents. Parallèlement de nombreuses méthodes d'analyse de séquence comprennent la reconnaissance de signaux courts, plus ou moins conservées dans la séquence, ce qui représente principalement des sites de liaison pour diverses protéines [36]. Une approche commune pour trouver des motifs réglementaires, est de relier des motifs à des niveaux d'expression des gènes en utilisant une approche de régression. La sélection des fonctionnalités peut ensuite être utilisé pour rechercher les motifs qui maximisent l'ajustement du modèle de régression [37]. Dans [38], les auteurs démontrent les avantages d'employer la sélection d'attributs, en utilisant l'entropie caractéristique de classe comme filtre pour éliminer les caractéristiques non pertinentes.

- **Traitement des données biopuces :** l'avènement des jeux de données de puces à ADN a stimulé une nouvelle ligne de recherche en bioinformatique. Les données de biopuces constituent un grand défi pour les techniques de calcul, en raison de leur grande dimension (jusqu'à plusieurs dizaines de milliers de gènes) et de leurs petites tailles d'échantillon [39]. En outre, des complications expérimentales supplémentaires comme le bruit et la variabilité rendent l'analyse des données de puces à ADN un domaine très passionnant. Afin de faire face aux caractéristiques particulières de ces données, la nécessité évidente de réduction de dimension a fait objet de plusieurs recherches scientifiques [40]–[43]. En raison de la haute dimensionnalité de la plupart des analyses micropuces, les techniques de sélection rapides et efficaces, telles que les méthodes de filtrage univariées, ont attiré plus d'attention.

1.7. CONCLUSION

Dans ce chapitre, nous avons présenté des notions théoriques sur le principe de sélection d'attributs selon la pertinence et la diversité d'information portée afin d'optimiser la classification. Nous avons ainsi présenté les études réalisées dans l'état de l'art concernant la procédure de sélection d'attributs. Cette procédure étant basée sur quatre étapes principales : La génération d'attribut, la fonction permettant leur évaluation, le critère

d'arrêt de l'opération et enfin la validation de par le test de l'efficacité de l'ensemble des attributs. Dans le prochain chapitre, nous allons utiliser ces notions pour une étude expérimentale et comparative entre différents filtres.

Chapitre II.

ETUDE DES METHODES DE SELECTION D'ATTRIBUTS

| | | |
|--------|--|----|
| 2.1. | Introduction | 28 |
| 2.2. | Materiels et méthodes | 29 |
| 2.2.1. | Présentation de l'ensemble de données | 29 |
| 2.2.2. | Classifieurs utilisés | 31 |
| 2.2.3. | Evaluation de la base de donnée originale..... | 31 |
| 2.2.4. | Prétraitement de la base | 32 |
| 2.2.5. | Utilisation de la base dans la littérature..... | 34 |
| 2.3. | Sélection des attributs avec l'approche Filter | 37 |
| 2.3.1. | Approche univarié | 37 |
| 2.3.2. | Approche multivarié | 42 |
| 2.4. | Résultats obtenus | 45 |
| 2.4.1. | Filtre univarié | 46 |
| 2.4.2. | Filtre multivarié | 53 |
| 2.5. | Discussion des résultats | 54 |
| 2.6. | Conclusion | 58 |

2.1. INTRODUCTION

L'identification des attributs pertinents et la suppression des attributs redondants est une étape cruciale pour une classification efficace. De nombreux algorithmes de sélection ont été proposés dans la littérature. Nous avons présenté dans le premier chapitre la procédure générale de sélection et nous avons vu qu'il existe trois approches générales de sélection d'attributs: *wrapper*, *embedded* et *filter*. Ces méthodes permettent de ne sélectionner que les attributs ou les ensembles d'attributs les plus pertinents afin d'atteindre leurs objectifs de classification.

Dans ce chapitre une panoplie de méthodes de sélection de type « filter » sera présentée et expérimentée. Ce choix est justifié par la simplicité de ces méthodes, leur temps de calcul, et principalement pour leur indépendance de l'algorithme de classification. Nous nous sommes intéressés à un problème de classification pour l'aide au diagnostic médical et plus particulièrement à la détection des arythmies cardiaques.

Selon l'organisation mondiale de la santé, les maladies cardio-vasculaires sont la première cause de mortalité dans le monde. En effet, on estime à 17,5 millions le nombre de décès dû aux maladies cardiovasculaires, soit 31% de la mortalité mondiale totale [44]. Il est possible de prévenir la plupart des maladies cardiovasculaires en s'attaquant aux facteurs de risque comportementaux (tabagisme, mauvaise alimentation et obésité, sédentarité et utilisation nocive de l'alcool) à l'aide de stratégies à l'échelle de la population [44]. Les personnes souffrant de maladies cardiovasculaires ou exposées à un risque élevé de maladies cardiovasculaires (du fait de la présence d'un ou plusieurs facteurs de risque comme l'hypertension, le diabète, l'hyperlipidémie ou une maladie déjà installée) nécessitent une détection précoce et une prise en charge au moment opportun avant qu'il ne soit trop tard [44]. Les arythmies cardiaques font références à n'importe quel changement du rythme du cœur (trop faible, trop fort ou irrégulier). À chaque battement cardiaque, un signal électrique se propage à partir du haut du cœur vers le bas. Comme le signal se déplace, il provoque le cœur à se contracter et pomper le sang. Ce signal peut être détecté en utilisant des électrocardiogrammes (ECG).

Au cours des dernières décennies, plusieurs méthodes automatisées pour la détection des arythmies ont été développées. Ces méthodes simplifient la tâche de surveillance et offrent une aide aux médecins afin d'améliorer l'efficacité du diagnostic. En effet, l'électrocardiographie informatisée est actuellement une pratique bien établie, en soutenant le diagnostic humain. Ainsi, de nombreux algorithmes d'apprentissage automatique ont été proposés pour la détection des arythmies cardiaques, en utilisant l'ensemble de données d'arythmie UCI.

Nous présentons au début de ce chapitre les principaux outils utilisés dans cette étude en décrivant la base de données, son prétraitement ainsi que les principales méthodes déjà réalisées dans la littérature. Par la suite, nous entamons une étude expérimentale avec plusieurs filtres de type univarié et multivarié. Les résultats obtenus sont comparés et discutés.

2.2. MATERIELS ET METHODES

Toutes les expérimentations réalisées dans ce travail ont été exécutées sur un ordinateur processeur pentium(R), CPU Dual-Core T4400 2,20GHz. Nous avons utilisé le logiciel open source Weka (Waikato Environment for Knowledge Analysis) [45], [46] pour la sélection d'attributs et la bibliothèque PRTools [47] disponible sous Matlab pour la classification.

2.2.1. PRESENTATION DE L'ENSEMBLE DE DONNEES

La base de données utilisé dans ce travail « Arrhythmia » concerne des patients turques [48]. Elle contient 279 attributs, dont 206 ont une valeur linéaire et 73 ont une valeur nominale. Notre objectif est de faire la distinction entre la présence et l'absence des troubles du rythme cardiaque. Ainsi, nous avons regroupé l'ensemble de données en deux catégories: 245 enregistrements (classe 01) font référence à un ECG normal et 185 enregistrements (classes 02 à 15) font référence à un ECG anormal (présence d'arythmie). Notant que ce jeu de données est disponible dans le répertoire UCI [49]. Les informations détaillées sur les attributs sont présentées dans le Tableau II-1 ci-dessous.

Tableau II-1 Attributs de la base « Arrhythmia »

| <i>N° Attribut</i> | <i>Nom</i> | <i>Type</i> |
|--------------------|---|---|
| 1 | Age | linéaire |
| 2 | Sexe (0 = masculin, 1 = féminin) | nominal |
| 3 | Hauteur (cm) | linéaire |
| 4 | Poids (kg) | linéaire |
| 5 | Durée du QRS (ms) | linéaire |
| 6 | Intervalle PR (ms) | linéaire |
| 7 | Intervalle QT (ms) | linéaire |
| 8 | Intervalle T (ms) | linéaire |
| 9 | Intervalle P (ms) | linéaire |
| 10 | QRS (degré) | Linéaire |
| 11 | T (degré) | Linéaire |
| 12 | P (degré) | Linéaire |
| 13 | QRST (degré) | Linéaire |
| 14 | J (degré) | linéaire |
| 15 | Fréquence cardiaque | linéaire |
| 16.. 27 | Du canal DI : Onde Q, onde R, Onde S, ondes R' (petit pic juste après R), Onde S', Nombre de déflexions intrinsèques, Existence d'ondes R déformées, Existence d'une dérivation diphasique de l'onde R, Existence de l'onde P déchiquetée, Existence d'une dérivation diphasique de l'onde P, Existence d'ondes T déformées, Existence d'une dérivation diphasique de l'onde T, | De 16 à 21: Linéaire et de 22 à 27 nominal |
| 28.. 39 | Du canal DII : (similaire à 16.. 27 du canal DI) | |
| 40.. 51 | Des canaux DIII: (similaire à 16.. 27 du canal DI) | |
| 52.. 63 | Du canal AVR: (similaire à 16.. 27 du canal DI) | |
| 64.. 75 | Du canal AVL: (similaire à 16.. 27 du canal DI) | |
| 76.. 87 | De canal AVF: (similaire à 16.. 27 du canal DI) | |
| 88.. 99 | Du canal V1: (similaire à 16.. 27 du canal DI) | |
| 100.. 111 | Du canal V2: (similaire à 16.. 27 du canal DI) | |
| 112.. 123 | Du canal V3: (similaire à 16.. 27 du canal DI) | |
| 124.. 135 | Du canal V4: (similaire à 16.. 27 du canal DI) | |
| 136.. 147 | Du canal V5: (similaire à 16.. 27 du canal DI) | |
| 148.. 159 | Du canal V6: (similaire à 16.. 27 du canal DI) | |
| 160.. 169 | Du canal DI : Onde JJ, onde Q, onde R, S, R', S', onde P, onde T, QRSA (Somme des zones de tous les segments divisées par 10 sachant que Zone = largeur * hauteur / 2), QRSTA (QRSTA= QRSA + 0,5 * largeur de l'onde T * 0,1 * hauteur de l'onde T) | Linéaire |
| 170.. 179 | Du canal DII: (similaire à 160.. 169 du canal DI) | Linéaire |
| 180.. 189 | Du canal DIII: (similaire à 160.. 169 du canal DI) | Linéaire |

| | | |
|-----------|---|----------|
| 190.. 199 | Du canal AVR: (similaire à 160.. 169 du canal DI) | Linéaire |
| 200.. 209 | Du canal AVL: (similaire à 160.. 169 du canal DI) | Linéaire |
| 210.. 219 | De canal AVF: (similaire à 160.. 169 du canal DI) | Linéaire |
| 220.. 229 | Du canal V1: (similaire à 160.. 169 du canal DI) | Linéaire |
| 230.. 239 | Du canal V2: (similaire à 160.. 169 du canal DI) | Linéaire |
| 240.. 249 | Du canal V3: (similaire à 160.. 169 du canal DI) | Linéaire |
| 250.. 259 | Du canal V4: (similaire à 160.. 169 du canal DI) | Linéaire |
| 260.. 269 | Du canal V5: (similaire à 160.. 169 du canal DI) | Linéaire |
| 270.. 279 | Du canal V6: (similaire à 160.. 169 du canal DI) | Linéaire |

2.2.2. CLASSIFIEURS UTILISES

Afin de tester les performances des filtres, nous avons divisé la base de données en deux ensemble : 2/3 pour l'apprentissage et 1/3 pour le test. Ensuite, nous avons appliqué six (06) classifieurs de nature très différentes, disponibles dans la boîte à outils "PRTools" [47]. Le Tableau II-2 représente l'abréviation et le nom des classifieurs utilisés.

Tableau II-2 Abréviation des classifieurs utilisés

| <i>Abréviation</i> | <i>Nom du Classifieur</i> |
|--------------------|--|
| <i>Fisherc</i> | Linéaire moindres carrés de Fisher |
| <i>Knnc</i> | Classifieur K-plus proche voisin |
| <i>Dtc</i> | Classifieur Arbre de décision -arbre de Verzakov- |
| <i>Naivebc</i> | Classifieur Naive Bayes |
| <i>Svc</i> | Classifieur Support Vector Machine |
| <i>Bpxnc</i> | Réseaux de neurone avec apprentissage par rétro-propagation vers l'avant |

2.2.3. ÉVALUATION DE LA BASE DE DONNEE ORIGINALE

Nous avons évalué la base de données originale, à savoir sans prétraitement (en dehors du remplacement des valeurs manquantes) et sans réduction du vecteur d'entrée. L'évaluation est présentée dans le Tableau II-3 selon l'erreur de classification (E), le taux de spécificité (Sp %) et le taux de sensibilité (Se %).

Tableau II-3 Résultats obtenus sans prétraitement de la base de données

| <i>Classifieur</i> | <i>E</i> | <i>Se (%)</i> | <i>Sp (%)</i> |
|--------------------|----------|---------------|---------------|
| <i>Fisherc</i> | 0,495 | 47,69 | 53,27 |
| <i>Knnc</i> | 0,399 | 69,23 | 59,40 |
| <i>Dtc</i> | 0,24 | 75 | 77,17 |
| <i>Naivebc</i> | 0,272 | 84,61 | 69,167 |
| <i>Svc</i> | 0,44 | 54,05 | 58,16 |
| <i>Bpxnc</i> | 0,325 | 68,57 | 67,65 |

La meilleure performance en termes de taux d'erreur est obtenu en utilisant le classifieur *Dtc* avec $e=0,24$ alors que la plus mauvaise performance est associée au classifieur *Fisherc* ayant un très large taux d'erreur de **0,495**. Le pourcentage du taux d'erreur moyen obtenu par l'ensemble des classifieurs est d'environ **36,2%**. La meilleure sensibilité est obtenue par le classifieur *Naivebc* (**84,61%**) et la plus mauvaise par *Fisherc* (**47,69%**). Tandis que la meilleure spécificité est obtenue par le classifieur *Dtc* (**77,17%**) et la plus mauvaise par *Fisherc* (**53,27%**).

2.2.4. PRETRAITEMENT DE LA BASE

Nous avons remarqué , qu'en utilisant la base de données originale, les résultats obtenus sont très mauvais. Dans le but de les améliorer nous avons traiter ce problème de sa racine, à savoir le prétraitement de la base. Pour cela, nous avons appliqué un ré-échantillonnage des données en utilisant la méthode non supervisée « resample » [50], [51]. Le nombre de cas dans l'ensemble des données générées peut être spécifié. Dans cette étude, nous avons conservé 100% de la base sans remplacement. Les résultats obtenus sont résumés dans le Tableau II-4.

Tableau II-4 Résultats obtenus avec ré-échantillonnage des données

| <i>Classifieur</i> | <i>E</i> | <i>Se (%)</i> | <i>Sp(%)</i> |
|--------------------|----------|---------------|--------------|
| <i>Fisherc</i> | 0,231 | 70,37 | 81,42 |
| <i>Knnc</i> | 0,292 | 66,67 | 75 |
| <i>Dtc</i> | 0,130 | 84,41 | 91,58 |
| <i>Naivebc</i> | 0,229 | 95,35 | 75,19 |
| <i>Svc</i> | 0,182 | 77,63 | 85,42 |
| <i>Bpxnc</i> | 0,160 | 85,07 | 84,76 |

Le classifieur qui minimise le plus le taux d'erreur est le classifieur *Dtc* avec $e=0,130$. Inversement, le classifieur qui maximise le taux d'erreur est le classifieur *knnc* avec $e=0,292$. La moyenne générale du taux d'erreur est de **18,22%**. La meilleure sensibilité est obtenue par le classifieur *Naivebc* (**95,35%**) et la plus mauvaise par *knnc* (**66,67%**). Tandis que la meilleure spécificité est obtenue par le classifieur *Dtc* (**91,58%**) et la plus mauvaise par *knnc* (**75%**).

Le ré-échantillonnage des données a nettement amélioré les résultats. Une comparaison peut être observée sur la Figure II.1 . Ainsi, la moyenne de l'erreur de tous les classifieurs a été diminuée de plus de **19%**. Vu l'amélioration apportée lors de cette expérimentation, nous avons considéré cette nouvelle base de données pour la suite du travail. Toutefois, ils existent d'autres solution de prétraitement de données qui peuvent être intéressantes, nous citons a titre d'exemple SMOTE [52] et Bootstrap [53], [54]. Nous n'avons pas pu explorer ces solutions puisque les résultats obtenus en utilisant ré-échantillonnage avec la méthode « resample » sont assez satisfaisants.

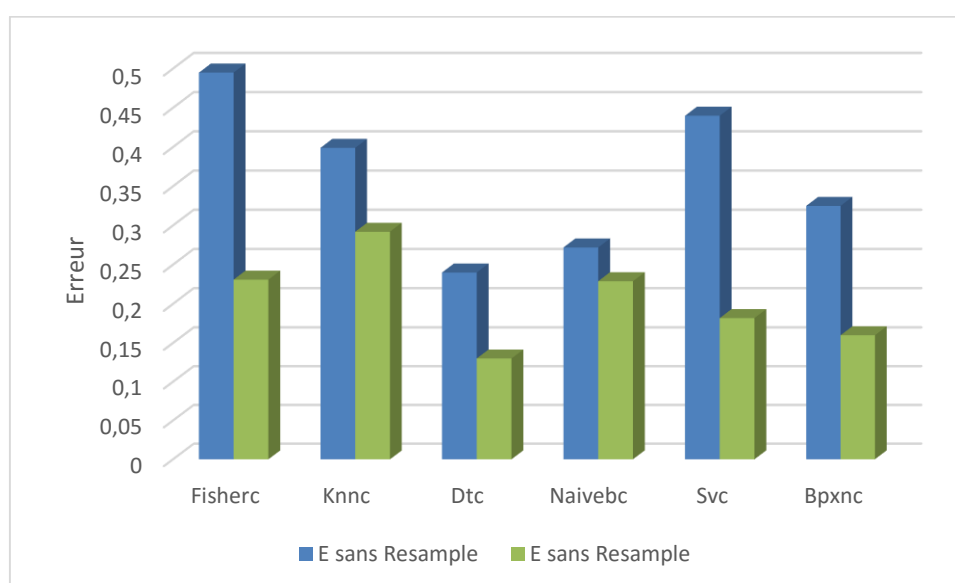


Figure II.1 Comparaison de l'erreur « avant » et « après » ré-échantillonnage

2.2.5. UTILISATION DE LA BASE DANS LA LITTERATURE

La base de données «Arrhythmia» est très riche en information. Ainsi, elle a fait l'objet de plusieurs recherches dans la littérature. Nous avons récapitulé les informations jugées utiles, sur la majorité des travaux récents, dans le Tableau II-5. Ainsi, nous remarquons que le taux de de précision dans les différentes expérimentations varie entre 66% et 95%.

Il convient à mentionner, qu'à partir de ce tableau, tous les chercheurs ont réduit le nombre de paramètres d'entrée que ce soit par sélection ou par extraction des attributs. Notant que l'approche de sélection est la plus utilisée vu que la majorité des chercheurs dans le domaine médical préfèrent ne pas perdre la sémantique des attributs afin qu'elle puisse être, éventuellement, analysée par un médecin expert.

Tableau II-5 Travaux réalisés en utilisant la base de données « Arrhythmia »

| <i>Auteurs</i> | <i>Nb classes</i> | <i>Méthode de sélection d'attributs</i> | <i>Type de réduction</i> | <i>Nombre d'attributs retenus</i> | <i>Classifieur</i> | <i>Résultats ¹(%)</i> |
|---------------------------------|-------------------|---|--------------------------|---|-----------------------------------|----------------------------------|
| <i>Namsrai et al. 2013 [55]</i> | 2 | Méthodes ensemblistes en utilisant 4 classifieurs et 3 sous ensemble de données | Wrapper | Vote majoritaire entre 3 ensembles d'attributs, Ils contiennent respectivement 24, 27, and 12 attributs | Naïve Bayes | 77,78 |
| | | | | | SVM | 66,67 |
| | | | | | Decision Tree | 95,24 |
| | | | | | Bayes Network | 91,43 |
| <i>Sotto et al. 2016[56]</i> | 16 | Programmation Kaizen + Programmation génétique linéaire | Extraction | 26 | Forêt aléatoire | 77,69 (±0,7) |
| | 2 | | | | 86,39 (±0,9) | |
| <i>Uyar et Gurgen 2007[57]</i> | 2 | Choix d'un cardiologue | Manuelle | 9 | Combinaison en série de SVM et LR | 0 (Rejet) 23,9 (Erreur) |
| | | | | | SVM-RO | 31,5 (Rejet) 10,8 (Erreur) |
| | | | | | SVM-LR | 11,9 (Rejet) 19,5 (Erreur) |
| | | | | | SVM-kNN | 13,0 (Rejet) 15,2 (Erreur) |
| <i>Gozde et al. 2014 [58]</i> | 2 | GA-NN | Wrapper | 9 | GA-NN | 86,75 |

¹ Les résultats sont donnés, de façon générale, selon le taux précision sauf quelques exceptions indiquées dans le tableau.

| | | | | | | |
|-------------------------------------|----|--|---------------------|--------------------------|--|--|
| <i>Khare et al. 2012 [59]</i> | 2 | Spearman Rank Correlation + PCA | Filter + extraction | 115 (Avant l'extraction) | SVM (Parametres: cost=.1 et Sigma=.002) | 85,98 |
| <i>Shilaskar et Swati 2013 [60]</i> | 2 | Model hybrid (Forward Feature Inclusion+ Back-elimination Feature Selection + Forward Feature Selection) | Filter | 23 | | 88 |
| <i>Deepthi et al. 2016 [61]</i> | 16 | ACP | Extraction | | Vote majoritaire (SVM, Arbre de décision, Naïve Bayes, k-NN et analyse des discriminants) LDA CART | 91,11 |
| <i>Bahadorinia et al. 2014 [62]</i> | 5 | ACP | Extraction | 16 | ANN+PSO ANN+LVQ ANN+GA | 85,36 78,04 80,48 |
| <i>Mitra et Samanta 2013 [63]</i> | 2 | Suppression des attributs contenant des valeurs manquantes ou nulles et utilisation de la méthode CFS + Algorithme de recherche linéaire forward | Filter | 18 | IBPLN LM | 86,02 87,72 |
| <i>Jadhav et al. 2013[64]</i> | 2 | Ensemble de classifieurs basé sur RS | Wrapper | 15-20 | Ensemble de classifieurs basé sur RS (PART) Bagging (J48) Bagging (Forêts aléatoires) Adaboost M1 (j48) Adaboost M1 (Forêts aléatoires) Dagging (règles JRip) | 91,11 88,89 84,44 88,89 80,00 80,00 |

2.3. SELECTION DES ATTRIBUTS AVEC L'APPROCHE FILTER

La fonction d'évaluation sert à mesurer soit la pertinence des attributs en les appréciant de manière individuelle, par l'utilisation d'un algorithme de sélection avec classement des attributs (filtre univarié), ou bien la pertinence des sous-ensembles d'attributs générés par l'une des différentes méthodes de génération, en utilisant un algorithme de recherche de sous-ensembles (filtre multivarié).

2.3.1. APPROCHE UNIVARIE

Dans ce type d'approche, les différents filtres permettent d'évaluer chaque attribut d'une manière indépendante. Par la suite une méthode de recherche « Ranker » est utilisé pour générer une liste classée des attributs en considérant leurs scores. Chaque score est calculé selon une méthode d'évaluation prédéfinie. Nous présentons dans la suite sept types d'évaluateurs ainsi que le classement des attributs retenus.

2.3.1.1. METHODE GAIN D'INFORMATION

La méthode nommée *InfoGainAttributeEval* évalue la valeur d'un attribut en mesurant le gain d'information par rapport à la classe. Le gain d'information (appelé aussi « Divergence de Kullback-Leibler » ou « entropie relative ») est une mesure de la dissimilarité entre deux lois de probabilités.

$$\text{Infogain}(\text{classe}, \text{attribut}) = H(\text{Classe}) - H(\text{Classe} | \text{Attribut}) \quad (4)$$

Notant que le gain d'information mesure l'efficacité d'un attribut A avec la réduction de l'entropie. L'entropie E d'un ensemble d'exemples S est évaluée comme suit : (supposons que l'attribut à prédire prenne M valeurs distinctes définissant M classes C_1, \dots, C_M)

$$E(S) = - \sum_{i=1}^M p_i \log_2 p_i \quad (5)$$

Où p_i désigne la proportion d'exemples de S appartenant à la i -ème C_i

A partir de l'entropie, on peut calculer *InfoGain* :

$$InfoGain(S, A) = E(S) - \sum_k^{k=m} \frac{|S_k|}{|S|} E(S_k) \quad (6)$$

Où l'attribut A prend les valeurs a_1, \dots, a_m et S_k est le sous-ensemble de S pour lequel l'attribut A prend la valeur a_k .

Le Tableau II-6 représente le score de gain d'information des 35 meilleurs attributs selon le calcul obtenu durant la phase d'apprentissage en appliquant le filtre *InfoGain* calculé par les formules cités ci-dessus.

Tableau II-6 Score des meilleurs attributs en utilisant la méthode InfoGain

| | | | | | | | | | | | | |
|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| N° | 113 | 227 | 198 | 261 | 232 | 237 | 101 | 14 | 5 | 11 | 196 | 51 |
| InfoGain | 0,169 | 0,167 | 0,159 | 0,149 | 0,141 | 0,139 | 0,136 | 0,127 | 0,126 | 0,123 | 0,12 | 0,118 |
| N° | 276 | 104 | 236 | 268 | 176 | 231 | 13 | 89 | 116 | 206 | 239 | 4 |
| InfoGain | 0,113 | 0,112 | 0,111 | 0,109 | 0,105 | 0,105 | 0,103 | 0,102 | 0,102 | 0,1 | 0,096 | 0,095 |
| N° | 100 | 207 | 136 | 178 | 269 | 99 | 278 | 166 | 75 | 111 | 259 | |
| InfoGain | 0,095 | 0,093 | 0,093 | 0,093 | 0,09 | 0,09 | 0,086 | 0,086 | 0,085 | 0,085 | 0,084 | |

2.3.1.2. METHODE GAINRATIO

La méthode nommée *GainRatioAttributeEval* évalue la valeur d'un attribut en mesurant le rapport de gain par rapport à la classe.

$$GainRatio (classe, attribut) = \frac{H (Classe) - H (Classe | Attribut)}{H (Attribut)} \quad (7)$$

En introduisant une information de partitionnement, *GainRation* ajuste *InfoGain* avec l'entropie du partitionnement. On peut le calculer avec les formules suivantes :

$$SplitInfo(S, A) = E(S) - \sum_k^{k=m} \frac{|S_k|}{|S|} \log_2 \frac{|S_k|}{|S|} \quad (8)$$

$$GainRatio(S, A) = \frac{InfoGain(S, A)}{SplitInfo(S, A)} \quad (9)$$

Les scores obtenus pour les 35 meilleurs attributs en utilisant *GainRatio* sont représentés dans le Tableau II-7.

Tableau II-7 Scores des meilleurs attributs en utilisant la méthode GainRatio

| | | | | | | | | | | | | |
|------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| N° | 99 | 111 | 237 | 247 | 221 | 242 | 236 | 249 | 260 | 212 | 102 | 233 |
| GainRatio | 0,246 | 0,242 | 0,239 | 0,232 | 0,232 | 0,227 | 0,222 | 0,222 | 0,217 | 0,217 | 0,217 | 0,217 |
| N° | 229 | 241 | 227 | 198 | 64 | 125 | 202 | 14 | 219 | 235 | 107 | 217 |
| GainRatio | 0,217 | 0,217 | 0,215 | 0,213 | 0,206 | 0,206 | 0,206 | 0,205 | 0,2 | 0,2 | 0,2 | 0,2 |
| N° | 182 | 256 | 135 | 225 | 140 | 65 | 250 | 32 | 120 | 216 | 29 | |
| GainRatio | 0,2 | 0,199 | 0,194 | 0,194 | 0,194 | 0,194 | 0,188 | 0,188 | 0,188 | 0,188 | 0,187 | |

2.3.1.3. METHODE KHI2

La méthode nommée *ChiSquaredAttributeEval* évalue les attributs en estimant, à l'aide d'un test de *khi2*, l'association entre l'attribut et une classe donnée. Le test *khi2* (*Chi2* ou χ^2) proposé par le statisticien Karl Pearson en 1900 [65] [66], permet de déterminer la probabilité que les lignes et les colonnes d'un tableau croisé soient indépendantes. Cela signifie que :

- Le fait d'appartenir à l'une des modalités de la première variable n'a aucune influence sur la modalité d'appartenance de la seconde variable ;
- Les pourcentages lignes du tableau croisé sont les mêmes pour toutes les lignes ;
- Les pourcentages colonnes du tableau croisé sont les mêmes pour toutes les colonnes.

Le test du *Khi2* se base sur la valeur du χ^2 du tableau, qui est une mesure de l'écart entre le tableau observé et le tableau qu'on aurait obtenu si les variables étaient parfaitement

indépendantes, et sur le nombre de degrés de liberté du tableau, qui dépend du nombre de lignes et de colonnes. Le calcul du χ^2 se fait en deux phases :

A partir du tableau de contingence, il faut calculer les effectifs théoriques qui correspondent à l'hypothèse nulle d'indépendance. Le calcul de ces effectifs se fait de la manière suivante :

$$\text{Total de la ligne concernée} * \text{Total de la colonne concernée} / \text{Total général}$$

Pour chaque cellule, le χ^2 se calcul avec la formule suivante :

$$(\text{Observations} - \text{Effectifs théoriques})^2 / \text{Effectifs théoriques}.$$

Les scores des 35 meilleurs attributs en utilisant χ^2 sont représentés dans le Tableau II-8.

Tableau II-8 Score des meilleurs attributs en utilisant la méthode Khi2

| | | | | | | | | | | | | |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| N^\bullet | 113 | 227 | 232 | 198 | 261 | 101 | 11 | 5 | 196 | 237 | 14 | 276 |
| χ^2 | 54,48 | 52,85 | 45,89 | 45,85 | 45,79 | 44,56 | 42,84 | 41,96 | 41,89 | 39,94 | 38,3 | 38,2 |
| N^\bullet | 231 | 51 | 176 | 13 | 104 | 206 | 268 | 178 | 236 | 89 | 116 | 269 |
| χ^2 | 36,03 | 36,03 | 35,97 | 35,66 | 34,78 | 34,03 | 33,98 | 32,92 | 31,85 | 31,48 | 31,38 | 31,21 |
| N^\bullet | 239 | 136 | 278 | 166 | 100 | 207 | 259 | 4 | 248 | 75 | 168 | |
| χ^2 | 31,15 | 30,88 | 30,48 | 30,16 | 29,79 | 29,63 | 29,56 | 28,78 | 28,4 | 28,24 | 27,91 | |

2.3.1.4. METHODE RELIEFF

La méthode nommée *ReliefAttributeEval* évalue la valeur d'un attribut par échantillonnage aléatoire répété d'une instance et prend en compte la valeur de l'attribut donné pour l'instance la plus proche de la même classe ou de classe différente. Elle peut fonctionner sur des données de classe discrètes ou continues. Cet algorithme a été introduit dans [67] sous le nom de *Relief* puis amélioré et adapté au cas multi-classes sous le nom de *ReliefF* [68]. Les scores des 35 meilleurs attributs en utilisant *ReliefF* sont représentés dans le Tableau II-9

Tableau II-9 Score des meilleurs attributs en utilisant la méthode *ReliefF*

| | | | | | | | | | | | | |
|-----------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| <i>N°</i> | 92 | 39 | 40 | 101 | 2 | 44 | 251 | 102 | 17 | 14 | 65 | 46 |
| <i>ReliefF</i> | 0,038 | 0,034 | 0,030 | 0,029 | 0,029 | 0,028 | 0,028 | 0,027 | 0,026 | 0,026 | 0,025 | 0,025 |
| <i>N°</i> | 168 | 41 | 113 | 90 | 99 | 209 | 111 | 76 | 190 | 241 | 178 | 77 |
| <i>ReliefF</i> | 0,023 | 0,023 | 0,022 | 0,022 | 0,021 | 0,021 | 0,02 | 0,02 | 0,018 | 0,018 | 0,018 | 0,018 |
| <i>N°</i> | 255 | 198 | 7 | 266 | 276 | 16 | 64 | 240 | 196 | 176 | 249 | |
| <i>ReliefF</i> | 0,017 | 0,017 | 0,017 | 0,017 | 0,017 | 0,017 | 0,015 | 0,015 | 0,014 | 0,014 | 0,014 | |

2.3.1.5. METHODE INCERTITUDE SYMETRIQUE

La méthode nommée *SymmetricalUncertAttributeEval* (*SymmU*) évalue un attribut en mesurant son incertitude symétrique par rapport à la classe. Les résultats de cette méthodes sont décrits dans le tableau II-10.

$$SymmU(Classe, Attribut) = \frac{2 * (H(Classe) - H(Classe | Attribut))}{H(Classe) + H(Attribut)}. \quad (10)$$

Tableau II-10 Score des meilleurs attributs en utilisant la méthode *SymmU*

| | | | | | | | | | | | | |
|---------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| <i>N°</i> | 227 | 198 | 237 | 113 | 14 | 232 | 101 | 236 | 276 | 196 | 99 | 104 |
| <i>SymmU</i> | 0,189 | 0,184 | 0,178 | 0,160 | 0,158 | 0,155 | 0,154 | 0,150 | 0,141 | 0,137 | 0,133 | 0,130 |
| <i>N°</i> | 111 | 116 | 206 | 51 | 11 | 256 | 77 | 219 | 231 | 247 | 221 | 100 |
| <i>SymmU</i> | 0,127 | 0,126 | 0,124 | 0,12 | 0,118 | 0,116 | 0,115 | 0,115 | 0,114 | 0,114 | 0,114 | 0,114 |
| <i>N°</i> | 261 | 207 | 136 | 5 | 242 | 239 | 199 | 92 | 29 | 240 | 230 | |
| <i>SymmU</i> | 0,113 | 0,112 | 0,11 | 0,108 | 0,107 | 0,105 | 0,104 | 0,104 | 0,104 | 0,104 | 0,103 | |

2.3.1.6. METHODE SVM

Cette méthode nommée *SVMAttributeEval* évalue les attributs en utilisant une élimination récursive des attributs avec le classifieur *SVM* (*Support Vector Machine*). Les attributs sont sélectionnés un par un en fonction de la taille de leurs coefficients de réapprentissage du fait que le classifieur *SVM* permet d'attribuer un poids à chaque attribut. Divers paramètres sont transmis au classifieur *SVM* : complexité, epsilon, la tolérance, et la méthode de

filtrage utilisée [69]. Le classement des 35 meilleurs attributs en utilisant *SVM* sont représentés dans le Tableau II-11.

Tableau II-11 Rang meilleurs attributs en utilisant la méthode SVM

| | | | | | | | | | | | | |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| N° | 92 | 276 | 99 | 90 | 206 | 190 | 4 | 68 | 15 | 102 | 46 | 170 |
| RangSVM | 278 | 277 | 276 | 275 | 274 | 273 | 272 | 271 | 270 | 269 | 268 | 267 |
| N° | 81 | 146 | 34 | 60 | 266 | 62 | 24 | 77 | 40 | 162 | 120 | 259 |
| RangSVM | 266 | 265 | 264 | 263 | 262 | 261 | 260 | 259 | 258 | 257 | 256 | 255 |
| N° | 160 | 111 | 208 | 29 | 56 | 253 | 116 | 117 | 35 | 9 | 211 | |
| RangSVM | 254 | 253 | 252 | 251 | 250 | 249 | 248 | 247 | 246 | 245 | 244 | |

2.3.1.7. METHODE ONER

Cette méthode nommée *OneRAttributeEval* évalue la valeur d'un attribut en utilisant le classifieur *OneR* (*one-attribute-rule*) [70]. Le principe de la conception de cet algorithme consiste à trouver l'attribut qui présente le moins d'erreurs de prédiction possibles afin de l'utiliser par la suite. Le rang des 35 meilleurs attributs en utilisant *OneR* sont représentés dans le Tableau II-12.

Tableau II-12 Score des meilleurs attributs en utilisant le filtre OneR

| | | | | | | | | | | | | |
|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| N° | 196 | 101 | 113 | 168 | 11 | 248 | 178 | 227 | 9 | 237 | 10 | 5 |
| RangOneR | 75,19 | 73,64 | 72,87 | 70,54 | 69,77 | 68,22 | 68,22 | 67,83 | 67,05 | 67,05 | 67,05 | 66,67 |
| N° | 125 | 232 | 89 | 172 | 148 | 99 | 207 | 246 | 92 | 225 | 259 | 177 |
| RangOneR | 66,67 | 66,28 | 66,28 | 66,28 | 66,28 | 65,89 | 65,89 | 65,89 | 65,89 | 65,89 | 65,89 | 65,89 |
| N° | 202 | 162 | 276 | 268 | 116 | 222 | 239 | 136 | 68 | 29 | 235 | |
| RangOneR | 65,89 | 65,5 | 65,5 | 65,5 | 65,5 | 65,12 | 64,73 | 64,73 | 64,34 | 64,34 | 64,34 | |

2.3.2. APPROCHE MULTIVARIE

Dans ce type d'approche, on prend un sous-ensemble d'attributs et on renvoie une mesure numérique qui guide la recherche. Deux méthodes de filtres multivariés ont été utilisé (Méthode CFS et méthode de consistance):

2.3.2.1. METHODE CFS

Cette méthode nommée *CfsSubsetEval* évalue la capacité prédictive de chaque attribut individuellement et le degré de redondance entre eux, préférant les ensembles d'attributs qui sont fortement corrélés avec la classe, mais avec une faible intercorrélacion. Une option ajoute itérativement les attributs qui ont la plus forte corrélation avec la classe, à condition que l'ensemble ne contienne pas déjà un attribut dont la corrélation avec l'attribut en question est encore plus élevée. Les algorithmes de recherche traversent l'espace d'attribut pour trouver un bon sous-ensemble. A l'instar de la sous-section précédente, nous avons utilisé le logiciel open source Weka pour obtenir les sous ensembles d'attributs avec le filtre *CfsSubsetEval*, en ce basant sur quatre algorithmes de recherche différents.

a. ALGORITHME BESTFIRST

L'algorithme *BestFirst* (ou premier d'abord) recherche l'espace de sous-ensembles d'attributs par escalade de glouton (*greedy hill climbing*) avec un centre de retour en arrière. L'idée principale est d'utiliser une fonction d'évaluation afin d'estimer l'intérêt des nœuds et développer le nœud le plus intéressant. Le nœud à développer est choisi selon une fonction d'évaluation tandis que le coût du chemin le plus court pour se rendre au but est estimé par une fonction heuristique.

Best-First peut être initiée par l'ensemble vide d'attributs et recherche vers l'avant, ou par l'ensemble complet d'attributs et recherche vers l'arrière, ou encore à tout moment par la recherche dans les deux directions (en tenant compte de toutes les possibilités d'ajout et de suppression d'attributs uniques à un moment donné).

L'algorithme *BestFirst* a retenu 25 attributs :

4,14,75,77,92,102,111,113,168,178,186,196,198,199,221,227,232,235,237,246,249,260,261,263,276.

b. ALGORITHME GENETICSEARCH

La recherche génétique utilise un algorithme génétique simple [17]. Les paramètres comprennent la taille de la population, le nombre de générations, et les probabilités de croisement et de mutation.

Les attributs retenus en utilisant l'algorithme génétique sont au nombre de 70: 2,3,5,10,13,20,27,32,35,36,40,41,44,46,50,51,54,59,68,72,73,84,85,88,90,92,98,99,101,105,109,112,118,123,125,126,133,135,140,144,152,157,161,166,167,173,178,186,189,190,192,196,198,199,201,203,210,217,232,237,240,242,245,246,256,261,271,274,276,278.

c. ALGORITHME GREEDYSTEPWISE

Il effectue une recherche gloutonne (*greedy*) vers l'avant (*forward*) ou vers l'arrière (*backward*) à travers l'espace de sous-ensembles d'attributs. Il peut commencer sans aucun attribut, avec tous les attributs ou d'un point arbitraire dans l'espace. Le processus s'arrête lorsque l'addition/suppression de tous les attributs restants donne une diminution de l'évaluation. Il peut également produire une liste de classement des attributs en traversant l'espace d'un côté à l'autre et en enregistrant l'ordre selon lequel les attributs sont sélectionnés.

En utilisant la recherche vers l'avant (*forward*), nous avons obtenu 25 attributs :

4,14,75,77,92,102,111,113,168,178,186,196,198,199,221,227,232,235,237,246,249,260,261,263,276.

En appliquant la recherche vers l'arrière, nous avons obtenu 29 attributs :

4,13,14,17,75,77,92,99,102,113,168,178,186,196,198,199,221,223,227,232,236,237,247,249,256,260,261,263,276.

2.3.2.2. METHODE DE CONSISTANCE

Cette méthode appelée *ConsistencySubsetEval* évalue les ensembles d'attributs par le degré de cohérence dans les valeurs de la classe lorsque les instances d'apprentissage sont

projetées sur l'ensemble. La cohérence d'un sous-ensemble d'attributs ne peut jamais améliorer celle de l'ensemble, de sorte que l'évaluateur est généralement utilisé en conjonction avec une recherche aléatoire ou exhaustive qui cherche le plus petit sous-ensemble avec une cohérence qui est la même que celle de l'ensemble d'attributs complet. Cependant, les deux types de recherche sont connus pour leur complexité, c'est pourquoi nous avons utilisé un ordinateur plus puissant (Processus i7-3770 CPU 3.40 GHz, RAM 8 Go). Malgré cela, après plusieurs semaines d'exécution, nous avons décidé d'abandonner la recherche avec l'algorithme exhaustive, au vue de sa complexité très élevée.

d. ALGORITHME DE RECHERCHE ALEATOIRE

Cet algorithme effectue une recherche aléatoire (*RandomSearch*) dans l'espace de sous-ensembles d'attributs. Si aucun réglage de démarrage n'est fourni, la recherche aléatoire commence à partir d'un point aléatoire et signale le meilleur sous-ensemble trouvé. Si un ensemble de démarrage est fourni, des recherches aléatoires sont effectuées pour des sous-ensembles qui sont aussi bons ou meilleurs que le point de départ avec le même nombre d'attributs ou moins [71]. Cependant cet algorithme a une grande complexité algorithmique, nous avons attendu 8 jours pour obtenir un sous ensemble de 134 attributs :

1,3,5,6,7,8,9,10,12,14,15,16,17,21,22,24,26,29,32,35,36,38,39,43,45,47,48,55,56,57,58,60,62,63,66,68,69,71,72,74,75,79,81,82,84,89,90,91,92,96,97,99,100,105,106,109,111,113,116,120,121,124,127,128,129,130,131,135,140,142,145,149,151,153,161,163,165,166,168,170,171,172,173,175,176,179,181,182,183,184,185,189,190,191,193,195,202,204,206,207,210,213,214,221,223,225,227,228,230,233,236,237,239,240,241,243,245,246,248,251,254,257,258,261,266,267,268,269,270,271,272,275,276,278.

2.4. RESULTATS OBTENUS

Dans cette section, nous allons présenter les résultats obtenus selon le type du filtre, à savoir univarié et multivarié.

2.4.1. FILTRE UNIVARIE

Afin de tester ces méthodes, nous avons utilisé les classifieurs cités précédemment en variant le nombre d'attributs de : 5,10, 15, jusqu'à 35.

2.4.1.1. METHODE INFOGAIN

L'erreur obtenue pour chaque classifieur en utilisant le filtre *InfoGain* est résumée dans le Tableau II-13. Nous remarquons pour les algorithmes suivants que :

- *Fisherc* a obtenu le meilleur taux d'erreur en utilisant **35** attributs ;
- *Knnc* a obtenu le meilleur taux d'erreur en utilisant **10** attributs ;
- *Dtc* a obtenu le meilleur taux d'erreur en utilisant **20** ou **25** attributs ;
- *Naivebc* a obtenu le meilleur taux d'erreur en utilisant **35** attributs ;
- *Svc* a obtenu le meilleur taux d'erreur en utilisant **30** attributs ;
- *Bpxnc* a obtenu le meilleur taux d'erreur en utilisant **25** attributs ;
- En moyenne, le meilleur taux d'erreur (**0,204**) a été obtenu en utilisant **35** attributs ;
- La majorité des classifieurs ont obtenu le meilleur taux d'erreur en utilisant **25** ou **35** attributs.

Tableau II-13 Erreur obtenue avec la méthode *InfoGain*

| <i>Classifieur/NbAttributs</i> | 5 | 10 | 15 | 20 | 25 | 30 | 35 |
|--------------------------------|----------|-----------|-----------|-----------|-----------|-----------|-----------|
| <i>Fisherc</i> | 0,384 | 0,284 | 0,271 | 0,258 | 0,268 | 0,244 | 0,233 |
| <i>Knnc</i> | 0,211 | 0,169 | 0,167 | 0,201 | 0,245 | 0,237 | 0,213 |
| <i>Dtc</i> | 0,23 | 0,206 | 0,16 | 0,151 | 0,151 | 0,231 | 0,167 |
| <i>Naivebc</i> | 0,302 | 0,266 | 0,233 | 0,241 | 0,255 | 0,236 | 0,214 |
| <i>Svc</i> | 0,388 | 0,315 | 0,281 | 0,251 | 0,251 | 0,23 | 0,237 |
| <i>Bpxnc</i> | 0,317 | 0,24 | 0,168 | 0,193 | 0,16 | 0,184 | 0,162 |

2.4.1.2. METHODE GAINRATIO

En observant l'erreur obtenue en utilisant le filtre *GainRatio* résumé dans le Tableau II-14, nous remarquons que les algorithmes suivants:

- *Fisherc* a obtenu le meilleur taux d'erreur en utilisant **20** attributs ;
- *Knnc* a obtenu le meilleur taux d'erreur en utilisant **30** attributs ;
- *Dtc* a obtenu le meilleur taux d'erreur en utilisant **25** et **35** attributs ;
- *Naivebc* a obtenu le meilleur d'erreur taux en utilisant **35** attributs ;
- *Svc* a obtenu le meilleur d'erreur taux en utilisant **35** attributs ;
- *Bpxnc* a obtenu le meilleur taux d'erreur en utilisant **35** attributs ;
- En moyenne, le meilleur taux (**0,226**) a été obtenu en utilisant **35** attributs ;
- La majorité des classifieurs ont obtenu le meilleur taux d'erreur en utilisant **35** attributs.

Tableau II-14 Erreur obtenue avec la méthode *GainRatio*

| <i>Classifieur/NbAttributs</i> | 5 | 10 | 15 | 20 | 25 | 30 | 35 |
|--------------------------------|----------|-----------|-----------|-----------|-----------|-----------|-----------|
| <i>Fisherc</i> | 0,38 | 0,349 | 0,309 | 0,285 | 0,313 | 0,314 | 0,29 |
| <i>Knnc</i> | 0,284 | 0,255 | 0,285 | 0,208 | 0,165 | 0,162 | 0,22 |
| <i>Dtc</i> | 0,351 | 0,268 | 0,229 | 0,173 | 0,172 | 0,172 | 0,18 |
| <i>Naivebc</i> | 0,354 | 0,313 | 0,284 | 0,265 | 0,269 | 0,275 | 0,26 |
| <i>Svc</i> | 0,356 | 0,339 | 0,302 | 0,292 | 0,293 | 0,295 | 0,25 |
| <i>Bpxnc</i> | 0,321 | 0,278 | 0,261 | 0,209 | 0,216 | 0,218 | 0,17 |

2.4.1.3. METHODE CHI2

En observant l'erreur obtenue en utilisant le filtre *Chi2* résumé dans le Tableau II-15. Nous remarquons que les algorithmes suivants:

- *Fisherc* a obtenu le meilleur taux d'erreur en utilisant **30 et 35** attributs ;
- *Knnc* a obtenu le meilleur taux d'erreur en utilisant **15** attributs ;
- *Dtc* a obtenu le meilleur taux d'erreur en utilisant **25** attributs ;
- *Naivebc* a obtenu le meilleur taux d'erreur en utilisant **20** attributs ;
- *Svc* a obtenu le meilleur taux d'erreur en utilisant **35** attributs ;
- *Bpxnc* a obtenu le meilleur taux d'erreur en utilisant **30** attributs ;
- En moyenne, le meilleur taux d'erreur (**0,2**) a été obtenu en utilisant **35** attributs ;
- La majorité des classifieurs ont obtenu le meilleur taux d'erreur en utilisant **30 et 35** attributs.

Tableau II-15 Erreur obtenue avec la méthode *Chi2*

| <i>Classifieur/NbAttributs</i> | <i>5</i> | <i>10</i> | <i>15</i> | <i>20</i> | <i>25</i> | <i>30</i> | <i>35</i> |
|--------------------------------|----------|-----------|-----------|-----------|-----------|-----------|-----------|
| <i>Fisherc</i> | 0,384 | 0,313 | 0,273 | 0,253 | 0,278 | 0,237 | 0,237 |
| <i>Knnc</i> | 0,211 | 0,204 | 0,159 | 0,179 | 0,196 | 0,211 | 0,221 |
| <i>Dtc</i> | 0,23 | 0,161 | 0,204 | 0,15 | 0,138 | 0,15 | 0,147 |
| <i>Naivebc</i> | 0,302 | 0,29 | 0,227 | 0,213 | 0,241 | 0,248 | 0,237 |
| <i>Svc</i> | 0,388 | 0,3 | 0,244 | 0,259 | 0,247 | 0,239 | 0,219 |
| <i>Bpxnc</i> | 0,321 | 0,28 | 0,206 | 0,163 | 0,134 | 0,131 | 0,138 |

2.4.1.4. METHODE RELIEFF

En considérant l'erreur obtenue en utilisant le filtre *Relieff* résumé dans le Tableau II-16 nous remarquons que les algorithmes suivant :

- *Fisherc* a obtenu le meilleur taux d'erreur en utilisant **20** attributs ;
- *Knnc* a obtenu le meilleur taux d'erreur en utilisant **10** attributs ;
- *Dtc* a obtenu le meilleur taux d'erreur en utilisant **30** attributs ;
- *Naivebc* a obtenu le meilleur taux d'erreur en utilisant **20** attributs ;
- *Svc* a obtenu le meilleur taux d'erreur en utilisant **25** attributs ;
- *Bpxnc* a obtenu le meilleur taux d'erreur en utilisant **25** attributs ;
- En moyenne, le meilleur taux d'erreur (**0,201**) a été obtenu en utilisant **30** attributs ;
- La majorité des classifieurs ont obtenu le meilleur taux d'erreur en utilisant **30** attributs.

Tableau II-16 Erreur obtenue avec la méthode *Relieff*

| <i>Classifieur/NbAttributs</i> | <i>5</i> | <i>10</i> | <i>15</i> | <i>20</i> | <i>25</i> | <i>30</i> | <i>35</i> |
|--------------------------------|----------|-----------|-----------|-----------|-----------|-----------|-----------|
| <i>Fisherc</i> | 0,36 | 0,386 | 0,305 | 0,253 | 0,279 | 0,278 | 0,305 |
| <i>Knnc</i> | 0,189 | 0,129 | 0,141 | 0,179 | 0,154 | 0,184 | 0,164 |
| <i>Dtc</i> | 0,26 | 0,146 | 0,187 | 0,15 | 0,141 | 0,119 | 0,15 |
| <i>Naivebc</i> | 0,308 | 0,308 | 0,286 | 0,213 | 0,23 | 0,24 | 0,224 |
| <i>Svc</i> | 0,341 | 0,367 | 0,255 | 0,259 | 0,273 | 0,231 | 0,281 |
| <i>Bpxnc</i> | 0,257 | 0,221 | 0,187 | 0,163 | 0,231 | 0,156 | 0,169 |

2.4.1.5. METHODE INCERTITUDE SYMETRIQUE

L'erreur obtenue en utilisant le filtre *SymmU* est résumé dans le Tableau II-17. Nous remarquons que les algorithmes suivants :

- *Fisherc* a obtenu le meilleur taux d'erreur en utilisant **30** attributs ;
- *Knnc* a obtenu le meilleur taux d'erreur en utilisant **5** attributs ;
- *Dtc* a obtenu le meilleur taux d'erreur en utilisant **15** attributs ;
- *Naivebc* a obtenu le meilleur taux d'erreur en utilisant **10** attributs ;
- *Svc* a obtenu le meilleur taux d'erreur en utilisant **30** attributs ;
- *Bpxnc* a obtenu le meilleur taux d'erreur en utilisant **35** attributs ;
- En moyenne, le meilleur taux d'erreur (**0,211**) a été obtenu en utilisant **35** attributs ;
- La majorité des classifieurs ont obtenu le meilleur taux d'erreur en utilisant **30** attributs.

Tableau II-17 Erreur obtenue avec la méthode SymmU

| Classifieur/NbAttributs | 5 | 10 | 15 | 20 | 25 | 30 | 35 |
|-------------------------|-------|-------|-------|-------|-------|-------|-------|
| <i>Fisherc</i> | 0,351 | 0,362 | 0,312 | 0,366 | 0,266 | 0,21 | 0,247 |
| <i>Knnc</i> | 0,128 | 0,182 | 0,215 | 0,281 | 0,189 | 0,227 | 0,174 |
| <i>Dtc</i> | 0,197 | 0,184 | 0,136 | 0,26 | 0,17 | 0,16 | 0,153 |
| <i>Naivebc</i> | 0,275 | 0,244 | 0,282 | 0,334 | 0,264 | 0,294 | 0,294 |
| <i>Svc</i> | 0,362 | 0,328 | 0,288 | 0,336 | 0,225 | 0,199 | 0,245 |
| <i>Bpxnc</i> | 0,177 | 0,194 | 0,176 | 0,312 | 0,206 | 0,204 | 0,153 |

2.4.1.6. METHODE SVM

En observant l'erreur obtenue en utilisant le filtre *SVM* résumé dans Tableau II-18 Nous remarquons que les algorithmes suivants:

- *Fisherc* a obtenu le meilleur taux d'erreur en utilisant **20** attributs ;
- *Knnc* a obtenu le meilleur taux d'erreur en utilisant **15** attributs ;
- *Dtc* a obtenu le meilleur taux d'erreur en utilisant **30** attributs ;
- *Naivebc* a obtenu le meilleur taux d'erreur en utilisant **35** attributs ;
- *Svc* a obtenu le meilleur taux d'erreur en utilisant **35** attributs ;

- *Bpxnc* a obtenu le meilleur taux d'erreur en utilisant **30** attributs ;
- En moyenne, le meilleur taux d'erreur (**0.199**) a été obtenu en utilisant **30** attributs ;
- La majorité des classifieurs ont obtenu le meilleur taux d'erreur en utilisant **30 et 35** attributs.

Tableau II-18 Erreur obtenue avec la méthode SVM

| <i>Classifieur/NbAttributs</i> | 5 | 10 | 15 | 20 | 25 | 30 | 35 |
|--------------------------------|----------|-----------|-----------|-----------|-----------|-----------|-----------|
| <i>Fisherc</i> | 0,272 | 0,229 | 0,229 | 0,217 | 0,261 | 0,239 | 0,244 |
| <i>Knnc</i> | 0,189 | 0,168 | 0,156 | 0,318 | 0,22 | 0,252 | 0,243 |
| <i>Dtc</i> | 0,183 | 0,205 | 0,191 | 0,208 | 0,208 | 0,145 | 0,189 |
| <i>Naivebc</i> | 0,198 | 0,235 | 0,218 | 0,214 | 0,198 | 0,203 | 0,195 |
| <i>Svc</i> | 0,258 | 0,246 | 0,234 | 0,244 | 0,256 | 0,204 | 0,193 |
| <i>Bpxnc</i> | 0,22 | 0,261 | 0,187 | 0,179 | 0,179 | 0,151 | 0,155 |

2.4.1.7. METHODE ONER

En observant l'erreur obtenue en utilisant le filtre *OneR* résumé dans Tableau II-19 nous remarquons que les algorithmes suivants :

- *Fisherc* a obtenu le meilleur taux d'erreur en utilisant **20** attributs ;
- *Knnc* a obtenu le meilleur taux d'erreur en utilisant **5 et 10** attributs ;
- *Dtc* a obtenu le meilleur taux d'erreur en utilisant **15** attributs ;
- *Naivebc* a obtenu le meilleur taux d'erreur en utilisant **25** attributs ;
- *Svc* a obtenu le meilleur taux d'erreur en utilisant **25** attributs ;
- *Bpxnc* a obtenu le meilleur taux d'erreur en utilisant **25** attributs ;
- En moyenne, le meilleur taux d'erreur (**0,285**) a été obtenu en utilisant **25** attributs ;
- La majorité des classifieurs ont obtenu le meilleur taux d'erreur en utilisant **25** attributs.

Tableau II-19 Erreur obtenue avec la méthode OneR

| <i>Classifieur/NbAttributs</i> | 5 | 10 | 15 | 20 | 25 | 30 | 35 |
|--------------------------------|----------|-----------|-----------|-----------|-----------|-----------|-----------|
| <i>Fisherc</i> | 0,316 | 0,326 | 0,314 | 0,251 | 0,28 | 0,275 | 0,291 |
| <i>Knnc</i> | 0,187 | 0,187 | 0,262 | 0,253 | 0,257 | 0,257 | 0,248 |
| <i>Dtc</i> | 0,282 | 0,164 | 0,148 | 0,203 | 0,181 | 0,172 | 0,196 |

| | | | | | | | |
|----------------|-------|-------|-------|-------|-------|-------|-------|
| <i>Naivebc</i> | 0,277 | 0,266 | 0,265 | 0,285 | 0,253 | 0,26 | 0,255 |
| <i>Svc</i> | 0,313 | 0,311 | 0,307 | 0,229 | 0,22 | 0,239 | 0,259 |
| <i>Bpxnc</i> | 0,297 | 0,277 | 0,197 | 0,306 | 0,17 | 0,203 | 0,203 |

D'un autre côté si on prend en considération les performances individuelles de chaque classifieur (voir Figure II.2) nous remarquons que :

- > Le classifieur avec le meilleur taux, c'est-à-dire le classifieur qui minimise le plus l'erreur de classification pour chaque méthode est résumé ci-dessous :

Méthode InfoGain : classifieur *Dtc* en utilisant **20** ou **25** attributs ;
Méthode GainRatio : classifieur *Knnc* en utilisant **30** attributs ;
Méthode Chi2 : classifieur *Bpxnc* en utilisant **30** attributs ;
Méthode ReliefF : classifieur *Dtc* en utilisant **30** attributs ;
Méthode SymmU : classifieur *Knnc* en utilisant **5** attributs ;
Méthode SVM : classifieur *Dtc* en utilisant **30** attributs ;
Méthode OneR : classifieur *Dtc* en utilisant **15** attributs.

- > Parallèlement, le classifieur le moins performormant pour chaque méthodes est :

Méthode InfoGain : classifieur *Svc* en utilisant **5** attributs ;
Méthode GainRatio : classifieur *Fisherc* en utilisant **5** attributs ;
Méthode Chi2 : classifieur *0.388* en utilisant **5** attributs ;
Méthode ReliefF : classifieur *0,386* en utilisant **10** attributs ;
Méthode SymmU : classifieur *Fisherc* en utilisant **20** attributs ;
Méthode SVM : classifieur *Knnc* en utilisant **20** attributs ;
Méthode OneR : classifieur *Fisherc* en utilisant **10** attributs.

- > Le classifieur avec le meilleur qui minimise le taux d'erreur selon selon la moyenne globale en variant le nombre d'attributs pour chaque méthode est résumé ci-dessous :

Méthode InfoGain : classifieur *Dtc* avec la moyenne de **0,185** ;
Méthode GainRatio : classifieur *Dtc* avec la moyenne de **0,2** ;
Méthode Chi2 : classifieur *Dtc* avec la moyenne de **0,168** ;
Méthode ReliefF : classifieur *Knnc* avec la moyenne de **0,162** ;
Méthode SymmU : classifieur *Dtc* avec la moyenne de **0,18** ;
Méthode SVM : classifieur *Dtc* avec la moyenne de **0,189** ;
Méthode OneR : classifieur *Dtc* avec la moyenne de **0,192**.

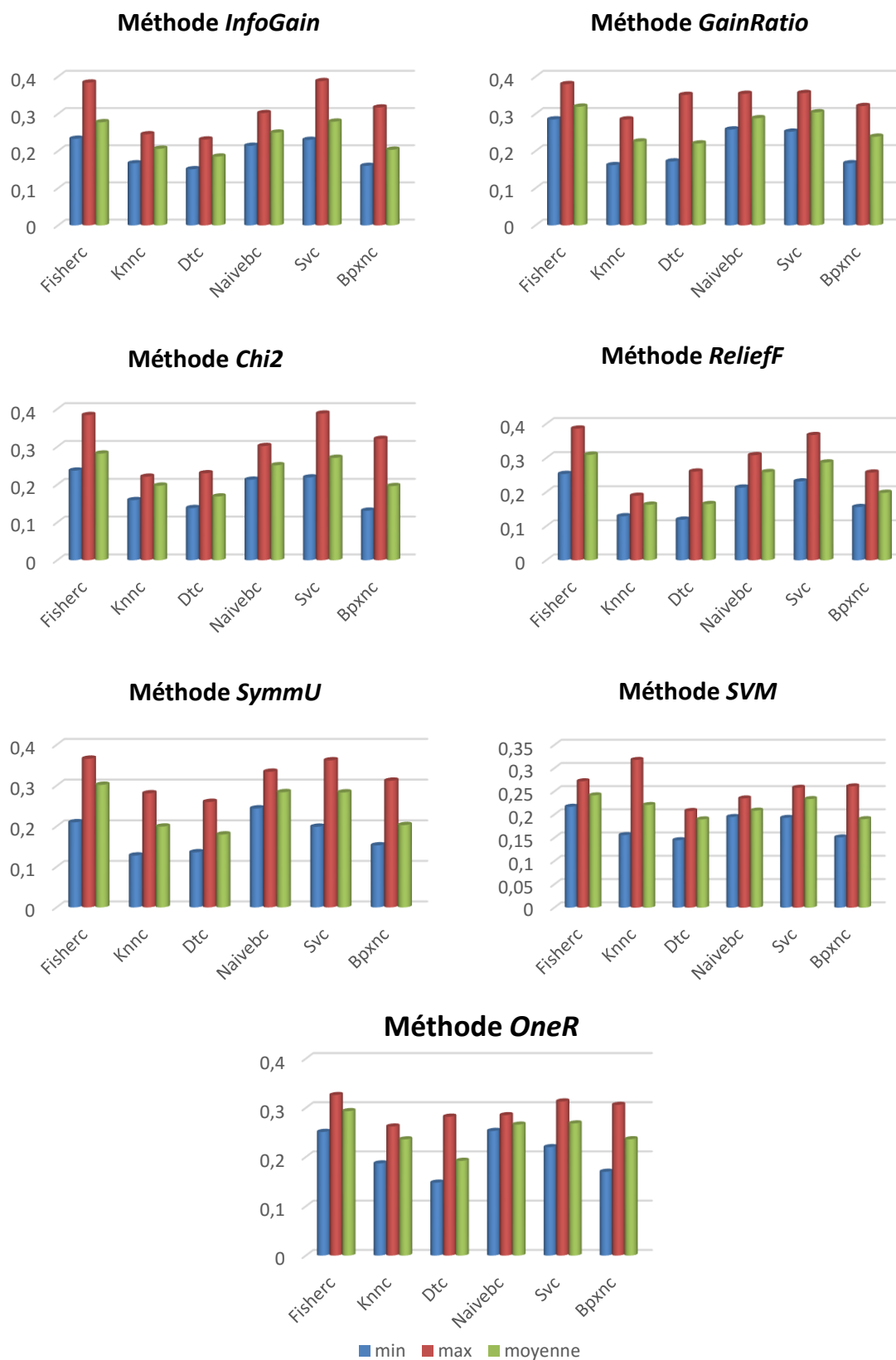


Figure II.2 Comparaison des classifieurs en appliquant les différentes méthodes

2.4.2. FILTRE MULTIVARIE

2.4.2.1. METHODE CFS

Afin de tester les attributs obtenus avec le filtre CFS en utilisant les différents algorithmes de recherches (quatre), nous les avons évalués avec les mêmes classifieurs cités précédemment. Les résultats sont résumés dans Tableau II-20.

Tableau II-20 Résultats obtenus en utilisant *CFS*

| <i>Classifieur/Algo de recherche</i> | <i>BestFirst</i> | <i>Forward</i> | <i>Backward</i> | <i>GeneticAlgorithm</i> |
|--------------------------------------|------------------|----------------|-----------------|-------------------------|
| <i>Fisherc</i> | 0,254 | 0,263 | 0,277 | 0,251 |
| <i>Knnc</i> | 0,146 | 0,203 | 0,147 | 0,259 |
| <i>Dtc</i> | 0,127 | 0,15 | 0,165 | 0,144 |
| <i>Naivebc</i> | 0,212 | 0,212 | 0,208 | 0,231 |
| <i>Svc</i> | 0,202 | 0,287 | 0,271 | 0,291 |
| <i>Bpxnc</i> | 0,137 | 0,198 | 0,231 | 0,182 |

Nous remarquons que le meilleur résultat a été obtenu par le classifieur *Dtc* en utilisant l'algorithme *BestFirst* ($e=0,127$) et le plus mauvais a été obtenu par le classifieur *Svc* en utilisant l'algorithme *Forward* ($e=0,287$).

2.4.2.2. METHODE CONSISTANCE

Pour le filtre nommé *ConsistencySubsetEval*, nous avons utilisé un algorithme de recherche aléatoire (Random Search), les résultats sont résumés dans le Tableau II-21.

Tableau II-21 Résultats obtenus en utilisant *ConsistencySubsetEval*

| <i>Classifieur/Algo de recherche</i> | <i>RansomSearch</i> |
|--------------------------------------|---------------------|
| <i>Fisherc</i> | 0.273 |
| <i>Knnc</i> | 0.257 |
| <i>Dtc</i> | 0.220 |
| <i>Naivebc</i> | 0.119 |
| <i>Svc</i> | 0.212 |
| <i>Bpxnc</i> | 0.207 |

2.5. DISCUSSION DES RESULTATS

Dans ce chapitre, nous avons comparé plusieurs méthodes de filtrage en les subdivisant en deux catégories. Dans la première catégorie, nous avons exploré les filtres de type univarié. D'après les résultats obtenus, nous pouvons en tirer les remarques suivantes :

- ⦿ Les filtres de type univarié sont rapides et très simples à utiliser. En effet, dans ce type de filtre, généralement, on compare uniquement le vecteur classe avec chaque vecteur d'attributs afin d'obtenir un score selon la méthode d'évaluation choisie. Le seul filtre qui a mis un peu plus de temps d'exécution est le filtre *SVM*. Le nombre d'attributs optimal ne peut pas être déterminé systématiquement, il dépend généralement :
 - Du classifieur ;
 - Du filtre utilisé ;
 - De la base de données utilisée (Nous avons étudié dans [72] la sélection d'attributs en utilisant la base de donnée « *Leukemia* » avec le filtre *Chi2*. Nous avons remarqué que plus le nombre d'attributs diminue, plus le taux de classification augmente. Le meilleur taux a été obtenu en utilisant un seul attribut . Notant que nous avons utilisé un arbre de décision de type *C4.5* pour la classification).
- ⦿ La comparaison des performances des classifieurs d'ordre général en variant le nombre d'attributs, résumé dans la Figure II.3. Nous remarquons que les résultats sont très proches mais il convient à noter que le filtre *SVM* fournit, relativement, de meilleures performances par rapport aux autres filtres.

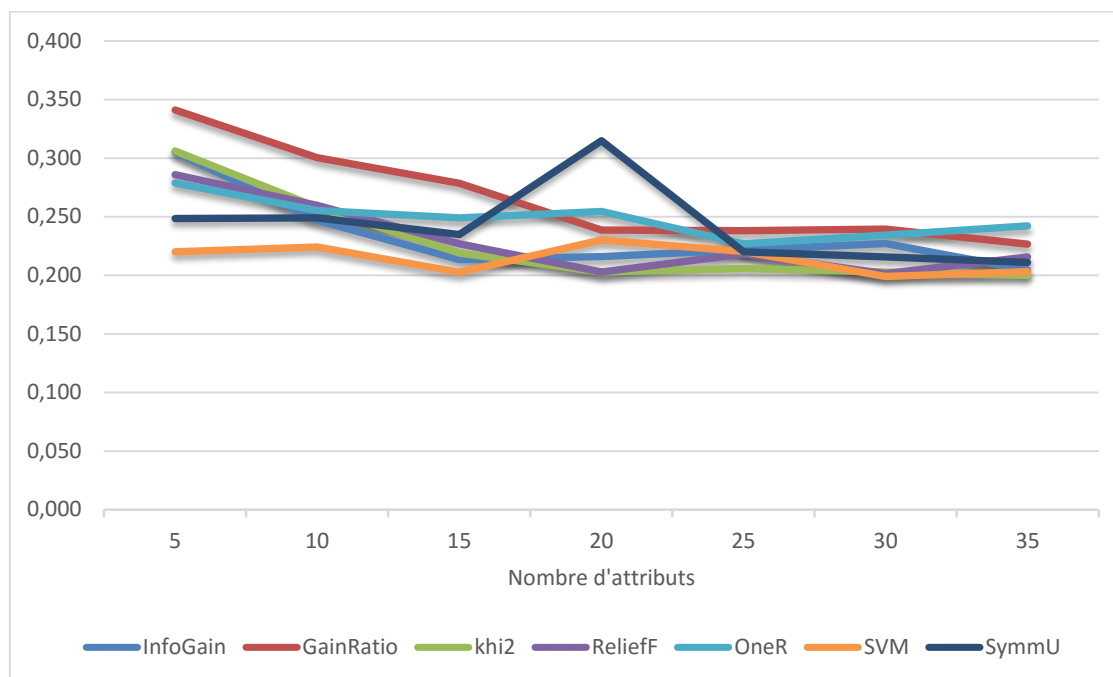


Figure II.3 Moyenne d'erreur en variant le nombre d'attributs

- ⊙ L'approche univarié a l'avantage d'être très rapide avec la plus faible complexité algorithmique par rapport à n'importe quel sélecteur de variable. Cependant, cette approche ne prend pas en considération l'interaction entre attributs. En effet, bien qu'elle sélectionne les attributs les plus pertinents, elle néglige les attributs redondants qui peuvent être source de bruit.
- ⊙ La deuxième catégorie concerne l'approche Multivarié. Nous avons appliqué deux évaluateurs (*CFS* et consistance) avec différents algorithmes de recherches. Dans ce type de méthode, le nombre d'attributs est déterminé automatiquement selon l'algorithme de recherche. En observant la Figure II.4, nous remarquons que la base obtenue grâce à l'évaluateur *CFS* et l'algorithme *BestFirst* a permis d'aboutir au meilleur résultat pour le plus de classifieurs.

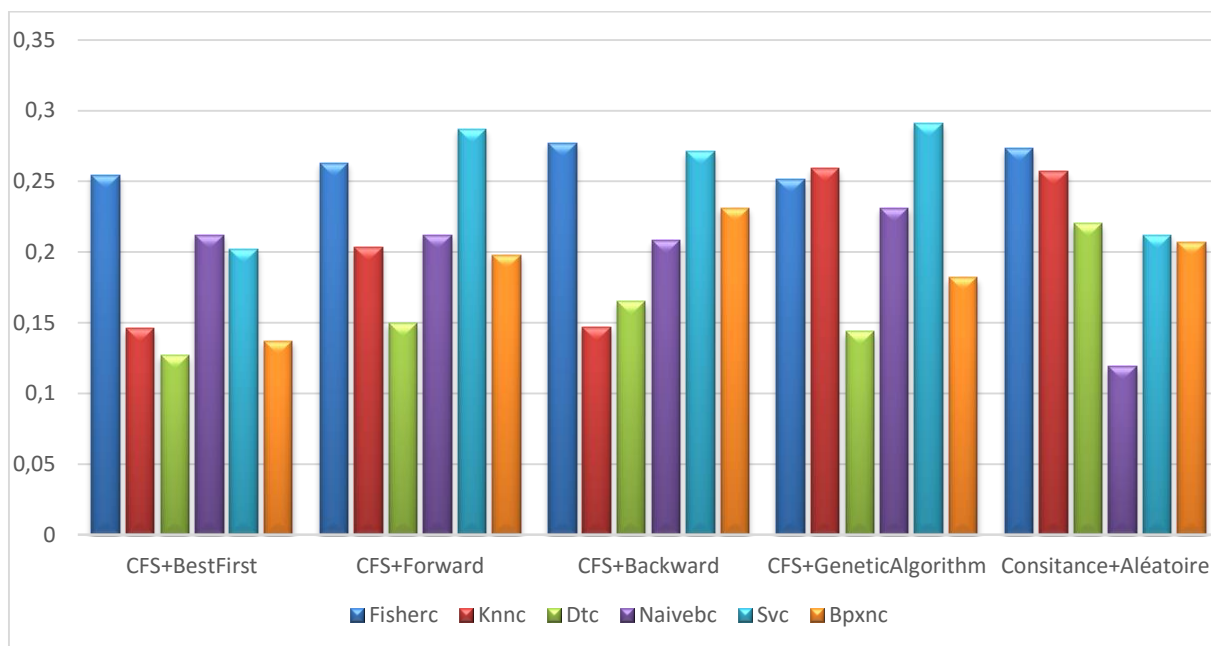


Figure II.4 Comparaison entre filtre avec l'approche multivarié

Afin de choisir le filtre qui sera utilisé pour la suite de cette thèse, nous avons comparé les meilleurs filtres pour chaque catégorie, à savoir : SVM (catégorie univarié) et CFS+BestFirst (catégorie multivarié).

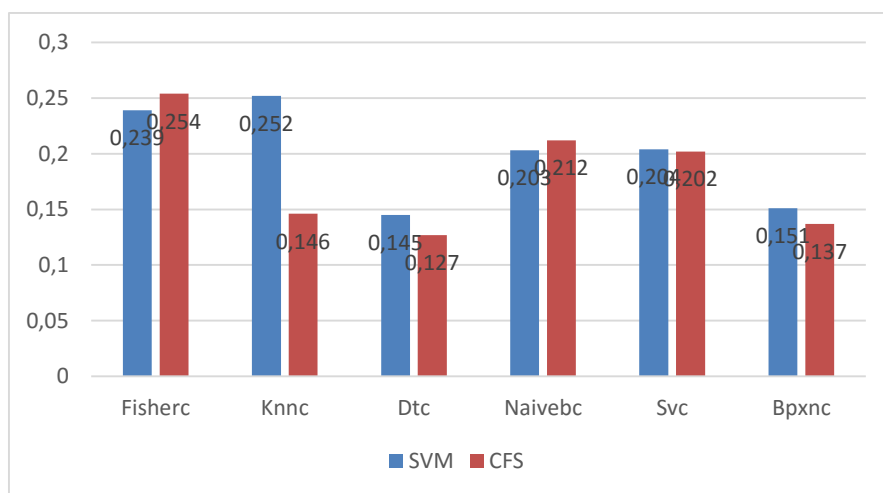


Figure II.5 Comparaison de l'erreur entre CFS+BF et SVM

En premier lieu, nous avons récapitulé la comparaison des performances selon le taux d'erreur dans la Figure II.5. Nous remarquons que, excepté pour le classifieurs *Knnc*, il n'y

a pas une très grande différence d'erreur entre les deux filtres. Cependant, peut-on stipuler que cette petite différence est significative ? Afin de répondre à cette question, nous avons utilisé le test de comparaison appelé « *McNemar* ».

Le test de McNemar est un test statistique utilisé sur les données nominales appariées. Il est appliqué sur un tableau de contingence 2×2 (Tableau II-22). Son nom provient de du nom du chercheur Quinn McNemar, qui l'a introduit en 1947 [73]. L'hypothèse nulle (H_0) d'homogénéité marginale indique que les deux probabilités marginales pour chaque résultat sont les mêmes.

Tableau II-22 Tableau de contingence 2x2 pour le test *McNemar*

| | <i>CFS+BestFirst Correct (1)</i> | <i>CFS+BestFirst Erreur (0)</i> |
|-------------------------------|----------------------------------|---------------------------------|
| <i>Filtre SVM Correct (1)</i> | N_{11} | N_{10} |
| <i>Filtre SVM Erreur (0)</i> | N_{01} | N_{00} |

Où :

- N_{11} nombre d'exemples bien classés par le filtre SVM et le filtre CFS+BestFirst ;
- N_{01} nombre d'exemples mal classés par le filtre SVM mais bien classés par le filtre SVM ;
- N_{10} nombre d'exemples bien classés par le filtre SVM mais mal classés par le filtre SVM ;
- N_{00} nombre d'exemples mal classés en utilisant et le filtre SVM et le filtre CFS+BestFirst.

Le test *McNemar* consiste à calculer χ^2 avec '1' degré de liberté selon la modification de Nemar (équation 11). Sachant que l'hypothèse nulle est rejetée si $\chi^2 > 3,84159$.

$$\chi^2 = \frac{(|N_{01} - N_{10}| - 1)^2}{N_{01} + N_{10}} \quad (11)$$

Afin de comparer entre les filtres, nous avons calculé l'hypothèse nulle du même classifieur en utilisant deux distributions de donnée différentes (générées selon les deux filtres). Les résultats obtenus pour chaque classifieur sont représentés par le Tableau II-23.

Tableau II-23 Comparaison statistique entre les meilleurs filtres

| <i>Classifieur</i> | <i>Valeur χ^2</i> |
|--------------------|-----------------------------------|
| <i>Fisher</i> | 22,82 |
| <i>Knn</i> | 52,81 |
| <i>Dtc</i> | 46,51 |
| <i>Naivebc</i> | 25,68 |
| <i>Svc</i> | 34,26 |
| <i>Bpxnc</i> | 34,22 |

Nous remarquons que l'hypothèse nulle de chaque classifieur est rejetée. Autrement dit les résultats obtenus par chaque classifieur en utilisant les deux filtres sont différents. Dans ce cas, nous affirmons que la majorité des classifieurs ont une meilleure performance en utilisant le filtre CFS+BestFirst par rapport au filtre SVM.

Après cette étude expérimentale et comparative, nous avons retenu le filtre CFS et l'algorithme de recherche BestFirst. Ce choix est justifié non seulement par les performances obtenues, mais aussi pour le fait que le sous-ensemble d'attributs soit de faible taille, d'autant plus qu'il est défini automatiquement selon l'algorithme de recherche choisi.

2.6. CONCLUSION

La sélection d'attributs a toujours suscité beaucoup d'intérêt dans la littérature et en particulier dans le domaine médical. Dans ce chapitre, nous avons réalisé une étude expérimentale comparative en s'intéressant particulièrement aux méthodes de type « filter » pour leur totale indépendance du classifieur utilisé. Il a été remarqué que les méthodes de type multivarié ont plus d'avantage concernant la considération des interactions entre les attributs. A l'issue de cette étude, nous sommes arrivés à la conclusion que ces résultats peuvent être améliorés si l'on optimise la méthode de sélection d'attributs et/ou les méthodes de classification. Ce dernier point fait l'objet de l'une des contributions effectuées dans cette thèse qui sera présenté dans le chapitre 4.

Chapitre III.

VERS UNE CLASSIFICATION ENSEMBLISTE

| | | |
|--------|---|----|
| 3.1. | Introduction | 60 |
| 3.2. | Terminologies et taxonomies | 62 |
| 3.2.1. | Classifieur de base | 62 |
| 3.2.2. | Définitions et motivations | 63 |
| 3.2.3. | Structures des classifieurs ensembliste..... | 65 |
| 3.3. | Etat de l'art des approches ensemblistes | 72 |
| 3.3.1. | Ensemble de classifieurs homogènes | 72 |
| 3.3.2. | Ensemble de classifieurs hétérogènes | 74 |
| 3.4. | Critères de sélection des classifieurs de base | 76 |
| 3.4.1. | Algorithmes de recherches | 77 |
| 3.4.2. | Le rôle de la diversité pour la sélection des classifieurs..... | 78 |
| 3.4.3. | Quantification de la diversité..... | 79 |
| 3.5. | Conclusion..... | 82 |

3.1. INTRODUCTION

La fusion d'information est apparue initialement dans le domaine militaire dont l'objectif était de gérer d'importantes quantités de données en provenance de sources multiples [74]–[76]. Depuis, ce principe a été étendu à d'autres domaines à vocation civile donnant ainsi naissance à plusieurs applications [15], [77], [78].

Deux faits importants ont contribué à la dynamique énorme de la notion de fusion de données que nous observons ces dernières années [79]. D'un côté, l'arrivée des systèmes de détection multimodaux qui fournissent des données coordonnées de plusieurs capteurs, de différents types, appuyés par l'immense contenu de l'information dans les systèmes d'information interconnectés [76]. D'un autre côté, les méthodes de classification individuelles fournissent des connaissances de remplacement, même en l'absence des données. Il s'est avéré que même si elle est appliquée à la même tâche en utilisant les mêmes données, une décision conjointement émise par des classifieurs combinés est potentiellement plus efficace que n'importe quelle décision individuelle [80]–[82].

La fusion d'informations peut être définie comme la combinaison d'informations, souvent imparfaites et hétérogènes, afin d'obtenir une information globale utile, plus complète, de meilleure qualité, et permettant d'améliorer la prise de décision [83]. Elle peut être réalisée sur différents niveaux d'abstraction qui sont étroitement liés: fusion au niveau des données, fusion au niveau des attributs et fusion des décisions [80]. La Figure III.1 montre le processus général de fusion au niveau des données.

Wald [84] a défini **la fusion des données** comme étant le cadre formel dans lequel s'expriment les moyens et les techniques permettant l'alliance de données provenant de sources diverses. C'est un processus visant l'obtention d'information de meilleure qualité. Par exemple, une restitution cartographique satellite peut être le résultat de la fusion de données optiques, de données image infrarouge et de données image radar.

La fusion au niveau des attributs permet à la fois de fournir les informations les plus discriminatoires, originaires de multiples ensembles d'attributs, et d'éliminer les informations redondantes qui résultent de la corrélation entre les ensembles d'attributs distinctes. En d'autres termes, la fonction de fusion est capable de dériver et de gagner des ensembles de vecteurs de caractéristiques les plus efficaces et les moins dimensionnelles qui profitent à la décision finale [85].

La fusion au niveau des décisions, ou fusion de haut niveau, permet de synthétiser plusieurs décisions dans le cadre de la classification. Dans ce type de fusion différents classifieurs sont combinés afin de choisir la meilleure décision pour un problème de classification donné. Ce type de fusion est le plus utilisé dans la littérature.

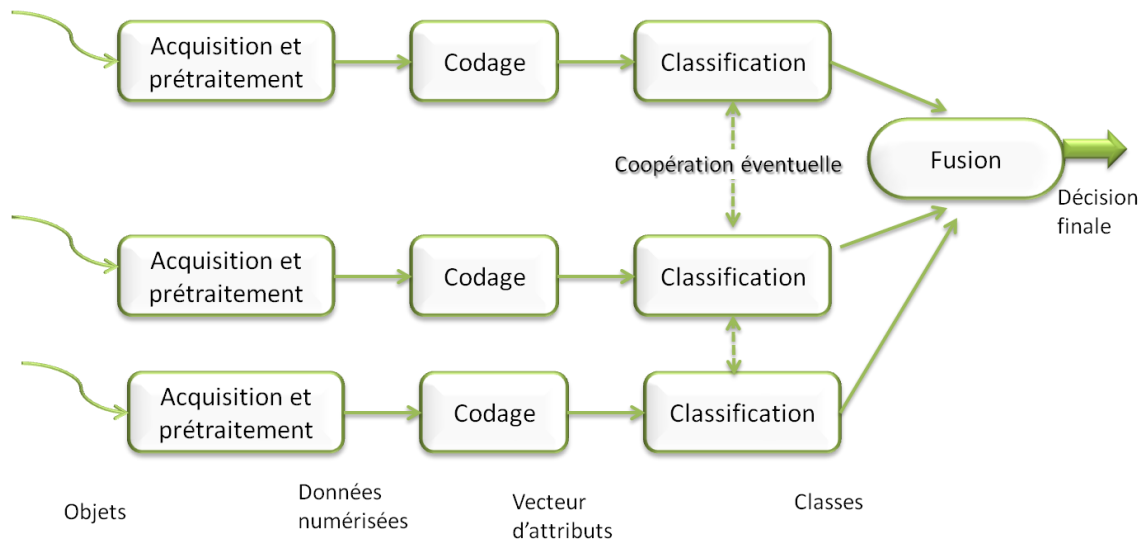


Figure III.1 Processus général de fusion des données

Dans ce chapitre, afin d'atteindre notre objectif en améliorant l'exactitude du système, nous nous intéressons particulièrement à la fusion au niveau des décisions, du fait que nous avons une seule source de données avec les mêmes caractéristiques pour tous les échantillons. Nous débutons ce chapitre par définir quelques notions de base dans le domaine. Ensuite, nous présentons une variété d'approches de combinaisons de classifieurs proposées dans la littérature. Et nous terminons par le critère de sélection des membres parmi un ensemble de classifieurs ainsi qu'une conclusion du chapitre.

3.2. TERMINOLOGIES ET TAXONOMIES

Les nombreux ateliers internationaux sur les systèmes multi-classifieurs (MCS), tenus depuis l'an 2000 (*Multiple Classifier Systems, International Workshop*) [86], ont joué un rôle central dans l'organisation, la systématisation et le développement davantage de connaissances dans le domaine de la combinaison des classifieurs [87]. Par ailleurs, Cela a abouti à des termes différents pour la même notion (notons que certaines notions ne sont pas absolument identiques). Par exemple :

- Système multi-classifieurs (MCS) = Ensemble de classifieurs (EoC) = Classifieurs ensemblistes = Combinaison des classifieurs = Fusion au niveau de décision = Fusion de haut niveau = méta-classifieurs = Comité de décideurs
- Classifieur = classifieur de base = hypothèse = apprenant = expert
- Sortie = décision = classe
- Exemple = instance = cas = point donné = objet

3.2.1. CLASSIFIEUR DE BASE

Ce terme est utilisé pour indiquer les composants de base d'un système multi-classifieurs (MCS). Autrement dit, un MCS est constitué d'un ensemble de classifieurs de base [87]. Les classifieurs de base doivent être différents, mais ils doivent aussi être comparables pour que l'on puisse combiner leurs sorties [88]. Un ensemble cohérent de différents classifieurs peut être généré sous l'environnement suivants:

1. Choix de paramètres différents (tels que le nombre de voisins dans la règle des k-plus proches voisins, le noyau dans SVM, ...etc) ;
2. Différentes architectures (telles que la taille d'un réseau de neurones) ;
3. Classifieurs de nature différente ou hétérogène (tels que : réseau de neurones, SVM et arbre de décision) ;
4. Initialisations différentes ;
5. Classifieurs entraînés sur des ensembles d'apprentissage différents ;

6. Ensembles d'attributs différents (tels que la génération des sous-ensembles aléatoires sur un grand ensemble d'attributs).

La taille d'un ensemble de classifieurs de base détermine la dimensionnalité d'entrée pour le combinateur (le nombre de classes multiplié par le nombre de classifieurs de base). Il est à noter que certains auteurs utilisent ce terme uniquement lorsque le système de classification multiple est conçu à l'aide d'un modèle unique de classification (par exemple, un arbre de décision). Plusieurs versions de ce classifieur de base étant générées pour construire le système de classification.

3.2.2. DEFINITIONS ET MOTIVATIONS

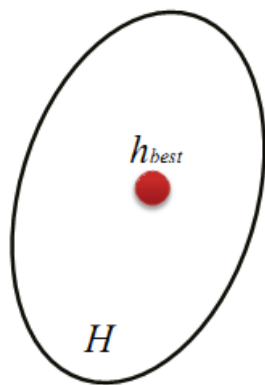
Selon l'encyclopédie de l'apprentissage automatique [3], il n'existe pas un modèle d'apprentissage capable de traiter n'importe quelle distribution de donnée. En effet, l'approche classique pour concevoir un système de reconnaissance qui se concentre sur la recherche du meilleur classifieur individuel présente quelques inconvénients sérieux [89].

Dietterich [90] propose trois qualités pour lesquelles un ensemble de classifieurs peut être meilleur qu'un classifieur individuel : statistique, computationnelle et représentationnelle. Les améliorations dans les tâches de classification sont souvent obtenues en agréant un groupe de classifieurs (appelés classifieurs de base) en tant que comité d'ensemble et en rendant la prédiction des données du test de manière consensuelle. L'objectif de la conception et de l'utilisation des méthodes ensemblistes est de parvenir à une classification plus précise ainsi qu'à une meilleure généralisation [91].

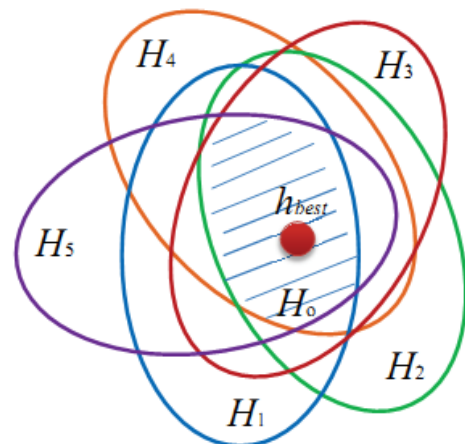
Supposons que les données d'apprentissage ne sont pas bruitées, sans valeurs manquantes, et suffisamment nombreuses pour représenter le modèle dans la Figure III.2. Ainsi, nous souhaitons que la classe choisie sur l'ensemble des données représente la meilleure hypothèse (règle de classification appelée h_{best}) de classification indiquée par le cercle dans la Figure III.2. Cependant, en pratique, les ensembles de données d'apprentissage sont souvent composés d'une taille d'échantillons réduite, d'une grande dimensionnalité et d'un rapport de bruit élevé. Par conséquent, l'obtention de la meilleure hypothèse de classification n'est souvent pas commune. En effet, il existe un grand

nombre d'hypothèses sous-optimales dans l'espace de l'hypothèse H qui peuvent correspondre aux données d'apprentissage mais qui ne généralisent pas bien sur des données futures.

La conception de classifieurs multiples en manipulant les données d'apprentissage de manière intelligente permet d'obtenir un espace d'hypothèse différent avec chaque classifieur (H_1, H_2, \dots, H_L , où L est le nombre de classifieurs), ce qui peut conduire à un espace d'hypothèses de chevauchement rétréci (H_0) en combinant les sorties des classifieurs de base. Ceci est réalisé à l'aide des méthodes d'intégration qui profitent de la région chevauchante (comme par exemple la moyenne ou le vote majoritaire). Par conséquence, l'ensemble composé d'une telle manière semble souvent plus précis.



(a) Espace d'hypothèse d'un seul classifieur



(b) Espace d'hypothèse d'un ensemble de classifieurs

Figure III.2 Illustration de l'espace partitionnement de la meilleure hypothèse avec un ensemble de classifieurs

Un grand nombre de méthodes pour créer et combiner plusieurs classifieurs ont été proposées dans la littérature au cours des dernières années. Les résultats rapportés ont démontré les bonnes performances réalisables par la combinaison de plusieurs classifieurs.

3.2.3. STRUCTURES DES CLASSIFIEURS ENSEMBLISTE

Il n'existe toujours pas un accord universel sur une structure générale ou une taxonomie de l'ensemble du domaine. Cependant, plusieurs architectures peuvent être envisagées au niveau de la combinaison:

Selon l'agencement des classifieurs il existe trois types de combinaison [23], parallèle, séquentielle et hybride.

- La **combinaison séquentielle** (en cascade), est organisée en niveaux successifs de décision. L'objectif étant de réduire progressivement le nombre de classes possibles. Pour chaque niveau, il existe un seul classifieur qui prend en compte la réponse fournie par le classifieur placé en amont afin de traiter les rejets ou de confirmer la décision obtenue sur la classe qui lui est présentée.
- Inversement, la **combinaison parallèle** laisse dans un premier temps les différents classifieurs opérés indépendamment les uns des autres. Ensuite, les décisions individuelles sont fusionnées. Deux choix de fusions sont possibles : une fusion démocratique qui ne favorise aucun classifieur en particulier et une fusion dirigée qui consiste à attribuer un poids à la sortie de chaque classifieur en fonction de ses performances.
- La **combinaison hybride** est une approche basée sur la combinaison d'architectures séquentielle et parallèle afin de tirer pleinement avantage de chacun des classifieurs utilisés. Ce type d'approche permet de générer de nombreux schémas de coopération qui peuvent rapidement devenir complexes à optimiser.

Selon le type de sorties des classifieurs, il existe trois type de combinaisons [92] :

- **Type classe** : les sorties de type classe sont les moins riches en information et les plus simples à utiliser. Son principal avantage est qu'elle peut être utilisée pour tout type de sortie de classifieur, quelle que soit sa structure.
- **Type rang** : les méthodes de type rang combinent des listes de propositions de classifieurs ordonnées de manière décroissante selon leur préférence. Les rangs

attribués par chaque classifieur peuvent être considérés comme étant des votes pour toutes les classes. Ayant un nombre N de classes, chaque classifieur attribue un poids N à la classe placée au début de la liste, un poids $N-1$ à la seconde classe et ainsi de suite. La classe placée à la fin de la liste reçoit un seul vote.

- **Type mesure** : les méthodes de ce type combinent des classifieurs en utilisant leur degré de confiance attribué à chaque classe. Cependant, étant donné que les sorties des classifieurs ne sont pas toujours comparables, une normalisation est souvent nécessaire.

Selon le type de coopération des classifieurs, Kuncheva [93] a montré la différence entre la fusion et la sélection :

- La **fusion** ou agrégation est une opération qui permet de combiner les sorties de tous les classifieurs dans l'objectif d'atteindre un consensus. Dans ce cas, tous les classifieurs participent de manière indépendante à la prise de décision sur le même problème. Il est à noter que toutes les sorties sont combinées par un certain opérateur de combinaison.
- la **sélection** est une opération qui consiste à prendre un choix dynamique des "meilleurs" classifieurs parmi un ensemble de classifieurs possibles afin d'identifier la classe inconnue.

Selon leur capacité d'apprentissage, on peut distinguer [93] :

- Les **méthodes paramétriques** (sans apprentissage) se contentent d'utiliser seulement et simplement des informations du premier ordre (sorties de classifieurs). Elles sont faciles à implémenter et ne nécessitent pas de phase d'apprentissage. Toutefois, l'inconvénient de ces méthodes réside dans le fait qu'elles traitent les classifieurs de la même manière, ce qui ne permet pas prendre en considération leur capacité individuelle. Elles peuvent être divisées en trois types de sortie de classifieur: classe, rang et mesure.
- Les **méthodes non-paramétriques** (avec apprentissage) permettent de chercher et adapter les paramètres à utiliser dans la combinaison suivant la base des

exemples disponibles durant la phase d'apprentissage. Comparées aux méthodes de combinaison paramétriques, ces méthodes sont plus complexes à mettre en oeuvre. Elles utilisent des paramètres supplémentaires calculés pendant la phase d'apprentissage. La performance de ces méthodes dépend alors de la bonne estimation de ces paramètres.

Selon le fait de générer de nouveaux classifieurs de base, il existe deux méthodes [94] :

- **les méthodes d'ensemble non génératives** : ces méthodes se limitent à combiner de manière appropriée un ensemble de classifieurs de base existant (éventuellement bien conçus). Dans ce cas, l'accent est mis sur les différentes techniques de combinaison ainsi que sur le choix de l'ensemble de classifieurs à combiner.
- **les méthodes d'ensemble génératives** : contrairement aux méthodes non génératives, ces méthodes génèrent de nouveaux classifieurs de base qui agissent sur l'algorithme d'apprentissage de base ou sur la structure de l'ensemble de données afin d'essayer d'améliorer activement la diversité et l'exactitude des classifieurs de base. Dans ce cas, l'accent est mis sur la façon dont divers apprenants de base sont construits.

Plusieurs structures globales ont été proposées dans la littérature. Dans le travail de [23], les auteurs ont présenté une structure globale sur les méthodes de combinaison parallèle résumée dans la Figure III.3. D'un autre côté, Matteo et Giorgio [94] ont proposé la structure représentée dans la Figure III.4.

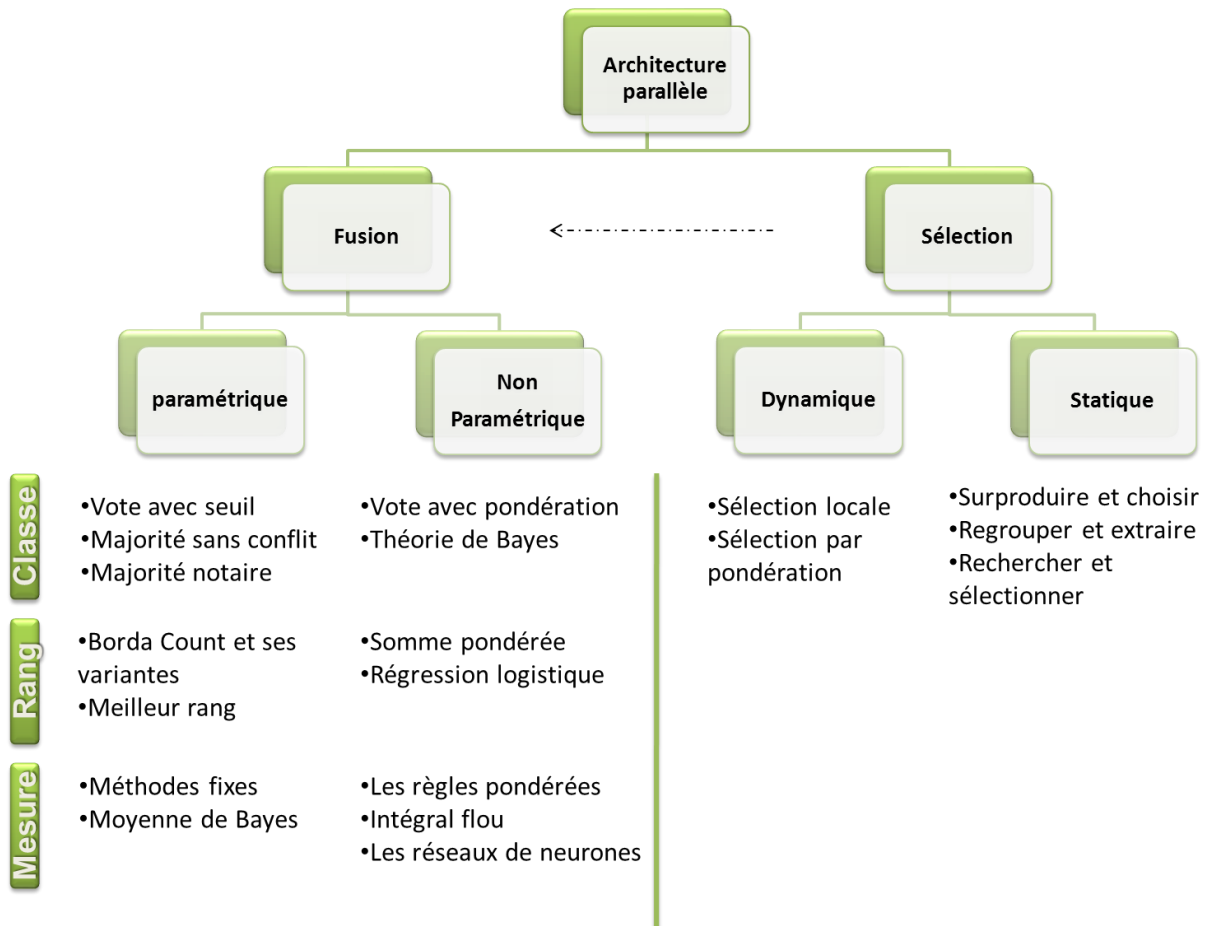


Figure III.3 Panorama global des méthodes de combinaison parallèle des classifieurs selon les travaux de Zouari et al. [23]

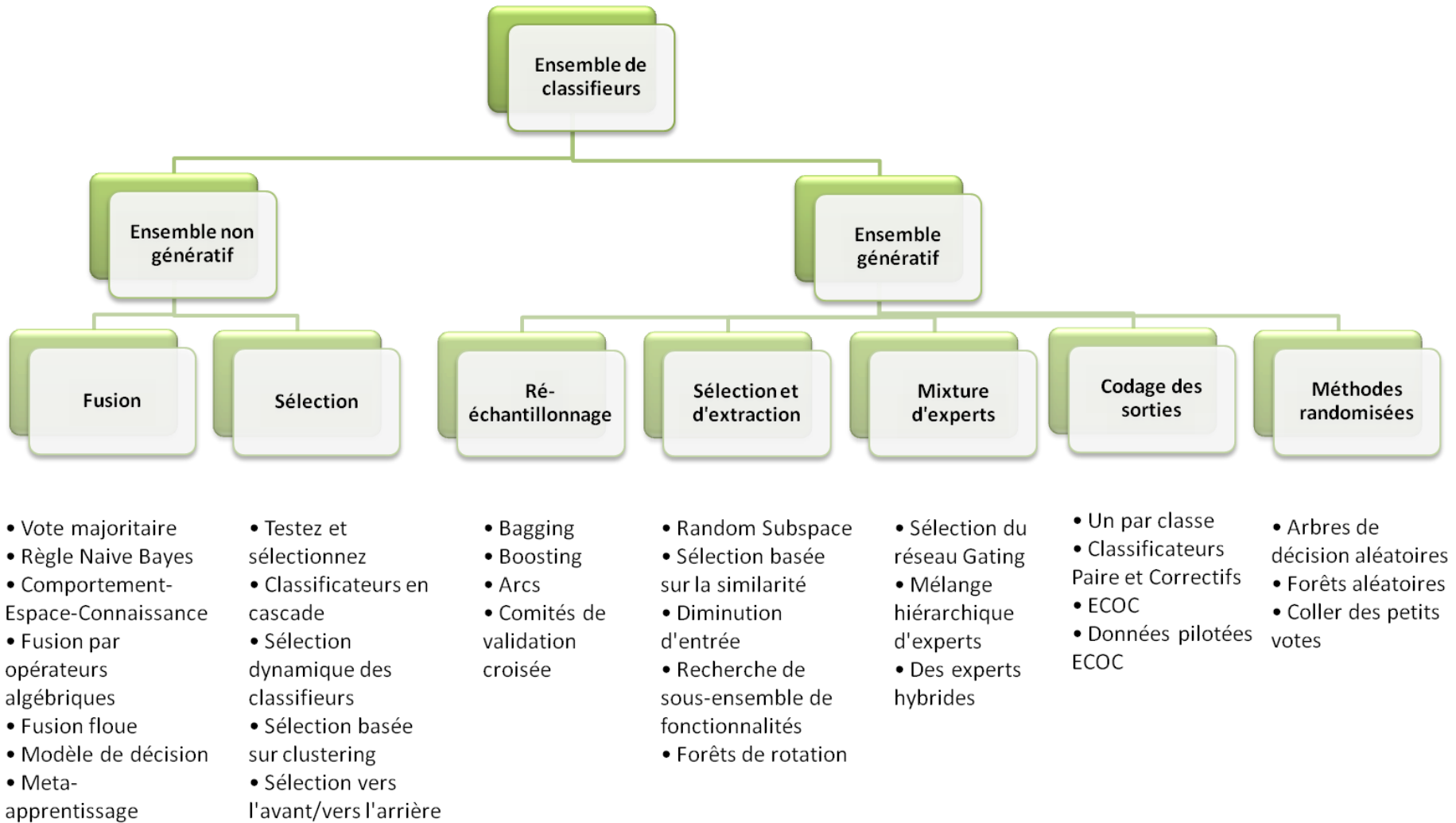


Figure III.4 Panorama global des méthodes de combinaison des classifieurs selon Matteo et Giorgio [94]

La combinaison d'un ensemble de classifieurs est souvent basée sur le principe du vote, en particulier de type classe. Ce principe a été formalisé par le mathématicien et philosophe *Condorcet* il y a maintenant plus de deux siècles [95]. Depuis, plusieurs méthodes de vote ont été proposées. Elles sont utilisées essentiellement dans les élections.

Dans le cadre de la combinaison, les méthodes de vote consistent à interpréter chaque sortie d'un classifieur comme un vote pour l'une des classes possibles. La classe ayant un nombre de votes supérieur à un seuil préfixé est retenue comme décision finale. On peut classer ces méthodes en plusieurs catégories :

Vote avec seuil : Les méthodes de vote peuvent pratiquement toutes être dérivées de la règle avec seuil exprimée par :

$$E(x) = \begin{cases} C_i & \text{Si } \sum_{j=1}^L e_{i,j} = \max_{t=1}^N \sum_{j=1}^L e_{t,j} \geq \lambda \cdot L \\ \text{Rejet} & \text{Sinon} \end{cases} \quad (12)$$

Où pour chaque objet x :

- L est le nombre de classifieurs
- C_i est la i -ème classe retenue par le vote attribué par le j -ème classifieur e_j à x .
- λ correspond à la proportion de classifieurs devant répondre la même classe pour que cette classe soit retenue comme résultat de la combinaison. En fonction de la valeur que peut prendre λ , les votes suivants sont distingués :

Pour $\lambda = 0$, il s'agit du vote à la pluralité :

- > La classe qui reçoit le plus de votes est élue comme classe finale ;
- > Si toutes les classes ont le même nombre de votes, il y a rejet.

Pour $\lambda = 0,5$, il s'agit du vote à la majorité

- > La classe qui reçoit plus de la moitié des votes est élue ;

- > Probabilité de produire une décision finale erronée quand la majorité des classifieurs ne proposent pas la vraie classe. Ceci implique qu'il faut un grand nombre de classifieurs pour que la solution soit moyennement correcte.

Pour $\lambda = 1$, il s'agit du vote à l'unanimité

- > La classe finale est élue si tous les classifieurs proposent cette réponse. Autrement, un rejet est automatique ;
- > Ce vote est plutôt fiable. Cependant, il présente l'inconvénient majeur de produire un taux de reconnaissance assez faible, d'autant plus lorsque le nombre de classifieurs augmente.

La Figure III.5 représente les différents modèles de consensus dans un groupe de 10 décideurs: l'unanimité, la majorité simple, et la pluralité. Dans les trois cas, la décision finale du groupe est représentée par les décideurs portant la couleur noire.



Figure III.5 Modèles de consensus (unanimité, majorité et pluralité)

Majorité sans conflit : Le principe de la majorité sans conflit est identique à celui de la majorité unanime présenté ci-dessus, à la différence que l'on autorise les classifieurs à effectuer un rejet. Un classifieur proposant de rejeter un élément n'a donc aucun poids dans ce système. Dans ce cas, lorsque tous les classifieurs rejettent sauf un, alors c'est la sortie de ce dernier classifieur qui sera conservée comme résultat de la combinaison. Il n'y a rejet total que lorsque tous les classifieurs ont proposé de rejeter l'élément ou en cas d'égalité de votes.

Vote avec pondération : Dans le vote avec pondération, la réponse de chaque classifieur e_j est pondérée par un coefficient ω_j indiquant son importance dans la combinaison.

$$E(x) = \begin{cases} C_i & \text{Si } \sum_{j=1}^L \omega_j e_{i,j} = \max_{t=1}^N \sum_{j=1}^L \omega_j e_{t,j} \\ \text{Rejet} & \text{Sinon} \end{cases} \quad (13)$$

Il existe plusieurs façons de déterminer les coefficients ω_j , ils peuvent par exemple avoir été optimisés par algorithme génétique [96] ou selon la fiabilité de chaque expert en se basant sur leurs performances individuelles [97].

3.3. ETAT DE L'ART DES APPROCHES ENSEMBLISTES

Malgré les nombreuses approches ensemblistes proposées dans la littérature pour la conception d'un ensemble de classifieurs, mais jusqu'à présent aucune méthode n'a donné entière satisfaction, et ce domaine de recherche reste encore un problème ouvert [98]. Dans cette sous-section, nous présentons les principaux travaux réalisés dans ce domaine selon la nature des classifieurs de base, qu'ils soient homogènes ou hétérogènes [99]–[101].

3.3.1. ENSEMBLE DE CLASSIFIEURS HOMOGENES

Les mécanismes qui sont utilisés pour construire un ensemble de classifieurs homogènes se basent soit sur l'emploi de différents sous-ensembles de données d'apprentissage en utilisant une méthode de classification unique, ou bien sur l'utilisation de différents paramètres d'apprentissage avec une méthode de classification unique [100]. Cet axe de recherche a suscité beaucoup d'intérêt dans la littérature :

Bagging (Bootstrap aggregating) est une approche de construction d'ensemble qui utilise différents sous-ensembles de données d'apprentissage avec une méthode de classification unique [102]. Étant donné un ensemble d'apprentissage de taille t , *Bagging* attire des instances aléatoires t de l'ensemble de données avec remplacement (à l'aide d'une distribution uniforme). Ces t instances sont apprises, et ce processus est répété plusieurs fois. Étant donné que le tirage au sort est effectué avec remplacement, les instances tirées contiendront des doublons et des omissions par rapport à l'ensemble d'apprentissage

initiale. Chaque cycle à travers le processus aboutit à un classifieur. Après la construction de plusieurs classifieurs, les sorties de chaque classifieur sont combinés pour produire la prédiction finale [102].

Une autre approche appelée « **Boosting** » utilise également une méthode d'apprentissage unique avec différents sous-ensembles de données d'apprentissage [103]. Sa structure globale est similaire à celle de la méthode *Bagging*, à la différence qu'elle conserve la trace de la performance de l'algorithme d'apprentissage et se concentre sur les cas qui ne sont pas correctement appris. Au lieu de choisir les instances t d'apprentissage à l'aide d'une distribution uniforme de manière aléatoire, les exemples d'apprentissage sont sélectionnés en favorisant les instances qui ne sont pas bien classées. Après plusieurs cycles, la prédiction est réalisée selon un vote pondéré des prédictions de chaque classifieur. Ainsi, les poids sont proportionnels à la précision de chaque classifieur sur son ensemble d'apprentissage. L'algorithme le plus connu de l'approche *Boosting*, appelée « *AdaBoost* », a été proposée par Friedman et al. [104].

Les *forêts aléatoires* (plus connus sous *Random Forest*) sont une combinaison d'arbres de décision, où chaque arbre dépend des valeurs d'un vecteur aléatoire indépendamment échantillonné et avec la même distribution pour tous les arbres de la forêt [105]. L'erreur de généralisation d'une forêt d'arbres dépend de la force des arbres individuels dans la forêt et de la corrélation entre eux. L'utilisation d'une sélection aléatoire de caractéristiques pour diviser chaque nœud donne des taux d'erreur qui se comparent favorablement à *Adaboost* [105].

La méthode *Bagging* exploite le principe de la diversité en utilisant les différents ensembles de données perturbées dans la base d'apprentissage des classifieurs. Autrement dit, chaque classifieur de base est entraîné sur un sous-ensemble d'échantillons pour obtenir une hypothèse de classification légèrement différente, puis il est combiné avec les autres classifieurs pour former l'ensemble. D'un autre côté, la méthode *Random Forest* utilise le même principe mais avec différents ensembles d'attributs. En ce qui concerne la méthode *boosting*, la diversité est obtenue en augmentant les poids des échantillons mal classés de manière itérative. Généralement, ces trois méthodes utilisent des arbres de décision pour

leur sensibilité aux petits changements sur l'ensemble d'apprentissage, et sont donc adaptées à la procédure de perturbation appliquée aux données d'apprentissage [91].

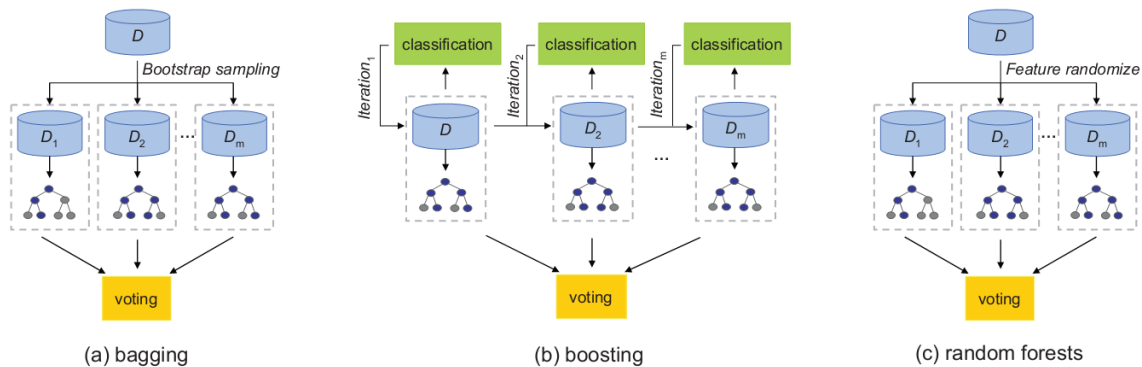


Figure III.6 Illustration schématique des trois méthodes d'ensemble homogènes populaires

Il existe aussi des combinaison de classifieurs homogènes avec différents paramètres d'apprentissage sur le même ensemble de données, comme par exemple dans [106] où des techniques de réseaux de neurones ont été employées à base des méthodes d'apprentissage avec différentes topologies, différents poids initiaux et différents paramètres.

3.3.2. ENSEMBLE DE CLASSIFIEURS HETEROGENES

Ce type d'ensemble permet la combinaison d'hypothèses produites à partir de méthodes d'apprentissage différentes (des hypothèses de nature hétérogène), appliquées à un même échantillon d'apprentissage. Les travaux de la littérature dans ce domaine portent soit sur la manière de combiner les classifieurs efficacement, soit sur la sélection des classifieurs.

Le vote est la méthode la plus simple pour combiner les prédictions à partir de classifieurs multiples [107]. Dans sa forme la plus simple, appelée pluralité ou vote majoritaire, chaque modèle de classification contribue à un seul vote. La prédiction collective est réalisée par la majorité des votes, c'est-à-dire que la classe ayant le plus de votes est la prédiction finale. En revanche, dans le vote pondéré, les classifieurs ont des degrés d'influence variables sur la prévision collective qui est relative à leur précision prédictive. Chaque classifieur est lié à un poids spécifique déterminé par ses performances (par exemple, précision, modèle de coût) sur un ensemble de validation. La prévision finale est décidée

en résumant tous les votes pondérés et en choisissant la classe avec l'agrégat le plus élevé [100].

Parmi les méthodes qui permettent la combinaison des classifieurs hétérogènes, nous trouvons la méthode « *Stacking* » [108]. Cette méthode vise à améliorer l'efficacité en exécutant un certain nombre de processus d'apprentissage et en combinant les résultats collectifs. La principale différence entre la méthode du vote et la méthode Stacking est que ce dernier combine les classifieurs de base d'une manière non linéaire. La tâche de combinaison intègre les classifieurs de base calculés indépendamment dans un classifieur de niveau supérieur, un méta-classifieur, en réapprenant l'ensemble d'apprentissage au niveau supérieur (méta-niveau). Cet ensemble est créé en utilisant les prédictions des classifieurs de base sur l'ensemble de validation comme valeurs d'attribut et la classe désirée comme cible. Les auteurs ont montré que la généralisation réussie nécessite l'utilisation de distributions de classes de sortie plutôt que de prédictions de classe. Dans leurs expérimentations, seul l'algorithme MLR (un discriminant linéaire) était approprié pour être utilisé comme un classifieur de niveau 1 [108].

La généralisation en cascade est un autre algorithme qui appartient à la famille des algorithmes Stacking. Cet algorithme utilise l'ensemble des classifieurs séquentiellement, à chaque étape exécutant une extension des données d'origine par l'insertion de nouveaux attributs [109]. Les nouveaux attributs sont dérivés de la distribution de classes de probabilité donnée par un classifieur de base. Dans [110] les auteurs ont proposé d'étendre la méthode Stacking avec des distributions de probabilité et une régression linéaire multi-réponses. Ils ont montré que cette dernière extension est plus performante que les méthodes existantes et mieux que de sélectionner le meilleur classifieur par validation croisée.

D'un autre côté Kuncheva a donné des formules pour l'erreur de classification en utilisant les règles fixes suivantes: moyenne, minimum, maximum, médian, vote majoritaire et produit [111]. Pour une probabilité a posteriori uniformément répartie, la méthode minimale/maximale ont fait le meilleur taux; Tandis que pour les erreurs normalement réparties, les règles de fusion ont donné globalement une performance très similaire. Il existe d'avantage d'architectures qui permettent de fusionner des classifieurs hétérogènes.

Parallèlement, des recherches théoriques et expérimentales ont montré que la sélection de classifieurs est un processus primordial avant d'utiliser n'importe quelle architecture.

Mendiàdua et al. [112] ont présenté et testé la sélection dynamique des classifieurs avec la validation croisée puis ils ont proposé une approche qui a les mêmes stratégies que la sélection dynamique des classifieurs en utilisant la méthode K-plus proche voisins égalité (K-NNE) pour obtenir la précision locale. Notant que K-NNE est une extension de K-NN dans laquelle toutes les classes sont traitées indépendamment [112]. Ils ont réalisé une autre méthode basée sur la sélection des classifieurs simples appropriés par la moyenne d'un algorithme évolutif [113].

Dans [114], les auteurs ont combiné les avantages de la fusion des classifieurs et de la sélection dynamique selon leurs performances individuelles afin de choisir le sous-ensemble qui participe au vote final. Dans le même contexte, Canuto et al. [115] ont montré comment le choix des classifieurs de bases peut affecter la performance de plusieurs méthodes de combinaison (méthodes basées sur la sélection et/ou la fusion). Ils ont conclu que les précisions les plus élevées ont été presque toujours obtenues en utilisant des structures hybrides.

Kuncheva a présenté une combinaison de sélection de classifieurs et de fusion en utilisant l'inférence statistique pour basculer entre les deux [93]. Les auteurs ont étudié les relations entre les différentes méthodes de combinaison de classifieurs et les mesures de diversité dans la combinaison des classifieurs.

3.4. CRITERES DE SELECTION DES CLASSIFIEURS DE BASE

Si nous avons un classifieur parfait, alors aucun ensemble n'est nécessaire [116]. Étant donné un grand nombre de classifieurs différents, il existe un certain nombre de stratégies de combinaison possible à suivre et il n'est généralement pas clair lequel peut être le meilleur pour un problème particulier [117]. La stratégie la plus simple pourrait être de sélectionner un classifieur unique, à savoir le classifieur le plus performant sur les données d'apprentissage et de l'appliquer sur la base de test [118]. Une telle approche, bien qu'elle

est la plus simple, ne garantit pas les performances optimales [119]. En outre, il est possible qu'au moins certains sous-ensembles de classifieurs puissent conjointement surpasser le meilleur classifieur s'ils sont combinés de façon appropriée. Pour assurer une performance optimale, une conception de classifieurs multiples doit être en mesure de sélectionner le sous-ensemble de classifieurs le plus optimal dans le sens où il produit la plus haute performance possible pour un combineur particulier.

3.4.1. ALGORITHMES DE RECHERCHES

Il existe beaucoup d'algorithmes de recherche pour la trouver le meilleur sous ensembles de classifieurs [117], [118]. Parmi les algorithmes de recherches le plus utilisés dans la littérature nous citons :

- **Single best (SB)** : La sélection du classifieur le plus performant est le choix le plus simple et plus largement adopté. Une méthode est très utilisée grâce à sa simplicité et sa faible complexité de recherche. Dans la littérature, les méthodes employant des techniques de combinaison, présentent souvent la méthode **SB** comme une référence de comparaison.
- **N best (NB)** : cette méthode représente une extension de **SB** et dans laquelle plusieurs combinaisons de classifieurs sont évaluées et les N classifieurs, qui présentent la performance de combinaison la plus optimale, sont sélectionnés. L'algorithme **NBest** est très coûteux en termes de complexité calculatoire. En effet, ceci nécessite d'évaluer tous les classifieurs selon leur performance et vérifier par la suite toutes les combinaisons possibles pour trouver la valeur de N .
- **Forward search (FS)** : ou recherche en avant, c'est une méthode intuitive. A chaque itération, un nouveau classifieur, qui minimise la valeur de l'erreur de l'ensemble des classifieurs, est ajouté. Si cette erreur ne peut être réduite pour aucun classifieur, la recherche s'arrête à la dernière itération de combinaison possible. L'erreur est le critère le plus intuitif pour la sélection de classifieurs. En général, toute mesure d'optimalité peut être intégrée comme critère de sélection. A noter qu'à chaque itération, le critère de sélection est évalué pour tous les classifieurs restants ce qui engendre une complexité quadratique.

- **Backward search (BS):** à l'inverse de L'algorithme de FS, Backward search (la recherche en arrière) commence par l'ensemble des classifieurs et élimine à chaque itération le classifieur qui ne minimise pas l'erreur.

3.4.2. LE ROLE DE LA DIVERSITE POUR LA SELECTION DES CLASSIFIEURS

La qualité du système multi-classifieurs repose principalement sur la qualité du critère de sélection. La meilleure performance individuelle a toujours été un indicateur universel pour la sélection des meilleurs classifieurs de base. Cependant l'applicabilité du critère de performance individuel pour la sélection des classifieurs est très limitée [117]. Le problème majeur est l'incohérence de l'évaluation, car l'ajout de classifieurs plus mauvais que les premiers (dans le sens performance individuelle) ne produit pas forcément des combinaisons médiocre.

D'un autre côté, plusieurs chercheurs du domaine sont convaincus que la combinaison de classifieurs ne peut être efficace que si les classifieurs individuels sont différents et pas nécessairement indépendants [120]. Ainsi, l'étude du rôle de la diversité, appelée aussi complémentarité ou orthogonalité, dans l'amélioration des performances d'un ensemble de classifieurs constitue actuellement une piste de recherche prometteuse. Il est prouvé qu'il est préférable de combiner des classifieurs moins performants mais qui font des erreurs différentes, plutôt que des classifieurs très performants mais qui présentent des erreurs identiques [121].

Cependant, la diversité n'est pas suffisante dans un ensemble de classifieurs, il faut tenir compte aussi de la performance des classifieurs à combiner, nous parlons du compromis « diversité-performance ». Kuncheva [122] a montré que lorsque les performances des classifieurs augmentent, la diversité de l'ensemble diminue. Il est alors intéressant d'avoir un bon compromis entre diversité et performance [23]. De nombreux travaux ont abordé de façon expérimentale l'étude de la diversité, la question posée est de connaître si deux classifieurs sont divers ?

3.4.3. QUANTIFICATION DE LA DIVERSITE

Dans la littérature, ces mesures sont classées en deux catégories: les mesures pairwise se calculent pour chaque paire de classifieurs. La diversité d'un ensemble de L classifieurs est obtenue en faisant la moyenne de $\frac{L(L-1)}{2}$ mesures calculées sur les paires de classifieurs. Les mesures non pairwise permettent de calculer la diversité de tous les éléments de l'ensemble de classifieurs.

Pour quantifier les deux types de mesure, nous comparons les sorties des classifieurs qui sont sous forme binaire: « 1 » montre que le classifieur a donné la bonne solution et « 0 » dans le cas contraire. Soit S la taille de ce vecteur (le nombre de sortie de chaque classifieur e_i tel que i varie de 1 jusqu'au nombre total des classifieurs), donc chaque sortie s varie de 1 jusqu'à S .

3.4.3.1. MESURES "PAIRWISE"

Ces mesures consistent à calculer la diversité entre chaque couple de classifieurs et ensuite faire la moyenne des valeurs calculées. Pour L classifieurs, la moyenne M de chacune des mesures « pairwise » $M_{i,j}$ pour toutes les paires $(e_i; e_j)$ est calculée par:

$$M = \frac{2}{SL(L-1)} \sum_{i=1}^{L-1} \sum_{j=i+1}^L M_{i,j} \quad (14)$$

Pour calculer M , il faut tout d'abord déterminer la relation entre chaque couple de classifieurs (voir **Erreur ! Source du renvoi introuvable.**).

Tableau III-1 Relation entre les sorties de deux classifieurs

| | $e_j \text{ correcte}(1)$ | $e_j \text{ incorrecte}(0)$ |
|-----------------------------|---------------------------|-----------------------------|
| $e_i \text{ correcte}(1)$ | N^{11} | N^{10} |
| $e_i \text{ incorrecte}(0)$ | N^{01} | N^{00} |

- *Q statistique (Yule [123])*

$$Q_{Stat} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}} \quad (15)$$

- *Coefficient de corrélation ρ (Kuncheva and Whitaker [124])*

$$M_{i,j} = \frac{N^{11}N^{00} - N^{01}N^{10}}{\sqrt{(N^{11} + N^{10})(N^{01} + N^{00})(N^{11} + N^{01})(N^{10} + N^{00})}} \quad (16)$$

Q statistique et ρ varient entre -1 et 1 :

La valeur -1 indique que les classifieurs proposent des solutions différentes.

La valeur 1 indique que les classifieurs sont identiques.

Pour des classifieurs statistiquement indépendants, la valeur est égale à 0 .

- *Mesure de désaccord D (Skalak [125])*

$$D = \frac{N^{01} + N^{10}}{N^{11} + N^{10} + N^{01} + N^{00}} \quad (17)$$

- *Mesure de similarité SI (Lecce et al. [126])*

$$SI = \frac{N^{00} + N^{11}}{N^{11} + N^{00} + N^{01} + N^{10}} \quad (18)$$

D et SI Varient entre 0 et 1

La valeur 1 indique que les classifieurs sont identiques

La valeur 0 indique qu'ils proposent des réponses différentes.

- *Mesure de kappa κ (Cohen [127])*

Soit N_{ij} le nombre d'exemples reconnus de la classe C_i par le premier classifieur et de la classe C_j par le deuxième classifieur, N^{i*} , le nombre d'exemples reconnus C_i par le premier classifieur, et N^{*i} , le nombre d'exemples reconnus de la classe C_i par le deuxième classifieur.

On peut alors définir κ par l'équation (19).

$$\kappa = \frac{\theta_1 - \theta_2}{1 - \theta_2} \quad (19)$$

Sachant que θ_1 estime la probabilité lorsque les deux classifieurs soient d'accord et θ_2 estime la probabilité lorsque les deux classifieurs soient d'accord simplement par hasard (puisque chaque classifieur propose sa décision aléatoirement sans connaître la réponse de l'autre classifieur).

$$\theta_1 = \frac{\sum_{i=1}^N N^{ii}}{S} \quad (20)$$

$$\theta_2 = \sum_{i=1}^N \left(\frac{N^{i*}}{S} \cdot \frac{N^{*i}}{S} \right) \quad (21)$$

La valeur de κ est comprise entre -1 et 1.

$\kappa = -1$ indique que les classifieurs sont très différents ;

$\kappa = 1$ indique le contraire.

- **Ratio entre les erreurs R (Aksela [128])**

$$R = \frac{N_{different}^{00}}{N_{meme}^{00}} \quad (22)$$

$N_{different}^{00}$ est le nombre de fois où deux classifieurs proposent en même temps des erreurs différentes. N_{meme}^{00} est le nombre de fois où deux classifieurs proposent les mêmes erreurs.

Une valeur élevée de R signifie que les classifieurs sont différents.

- **Weighted Count of Errors and Correct results (WCEC) (Aksela [128])**

Il s'agit d'une mesure qui tient compte en même temps du résultat correct et incorrect des classifieurs. Une valeur faible de cette mesure signifie que les classifieurs sont identiques.

$$M_{ij} = N^{11} + \frac{1}{2}(N^{01} + N^{10}) - N_{different}^{00} - 5N_{meme}^{00} \quad (23)$$

3.4.3.2. MESURES "NON-PAIRWISE"

Il est possible de quantifier la diversité de manière globale relativement à un ensemble de classifieurs avec les mesures suivantes :

- *L'entropie E (Kuncheva and Whitaker [124])*

$$E = \frac{1}{S} \frac{2}{L-1} \sum_{i=1}^S \min \left\{ \sum_{j=1}^L y_{ij}, L - \sum_{j=1}^L y_{ij} \right\} \quad (24)$$

où S désigne le nombre d'exemples d'apprentissage et L le nombre de classifieurs. Pour l'entropie, E varie entre 0 et 1 .

$E=0$ indique qu'il n'y a pas de différence entre les classifieurs ;

$E=1$ indique une diversité élevée.

Il existe d'autres mesures de type non-pairwise, comme par exemple : Mesure de difficulté θ [129], Erreur relative [130], Diversité généralisée [118], Exponentiel du nombre d'erreurs, Percentage correct diversity measure, Coincident failure [118].

3.5. CONCLUSION

Dans ce chapitre nous avons présenté les différentes terminologies et architectures pour la combinaison d'un ensemble de classifieurs, ainsi que les différentes méthodes et les travaux réalisés dans ce domaine. Selon la multitude des travaux existants, nous remarquons que la combinaison des classifieurs hétérogènes en utilisant une approche hybride, basée à la fois sur la sélection et la fusion des classifieurs, donne de meilleures performances par rapport aux approches classique. Cependant la meilleure sélection des classifieurs de base reste un problème ouvert. Dans le chapitre qui suit nous proposons une solution pour sélectionner les classifieurs avec une approche basée sur un compromis diversité/performance.

Chapitre IV.

NOUVELLE APPROCHE ENSEMBLISTE POUR LA DETECTION DES ARYTHMIES CARDIAQUES

| | | |
|--------|---|-----|
| 4.1. | Introduction | 84 |
| 4.2. | Matériels et méthodes | 85 |
| 4.3. | Fusion des classifieurs par règles fixes | 87 |
| 4.4. | Sélection des classifieurs de base | 89 |
| 4.4.1. | Sélection dynamique des classifieurs | 89 |
| 4.4.2. | Recherche <i>SingleBest</i> | 90 |
| 4.4.3. | Recherche <i>N-Best</i> | 90 |
| 4.4.4. | Recherche vers l'avant..... | 90 |
| 4.4.5. | Discussion des résultats | 91 |
| 4.5. | Méthode proposée pour la sélection des classifieurs de base | 92 |
| 4.5.1. | Description détaillée | 92 |
| 4.5.2. | Résultats et discussion | 94 |
| 4.6. | Etude comparative avec la littérature | 95 |
| 4.7. | Classification ensembliste des données biopuces..... | 98 |
| 4.7.1. | Présentation des bases de données considérées | 98 |
| 4.7.2. | Présentation générale des données biopuces | 99 |
| 4.7.3. | Prétraitement des données | 101 |
| 4.7.4. | Résultats de la combinaison par règles fixes | 103 |
| 4.7.5. | Résultats obtenus en utilisant notre approche..... | 104 |
| 4.7.6. | Comparaison avec la littérature : données biopuces..... | 106 |
| 4.8. | Conclusion | 109 |

4.1. INTRODUCTION

La reconnaissance des arrhythmies cardiaques est un sujet d'une importance majeure qui nécessite d'être à la fois rapide et précise [44]. La sensibilité des conséquences qui peuvent résulter d'une mauvaise identification et/ou une lente analyse rend le suivi manuel non seulement coûteux, mais aussi inefficace. Ce problème a longtemps préoccupé la communauté d'apprentissage automatique et leur a fortement incité à proposer des solutions sérieuses [131]. Plusieurs méthodes dites intelligentes permettant la détection d'arythmies ont été proposées. Ces méthodes ont simplifié les tâches de suivi des maladies et ont offert aux cliniciens un moyen d'assistance très efficace. Dans [132], les auteurs ont examiné différents systèmes de diagnostic cardiaque assisté par ordinateur, et ont présenté les méthodes d'analyse, les défis abordés et l'avenir du dépistage des maladies cardiovasculaires.

Parmi les classifieurs proposés pour la détection automatique des arythmies cardiaques en utilisant l'ensemble de données d'arythmies UCI, nous pouvons citer l'algorithme des Intervalles de fonctions de vote (*VFIS*) [48], le réseau de neurones modulaire [133], le modèle généralisé de réseau neuronal *feed-forward* avec algorithme statique de rétro-propagation [134], *SVM* avec *Spearman Rank Correlation* et analyse en composantes principales (*ACP*) [59], *LS-SVM* avec *Score Fisher (FS)* [135], Système de reconnaissance immunitaire artificiel avec prétraitement pondéré flou [136], *OneRule*, *J48* et *Naive Bayes* [137]. Ces méthodes ont obtenu des performances compétitives qui peuvent être améliorées en choisissant judicieusement la bonne méthode de classification.

C'est ainsi que plusieurs méthodes ensemblistes ont été proposées, telles que: Ensemble de classifieurs basé sur un sous espace aléatoire (*Ransom subspace*) en se basant sur l'arbre *PART* comme classifieur de base [64], une approche multi-vue d'apprentissage [138], Méthodes ensemblistes avec vote majoritaire en utilisant 4 classifieurs et 3 sous ensemble de données pour la sélection des attributs et la classification [55], La méthode Random Forest (forêt aléatoire) avec la stratégie de ré-échantillonnage de données [139], méthodes d'induction pour des arbres de décision incrémentaux [140], *AdaBoost.M1* + *C4.5*, *multi-Boost* + *C4.5*, forêt aléatoire et forêt de rotation [141]. Nous avons remarqué

qu'en règle générale, les performances de ces méthodes ensemblistes dépendent du choix des classifieurs de base. Dans ce chapitre, nous portons une considération particulière au choix des classifieurs et en particulier à l'apport de la diversité entre ces derniers.

L'objectif principal de ce chapitre est d'améliorer les performances de la reconnaissance des arythmies cardiaques et d'automatiser la sélection des classifieurs de base tout en gardant leurs paramètres par défaut. Ainsi, nous proposons un algorithme de recherche basé sur la diversité des classifieurs et permettant de sélectionner un sous ensemble de classifieurs hétérogènes qui seront par la suite combinés avec des règles fixes [111]. Cet algorithme sera testé par la suite sur des données biopuces.

4.2. MATERIELS ET METHODES

Dans ce chapitre, nous avons réalisé l'étude expérimentale sur un ordinateur avec processeur Dual-Core 2,20GHz. Concernant la classification, nous avons travaillé avec la bibliothèque *PRTools* [47] disponible sous Matlab.

Les jeux de données employés sont issus de la base de données *Arrhythmia* [48] en utilisant le prétraitement et la sélection d'attributs du Chapitre II. Nous avons appliqué vingt-cinq (25) classifieurs de nature très différente (hétérogènes), disponibles dans la boîte à outils *PRTools*. Le Tableau IV-1 représente l'abréviation et les noms des classifieurs utilisés.

Tableau IV-1 Abréviations des classifieurs utilisés

| <i>Abréviation</i> | <i>Nom du classifieur</i> |
|--------------------|--|
| <i>Fisher</i> | Fisher's Least Square Linear |
| <i>Ldc</i> | Classifieur Linéaire Bayes normal |
| <i>Loglc</i> | Classifieur linéaire logistique |
| <i>Nmc</i> | Classifieur de la moyenne la plus proche |
| <i>Nmsc</i> | Classifieur Moyenne la plus proche en escalade |
| <i>Subsc</i> | Classifieur subspatial |
| <i>Knnc</i> | Classifieur K-plus proche voisin |
| <i>Weakc</i> | Classifieur faible |
| <i>Stumpc</i> | Classifieur décision stump |

| | |
|----------------|---|
| <i>Parzenc</i> | Classifieur Optimisation de Parzen |
| <i>Dtc</i> | Classifieur Arbre de décision -arbre de Verzakov- |
| <i>Randomc</i> | Classifieur forêt aléatoire |
| <i>Naivebc</i> | Classifieur Naive Bayes |
| <i>Perlc</i> | Classifieur perceptron linéaire |
| <i>Rbnc</i> | Classifieur base radiale réseau fonction neuronale |
| <i>Rnnc</i> | Classifieur Aléatoire Neural Net |
| <i>Vpc</i> | Classifieur perceptron vote |
| <i>Drbmc</i> | Classifieur formé par Machine Boltzmann Discriminative restreinte |
| <i>Svc</i> | Classifieur Support Vector Machine |
| <i>Nusvc</i> | Classifieur Support Vector : algorithme NU |
| <i>Bpxnc</i> | Réseaux de neurone avec apprentissage par rétro-propagation vers l'avant |
| <i>Lmnc</i> | Réseau de neurones avec apprentissage par Levenberg-Marquardt vers l'avant |
| <i>Pcldc</i> | Classifieur linéaire utilisant l'expansion PC sur les données conjointe |
| <i>Klldc</i> | Classifieur linéaire construit sur l'expansion KL de la matrice de covariance commune |
| <i>Rbsvc</i> | Classifieur Support Vector basé sur Radiale automatique |

Afin de tester les performances de ces classifieurs, nous avons divisé la base de données en deux ensembles : 2/3 pour l'apprentissage et 1/3 pour le test. Par la suite, l'évaluation est donnée selon l'erreur de classification (E), la sensibilité (Se) et la spécificité (Sp). Les résultats sont représentés sur le Tableau IV-2.

Tableau IV-2 Performances des classifieurs de base

| <i>Classifieur</i> | <i>E</i> | <i>Se (%)</i> | <i>Sp (%)</i> |
|--------------------|----------|---------------|---------------|
| <i>Fisherc</i> | 0,254 | 78,33 | 75,00 |
| <i>Ldc</i> | 0,237 | 77,27 | 77,36 |
| <i>Loglc</i> | 0,202 | 81,82 | 80,19 |
| <i>Nmc</i> | 0,296 | 72,88 | 71,68 |
| <i>Nmsc</i> | 0,268 | 73,85 | 74,77 |
| <i>Subsc</i> | 0,449 | 46,58 | 73,08 |
| <i>Knnc</i> | 0,146 | 91,94 | 83,64 |
| <i>Weakc</i> | 0,623 | 32,63 | 42,86 |
| <i>Stumpc</i> | 0,427 | 77,78 | 60,39 |
| <i>Parzenc</i> | 0,187 | 92,59 | 78,81 |
| <i>Dtc</i> | 0,127 | 83,54 | 90,32 |
| <i>Rfc</i> | 0,213 | 68,37 | 89,19 |
| <i>Naivebc</i> | 0,208 | 86,21 | 78,07 |
| <i>Perlc</i> | 0,329 | 64,29 | 70,59 |

| | | | |
|--------------|-------|--------|-------|
| <i>Rbnc</i> | 0,44 | 100,00 | 59,51 |
| <i>Rnnc</i> | 0,22 | 80,00 | 78,50 |
| <i>Vpc</i> | 0,293 | 71,43 | 72,48 |
| <i>Drbmc</i> | 0,21 | 82,54 | 78,90 |
| <i>Svc</i> | 0,202 | 81,82 | 80,19 |
| <i>Nusvc</i> | 0,2 | 80,88 | 80,77 |
| <i>Bpxnc</i> | 0,137 | 85,14 | 87,76 |
| <i>Lmnc</i> | 0,163 | 84,29 | 84,31 |
| <i>Pcldc</i> | 0,237 | 77,27 | 77,36 |
| <i>Klldc</i> | 0,237 | 77,27 | 77,36 |
| <i>Rbsvc</i> | 0,15 | 84,72 | 86,00 |

4.3. FUSION DES CLASSIFIEURS PAR REGLES FIXES

Il existe de nombreuses méthodes de combinaison des classifieurs hétérogènes, et selon [142] la combinaison par règle fixe ou par vote sont de loin les plus couramment utilisés dans la pratique. En effet, bien qu'un combinateur puisse être spécifiquement choisi pour optimiser la performance dans une application particulière, ces règles de combinaison montrent toujours un bon comportement à travers de nombreux problèmes, et sont assez simples à appliquer [142].

Le principe de base derrière les règles fixes est que les classifieurs sont indépendants et estiment les probabilités à posteriori des classes [82]. Ainsi, pour reconnaître le type d'arythmie désigné par x , une règle de décision sur x notée $E(x)$ est utilisée pour sélectionner la $i^{\text{ème}}$ classe C_i pour laquelle la $i^{\text{ème}}$ probabilité postérieure P_i est la plus élevée.

$$E(x) = \begin{cases} C_i & \text{if } \max_{i=1}^n P_i = \max_{m=1}^n P_m \\ \text{Rejet Sinon} & \end{cases} \quad (25)$$

Où $m_{i,j}$ désigne la mesure de confiance de la classe C_i par le classifieur e_j , n désigne le nombre de classes et L désigne le nombre de classifieurs. La probabilité a posteriori P_m peut être calculée selon les règles suivantes:

Règle Maximum (Max):

$$P_m = \max_{j=1}^L m_{i,j} \quad (26)$$

Règle Minimum (Min):

$$P_m = \min_{j=1}^L m_{i,j} \quad (27)$$

Règle médiane (Median):

$$P_m = \begin{cases} \frac{m_{i,\frac{L}{2}} + m_{i,\frac{L+2}{2}}}{2} \\ m_{i,\frac{L+1}{2}} \end{cases} \quad (28)$$

Règle produit (Product):

$$P_m = \prod_{j=1}^L m_{i,j} \quad (29)$$

Règle linéaire:

$$P_m = \lambda \sum_{j=1}^L m_{i,j} \quad (30)$$

Où λ est une constante. Ainsi si $\lambda=1$ alors nous obtenons la règle somme, et si $\lambda = \frac{1}{L}$ alors nous obtenons la moyenne simple.

Dans ce chapitre, les classifieurs hétérogènes sont fusionnés en utilisant les règles fixes ainsi que le vote majoritaire (*Voting*). Les résultats obtenus sont résumés dans le Tableau IV-3.

Tableau IV-3 Erreurs obtenus en fusionnant les classifieurs par règles fixes

| <i>Classifieur</i> | <i>Erreur</i> |
|--------------------|---------------|
| <i>Product</i> | 0,553 |
| <i>Mean</i> | 0,228 |
| <i>Median</i> | 0,224 |
| <i>Max</i> | 0,546 |
| <i>Min</i> | 0,623 |
| <i>Voting</i> | 0,183 |

4.4. SELECTION DES CLASSIFIEURS DE BASE

La sélection des classifieurs de base est un réel challenge pour la création d'un ensemble de classifieurs [143]. Cette tâche est liée au choix d'un sous-ensemble de classifieurs parmi tout un pool de classifieurs de base [144]. Nous distinguons deux types de sélection : statique et dynamique [145]. Dans la sélection statique, un sous-ensemble de classifieurs est sélectionné pour créer un classifieur ensembliste. Ce dernier est utilisé dans la classification de tous les objets de l'ensemble de test. La principale problématique dans ce cas est de trouver une fonction d'évaluation pertinente pour sélectionner les classifieurs [144]. La fonction la plus utilisée dans la littérature est l'erreur générée [117]. Il est à noter que selon la fonction d'évaluation choisie, il existe différents algorithmes de recherche. D'un autre côté, la sélection dynamique consiste pour chaque échantillon inconnu, à sélectionner un sous-ensemble spécifique de classifieurs [146]. Cela signifie que différents ensembles de classifieurs sont choisis pour des objets distincts de la base de test. Dans ce type de sélection, le classifieur est sélectionné et affecté à l'échantillon selon différents attributs [147] ou selon différentes régions de décision [148].

4.4.1. SELECTION DYNAMIQUE DES CLASSIFIEURS

La sélection dynamique des classifieurs (*Dynamic Classifier Selection*) a été testé avec un l'algorithme *Dcs* avec en utilisant la bibliothèque *Prtools*. Ainsi, le taux d'erreur de classification obtenu est de **0,146**.

4.4.2. RECHERCHE *SINGLEBEST*

Le principe de cet recherche est de choisir le classifieur le plus performant durant la phase d'apprentissage. Dans cette expérience, le *SingleBest* est le classifieur *knnc* avec un taux d'erreur de *0,146*.

4.4.3. RECHERCHE *N-BEST*

La sélection des N meilleurs classifieurs (*N-Best*) est plus complexe. Pour valider la valeur de N , les combinaisons sont évaluées en augmentant le nombre de classifieur jusqu'à ce que les résultats obtenus, selon des critères prédéfinis, permettent de sélectionner la valeur optimale de N . Le Figure IV.1 représente les performances de la fusion en incrémentant le nombre N par 1. Nous constatons que la majorité des combinateurs ont obtenu de meilleurs résultats lorsque $N = 3$.

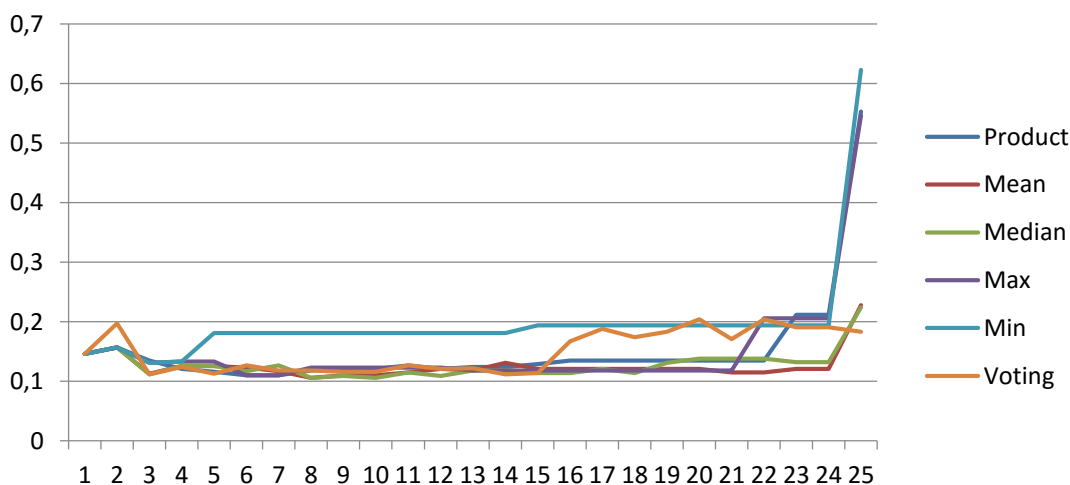


Figure IV.1 Erreur obtenue en utilisant la recherche *N-Best*

4.4.4. RECHERCHE VERS L'AVANT

La recherche progressive vers l'avant retient à chaque itération le classifieur qui améliore la valeur du critère de sélection. En général, toutes les mesures d'optimalité peuvent être incorporées en tant que critères de sélection. Dans cette expérimentation, nous utilisons comme mesure la diversité Q -statistique. En appliquant la recherche vers l'avant 8 classifieurs ont été retenus: *subsc*, *weakc*, *stumpc*, *parzenc*, *randomforestc*, *perlc*, *rbnc* and

bpxnc. Les résultats de l'erreur en fusionnant ces classifieurs par les règles fixes sont résumés dans le Tableau IV-4.

Tableau IV-4 L'erreur de la fusion en utilisant la recherche en avant

| <i>Combineur</i> | <i>Erreur</i> |
|------------------|---------------|
| <i>3Product</i> | 0.523 |
| <i>Mean</i> | 0.191 |
| <i>Median</i> | 0.149 |
| <i>Maximum</i> | 0.523 |
| <i>Minimum</i> | 0.523 |
| <i>Voting</i> | 0.251 |

4.4.5. DISCUSSION DES RESULTATS

Dans ce chapitre, nous avons testé dans un premier temps chaque classifieur individuellement (Les résultats sont données sur le tableau Tableau IV-2). Nous remarquons que le taux d'erreur le plus bas a été obtenu par *Dtc* ($e = 0,127$) tandis que le plus haut taux a été obtenu par *Weakc* ($e=0,623$), la moyenne générale de l'erreur est de **0,258**.

Dans un deuxième temps, dans le but d'améliorer ces résultats, nous avons combiné les sorties des classifieurs en se basant sur les règles fixes et le vote majoritaire [111], [149]. Nous remarquons que les résultats n'ont pas amélioré les résultats par rapport aux classifieurs individuels, en particulier pour les règle max, min et produit (voir Tableau IV-3).

C'est ainsi que nous avons observé la nécessité du choix adéquat des classifieurs de base. En utilisant une sélection dynamique de classifieur nous obtenons une erreur de classification de **0,146**. Nous notons une nette amélioration par rapport à la fusion de tous les classifieurs par règle fixe. Ce résultat confirme l'importance d'appliquer une sélection préliminaire des classifieurs.

En troisième étape, nous utilisons la sélection statique avec l'algorithme « *Single-Best* », ce qui a conduit à l'obtention d'un taux d'erreur de **0,146**, également obtenu en employant la

sélection dynamique. Donc, aucune différence de résultat n'a été remarqué entre la sélection statique et la sélection dynamique sur ce jeu de données.

Après avoir testé la fusion par règles fixes ainsi que la sélection statique et dynamique du meilleur classifieur, nous avons effectué une sélection d'un sous ensemble de classifieurs de base que nous avons ensuite fusionné. Pour cela, nous avons employé deux algorithmes: l'algorithme *N-Best*, qui se base sur l'erreur de classification comme unique critère d'évaluation ainsi que l'algorithme de recherche vers l'avant qui se base uniquement sur la mesure de corrélation *Q-Statistic*. Les résultats obtenus montrent que les performances des règles de combinaison se sont nettement améliorées en utilisant la recherche *N-Best*. Inversement, les taux d'erreurs ont augmenté en utilisant l'algorithme de recherche vers l'avant. Cependant, la méthode *N-Best* est nettement plus complexe par rapport à la recherche vers l'avant.

Ainsi, en tirant les avantages de chacune des deux approches présentées ci-dessus, nous proposons dans la section suivante un algorithme de recherche qui se base à la fois sur les performances individuelles des classifieurs durant la phase d'apprentissage mais aussi sur la corrélation entre eux, tout en poussant le compromis vers les performances individuelles. Ceci a été justifié par les résultats précédents.

4.5. METHODE PROPOSEE POUR LA SELECTION DES CLASSIFIEURS DE BASE

4.5.1. DESCRIPTION DETAILLEE

Notre méthode a été publiée dans [4]. Son schéma global est décrit sur la Figure IV.2. Nous avons divisé un ensemble de données en deux parties (2/3 pour l'apprentissage et 1/3 pour le test). Ensuite, nous avons appliqué le filtre *CFS* avec la méthode de recherche « *BestFirst* ». Après réduction de la dimensionnalité, nous avons évalué chaque classifieur de base individuellement en mesurant l'erreur de classification. Par la suite, nous avons appliqué l'algorithme de recherche décrit dans le Tableau IV-5 pour la sélection des classifieurs. Notre objectif est d'obtenir un ensemble de classifieurs efficaces et diversifiés

(en utilisant Q -statistique) tout en réduisant la complexité. Cet algorithme permet de retenir uniquement les classifieurs qui sont plus performants que la moyenne général de l'erreur générée. Nous avons trié les meilleurs classifieurs selon leur performance durant la phase d'apprentissage. La sélection des classifieurs est procédée de manière itérative. A chaque itération, un classifieur est éliminé de telle sorte que le sous-ensemble des classifieurs retenus donne une diversité plus importante selon le test Q -statistique. A la fin des itérations, nous fusionnons les sorties des membres sélectionnés à l'aide des règles fixes.

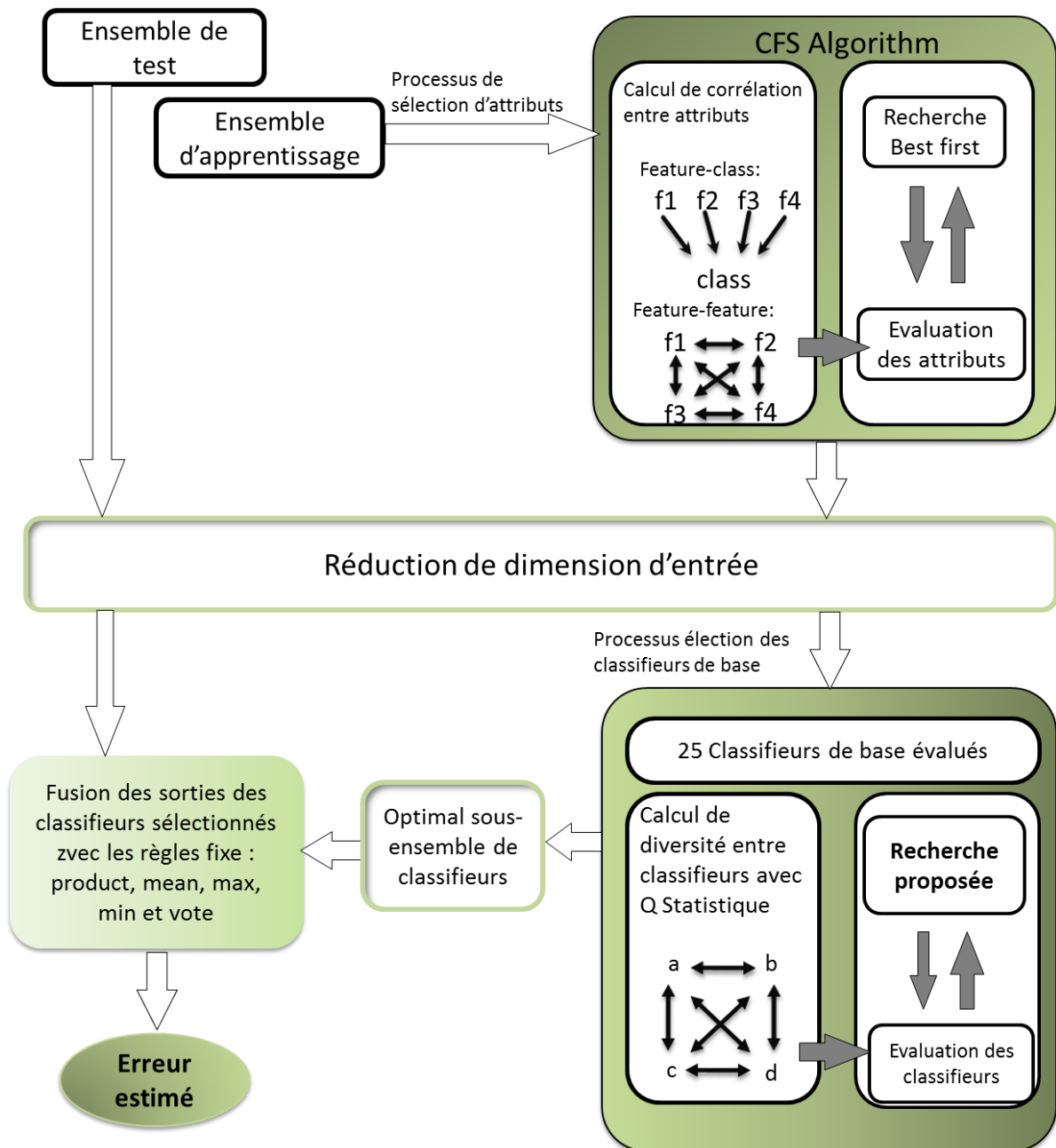


Figure IV.2 Schéma global de l'approche proposée

Tableau IV-5 Pseudo-code de l'algorithme proposé

Let E be an ensemble of L heterogenous classifiers
denoted D_i ($i = 1, \dots, L$),

$$E = \{D_1, D_2, \dots, D_L\}$$

for $i = 1, \dots, L$, sort classifiers according to $error(D)$

$$\bar{e} \leftarrow 1/L \sum_{i=1}^L error(D_i)$$

While $error(D_i) > \bar{e}$ do

$$E = E - \{D_j\}$$

End do

$$L' \leftarrow Card(E)$$

for $i = 1$ to $L' - 1$

for $k = i + 1$ to L'

$$Q_{i,k} \leftarrow \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}}$$

End for

End for

$$Q_{av} \leftarrow \frac{2}{L'(L' - 1)} \sum_{i=1}^{L'-1} \sum_{k=i+1}^{L'} Q_{i,k}$$

For each $D_i \in E$

$$Q'_{av} \leftarrow Q_{av}(E - \{D_i\})$$

if $Q'_{av} < Q_{av}$ then $Q_{av} = Q'_{av}$

else $E = E - \{D_i\}$

End if

4.5.2. RESULTATS ET DISCUSSION

En appliquant l'algorithme de recherche que nous proposons ci-dessus, trois (3) classifieurs ont été retenus, à savoir : *dtc*, *bpxnc* et *randomforestc* et la valeur de Q_{av} égale à -0,3359. Le Tableau IV-3 résume l'erreur de classification obtenue.

Figure IV.3 Résultats d'évaluation de l'algorithme de recherche proposé

| | <i>Erreur</i> | <i>Sp %</i> | <i>Pr %</i> |
|----------------|---------------|-------------|-------------|
| <i>Product</i> | 0,108 | 89,04 | 89,9 |
| <i>Mean</i> | 0,086 | 88,46 | 93,62 |
| <i>Median</i> | 0,102 | 85,19 | 93,41 |
| <i>Maximum</i> | 0,128 | 88,57 | 87,25 |
| <i>Minimum</i> | 0,128 | 88,57 | 87,25 |
| <i>Voting</i> | 0,102 | 85,19 | 93,41 |

La méthode proposée permet d'obtenir une précision de classification remarquable qui varie entre 87,78% en utilisant le combinatoire « maximum » et 91,28% en utilisant le combinatoire « moyenne ».

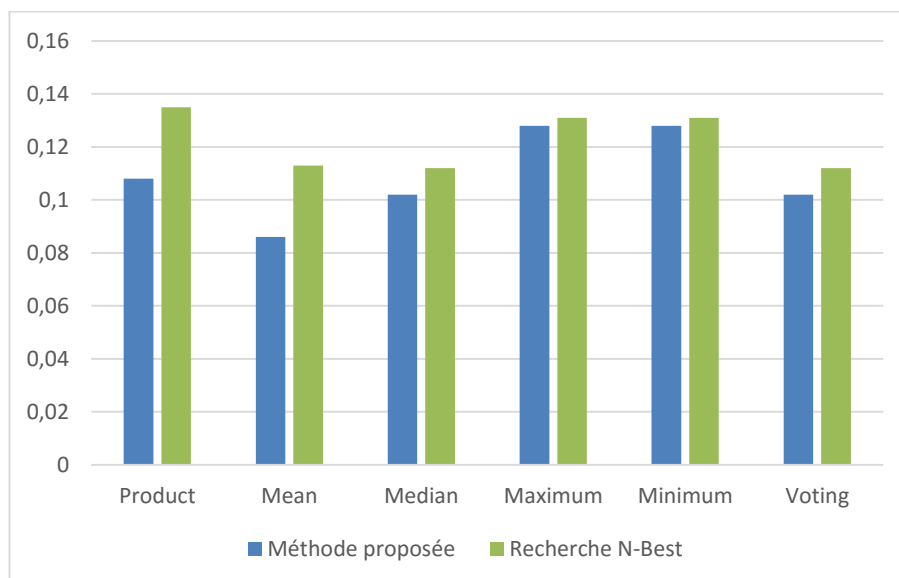


Figure IV.4 Comparaison du taux d'erreur entre la méthode proposée et la recherche *N-Best*

La Figure IV.4 montre une comparaison de notre approche avec la méthode «*N-Best*» (celle qui a obtenu le meilleur résultat dans l'étape précédente), nous remarquons que la méthode proposée a diminué l'erreur. Ainsi nous déduisons que la combinaison entre la performance et la diversité des classifieurs comme critère d'évaluation dans le choix des classifieurs offre de meilleures performances. En effet, la diversité entre les classifieurs ne suffit pas de choisir le jeu le plus optimal de classifieurs. D'un autre côté, le choix du nombre de classifieur se fait automatiquement dans la méthode proposée.

4.6. ETUDE COMPARATIVE AVEC LA LITTERATURE

Afin de se situer par rapport aux approches existantes dans la littérature, nous avons comparé, le taux de bonne classification de la méthode proposée avec celui des différentes études existantes dans la littérature (utilisant le même ensemble de données des arythmies cardiaques provenant du référentiel d'apprentissage automatique UCI avec deux classes). Cette comparaison est récapitulée dans le Tableau IV-6.

Tableau IV-6 Comparaison avec les résultats de la littérature

| Méthode | Référence | Classification(%) |
|---|-----------------------------|--------------------------|
| <i>VFI5-GA</i> | Guvenir et al. 1997 [48] | 68 |
| <i>Modular neural network</i> | Jadhav et al. 2011 [133] | 82.22 |
| <i>GFFNN Model</i> | Jadhav et al. 2012 [134] | 82.35 |
| <i>Spearman Rank Correlation – PCA – SVM</i> | Khare et al. 2012 [59] | 85.98 |
| <i>FS- LS-SVM</i> | Ersen 2013 [135] | 82.09 |
| <i>AIRS with fuzzy weighted pre-processing</i> | Polat et al. 2006 [136] | 81.71 |
| <i>OneR</i> | Soman et al. 2005 [137] | 58.09 |
| <i>J48</i> | | 74.26 |
| <i>Naive Bayes</i> | | 75.00 |
| <i>Random subspace using PART</i> | Jadhav et al. 2013 [64] | 91.11 |
| <i>Multi-view ensemble learning</i> | Sakar et al. 2013 [138] | 71.10 |
| <i>Ensemble classifiers by using a feature selection schema</i> | Namsrai et al. 2013 [55] | 95.24 |
| <i>Random forest with data resampling</i> | Ozcift 2011 [139] | 90.00 |
| <i>Incremental decision trees induction method</i> | Piao et al. 2012 [140] | 78.3 |
| <i>Mean combiner with dtc, bpxnc and randomforestc</i> | Méthode proposée [4] | 91.28 |

Nous remarquons que le taux de classification obtenu par notre approche (91,28%) en se basant sur la règle « mean » est supérieur à ceux obtenus par les autres méthodes, à l'exception de la méthode proposée par [55] où le taux de classification est de 95%. Dans ce dernier, les auteurs utilisent un schéma de sélection d'attributs pour identifier les meilleurs ensembles d'attributs, où chaque sous ensemble forme un modèle de classification. A la fin, un vote majoritaire est utilisé pour déterminer le meilleur sous ensemble. Cependant, l'approche qu'ils ont proposé pour la sélection est de type *Wrapper* et elle a une large complexité algorithmique.

D'autre part, la méthode que nous proposons automatise la sélection de l'ensemble de classifieurs et conserve uniquement les classifieurs les plus efficaces et diversifiées, indépendamment de l'ensemble initial de classifieurs.

D'un autre côté, nous avons appliqué sur le même ensemble de données « Arrhythmia » à des méthodes ensemblistes de classifieurs homogènes. Nous avons utilisé le logiciel Weka avec les paramètres par défaut, les résultats sont représentés dans le Tableau IV-7. Les différentes méthodes ensemblistes que nous avons utilisé sont résumés ci-dessous (notant que les méthodes *Bagging*, *Boosting* et *RandomForest* sont présentés dans le chapitre 3).

Dagging : ce méta-classifieur crée un certain nombre de blocs disjoints et stratifiés sur les données. Il alimente chaque bloc de données par une copie du classifieur de base fourni. Les prédictions sont effectuées selon la moyenne, puisque tous les classifieurs de base générés sont placés dans un meta classifieur de Vote [108].

MultiBoosting : c'est une extension de la technique *AdaBoost* qui vise à former des comités de décision. Elle peut être vue comme un combinant *AdaBoost* avec agitation (*Wagging*), qui est capable d'exploiter à la fois le biais élevé d'*AdaBoost* et la réduction de la variance par l'agitation [150].

RandomSubSpace : cette méthode construit un classifieur basé sur l'arbre de décision qui maintient la précision la plus élevée sur les données d'apprentissage et améliore la précision de généralisation. Le classifieur est constitué de plusieurs arbres construits systématiquement par sélection pseudo-aléatoire de sous-ensembles de composantes du vecteur d'attributs. Ainsi, les arbres construits dans les sous-espaces sont choisis aléatoirement [151].

Rotation Forest : est une méthode qui a été proposée pour construire des ensembles de classifieurs en utilisant des arbres de décision formés indépendamment [152].

Decorate : (en anglais *Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples*) est un méta-classifieur qui consiste à construire de divers ensembles de classifieurs en utilisant des exemples d'apprentissage artificiels spécialement construits [153].

Tableau IV-7 Résultats de la fusion par classifieurs homogènes

| Méthode de combinaison | Classifieur de base | Référence | E |
|------------------------|---------------------|---------------------------------|-------|
| Bagging | REPTree | Breiman (1996) [102] | 0,186 |
| AdaBoost | DecisionStump | Freund et Schapire (1996) [104] | 0,209 |
| Dagging | SMO | Ting et witten (1997) [108] | 0,162 |
| MultiBoost | DecisionStump | Webb (2000) [150] | 0,303 |
| RandomSubSpace | REPTree | Ho (1998) [151] | 0,209 |
| RotationForest | C4.5 | Rodriguez (2006) [152] | 0,163 |
| Decorate | C4.5 | Melville (2004) [153] | 0,116 |

Nous remarquons que la méthode de fusion *DECORATE* a obtenu de meilleurs résultats par rapport aux autres méthodes de fusion des classifieurs homogènes. Notant que cette méthode se base principalement sur la diversité. Toutefois, en utilisant les mêmes ensembles d'apprentissage et de test, notre approche s'est avéré plus performante en termes d'erreur de classification. Pour rappel l'erreur que nous avons obtenu pour tous les règles varie entre 0,086 et 0,128.

4.7. CLASSIFICATION ENSEMBLISTE DES DONNEES BIOPUCES

Afin de consolider davantage l'efficacité de notre méthode, nous avons décidé d'étendre les tests à d'autres types de données. Ainsi, nous avons opté pour ce que l'on appelle les données biopuces. Ce choix est motivé par leur grande dimension d'attributs tout en ayant une petite population. Dans cette section, nous présentons tout d'abord les bases de données considérées, ainsi que le context général des données biopuces. Ensuite, nous exposons et discutons les résultats obtenus.

4.7.1. PRESENTATION DES BASES DE DONNEES CONSIDEREES

Le nombre de jeux des données biopuces a considérablement augmenté ces dernières années. Ainsi, un grand nombre est mis librement à disposition des chercheurs. La majorité de ces données concerne le transcriptome humain. Dans cette partie, nous avons utilisé trois différents jeux de données. Notre objectif est de construire un modèle, à partir des données d'expression, qui détecte la présence d'un cancer ou son type :

Données sur la leucémie (*Leukemia*) [40]: cette maladie se caractérise par une atteinte massive de la moelle osseuse, due au développement de lymphomes malins. On distingue les leucémies aiguës lymphoblastiques (notées **ALL** par la suite pour *Acute Lymphoblastic Leukemia*) des leucémies aiguës myéloblastiques (notées **AML** pour *Acute Myeloid Leukemia*). La distinction entre ces deux formes est essentielle pour le succès des thérapies envisagées : le traitement diffère selon l'une ou l'autre de ces deux classes de leucémie. Le

jeu de données comporte 38 patients (27 patients ALL et 11 patients AML) et concerne 3571 gènes.

Données sur le cancer du côlon (Colon) [41] : le cancer du côlon (ou cancer colorectal) se développe à partir des cellules qui tapissent la paroi interne du côlon. Ce jeu de données contient le profil d'expression de 2000 gènes sur 62 patients. 40 d'entre eux sont atteints d'un cancer du colon et 22 sont sains.

Données sur le cancer du prostate (Prostate) [154]: le cancer de la prostate correspond au développement de cellules cancéreuses dans la prostate. Dans environ 95 % des cas, il s'agit d'un adénocarcinome, c'est-à-dire d'un cancer glandulaire. Cet ensemble de données contient 12625 gènes et 136 échantillons. 59 de ces échantillons proviennent de tissus normaux et les reste sont des tissus tumoraux [18].

4.7.2. PRESENTATION GENERALE DES DONNEES BIOPUCES

L'ADN est la forme de stockage de l'information génétique de tous les êtres vivants. La technologie des puces à ADN ou biopuces, connaît à l'heure actuelle un essor exceptionnel et suscite un formidable intérêt dans la communauté scientifique. Cette technologie a été développée au début des années 1990 et permet la mesure simultanée des niveaux d'expression de plusieurs milliers de gènes, voire d'un génome entier. L'utilité de ces informations est scientifiquement incontestable car la connaissance du niveau d'expression d'un gène constitue une avancée vers sa fonction, mais également vers le criblage de nouvelles molécules et l'identification de nouveaux médicaments et de nouveaux outils de diagnostic [155]–[157].

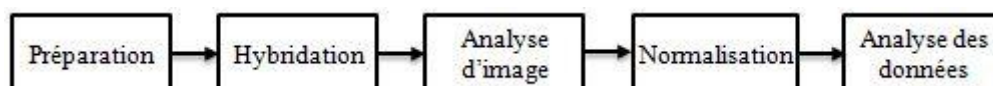


Figure IV.5 Etapes d'une analyse par puces à ADN

Une puce ADN (appelée *DNA microarray* en anglais) est constituée de fragments d'ADN immobilisés sur un support solide, de manière ordonnée [158]. Comme indiqué dans la Figure IV.5, les différentes phases d'une analyse par puces ADN sont :

1. **Préparation des cibles** : Il s'agit d'extraire les ARN messager d'un échantillon biologique à analyser. Une mauvaise purification peut conduire à une augmentation des bruits de fond sur la lame.
2. **L'hybridation** : Consiste à marquer les deux échantillons pour ensuite les hybrider et les nettoyer. Les échantillons sont marqués par des substances fluorescentes, Cy3 (en vert) et Cy5 (en rouge). Ce processus d'hybridation est réalisé dans une station fluide (four) pour favoriser les liaisons entre séquences complémentaires.
3. **Acquisition et analyse des images** : L'obtention des images est réalisée par lecture des puces sur des scanners de haute précision, adaptés aux marqueurs utilisés. Le procédé de détection combine deux lasers, pour exciter les fluorochromes Cy3 et Cy5. On obtient alors deux images dont le niveau de gris représente l'intensité de la fluorescence lue. Si on remplace les niveaux de gris par des niveaux de vert pour la première image et par des niveaux de rouge pour la seconde, on obtient en les superposant une image en fausses couleurs composée de spots allant du vert au rouge quand un des fluorophores domine, en passant par le jaune (même intensité pour les deux fluorophores). Le noir symbolise l'absence de signal. L'intensité du signal de fluorescence pour chaque couple (gène,spot) est proportionnel à l'intensité d'hybridation donc à l'expression du gène ciblé (voir Figure IV.6).

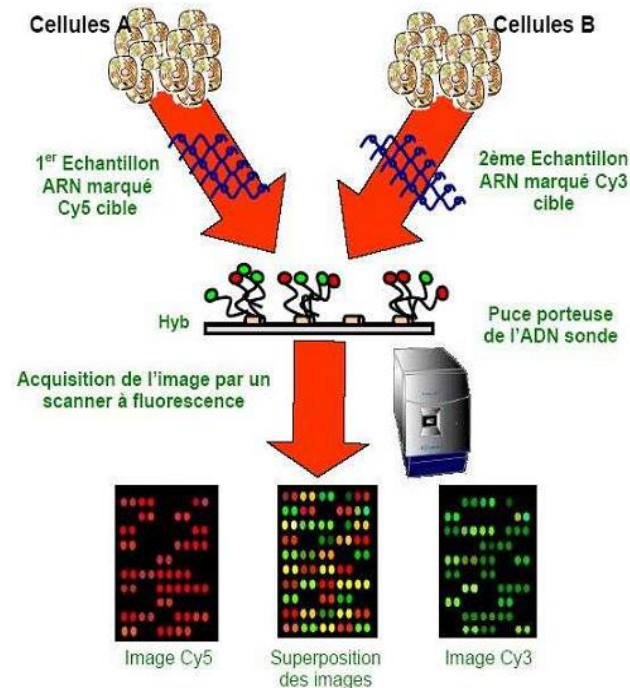


Figure IV.6 Processus d'acquisition de l'image

4. **Normalisation des données** : La normalisation consiste à ajuster l'intensité globale des images acquises sur chacun des deux canaux rouge et vert, de manière à corriger les différences systématiques entre les échantillons sur la même lame, qui ne représentent pas de variations biologiques entre les échantillons et qui tendent à déséquilibrer le signal de l'un des canaux par rapport à l'autre.
5. **Présentation des données de puces à ADN** : Après les transformations, les données recueillies pour l'étude d'un problème donné sont regroupées sous forme de matrice avec une ligne par couple (gène, sonde) et une colonne par échantillon. Les matrices de données qui sont actuellement disponibles ont donc les caractéristiques suivantes : la grande dimensionnalité due au nombre élevé de descripteurs (gènes) et le nombre limité d'échantillons.

4.7.3. PRETRAITEMENT DES DONNEES

Afin de remédier au problème de dimensionnalité, nous avons utilisé le filtre *CFS* (*Correlation Feature Selection*) avec l'algorithme de recherche *BestFirst*. Dans cette étude expérimentale, nous avons appliqué les 25 classifieurs présentés précédemment. Pour

l'évaluation, nous avons retenu les sous-ensembles de test et d'apprentissage par défaut. Ainsi, pour la base de données leukemia : 38 échantillons sont appliqués pour l'apprentissage et 34 pour le test, pour la base du prostate: 102 pour l'apprentissage et 34 pour le test. Pour la base de données Côlon, nous l'avons divisé en 34 instances pour l'apprentissage et 28 instances pour le test. Le nombre d'attributs retenus pour chaque ensemble de données est représenté en gras dans le Tableau IV-8.

Tableau IV-8 Nombre d'attributs pour les différents ensembles de données

| | Nombre d'attributs | Nombre d'attributs en utilisant CFS+BF | Nombre d'instances pour la phase d'apprentissage | Nombre d'instances pour la phase de test |
|-----------------|---------------------------|---|---|---|
| <i>Leukemia</i> | 7129 | 36 | 38 | 34 |
| <i>Prostate</i> | 12600 | 89 | 102 | 34 |
| <i>Colon</i> | 2000 | 13 | 34 | 28 |

Le taux d'erreur obtenu, en utilisant le filtre *CFS*, pour chaque ensemble de données est représenté dans le Tableau IV-9.

Tableau IV-9 Taux d'erreur des données biopuces en utilisant le filtre *CFS*

| ensemble de données/ Classifieur | Leucémie | Prostate | Colon |
|---|-----------------|-----------------|--------------|
| <i>Fisherc</i> | 0,457 | 0 | 0,206 |
| <i>Ldc</i> | 0,525 | 0 | 0,211 |
| <i>Loglc</i> | 0,507 | 0 | 0,456 |
| <i>Nmc</i> | 0,071 | 0,098 | 0,183 |
| <i>Nmsc</i> | 0,143 | 0,048 | 0,261 |
| <i>Subsc</i> | 0,05 | 0,089 | 0,183 |
| <i>Knnc</i> | 0 | 0 | 0,211 |
| <i>weakc</i> | 0,214 | 0,107 | 0,25 |
| <i>stumpc</i> | 0,236 | 0,144 | 0,406 |
| <i>parzenc</i> | 0,036 | 0,01 | 0,211 |
| <i>dtc</i> | 0,086 | 0 | 0,378 |
| <i>randomfc</i> | 0,179 | 0 | 0,428 |
| <i>naivebc</i> | 0,143 | 0,029 | 0,361 |
| <i>perlc</i> | 0,061 | 0 | 0,211 |
| <i>rbnc</i> | 0,5 | 0,29 | 0,5 |
| <i>rnnc</i> | 0,254 | 0,059 | 0,439 |
| <i>vpc</i> | 0,025 | 0,02 | 0,211 |

| | | | |
|--------------|-------|------|-------|
| <i>drbmc</i> | 0,025 | 0 | 0,183 |
| <i>svc</i> | 0,025 | 0 | 0,211 |
| <i>nusvc</i> | 0,025 | 0,01 | 0,211 |
| <i>bpxnc</i> | 0,061 | 0 | 0,328 |
| <i>lmnc</i> | 0,264 | 0 | 0,306 |
| <i>pcldc</i> | 0,525 | 0 | 0,211 |
| <i>klldc</i> | 0,525 | 0 | 0,211 |
| <i>rbsvc</i> | 0,321 | 0 | 0,261 |

Nous remarquons que dans l'ensemble de données *Leukemia*, la meilleure performance a été obtenue par les classifieurs *knnc* ($e=0$) et la plus mauvaise performance a été réalisée par les classifieurs *Ldc*, *Pcldc* et *Klldc* ($e=0,525$). Dans l'ensemble de données Prostate, nous remarquons que la majorité des classifieurs ont obtenus de très bons résultats où la majorité des classifieurs, à savoir 14 classifieurs, a produit un taux d'erreur de 0% tandis que le plus mauvais taux est de 0,29. Généralement, dans ce genre de situation, nous n'avons pas une nécessité de combiner des classifieurs. Toutefois à titre indicatif, nous allons fusionner les sorties des classifieurs même pour cet ensemble de données. Parallèlement, dans l'ensemble de donnée *Colon*, la meilleure performance a été obtenue par les classifieurs *Drbmc*, *Subsc* et *Nmc* ($e=0,183$) et la plus mauvaise performance a été réalisée par le classifieur *rbnc* ($e=0,5$).

4.7.4. RESULTATS DE LA COMBINAISON PAR REGLES FIXES

Après avoir testé chaque classifieur de base individuellement, nous avons fusionné les sorties de tous les classifieurs en utilisant les règles fixes. Les résultats sont résumés dans le Tableau IV-10.

Tableau IV-10 fusion de tous les classifieurs de base par les règles fixes : application aux données biopuces

| Ensembles de données/ Règles de combinaison | Leucémie | Prostate | Colon |
|--|-----------------|-----------------|--------------|
| <i>Product</i> | 0,5 | 0,5 | 0,211 |
| <i>Mean</i> | 0,036 | 0 | 0,183 |
| <i>Median</i> | 0,036 | 0 | 0,211 |
| <i>Maximum</i> | 0,464 | 0,02 | 0,211 |

| | | | |
|----------------|-------|-------|-------|
| <i>Minimum</i> | 0,286 | 0,049 | 0,161 |
| <i>Voting</i> | 0,071 | 0 | 0,211 |

4.7.5. RESULTATS OBTENUS EN UTILISANT NOTRE APPROCHE

Dans cette partie, nous avons testé l'algorithme proposé (voir la Figure IV.2 présentée ci-dessus), les résultats obtenus sont schématisés dans le Tableau IV-11.

Tableau IV-11 Méthode proposée : application aux données biopuces

| <i>Ensembles de données/ Règles de combinaison</i> | <i>Leucémie</i> | <i>Prostate</i> | <i>Colon</i> |
|--|-----------------|-----------------|--------------|
| <i>Product</i> | 0,5 | 0,5 | 0,183 |
| <i>Mean</i> | 0 | 0 | 0,283 |
| <i>Median</i> | 0 | 0 | 0,183 |
| <i>Maximum</i> | 0,393 | 0 | 0,183 |
| <i>Minimum</i> | 0,286 | 0 | 0,211 |
| <i>Voting</i> | 0,071 | 0 | 0,211 |

Les résultats obtenus sont très satisfaisants pour les deux ensembles de données « Leucémie » et « Prostate », excepté pour la règle « produit » où $e=0,5$.

Afin d'observer l'apport de la méthode proposée, nous avons comparé les résultats obtenus, en fusionnant les sorties de tous les classifieurs par règles fixes, avec ceux obtenus en sélectionnant d'abord l'ensemble optimal, selon notre méthode, avant de procéder à leur fusion. Notant que toutes les expérimentations sont appliquées sur les mêmes ensembles de données.

Concernant l'ensemble de données « Leukemia », la comparaison est schématisée dans la Figure IV.7. Nous remarquons que notre approche a amélioré les résultats pour les règles *Mean*, *Median* et *Maximum*, tandis que pour les autres règles l'erreur est restée la même. Le meilleur taux d'erreur de 0% est obtenu par les règles *Mean* et *Median*.

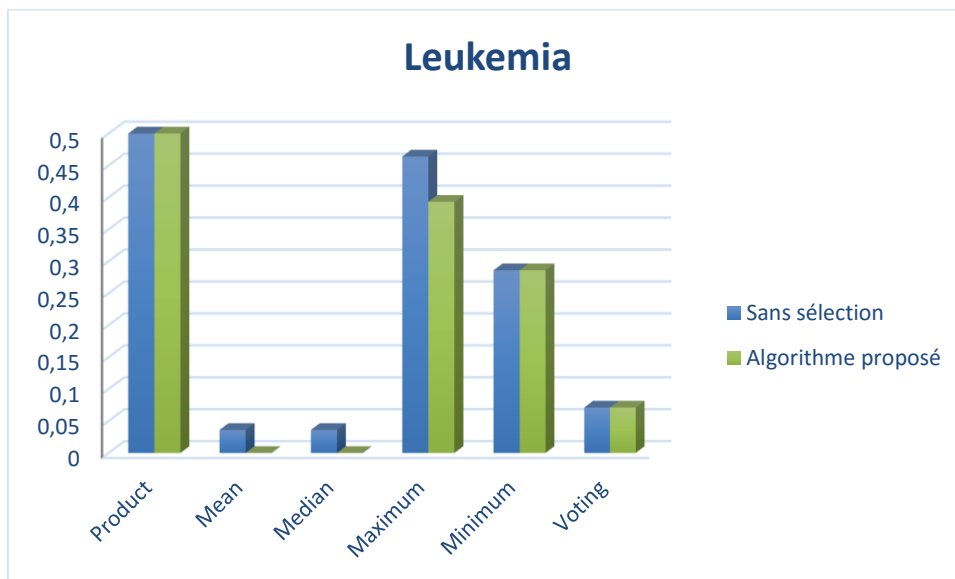


Figure IV.7 Comparaison du taux d'erreur pour la détection des types de leucémies

Pour l'ensemble de données « Prostate », la comparaison du taux d'erreur obtenu, en utilisant notre approche pour sélectionner les classifieurs de base, ou en utilisant tous les classifieurs, est décrite sur la Figure IV.8. Nous remarquons que notre approche a amélioré les résultats pour les règles *Minimum* et *Maximum*, tandis que pour les autres règles l'erreur est restée la même. Le meilleur taux d'erreur de *0%* est obtenu par toutes les règles, à l'exception du *Produit*.

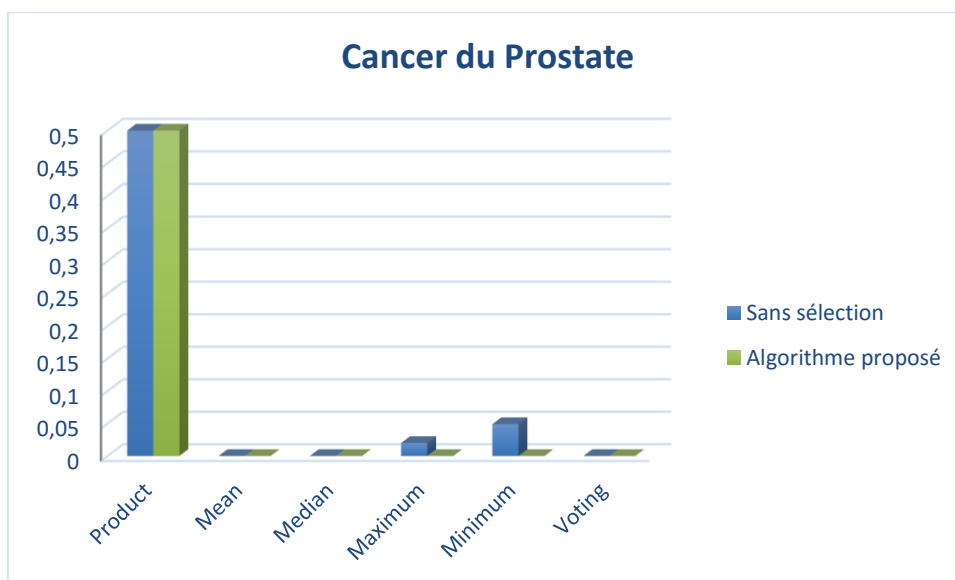


Figure IV.8 Comparaison du taux d'erreur pour la détection du cancer de prostate

Le dernier ensemble de données a pour objectif de distinguer entre la présence et l'absence du cancer du colon. Nous remarquons que pour la majorité des règles de fusion, notre algorithme a amélioré les résultats avec un taux d'erreur $e=0,183$. Cependant, le meilleur taux a été obtenu par la règle minimum en utilisant tous les classifieurs avec $e=0,150$

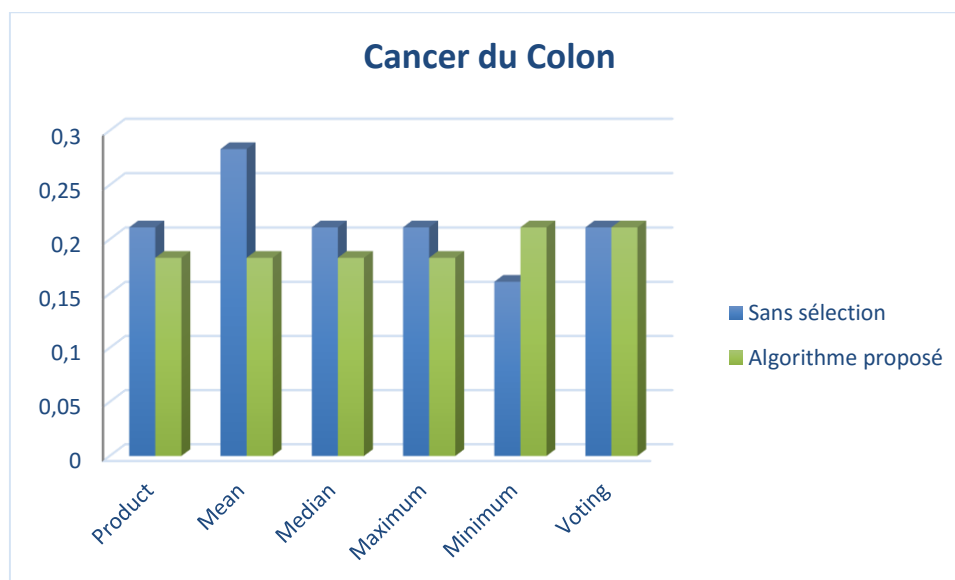


Figure IV.9 Comparaison du taux d'erreur pour la détection du cancer du colon

4.7.6. COMPARAISON AVEC LA LITTERATURE : DONNEES BIPUCES

Afin de se situer par rapport aux différents travaux publiés dans la littérature, nous avons comparé, dans le Tableau IV-12, les résultats obtenus en utilisant notre approche avec ceux de la littérature

Tableau IV-12 Etat de l'art sur les données bopuces

| Référence | Méthode utilisée | Taux d'erreur (%) | | |
|-----------------------------------|------------------------------------|-------------------|-------|----------|
| | | Leukemia | Colon | Prostate |
| Bolton- Canedo et al. [159] | Filtre Ensemble + classifieur C4.5 | 11,96 | 13,1 | 11,81 |
| | Filtre C4.5 + classifieur C4.5 | 23,04 | 23,81 | 19,78 |
| | Filtre CFS + classifieur C4.5 | 16,25 | 19,29 | 22,97 |
| | Filtre Cons + classifieur C4.5 | 14,82 | 16,43 | 16,92 |

| | | | | |
|-----------------------------|--|-------|-------|-------|
| | Filtre INT + classifieur C4.5 | 16,25 | 10,95 | 18,46 |
| | Filtre InfoGain + classifieur C4.5 | 12,5 | 14,29 | 9,4 |
| | Filtre ReliefF + classifieur C4.5 | 11,25 | 22,62 | 10,27 |
| | Filtre Ensemble + classifieur NaiveBayes | 4,11 | 16,19 | 41,87 |
| | Filtre NaiveBayes + classifieur NaiveBayes | 1,25 | 47,86 | 44,78 |
| | Filtre CFS + classifieur NaiveBayes | 2,86 | 18,1 | 36,7 |
| | Filtre Cons + classifieur NaiveBayes | 10,54 | 20 | 39,01 |
| | Filtre INT + classifieur NaiveBayes | 4,11 | 19,29 | 30,93 |
| | Filtre InfoGain + classifieur NaiveBayes | 4,11 | 22,86 | 42,8 |
| | Filtre ReliefF + classifieur NaiveBayes | 4,29 | 14,52 | 40,27 |
| | Filtre Ensemble + classifieur IB1 | 5,54 | 19,05 | 12,53 |
| | Filtre IB1 + classifieur IB1 | 9,46 | 26,62 | 20,11 |
| | Filtre CFS + classifieur IB1 | 5,54 | 19,29 | 21,04 |
| | Filtre Cons + classifieur IB1 | 9,46 | 26,43 | 20,11 |
| | Filtre INT + classifieur IB1 | 4,11 | 22,86 | 20,05 |
| | Filtre InfoGain + classifieur IB1 | 6,61 | 20,95 | 16,87 |
| | Filtre ReliefF + classifieur IB1 | 6,96 | 25,71 | 12,58 |
| <i>Dettling</i> [160] | BagBoost | 4,08 | 16,1 | 7,53 |
| | Boosting | 5,67 | 19,14 | 8,71 |
| | RandomForest | 1,92 | 14,86 | 9 |
| | SVM | 1,83 | 15,05 | 7,88 |
| | PAM | 3,75 | 11,9 | 16,53 |
| | DLDA | 2,92 | 12,86 | 14,18 |
| | kNN | 3,83 | 16,38 | 10,59 |
| <i>Cadenas et al.</i> [161] | Fuzzy RandomForest | 1,39 | 6,45 | 2,94 |

| | | | | |
|--------------------------------------|---|----------|-------------|----------|
| <i>Kohbalan Moorthy et al.</i> [162] | Meilleurs résultats en utilisant la méthode RandomForest pour la sélection des intervalles des gènes | 7,52 | 15,39 | 5,84 |
| Méthode proposée | CFS et BestFirst pour la sélection des gènes + algorithme proposé par fusion avec la règle médiane | 0 | 18,3 | 0 |

Bolon-Canedo et al. [159], ont testé trois classifieurs différents, à savoir *C4.5*, *NaiveBayes* et *IB1* (le plus proche voisin), sur différentes distributions de données issues de l'application de plusieurs types de sélection d'attributs. Le premier type, est un filtre ensembliste (qui représente un vote entre les sous ensemble d'attributs retenus en utilisant les métriques qui suivent : *CFS*, *Cons*, *INT*, *InfoGain* et *ReliefF*), le second est un wrapper qui dépend du classifieur, les autres filtre sont respectivement : *CFS (Correlation Feature Selection)*, *Cons (Consistance)*, *INT (INTERECT* qui est un algorithme basé sur l'incertitude symmitrique), *InfoGain (Information Gain) et ReliefF*. Dans ce travail, la meilleure performance pour l'ensemble de données « *Leukemia* » est obtenu en utilisant une sélection d'attributs de type *Wrapper* avec le classifieur *NaiveBayes* ; le taux d'erreur généré est égal à *1,25*. Pour la base « *Colon* », le meilleur taux d'erreur est obtenu par le classifieur *C4.5* en utilisant le filtre *INT (e=10,95)*. Pour la base « *Prostate* », le meilleur taux d'erreur est obtenu par le classifieur *C4.5* avec le filtre *InfoGain* tel que *e=9,4*.

Dettling [160] a préféré sélectionner les *200 gènes* les plus descriptives en utilisant la *statistique de Wilcoxon* pour la sélection des attributs. Il a proposé d'appliquer la méthode *Boosting* en utilisant *100 itérations* avec un nouvel algorithme « *LogitBoost* ». Ensuite, il a comparé ses résultats avec des méthodes existantes, telles que *RandomForest*, *KNN* et *SVM*. Le meilleur classifieur pour chaque base de donnée est *SVM* pour « *Leukemia* » avec *e=1,83*, *PAM* pour « *Colon* » avec *e=11,9* et *BagBoost* pour « *Prostate* » avec *e=7,53*.

Cadenas et al. [161] ont utilisé les règles floues avec les forêts aussi bien pour la sélection d'attributs que pour la classification. Ils ont obtenu des résultats très performants. Toutefois, leur méthode est très coûteuse en terme de complexité algorithmique.

Nous remarquons que les résultats que nous avons obtenus pour les base « *Leukemia* » et « *Prostate* » sont bien meilleurs que la littérature avec un taux d'erreur de 0%. Quant à la base « *Colon* », nos résultats restent compétitifs avec une complexité plus réduite par rapport aux méthodes *wrapper*.

4.8. CONCLUSION

L'objectif principal de la combinaison des classifieurs est d'améliorer leur capacité de généralisation et la performance de reconnaissance. Le choix correct des classifieurs de base est fondamental pour assurer la performance globale d'un ensemble. C'est pourquoi nous avons proposé de sélectionner les classifieurs de base en se basant principalement sur leurs performances individuelles ainsi que sur la diversité mutuelle entre classifieurs. Une fois le sous-ensemble validé, nous pouvons fusionner leurs décisions. L'approche proposée a été testée, comparée à la littérature et validée avec succès en considérant des bases de données d'arythmie cardiaque ainsi que plusieurs bases de données biopuces. Les résultats ont permis de démontrer l'efficacité de notre méthode par rapport à l'état de l'art.

Conclusion générale et perspectives

Les travaux présentés dans ce mémoire abordent le problème de classification des données biomédicales et concernent en particulier la sélection des attributs pertinents ainsi que la combinaison parallèle des classifieurs hétérogènes.

Nos principaux objectifs étant d'apporter une contribution à l'évaluation, la compréhension et l'amélioration du comportement des méthodes ensemblistes dans le domaine biomédical ainsi que de mettre en valeur le rôle clé de la sélection des classifieurs de base. L'approche que nous avons proposée pour atteindre ces objectifs a été testée sur des bases de données réelles issues de plusieurs répertoires médicaux mises à disposition des chercheurs.

Dans ce contexte, la première partie de ce travail a été consacrée à l'analyse des travaux effectués dans la littérature sur la sélection d'attributs qui se distinguent principalement en trois types : les méthodes *wrapper* (enveloppantes), *embedded* (intégrées) et *filter* (les filtres). Afin d'approfondir ces connaissances, nous avons réalisé une étude sur les différentes méthodes de sélection de variables. L'étude a été restreinte à l'approche « filter » de par sa simplicité et son indépendance du modèle de classification.

Cette expertise nous a conduit à retenir la méthode basée sur l'algorithme de recherche *BestFirst* et le filtre *CFS* (*Correlation Feature Selection*) et qui présente les meilleurs résultats en sélection d'attributs. Le choix s'est porté sur le filtre *CFS* de par ses caractéristiques théoriques qui permettent de conserver les attributs pertinents et d'éliminer les attributs redondants.

Dans la deuxième partie, les différentes méthodes de combinaison qui existent dans l'état de l'art ont été étudiées et présentées. Cela nous a permis de mettre au point les différentes caractéristiques de chaque méthode ainsi que les principales motivations des chercheurs dans l'évaluation du comportement de ces méthodes. Par ailleurs, l'accent a été porté sur le rôle de la diversité dans la sélection des classifieurs de base. Cette partie a été appuyée par une étude expérimentale dans laquelle nous nous sommes particulièrement intéressés par la

combinaison des classifieurs hétérogènes en utilisant des règles fixes et le vote majoritaire. L'objectif principal de la combinaison des classifieurs est d'améliorer leur capacité de généralisation et la performance de reconnaissance. Un choix correct des classifieurs de base est d'un grand apport pour la performance globale d'un ensemble. De ce fait, notre contribution porte sur la sélection des classifieurs de base selon le meilleur compromis entre leurs performances individuelles et leurs diversités mutuelles. Une fois le sous-ensemble validé, la fusion des décisions peut être entamée. Les résultats obtenus dans les expérimentations nous ont permis de mettre en évidence les idées suivantes :

- Le compromis entre la performance individuelle et la diversité mutuelle des classifieurs est un point clé qui a été testé sur plusieurs bases de données et a conduit à l'amélioration des performances.
- L'intérêt du prétraitement de l'ensemble des données : le ré-échantillonnage des données a permis d'améliorer considérablement les résultats.
- L'intérêt de la bonne sélection des attributs : la majorité des filtres testés ont amélioré les performances des classifieurs.
- L'intérêt du filtre multivarié par rapport au filtre univarié: bien qu'il présente une complexité algorithmique plus importante, le filtre multivarié prend en considération la relation inter-attributs et permet de fixer automatiquement le choix du nombre d'attributs.
- L'intérêt des multi classifieurs par rapport aux mono-classifieurs : l'étude expérimentale a confirmé l'étude théorique. Toutefois, il faut s'assurer que certaines conditions citées ci-dessous soient respectées.
- L'intérêt de la bonne sélection des classifieurs de base: en particulier s'ils sont hétérogènes.
- Le rôle du bon compromis entre les performances individuelles et la diversité entre classifieurs.

Toutefois, ce travail nous a conduit à développer des perspectives très intéressantes qui s'inscrivent parfaitement dans la continuité des recherches que nous souhaitons aborder

dans le futur. Tout d'abord, nous visons à court terme d'aborder d'autres schémas de combinaison en utilisant d'autres critères d'évaluation. Nous envisageons également de travailler sur la combinaison des classifieurs homogènes où il serait intéressant d'appliquer le même principe de compromis performance/diversité pour la sélection des classifieurs de base (telle que la sélection optimale des arbres dans une forêt aléatoire) .

Nous estimons qu'il sera très pertinent de lancer une réflexion approfondie sur les points suivants :

- Sélection ensembliste d'attributs ;
- Prétraitement des données déséquilibrées;
- Combinaison des classifieurs par règles d'induction ;
- Intégrer dans un modèle de classification ensembliste et distribué un système multi-agents afin de bénéficier de tous leurs avantages.

Ce travail m'a permis de me familiariser avec les techniques de l'apprentissage automatique, et particulièrement les approches ensemblistes, mais surtout de comprendre les enjeux pour fiabiliser les applications dédiées à l'aide au diagnostic médical. Je suis tout à fait consciente que cela ne constitue qu'une source d'inspiration pour des réflexions qui devront se poursuivre dans le futur.

REFERENCES BIBLIOGRAPHIQUES

- [1] A. Cornuéjols and L. Miclet, *Apprentissage artificiel : Concepts et algorithmes*, vol. 50, no. 5. 2003.
- [2] A. Konar, *Artificial intelligence and soft computing: Behavioral and cognitive modeling of the human brain*, vol. 15, no. 8. 2001.
- [3] C. Sammut and G. I. Webb, *Encyclopedia of Machine Learning*, Springer S. 2011.
- [4] A. Boublenza, M. A. Chikh, and S. Bouchikhi, "Classifier Set Selection for Cardiac Arrhythmia Diagnosis Using Diversity," *J. Med. Imaging Heal. Informatics*, vol. 5, no. 3, pp. 513–519, 2015.
- [5] Y. Bennani, S. Guérif, and E. Viennet, "Réduction des dimensions des données en apprentissage artificiel."
- [6] S. Guérif, "Réduction de dimension en Apprentissage Numérique Non Supervisé," l'Université Paris 13, 2006.
- [7] L. Yu and H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution," *Int. Conf. Mach. Learn.*, pp. 1–8, 2003.
- [8] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artif. Intell.*, vol. 97, no. 1, pp. 245–271, 1997.
- [9] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 2, pp. 153–158, 1997.
- [10] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, no. Mar, pp. 1157–1182, 2003.
- [11] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1–2, pp. 273–324, 1997.
- [12] Koller, Daphne, and Sahami Mehran, "Toward Optimal Feature Selection," 1996.
- [13] E. P. Xing, M. I. Jordan, and R. M. Karp, "Feature selection for high-dimensional genomic microarray data," *Proc. 18th Int. Conf. Mach. Learn.*, pp. 601–608, 2001.
- [14] M. A. Hall, "Correlation-based Feature Selection for Machine Learning," University of Waikato, 1999.
- [15] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, no. 3, pp. 131–156, 1997.
- [16] J. H. (John H. Holland and J. H., *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, 1975.
- [17] D. E. Goldberg and J. H. Holland, "Genetic Algorithms and Machine Learning," *Mach. Learn.*, vol. 3, no. 2/3, pp. 95–99, 1988.
- [18] M. Kudo and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers," *Pattern Recognit.*, vol. 33, no. 1, pp. 25–41, 2000.
- [19] M. Kalakech, "Sélection semi-supervisée d'attributs: Application à la classification de textures couleur," Université de Lille 1, 2011.
- [20] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.
- [21] Z. Zhu, Y.-S. Ong, and M. Dash, "Wrapper-filter feature selection algorithm using a memetic framework.," *IEEE Trans. Syst. Man. Cybern. B. Cybern.*, vol. 37, no. 1,

- pp. 70–6, Feb. 2007.
- [22] S. Della Pietra, V. Della Pietra, and J. Lafferty, “Inducing Features of Random Fields,” *IEEE Trans. PATTERN Anal. Mach. Intell.*, vol. 19, no. 4, 1997.
- [23] H. Khoufi Zouari, “Contribution à l’évaluation des méthodes de combinaison parallèle de classifieurs par simulation,” UNIVERSITE DE ROUEN U.F.R. DES SCIENCES ET TECHNIQUES, 2004.
- [24] L. Xu, A. Krzyzak, and C. Y. Suen, “Methods of combining multiple classifiers and their applications to handwriting recognition,” *IEEE Trans. Syst. Man. Cybern.*, vol. 22, no. 3, pp. 418–435, 1992.
- [25] M. SAMB, F. CAMARA, S. NDIAYE, Y. SLIMANI, and M. A. ESSEGHIR, “Approche de sélection d’attributs pour la classification basée sur l’algorithme RFE-SVM,” *Colloq. Africain sur la Rech. en Inform.*, vol. 17, no. 12, pp. 197–219, 2014.
- [26] S. Puuronen, A. Tsymbal, and I. Skrypnik, “Advanced local feature selection in medical diagnostics,” in *Computer-Based Medical Systems, 2000. CBMS 2000. Proceedings. 13th IEEE Symposium on*, 2000, pp. 25–30.
- [27] K. Rajeswari, V. Vaithyanathan, and S. V. Pede, “Feature selection for classification in medical data mining,” *Int. J. Emerg. Trends Technol. Comput. Sci.*, vol. 2, no. 2, pp. 492–497, 2013.
- [28] H. H. Inbarani, A. T. Azar, and G. Jothi, “Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis,” *Comput. Methods Programs Biomed.*, vol. 113, no. 1, pp. 175–185, 2014.
- [29] L. P. Hung, R. Alfred, A. Hijazi, and M. Hanafi, “A Review on Feature Selection Methods for Sentiment Analysis,” *Adv. Sci. Lett.*, vol. 21, no. 10, pp. 2952–2956, 2015.
- [30] R. L. Babu and S. Vijayan, “Wrapper based feature selection in semantic medical information retrieval,” *J. Med. Imaging Heal. Informatics*, vol. 6, no. 3, pp. 802–805, 2016.
- [31] G. Forman, “An extensive empirical study of feature selection metrics for text classification,” *J. Mach. Learn. Res.*, vol. 3, no. Mar, pp. 1289–1305, 2003.
- [32] T. Deselaers, D. Keysers, and H. Ney, “Features for image retrieval: an experimental comparison,” *Inf. Retr. Boston.*, vol. 11, no. 2, pp. 77–107, 2008.
- [33] K. Verma, B. K. Singh, P. Tripathi, and A. S. Thoke, “Review of Feature Selection Algorithms for Breast Cancer Ultrasound Image,” Springer International Publishing, 2015, pp. 23–32.
- [34] S. L. Salzberg, A. L. Delcher, S. Kasif, and O. White, “Microbial gene identification using interpolated Markov models,” *Nucleic Acids Res.*, vol. 26, no. 2, pp. 544–548, 1998.
- [35] Y. Saeys, P. Rouzé, and Y. de Peer, “In search of the small ones: improved prediction of short exons in vertebrates, plants, fungi and protists,” *Bioinformatics*, vol. 23, no. 4, pp. 414–420, 2007.
- [36] A. Keleş and A. Keleş, “ESTDD: Expert system for thyroid diseases diagnosis,” *Expert Syst. Appl.*, vol. 34, no. 1, pp. 242–246, Jan. 2008.
- [37] M. G. Tadesse, M. Vannucci, and P. Liò, “Identification of DNA regulatory motifs using Bayesian variable selection,” *Bioinformatics*, vol. 20, no. 16, pp. 2553–2561, 2004.
- [38] H. Liu, H. Han, J. Li, and L. Wong, “Using amino acid patterns to accurately predict translation initiation sites,” *In Silico Biol.*, vol. 4, no. 3, pp. 255–269, 2004.
- [39] R. L. Somorjai, B. Dolenko, and R. Baumgartner, “Class prediction and discovery

- using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions.,” *Bioinformatics*, vol. 19, no. 12, pp. 1484–91, Aug. 2003.
- [40] T. R. Golub, “Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring,” *Science (80-.)*, vol. 286, no. 5439, pp. 531–537, Oct. 1999.
- [41] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, “Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays,” *Proc. Natl. Acad. Sci.*, vol. 96, no. 12, pp. 6745–6750, 1999.
- [42] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini, “Tissue classification with gene expression profiles,” *J. Comput. Biol.*, vol. 7, no. 3–4, pp. 559–583, 2000.
- [43] D. T. Ross, U. Scherf, M. B. Eisen, C. M. Perou, C. Rees, P. Spellman, V. Iyer, S. S. Jeffrey, M. de Rijn, M. Waltham, and others, “Systematic variation in gene expression patterns in human cancer cell lines,” *Nat. Genet.*, vol. 24, no. 3, pp. 227–235, 2000.
- [44] “OMS (Organisation Mondiale de la Santé), ‘Maladies cardiovasculaires,’” *WHO*, 2015.
- [45] M. A. Hall and G. Holmes, “Benchmarking Attribute Selection Techniques for Discrete Class Data Mining,” vol. 15, no. 3, pp. 1–16, 2003.
- [46] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA data mining software,” *ACM SIGKDD Explor.*, vol. 11, no. 1, pp. 10–18, 2009.
- [47] R. P. W. Duin, P. Juszczak, D. de Ridder, P. Paclik, E. Pekalska, and D. M. J. Tax, “PRTools, a Matlab toolbox for pattern recognition,” 2004. [Online]. Available: <http://www.prtools.org>.
- [48] H. A. Guvenir, B. Acar, G. Demiroz, and C. Ayhan, “A Supervised Machine Learning Algorithm for Arrhythmia Analysis,” *Comput. Cardiol. IEEE*, vol. 24, pp. 433–436, 1997.
- [49] M. Lichman, “UCI Machine Learning Repository,” *University of California, Irvine, School of Information and Computer Sciences*, 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>.
- [50] C. H. Yu, “Resampling methods: concepts, applications, and justification,” *Pract. Assessment, Res. Eval.*, vol. 8, no. 19, pp. 1–23, 2003.
- [51] P. I. Good, *Resampling methods: A Practical Guide to Data Analysis*. Springer, 2006.
- [52] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.
- [53] B. Efron and B. Efron, *The jackknife, the bootstrap and other resampling plans*, vol. 38. SIAM, 1982.
- [54] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*. CRC press, 1994.
- [55] E. Namsrai, T. Munkhdalai, and M. Li, “A Feature Selection-based Ensemble Method for Arrhythmia Classification,” *J Inf Process Syst*, vol. 9, no. 1, pp. 31–40, 2013.
- [56] L. F. D. P. Sotto, P. Unifesp, R. C. Coelho, P. Unifesp, and P. Unifesp, “Classification of Cardiac Arrhythmia by Random Forests with Features Constructed by Kaizen Programming with Linear Genetic Programming,” in

- GECCO '16, July 20-24, 2016, Denver, CO, USA, 2016, pp. 813–820.
- [57] A. Uyar and F. Gürgen, “Arrhythmia Classification Using Serial Fusion of Support Vector Machines and Logistic Regression,” in *IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, 2007, no. September, pp. 560–565.
- [58] G. Aslantas, F. G. S, and Albert Ali Salah, “GA-NN APPROACH FOR ECG FEATURE SELECTION IN RULE BASED ARRHYTHMIA CLASSIFICATION,” *Neural Netw. World*, vol. 14, no. 3, pp. 267–283, 2014.
- [59] S. Khare, A. Bhandari, S. Singh, and A. Arora, “Spearman Rank Correlation and Support Vector Machine Shreya,” in *Proceedings of the International Conference on SocProS*, 2012, pp. 591–598.
- [60] S. Shilaskar and A. Ghatol, “Feature selection for medical diagnosis : Evaluation for cardiovascular diseases,” *Expert Syst. Appl.*, vol. 40, no. 10, pp. 4146–4153, 2013.
- [61] S. Deepthi, A. Ravikumar, and R. V. Nair, “Evaluation of Classification Techniques for Arrhythmia Screening of Astronauts,” *Procedia Technol.*, vol. 24, pp. 1232–1239, 2016.
- [62] A. Bahadorinia, A. Dolatabadi, and A. Hajipour, “A Hybridized Artificial Neural Network and Optimization Algorithms for the Diagnosis Of Cardiac Arrhythmias,” *Adv. Comput. Sci. an Int. J.*, vol. 3, no. 4, pp. 51–58, 2014.
- [63] M. Mitra and R. K. Samanta, “Cardiac Arrhythmia Classification Using Neural Networks with Selected Features,” *Procedia Technol.*, vol. 10, pp. 76–84, 2013.
- [64] S. Jadhav, S. Nalbalwar, and A. Ghatol, “Feature elimination based random subspace ensembles learning for ECG arrhythmia diagnosis,” *Soft Comput. Springer*, 2013.
- [65] Y. Brunet-Moret, “Le test du Khi 2 de Pearson,” 1966.
- [66] R. L. Plackett, “Karl Pearson and the chi-squared test,” *Int. Stat. Rev. Int. Stat.*, pp. 59–72, 1983.
- [67] K. Kira and L. A. Rendell, *A practical approach to feature selection*. 1992.
- [68] M. Robnik-Šikonja and I. Kononenko, “Theoretical and Empirical Analysis of ReliefF and RReliefF,” *Mach. Learn.*, vol. 53, no. 1, pp. 23–69, 2003.
- [69] G. Ji, Z. Yang, and W. You, “PLS-Based Gene Selection and Identification of Tumor-Specific Genes,” vol. 41, no. 6, pp. 830–841, 2011.
- [70] R. C. Holte, “Very Simple Classification Rules Perform Well on Most Commonly Used Datasets,” *Mach. Learn.*, vol. 11, no. 1988, pp. 63–91, 1993.
- [71] H. Liu and R. Setiono, “SCALABLE FEATURE SELECTION FOR LARGE SIZED DATABASES.”
- [72] A. Boublenza and M. A. Chikh, “Filtre khi2 pour la sélection des gènes,” in *3ème journée doctorale de génie biomédicale*, 2013.
- [73] Q. McNemar, “Note on the sampling error of the difference between correlated proportions or percentages,” *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.
- [74] C. Rouchouze, “fusion de données Fusion de données exemples Défense et axes de recherche Data Fusion : Some Examples in Defence Area,” *Trait. du Signal*, vol. 11, no. 6, 1994.
- [75] A. N. Steinberg, C. L. Bowman, and F. E. White, “Revisions to the JDL data fusion model,” 1999.
- [76] D. L. Hall, *Mathematical Techniques in Multisensor Data Fusion*, Artech Hou. Norwood, MA, USA: Artech House, Inc., 1992.
- [77] J. Tang, S. Alelyani, and H. Liu, “Feature Selection for Classification: A Review,”

- Data Classif. Algorithms Appl.*, pp. 37–64, 2014.
- [78] L. A. Cotfas, C. Delcea, I. Roxin, and R. Paun, “New Trends in Intelligent Information and Database Systems,” *Stud. Comput. Intell.*, vol. 598, pp. 131–139, 2015.
- [79] A. Martin, “La fusion d’informations,” *Polycopié de cours ENSIETA-Réf*, vol. 21, no. 2, pp. 1–25, 2005.
- [80] D. Ruta, “Classifier diversity in combined pattern recognition systems,” University of Paisley, Scotland, 2003.
- [81] B. V Dasarathy, *Decision fusion*, vol. 1994. IEEE Computer Society Press, 1994.
- [82] J. Kittler, “Combining classifiers: A theoretical framework,” *Pattern Anal. Appl.*, vol. 1, no. 1, pp. 18–27, 1998.
- [83] I. Bloch, “Fusion d’informations numériques : panorama méthodologique,” *Journées Natl. la Rech. en Robot.*, 2005.
- [84] L. Wald, “Some terms of reference in data fusion,” *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 3, pp. 1190–1193, 1999.
- [85] J. Yang, J. Yang, D. Zhang, and J. Lu, “Feature fusion: parallel strategy vs. serial strategy,” *Pattern Recognit.*, vol. 36, no. 6, pp. 1369–1381, Jun. 2003.
- [86] J. Kittler, F. Roli, and I. MCS (Workshop) (1st : 2000 : Cagliari, *Multiple classifier systems : first international workshop, MCS 2000, Cagliari, Italy, June 21-23, 2000 : proceedings*. Springer, 2000.
- [87] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, A Wiley-In., vol. 47, no. 4. A Wiley-Interscience publication: A Wiley-Interscience publication, 2005.
- [88] P. T. Higgins, A. H. Choi, Y. Chen, J.-C. Fondeur, P. Li, J. Tian, X. Yang, S. Zhou, D. Day, C. Busch, G. Canon, M. Swayze, C. J. Tilton, D. Setlak, E. Tabassi, P. Grother, E. Tabassi, P. Grother, D. J. Buettner, G. Cannon, P. Statham, A. Yamada, G. Parziale, A. K. Jain, K. Nandakumar, F. Deravi, J. Larmouth, J. L. Wayman, R. McIver, A. Adler, S. Schuckers, A. Ross, and A. K. Jain, “Base Classifier,” in *Encyclopedia of Biometrics*, Boston, MA: Springer US, 2009, pp. 60–60.
- [89] P. Mordohai, G. Medioni, P. Fua, A. Ross, J. Soh, F. Deravi, A. Triglia, A. Bazin, F. Roli, J. Bigun, B. Roui-Abidi, and M. Abidi, “Multiple Classifier Systems,” in *Encyclopedia of Biometrics*, Boston, MA: Springer US, 2009, pp. 981–986.
- [90] T. G. Dietterich, *Multiple classifier systems*, vol. 1857. 2000.
- [91] P. Yang, Y. Hwa Yang, B. B. Zhou, and A. Y. Zomaya, “A Review of Ensemble Methods in Bioinformatics,” *Curr. Bioinform.*, vol. 5, no. 4, pp. 296–308, 2010.
- [92] A. K. Jain, R. P. W. Duin, and J. Mao, “Statistical Pattern Recognition : A Review,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–38, 2000.
- [93] L. I. Kuncheva, “Switching between selection and fusion in combining classifiers: an experiment,” *IEEE Trans. Syst. Man. Cybern. B. Cybern.*, vol. 32, no. 2, pp. 146–56, Jan. 2002.
- [94] R. Matteo and V. Giorgio, *Ensemble methods: a review*. 2001.
- [95] H. P. Young, “Condorcet’s theory of voting,” *Am. Polit. Sci. Rev.*, vol. 82, no. 4, pp. 1231–1244, 1988.
- [96] L. Lam and C. Y. Suen, “A theoretical analysis of the application of majority voting to pattern recognition,” in *Pattern Recognition, 1994. Vol. 2-Conference B: Computer Vision & Image Processing., Proceedings of the 12th IAPR International. Conference on*, 1994, vol. 2, pp. 418–420.
- [97] B. Achermann and H. Bunke, *Combination of classifiers on the decision level for*

- face recognition*, Citeseer. Citeseer: Citeseer, 1996.
- [98] R. Vilalta and Y. Drissi, “A perspective view and survey of meta-learning,” *Artif. Intell. Rev.*, vol. 18, no. 2, pp. 77–95, 2002.
- [99] D. Bahler and L. Navarro, “Methods for combining heterogeneous sets of classifiers,” in *17th Natl. Conf. on Artificial Intelligence (AAAI), Workshop on New Research Problems for Machine Learning*, 2000.
- [100] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, “Machine learning: A review of classification and combining techniques,” *Artif. Intell. Rev.*, vol. 26, no. 3, pp. 159–190, 2006.
- [101] M. Sewell, “Ensemble Learning,” *Ensemble Learn.*, vol. 4, pp. 1–34, 2012.
- [102] L. Breiman, “Bagging predictors,” *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
- [103] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” in *European conference on computational learning theory*, 1995, pp. 23–37.
- [104] Y. Freund, R. E. Schapire, and others, “Experiments with a new boosting algorithm,” in *Icml*, 1996, vol. 96, pp. 148–156.
- [105] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [106] R. Maclin, J. W. Shavlik, and others, “Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks,” in *IJCAI*, 1995, pp. 524–531.
- [107] C. Science and S. Engineering, “A Review of Ensemble Technique for Improving Majority Voting for Classifier,” vol. 3, no. 1, pp. 177–180, 2013.
- [108] K. M. Ting and I. H. Witten, “Issues in stacked generalization,” *J. Artif. Intell. Res.(JAIR)*, vol. 10, pp. 271–289, 1999.
- [109] J. Gama and P. Brazdil, “Cascade generalization,” *Mach. Learn.*, vol. 41, no. 3, pp. 315–343, 2000.
- [110] S. Džeroski and B. Ženko, “Is combining classifiers with stacking better than selecting the best one?,” *Mach. Learn.*, vol. 54, no. 3, pp. 255–273, 2004.
- [111] L. I. Kuncheva, “A theoretical study on six classifier fusion strategies,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 281–286, 2002.
- [112] I. Mendialdua, J. M. Martínez-Otzeta, I. Rodríguez-Rodríguez, T. Ruiz-Vazquez, and B. Sierra, “Dynamic selection of the best base classifier in One versus One,” *Knowledge-Based Syst.*, vol. 85, pp. 298–306, 2015.
- [113] I. Mendialdua, A. Arruti, E. Jauregi, E. Lazkano, and B. Sierra, “Classifier Subset Selection to construct multi-classifiers by means of estimation of distribution algorithms,” *Neurocomputing*, vol. 157, pp. 46–60, 2015.
- [114] S. Kotsiantis and P. Pintelas, “Recent advances in clustering: A brief survey,” *WSEAS Trans. Inf. Sci. Appl.*, vol. 1, no. 1, pp. 73–81, 2004.
- [115] A. M. P. Canuto, M. C. C. Abreu, L. de Melo Oliveira, J. C. Xavier, and A. de M. Santos, “Investigating the influence of the choice of the ensemble members in accuracy and diversity of selection-based and fusion-based methods for ensembles,” *Pattern Recognit. Lett.*, vol. 28, no. 4, pp. 472–486, 2007.
- [116] L. I. Kuncheva, “Diversity in multiple classifier systems,” *Inf. fusion*, vol. 6, no. 1, pp. 3–4, 2005.
- [117] D. Ruta and B. Gabrys, “Classifier selection for majority voting,” *Inf. Fusion*, vol. 6, no. 1, pp. 63–81, 2005.
- [118] D. Partridge and W. B. Yates, “Engineering multiversion neural-net systems,” *Neural Comput.*, vol. 8, no. 4, pp. 869–893, 1996.

- [119] F. Roli and G. Giacinto, "Design of multiple classifier systems," *Ser. Mach. Percept. Artif. Intell.*, vol. 47, pp. 199–226, 2002.
- [120] L. I. Kuncheva, C. J. Whitaker, C. A. Shipp, and R. P. W. Duin, "Is independence good for combining classifiers?," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, 2000, vol. 2, pp. 168–171.
- [121] M. Petrakos, J. A. Benediktsson, and I. Kanellopoulos, "The effect of classifier agreement on the accuracy of the combined classifier in decision level fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 11, pp. 2539–2546, 2001.
- [122] L. I. Kuncheva, "Switching between selection and fusion in combining classifiers: an experiment," *IEEE Trans. Syst. Man, Cybern. Part B*, vol. 32, no. 2, pp. 146–156, 2002.
- [123] G. U. Yule, "On the Association of Attributes in Statistics: With Illustrations from the Material of the Childhood Society, &c," *Philos. Trans. R. Soc. London A Math. Phys. Eng. Sci.*, vol. 194, no. 252–261, pp. 257–319, 1900.
- [124] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Mach. Learn.*, vol. 51, no. 2, pp. 181–207, 2003.
- [125] D. B. Skalak, "The sources of increased accuracy for two proposed boosting algorithms," in *Proc. American Association for Artificial Intelligence, AAAI-96, Integrating Multiple Learned Models Workshop*, 1996, vol. 1129, p. 1133.
- [126] V. Di Lecce, G. Dimauro, A. Guerriero, S. Impedovo, G. Pirlo, and A. Salzo, "Classifier combination: the role of a-priori knowledge," in *IWFHR*, 2000, vol. 7, pp. 143–152.
- [127] J. Cohen, "Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit.," *Psychol. Bull.*, vol. 70, no. 4, p. 213, 1968.
- [128] M. Aksela, "Comparison of classifier selection methods for improving committee performance," in *International Workshop on Multiple Classifier Systems*, 2003, pp. 84–93.
- [129] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, pp. 993–1001, 1990.
- [130] D. Ruta and B. Gabrys, "Classifier Selection for Majority Voting," *Inf. Fusion*, vol. 5, no. 1, pp. 63–81, 2005.
- [131] U. R. Acharya, K. P. Joseph, N. Kannathal, C. M. Lim, and J. S. Suri, "Heart rate variability: a review," *Med. Biol. Eng. Comput.*, vol. 44, no. 12, pp. 1031–1051, 2006.
- [132] R. J. Martis, U. R. Acharya, and H. Adeli, "Current methods in electrocardiogram characterization.," *Comput. Biol. Med.*, vol. 48, pp. 133–49, May 2014.
- [133] S. M. Jadhav, S. L. Nalbalwar, and A. A. Ghatol, "MODULAR NEURAL NETWORK BASED ARRHYTHMIA CLASSIFICATION SYSTEM USING ECG SIGNAL DATA," *Int. J. Inf. Technol. Knowl. Manag.*, vol. 4, no. 1, pp. 205–209, 2011.
- [134] S. Jadhav, S. Nalbalwar, and A. Ghatol, "Performance Evaluation of Generalized Feedforward Neural Network Based ECG Arrhythmia Classifier," *IJCSI Int. J. Comput. Sci. Issues*, vol. 9, no. 4, pp. 379–384, 2012.
- [135] E. Yilmaz, "An Expert System Based on Fisher Score and LS-SVM for Cardiac Arrhythmia Diagnosis," *Comput. Math. Methods Med.*, 2013.
- [136] K. Polat, S. Şahan, and S. Güneş, "A new method to medical diagnosis: Artificial immune recognition system (AIRS) with fuzzy weighted pre-processing and

- application to ECG arrhythmia,” *Expert Syst. Appl.*, vol. 31, no. 2, pp. 264–269, Aug. 2006.
- [137] T. Soman and P. Bobbie, “Classification of arrhythmia using machine learning techniques,” *WSEAS Trans. Comput.*, 2005.
- [138] C. O. Sakar, O. Kursun, H. Seker, F. Gurgen, N. Aydin, and O. Favorov, “Combining multiple views : Case studies on protein and arrhythmia features,” *Eng. Appl. Artif. Intell.*, pp. 1–7, 2013.
- [139] A. Ozçift, “Random forests ensemble classifier trained with data resampling strategy to improve cardiac arrhythmia diagnosis,” *Comput. Biol. Med.*, vol. 41, no. 5, pp. 265–71, May 2011.
- [140] M. Piao, Y. Piao, H. Shon, J. Bae, and K. Ryu, “Evolutional Diagnostic Rules Mining for Heart Disease Classification Using ECG Signal Data,” *Adv. Control Commun.*, pp. 673–680, 2012.
- [141] A. Jovic, K. Brkic, and N. Bogunovic, “Decision Tree Ensembles in Biomedical Time-Series Classification,” pp. 408–417, 2012.
- [142] Z. Huo, “Procedia Engineering Experimental Comparison of Six Fixed Classifier Fusion Rules,” in *2011 International Conference on Power Electronics and Engineering Application*, 2011, pp. 429–433.
- [143] R. Ranawana and V. Palade, “Multi-classifier systems: Review and a roadmap for developers,” *Int. J. Hybrid Intell. Syst.*, vol. 3, no. 1, pp. 35–61, 2006.
- [144] R. Burduk and K. Walkowiak, “Static Classifier Selection with Interval Weights of Base Classifiers,” in *Intelligent Information and Database Systems: 7th Asian Conference, ACIIDS 2015, Bali, Indonesia, March 23-25, 2015, Proceedings, Part I*, N. T. Nguyen, B. Trawiński, and R. Kosala, Eds. Cham: Springer International Publishing, 2015, pp. 494–502.
- [145] G. Giacinto and F. Roli, “Dynamic classifier selection based on multiple classifier behaviour,” *Pattern Recognit.*, vol. 34, pp. 1879–1881, 2001.
- [146] P. R. Cavalin, R. Sabourin, and C. Y. Suen, “Dynamic selection approaches for multiple classifier systems,” *Neural Comput. Appl.*, vol. 22, no. 3–4, pp. 673–688, Sep. 2011.
- [147] T. Woloszynski and M. Kurzynski, “A probabilistic model of classifier competence for dynamic ensemble selection,” *Pattern Recognit.*, vol. 44, no. 10, pp. 2656–2668, 2011.
- [148] K. Jackowski and M. Wozniak, “Method of classifier selection using the genetic approach,” *Expert Syst.*, vol. 27, no. 2, pp. 114–128, 2010.
- [149] F. Roli and G. L. Marcialis, “An experimental comparison of fixed and trained fusion rules for crisp classifier outputs.”
- [150] G. I. Webb, “Multiboosting: A technique for combining boosting and wagging,” *Mach. Learn.*, vol. 40, no. 2, pp. 159–196, 2000.
- [151] T. K. Ho, “The random subspace method for constructing decision forests,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, 1998.
- [152] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, “Rotation forest: A new classifier ensemble method,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1619–1630, 2006.
- [153] P. Melville and R. J. Mooney, “Constructing Diverse Classifier Ensembles using Artificial Training Examples,” no. August, pp. 505–510, 2003.
- [154] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V D’Amico, J. P. Richie, and others, “Gene expression correlates of

- clinical prostate cancer behavior,” *Cancer Cell*, vol. 1, no. 2, pp. 203–209, 2002.
- [155] T. Speed, *Statistical analysis of gene expression microarray data*. CRC Press, 2003.
- [156] D. Stekel, *Microarray bioinformatics*. Cambridge University Press, 2003.
- [157] T. S. Furey, N. Cristianini, N. Duffy, and W. David, “Support vector machine classification and validation of cancer tissue samples using microarray expression data,” *Bioinformatics, Oxford Univ. Press*, vol. 16, no. 10, pp. 906–914, 2000.
- [158] E. M. Southern, “DNA microarrays,” *DNA Arrays Methods Protoc.*, pp. 1–15, 2001.
- [159] V. Bolón-Canedo, N. Sánchez-Marroño, and A. Alonso-Betanzos, “An ensemble of filters and classifiers for microarray data classification,” *Pattern Recognit.*, vol. 45, no. 1, pp. 531–539, 2012.
- [160] M. Dettling, “BagBoosting for tumor classification with gene expression data,” *Bioinformatics*, vol. 20, no. 18, pp. 3583–3593, 2004.
- [161] D. P. and P. P. B. Jose M. Cadenas, M. Carmen Garrido, Raquel Martínez, “Gene Priorization for Tumor Classification Using an Embedded Method,” *Comput. Intell.*, no. 808, pp. 101–118, 2016.
- [162] K. Moorthy, M. Saberi, B. Mohamad, and S. Deris, “Multiple Gene Sets for Cancer Classification Using Gene Range Selection Based on Random Forest,” *Springer-Verlag Berlin Heidelb.*, pp. 385–393, 2013.