UNIVERSITÉ DU QUÉBEC À MONTRÉAL

MODÈLE DE PRÉDICTION DE CONTEXTE PERTINENT POUR LA MALADIE MPOC

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

LOKMAN SALEH

JUIN 2017

REMERCIEMENTS

Avant de commencer, je tiens tout d'abord à remercier mon directeur de recherche, M. Hamid Mcheick, ainsi que mon codirecteur de recherche, M. Hafedh Mili, des professeurs talentueux et des scientifiques passionnés, qui m'ont accompagné tout au long ma recherche. Merci pour vos conseils et critiques et pour la confiance accordée en acceptant d'encadrer ce travail de maîtrise.

Je suis profondément redevable de mes collègues du laboratoire de recherche LATECE, pour leur aide sincère et leurs expériences uniques, particulièrement Imen Benzarti et Choukri Djellali.

Ensuite, du fond de mon cœur, je dois exprimer ma plus profonde gratitude et mon affection à mon père Farzat, qui a financé tout ce travail, à ma mère, mes sœurs et mes frères pour m'avoir apporté un appui indéfectible et des encouragements continus tout au long de mes d'études et du processus de recherche et d'écriture de ce mémoire. Cette réalisation n'aurait pas été possible sans eux. Je vous remercie.

Enfin, je souhaite de remercier profondément toutes les personnes qui m'ont apporté leur aide de près ou de loin et qui ont contribué à mon succès tout au long de cette recherche.

PUBLICATIONS

S.Lokman, M.Hamid, A.Hicham, M.Hafedh, "Context Relevant Prediction Model for COPD domain using Bayesian Belief Network" Accepted to be published, Sensors Journal, MDPI, issue: Context Aware Environments and Applications, 2017.

S.Lokman, M.Hamid, A.Hicham, M.Hafedh, "HCES: Helper Context Engine System to Predict Relevant State of patients in COPD Domain using Naïve Bayesian" Accepted to be published, International Conference on Internet of Things and Machine Learning (IML 2017-ACM), ACM Digital Library. ISBN: 978-1-4503-7, Liverpool, United Kingdom. October 17-18, 2017.

Lokman Saleh, Hamid Mcheick, and Hicham Ajami. Present a tutorial in 5th International conference on Multi-displanary On E-Technologies (MCETECH'2017). Tutorial title: Context-aware applications for healthcare systems. May 17-19, 2017, Ottawa, Canada.

Hamid Mcheick, Hafedh Mili, Mohamed Dbouk, Caroline Gagné, Djamal Rebaine, Marc Gravel, Malak Khries, Achref Charmiti, Mario Leone, Hung-Tien Bui, Ghassan Fadlallah, Hicham Ajami, Lokman Saleh, Ubiquitous and Collaborative Computing (Poster), Proceeding of the 5Th Annual DIVA Workshop, NSERC-DIVA research strategy network: Developing the next generation Intelligent Vehicular Networks and Applications, Eds. A Boukerche & R. De Grande, pp. 372, February 16-18, 2016, Ottawa, Canada.

TABLE DES MATIERES

LISTE DES TABLEAUX xi
LISTE DES FIGURES xiii
RÉSUMÉ xvii
INTRODUCTION 1
0.1 Problématique 2
0.2 Objectifs 5
0.3 Méthodologie de recherche 6
0.4 Organisation du mémoire 8
CHAPITRE I
REVUE DE LITTERATURE DE CONTEXTE ET DES ALGORITHMES D'APPRENTISSAGE 11
1.1 Définition du contexte 11
1.2 Sensibilité du contexte 13
1.3 Modélisation du contexte 14
1.3.1 Modèle Clé-Valeur 15
1.3.2 Modèle Schéma de balisage 16
1.3.3 Modèle logique 16
1.3.4 Modèle orienté objet 17
1.3.5 Modèle de l'ontologie 18
1.4 Phase de prétraitement de l'apprentissage automatique 20
1.4.1 Discrétisation 21
1.4.2 Sélection des attributs pertinents 26
1.4.3 Structure de dépendance entre les attributs 34
1.5 Algorithmes d'apprentissage supervisé 39
1.5.1 Arbre de décision (ID3 et C4.5) 39

1.5.2 Bayésien naïf 43
1.5.3 Réseau bayésien 45
1.6 Conclusion 47
CHAPITRE II
MOTIFS DE TRAITEMENT DE LA MPOC ET LES SYSTÈMES INFORMATIQUES QUI LE SOUTIENNENT 49
2.1 Définition des maladies chroniques et de la MPOC 49
2.1.1 Qu'est-ce que la maladie pulmonaire obstructive chronique (MPOC)? 50
2.2 Effets biologiques et économiques de la MPOC 51
2.3 Définition de l'exacerbation 52
2.3.1 Conséquences des exacerbations 54
2.4 Systèmes informatiques existants pour suivre les exacerbations 55
2.4.1 Systèmes de télésanté 56
2.4.2 Systèmes de traitement automatique de la MPOC 61
2.4.3 Sélection des attributs pertinents de la MPOC 64
2.5 Analyse de travaux reliés et problèmes spécifiques 66
CHAPITRE III
MODÈLE DE SÉLECTION DES ATTRIBUTS PERTINENTS ET DE DÉTECTION DE LA MPOC; COMPARAISON DES ALGORITHMES D'APPRENTISSAGE 71
3.1 Introduction 71
3.2 Modèle proposé pour sélectionner les attributs pertinents et détecter l'exacerbation dans la MPOC 73
3.3 Acquisition de la base d'apprentissage 75
3.4 Comparaison des modèles de représentation de contexte 76
3.4.1 Choix de la représentation de contexte 76
3.4.2 Description de l'ontologie de la MPOC 79
3.5 Métriques d'évaluations dans l'apprentissage automatique 83
3.6 Sélection des algorithmes et résultats obtenus 86
3.6.4 Configuration de Weka 87
3.6.5 Comparaison des algorithmes de sélection des attributs pertinents 90
3.6.6 Comparaison et utilité des méthodes de discrétisation 92

3.6.7 Comparaison des méthodes de discrétisation et de sélection des attributs pertinents appliquées ensemble 94
3.6.8 Comparaison des algorithmes de dépendance 100
3.7 Conclusion 102
CHAPITRE IV
DEVELOPEMENT D'UNE APPLICATION CONTEXTUELLE : INTÉGRATION DE DIFFÉRENTES ALGORITHMES DANS NOTRE OUTIL DE PRÉDICTION, ÉTUDE DE CAS ET PRÉSENTATION DES RÉSULTATS OBTENUS 105
4.1 Introduction 105
4.2 Étude de cas du modèle de prédiction proposé 106
4.3 Technologies utilisées : Netica-Java, NetBeans et Weka 108
4.4 Conception et implémentation de l'application contextuelle 109
4.5 Application de l'heuristique Gain-Ratio pour arranger les attributs pertinents 114
4.6 Validation du modèle proposé et extension de la méthode Wrappers : WrappersPlus 117
4.6.1 Validation générale du modèle proposé 121
4.6.2 Résultat de l'application de *WrappersPlus* sur huit différentes bases d'apprentissage 121
CONCLUSION 123
APPENDICE A
EXEMPLES PRATIQUES : LES ALGORITHMES D'APPRENTISSAGES 127
APPENDICE B
L'ÉLARGISSEMENT D'ONTOLOGIE 137
APPENDICE C
CODE D'IMPLÉMENTATION D'APPLICATION CONTEXTUELLE 141
APPENDICE D
RÉSULTATS DE WEKA 149
BIBLIOGRAPHIE 159

LISTE DES TABLEAUX

Tableau Page

1.1 Avantages et désavantages de la discrétisation 22

3.1 Comparaison entre les approches de modélisation du contexte (Strang et Linnhoff-Popien, 2004) 77

3.2 Comparaison des différents algorithmes d'apprentissage avec la configuration par défaut de Weka 89

3.3 Comparaison des différents algorithmes d'apprentissage en appliquant les méthodes *Filters* et *Wrappers* 91

3.4 Influence de la discrétisation sur les classificateurs 93

3.5 Résultats obtenus par la métrique AUROC en appliquant la sélection d'attributs pertinents qui précède la discrétisation 96

3.6 Résultats obtenus par la métrique AUROC en appliquant la discrétisation qui précède la sélection d'attributs pertinents 98

3.7 Comparaison entre le réseau bayésien et le naïf bayésien en apprenant le réseau de croyance du réseau bayésien à partir de la base d'apprentissage 101

4.1 Variation de la précision de prédiction, en fonction du nombre d'attributs utilisé, en se basant sur l'ordre de la figure 4.8 115

4.2 *Wrappers* par rapport notre proposition *WrappersPlus*, en utilisant les algorithmes obtenus pour fournir un modèle MPAR 120

LISTE DES FIGURES

Figure Page

1.1 Éléments d'une balise XML 16
1.2 Sélecteur d'attributs *Wrappers* (Cornuéjols, 2006). 29
1.3 Sélecteur d'attributs *Filters* (Cornuéjols, 2006). 30
1.4 Tree Augmented Naïve Bayes (TAN) basé sur *Naïve Bayes (NB)* (Gama et Porto, 2008) 38
1.5 Construction de l'arbre de décision 41
1.6 Réseau de croyance correspondant à la classification naïve bayésienne 44
2.1 Voies aériennes inférieures (coupe intra-thoracique) (Busson, 2014) 50
2.2 *Obstructive Chronic Bronchitis and/or Emphysema* (McGill, 2016) 51
2.3 Système de surveillance des patients atteints de MPOC (Maiolo *et al.*, 2003) 57
2.4 *Health buddy (HB) device* 58
2.5 Surveillance du taux de respiration avant l'hospitalisation pour cause d'exacerbation (Yañez *et al.*, 2012) 62
2.6 Avis d'expert pour construire le réseau bayésien 63
2.7 Modèle prédictif basé sur le réseau bayésien, K2 et *Markov Blanket* (Himes *et al.*, 2009). 65
3.1 Modèle de sélection des attributs pertinents et de détection des exacerbations 73
3.2 Échantillon de la base d'apprentissage en format Excel 76

3.3 Ontologie décrivant toutes les entités possibles d'une application qui détecte l'exacerbation dans la MPOC (« voir appendice B») 80
3.4 Interface du *Protégé* pendant la création de l'ontologie MPOC 82
3.5 Interface du site Web *VOWL* permettant de visualiser l'ontologie de la MPOC 83
3.6 Matrice de confusion pour la classification binaire 84
3.7 Weka utilise par défaut le *Cross Validation* stratifiée 86
3.8 Structure du réseau bayésien réalisée avec l'algorithme K2, qui suppose que le nombre de parents de chaque attribut égal à 1, par défaut 89
3.9 Les algorithmes obtenus pour fournir un modèle de prédiction performant, autonome et raffiné (MPAR) 101
3.10 *Receiver Operating Characteristic (ROC)*, courbe correspondant au modèle final de la prédiction de l'exacerbation MPOC. AUROC = 81.5 %. 102
4.1 Étude de cas : les six étapes du modèle proposé, pour fournir une application contextuelle performante, autonome, raffiné et efficace (voir section 2.5). 107
4.2 NeticaJ.jar est intégré avec NetBeans dans notre projet 109
4.3 Modèle de classe en utilisant UML, pour implémenter le réseau bayésien. 110
4.4 Discrétisation *Fayyad & Irani's MDL* sur les 17 attributs 111
4.5 Sélection des attributs pertinents par Weka (*Wrapper-BestFirst*). 112
4.6 Structure du réseau bayésien, en utilisant la méthode TAN pour détecter l'exacerbation de la MPOC 112
4.7 Méthode pour trouver la meilleure *CutOff* 113
4.8 Arrangement des attributs pertinents, basé sur la mesure *GainRatio* 115
4.9 Interface de base de notre application présentant les huit symptômes primaires 116
4.10 Interface de notre application, avec les symptômes secondaires (partie droite de la figure) 116

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

Ac	Application contextuelle
ATS	*American Thoracic Society*
AUROC	*Area Under Receiver Operating Characteristic*
CFSsubsetEval	*Correlation based Feature Selection*
CPT	*Conditional probability table*
EFD	*Equal Frequency Discretization*
EWD	*Equal Width Discretization*
GR	*Gain Ratio*
IA	*Intelligence artificielle*
MPOC	Maladie pulmonaire obstructive chronique
NASA	National Aeronautics and Space Administration
OWL	*Web Ontology Language*
TAN	*Tree Augmented Naive Bayes*

RÉSUMÉ

En informatique, les applications contextuelles (AC) permettent la mise en place d'approches prometteuses et efficaces pour le traitement et le suivi de divers éléments contextuels. Dans le domaine médical, on utilise de plus en plus les AC pour soutenir le personnel médical appelé à surveiller la santé de patients, prédire leur état et prendre des décisions en conséquence. Ces tâches permettent d'améliorer l'état de santé des personnes atteintes de maladies qui nécessitent un traitement efficace et une surveillance continue afin d'éviter l'hospitalisation, ici les personnes atteintes de maladies pulmonaires obstructives chroniques (MPOC).

Nous avons utilisé les MPOC comme domaine d'application pour des raisons à la fois médicales et scientifiques. Environ 3.5 millions de Canadiens sont atteints de cette maladie pulmonaire, ce qui entraîne des coûts d'hospitalisation élevés. Aussi, il n'y a pas à ce jour de traitement curatif de ces maladies; le traitement actuel consiste à éviter les complications (crise pulmonaire ou exacerbation) par des systèmes logiciels présentant une efficacité limitée, puisqu'ils se limitent aux soins généraux de la MPOC, ne sont pas très précis et nécessitent la participation de spécialistes médicaux pour leur construction. En plus, ces logiciels identifient les attributs pertinents d'un patient atteint par cette maladie, par des approches inefficaces, ce qui donne souvent une réponse imprécise et incertaine.

Nous avons donc conçu et validé un modèle et un outil logiciel performant et autonome qui aide les patients et le personnel médical à prendre des décisions rapides et à mieux prédire les risques de complication de la MPOC en identifiant efficacement par anticipation et en ordre les attributs les plus pertinents. En premier lieu, cette recherche se concentre sur la compréhension du contexte, en proposant une ontologie qui décrit un système intelligent pour détecter l'exacerbation. En second lieu, nous comparons différents algorithmes de sélection d'attributs pertinents, de discrétisation et de raisonnement de l'état courant d'un patient selon ses données réelles. Cette comparaison permet de choisir la bonne combinaison d'algorithmes en se basant sur des métriques qui mesurent la performance du modèle prédictif choisi. En troisième lieu, sur la base de ces algorithmes identifiés, nous proposons un modèle prédictif appliqué à la MPOC, qui permet d'en identifier les attributs pertinents et de prédire les exacerbations d'un patient en temps réel. En outre, ce modèle est validé par huit différentes bases d'apprentissage (*Cancer, Spectfheart, etc.)*. À la fin, l'algorithme *Wrappers-BestFirst* est amélioré afin de minimiser le nombre d'attributs pertinents.

Mots-clés : applications contextuelles, MPOC, sensibilité au contexte, incertitude, apprentissage automatique, exacerbation, sélection des attributs pertinents.

INTRODUCTION

Les systèmes informatiques ubiquitaires ou pervasifs (SIP) constituent une nouvelle classe de systèmes d'information qui accommode les besoins et les volontés des utilisateurs en utilisant les technologies de l'information. Les SIP diffèrent des systèmes d'information de bureau (SIB): ils englobent un environnement complexe, dynamique et composé de plusieurs objets (par exemple : les capteurs), qui leur permettent de recevoir des quantités beaucoup plus importantes d'informations contextuelles que l'entrée d'utilisateur simple. Cette complexité des nouveaux systèmes implique l'utilisation de méthodes automatisées pour analyser et gérer la masse des données captées.

Les chercheurs du domaine de l'apprentissage automatique ont développé des algorithmes de raisonnement qui détectent automatiquement des modèles (*patterns*) dans les données. Ces modèles sont utilisés pour découvrir de nouvelles connaissances. Parmi les caractéristiques communes à ces modèles d'apprentissage, on peut citer le fait qu'ils ne se limitent pas à un domaine précis. Dans ce contexte, les SIP peuvent profiter des algorithmes d'apprentissage pour créer leur propre modèle dans différents domaines d'application, ce qui les rend capables de prendre une décision à chaque nouvelle situation, ce qu'on appelle « adaptation dynamique du système selon son contexte », ou *sensibilité au contexte*.

La sensibilité au contexte permet de suivre les habitudes des usagers et d'y répondre de façon dynamique selon les changements de contexte (Li *et al.*, 2015). Elle implique aussi de nouveaux défis et en ce sens, des mécanismes sont utilisés pour réduire la complexité de l'adaptation, par exemple quand les données observées sont imparfaites,

incomplètes, ambiguës, etc. (Mcheick *et al.*, 2015). Ces mécanismes reposent aussi sur les méthodes d'apprentissage automatique qui traitent l'incertitude des données capturées, comme le réseau bayésien.

La sensibilité au contexte peut être exploitée par une application contextuelle (AC). Cette dernière permet la mise en place d'approches prometteuses permettant de réaliser le traitement et le suivi de divers éléments contextuels (Najar, 2014) (Dey et Häkkilä, 2008). Dans le domaine médical, ces tâches permettent d'améliorer l'état de santé des personnes atteintes de maladies qui nécessitent une surveillance continue et de leur éviter l'hospitalisation. C'est le cas, par exemple, des personnes atteintes de maladies pulmonaires obstructives chroniques (MPOC), qui ne peuvent pas sortir à l'extérieur par temps humide. Cependant, des défis importants, qui sont énumérés à la prochaine section, restent à résoudre afin d'éviter plusieurs lacunes lors de la construction d'une application contextuelle.

Les sections suivantes abordent la problématique, les objectifs, la méthodologie de recherche et l'organisation de ce mémoire.

0.1 Problématique

Fournir un soutien assisté par ordinateur aux patients atteints de MPOC et au personnel médical (médecins, infirmières, etc.) leur permet de détecter une crise pulmonaire, comporte de multiples avantages. Cela peut entraîner une diminution du risque de complications dues aux MPOC et du coût des soins de santé liés à l'hospitalisation, alléger la charge de travail du personnel soignant, etc. Toutefois, le processus de surveillance d'un patient assisté par ordinateur présente plusieurs défis en raison de sa complexité. En effet, il faut se demander: Comment les données pertinentes des patients

sont-elles sélectionnées et interprétées? Quelle infrastructure logique est la plus performante? Comment l'application contextuelle évoluera-t-elle dans le futur?

Ces questions se détaillent comme suit :

1) Comprendre et identifier les éléments pertinents du contexte en général et celui du domaine médical en particulier reste encore un défi important (Montserrat-Capdevila *et al.*, 2016). En outre, la sélection des attributs pertinents est un sujet de recherche qui appartient au domaine de l'apprentissage automatique. Dans ce cadre-là, un rapprochement entre l'analyse de données biologiques et l'apprentissage automatique est une perspective intéressante à explorer (Slimani *et al.*, 2014). De plus, la recherche actuelle permet rarement au concepteur d'une AC de déterminer ce qui constitue un contexte pertinent (Khattak *et al.*, 2014) (Mostefaoui *et al.*, 2004). Dans le cadre de cette recherche, nous voulons identifier les attributs pertinents qui indiquent la crise pulmonaire ou l'exacerbation de la personne atteinte de MPOC en utilisant des méthodes issues du domaine de l'apprentissage automatique. Cette sélection d'attributs permet de réaliser un outil de prédiction raffiné, dans le sens d'une interaction facilitée avec le patient. En outre, dans le cas d'urgence les attributs pertinents peuvent être plus qu'on puisse les observer dans un court laps de temps. Dans ce contexte, l'ordonnancement de ces attributs va garantir une précision minimale, à la place de faire une observation aléatoire et non complete qui perd de la précision de prédiction.

2) En outre, comme les causes d'exacerbation dans les MPOC ne sont pas connues (Lareau *et al.*, 2014) et que chaque personne en présente des signes et symptômes légèrement différents (Van der Heijden *et al.*, 2014), la prédiction d'exacerbations est traitée dans un cadre d'incertitude, où le traitement logique *"If-Else"* ne fonctionne pas (Parsons et Kubat, 1994). Par ailleurs, l'incertitude est une caractéristique inévitable de l'application contextuelle, en raison de la

nature des données détectées, qui pourraient être imparfaites, incomplètes, erronées ou ambiguës (Mcheick *et al.*, 2015). Pour ces raisons, la comparaison de plusieurs algorithmes lors de l'apprentissage automatique est indispensable afin de créer une infrastructure logique performante et ainsi obtenir une bonne précision de prédiction.

3) Enfin, l'objectif de la plupart des systèmes ubiquitaires et contextuels est d'automatiser le traitement des MPOC. Cependant, au fondement de l'instruction de ces systèmes, nous remarquons l'intervention marquée des médecins. Nous pensons que cette intervention peut mener à la génération d'un système rigide, qui ne sera pas capable d'évoluer dans le futur. Par exemple, dans le cas du réseau bayésien, la dépendance entre les symptômes peut être constatée par des experts médicaux (Van der Heijden *et al.*, 2013), mais la structure (réseau de croyance) à base d'experts peut-être différente d'un expert à l'autre et difficile à faire évoluer en l'absence de ce dernier (même chose pour la discrétisation et la sélection d'attributs pertinents). C'est pourquoi nous avons conçu et validé, par l'usage de l'apprentissage automatique, un système de prédiction autonome et performant ne nécessitant pas l'intervention d'experts, ce qui est en soi une contribution importante à la recherche portant sur le traitement préventif des MPOC.

Pour formuler notre objectif, nous soulevons les questions suivantes, qui seront adressées dans ce mémoire afin de développer notre modèle de prédiction final:

- Comment peut-on concevoir une ontologie qui explique les scénarios de patients MPOC pour détecter leur exacerbation en utilisant une application contextuelle ?
- Comment peut-on identifier une bonne méthode pour discrétiser les informations continues dans la base de données de MPOC au lieu de référer, pour ce faire, à des experts?

- Quel est l'algorithme qui permet de mieux sélectionner les attributs pertinents de cette maladie?
- Quel est le meilleur algorithme, dans le cas du réseau bayésien, permettant de réaliser la dépendance entre les attributs prédictifs et ainsi de détecter les exacerbations de la MPOC?
- Quel est l'algorithme d'apprentissage le plus performant pour prédire les exacerbations de la MPOC?
- Comment peut-on améliorer la précision de la prédiction dans les MPOC?
- Comment peut-on arranger les attributs pertinents pour que le système prédictif puisse commencer par préciser les éléments prédicteurs les plus importants dès le départ?
- Comment peut-on évaluer le modèle prédictif final et quelle est la métrique d'évaluation la plus importante?
- Comment peut-on améliorer l'algorithme de sélection *Wrapper-BestFirst*?

0.2 Objectifs

Notre objectif est de concevoir et de valider un modèle et un outil logiciel autonome et performant permettant aux patients et au personnel médical (médecins ou infirmières) de prendre des décisions rapides en identifiant par anticipation et en ordre les attributs les plus pertinents des MPOC et ainsi, de prédire les crises pulmonaire ou les exacerbations des patients. Essentiellement, nous voulons répondre à la question suivante :

Comment peut-on concevoir et valider un modèle de prédiction autonome, performant, permettant de bien discriminer entre les états d'un patient ?

Pour y parvenir, un modèle prédictif du processus de sélection des attributs pertinents et de détection des exacerbations est conçu et validé. Ce modèle :

i) Réduit les efforts du personnel médical requis pour diagnostiquer l'état de patients par un moyen pouvant être appliqué sur place (clinique, urgence, etc.).

ii) Améliore la précision de diagnostic de l'état de patients dans le cas de suivi. Soulignons que ce diagnostic est offert toujours par le médecin.

iii) Fournit un outil de prédiction autonome capable d'évoluer dans le futur sans l'intervention d'experts médicaux.

0.3 Méthodologie de recherche

Notre démarche globale s'est déroulée selon les phases suivantes :

- Survoler et comprendre les différentes définitions de contexte afin d'adopter la plus pertinente dans le cadre de cette recherche, cela permettra de comprendre le terrain du notre travail et le scénario qu'il faut le prendre en compte dans la première question de notre objective (au chapitre 1).
- Effectuer une revue de la littérature portant sur les modèles de représentation du contexte, tels que le modèle clé-valeur, le schéma de balisage, l'ontologie, etc. Cette revue nous permettra de prouver notre choix d'ontologie (au chapitre 1).
- Effectuer une revue des algorithmes utilisés pour l'apprentissage automatique, comme les algorithmes du raisonnement, de la discrétisation et de la sélection des attributs pertinents, pour les comprendre et les améliorer comme le cas du *Wrappers* (au chapitre 1).

- Survoler le thème des MPOC afin d'en comprendre les influences sur la vie des personnes. Dans le même cadre, nous expliquerons l'exacerbation et ses conséquences pour les patients atteints de MPOC (au chapitre 2).
- Effectuer une revue de la littérature portant sur les systèmes informatiques permettant le suivi des MPOC, cela permettra de détecter les points manquants dans les anciens travaux (au chapitre 2).
- Représenter les entités de notre système prédictif par une ontologie capable de détecter l'exacerbation de la MPOC (au chapitre 3).
- Comparer plusieurs algorithmes utilisés lors des phases de prétraitement et de traitement de l'apprentissage automatique, pour choisir les plus performants selon une métrique de classification (au chapitre 3).
- Sélectionner une combinaison d'ensemble des algorithmes dans l'apprentissage automatique, de façon ordonnée et performante selon notre test, afin que nous puissions proposer un modèle prédictif pouvant être généralisé et appliqué à n'importe quel système de prédiction (au chapitre 3).
- Évaluer le modèle prédictif en utilisant l'outil d'apprentissage Weka (au chapitre 3).
- Implémenter et valider une application contextuelle en Java, dans laquelle nous intégrons les algorithmes choisis à l'étape précédente (au chapitre 4).
- Valider le modèle de prédiction proposé par huit bases d'apprentissage (au chapitre 4).
- Proposer un algorithme *WrappersPlus* comme une extension de l'algorithme de sélection *Wrapper-BestFirst*, qui est validé par huit bases d'apprentissage (au chapitre 4). Notons que nous allons utiliser huit bases d'apprentissage différentes, comme : *Cancer, Spectfheart, etc.*
- Formuler nos conclusions et identifier les perspectives de recherches futures (Conclusion).

0.4 Organisation du mémoire

Le présent mémoire est structuré de la façon suivante:

Premièrement, au chapitre 1, nous étudions les définitions et les modèles de représentation de contexte afin de mieux comprendre ses différents éléments. Ainsi, une étude approfondie des quatre phases du processus de raisonnement (la discrétisation des attributs continus, la sélection des attributs pertinents, la réalisation du réseau de dépendance entre les attributs de données et le raisonnement) est menée.

À travers le chapitre 2, nous décrivons les avantages d'étudier la MPOC, en commençant par sa définition générale et ses influences sur la vie des personnes. Dans le même cadre, nous expliquons l'exacerbation et ses conséquences dangereuses pour les patients atteints de MPOC. Par la suite, nous effectuons un survol général des travaux en informatique qui ont abordé la prédiction des manifestations de la MPOC, en particulier l'exacerbation.

Au chapitre 3, nous proposons une ontologie pour un système intelligent de MPOC. Cette ontologie illustre un scénario pour détecter l'exacerbation par une application contextuelle, en se basant sur les algorithmes de chaque phase mentionnée ci-dessus. Ensuite, ces algorithmes sont comparés entre eux, pour déterminer ceux qui sont les plus performants selon la métrique d'évaluation utilisée AUROC, dans le but de proposer un modèle de prédiction employant des mécanismes différents que ceux utilisés lors d'études précédentes et comportant plusieurs avantages.

Au chapitre 4, nous allons implémenter et valider en Java un outil prédictif performant, autonome permettant de prédire l'exacerbation des personnes atteintes de MPOC en nous basant sur le modèle proposé au chapitre 3. Au même chapitre, nous validons le modèle proposé par huit bases d'apprentissages. Ainsi, en nous basant sur ces bases d'apprentissage, nous nous avançons davantage d'un point de vue algorithmique en

proposant et validant *WrappersPlus,* une amélioration de l'algorithme *Wrapper-BestFirst* permettant de sélectionner un nombre plus restreint d'attributs pertinents tout en conservant une bonne capacité prédictive.

À la fin, nous allons conclure ce mémoire et exposer certaines perspectives prometteuses pour des recherches futures.

CHAPITRE I

REVUE DE LITTERATURE DE CONTEXTE ET DES ALGORITHMES D'APPRENTISSAGE

Comme nous l'avons mentionné à l'introduction, notre objectif est de concevoir et de valider un modèle de prédiction performant et autonome appliqué à une application contextuelle. Ce modèle aide le personnel médical à choisir les attributs pertinents de façon ordonnée et à prédire les exacerbations de la maladie pulmonaire obstructive chronique (MPOC). Ce chapitre fait le survol des trois premières et principales étapes de la construction d'une application contextuelle (AC) en mettant en évidence différentes techniques dont rend compte la littérature scientifique. Les étapes de l'AC qui sont expliquées dans ce chapitre sont : i) la définition de contexte, ii) la modélisation de contexte et iii) le raisonnement. Cette troisième étape représente la partie responsable de la prédiction dans une AC. Compte tenu de son importance, nous avons recensé plusieurs algorithmes utilisés lors des deux phases de prétraitement (discrétisation, sélection d'attributs pertinents, réseau de croyance) et de traitement (arbre de décision, naïf bayésien et réseau bayésien) des données lors de l'apprentissage automatique.

1.1 Définition du contexte

La présence du contexte dans une application contribue à l'amélioration de la compréhension et à l'identification de l'action autour (Kirsch Pinheiro, 2006).

Contrairement à une application de bureau ordinaire, une application appartenant à un environnement ubiquitaire peut changer dynamiquement d'environnement. Cela implique nécessairement la considération du contexte, ce qui permet d'obtenir une adaptabilité dynamique du contexte de l'utilisateur et de mieux répondre à son besoin (Najar *et al.*, 2009).

De nombreux scientifiques se sont interrogés sur la signification réelle de la notion de contexte dans l'environnement de l'informatique ubiquitaire. Proposer une définition claire, aide à formaliser la notion du contexte (Chaari *et al.*, 2005) et à l'utiliser efficacement (Dey, 2001). La définition aide également à orienter l'attention sur l'activité en cours (Brézillon, 2005) et à guider le mécanisme d'adaptation (Najar, 2014).

La première tentative de définition de la notion de contexte a été menée par (Schilit, B. N. et Theimer, 1994). Ces auteurs limitent la définition à la localisation de l'utilisateur, l'identification des personnes et des objets qui les entourent et aux modifications apportées à ces objets. Ces auteurs se sont appliqués à identifier des éléments du contexte au lieu de comprendre sa signification réelle. Cette direction a été empruntée par plusieurs autres chercheurs (Brown *et al.*, 1997), (Ryan *et al.*, 1999).

Par ailleurs, (Pascoe, 1998) (Dey, 2001) et (Kirsch Pinheiro, 2006) ont estimé qu'il est impossible d'énumérer tous les aspects importants ou valables relatifs à toutes les applications utilisées. Ces auteurs ont accordé une grande attention à la notion générale de contexte et lui ont accordé un sens plus opérationnel, permettant de développer des applications ubiquitaires. En particulier, (Dey, 2001) a raffiné les définitions précédentes, en donnant la définition générale du contexte qui suit:

> [...] Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

Cette définition est largement acceptée par les chercheurs, compte tenu de sa généralité par rapport aux exemples précédents et de son insistance sur les entités qui sont pertinentes pour l'interaction entre l'application et l'utilisateur.

Néanmoins, certains auteurs comme (Chaari *et al.*, 2005; Greenberg, 2001) trouvent que cette généralité rend l'identification des éléments du contexte difficile. En outre, Chaari (2005) mentionne que cette définition ne permet pas de faire la distinction entre les données appartenant au contexte et celles appartenant à l'application elle-même. Cette séparation s'avère pourtant essentielle pour assurer la flexibilité lors de la modélisation d'un système sensible au contexte.

De notre point de vue, nous constatons que toutes les définitions du contexte existantes sont soit très générales, ce qui rend la formalisation du contexte très difficile, soit spécifiques à un domaine particulier. Nous proposons donc d'adopter la définition de (Li *et al.*, 2015), pour qui le contexte est: « any piece of information that can represent changes of the circumstances (either static or dynamic). Further, it could be useful for understanding the current situation and predicting potential changes ». Cette définition peut être adaptée au contexte médical, parce qu'elle prend en considération le processus de prédiction. Ce processus est important, car il permet aux patients et au personnel médical de détecter la possibilité que survienne une maladie en général, le risque d'exacerbation dans la MPOC dans le cas de la présente recherche.

1.2 Sensibilité du contexte

Les systèmes sensibles au contexte sont en mesure d'adapter leurs opérations au contexte actuel, ce qui permet d'accroître la convivialité et l'efficacité de l'expérience de l'utilisateur sans son intervention explicite (Baldauf *et al.*, 2007).

Dans la littérature, la première mention de la sensibilité du contexte est due à (Want *et al.*, 1992). Par la suite, le terme a été plus clairement défini par (Schilit, B. N. et Theimer, 1994), qui considère la sensibilité au contexte comme la capacité d'une application à découvrir et à réagir aux changements de l'environnement qui entoure l'utilisateur. Plusieurs ont tenté de définir cette notion. Dans la communauté de l'informatique ubiquitaire, la définition la plus employée est celle de (Dey, 2000), qui, très générale, propose qu'« [u]n système est sensible au contexte s'il utilise le contexte pour fournir des informations et/ou des services pertinents à l'utilisateur, où la pertinence dépend de la tâche de l'utilisateur ».

Cette dernière définition est suffisamment précise pour s'adapter à n'importe quelle application contextuelle (Li *et al.*, 2015), c'est pourquoi nous avons choisi de l'adopter dans le cadre de cette recherche.

1.3 Modélisation du contexte

La modélisation du contexte est la deuxième phase la plus importante, après la définition, dans une application contextuelle. La modélisation comporte un intérêt pratique, puisqu'elle permet de décrire l'environnement et la situation de l'utilisateur (Liu *et al.*, 2011), de réaliser une conception compréhensible et facile du contexte (Kirsch Pinheiro, 2006), de déterminer la capacité d'adaptation du système (Najar *et al.*, 2009), d'assurer et de formaliser la définition du contexte (Najar *et al.*, 2009) et de maintenir et de faire évoluer le système sensible au contexte (Bettini *et al.*, 2010). Selon Dey (Dey, 2001), l'utilisation efficace de la notion de contexte ne se résume pas seulement à sa compréhension, mais s'étend également à sa conception sur un support architectural. En plus, l'approche de modélisation aide à choisir la méthode de raisonnement convenant le mieux au traitement du contexte (Topcu, 2011).

Dans ce contexte, plusieurs approches de modélisation ont été proposées dans différents domaines d'application contextuelle, car il n'y a pas de consensus sur la meilleure approche dans tous les domaines (Moore *et al.*, 2007). Pour cela (Strang et Linnhoff-Popien, 2004) présente six approches principales permettant de construire des modèles contextuels, le *modèle Clé-Valeur,* le *modèle Schéma de balisage*, le *modèle Graphique*, le *modèle Orienté Objet*, le *modèle logique* et le *modèle basé sur l'Ontologie*. Khattak et son équipe de recherche soutenaient en 2014 que la plupart des études antérieures portant sur la modélisation ou la représentation de contexte sont incomplètes (Khattak *et al.*, 2014). Ils ont donc travaillé à trois autres modèles, auxquels (Li *et al.*, 2015) a ajouté, plus tard en 2015, trois nouveaux modèles, pour un total de dix modèles principaux permettant de modéliser le contexte.

Dans les prochaines sections, nous distinguons quatre approches de modélisation qui sont utilisées fréquemment dans la communauté informatique et qui représentent les modèles communs de l'étude (Khattak *et al.*, 2014; Li *et al.*, 2015; Strang et Linnhoff-Popien, 2004) et (Aarab *et al.*, 2016).

1.3.1 Modèle Clé-Valeur

Le modèle de paires <clé,valeur> est la structure de données la plus simple pour la modélisation de l'information contextuelle. Ce modèle a été introduit par (Schilit, B. *et al.*, 1994) pour structurer le contexte sous la forme d'une paire ordonnée (C, V), où '*C*' est la clé qui représente un élément dans l'environnement d'une application et '*V*', la valeur de contexte actuellement capturée pour cet élément.

L'avantage de ce modèle est qu'il est une façon simple et rapide de décrire le contexte en utilisant les chaînes de caractères (*strings*). Cependant, il présente aussi plusieurs inconvénients, comme l'absence d'une norme pour le mettre en œuvre, l'impossibilité

de construire une structure sophistiquée ou de définir les relations entre ses entités et le manque d'extensibilité (ou la réutilisabilité). Pour conclure, ce modèle est considéré utile pour des applications de contexte simples, qui ne nécessitent pas le partage de données (Khattak *et al.*, 2014).

1.3.2 Modèle Schéma de balisage

Le modèle Schéma de balisage est représenté par une structure hiérarchique de balises. Ces derniers stockent les informations de contexte en utilisant des attributs qui peuvent être arbitrairement imbriqués et du contenu qui peut être défini de manière récursive par d'autres balises (Figure 1.1). L'expressivité de ce modèle est accessible par RDF/S et XML (Najar, 2014).

Figure 1.1 : Éléments d'une balise XML

Ce modèle est caractérisé par l'interopérabilité qu'elle permet entre les différents domaines d'application et pour sa capacité de récupérer les données efficacement. D'autre part, il présente plusieurs inconvénients, comme le manque de confidentialité et la difficulté d'établir les relations entre de nombreuses entités du contexte.

Ce modèle fournit un langage plus expressif que le modèle Clé-Valeur, mais il demande plus d'efforts pour maintenir l'aspect dynamique du contexte (Khattak *et al.*, 2014).

1.3.3 Modèle logique

Ce type de modèle est construit en représentant les éléments de contexte, comme des faits atomiques. Les relations entre ces éléments sont réalisées en utilisant des règles ou des expressions basées sur les connecteurs $\wedge$, $\vee$, $\neg$ et $\rightarrow$ (« et », « ou », « non, » et « implique »), ce qu'on appelle la « logique de premier ordre » (Gray et Salber, 2001).

Ce modèle comporte plusieurs avantages: il est riche en termes d'expressions et de raisonnements logiques et les informations de haut niveau peuvent extraire des informations de bas niveau. Par contre, le principal inconvénient de ce modèle est qu'il est construit en fonction d'une application précise, c'est-à-dire que dans la plupart des cas, on ne peut pas le réutiliser. Aussi, il est assez rigide pour supporter l'incertitude.

Ce type de modèle peut convenir aux applications d'intelligence artificielle où un niveau abstrait d'informations (niveau élevé) est nécessaire pour arriver à la connaissance et où on utilise des règles spécifiques.

1.3.4 Modèle orienté objet

Le concept de base du modèle orienté objet est le paradigme de programmation du même nom, qui permet de définir un petit nombre de propriétés, de fonctions et de règles permettant de simplifier la représentation des connaissances dans des domaines et systèmes très complexes (Bouzy et Cazenave, 1997). Ce modèle peut être visualisé en utilisant le langage de modélisation UML.

L'usage de ce modèle comporte plusieurs avantages. D'abord, il est capable de réaliser des relations complexes en se basant sur le langage UML. De plus, il est supporté par le langage de programmation Java et soutient l'encapsulation, la réutilisabilité, l'héritage et le polymorphisme (Khattak *et al.*, 2014), (Li *et al.*, 2015). Ce modèle éprouve cependant quelques difficultés de récupération des données à cause de

l'encapsulation de données; par la suite, l'information ne peut être vue par d'autres applications.

1.3.5 Modèle de l'ontologie

Afin d'obtenir une représentation du contexte présentant une forte sémantique, nous pouvons opter pour le modèle de l'ontologie. En informatique, l'ontologie décrit le sens spécifique de ce qu'on veut dire lorsqu'un mot peut avoir plusieurs significations. Par exemple, si on met la balise *fname* dans le XML (schéma de balisage), on ne saura pas immédiatement le sens de ce mot, qui pourrait référer autant à *family name* qu'à *father name*. Il faut donc définir le sens de *fname* pour le comprendre dans son contexte, mais la définition par une phrase ne fournit pas une signification suffisamment précise et formelle pour l'utiliser dans un programme. À cet effet, l'ontologie va nous permettre de préciser le vrai sens d'un objet. On peut dire que l'ontologie est un dictionnaire qui permet de bien expliquer l'environnement basé sur le langage expressif OWL.

La maturité ou l'expressivité d'ontologie est venue du Web sémantique (OWL, RDFS/OWL), qui traduit le modèle graphique de l'ontologie en un code exploitable et interopérable par un logiciel (Héon, 2014). À cet effet, les ontologies fournissent une description formelle et sémantique des informations de contexte en termes de concepts (idées ou classes), d'objets (instance ou fait) et de propriétés ou de relations (Najar, 2014). Le mot « ontologie » vient de deux mots grecs: *onto* (existence, être réel) et *logia* (science, ou étude de…).

La définition d'ontologie la plus courante et admise dans le milieu de la recherche est celle proposée par (Studer *et al.*, 1998): « an ontology is a formal, explicit specification of a shared conceptualization ». *Conceptualization* réfère au modèle abstrait d'un certain phénomène dans le monde. *Shared* indique la connaissance consensuelle, qui

n'est pas propre à l'opinion d'un individu, mais plutôt validée par un groupe afin que le sens soit clair lorsque nous le partageons. *Explicit* signifie que le type de concept utilisé et les contraintes sur son utilisation sont explicitement définis. Finalement, *formal* indique que l'ontologie doit être lisible par la machine.

Une autre définition récente de l'ontologie est celle proposée par (Van Nguyen *et al.*, 2010). Selon son article, « l'ontologie est une description explicite et formelle des concepts dans un domaine du discours particulier, et elle fournit un vocabulaire pour la représentation des connaissances du domaine et pour décrire ses situations spécifiques ». L'importance de l'ontologie est bien illustrée dans les propriétés d'expressivité, puisqu'elle supporte le mécanisme de raisonnement logique et la réutilisation. Cependant, l'ontologie ne peut pas régler l'incertitude et sa représentation est parfois complexe (Aarab *et al.*, 2016).

Jusqu'à maintenant, nous avons survolé les deux premières phases essentielles de la construction d'une application contextuelle. Premièrement, nous avons effectué un survol des différentes définitions de la notion de contexte et présenté celle, parmi les définitions existantes, qui nous a semblé la mieux applicable à notre démarche. Deuxièmement, nous avons passé en revue un ensemble d'approches de modélisation, afin de choisir celle présentant les propriétés les plus adéquates pour représenter un système intelligent capable d'assurer le suivi des patients atteints de MPOC. Le chapitre III, qui représente notre contribution, détaille les raisons de notre choix et la construction de notre modèle.

Dans ce chapitre (I), la troisième partie, qui est la plus intéressante du processus de développement d'une application contextuelle, consiste à établir des méthodes de raisonnement. Compte tenu de l'importance de l'influence de ces méthodes sur la performance et la capacité adaptative d'une application contextuelle, et considérant l'inquiétude dans le traitement de la MPOC, les méthodes de raisonnement employées doivent être capables de supporter le traitement de données ambiguës, complexes,

floues, incomplètes, continues, etc. Ces données peuvent être observées par un patient ou le personnel médical, ou encore être détectées par des capteurs. Dans ce contexte, nous considérons que la revue de plusieurs types d'algorithmes d'apprentissage automatique peut nous permettre de trouver, par la combinaison de leurs performances, une application contextuelle présentant plusieurs caractéristiques spéciales (autonome, capable de détecter l'exacerbation dans la MPOC avec une grande capacité prédictive, etc.).

1.4 Phase de prétraitement de l'apprentissage automatique

L'apprentissage automatique est un type d'intelligence artificielle (IA) qui se concentre sur le passage automatique des informations à une connaissance, pour fournir des ordinateurs capables d'apprendre sans être explicitement programmés (Murphy, 2012). Dans un autre sens, c'est une manière originale et récente de voir les sciences, car si « les sciences nous aident à comprendre notre environnement, pourquoi ne joueraient-elles pas le même rôle pour les ordinateurs? » (Olivier, 2006). Dès le début, l'apprentissage automatique a été conçu réaliser l'analyse de données médicales, surtout lorsque l'évolution numérique a fourni des moyens (capteurs) peu coûteux permettant de recueillir et de stocker les informations. Par exemple, les algorithmes d'apprentissage sont utiles au médecin lors du diagnostic des patients, afin d'améliorer la vitesse, la précision et la fiabilité de diagnostic, et peuvent également être utilisés pour former des étudiants non spécialistes (Kononenko, 2001).

Cette importance donnée à l'apprentissage automatique, spécifiquement l'apprentissage automatique supervisé dans le domaine médical, nous a incité à utiliser l'apprentissage automatique comme entité de raisonnement de notre application contextuelle. Dans les pages qui suivent, nous mettons en évidence plusieurs algorithmes existants qui sont employés lors des deux phases de prétraitement et de

traitement de données, afin de choisir ceux qui sont utiles à notre application contextuelle en nous basant sur leur rôle (voir le chapitre II) et leur efficacité prédictive (précision) dans le cas de la MPOC (voir le chapitre III).

Dans les prochaines sous-sections, notre revue se concentre sur les algorithmes utilisés lors de la phase du prétraitement de données, à savoir la discrétisation et la sélection d'attributs pertinents. Nous traitons également brièvement de deux algorithmes permettant de construire un réseau de croyances à partir de données dans le cas du réseau bayésien.

1.4.1 Discrétisation

L'apprentissage supervisé ou la classification vise à sélectionner un état parmi plusieurs dans la classe d'attributs. Ce processus de classification peut avoir une base d'apprentissage qui contient des attributs continus ou nominaux. Cependant, la plupart des classificateurs existants ne supportent pas cette hétérogénéité (soit la présence de deux types d'attributs). Dans ce contexte, nous avons constaté la présence de deux groupes des classificateurs, un qui supporte les attributs mixtes (nominaux et continus), comme le naïf bayésien (Ting, 1995), et l'autre, qui peut traiter les attributs nominaux seulement, comme le réseau bayésien et l'arbre de décision (Witten et Frank, 2011). Pour exécuter ces derniers classificateurs, nous devons donc appliquer une méthode qui transfère les attributs continus aux nominaux. Cette méthode s'appelle la discrétisation.

Le mot nominal ou discret réfère à un nombre fini de valeurs possibles (par exemple: rapide / moyen / lent, long / court, 1/2/3, etc.). Les attributs continus réfèrent pour leur part à une gamme infinie de valeurs possibles (par exemple, 0.1, 0.03, 0.2, 0.41, etc.).

Selon (Wang et Valtorta, 2012) la discrétisation se définit comme suit :

> [...] the process of converting the range of possible values associated with a continuous data item (e.g. a double precision number) into a number of sub-ranges each identified by a unique integer label; and converting all the values associated with instances of this data item to the corresponding integer labels.

La discrétisation est une étape essentielle lors du prétraitement de données, mais, pas justement pour homogénéiser les attributs dont certains sont continus et d'autres sont nominaux. Cette étape peut également être utile pour améliorer la performance des algorithmes d'apprentissage qui gèrent la présence d'attributs mixtes (nominaux et continus) (Butterworth *et al.*, 2004; Dougherty *et al.*, 1995). Nous avons résumé dans le tableau 1.1 plusieurs avantages et inconvénients de la discrétisation automatisée qui ont été recensés par la littérature scientifique.

Tableau 1.1 : Avantages et désavantages de la discrétisation

Avantages	**Désavantages**
1. Certaines méthodes d'apprentissage ne peuvent pas gérer les attributs continus. 2. Pour l'interprétation humaine, un ensemble d'intervalles est plus cognitif qu'une série de numéros. 3. Le traitement des données est plus rapide avec un nombre réduit d'états. 4. Homogénéisation de la nature des données (continues ou nominales). 5. C'est un moyen d'améliorer la performance du système de prédiction.	1. Peut conduire à la perte d'information (McGeachie *et al.*, 2014). 2. Pendant la discrétisation, un compromis doit être trouvé entre la qualité de l'information (c'est-à-dire qu'un intervalle doit contenir des informations homogènes) et la qualité statistique (c'est-à-dire que la taille de chaque intervalle doit être suffisante pour assurer sa généralisation) (Kotsiantis et Kanellopoulos, 2006).

La meilleure méthode de discrétisation est celle qui reflète la distribution originale de l'attribut continu et maintient les modèles *(patterns)* cachés dans cet attribut sans

ajouter de parasites (Lustgarten *et al.*, 2008). En réalité, l'expert du domaine (par exemple le médecin) est celui qui fait la meilleure discrétisation, parce qu'il peut adapter les intervalles de données au contexte de l'étude et présente un sens aux attributs transformés (Ricco, 2010). Cependant, avec une grande base d'apprentissage, le coût de l'expert serait prohibitif et parfois, l'expert n'est pas disponible et il est nécessaire de trouver des méthodes automatisées pour discrétiser les attributs prédictifs.

Soulignons qu'il n'est pas possible de faire une revue complète de toutes les méthodes de discrétisation existantes dans ce mémoire. Cependant, nous nous concentrons donc sur celles qui sont mentionnées dans le livre de (Witten et Frank, 2011). Ce livre constitue la documentation du Weka *Explorer*, une collection des algorithmes d'apprentissage automatique qui est utilisable et bien connue dans ce domaine (Weka, 2011). Autrement, ces méthodes de discrétisation sont utilisées par l'outil d'apprentissage automatique *Tanagra* (Ricco, Mai 2010). Ces deux références proposent donc quatre algorithmes et méthodes de discrétisation des attributs continus, qui se divisent en deux catégories: la discrétisation non supervisée et la discrétisation supervisée.

1.4.1.1 Discrétisation non supervisée

L'approche de la discrétisation non supervisée consiste à quantifier les intervalles de chaque attribut en l'absence de toute connaissance de la classe d'attribut. Parmi ces approches, nous avons sélectionné *Equal Width Discretization (EWD)* et *Equal Frequency Discretization (EFD).*

A. Equal Width Discretization (EWD) :

Cette méthode divise les données de chaque attribut en *k* intervalles de taille égale. La largeur de chaque intervalle est égale à *w*. *w*= (max-min)/k, max = le numéro maximum dans un attribut, et min = le contraire.
Après la discrétisation, les frontières des intervalles de l'attribut qui était continu apparaissent comme suit : min, min + w, ..., min + (k-1) w.

B. Equal Frequency Discretization (EFD):
Cette méthode divise les données de chaque attribut continu en *k* intervalles, et chaque intervalle contient environ *N / k records* ou cas. N est le nombre total de *records*.

Selon (Boulle, 2005), dans la plupart des cas, le nombre d'intervalles *(K)* pour les deux méthodes est toujours fixé à 10. En outre, Weka a mis 10 comme nombre d'intervalles par défaut pour les deux méthodes.

1.4.1.2 Discrétisation supervisée

La deuxième approche de discrétisation qui prend en compte la classe d'attribut lors de la discrétisation s'appelle *discrétisation supervisée* et contient les algorithmes suivants:

A. Fayyad & Irani's Minimum Description Length (MDL) criterion

Cet algorithme repose sur les deux idées principales suivantes :

a. D'abord, les valeurs d'un attribut sont triées dans un ordre croissant. Ensuite, le point central entre chaque deux valeurs successives de l'attribut est évalué comme un *cut point,* soit le point qui divise un attribut en deux intervalles.

b. Cet algorithme sélectionne le meilleur *cut point* de toute la gamme des valeurs existantes *(cut points)* d'un attribut. Cette sélection est réalisée en utilisant la

formule de l'*Information Gain* basée sur l'entropie de Shannon. Par la suite, chacun de deux intervalles obtenus par le meilleur *cut point* est discrétisé récursivement de la même façon. Cette discrétisation binaire se déroule de façon continue, jusqu'à ce qu'un nombre maximal d'intervalles d'un attribut soit atteints. Le nombre maximal est représenté comme la condition d'arrêt *MDL criterion*.

B. Kononenko's Minimum Description Length (MDL) criterion method (Kononenko, 1995)

Selon (Ismail et Ciesielski, 2003), cette méthode est très similaire à celle de Fayyad et Irani, sauf qu'elle comprend un ajustement lorsque plusieurs attributs doivent être discrétisés. Ismail et Ciesielski ajoutent que cet algorithme fournit aussi une correction du biais de la mesure d'entropie qui peut diriger un attribut vers de nombreux intervalles.

Pour réviser la formule de la *MDL criterion,* vous pouvez consulter (Ismail et Ciesielski, 2003). Pour plus d'informations à propos de l'algorithme *Fayyad & Irani's MDL*, voir les références suivantes: (Irani, 1993) ou (Witten et Frank, 2011).

1.4.1.3 Discussion

Les méthodes non supervisées EWD et EFD sont attrayants en raison de leur simplicité. Ainsi, (Witten et Frank, 2011) ont constaté que EFD peut donner un excellent résultat de discrétisation. En revanche, (De Bosschere, 2013) a conclu que EWD est la meilleure méthode dans le cadre de ses recherches. Par ailleurs, (Lustgarten *et al.*, 2008) ont démontré que dans la classification, la discrétisation supervisée est plus avantageuse que la discrétisation non supervisée. Précisément, (Kotsiantis et

Kanellopoulos, 2006) ont trouvé que la méthode de Fayyad et Irani offre de meilleurs résultats.

Au regard de cette revue de littérature, nous concluons qu'il n'existe pas de résultat unifié dans ce domaine. (Garcia *et al.*, 2013) soutiennent cette opinion. Cette équipe de recherche a empiriquement analysé de nombreuses méthodes de discrétisation et n'est arrivée à aucune conclusion probante concernant la méthode la plus performante de discrétisation. Pour cette raison, il est important de réaliser sa propre expérimentation afin de choisir la meilleure méthode de discrétisation pour un contexte spécifique.

1.4.2 Sélection des attributs pertinents

> « It seems that perfection is reached not when there is nothing left to add, but when there is nothing left to take away ».
>
> Antoine de Saint-Exupery

Théoriquement, si nous voulons traiter un grand nombre d'attributs, nous devons atteindre plus de pouvoir discriminant dans la classe d'attributs. Cependant, l'expérience pratique avec les algorithmes d'apprentissage automatique a montré que ce n'est pas toujours le cas (Hall, 1999). Depuis la dernière décennie, la sélection des attributs pertinents est un sujet de recherche qui a intéressé différents domaines, en particulier celui de l'analyse de données biologique (Slimani *et al.*, 2014). L'importance de la sélection des attributs pertinents est représentée par la suppression des attributs qui n'ajoutent pas de connaissance à un classificateur et qui sont redondants[1].

[1] Un attribut est dit redondant si un ou plusieurs des autres attributs sont fortement corrélées avec lui.

De plus, une fois effectuée la suppression des attributs, le meilleur sous-ensemble restant est celui qui peut donner la plus grande capacité discriminante à un classificateur. La meilleure façon de sélectionner les attributs pertinents est de façon manuelle, avec les experts du domaine (ex. : les médecins), en se basant sur la compréhension approfondie du problème et sur ce que signifient réellement les attributs. Cependant, les méthodes automatiques peuvent également être utiles (remplacer les experts, faciliter le tâche de sélection, etc.) (Witten et Frank, 2011).

Selon la thèse de doctorat de (Guérif, 2006), la réduction de la dimension de la base d'apprentissage par la suppression des attributs inutiles peut avoir plusieurs objectifs :

1. Améliorer la performance (l'exactitude et l'efficacité) des algorithmes d'apprentissage, spécifiquement dans le cas de la classification.
2. Accélérer le temps d'apprentissage et faciliter la découverte de connaissances lors de l'entraînement.
3. Offrir une base d'apprentissage compacte et facile à interpréter par l'être humain.
4. Réduire l'espace de stockage.

D'autre part, la sélection d'attributs pertinents peut engendrer la perte d'informations dans certains cas. Selon l'analyse menée dans le cadre de cette recherche, une méthode d'évaluation adaptée à un problème précis peut permettre d'éviter ce problème.

1.4.2.1 Procédure de sélection des attributs pertinents

Selon (El Ferchichi, 2013), toutes les procédures de sélection des attributs pertinents tentent de trouver le plus petit groupe d'attributs capable de garantir les deux conditions suivantes :

1. La précision de la classification ne doit pas s'affaiblir;

2. La distribution des classes[2] doit être proche de la distribution originale.

Intuitivement, pour choisir les attributs pertinents, il faut évaluer 2^N sous-ensembles (le nombre *2* représente la nature *sélectionnée* ou *non sélectionnée* d'un attribut et N est le nombre total d'attributs), dans le but de trouver celui qui respecte les deux conditions ci-dessus. Cette méthode est idéale et est la seule façon de choisir le meilleur sous-ensemble, mais elle est exhaustive et très coûteuse à mettre en pratique (Hall, 2000).

Plusieurs algorithmes ont été proposés pour résoudre ce problème en réduisant la complexité de la recherche. De plus, tous les algorithmes de sélection des attributs pertinents ont besoin d'un point de départ, d'une stratégie d'évaluation et d'un critère d'arrêt. Voici en quoi consistent ces quatre étapes:

1. Le point de départ : est un ensemble d'attributs à partir duquel le processus de sélection peut commencer à affecter la direction de la recherche. Par exemple, la recherche peut commencer avec tous les attributs qui sont dans la base d'apprentissage ou avec aucun attribut.
2. L'organisation de recherche : est la stratégie qui génère un sous-ensemble d'attributs qui sera évalué par la méthode d'évaluation. Dans ce contexte, les stratégies de recherche heuristiques sont plus réalisables que celles qui sont exhaustives, et peuvent donner de bons résultats (Hall, 2000) (par exemple le *BestFirst Heuristic Research*).
3. La stratégie d'évaluation : est la façon dont les sous-ensembles d'attributs sélectionnés peuvent être évalués par une méthode de recherche. La stratégie d'évaluation est le principal facteur permettant de différencier les algorithmes de sélection et les attributs pertinents dans l'apprentissage automatique. Le rôle de cette stratégie est de mesurer le pouvoir discriminant d'un sous-ensemble

[2] Les classes sont les états de la classe d'attribut.

d'attributs afin de distinguer entre les différents états de la classe d'attributs, par exemple l'*Information Gain*.

4. Le critère d'arrêt : est employé pour arrêter la recherche à travers l'espace des attributs existants. Ce critère est défini en fonction de la procédure de recherche et de la stratégie d'évaluation.

1.4.2.2 Types de méthodes de sélection existants

Les algorithmes de sélection des attributs pertinents se divisent en deux grandes catégories: les *Wrappers,* qui utilisent l'algorithme d'apprentissage lui-même pour évaluer l'utilité d'un sous ensemble d'attributs (Figure 1.2) et les *filters,* qui permettent d'évaluer les sous-ensembles selon des heuristiques ou des mesures générales (Figure 1.3).

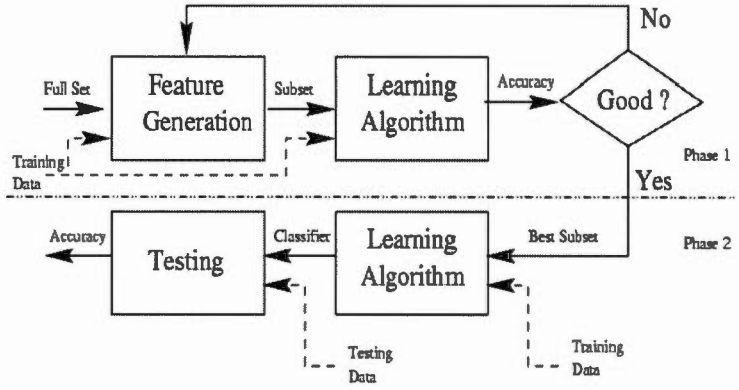


Figure 1.2 : Sélecteur d'attributs *Wrappers* (Cornuéjols, 2006).

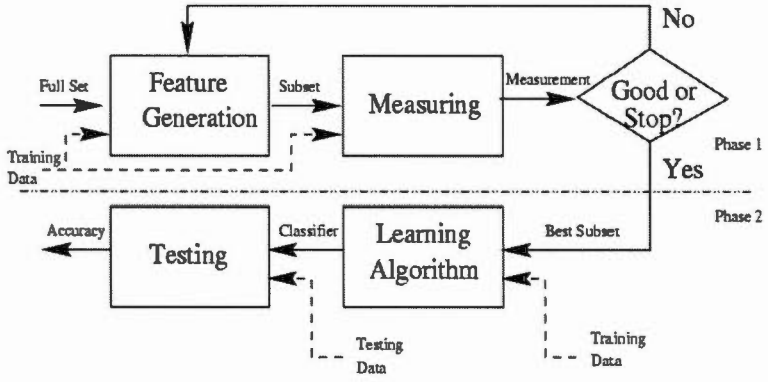


Figure 1.3 : Sélecteur d'attributs *Filters* (Cornuéjols, 2006).

Les méthodes de *filtrage* sont principalement utilisées en pratique (Goswami et Chakrabarti, 2014) et sont en général beaucoup plus rapides que les *Wrappers*, ce pour quoi elles sont plus utilisées pour les bases d'apprentissage de grande dimension (Hall, 1999). Cependant, avec les *Wrappers,* le résultat est concrètement intégré à l'algorithme de classification utilisé.

1.4.2.3 Méthodes Wrappers

Ces méthodes utilisent le classificateur lui-même pour évaluer la qualité d'un sous-ensemble particulier. Ce sous-ensemble est choisi par l'algorithme de recherche, qui est enroulé *("wrapped")* autour du classificateur jusqu'à l'obtention du sous-ensemble d'attributs le plus pertinent, en respectant le critère d'arrêt (Figure 1.2).

Le facteur clé qui fait la différence entre les mêmes méthodes *Wrappers,* est l'algorithme de recherche utilisée. Dans Weka *Explorer*, nous avons identifié plusieurs algorithmes de recherche, mais nous présentons ici ceux qui sont les plus communément utilisés par la communauté informatique, soit l'algorithme génétique et le BestFirst.

A. L'algorithme génétique (GA)

L'algorithme génétique est capable d'explorer efficacement un grand espace de recherche (Karegowda *et al.*, 2010). C'est un algorithme de recherche inspiré du principe de la sélection naturelle, et dont l'idée de base est de faire évoluer une population initiale d'individus afin d'évaluer la population finale et choisir l'individu qui a le score le plus élevé et qui représente la solution.

Dans notre cas, une population sera un ensemble d'individus. Un individu est un sous-ensemble d'attributs qui représente une des solutions. Un gène sera une partie de la solution comme un attribut. Ainsi, la génération est une itération dans l'algorithme lorsqu'on applique à la population les trois opérateurs suivants: reproduction, croisement et mutation.

Cette procédure *wrapped* autour d'un algorithme d'apprentissage jusqu'à ce que le critère d'arrêt soit satisfait. Selon *Weka*, ce critère correspond au nombre de générations à évaluer, qui est égal à 20 par défaut. Ainsi que, initialement la population égale à 20, et la taille de chaque individu (le nombre d'attributs dans un sous-ensemble) est aléatoirement choisie.

B. L'algorithme BestFirst

C'est la méthode qui est la préférée en recherche et qui fonctionne avec le sélecteur Wrapper (Kohavi, 1995b). Habituellement, le *BestFirst* commence par un ensemble vide *(Forward Selection)*, puis génère toutes les possibilités des sous-ensembles qui contiennent un seul attribut. Ensuite, le sous-ensemble présentant la valeur d'évaluation la plus élevée est choisie. Ce sous-ensemble est développé de la même manière, en ajoutant un nouvel attribut. C'est-à-dire, on fait par la suite l'évaluation avec toutes les possibilités des deux attributs, et ainsi de suite. S'il en a le temps, le chercheur *BestFirst* explorera tout l'espace d'attributs (Witten et Frank, 2011), ce pour quoi l'utilisation d'un critère d'arrêt est courante. Dans *Weka,* ce critère correspond au

nombre d'attributs consécutifs et non améliorés avant de terminer la recherche. Il est fixé à 5 par défaut.

1.4.2.4 Méthodes de filtrage

Les premières approches de sélection d'attributs pertinents qui ont été développées étaient les méthodes de filtrage (Hall, 2000). Tous les algorithmes de filtrage utilisent des heuristiques générales pour évaluer un sous-ensemble par rapport à la classe d'attribut. La méthode d'évaluation est le facteur clé qui fait la différence entre les mêmes méthodes *filters*.

En fait, il existe plusieurs algorithmes de filtrage. Dans ce mémoire, nous avons traité des deux méthodes qui sont les plus employées en recherche et qui ont été intégrées au système d'apprentissage *Weka*: A) la *Correlation-based Feature Selection (CFSsubsetEval)* et B), la *Gain Ratio Attribute Eval.*

A. Correlation based Feature Selection (CFSsubsetEval)

La plupart des algorithmes de filtrage existants ne fonctionnent qu'avec des attributs discrets. La *Correlation based Feature Selection algorithm* peut être appliqué aux attributs continus et discrets permettant d'en réduire la dimensionnalité jusqu'à cinquante pour cent dans la plupart des cas. Dans cette partie, nous allons présenter la méthode qui convient avec les attributs continus[3].

Comme pour toutes les méthodes *filters,* le cœur de l'algorithme de *CFSsubsetEval* est l'heuristique d'évaluation. Cette heuristique tient compte de l'influence d'un attribut et

[3] Pour plus d'informations sur la méthode qui convient avec les attributs discrets, vous pouvez consulter Mark A Hall, «Correlation-based feature selection of discrete and numeric class machine learning», (2000).

de l'inter-corrélation entre les attributs, sur la classe d'attribut (*target*). L'hypothèse sur laquelle se fonde cette heuristique est la suivante: « Good feature subsets contain features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other » (Hall, 2000).

La formule utilisée par l'algorithme de *CFSsubsetEval* pour évaluer un sous-ensemble d'attributs continus est la suivante:

$$r_{zc} = \frac{k\overline{r_{zi}}}{\sqrt{k + k - (k-1)\overline{r_{ii}}}} \qquad (1.1)$$

Où,

r_{zc} = La corrélation « mérite » entre un sous-ensemble c contenant k attributs et la classe d'attribut z;

k = Le nombre d'attributs dans le sous-ensemble;

$\overline{r_{zi}}$ = La moyenne de corrélation entre les attributs et la classe d'attribut et

$\overline{r_{ii}}$ = La moyenne de l'inter-corrélation (attribut-attribut).

D'abord, l'algorithme *CFSsubsetEval* calcule la moyenne des $\overline{r_{zi}}$ et $\overline{r_{ii}}$ puis utilise la méthode de recherche pour trouver un sous-ensemble. Cette méthode est le *BestFirst*. Le critère d'arrêt dans cette méthode de recherche est le nombre d'attributs consécutifs et non améliorés la valeur de r_{zc} avant de terminer la recherche.Ce nombre est fixé à 5 selon Weka et le pseudo-code de cet algorithme (Hall, 2000).

B. Gain Ratio Attribute Eval

La mesure *Gain Ratio (GR)* permet d'évaluer chaque attribut en mesurant son taux de gain par rapport à la classe d'attributs. La mesure GR est la modification de *l'information Gain* ou *Gain* permettant de réduire son biais[4] en utilisant *SplitInfo*.

$$\text{GainRatio (T, A)} = \frac{Gain\,(T,\ A)}{SplitInfo\,(T,\ A)} \qquad (1.2)$$

T = classe d'attribut;
A = attribut sélectionné pour mesurer son gain.

Gain Ratio Attribute Eval utilise la méthode de recherche *Ranker,* qui classe les attributs par ordre croissant, en évaluant individuellement chaque attribut à l'aide de la mesure GR. Les attributs de classement (*rank*) bas sont filtrés pour former un nouveau sous-ensemble réduit d'attributs (Priyadarsini *et al.*, 2011).

La mesure GR est compatible avec les attributs continus parce qu'elle fait la discrétisation binaire *(binary splits)* intérieurement, selon la documentation du Weka (Witten et Frank, 2011). Dans notre expérimentation, nous proposons le seuil (*rank*) pour exclure les attributs qui ne sont pas pertinents. Ce seuil est égal à *alpha=Max(Rank)/2.* C'est-à-dire, lorsqu'un attribut a la mesure de *GainRatio* inférieure à alpha, il est éliminé de la base d'apprentissage.

1.4.3 Structure de dépendance entre les attributs

[4] Pour plus de détail concernant la formule *GR,* vous pouvez consulter la section 1.5.1 (id3 et C4.5).

Le réseau bayésien a l'avantage de créer une représentation graphique avec des relations complexes qui permettent de comprendre les dépendances entre les attributs. En revanche, les autres approches d'apprentissage sont limitées à une représentation unique, entre la classe d'attributs et les attributs prédictifs.

Le réseau bayésien est un graphe orienté acyclique dans lequel les nœuds représentent les attributs et les arêtes entre les nœuds représentent souvent des relations probabilistes et non des relations de cause à effet (Himes *et al.*, 2009), bien que des relations causales puissent être trouvées dans un réseau bayésien. Établir la preuve qu'une relation causale existe nécessite l'étude approfondie et isolée de la relation entre chaque couple de variables par un expert (ex. : médecin). Cette approche à base d'experts est la façon la plus intuitive de construire la structure de dépendance. Cependant, la structure à base d'experts comporte plusieurs désavantages: elle peut donner des résultats différents d'un expert à l'autre, qui sont difficiles à obtenir en l'absence de ce dernier et inexacts pour un environnement continuellement modifié. Elle prend beaucoup de temps à construire et parfois, les connaissances spécialisées pour le faire sont indisponibles (Lerner et Malka, 2011).

Pour éviter tous ces défis, la création d'une structure de réseau bayésien à partir des données peut faciliter et accélérer la construction du réseau. Ce dernier est basé sur les dépendances probabilistes qui peuvent quantitativement désigner la corrélation entre les attributs.

La définition du problème de construction d'un réseau bayésien à partir de données peut être déclarée comme suit. Étant donné un ensemble d'attributs $D = X_1, \ldots, X_n$, nous devons trouver un réseau B qui correspond bien à D, dans le but de maximiser la probabilité conditionnelle d'un attribut, étant donné les états des autres attributs.

L'idée naïve de la construction d'un réseau bayésien à partir de la base d'apprentissage est de parcourir tous les réseaux possibles, en choisissant le graphe qui présente le score le plus élevé selon la méthode d'évaluation utilisée. Cela dit, ce parcours exhaustif est super-exponentiel (Robinson, 1977). Alors, pour effectuer le parcours dans un temps raisonnable, l'heuristique est la solution à l'étape de la recherche. Dans ce contexte, les deux algorithmes les plus utilisés pour identifier le réseau de dépendance le plus probable à partir de la base d'apprentissage, sont TAN et K2. Ces algorithmes ne supportent que les attributs discrets (Friedman *et al.*, 1997; Witten et Frank, 2011).

A. L'algorithme K2

L'utilisation de l'algorithme K2 est une approche commune et efficace, selon la communauté de recherche (Witten et Frank, 2011). Cet algorithme recherche de manière heuristique pour identifier le réseau de dépendance le plus probable à partir d'une base d'apprentissage.

Cette méthode limite l'espace de recherche par l'ordonnancement des nœuds. Alors, en supposant un ordre donné (aléatoire) des nœuds (attributs), de sorte qu'un nœud ne puisse pas être le parent d'un nœud qui précède (*le premier nœud ne peut pas avoir de parents*). Cette méthode traite chaque nœud à son tour en lui ajoutant des arêtes des nœuds précédemment traités. À chaque addition d'arêtes, la *mesure bayésienne*[5] est calculée et les parents qui maximisent le score du réseau (mesure bayésienne) parmi les parents possibles sont choisis.

Pendant ce processus, un nombre maximum de parents doit être précisé. Habituellement, ce nombre varie entre N=1,2, ou 3. Tel que pour chaque nœud, le

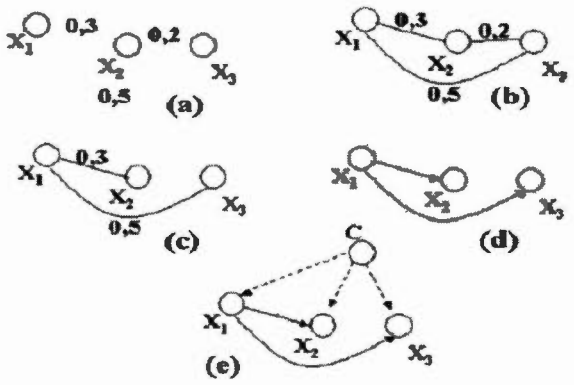
[5] Une métrique pour calculer le score du réseau qui a été attribuée à Gregory F Cooper et Edward Herskovits, «A Bayesian method for the induction of probabilistic networks from data», *Machine learning* 9, no. 4 (1992).

premier parent est une flèche à partir de la classe d'attribut (*c'est-à-dire que pour N=1, le réseau reste comme celui du naïf bayésien*). Ainsi que, le deuxième ou le troisième parent sont sélectionnés selon l'ordonnancement des nœuds et la *mesure bayésienne.* Pour plus d'informations, voir: (Cooper et Herskovits, 1992) et (Witten et Frank, 2011).

B. L'algorithme TAN (Tree Augmented Naive Bayes)

Comme son nom indique, cette méthode prend le réseau du classificateur naïf bayésien, et ajoute des arêtes supplémentaires entre les nœuds, sous forme un arbre. Cette méthode a montré une performance excellente, en dépit de sa simplicité et de fortes hypothèses d'indépendance (Friedman *et al.*, 1997). Le réseau *TAN* est restreint par le nombre de parents des attributs. Dans ce réseau, la classe d'attribut n'a pas de parents, et tous les autres attributs ont comme parent la classe d'attribut, et au maximum un autre attribut.

Donc, TAN s'obtient en trouvant le meilleur arbre qui relie les observations (les attributs sans la classe d'attribut) en utilisant la formule *Conditional Mutual Information* proposée par (Chow et Liu, 1968). Puis on combine toutes les observations reliées à la classe d'attribut. Un exemple de modèle TAN est représenté à la figure 1.4.



C = classe d'attribut
$X_1 \ldots X_3$ = des attributs ou observations, le numéro entre deux nœuds est l'information mutuelle

Figure 1.4 : Tree Augmented Naïve Bayes (TAN) basé sur *Naïve Bayes (NB)* (Gama et Porto, 2008)

Pour construire le TAN, (Friedman *et al.*, 1997) propose la procédure suivante :

1. Calculer *conditional mutual information* (basé sur l'entropie) entre toutes les paires d'attributs étant donnée la classe d'attribut. (Figure 1.4 (a))
2. Construire un arbre qui maximise l'information mutuelle entre chaque paire d'attributs. (Figure 1.4 (b et c))
3. Transformer l'arbre non dirigé en un arbre dirigé en choisissant une variable racine et la direction de toutes les arêtes à l'extérieur de celui-ci. (Figure 1.4 (d))
4. Relier toutes les observations à la classe d'attributs. (Figure 1.4 (e))

Nous venons d'expliquer les méthodes principales pouvant être utilisées à l'étape du prétraitement de la base d'apprentissage (discrétisation, sélection des attributs pertinents) ainsi que la structure de dépendance dans le réseau bayésien. À la prochaine section, nous expliquerons trois algorithmes d'apprentissage pour l'étape de traitement des données: l'arbre de décision, le naïf bayésien et le réseau bayésien.

Rappelons que nous voulons utiliser cet ensemble d'algorithmes pour proposer un modèle de prédiction performant permettant de créer une application contextuelle, ici employable pour le traitement des personnes atteintes de MPOC. Le modèle doit combiner les meilleurs algorithmes aux phases de prétraitement et de traitement des données dans le but de prendre en considération plusieurs aspects manquants des travaux antérieurs (voir à ce sujet le chapitre II).

1.5 Algorithmes d'apprentissage supervisé

L'apprentissage automatique est habituellement divisé en deux types principaux: l'apprentissage prédictif ou supervisé et l'apprentissage descriptif ou non supervisé. Une explication détaillée de ce qui fait la différence entre les deux se trouve à l'appendice A.4.

Dans le cas de notre recherche, la classification sera objective, car la classe d'attributs va distinguer entre les patients qui sont susceptibles d'avoir une exacerbation et ceux qui ne sont pas à risque. Parmi les nombreux algorithmes de classification proposés dans la littérature, nous avons choisi d'utiliser ceux qui sont les plus employés dans la communauté informatique, en nous basant sur le livre de (Stéphane, 2012): l'arbre de décision (ID3 et C4.5), le naïf bayésien et le réseau bayésien. Nous avons choisi de travailler avec plusieurs algorithmes, parce qu'aucun algorithme ne peut prétendre à lui seul résoudre tous les problèmes. Cette question est abordée par plusieurs scientifiques (Wolpert, 1996).

1.5.1 Arbre de décision (ID3 et C4.5)

La construction d'arbres de décision à partir de données est une discipline établie depuis longtemps. Les statisticiens en attribuent la paternité à Sonquist et Morgan (1963). Un arbre de décision est une structure semblable au *FlowChart*[6], qui divise progressivement un ensemble d'entraînement *(training set)* en sous-ensembles de plus en plus petits, basés sur le nœud interne (Figure 1.5, nœud gris clair), qui représente le « test » sur un attribut. Chaque branche représente le résultat du test (Figure 1.5, nœud gris foncé), et chaque nœud de feuille représente un état dans la classe d'attribut. Les chemins de la racine aux feuilles représentent les règles de la classification.

La construction des arbres de décision à partir de données est réalisée par le moyen de plusieurs méthodes: la méthode Chi-Square Automatic Interaction Detection (CHAID), Quick Unbiased Efficient Statistical Trees (QUEST), Inductive Decision Tree (ID3) et son successeur C4.5.

Selon (Rakotomalala, 2005), la différence entre cet ensemble de méthodes, repose sur les points suivants:

1. La stratégie adoptée pour choisir l'attribut de segmentation (Figure 1.5, nœud gris foncé). Exemple de stratégie : l'*information Gain* pour l'ID3 et le *Gain Ratio* pour C4.5.
2. La stratégie de discrétisation lorsque les attributs dans la base d'apprentissage sont continus (ex. Binaire avec C4.5, Non supporté pour ID3).
3. Les règles qui définissent la taille de l'arbre ou la condition d'arrêt, pour mettre un état de la classe d'attribut comme une feuille (ex : un état dans la classe d'attribut est une feuille lorsqu'un état de l'attribut de la segmentation fait l'état de la classe d'attribut pure (Sieben et Gather, 2007)).

[6] Est un type de diagramme qui représente un algorithme, un flux de travail ou un processus.

La stratégie adoptée pour choisir l'attribut de segmentation est le facteur clé dans la construction de l'arbre de décision. Nous allons expliquer cette stratégie pour ID3 (Quinlan, J. Ross, 1986) et C4.5 (Quinlan, J Ross, 1993), qui sont probablement les méthodes le plus populaires dans le domaine de l'apprentissage automatique (Salzberg, 1994).

Dans les algorithmes d'ID3 et C4.5, la stratégie de segmentation est basée sur l'entropie de Shannon (Shannon, 1948). L'entropie est une fonction symétrique qui nous permet de calculer la pureté d'un attribut. (Un exemple simple de l'entropie de Shannon se trouve à l'appendice A.1.)

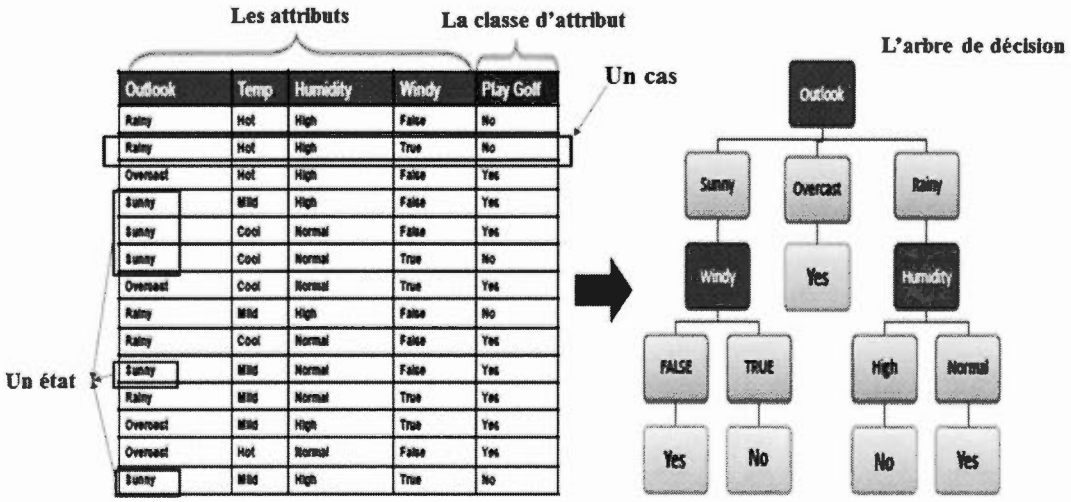


Figure 1.5 : Construction de l'arbre de décision

En général, si on considère une distribution de probabilité $P = (p_1, p_2..., p_c)$ de chaque état d'un attribut 'S', où p_i est la probabilité de i-ème état de 'S', la formule générale d'entropie sera la suivante:

$$\text{Entropy (S)} = -\sum_{i=1}^{C} p_i \text{Log}_2(p_i) \qquad (1.3)$$

La fonction d'entropie vérifie la propriété suivante : elle prend son minimum lorsque tous les cas se trouvent dans un même état, c'est-à-dire lorsque l'attribut est pur, et son maximum lorsque les états sont équirépartis.

Maintenant, pour mesurer l'effet d'un attribut particulier sur la pureté de la classe d'attribut, l'*Information Gain* est utilisé. La formule de l'*Information Gain* est la suivante:

$$\text{Gain}(T, A) = \text{Entropy}(T) - \sum_{v \in A} \frac{|T_v|}{|T|} \text{Entropy}(T_v) \qquad (1.4)$$

T = La classe d'attribut (*target*) ;
A = L'attribut sélectionné pour mesurer son gain ;
$|T_v|$ = Le nombre de cas du T dans lesquels l'attribut A prend l'état ou la valeur v et
$|T|$ = Le nombre de cas du T.

Cette mesure (Gain) a été utilisée dans l'algorithme d'ID3 pour sélectionner l'attribut le plus pertinent à chaque itération d'algorithme. Un exemple simple qui décrit la construction de l'arbre de décision avec ID3 se trouve à l'appendice A.1.2.

Quinlan a utilisé l'*Information Gain* pour l'algorithme d'ID3 (Quinlan, J. Ross, 1986). Ensuite, en 1993, il a proposé le GainRatio pour l'algorithme C4.5 (Quinlan, J Ross, 1993). C4.5 est l'évolution de l'ID3, où la notion de GainRatio tend à régler le problème des attributs qui ont un grand nombre d'états. Par exemple, si nous avons un attribut 'A' qui a des états différents pour chaque cas, alors Gain (T, A) est maximal, car $Entropy\ (T_v) = 0$. Afin d'éviter cette situation, Quinlan suggère dans le C4.5 d'utiliser le rapport suivant au lieu de l'*Information Gain* (Ingargiola, 1996):

$$\text{GainRatio}(T, A) = \frac{\text{Gain}(T, A)}{\text{SplitInfo}(T, A)} \qquad (1.5)$$

La formule de la SplitInfo est la suivante:

$$\text{SplitInfo}\,(T, A) = -\sum_{i=1}^{C} \frac{|T_i|}{|T|} \text{Log}_2 \frac{|T_i|}{|T|} \qquad (1.6)$$

Où T_i est la partition de T (T est la classe d'attribut) induite par i-ème état de A.

1.5.2 Bayésien naïf

Le naïf bayésien est un type de classification simple et facile à construire qui est basé sur le théorème de Bayes. On dit que cette approche est « naïve », parce que son principe de base repose sur l'hypothèse d'indépendance conditionnelle (Kumari, 2014). Cela signifie que les attributs sont conditionnellement indépendants entre eux étant donné la classe d'attribut. La performance concurrentielle de cette méthode dans le contexte du classement d'apprentissage supervisé est surprenante, car l'hypothèse d'indépendance conditionnelle sur laquelle cette méthode est basée est rarement vraie dans les applications du monde réel. Cela dit, (Zhang, 2004) a montré qu'il existe des raisons théoriques à cette efficacité inattendue.

Le naïf bayésien est largement utilisé pour la classification et il a prouvé son efficacité dans de nombreuses applications pratiques, comme la classification de textes, le diagnostic médical et la gestion de performance d'un système (Rish, 2001). Le naïf bayésien est considéré comme une forme spéciale et simple d'un réseau bayésien. De manière plus claire, elle est incarnée dans un réseau de croyances particulières où la classe d'attribut n'a pas de parents et où la classe d'attribut est la seule parente pour chaque autre attribut. La figure 1.6 montre un exemple du réseau naïf bayésien qui représente la base d'apprentissage à la figure 1.5.

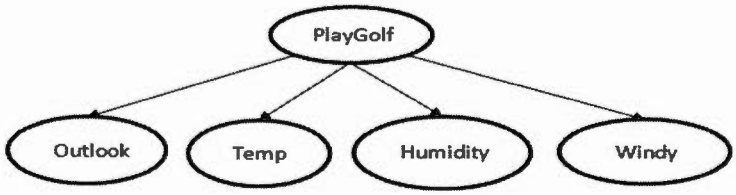


Figure 1.6 : Réseau de croyance correspondant à la classification naïve bayésienne

Le théorème de Bayes fournit le moyen de calculer la probabilité *a posteriori* et est représenté par la formule suivante :

$$P(A \mid B) = \frac{P(A,B)}{P(B)} \qquad (1.7)$$

A et B, deux événements quelconques d'un ensemble E muni d'une loi de probabilité P.

Selon la formule de Bayes, nous pouvons écrire la relation suivante :

$$P(C \mid F1, F2 \ldots Fn) = \frac{P(C) \cdot P(F_1, F_{2\ldots} F_n \mid C)}{P(F_1, F_{2\ldots} F_n)} \qquad (1.8)$$

Où 'C'est la classe d'attribut (ex: PlayGolf, figure 1.5) conditionnée par plusieurs attributs *Fi* (ex. Outlook, Temp, etc.).

En tenant compte l'hypothèse d'indépendance :

$$\text{Alors, } P(F_1, F_{2\ldots} F_n \mid C) = \prod_{i=1}^{n} P(F_i \mid C)$$

$$P(C \mid F_1, F_{2\ldots} F_n) = \frac{1}{Z} P(C) \prod_{i=1}^{n} P(F_i \mid C) \qquad (1.9)$$

Où 'Z' est appelé la constante de normalisation, qui peut être calculée par la somme des probabilités *a posteriori* $P(C \mid F_1, F_{2\ldots} F_n) + P(\bar{C} \mid F_1, F_{2\ldots} F_n) = 1$.

Cet ensemble de formules est utilisé lorsque la base d'apprentissage est catégorique ou discrète. (Un exemple illustratif se trouve à l'appendice A.2.3.)

Dans le cas où la base est continue, nous devons la transformer en *catégories* par l'utilisation des méthodes de discrétisation proposées dans la littérature (ex: *equal frequency, equal width,* etc.) (Yang et Webb, 2002). Cependant, dans certains cas, la discrétisation peut gaspiller l'information discriminante (Hand et Yu, 2001). Pour cette raison, une autre hypothèse est utilisée pour le naïf bayésien.

Cette hypothèse est que les valeurs continues associées à chaque attribut continu sont réparties selon la distribution gaussienne ou normale (Zhang, 2004). Cette proposition est importante pour notre recherche, qui se concentre sur l'étude des facteurs ou des attributs biologiques des patients atteints de MPOC, parce que beaucoup d'attributs biologiques vont suivre la distribution normale (Bland et Altman, 1996).

La densité de la distribution normale est définie par deux paramètres : la moyenne et l'écart-type.

Moyenne : $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$; Écart-type : $\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2}$

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}}\, e^{\frac{-(x-\mu)^2}{2\sigma^2}} \qquad (1.10)$$

Un exemple explicatif de la distribution normale dans la classification naïve bayésienne se trouve à l'appendice A.2.4.

1.5.3 Réseau bayésien

Malgré l'importance que prend le réseau de neurones en recherche actuellement, le réseau bayésien a été étudié beaucoup plus en profondeur (Naïm *et al.*, 1999). Il est considéré comme l'approche la plus appropriée pour modéliser l'incertitude dans l'intelligence artificielle et a été employé pour le développement d'une grande gamme d'applications médicales (Simões *et al.*, 2015) : les systèmes de surveillance en santé (Tsui *et al.*, 2003), la détection des tumeurs cérébrales (Reynolds *et al.*, 2007), la prédiction des risques de décès chez les patients ayant subi une chirurgie cardiaque (Verduijn *et al.*, 2007) et l'identification des patients à risque d'exacerbations liées à l'asthme (Sanders et Aronsky, 2006b). De plus, la National Aeronautics and Space Administration (NASA) a utilisé le réseau bayésien dans son application *Vista* pour fournir des conseils à propos de la possibilité d'échecs dans les systèmes de propulsion de la navette spatiale (Naïm *et al.*, 1999).

Le réseau bayésien est un graphe orienté et acyclique, où les nœuds représentent des attributs et où les arêtes sont des relations de dépendance. Ainsi, un nœud ayant une flèche entrante dépend du nœud d'où provient la flèche. L'orientation fournit un moyen simple et efficace pour exprimer les hypothèses de dépendance entre les attributs, guide la manière d'interpréter, économise un temps énorme calcul pendant l'inférence et minimise l'espace occupé dans la mémoire (Olivier, 2006). Cette structure de dépendance peut être réalisée en s'appuyant sur les connaissances d'experts, ou encore sur des données d'apprentissage lorsque les dépendances entre les attributs ne sont pas claires (section 1.4.3). De plus, chaque nœud du réseau bayésien peut présenter plusieurs états, qui sont énumérés dans un tableau de probabilités conditionnelles (CPT), en fonction de tous les états possibles de leurs parents.

D'autre part, le réseau bayésien peut faire une bonne inférence seulement à l'aide des observations disponibles. Autrement dit, il n'a pas besoin d'avoir toutes les connaissances d'attributs pour faire l'inférence. De plus, il ne se limite pas à savoir quel sera l'état le plus probable d'un nœud en fonction des informations observées,

c'est-à-dire P(état/cause), mais permet aussi de trouver les causes les plus probables de l'état d'un nœud donné qui est équivalent à P(cause/état).

Formellement, un réseau bayésien est défini par la paire (G, V), où 'G' représente un graphe et 'V', l'ensemble d'attributs $\{V_1, V_2 \ \dots \ , V_n\}$ dans le graphe G.

L'inférence dans le réseau bayésien est basée sur la probabilité de conjointe suivante :

$$P(V_1, V_2 \ \dots \ , V_n) = \prod_{i=1}^{n} P(V_i | Pa(V_i)) \qquad (1.11)$$

$Pa(V_i)$ est l'ensemble de parents du nœud V_i.

Cette probabilité de conjointe repose sur l'hypothèse de l'indépendance conditionnelle (D-séparation) pour la circulation des informations dans le réseau bayésien. Les trois règles (chaîne, cause commune et effet commun) de D-séparation servent à simplifier et réduire le calcul dans le réseau bayésien. L'explication de ces trois règles se trouve à l'appendice A.3.5. Aussi, un exemple simple du réseau bayésien illustrant son application pour la prédiction d'une maladie se trouve à l'appendice A.3.6.

1.6 Conclusion

Au cours de ce chapitre, nous avons décrit les trois premières entités (définition du contexte, son modélisation et raisonnement) principales permettant de construire une application contextuelle ou sensible au contexte. Nous avons donc adopté une définition qui convient à notre contexte et nous avons expliqué les choix disponibles afin de le modéliser. Finalement, nous avons détaillé douze algorithmes qui sont utilisés lors des phases de prétraitement et traitement de l'apprentissage automatique.

Au prochain chapitre, nous allons décrire les motivations biologiques, économiques et informatiques qui nous incitent à systématiser efficacement l'identification des paramètres pertinents au traitement de l'exacerbation dans la MPOC. Au même chapitre, nous allons aborder les défis informatiques posés par notre démarche.

CHAPITRE II

MOTIFS DE TRAITEMENT DE LA MPOC ET LES SYSTÈMES INFORMATIQUES QUI LE SOUTIENNENT

Les problèmes des attributs pertinents et des raisonnements sur ces derniers dont nous avons discuté au chapitre I pourraient être identifiés dans plusieurs domaines d'application, particulièrement dans le domaine médical. Dans ce chapitre, nous insistons sur la compréhension de la maladie pulmonaire obstructive chronique (MPOC) et son exacerbation, ainsi que sur leurs effets économiques et biologiques sur les patients. D'autre part, nous rendons compte des travaux informatiques reliés à ces effets et à la surveillance à domicile des patients atteints de MPOC. À la fin de ce chapitre, notre analyse nous permet d'énumérer les défis des différents systèmes informatiques dont traite la littérature, ainsi que l'utilité des algorithmes présentés au chapitre I (voir les détails dans la section 2.5). Rappelons que notre objectif est de concevoir une application contextuelle performante et autonome pour résoudre ces défis.

2.1 Définition des maladies chroniques et de la MPOC

En général, les maladies chroniques nécessitent un traitement médical à long terme (Megari, 2013), c'est-à-dire qu'il peut durer trois mois ou plus, selon la définition du *Centre national des statistiques de santé en Amérique.* Souvent, ces maladies ne disparaissent pas (Van der Heijden *et al.*, 2014) et limitent l'état fonctionnel et la

productivité des patients. Parmi ces maladies, nous pouvons compter le diabète, le cancer, les maladies pulmonaires, etc. (Brazeau, 2005). Au Canada, les maladies pulmonaires ont un caractère spécial, puisque près de 3,5 millions de Canadiens vivent avec ce type de maladie, qui englobe l'asthme et MPOC (Canada, 2007). La MPOC et l'asthme sont les deux maladies respiratoires les plus répandues, qui touchent les voies aériennes et d'autres parties du poumon (Geneviève, 2013). Spécifiquement, le traitement de la MPOC coûte environ trois fois plus cher que les asthmes (van-Mölken et Feenstra, 2001).

2.1.1 Qu'est-ce que la maladie pulmonaire obstructive chronique (MPOC)?

La broncho-pneumanopathie chronique obstructive (BPCO) est un terme générique de la maladie pulmonaire obstructive chronique. Au Québec, elle est nommée au MPOC ou *Chronic Obstructive Pulmonary Disease (COPD)* (Québec, 2014). Lors d'un fonctionnement normal des poumons, l'air entre par le nez ou la bouche, pour arriver aux bronches et le faire passer aux bronchioles. Les bronchioles amènent l'oxygène aux alvéoles pulmonaires (Figure 2.1). La paroi de chaque alvéole est comme un tissu très fin qui permet à l'oxygène de passer vers le sang sans laisser le sang fuir (Brazeau, 2005).

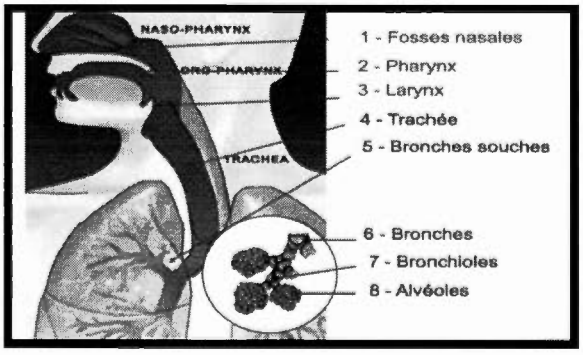


Figure 2.1: Voies aériennes inférieures (coupe intra-thoracique) (Busson, 2014)

Un patient est atteint de MPOC lorsque les poumons sont infectés. L'infection est un mélange d'obstruction des petites voies aériennes et de destruction des alvéoles, des phénomènes qu'on appelle respectivement la bronchite chronique et l'emphysème (Figure 2.2).

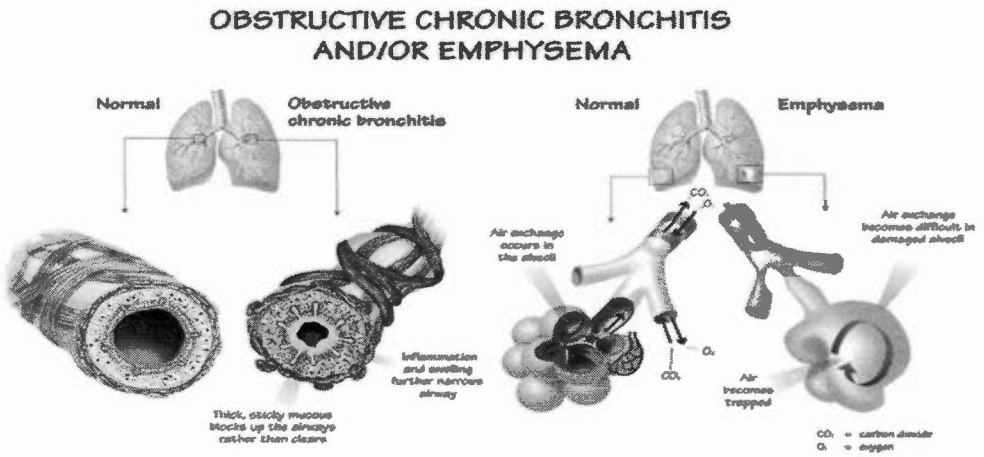


Figure 2.2 : *Obstructive Chronic Bronchitis and/or Emphysema* (McGill, 2016)

L'emphysème est défini comme la destruction des membranes qui séparent les alvéoles avec l'élargissement irréversible des espaces aériens distaux des bronchioles terminales et sans signe de fibrose[7]. La bronchite chronique est définie comme «*productive cough that is present for a period of 3 months in each of 2 consecutive years in the absence of another identifiable cause of excessive sputum production*». (Juvelekian, 2012).

2.2 Effets biologiques et économiques de la MPOC

[7] La fibrose pulmonaire est une lésion qui rend les poumons épais, raide et cicatrisée.

La MPOC est un problème majeur et constitue la principale cause de la mortalité dans le monde entier (Mannino et Buist, 2007). En chiffre, elle est considérée comme la troisième cause de décès chez les Américains (Juvelekian, 2012), la quatrième au Canada, hommes et femmes confondus (Québec., 2016a) et le cinquième fardeau médical dans le monde d'ici l'an 2020 (Murray et Lopez, 1997). La MPOC est la seule maladie pour laquelle le taux de mortalité augmente continuellement (Juvelekian, 2012). Financièrement, 32 milliards de dollars de frais ont été associés à cette maladie aux États-Unis en 2010 et ces charges risquent d'augmenter jusqu'au 49 milliards de dollars en 2020 (Ford *et al.*, 2015). Au Canada, la MPOC a un impact économique tout aussi majeur: les dépenses qui y sont associées remontent à plus de 1,67 milliard de dollars en 2004 (O'Donnell *et al.*, 2004). Elle représente un problème important en terme économique pour la société, qui doit être résolu.

Malheureusement, à ce jour il n'y a pas de traitement curatif de la MPOC. Les traitements actuels visent à ralentir l'évolution de cette maladie, à réduire les symptômes, à améliorer la qualité de vie des personnes atteintes (Québec., 2016a) et à éviter la crise pulmonaire ou l'exacerbation, qui ont les mêmes conséquences sur la santé qu'une crise cardiaque (Thoracologie, Février 2010). L'exacerbation est l'événement le plus important dans la progression de la MPOC et elle conduit à une augmentation de la mortalité (Connors Jr *et al.*, 1996). Nous l'expliquons plus en détail dans à la section suivante.

2.3 Définition de l'exacerbation

En 1995, l'American Thoracic Society (ATS) a reconnu que le phénomène de l'exacerbation chez les patients atteints de MPOC est difficile à définir parce que son mode de développement est mal compris (Ats, 1995). Selon (Rodriguez-Roisin, 2000a), les difficultés de proposer une définition standard de l'exacerbation, sont dues

aux raisons suivantes: 1) la fluctuation et la diversité des symptômes observés et 2), les comorbidités ou les causes d'exacerbation qui peuvent apparaître avec une respiration normale.

Généralement, l'exacerbation est définie par l'aggravation de symptômes qui obligent à se soumettre à des soins de santé imprévus pour en atténuer les effets (Van der Heijden *et al.*, 2013). En outre, plusieurs définitions suggèrent que l'exacerbation consiste en une fonction pulmonaire altérée, un événement aigu ou encore en une aggravation soudaine des symptômes de la MPOC.

En plus de ces ambiguïtés de définition, les symptômes et les facteurs qui influent sur la gravité et la fréquence des exacerbations sont inconnus ou non spécifiques à la MPOC (Lareau *et al.*, 2014). Ces symptômes peuvent varier en fonction de la quantité de dommages aux poumons. Ils peuvent également fluctuer (Healthline.com, 2014) et s'exprimer de façon légèrement différente selon chaque personne (Van der Heijden *et al.*, 2014). Par ailleurs, aucun marqueur biologique ne permet de démontrer de manière fiable la différence entre un état stable de la MPOC et une exacerbation (Hurst *et al.*, 2006).

Malgré cette méconnaissance des symptômes de l'exacerbation, plusieurs chercheurs ont mis en évidence quelques signes ou symptômes pertinents, qui peuvent proportionnellement influencer la détection des exacerbations. On peut citer parmi eux le tabagisme (Nizet *et al.*, 2005), les infections respiratoires (Fagon et Chastre, 1996), l'indice de masse corporelle (Tantucci *et al.*, 2008), les symptômes dépressifs (De Voogd *et al.*, 2009), l'hospitalisation précédente pour une exacerbation, le genre (homme/femme), la capacité vitale forcée (CVF) (soit le volume d'air expulsé avec force mesuré à l'aide d'un spiromètre) (Montserrat-Capdevila *et al.*, 2016), etc.

En raison du nombre important de symptômes à considérer et de leur ambiguïté, le risque d'exacerbation est totalement incertain chez les patients de la MPOC.

Cependant, l'incertitude est un sujet bien traité dans le domaine de l'apprentissage automatique en informatique, où elle est définie comme une situation d'information inadéquate où le traitement logique *"If-Else"* ne permet pas de traiter le phénomène (Parsons et Kubat, 1994). L'incertitude peut être de trois sortes: l'inexactitude, le manque de fiabilité et l'ignorance (Mcheick *et al.*, 2015).

Dans ce contexte, nous nous sommes intéressé à l'incertitude selon deux points de vue: i) la sélection des attributs ou des symptômes pertinents en utilisant des algorithmes informatiques et ii), la prédiction de la survenue de l'exacerbation avec une bonne précision (les sections 2.3.1 et 2.4 reviennent sur ces points).

2.3.1 Conséquences des exacerbations

Les exacerbations de la MPOC sont des événements importants, qui peuvent avoir des conséquences particulières et différentes de celles de la MPOC stable. En effet, l'exacerbation peut amener un patient à :

1. La mort (Lareau *et al.*, 2014).
2. La dégradation de la qualité de vie (Viegi *et al.*, 2007).
3. La détérioration rapide de la fonction respiratoire ou la maladie MPOC (Burt et Corbridge, 2013), qui peut rester de plusieurs journées jusqu'à quelques semaines (Seemungal *et al.*, 2000).

Ainsi, l'exacerbation est le principal facteur des visites médicales et des hospitalisations (Québec., 2016b). Seulement 50 % de toutes les exacerbations sont signalées aux médecins (Seemungal *et al.*, 1998). Cet ensemble de conséquences fait de la prévention des exacerbations un objectif particulièrement important à atteindre.

Habituellement, pour éviter une maladie, nous pensons à l'utilisation de médicaments. Mais, la médication ne permet pas de guérir la MPOC mais seulement de dilater les bronches afin d'apporter plus d'air aux alvéoles (Canada, S., 2010). Cependant, la détection rapide d'une exacerbation aide à réduire ses effets, à faciliter la récupération des poumons (Wilkinson *et al.*, 2004) et à empêcher l'aggravation de la maladie. Ainsi, le suivi quotidien de la MPOC est indispensable, pour garantir la détection rapide d'une exacerbation.

2.4 Systèmes informatiques existants pour suivre les exacerbations

L'incertitude, les causes qui ne sont jamais connues (nombre indéterminé des symptômes), la nature insidieuse de la maladie et l'ambiguïté de sa définition sont les points abstrus de l'exacerbation de la MPOC. Cette maladie progresse rapidement, nécessite un traitement rapide et il est difficile de déterminer quand la consultation d'un pneumologue est nécessaire (Berkhof *et al.*, 2015). Ces facteurs perturbent le patient, qui doit prendre des décisions utiles pour sa santé et son budget. Or, cette problématique peut être adressée par une méthode informatique qui peut surveiller quotidiennement et à domicile les patients, afin de prédire de façon performante les risques d'exacerbation. Dans le cas de la MPOC, le nombre indéterminé de symptômes nous incite à sélectionner les attributs (symptômes) pertinents indiquant l'exacerbation en utilisant de nouveaux algorithmes informatiques.

Ce suivi est nécessaire dans le cadre domestique (McKinstry *et al.*, 2009) et offre l'opportunité d'une intervention précoce si nécessaire. L'utilisation des méthodes informatiques s'accompagne de plusieurs avantages, dont les principaux sont les suivants :

1. Éviter les visites médicales imprévues et réduire les coûts d'hospitalisation.

2. Accorder plus de responsabilité aux patients, qui peuvent gérer eux-mêmes leur maladie, et du fait augmenter leurs compétences d'autogestion et leur compréhension du phénomène de l'exacerbation.
3. Soulager l'engorgement des salles d'urgence des hôpitaux et des cliniques médicales.
4. Améliorer la qualité de la vie liée à la santé du patient (*"health-related quality of life"* ou *HRQoL*) (Van der Heijden *et al.*, 2013).
5. Prolonger la vie (Jensen *et al.*, 2012).

En fait, la surveillance de la MPOC est un sujet qui a beaucoup suscité l'intérêt depuis un certain temps, mais le traitement automatique des données observées concernant cette maladie est généralement limitée (Van der Heijden *et al.*, 2013) (Trappenburg *et al.*, 2008). Pour cette raison, la prochaine section fait la revue des différents types des systèmes de surveillance existants. Notre revue de littérature nous a permis de distinguer trois types distincts de travaux antérieurs:

- Les systèmes de télésanté (ou de communication à distance) avec les patients;
- Les systèmes qui offrent des alertes automatiques ou qui font une interprétation automatique des données;
- Les systèmes qui concernent la sélection des attributs pertinents dans la MPOC.

2.4.1 Systèmes de télésanté

Dans le domaine de la gestion systématique à long terme de la MPOC, les premiers travaux qui ont été réalisés ont placé l'accent sur la télésanté. Cette technologie a fourni un moyen de communication entre le patient et son médecin, dans le but d'améliorer et de personnaliser le soin de santé à distance.

La littérature scientifique sur le sujet rend compte des systèmes suivants :

a) Le premier système de surveillance de la MPOC a été rapporté par (Maiolo *et al.*, 2003). Dans ce projet, les patients ont été suivis chez eux et seules les informations qui concernent la détermination de la saturation artérielle en oxygène (SAO_2) et la fréquence cardiaque ont été capturées. Ensuite, ces informations ont été transmises automatiquement deux fois par semaine au centre de traitement de l'hôpital en utilisant une ligne téléphonique normale. Les spécialistes ont analysé manuellement les informations reçues et appelaient le patient pour commenter ses symptômes ou pour apporter des changements nécessaires aux prescriptions (Figure 2.3).

La télésurveillance des patients par des médecins peut améliorer la fiabilité de la surveillance, mais ce système comporte plusieurs inconvénients. En particulier, la réponse prend beaucoup de temps pour détecter une rechute de la MPOC, alors que cette maladie requiert une intervention précoce (Himes *et al.*, 2009). De plus, ce système coûte cher, car il nécessite une analyse manuelle par des spécialistes.

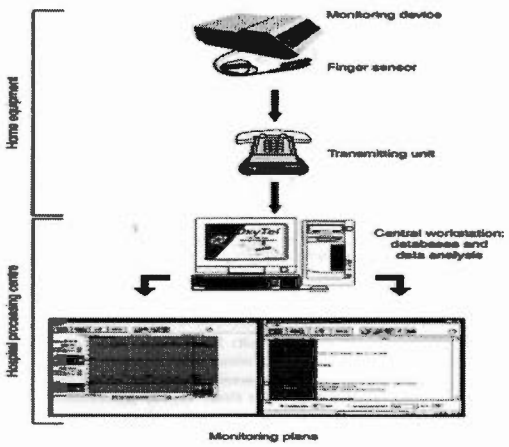


Figure 2.3 : Système de surveillance des patients atteints de MPOC (Maiolo *et al.*, 2003)

b) Dans le projet de (Vontetsianos *et al.*, 2005), la surveillance de la MPOC est réalisée comme suit : l'infirmière visite les patients d'une façon planifiée, en moyenne 0.8 fois par mois. L'infirmière est équipée d'un ordinateur portable et d'un certain nombre de dispositifs médicaux (ex. spiromètre, oxymétrie, etc.) ainsi que d'une caméra de vidéoconférence lui permettant de consulter les spécialistes de l'hôpital. Les spécialistes reçoivent par l'infirmière les possibles indices d'exacerbation et à leur tour conseillent les patients pour améliorer leur qualité de vie.

   L'avantage majeur de ce système réside dans le fait que le patient peut recevoir des rétroactions personnalisées de la part du médecin. Cependant, le traitement manuel est très coûteux et le nombre de visites n'est pas suffisant pour assurer le suivi d'une maladie insidieuse comme la MPOC.

c) (Trappenburg *et al.*, 2008) rendent compte d'un système de surveillance fourni aux patients atteints de MPOC pour détecter les risques d'exacerbation. Chaque patient a répondu personnellement aux questionnaires sur une base quotidienne en utilisant un appareil à quatre boutons appelé *Health Buddy* (Figure 2.4). L'évaluation de ces réponses est fournie par des infirmières qui les examinent à distance du lundi à vendredi.

Figure 2.4 : *Health buddy (HB) device*

Les avantages de ce système sont similaires aux précédents, mais ce système peut, en plus, améliorer la capacité d'autogestion des patients qui observent leurs propres symptômes sans les instructions d'une infirmière. Ses désavantages sont aussi semblables à ceux des systèmes précédents, en considérant aussi qu'il implique le besoin continu du réseau internet pour accomplir le test.

d) Actuellement, (Crooks, 2016) introduit le projet United4Health pour la télésurveillance et le traitement des maladies chroniques. Ce projet concerne les patients de la MPOC qui sont admis à l'hôpital avec une exacerbation. À la sortie d'une hospitalisation, les patients vont être équipés par du matériel de télésurveillance, y compris la vidéoconférence et une oxymétrie de pouls, pour les utiliser à la maison. L'infirmière, qui est à l'hôpital, reçoit régulièrement les données concernant l'état de santé du patient et organise un appel téléphonique, un test vidéo, une visite à domicile ou encore à l'hôpital selon le besoin. C'est l'infirmière qui décide quand le matériel de télésurveillance peut être retiré du domicile; le patient continue alors à s'autogérer. La limite de ce système est que l'infirmière doit être prête à toute éventualité et le patient toujours conscient et attentif de ses symptômes suite au retrait du matériel de télésurveillance.

Même si un système comme celui de (Crooks, 2016) existe actuellement, plusieurs chercheurs ont étudié, depuis 2008, l'effet de la télésanté sur la qualité de vie d'un patient. Dans ce contexte, (Polisena *et al.*, 2010) ont effectué une revue de littérature systématique concernant la télésanté à domicile, par rapport aux soins habituels de la MPOC. La revue a été faite sur dix systèmes: quatre études ont comparé la télésurveillance à domicile par rapport au mode de soin habituel (ex: aller à la clinique) et six ont comparé l'assistance téléphonique au mode de soin habituel. La différence entre le soin téléphonique et la télésurveillance est que ce dernier système peut envoyer

les informations électroniquement. La conclusion de cette étude est que le taux de mortalité chez les patients traités par support téléphonique est plus élevé que chez les patients traités par les soins habituels. Ainsi, les interventions de télésurveillance à domicile donnent des résultats semblables ou meilleurs que les soins habituels pour la qualité de vie et la satisfaction des patients. Similairement, (McLean *et al.*, 2012) ont confirmé que la télésanté peut réduire de façon significative le coût de la présence à l'hôpital, mais qu'elle a peu d'effet sur le risque de décès.

Récemment, dans les travaux de (Berkhof *et al.*, 2015), des infirmières ont été embauchées pour faire des appels téléphoniques aux patients atteints de MPOC. Les patients ont été contactés toutes les deux semaines pour répondre à un bref questionnaire, le *Clinical COPD Questionnaire (CCQ)* (Van der Molen *et al.*, 2003). Le score total de la CCQ est enregistré dans une base de données et si un changement est observé par rapport au score de CCQ précédent, les pneumologues sont avisés d'appeler immédiatement le patient. Cette étude a montré que la télémédecine par téléphone n'améliore pas l'état du patient. Au contraire, la visite régulière à la clinique améliore significativement le score de CCQ.

En plus du taux de la mortalité qui augmente à l'usage de ce type de systèmes de télésanté, nous constatons que la plupart des architectures proposées dans ce domaine présentent communément les inconvénients qui sont énumérés ici:

1. Il n'y a pas d'alertes immédiates aux patients en présence d'un risque d'exacerbation (Maiolo *et al.*, 2003).
2. Les systèmes de télésanté n'améliorent pas l'état du patient en comparaison à la visite régulière à la clinique (Berkhof *et al.*, 2015).
3. La réalisation de ce type de projet, qui implique par exemple la visite répétée de l'infirmière, coûte cher aux patients.
4. La connexion au réseau internet est toujours nécessaire pour traiter l'état du patient (Trappenburg *et al.*, 2008).

5. Avec ce type de système, il n'y a pas de traitement concentré sur la détection précoce des exacerbations.

2.4.2 Systèmes de traitement automatique de la MPOC

Les systèmes automatisés éliminent le besoin d'interférence des spécialistes pour accomplir les tâches primaires de la détection d'une maladie. Dans ce contexte, plusieurs chercheurs ont proposé des systèmes automatisés pour assurer le suivi de la MPOC, dans le but d'améliorer les systèmes de surveillance existants et de réduire les coûts d'hospitalisation. Les systèmes relevant de ce type de surveillance sont les suivants:

a) Selon (Halpin *et al.*, 2011), il y a une forte corrélation entre les facteurs météorologiques et l'incidence d'exacerbation de la MPOC, qui présente une forte saisonnalité. En fonction des prévisions météo, ces auteurs ont évalué un modèle prédictif pouvant détecter l'exacerbation chez les patients atteints de MPOC. Ce modèle est exécuté chaque semaine, pour prédire si l'état d'exacerbation va être normal ou s'il va augmenter dans les jours qui suivent. Si un risque est détecté, un appel automatique alerte les patients. Selon cette étude, ce système a montré son efficacité sans équivoque, mais il lui manque encore une réponse appropriée et immédiate pour chaque patient.

b) (Yañez *et al.*, 2012) ont conclu dans leurs recherches que le taux de la respiration augmente significativement quelques jours avant une exacerbation, ce qui nécessite une hospitalisation. La figure 2.5 représente l'augmentation du taux de respiration dans le cadre d'une étude portant sur 89 patients.

Dans cette étude, la capacité discriminante de la fréquence respiratoire pour détecter l'exacerbation deux jours avant l'hospitalisation est représentée par *Area Under Receiver Operating Characteristic (AUROC)* = 76 %. L'obstacle principal de cette méthode est que le système oxygénothérapie à domicile utilisé n'est pas un moyen pratique (facile) pour surveiller la respiration. De plus, certains problèmes liés à l'oxygénothérapie pendant l'exercice, le sommeil, etc. sont mentionnés par (Sandberg et Fleetham, 2013). En outre, la capacité prédictive d'exacerbation est modérée avec AUROC = 76%.

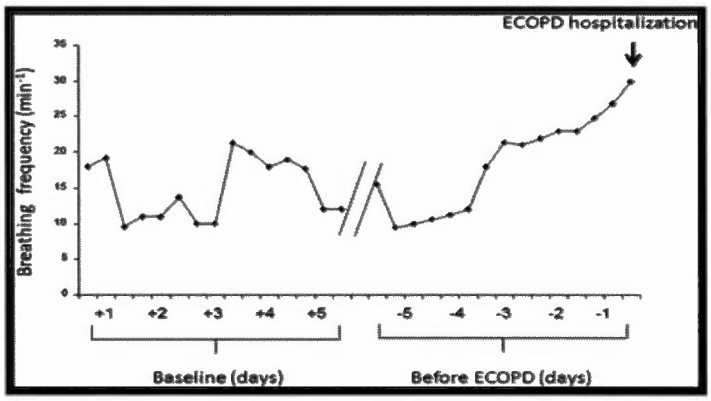


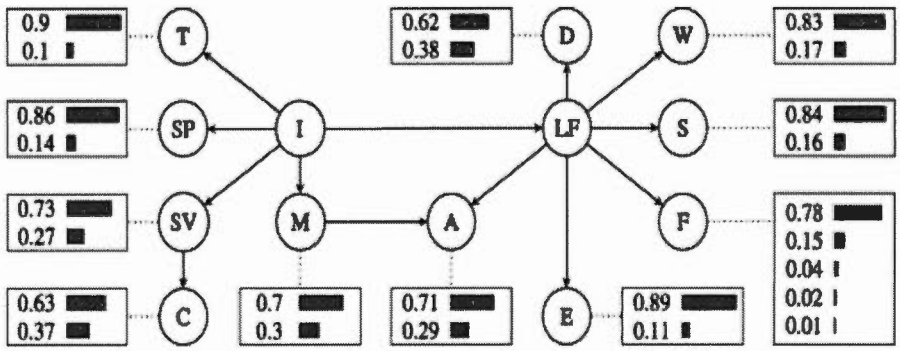
Figure 2.5 : Surveillance du taux de respiration avant l'hospitalisation pour cause d'exacerbation (Yañez *et al.*, 2012)

c) Pour détecter l'exacerbation chez les patients atteints de MPOC, (Van der Heijden *et al.*, 2013) ont utilisé le réseau bayésien (Figure 2.6) et l'ont implémenté sur un appareil mobile. La performance de leur système est prometteuse et son utilisation facile et pratique. Cela dit, ce système n'est pas autonome pour les raisons suivantes:

   1. Les attributs pertinents qui sont liés à l'apparition d'exacerbations sont identifiés préalablement par des experts (deux pneumologues).

2. Les experts ont réalisé le réseau de croyance ou dépendance entre les symptômes.

Ainsi, la structure de ce système est basée sur les comportements des experts. Ces comportements peuvent différer d'un expert à un autre et ne sont observables qu'en leur présence. Les données peuvent aussi être inexactes lorsqu'elles sont tirées d'un environnement continuellement modifié (Himes *et al.*, 2009), comme c'est le cas d'exacerbation, qui présente des symptômes non spécifiques ou inconnus (Lareau *et al.*, 2014; Rodriguez-Roisin, 2000b; Seemungal *et al.*, 1998). Alors, les méthodes automatiques qui remplacent les experts sont utiles (Witten et Frank, 2011) pour donner au système prédictif la capacité d'évoluer dans le temps.



A = activity, C = cough, D = dyspnea, E = exacerbation, F = FEV1, I =infection, LF = lung function, M = malaise, S = SpO2, SP = sputum purulence, SV = sputum volume, T = temperature and W = wheeze.

Figure 2.6 : Avis d'expert pour construire le réseau bayésien

d) En 2014, (Van der Heijden *et al.*, 2014) ont analysé une base d'apprentissage de la MPOC, dans le but de prédire la survenue d'exacerbation en utilisant *Temporal Node Bayesian Network (TNBN) (Arroyo-Figueroa, 1999), Expectation-Maximisation (EM) algorithm* et *block bootstrap* (pour

augmenter la quantité de données de la base d'apprentissage). La contribution fondamentale de cette étude est d'avoir employé l'ensemble de ces algorithmes pour apprendre un modèle temporel qui est représenté par *Temporal Node Bayesian Network (TNBN)*.

Ici, les auteurs souffrent de l'observation manquante (30%) et de la petite taille de l'échantillon (10 patients) dans la base d'apprentissage. En effet, avoir plus de données permet aux données de « parler d'elles-mêmes » au lieu de s'appuyer sur des heuristiques non prouvées et approximatives pour agrandir la base d'apprentissage (Flachaire, 2000). Toutefois, l'utilisation du *block bootstrap* pour agrandir la base d'apprentissage rend la base plus artificielle que si elle avait reproduit la réalité de la maladie.

e) (Ryynänen *et al.*, 2013), dans leur recherche, ont développé un réseau bayésien pour prédire la mortalité et de mesurer la *HRQoL (Health-Related Quality of Life*) chez les patients atteints de MPOC, dans le but d'améliorer leur qualité de vie. Le point faible de ce modèle est que la précision de la prédiction est faible avec AUROC = 69%, testé par 10-*Folds Cross Validation*. Aussi, cette étude ne traite pas de l'exacerbation.

2.4.3 Sélection des attributs pertinents de la MPOC

En raison de l'ambiguïté et du large nombre de facteurs qui peuvent influencer la détection d'exacerbations dans la MPOC, plusieurs systèmes informatiques se sont concentrés sur les facteurs prédictifs ou pertinents de la MPOC, qui sont énumérés ici-bas:

a) Récemment, (Himes *et al.*, 2009) ont identifié les attributs pertinents qui peuvent détecter la progression de l'asthme vers la MPOC, en utilisant le réseau bayésien. À cet effet, les auteurs se sont concentrés sur les nœuds qui modulent directement la MPOC. Ils ont utilisé *K2* (Cooper et Herskovits, 1992) pour identifier le réseau le plus probable à partir de données et l'heuristique *Markov Blanket* permettant de maintenir les relations qui influent directement la MPOC, et supprimer le reste des relations (Figure 2.7).

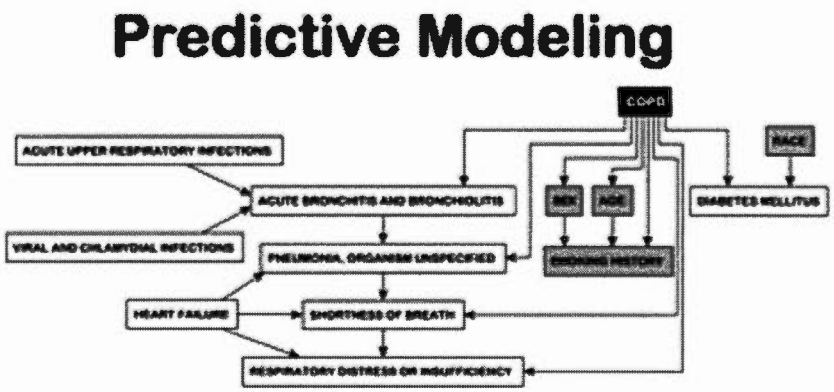


Figure 2.7 : Modèle prédictif basé sur le réseau bayésien, K2 et *Markov Blanket* (Himes *et al.*, 2009).

Le modèle prédictif de la MPOC chez les patients souffrant d'asthme est robuste et il a été évalué par 5-*Folds Cross Validation*, auprès de 9 349 patients. Au cours de cette étude, les chercheurs ont conclu que l'âge est la variable prédictive la plus forte, en utilisant la métrique d'évaluation AUROC, qui est égale à 81%. Cependant, cette étude ne prend pas en compte l'exacerbation.

b) En 2012, (Raghavan *et al.*, 2012) ont étudié la prévalence de la MPOC. Cette étude représente une partie de la *Canadian Obstructive Lung Disease (COLD) Study*. L'objectif de cette étude est d'identifier une combinaison de huit

éléments du questionnaire CAT[8] (ou *COPD Assessment Test*, un outil simple pour évaluer l'état de santé déficient dans la BPCO) avec d'autres prédicteurs connus de la MPOC (par exemple, les antécédents de tabagisme, l'âge, etc.) pour qu'ils puissent détecter l'exacerbation de façon aussi précise qu'avec la spirométrie, considérant que la spirométrie est très importante pour détecter cette maladie (Paliwal, 2012). Dans un autre sens, ces prédicteurs vont aider à identifier les patients à risque d'exacerbations de la MPOC pour lesquels le test de spirométrie est recommandé, en utilisant le modèle *Stepwise logistic regression.* La capacité prédictive de ce modèle est modérée (AUROC = 77 %).

c) (Amalakuhan *et al.*, 2012) compte sur le modèle *Random-Forest (RF)* pour déterminer les facteurs qui sont fortement corrélés avec les réadmissions à l'hôpital liées à la MPOC, et pour créer une application qui peut identifier les patients présentant un risque élevé de réadmission à l'hôpital dans l'année qui suit l'admission. L'indice de l'admission a été défini comme étant la première admission à l'hôpital à l'intérieur d'une période de douze mois consécutifs. L'AUROC du système est modéré et égal à 75%.

2.5 Analyse de travaux reliés et problèmes spécifiques

Dans cette revue de littérature, nous avons distingué trois types de travaux antérieurs qui sont fortement reliés au traitement de la maladie MPOC.

Premièrement, la recherche liée à la MPOC a insisté sur la facilitation de la communication à distance entre le patient et le médecin. Cependant, cette approche comporte plusieurs inconvénients: le manque d'automatisme au niveau de l'alerte, le

[8] *www.catestonline.org/english/index.htm*

manque de traitement concentré sur la détection précoce des exacerbations, le coût élevé suite à la visite répétée de l'infirmière et le nombre plus élevé de décès lorsqu'est utilisé ce type de système, etc.

Deuxièmement, nous avons étudié les systèmes qui font une interprétation automatique des symptômes observés. Dans ces systèmes, nous avons remarqué plusieurs points importants qui conduisent au manque d'un outil parfaitement automatique à la disposition des patients.

1. Le système ne peut pas donner une réponse immédiate à l'état de santé du patient (Halpin *et al.*, 2011).
2. L'outil utilisé ne permet pas de surveiller la respiration (Yañez *et al.*, 2012).
3. Le processus de prédiction manque d'autonomie, ce qui limite l'évolution du système dans le temps (Van der Heijden *et al.*, 2013).
4. La prédiction n'est pas prometteuse avec une précision finale modérée ou faible selon la métrique d'évaluation AUROC (Hanley et McNeil, 1982). Pour résumer, la performance des modèles proposés dans la littérature que nous avons abordés ici va comme suit : (Yañez *et al.*, 2012) = 76 %, (Ryynänen *et al.*, 2013) = 69 %, (Raghavan *et al.*, 2012) = 77 %, (Amalakuhan *et al.*, 2012) = 75 %.

Finalement, plusieurs chercheurs ont travaillé sur les facteurs prédictifs de la MPOC. Par exemple, (Himes *et al.*, 2009) utilise le réseau bayésien et le *Markov Blanket*, mais cette heuristique présente une relation forte avec le réseau bayésien (Sinoquet et Mourad, 2014), ce qui limite son utilisation avec d'autres méthodes d'apprentissage, comme le naïf bayésien ou l'arbre de décision. En outre, les autres travaux couvrant la sélection des attributs pertinents dans la MPOC, comme ceux de (Raghavan *et al.*, 2012) (Amalakuhan *et al.*, 2012), n'affichent pas une bonne performance, avec un AUROC de 77% et de 75 %, respectivement. Cette situation nous incite à utiliser de nouvelles méthodes de sélection des attributs pertinents pour le domaine de la MPOC.

Nous concluons que pour la plupart des projets antérieurs concernant le traitement de la MPOC, les attributs sont déjà catégorisés par les experts ou le classificateur lui-même (comme le naïf bayésien) traite les attributs continus sans avoir besoin de les catégoriser. Par exemple, les patients ont été divisé par l'expert en quatre groupes d'âge: les 18-44 ans, les 45-64 ans, les 65-74 ans et les 75 ans et plus (Himes *et al.*, 2009). Aussi, dans (Van der Heijden *et al.*, 2013), toutes les variables, sauf FEV_1 (*Forced expiratory volume*), sont binaires. FEV_1 est divisé en cinq catégories basées sur l'avis d'experts. (Ryynänen *et al.*, 2013) ont utilisé le naïf bayésien pour traiter les attributs continus sans aucune catégorisation. Le résultat dans cette étude (AUROC = 69 %) est un mauvais résultat selon l'évaluation de (Sandelowsky *et al.*, 2011). Ainsi, la catégorisation ou la discrétisation automatique en l'absence d'expert dans le contexte de la MPOC est une nouvelle contribution de notre recherche au domaine.

Concernant la sélection d'attributs pertinents et la discrétisation, ils sont les deux points essentiels du prétraitement des données pendant le processus de classification. Cela dit, pour garantir une bonne précision de prédiction, nous constatons que l'ordonnancement de ces méthodes est indispensable, parce que :

1. Selon notre recherche, aucun chercheur n'affirme que la discrétisation doive être effectuée avant la sélection d'attributs, ou le contraire.
2. Les méthodes de sélection des attributs pertinents peuvent supporter les attributs continus, comme le *CFSsubsetEval*. Il n'est donc pas nécessaire de faire la discrétisation en premier. Dans un autre sens, les méthodes de sélection peuvent ne pas nécessiter de discrétisation. Ainsi, on peut utiliser cette dernière seulement pour l'algorithme de traitement, comme le réseau bayésien ou l'arbre de décision.

En résumé, nous cherchons à concevoir et à valider un modèle de prédiction qui peut résoudre la plupart des problèmes que nous avons soulevés. Notre modèle va être

réalisé par une application contextuelle qui aide le personnel médical et/ou les patients à prédire les exacerbations de la MPOC. Ce modèle de prédiction présente les caractéristiques suivantes:

1. Autonome : c'est-à-dire qu'il est capable d'évoluer avec le temps sans nécessiter l'intervention d'experts médicaux.
2. Performant: il arrive à de bons résultats par la comparaison et la formulation d'un ordre des algorithmes dans le domaine d'apprentissage.
3. Raffiné: il sélectionne les attributs pertinents à l'exacerbation de la MPOC en utilisant de nouveaux algorithmes dans ce contexte. La sélection d'attributs pertinents facilite et simplifie l'interaction entre le patient et l'application contextuelle. Ainsi, la nouvelle utilisation de la méthode de discrétisation peut aussi faciliter l'interaction, en réduisant le nombre d'états pour chaque attribut.
4. Efficace : c'est-à-dire qu'il prend en compte l'ordonnancement des attributs pertinents dans notre application contextuelle. Ainsi, l'attribut le plus fort apparaît en premier et les autres en ordre décroissant d'importance, jusqu'à l'attribut le moins prédictif, qui apparaît à la fin de la liste de symptômes. Cette liste contient les attributs que le patient doit considérer. Cette contribution est nouvelle en général et nous l'appliquons à la MPOC. La démonstration de cette idée se trouve au chapitre IV, où nous faisons l'implémentation et la validation de notre application contextuelle.

Essentiellement, ces quatre caractéristiques seront prises en compte par notre application contextuelle, qui devra être capable de détecter l'exacerbation chez les patients de la MPOC. Pour développer cette application contextuelle, un modèle général qui inclut toutes les étapes nécessaires est proposé au début du chapitre III. Les trois premiers points (autonome, performante et raffinée) sont validés au chapitre III et le dernier point (efficace) au chapitre IV, parce qu'il est la dernière tâche réalisée par notre application contextuelle.

Pour généraliser le modèle proposé à différents domaines d'application, une procédure de validation a été accomplie sur huit bases d'apprentissage (*Cancer, Spectfheart, etc.*) à la fin du chapitre IV.

CHAPITRE III

MODÈLE DE SÉLECTION DES ATTRIBUTS PERTINENTS ET DE DÉTECTION DE LA MPOC; COMPARAISON DES ALGORITHMES D'APPRENTISSAGE

3.1 Introduction

Au chapitre I, nous avons analysé les algorithmes de discrétisation, de sélection des attributs pertinents et de raisonnement, ainsi que la modélisation des applications contextuelles. Le chapitre II décrit les systèmes informatiques existants qui permettent d'évaluer l'état du patient atteint de MPOC lors d'un diagnostic, en milieu médical ou lors d'un contrôle à domicile. Au chapitre II, nous avons énuméré plusieurs défis de ces systèmes, qui concernent principalement l'identification des attributs pertinents, l'automatisation de la construction du modèle prédictif, la faiblesse de précision de la prédiction et la sélection des attributs les plus pertinents (Section 2.5).

Pour répondre à ces défis, nous proposons dans ce chapitre un modèle prédictif en six étapes appliqué à la MPOC, qui utilise essentiellement les algorithmes présentés au chapitre I. Ce modèle aide le personnel médical à prendre efficacement des décisions et à surveiller le patient à domicile, en l'avisant automatiquement (selon ses observations) lorsqu'il est à risque d'exacerbation. Ce modèle de prédiction est validé par la métrique AUROC.

Lors du développement de ce modèle, les principales tâches se déroulent comme suit:

1. Nous représentons les entités de notre système automatique de suivi de la MPOC et leurs relations par une ontologie (Section 3.4).
2. Nous comparons quatre algorithmes de sélection des attributs pertinents pour choisir celui qui retourne le sous-ensemble d'attributs le plus pertinent. Cet algorithme (*Wrapper-BestFirst*) n'a pas encore été utilisé dans le contexte de la MPOC, selon notre revue de littérature (Chapitre II - Section 2.5).
3. Une comparaison a également été réalisée entre quatre algorithmes de discrétisation, pour choisir celui qui donne au classificateur la capacité discriminante la plus élevée. L'utilisation de ces algorithmes de discrétisation est nouvelle dans le contexte de la MPOC. Rappelons que les approches existantes requièrent la participation d'experts (Chapitre II - Section 2.5).
4. Nous comparons aussi les deux algorithmes de dépendance entre les attributs (TAN et K2) et utilisons TAN, car il améliore la capacité prédictive de la détection des exacerbations (Section 3.6.8).
5. Pour assurer une bonne performance de la prédiction des exacerbations, nous formulons un ordre entre les algorithmes de discrétisation et ceux de sélection, par rapport à différents algorithmes d'apprentissage (naïf bayésien, arbre de décision ou réseau bayésien), en utilisant la métrique *Area Under Receiver Operating Characteristic (AUROC)*. Soulignons que la performance n'était pas au rendez-vous dans la plupart des travaux existants (Yañez *et al.*, 2012) (Ryynänen *et al.*, 2013) (Raghavan *et al.*, 2012) (Amalakuhan *et al.*, 2012).

La comparaison de ces ensembles d'algorithmes permet de choisir un seul algorithme lors de chaque phase (discrétisation, sélection, dépendance, traitement). Dans ce chapitre, ces algorithmes seront proposés dans un modèle qui sert généralement à

sélectionner les attributs pertinents et à prédire l'exacerbation avec une grande efficacité, tout en remédiant aux problèmes énumérés à la section 2.5.

Nous proposons notre modèle à la section 3.2 et décrivons l'étape d'acquisition de la base d'apprentissage de la MPOC à la section 3.3. L'ontologie de la MPOC est présentée dans la section 3.4 et les métriques d'évaluations sont détaillées dans la section 3.5. À la section 3.6, notre modèle est validé par une comparaison entre les différentes phases, en nous basant sur la métrique d'évaluation AUROC et sur la base d'apprentissage de la MPOC.

3.2 Modèle proposé pour sélectionner les attributs pertinents et détecter l'exacerbation dans la MPOC

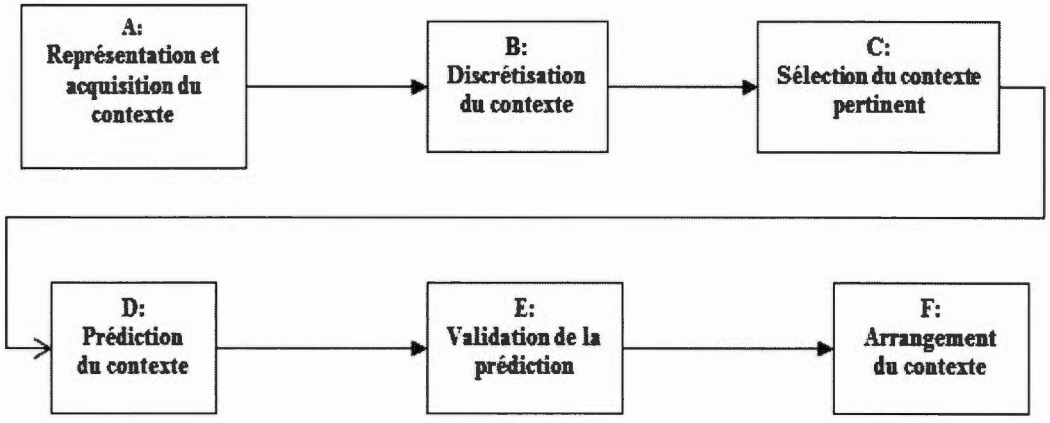


Figure 3.1 : Modèle de sélection des attributs pertinents et de détection des exacerbations

Le modèle proposé est représenté par six étapes (Figure 3.1). Son objectif ultime est d'être réalisé dans une application contextuelle (Figures 4.9 et 4.10). Principalement, ces étapes servent à sélectionner les attributs pertinents et à prédire l'exacerbation dans

la MPOC avec une haute efficacité. Dans le cas d'exacerbation de la MPOC, nous avons démontré l'efficacité de ce modèle (Figure 3.9) ainsi que sa généralité, en le testant sur huit bases d'apprentissage (*Cancer*, *Spectfheart, etc.)*(Chapitre IV, Section 4.6).

Ces étapes sont décrites comme suit :

Étape A : L'acquisition de la base d'apprentissage est la première étape et la plus difficile du travail lié à l'apprentissage automatique. En raison de la difficulté d'obtenir cette base pour tous les éléments contextuels disponibles dans un environnement, sa présence au début du processus peut nous aider à nous représenter comment notre scénario principal peut être réalisé.

Étape B : L'algorithme de discrétisation supervisée est appliqué (*Fayyad & Irani's MDL*).

Étape C : Les attributs pertinents sont sélectionnés à l'aide de *Wrapper-BestFirst*.

Étape D : Le modèle de prédiction est créé à l'aide du réseau bayésien, en utilisant TAN.

Étape E : La performance du modèle prédictif est évaluée à l'aide de *Receiver Operating Characteristic (ROC) curves*, avec 10-*Folds Cross Validation* stratifié.

Étape F: Le contexte pertinent doit être arrangé de façon descendante pour qu'on puisse utiliser les plus pertinents en cas d'urgence. Cette étape est exposée au chapitre IV.

Ce modèle est validé aux sections suivantes de ce chapitre. Au chapitre IV, il est réalisé dans une application contextuelle qui prédit l'exacerbation dans la MPOC avec une grande efficacité.

Les deux sections suivantes (3.3 et 3.4) présentent la première étape de ce modèle : l'acquisition et la représentation du contexte.

3.3 Acquisition de la base d'apprentissage

Pour prédire l'exacerbation de la MPOC, nous avons travaillé avec une base d'apprentissage composée de 61 attributs et de 1985 patients. Cette base est disponible sur la page Web GitHub (Rajasekaran, 2015) de la source CrowdANALYTIX (Figure 3.2). CrowdANALYTIX est un site Web qui organise régulièrement des concours de prédiction par des données scientifiques, en s'appuyant sur l'intelligence collective d'une communauté de plus de 14 078 scientifiques à travers le monde (Analytix, 2015).

Le principal défaut de cette base d'apprentissage est que les noms d'attributs sont cryptés pour protéger la confidentialité des patients. Ainsi, l'étiquette (nom) de chaque attribut est représentée par une grande catégorie. Par exemple, les attributs qui représentent la catégorie *Lung Function* (comme FEV[9], FEV_{max}, etc.) sont nommés comme LungFun1, LungFun2, etc. Un échantillon de cette base est donné comme suit :

[9] Forced expiratory volume

DisStage2	LungFun1	LungFun2	LungFun3	LungFun4	LungFun5	LungFun6	LungFun7	LungFun8	LungFun9	LungFun10	LungFun11	L
0	0.958806818	0.366689053	0.784810127	0.545863309	0.456096694	0.3312	0.521297823	0.345814051	0.536017529	0.462908557	0.394909313	
0.857142857	0.985795455	0.41773002	0.734177215	0.517985612	0.465872734	0.33792	0.426650473	0.272191838	0.535195837	0.464744767	0.396433471	
0.142857143	0.947443182	0.462726662	0.658227848	0.632793765	0.542125844	0.46848	0.354992626	0.227387463	0.618734593	0	0.43011736	
0.714285714	0.534090909	0.306917394	0.481012658	0.292266187	0.336118023	0.268	0.239437842	0.053639041	0.285127362	0.27745134	0.238987959	
1	0.515625	0.343854936	0.392405063	0.333932854	0.463384287	0.3776	0.302767416	0.086453513	0.339085182	0.439037826	0.410150892	
0.714285714	0.447443182	0.280725319	0.392405063	0.315347722	0.442765731	0.35344	0.306497788	0.078249895	0.230895645	0	0.175430575	
0	0.974431818	0.520483546	0.582278481	0.336630695	0.345183079	0.264	0.371475666	0.097602019	0.360996987	0.3540213	0.330894681	
0.142857143	0.583806818	0.370047011	0.405063291	0.375	0.440810523	0.4064	0.245163529	0.062263357	0.411120241	0.493756886	0.574607529	
0.142857143	0.899147727	0.441235729	0.620253165	0.708333333	0.69303235	0.55904	0.532922703	0.263357173	0.718707204	0.729159016	0.65096784	
0.142857143	0.678977273	0.340496978	0.582278481	0.252098321	0.282083185	0.18784	0.291749805	0.059318469	0.264311148	0.258171135	0.238073464	
0	0.754261364	0.276695769	0.721518987	0.443045564	0.392641308	0.2856	0.432723172	0.206983593	0.468638729	0.403415351	0.356043286	
1	0.569602273	0.314304903	0.518987342	0.325839329	0.39210807	0.28064	0.218877418	0.123264619	0.365379348	0.394785163	0.347355586	
0.714285714	0.224431818	0.179986568	0.303797468	0.21882494	0.378954852	0.30096	0.262080333	0.057425326	0.255272528	0.378993757	0.354214297	

Figure 3.2 : Échantillon de la base d'apprentissage en format Excel

Selon CrowdANALYTIX, la signification de chaque catégorie est décrite comme suit. *Exacer* est la classe d'attributs et elle est catégorique, avec 1 = exacerbation et 0 = non-exacerbation. *Démographie* est une catégorie qui contient six attributs (âge, genre, etc.) discrets et binaires. *DiseaseStage* contient deux attributs discrets pour mesurer la sévérité de la MPOC. *LungFunction* contient 20 attributs continus dérivés du spiromètre. *DiseaseHistory* contient dix attributs discrets qui sont l'histoire de quelques symptômes de la MPOC, par exemple si le patient a déjà été hospitalisé pour un trouble (*illness*[10]) respiratoire et si oui, combien de fois. *OtherLungDisease* comprend 13 attributs discrets, qui sont les antécédents des problèmes pulmonaires de chaque patient, comme l'histoire de toux, l'asthme, etc. *RespiratoryQuestionnaire* se compose de cinq attributs continus mesurant la qualité de vie chez les patients atteints de MPOC. *Smoking* cette catégorie contient quatre attributs, trois continus et un discret.

3.4 Comparaison des modèles de représentation de contexte

3.4.1 Choix de la représentation de contexte

[10]Réfère ici aux sentiments qui peuvent survenir avec une maladie. Ex : la fatigue, la faiblesse, l'inconfort, la confusion, etc. Tony Ingram, «Disease vs. Illness», (2012)

Dans cette partie, nous développons une ontologie qui représente les entités principales et leurs relations, dans un scénario où on surveille les exacerbations des patients souffrant de MPOC par le moyen d'une application contextuelle. Notre choix d'ontologie est fondé sur plusieurs raisons. Voyons d'abord le tableau 3.1 (Strang et Linnhoff-Popien, 2004), qui présente l'enquête de modélisation la plus citée.

Tableau 3.1 : Comparaison entre les approches de modélisation du contexte (Strang et Linnhoff-Popien, 2004)

Critères / Modèles	Dc	Pv	Qua	Inc	For	App
Key-value	-	-	-	-	-	+
Schéma de balisage	+	++	-	-	+	++
Orienté Object	++	+	+	+	+	+
Modèle Logique	++	-	-	-	++	-
Ontologie	++	++	+	+	++	+

(++) score élevé; (+) score acceptable; (-) score médiocre

Cette évaluation (tableau 3.1) est faite pour évaluer l'efficacité des modèles de représentation dans le but de répondre aux exigences des applications contextuelles. Selon (Strang et Linnhoff-Popien, 2004), le meilleur modèle de représentation est celui qui est capable: i) de s'adapter aux systèmes dynamiquement distribués (*Distributed Composition [Dc]),* ii) d'offrir la possibilité de valider la structure du modèle crée (*Partial Validation [PV]),* et iii) de soutenir la qualité et la richesse des informations (*Richness and Quality of information [Qua]*) capturées (par exemple, l'ontologie garantit la gestion durable de l'information modélisée en utilisant la structure hiérarchique). (Strang et Linnhoff-Popien, 2004) ajoute à ce propos que le meilleur modèle de représentation contextuelle est celui qui gère les informations incomplètes et ambiguës recueillies par les capteurs (*Incompleteness and Ambiguity [Inc]*), et de construire de manière retraçable les entités de contexte pour offrir une compréhension partagée (*Level of Formality [For]*). De plus, ce modèle de représentation doit être

compatible avec l'infrastructure de l'informatique ubiquitaire (*Applicability to existing environments [app]*) (Emitzá, 2010) (Kabir, 2016).

Toutes ces exigences placent l'ontologie en tête de liste des modèles de représentation de contexte, selon (Strang et Linnhoff-Popien, 2004). En effet, le modèle d'ontologie continue de se démarquer jusqu'à aujourd'hui, du point de vue de son expressivité (Mansoor *et al.*, 2008), de sa sophistication (Baldauf *et al.*, 2007), de la représentation complexe qu'elle permet (Khattak *et al.*, 2014), de son interopérabilité (Kamberov, 2016) et de la possibilité de la réaliser dans plusieurs contextes (Kabir, 2016).

Bien que la pertinence de l'ontologie soit évidente, dans notre cas, le raisonnement d'ontologie est difficile à appliquer parce qu'il est basé sur une logique de premier ordre, ou « logique temporelle » (Gu *et al.*, 2005). Par exemple, le langage d'ontologie *OWL* est très expressif par rapport à d'autres langages, comme le RDFS. Cependant, *OWL* ne peut pas incorporer des informations probabilistes pour gérer les événements incertains. Or, la prise en compte de l'incertitude est importante pour détecter l'exacerbation dans la MPOC (Chapitre II). Dans le même contexte, l'incertitude est la raison pour laquelle (Gu *et al.*, 2004) ont étendu leur modèle d'ontologie à un modèle probabiliste, en définissant des marges de probabilité supplémentaires. En pratique, (Gu *et al.*, 2004) ont utilisé l'ontologie pour construire leur réseau de croyance, en se basant sur des experts.

Dans notre cas, l'ontologie ne contribue pas à faire de raisonnement, mais juste il nous permet de comprendre notre contexte (ex : symptômes et signes de la MPOC), et les différentes techniques utilisées (ex : les algorithmes d'apprentissage), ainsi que la relation entre eux, pour arriver à notre objective (prédiction d'exacerbation de la MPOC). En plus, l'ontologie nous serve à comprendre les trois points suivants :

1. Notre ontologie représente les relations entre la plupart des entités (contexte, techniques, location, médecin, patient) qui constituent un système intelligent

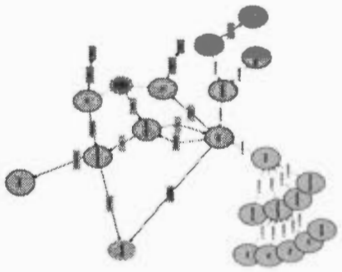
surveillant les patients atteints de MPOC. En nous basant sur cette ontologie, nous décrivons un scénario détaillé pour le patient de la MPOC, dans le but de détecter ses exacerbations.

2. Cette ontologie illustre l'autonomie de notre application contextuelle finale.
3. L'ontologie va nous permettre de faire évoluer facilement notre modèle de prédiction dans le futur.

3.4.2 Description de l'ontologie de la MPOC

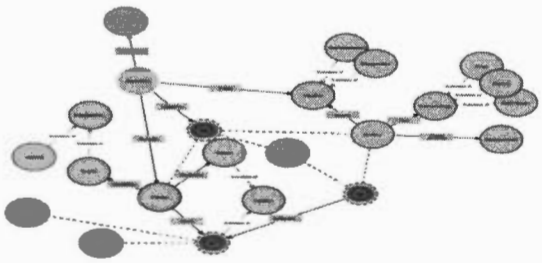
Selon notre revue de littérature, il n'existe à ce jour aucune ontologie qui réalise un *framework* contextuel pour un système détectant l'exacerbation dans la MPOC.

La figure 3.3 montre l'ontologie de notre système de suivi de la MPOC (l'élargissement de cette figure se trouve à l'appendice B, aux figures B.1, B.2, B.3). Dans cette ontologie, les cercles définissent les classes ou les concepts, et chaque flèche définit une relation entre ces classes. Les rectangles sont ajoutés pour afficher les noms des relations. En outre, le symbole ∪ est utilisé pour symboliser l'union de plusieurs concepts et le symbole ∩ pour signifier l'intersection entre eux.



Partie 1 : L'ontologie de la maladie MPOC (signes et symptômes)

Partie 3 : L'ontologie des algorithmes utilisées pour l'application contextuelle



Partie 2 : L'ontologie qui représente le scenario du patient en utilisant notre application contextuelle

Figure 3.3 : Ontologie décrivant toutes les entités possibles d'une application qui détecte l'exacerbation dans la MPOC (« voir appendice B»)

3.4.3 Scénario de l'ontologie

Considérons le patient fictif Bob souffrant de MPOC et résidant à son domicile. Il serait intéressant qu'une application contextuelle efficace lui offre la capacité de prendre soin

de lui-même en prédisant l'exacerbation ou la crise pulmonaire avant qu'elle ne se produise.

À la figure 3.3 (Partie 2) ou figure B.2, nous avons défini que l'application de prédiction doit être installée sur un serveur, pour en assurer l'accès au personnel médical et au patient, à l'hôpital ou à la maison. Cette application aide le personnel médical à mesurer efficacement les attributs pertinents de la MPOC. En outre, cette application peut aussi surveiller le patient à domicile. Une fois que ce dernier est avisé d'un risque d'exacerbation par cette application, et s'il n'est pas en mesure de contrôler son état à la maison, il se déplace à la clinique ou à l'hôpital. Dans ce cas, le personnel médical peut utiliser l'application pour analyser les observations détectées par le patient et ainsi mieux comprendre son état sans lui poser beaucoup de questions. Le personnel médical peut aussi utiliser l'application avec un nouveau patient pour l'aider à prendre une décision. Dans ce scénario, la classe « docteur » fait des examens supplémentaires, comme des radiographies, PaCO2[11], etc., pour prendre une décision finale ou pour confirmer le résultat obtenu par l'application.

De plus, dans cette ontologie, l'application obtient des données d'apprentissage tirées de l'historique des patients. Ces historiques sont placés à l'hôpital dans un serveur qui contient la base d'apprentissage figure 3.3 (Partie 1) ou figure B.1 Cette base est évolutive et incrémente au fil du temps les profils des patients. Chaque profil a un identificateur unique[12] et chaque ligne à ajouter dans la base d'apprentissage est assurée par le docteur. L'ensemble de *SubClasses* relié à la classe profil dans la figure B.1, représente les attributs (symptômes) de jeu de données que nous voulons acquérir pour détecter l'exacerbation. Nous avons obtenu cette base d'apprentissage sur le site de CrowdANALYTIX (Analytix, 2015). Les détails à propos de ces données sont décrits

[11] Soit la pression partielle en dioxyde de carbone dans le sang artériel.

[12] Unique, c'est-à-dire *Functional* dans la syntaxe d'OWL.

à la section 3.3. La figure 3.3 (Partie 3) ou figure B.3 illustre l'ensemble des algorithmes qui peuvent être utilisés pour créer le système de prédiction. Dans la même figure, l'union signifie qu'on a plusieurs choix d'algorithmes à utiliser. Habituellement, l'efficacité des algorithmes d'apprentissage automatique dépend de la base d'apprentissage. Il est donc utile de faire la comparaison entre ces algorithmes à chaque base d'apprentissage utilisée. Nous faisons notre propre comparaison à la section 3.6. Ces ensembles d'algorithmes sont expliqués au chapitre I (Sections 1.4 à 1.12). Ces trois ontologies dans la figure 3.3 (partie 1, 2 et 3) serve l'application contextuelle, pour cela l'entité de ce dernier est le point commun entre les trois ontologies (l'entité avec une frontière jaune). D'une autre façon, partie 1 apprend l'application, partie 2 explique le scénario d'utilisation de cette application par le patient ou les personnels médicaux, et la partie 3 met en évidence les algorithmes qu'il faut les utiliser pour produire une application contextuelle a plusieurs caractéristiques différentes de ce qui existe (section 2.5).

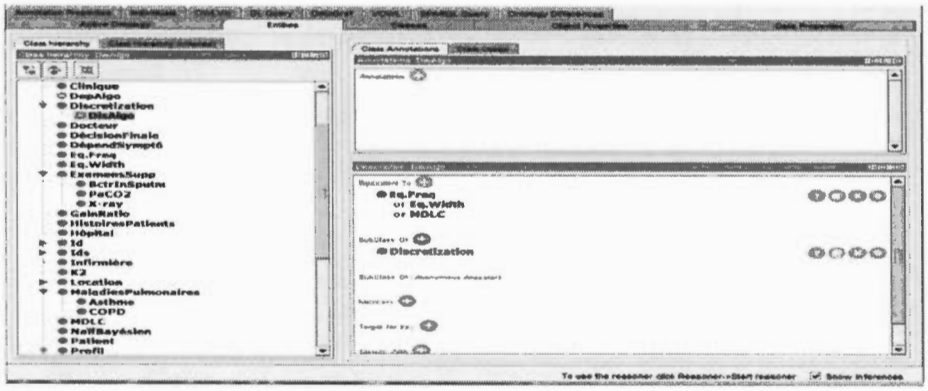
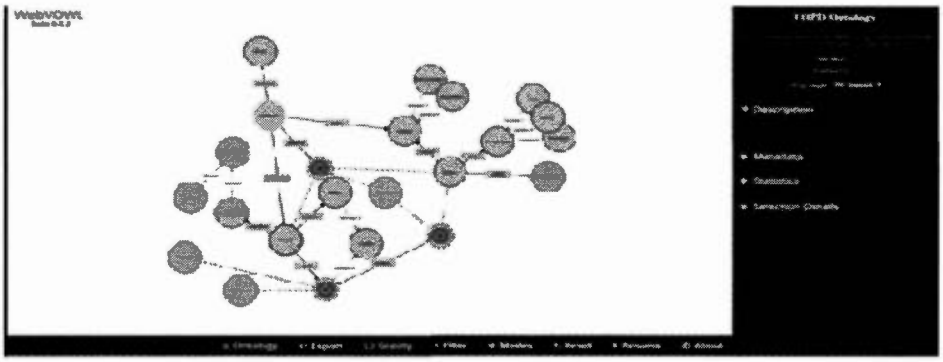


Figure 3.4 : Interface du *Protégé* pendant la création de l'ontologie MPOC

Cette ontologie nous permet de comprendre notre contexte et de mettre en évidence tous les concepts nécessaires qui peuvent être utilisés ou qui entrent ligne de compte lors de la création de notre application contextuelle. L'ontologie est créée par l'outil

Protégé (Figure 3.4) et utilise l'*Ontology Web Language (OWL)*. Le *Fichier*.owl qui est généré par *Protégé* peut être visualisé à la figure 3.5.

En conclusion, cette ontologie démontre l'importance de l'étape de la représentation dans le modèle que nous proposons, pour comprendre notre contexte. En outre, l'ontologie propose un système intelligent complet pour surveiller le patient souffrant de MPOC. Mais, dans les prochaines pages nous nous concentrons particulièrement sur l'infrastructure logique de ce système. Cette infrastructure est basée sur les cinq étapes restantes du notre modèle, qui sont précédemment décrites (Figure 3.1 B, C, D, E et F) en utilisant une base d'apprentissage fixe. Dans ce contexte, nous proposons d'abord des métriques d'évaluation de l'apprentissage automatique.



Source: http://vowl.visualdataweb.org/webvowl

Figure 3.5 : Interface du site Web *VOWL* permettant de visualiser l'ontologie de la MPOC

3.5 Métriques d'évaluations dans l'apprentissage automatique

L'évaluation de la performance des algorithmes d'apprentissage est un aspect fondamental de l'apprentissage automatique (Hall, 1999). Souvent, cette évaluation est basée sur la matrice de confusion suivante :

La décision du classifieur

	Décision Positifs	Décision Negatifs	
Étiquette Positifs	**Vrai Positifs, TP**	**Faux Négatifs, FN**	**En réalité**
Étiquette Négatifs	**Faux positifs, FP**	**Vrai Négatifs, TN**	

Figure 3.6 : Matrice de confusion pour la classification binaire

Les détails de cette matrice sont donnés comme suit :

- La décision positive : le classificateur prédit que le patient est susceptible d'avoir une exacerbation. La décision négative signifie le contraire.
- La mesure de vrais positifs (TP) : les cas qui sont correctement classés comme positifs.
- Les faux positifs (FP) : les cas qui sont en réalité négatifs, mais que le classificateur a considérés positifs.

Vrai / faux négatifs (TN et FN) sont respectivement comme TP et FP (à la place de positif, négatif).

Plusieurs métriques d'évaluation se basant sur la matrice de confusion de la figure 3.6 ont été utilisées dans la communauté informatique pour comprendre la performance de prédiction d'un classificateur. L'évaluation de la classification est souvent basée sur le **TPR** (*TruePositiveRate Or* sensitivity *Or recall*) = TP / (TP +FN), **FPR** (*FalsePositiveRate Or 1-Specificity*) = FP/ (FP +TN), **ROC** (*The receiver operating characteristic curve)*= une courbe de *TPR* en fonction de *FPR* aux différents points *cutoff* et qui se résume par *Area Under Curve* (AUROC ou AUC), **Accuracy** = (TP +TN) / (TP+TN+FP+FN), et **F-measure** = 2 (Recall * Précision) / (Recall + Précision), où la précision = TP / (TP +FP).

Plusieurs travaux de recherche ont considéré l'*accuracy* comme une métrique essentielle pour mesurer la capacité prédictive d'un classificateur (Wang et Valtorta, 2012), mais nous avons trouvé que cette métrique présente des problèmes dans quelque cas. La faiblesse de cette méthode est expliquée dans le site Web (Tryo.labs, 2013).

D'autre part, parmi les métriques d'évaluation existantes (*Recall, Specificty, F-measure, ROC, etc.),* le graphe ROC est souvent utilisé pour évaluer la prise de décision médicale dans les recherches biologiques en général (Himes *et al.*, 2009; Hoot et Aronsky, 2005; Sanders et Aronsky, 2006a; Van den Berge *et al.*, 2012; Van der Heijden *et al.*, 2013; Van der Heijden *et al.*, 2014). De plus, l'AUROC est largement adopté pour l'évaluation de l'apprentissage supervisé (Fawcett, 2004). Pour cette raison, les prochaines expérimentations vont être basées sur cette métrique. L'utilisation d'une seule mesure d'évaluation peut nous aider à rendre les expérimentations plus compréhensibles et synchronisées. La métrique AUROC se résume par les valeurs suivantes: excellent = 0,90 à 1, bonne = 0.80 - 0.90, acceptable = 0.70 - 0.80, pauvre = 0.60-0.70 et Fail = de 0.50 à 0.60 (Sandelowsky *et al.*, 2011).

Pour appliquer la métrique AUROC à un classificateur, il faut que la base d'apprentissage se décompose en deux parties: *TrainSet* et *TestSet*. Le *TrainSet* sert *à* entrainer le système d'apprentissage et le *TestSet* à l'évaluer. Dans cette étude, chaque observation (ligne) est un patient, et un seul patient peut apparaître uniquement dans le *TrainSet* ou dans le *TestSet*, jamais dans les deux. De cette façon, nous mesurons la capacité prédictive de la méthode d'apprentissage. Cette évaluation s'appelle la fiabilité du modèle (*reliability*) (Ryynänen *et al.*, 2013). Cependant, la partition binaire (*Test-Train*) n'est pas satisfaisante, car elle ne donne pas la chance à chaque point des données d'être dans la partie de validation. Les chercheurs parlent de *Cross Validation* pour remédier à ce problème.

Dans le *Cross Validation*, l'ensemble des données est divisé en k sous-ensembles d'attributs. Dans chaque validation, un de k sous-ensembles est utilisé comme test, et

les autres k-1 sous-ensembles sont utilisés pour l'entraînement. L'opération se répète k fois pour que chaque sous-ensemble soit utilisé exactement une fois comme test.

(Kohavi, 1995a) a proposé 10-*Folds Cross Validation* stratifiée comme un meilleur choix. Les *Folds* sont stratifiées, c'est-à-dire qu'elles contiennent à peu près la même proportion d'étiquettes (*labels*) que les jeux de données originaux. Weka utilise par défaut le *Cross Validation* stratifié et est représenté à la figure 3.7. Pendant notre expérimentation, nous allons utiliser le système logiciel Weka pour comparer les algorithmes utilisés. Plus de détails seront donnés à ce propos dans les prochaines sections

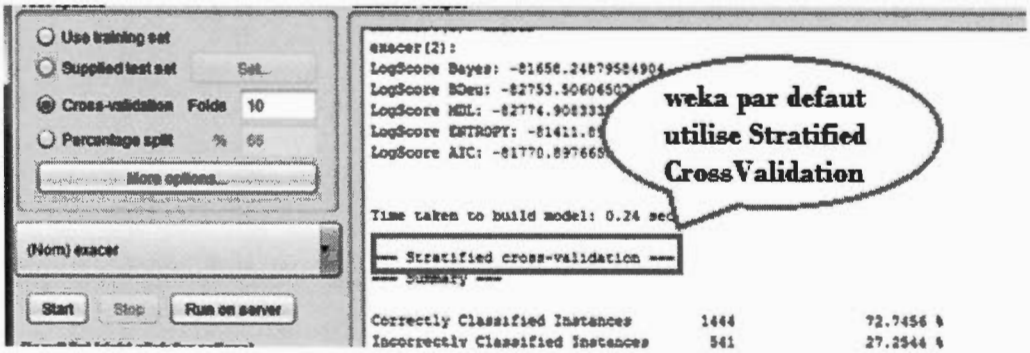


Figure 3.7 : Weka utilise par défaut le *Cross Validation* stratifiée

3.6 Sélection des algorithmes et résultats obtenus

Dans cette section, nous décrivons l'évaluation expérimentale permettant de choisir une chaîne d'algorithmes pouvant prédire l'exacerbation de la MPOC de manière performante et autonome, en sélectionnant les attributs pertinents.

Pour réaliser ces trois derniers buts (performance, autonomie et choix pertinent des attributs), nous comparons quatre algorithmes d'apprentissages traitant le réseau bayésien, l'arbre de décision (C4.5 et ID3) et le naïf bayésien, pour choisir celui qui est

le plus performant lorsque combiné avec un algorithme performant de discrétisation (*EqualWidthDiscretization (EFD)*, *EqualFrequencyDiscretization (EFD)*, *MDL Fayyad et Irani's MDL* ou *Kononenko's MDL*), et un autre de sélection des attributs pertinents (*CFSsubsetEval*, *GainRatioAttributEval ou Wrappers*).

En outre, nous avons investigué sur l'ordre de discrétisation et sur la sélection d'attributs pertinents. Dans notre cas, la discrétisation est précédée par la sélection, pour aider à améliorer la performance de prédiction du modèle de classification choisi. Ces ensembles de méthodes n'ont encore jamais été comparés dans le contexte de la détection de l'exacerbation dans la MPOC (Chapitre II – Section 2.5).

Pour réaliser cette comparaison, nous avons examiné l'ensemble des approches permettant d'accomplir l'apprentissage par le système logiciel Weka (Weka, 2011). Weka prend en charge plusieurs tâches de l'exploration de données, comme le prétraitement (par exemple, la discrétisation et la sélection des attributs pertinents), la classification, la visualisation des métriques d'évaluation (par exemple, AUROC) et les méthodes pour créer le réseau de dépendance entre les attributs dans le cas du réseau bayésien.

3.6.4 Configuration de Weka

Notre base d'apprentissage se compose de 61 attributs mixtes (continus et discrets) et de 1985 patients atteints de MPOC. Dans la première expérimentation, nous n'appliquons pas les méthodes de prétraitement sur la base d'apprentissage (sélection et discrétisation). En d'autres termes, nous utilisons la configuration par défaut du Weka pour montrer l'influence importante de ces deux méthodes sur la performance de la prédiction.

La configuration par défaut des classificateurs dans Weka est donnée comme suit:

id3 : est l'algorithme qui construit l'arbre de décision. Cet algorithme ne traite pas les attributs continus (Quinlan, J. Ross, 1986).

J4.8 : Ou encore l'algorithme C4.5, qui est l'extension d'ID3. L'algorithme de l'arbre de décision C4.5 (Quinlan, J. Ross, 1996) peut traiter les attributs continus en divisant intérieurement ces attributs en intervalles discrets (deux intervalles ou discrétisation binaire) pendant la construction de l'arbre de décision (Witten et Frank, 2011).

Naïf bayésien: Par défaut, le classificateur naïf bayésien suppose que les attributs suivent une distribution normale (*normally distributed*) pour traiter les attributs continus (Weka.net). En outre, le classificateur naïf bayésien accepte les attributs discrets (Witten et Frank, 2011).

Réseau bayésien : En général, le réseau bayésien ne supporte que les attributs discrets (Bouckaert, May 2008). Cependant, dans la configuration par défaut de Weka Explorer, l'adaptation des attributs continus est supportée par une discrétisation binaire pendant la construction des *Conditional Probability Tables (CPTs)*. Par défaut, l'algorithme qui identifie le réseau de dépendance dans le réseau bayésien est K2, en considérant 1 comme le nombre maximum de parents.

Dans notre expérimentation, la première étape consiste à comparer la capacité prédictive ou la performance de l'ensemble des algorithmes d'apprentissages en utilisant la configuration par défaut de Weka pour détecter l'exacerbation de la MPOC.

Tableau 3.2: Comparaison des différents algorithmes d'apprentissage avec la configuration par défaut de Weka

Test A	Configuration par défaut de Weka avec des attributs mixtes (continus et discrets)			
	61 attributs et 1985 patients en utilisant Weka			
10 - *Cross Validation*	Bayes		Arbre de décision	
ML / Métrique	Naïf bayésien	Réseau bayésien (K2)	ID3	C4.5
AUROC	**0.768**	**0.768**	-	0.564

Le signe (-) dans les tableaux qui représentent le résultat, signifie que l'algorithme ne supporte pas le cas.

Le Test A (Tableau 3.2) indique que le résultat du réseau bayésien correspond au naïf bayésien. Nous croyons ce résultat est possible en raison de la structure de dépendance de réseau bayésien utilisant K2 avec nombre de parents égal à 1 (Figure 3.8), qui correspond au naïf bayésien (attributs indépendants) dans la configuration par défaut de Weka. Il est à noter que le naïf bayésien utilise la distribution normale pour gérer les attributs continus, au lieu de faire la discrétisation binaire intérieurement comme le fait le réseau bayésien. L'arbre de décision avec l'algorithme C4.5 utilisant intérieurement la discrétisation binaire a échoué à détecter l'exacerbation dans la MPOC (AUROC = 56.4 %). Ainsi, ID3 ne supporte pas l'hétérogénéité (continue et discrète) des attributs.

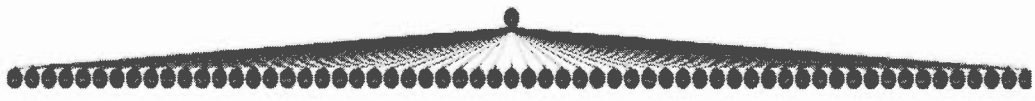


Figure 3.8 : Structure du réseau bayésien réalisée avec l'algorithme K2, qui suppose que le nombre de parents de chaque attribut égal à 1, par défaut

Dans le test A, nous constatons que le résultat obtenu (AUROC = 76.8 %) ne peut rivaliser avec ceux obtenus par d'autres travaux, comme ceux de (Raghavan *et al.*, 2012), qui obtiennent un AUROC = 77 %. De cette façon, nous considérons la méthode de discrétisation et de sélection des attributs pertinents lors des prochaines expérimentations pour améliorer la performance de la prédiction. En plus, ces deux méthodes comportent plusieurs avantages dans le contexte de la MPOC, qui sont décrits dans les deux prochaines sections.

3.6.5 Comparaison des algorithmes de sélection des attributs pertinents

Théoriquement, plus d'attributs devrait se traduire par plus de puissance et de pouvoir discriminant. Cependant, l'expérimentation pratique avec des algorithmes d'apprentissage automatique a montré que ce n'est pas toujours le cas (Hall, 1999). Dans ce travail, les objectifs en ce qui concerne la sélection d'attributs pertinents sont:

a. D'améliorer la performance de la prédiction.
b. De déterminer les attributs les plus pertinents pour détecter l'exacerbation de la MPOC, en utilisant des nouvelles méthodes.
c. De faciliter l'interaction du personnel médical avec notre outil de prédiction décrit au chapitre IV.

De plus, selon plusieurs chercheurs, la réduction de la dimension des données par la suppression des attributs inappropriés peut, entre autres, accélérer le temps d'apprentissage et faciliter la découverte de nouvelles connaissances dans les données (Guérif, 2006).

Dans cette section, nous comparons quatre algorithmes de sélection des attributs pertinents à l'aide de *Wrappers* et *Filters*. Cette comparaison aide à trouver le sous-

ensemble d'attributs le plus pertinent, soit celui qui a la capacité discriminante la plus élevée, afin de faire la distinction entre les états d'exacerbation et de non-exacerbation dans la classe d'attribut.

Au test B, le réseau bayésien et le C4.5 utilisent intérieurement la discrétisation binaire pour traiter les attributs continus, en se basant sur la configuration par défaut de Weka. ID3 ne divise pas intérieurement les attributs continus pendant la construction de l'arbre de décision.

Tous les détails concernant *CFSsubsetEval*, *GainRatio*, *Wrapper-BestFirst* et *Génétique*, ainsi que leurs configurations se trouvent au chapitre I (Section 1.4.2).

Tableau 3.3 : Comparaison des différents algorithmes d'apprentissage en appliquant les méthodes *Filters* et *Wrappers*.

La Sélection des Attributs pertinents	**Test B**	Les attributs initiaux sont mixtes (continus et discrets)						
		61 attributs et 1985 patients en utilisant Weka						
	10 - *Cross Validation*	Bayes				Arbre de décision		
	AUROC	Naïf bayésien		Réseau bayésien (K2)		ID3	C4.5	NAP
Filters	CFSsubsetEval	0.779		0.780		-	0.604	18
	Gain Ratio Attribute Eval	0.757		0.743		-	0.640	12
Wrappers	BestFirst	**0.794**	19	0.788	12	-	0.668	38	-
	Génétique	0.789	26	0.784	31	-	0.673	9	-

Au tableau 3.3 ci-dessus, NAP est l'abréviation du nombre d'attributs pertinents. Cependant, la partie droite des cellules qui sont découpées par une ligne pointée représente le nombre d'attributs pertinents sélectionnés pour chaque classificateur, en se basant sur les deux algorithmes de recherches *BestFirst* et *Génétique* dans *Wrappers*. La partie gauche représente le score de l'AUROC. Afin de préciser ces idées, *Wrappers* suppose que le classificateur lui-même évalue un sous-ensemble d'attributs choisi par la méthode de recherche. Par contre, pour les méthodes *Filters*, la méthode de sélection ne dépend pas du classificateur.

Grâce au tableau comparatif (Tableau 3.3) ci-dessus, nous observons que le naïf bayésien présente le score le plus élevé en utilisant *Wrapper-BestFirst*. Le score a augmenté de 2.6 % (AUROC = 0.794 %) par rapport au test A, et le nombre d'attributs a diminué à 19.

À la section suivante, nous mesurons l'influence de quatre méthodes de discrétisation (EWD, EFD, *Fayyad & Irani's MDL*, et *Kononenko's MDL*) sur la capacité prédictive des classificateurs, en utilisant les 61 attributs. Ensuite, nous combinons les méthodes de sélection avec celles de discrétisation dans les deux sens (discrétisation → sélection et sélection → discrétisation), dans le but de choisir l'ordre le plus performant pour la prédiction.

3.6.6 Comparaison et utilité des méthodes de discrétisation

La discrétisation est l'étape essentielle de prétraitement des données pour les algorithmes d'apprentissage qui traitent uniquement les données discrètes. En outre, dans ce travail, nous utilisons la discrétisation pour atteindre les objectifs suivants :

a. Améliorer la capacité prédictive des classificateurs (Butterworth *et al.*, 2004; Dougherty *et al.*, 1995).

b. Dans la plupart des projets antérieurs concernant le traitement de la MPOC, les attributs sont déjà discrétisées par un expert (Van der Heijden *et al.*, 2013), (Ryynänen *et al.*, 2013) (Himes *et al.*, 2009). Ainsi, la discrétisation automatique employée dans le contexte de la MPOC est une nouvelle contribution.
c. La discrétisation diminue le nombre d'états des attributs continus, ce qui a pour effet de faciliter l'interaction avec notre outil de prédiction final (Chapitre IV).

Dans cette expérimentation, les méthodes de discrétisation utilisées sont : *EqualWidthDiscretization (EWD), EqualFrequencyDiscretization (EFD), Fayyad & Irani's MDL* et *Kononenko's MDL*. Les résultats de ces méthodes sur les 61 attributs sont donnés au tableau 3.4.

Tableau 3.4: Influence de la discrétisation sur les classificateurs

Discrétisation	**Test C**	Les attributs initiaux sont mixtes (continus et discrets)			
		61 attributs et 1985 patients en utilisant *Weka*			
	10 - *Cross Validation*	Bayes		Arbre de décision	
	AUROC	Naïf bayésien	Réseau bayésien (K2)	ID3	C4.5
Non supervisée	EWD 10 Intervalles	0.767	0.767	0.541	0.621
	EFD 10 Intervalles	0.769	0.769	0.574	0.510
Supervisée	*Fayyad & Irani's MDL*	0.781	0.781	0.584	0.608
	Kononenko's MDL	**0.782**	**0.782**	0.579	0.613

Théoriquement, le résultat du naïf bayésien et du réseau bayésien doit être identique, en raison de la structure de dépendance du réseau bayésien en utilisant K2 avec un nombre de parents égal à 1 (configuration par défaut de Weka, Figure 3.8), qui correspond à la structure du naïf bayésien. En outre, dans ce test C, tous les attributs sont discrets, alors l'algorithme naïf bayésien utilise le théorème de Bayes dans l'inférence comme le réseau bayésien. D'autre part, l'algorithme ID3 donne un résultat parce que tous les attributs sont discrétisés.

Par conséquent, comme l'évaluation l'a montré, le réseau bayésien et le naïf bayésien avec la discrétisation de *Kononenko's MDL* retournent le score le plus élevé. Ce résultat (AUROC =78.2 %) est meilleur que celui du test A (AUROC = 76.8 %), où nous utilisons la base d'apprentissage originale. Cependant, la méthode de sélection des attributs pertinents *Wrapper-BestFirst* avec le naïf bayésien retourne un score encore plus grand, soit AUROC = 79.4 % (Test C).

Les algorithmes d'arbre de décision se sont améliorés avec les méthodes de sélection et de la discrétisation (tests B et C), mais, ils ne retournent pas un bon résultat par rapport au réseau bayésien et au naïf bayésien. Les résultats de la discrétisation sont rassemblés à l'appendice D.

3.6.7 Comparaison des méthodes de discrétisation et de sélection des attributs pertinents appliquées ensemble

Le test de sélection des attributs pertinents (test C) retourne le score le plus élevé avec un AUROC= 79.4 %, qui surpasse le score de la discrétisation (test D) retournant un AUROC =78.2 %. Ces deux scores sont meilleurs que le score du test A (AUROC =

76.8 %). Partant de cet effet, il est nécessaire de mettre en place une combinaison de ces deux méthodes (sélection et discrétisation) afin de tester leur incidence combinée. Cependant, pour garantir une bonne précision de prédiction, il est important de savoir laquelle doit être appliquée en premier lieu: sélection-discrétisation ou discrétisation-sélection. Cet ordre est important, parce que :

3. Selon la recherche existante, aucun chercheur n'affirme que la discrétisation doive être effectuée avant la sélection d'attributs, ou le contraire.
4. La méthode de la sélection d'attributs pertinents *CFSsubsetEval* peut supporter les attributs continus. Il n'est donc pas nécessaire de faire la discrétisation avant la sélection.

Nous avons constaté qu'il est important de choisir un ordonnancement de ces deux méthodes, afin d'obtenir une bonne performance de prédiction. Au tableau suivant, nous appliquons la sélection d'attributs avant la discrétisation, comme suit :

Tableau 3.5 : Résultats obtenus par la métrique AUROC en appliquant la sélection d'attributs pertinents qui précède la discrétisation

Sélection d'attributs → Discrétisation		Test D	Les attributs initiaux sont mixtes (continus et discrets)			
			61 attributs et 1985 patients en utilisant Weka			
		10 - *Cross Validation*	Bayes		Arbre de décision	
		AUROC	Naïf bayésien	Réseau bayésien (K2)	ID3	C4.5
Filters	**CFSsubsetEval**	EWD 10 Intervalles	0.780	0.780	0.588	0.496
		EFD 10 Intervalles	0.785	0.785	0.579	0.496
		Fayyad & Irani's MDL	0.789	0.789	0.515	0.666
		Kononenko's MDL	0.788	0.788	0.509	0.670
	Gain Ratio Attribute Eval	EWD 10 Intervalles	0.761	0.761	0.529	0.496
		EFD 10 Intervalles	0.761	0.761	0.566	0.496
		Fayyad & Irani's MDL	0.757	0.757	0.639	0.666
		Kononenko's MDL	0.757	0.757	0.639	0.666
Wrappers	**BestFirst**	EWD 10 Intervalles	0.793	0.781	-	0.496
		EFD 10 Intervalles	0.793	0.786	-	0.496
		Fayyad & Irani's MDL	0.798	0.795	-	0.610
		Kononenko's MDL	0.797	0.795	-	0.618

	Génétique	EWD 10 Intervalles	0.786	0.787	-	0.496
		EFD 10 Intervalles	0.785	0.786	-	0.496
		Fayyad & Irani's MDL	0.786	0.796	-	0.539
		Kononenko's MDL	0.786	0.785	-	0.539

Parmi les résultats, dont nous rendons compte au tableau 3.5, le meilleur est obtenu par la combinaison entre *Wrapper-BestFirst* pour la sélection, suivi de *Fayyad & Irani's MDL* pour la discrétisation, en utilisant le classificateur naïf bayésien pour l'apprentissage, avec comme résultat un AUROC = 79.8 %. Ce résultat est meilleur que celui que nous obtenons à faire seulement la sélection des attributs pertinents (Test B, AUROC= 79.4 %), ou seulement la discrétisation (Test C, AUROC =78.2 %).

Le tiret (-) avec l'algorithme ID3 indique que ce dernier ne traite pas les attributs continus pour être l'évaluateur des méthodes *Wrappers,* si nous appliquons ces dernières sur une base d'apprentissage dont les attributs sont continus.

D'autre part, le tableau 3.6 teste l'autre sens, c'est-à-dire que nous appliquons la discrétisation avant la sélection, pour analyser son influence sur les classificateurs.

Tableau 3.6 : Résultats obtenus par la métrique AUROC en appliquant la discrétisation qui précède la sélection d'attributs pertinents

Discrétisation → Sélection	Test E	Les attributs initiaux sont mixtes (continus et discrets)			
		61 attributs et 1985 patients en utilisant Weka			
	10 - *Cross Validation*	Bayes		Arbre de décision	
	AUROC	Naïf bayésien	Réseau bayésien (K2)	ID3	C4.5
EWD 10 Intervalles	CFSsubsetEval	0.780	0.780	0.550	0.496
	Gain Ratio Attribute Eval	0.730	0.729	0.522	0.496
	WrapperBestFirst	0.796	0.796	0.734	0.675
	WrapperGénétique	0.791	0.791	0.614	0.658
EFD 10 Intervalles	CFSsubsetEval	0.771	0.771	0.555	0.496
	Gain Ratio Attribute Eval	0.719	0.719	0.552	0.496
	WrapperBestFirst	0.799	0.801	0.757	0.562
	WrapperGénétique	0.792	0.791	0.623	0.496
Fayyad & Irani's MDL	CFSsubsetEval	0.795	0.795	0.547	0.664
	Gain Ratio Attribute Eval	0.768	0.768	0.605	0.673
	WrapperBestFirst	**0.802**	**0.802**	0.759	0.688
	WrapperGénétique	0.800	0.802	0.758	0.660

Kononenko's MDL	CFSsubsetEval	0.791	0.791	0.580	0.663
	Gain Ratio Attribute Eval	0.769	0.769	0.609	0.658
	WrapperBestFirst	0.800	0.800	0.763	0.692
	WrapperGénétique	0.798	0.801	0.756	0.670

Dans ce dernier test (Test E, Tableau 3.6), la discrétisation prend place avant la sélection d'attributs pertinents, en les appliquant au réseau bayésien ou au naïf bayésien. Il retourne un résultat important avec un AUROC = 80.2 %, un meilleur score que celui obtenu dans l'autre sens (Test E, AUROC = 79.8 %).

En plus, selon cet ordre (discrétisation →sélection), nous observons aussi que même si la capacité prédictive de l'arbre de décision (ID3 et C4.5) ne retourne pas le meilleur résultat, ce dernier s'améliore avec ces deux étapes.

Par conséquent, nous concluons que pour obtenir une bonne performance pour le réseau bayésien ou le naïf bayésien, nous devons appliquer sur la base d'apprentissage l'ordre d'algorithmes suivant :

1. Discrétisation avec *Fayyad & Irani's MDL.*
2. Sélection des attributs pertinents avec la méthode *Wrapper* en utilisant l'algorithme de recherche *BestFirst.*

Finalement, parmi toutes les expérimentations précédentes (tests A à E), nous avons observé que le réseau bayésien et le naïf bayésien, qui offrent les meilleurs résultats, sont en compétition pour être les prédicteurs les plus performants. En raison de cette similarité, nous avons comparé le naïf bayésien avec les diverses structures du réseau bayésien, pour choisir celle qui retourne la meilleure performance.

3.6.8 Comparaison des algorithmes de dépendance

La création d'une structure de réseau bayésien par des méthodes automatiques à partir des données comporte plusieurs avantages:

a. Elle facilite et accélère la création du réseau de dépendance.
b. Elle utilise une méthode automatique au lieu de compter sur des experts (Van der Heijden *et al.*, 2013).
c. Elle augmente la performance de la prédiction.

À cet effet, les deux algorithmes les plus utilisés sont TAN & K2 (chapitre I). Selon Weka, il n'y a pas de restriction du nombre de parents (P) dans K2. Pour cette raison, dans notre recherche nous nous sommes concentrés sur les trois premiers parents. Par contre, TAN a un nombre de parents fixe, qui est égal à 2 (Witten et Frank, 2011).

Rappelons l'utilisation des méthodes automatiques pour la discrétisation et pour construire le réseau de dépendance, produit un système de prédiction autonome. Ce système n'a pas besoin de l'intervention d'experts (ex: pneumologues) pour évoluer dans le futur. Cette combinaison représente une contribution de notre recherche.

Le tableau 3.7 (test F) démontre que la méthode TAN combinée au réseau bayésien, en appliquant la méthode de discrétisation *Fayyad & Irani's MDL,* suivie de la sélection des attributs pertinents (*Wrapper-BestFirst*) sur la base d'apprentissage, peut créer un système de prédiction performant et autonome pour détecter l'exacerbation dans la MPC, avec un AUROC = 81.5 % (Figure 3.10).

Tableau 3.7 : Comparaison entre le réseau bayésien et le naïf bayésien en apprenant le réseau de croyance du réseau bayésien à partir de la base d'apprentissage

Test F	**A-** Les attributs sont discrets, avec Fayyad & Irani's MDL **B-** La sélection utilise *Wrapper* avec l'algorithme de recherche *BestFirst*.	
10 - *Cross Validation*	Area Under Roc Curve - AUROC	*Nb des attributs pertinents*
Naïf bayésien	80.2 %	12
BN(K2) – 1 P	80.2 %	12
BN(K2) – 2 P	80.9 %	15
BN(K2) – 3 P	80.2 %	14
BN(TAN)	81.50 %	17

1. Discrétisation supervisée, avec *Fayyad & Irani's MDL*.
2. Sélection les attributs les plus pertinents (*Wrappers – BestFirst*).
3. TAN pour créer la structure de dépendance des données.
4. Cette séquence des algorithmes sera appliquée sur la méthode d'apprentissage, réseau bayésien.

Figure 3.9 : Les algorithmes obtenus pour fournir un modèle de prédiction performant, autonome et raffiné[13] (MPAR)

Les quatre algorithmes ci-dessus (Figure 3.9) retournent un bon résultat si on les compare avec plusieurs autres (tests A à F). La précision, qui était AUROC = 76.8 % au début, augmente jusqu'à AUROC = 81.5 % (Figure 3.10) si on applique les algorithmes de 1 à 4 (Figure 3.9). Ainsi, en appliquant ces quatre algorithmes le

[13] Raffiné signifie la sélection d'attributs pertinents (voir section 2.5).

modèle de prédiction va être autonome (pas besoin des experts médicaux), et raffiné en faisant la sélection d'attributs pertinents et la discrétisation (voir section 2.5).

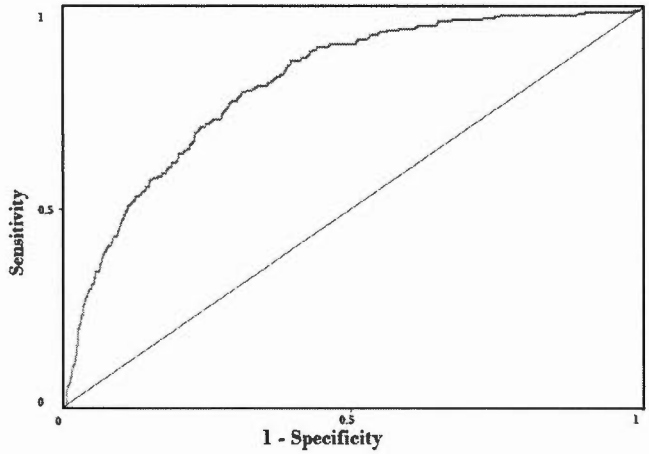


Figure 3.10 : *Receiver Operating Characteristic (ROC)*, courbe correspondant au modèle final de la prédiction de l'exacerbation MPOC. AUROC = 81.5 %.

Alors, ces algorithmes dont rend compte la figure 3.9 démontrent l'efficacité et l'importance des étapes B, C, et D du modèle proposé au début de ce chapitre (Section 3.2), qui offre une bonne précision (AUROC = 81.5 %). L'évaluation de ces quatre algorithmes (étape E) est dans la figure 3.10. L'utilité de la dernière étape (F) de notre modèle (Arrangement du contexte) est démontrée au chapitre IV. De plus, à la fin de ce dernier chapitre, nous démontrons clairement l'efficacité de ce modèle (essentiellement les algorithmes sélectionnés dans la figure 3.9) en l'évaluant avec huit bases d'apprentissage de différents domaines. Ces huit bases d'apprentissages sont utilisées aussi pour valider notre proposition *WrappersPlus* pour sélectionner les attributs pertinents (Section 4.6).

3.7 Conclusion

Notre travail offre la solution aux défis identifiés pour les systèmes de prédiction de contexte pertinent dans le cas de la MPOC, que voici:

1. Le système de surveillance ne peut pas donner une réponse immédiate aux patients (Halpin *et al.*, 2011; Maiolo *et al.*, 2003) et il coûte cher puisque basé sur le traitement manuel par le personnel médical (Vontetsianos *et al.*, 2005).
   Notre modèle de prédiction donne une réponse automatique et personnelle, en temps réels, en se basant sur les symptômes observés.
2. Le modèle de prédiction n'est pas autonome ni capable d'évoluer avec le temps, car il est basé sur l'analyse de spécialistes médicaux pendant sa construction (Van der Heijden *et al.*, 2013).
   Le modèle de prédiction que nous avons proposé est totalement automatique et utilise l'algorithme TAN pour construire le réseau de croyance, ainsi qu'une nouvelle méthode dans le contexte de la MPOC, Wrapper-BestFirst, pour sélectionner les attributs pertinents à la place du spécialiste.
3. La discrétisation automatique n'est pas supportée dans plusieurs travaux antérieurs (Himes *et al.*, 2009), (Ryynänen *et al.*, 2013), (Sandelowsky *et al.*, 2011), etc.
   Dans notre modèle de prédiction, l'algorithme de discrétisation supervisée est appliqué (Fayyad & Irani's MDL).
4. La performance n'est pas concluante dans la plupart des travaux antérieurs concernant la MPOC (par exemple: (Yañez *et al.*, 2012) AUROC= 76 %, (Ryynänen *et al.*, 2013) AUROC = 69 %, (Raghavan *et al.*, 2012) AUROC = 77 %, (Amalakuhan *et al.*, 2012) AUROC = 75 %.)
   Le modèle proposé offre une bonne performance, avec un AUROC = 81.5 %.
5. Ainsi, nous avons démontré que la discrétisation précédant la sélection d'attributs pertinents peut améliorer la performante de la prédiction des attributs pertinents.

Rappelons que les détails algorithmiques de l'ensemble des algorithmes utilisés sont donnés au chapitre I, ainsi que les motivations théoriques qui nous les ont fait choisir. La revue de littérature qui déclare la nouveauté de ces algorithmes dans le contexte de MPOC, ainsi que l'utilité de leur utilisation par rapport aux travaux antérieurs sur le sujet se trouve au chapitre II (Section 2.5).

En résumé, le modèle proposé a démontré son utilité dans le cas de la détection de l'exacerbation dans la MPOC, de façon performante, autonome et raffiné (sélection d'attributs pertinents). La dernière étape (F - Arrangement du contexte) de notre modèle est démontrée au chapitre IV, afin d'observer les attributs les plus pertinents en cas d'urgence. En outre, au même chapitre, nous réalisons ce modèle dans une application contextuelle qui bénéficie de son efficacité.

En plus, au chapitre IV, nous proposons une extension de l'algorithme de sélection des attributs pertinents *Wrapper*. Cette extension, est validée par huit bases d'apprentissage, pour minimiser le nombre d'attributs pertinents et garder la même capacité prédictive que l'algorithme *Wrapper*. En même temps, nous validons les quatre algorithmes obtenus par l'ensemble de comparaison (Test A-F) pour fournir un modèle MPAR (Figure 3.9).

CHAPITRE IV

DEVELOPEMENT D'UNE APPLICATION CONTEXTUELLE : INTÉGRATION DE DIFFÉRENTES ALGORITHMES DANS NOTRE OUTIL DE PRÉDICTION, ÉTUDE DE CAS ET PRÉSENTATION DES RÉSULTATS OBTENUS

4.1 Introduction

Grâce à une application contextuelle automatisée qui gère la maladie pulmonaire d'obstructive chronique (MPOC), il est possible de prédire à l'avance les exacerbations des patients, dans le but de i) aider le personnel médical à prendre les mesures nécessaires, ii) éviter l'hospitalisation et iii), réduire le risque de voir s'aggraver la maladie (Chapitre II, Section 2.4). Dans ce chapitre nous allons concevoir et valider une application contextuelle performante et autonome permettant d'aider les patients atteints de MPOC et le personnel médical. Pour réaliser cette application, nous nous sommes basés, au plan théorique, sur les comparaisons que nous avons présentés au chapitre III afin de sélectionner les algorithmes de notre Modèle de prédiction performant et autonome (MPAR) (Chapitre III, Figure 3.9), et au plan pratique sur la bibliothèque Netica-J, l'EDI NetBeans pour Java et Weka.

Pendant la réalisation de cette application, nous avons observé que le nombre d'attributs pertinents est toujours aussi grand après l'étape de leur sélection (17 attributs). Pour cette raison, nous proposons d'arranger les attributs pertinents de façon descendante, de sorte que l'attribut le plus important apparaisse au début et les autres à sa suite, jusqu'à l'attribut le moins important. Cet ordre est très utile pour aider le personnel médical à rendre un diagnostic suffisamment précis: en général, le médecin

n'en observe que quelques-uns, et pas les dix-sept symptômes (attributs). Nous appliquons cette nouvelle idée au contexte de la maladie MPOC.

De plus, en raison de l'expertise acquise avec les algorithmes au fil de ce processus de recherche, nous proposons à la fin de ce chapitre une extension de l'algorithme de sélection d'attributs pertinents *Wrappers* que nous avons appelée *WrappersPlus*. Cette extension est validée par huit bases d'apprentissage différentes, dans le but de minimiser le nombre d'attributs pertinents et de garder la même capacité prédictive que si on applique les algorithmes de MPAR en se basant sur *Wrappers-BestFirst* (Figure 3.9).

4.2 Étude de cas du modèle de prédiction proposé

La figure 4.1 montre une étude de cas des six étapes implémentées pour fournir une application contextuelle performante, autonome, raffinée et efficace. Ces étapes sont basées sur le modèle proposé au chapitre III (Figure 3.1), appliqué au cas de l'exacerbation dans la MPOC. Ces étapes sont les suivantes:

(A) Obtention d'un échantillon de la base d'apprentissage de CrowdANALYTIX. La représentation du contexte en ontologie est faite au chapitre III (Figure 3.3).

(B) Application de la méthode de discrétisation supervisée (*Fayyad & Irani's MDL*).

(C) Sélection des attributs pertinents avec *Wrapper-BestFirst.*

(D) Création du modèle de prédiction en utilisant TAN, qui permet de construire le réseau bayésien.

(E) Évaluation de la performance du modèle prédictif à l'aide de la métrique *Area Under Receiver Operating Characteristic Curves (AUROC)*, avec 10-*Fold Cross Validation* stratifiée pour déterminer le meilleur point *Cutoff*.

(F) Représentation de l'arrangement des attributs pertinents en utilisant *GainRatio*.

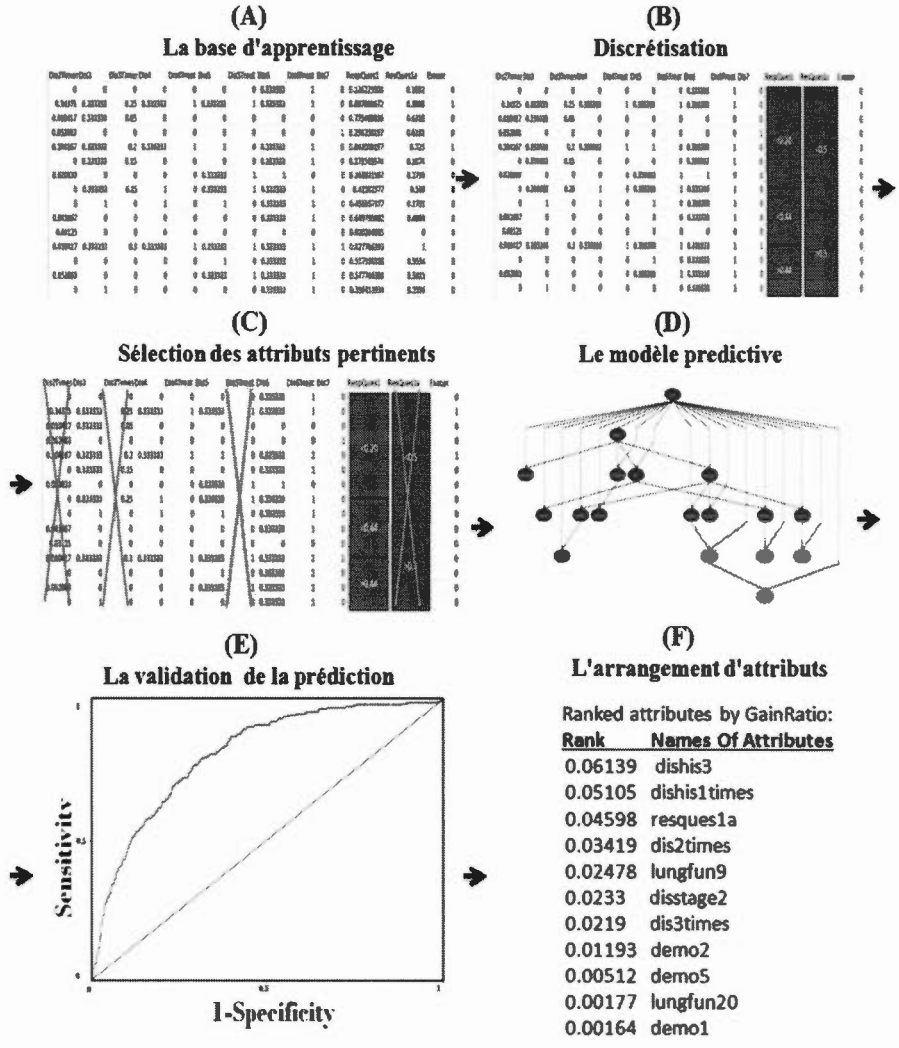


Figure 4.1 : Étude de cas : les six étapes du modèle proposé, pour fournir une application contextuelle performante, autonome, raffiné et efficace (voir section 2.5).

4.3 Technologies utilisées : Netica-Java, NetBeans et Weka

Les technologies utilisées pour le développement de notre application contextuelle sont Netica-Java (Netika-J), NetBeans, et Weka.

Netica-J offre l'API complète de Netica™ en Java. Netica™ est le logiciel le plus largement utilisé pour le développement du réseau bayésien (Netica, 2000). Les principaux clients de Netica™ sont: *American Board of Family Medicine*, *Department of National Defence-Canada*, etc (Manual, 2012). L'utilisation de l'API de Netica (Netica-J) nécessite une licence[14]. Netica-J[15] contient des classes et des méthodes en Java pour construire, apprendre, modifier, transformer, enregistrer et lire les réseaux bayésiens, ainsi qu'un moteur d'inférence puissant.

D'autre part, l'environnement de développement *open source* NetBeans se concentre sur la simplification du développement de l'application contextuelle Java, en fournissant des *buttons*, *frames*, etc. En plus, NetBeans permet d'ajouter une librairie externe permettant d'utiliser ses propres classes et méthodes pendant la programmation. Dans notre projet, nous ajoutons à notre projet dans NetBeans, la librairie Netica-J pour construire et gérer le réseau bayésien (Figure 4.2).

Finalement, Weka est utilisé pour accomplir la discrétisation, et la sélection d'attributs pertinents, ainsi que pour construire la dépendance entre ces attributs (Figure 4.4, 4.5 et 4.6).

---

[14] Grâce à une communication par message électronique avec Netica, nous avons obtenu une licence spéciale et gratuite pour 4 mois : «+SalehL/UQAM/Ex17-02-15,.....» ***Thanks Netica.***

[15] Cette librairie peut être obtenue à partir de la page suivante : www.norsys.com/netica-j.html.

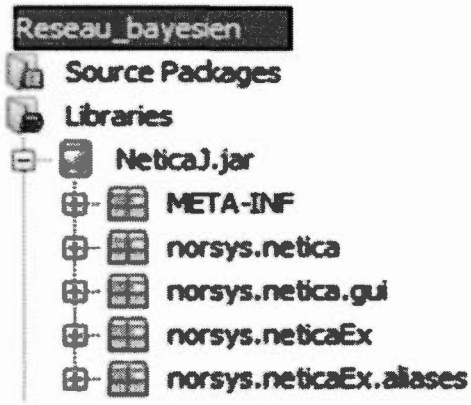


Figure 4.2 : NeticaJ.jar est intégré avec NetBeans dans notre projet

4.4 Conception et implémentation de l'application contextuelle

En utilisant NetBeans, la licence de Netica-J, Weka, la base d'apprentissage fournie par CrowdANALYTIX et les algorithmes appliqués dans notre modèle (Figure 4.1), nous développons notre application contextuelle en Java, qui peut être utilisée par les patients ou le personnel médical pour détecter les risques d'exacerbation chez les patients souffrant de MPOC.

Notre application contextuelle intègre les algorithmes de prédiction suivants : *Wrapper-BestFirst*, *Fayyad & Irani's MDL*, *TAN*, réseau bayésien, et *GainRatio*. Elle présente les caractéristiques suivantes :

- Performante : elle présente un AUROC= 81.5 % en utilisant le réseau bayésien.
- Autonome: elle est capable d'évoluer sans l'intervention d'experts en utilisant *Fayyad & Irani's MDL* pour la discrétisation et TAN pour le réseau de croyance.

- Raffinée: elle permet une interaction rapide avec l'utilisateur, grâce à *Wrapper-BestFirst et Fayyad & Irani's MDL*, qui réduisent parfaitement le nombre d'attributs et d'états à traiter.
- Efficace : nous arrangeons les attributs pertinents pour garantir une précision minimale en cas d'urgence (Section 4.5), en nous basant sur l'heuristique *GainRatio*.

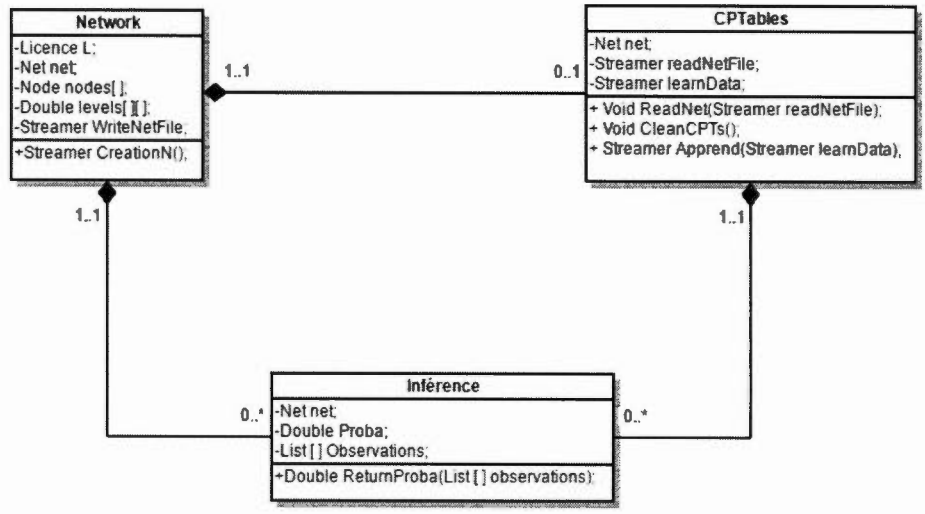


Figure 4.3 : Modèle de classe en utilisant UML, pour implémenter le réseau bayésien

Cette application est décrite par le diagramme de classe de la figure 4.3. Elle est représentée par trois classes principales. Ces classes ainsi que leurs relations sont expliquées comme suit :

Les relations entre les classes:

Notre application se compose de trois classes, entre ces dernières la relation de composition est bien apparue (les classes *CPTables* et *Inférence* sont deux composants du réseau (classe *Network*), ainsi *Inférence* est un composant du *CPTables*. La raison de cette composition est que les *Conditional Probability Tables (CPTs)* ne peuvent exister sans le réseau de croyance (*Network*). Ainsi, on ne peut pas appliquer

l'inférence bayésienne sans réseau ou CPTs. La cardinalité entre les classes s'explique comme suit : le réseau de croyance qui est créé par la classe *Network* peut avoir zéro ou un objet *CPTables* pour apprendre les CPTs de la base d'apprentissage, parce que le réseau ne peut pas remplir les CPTs, ainsi qu'il peut les remplir une fois par la base d'apprentissage. D'autre part, dans un cas d'inférence (classe *Inférence*) nous devons avoir un objet pour construire le *Network* et un autre pour apprendre ce réseau (remplir les CPTs par la classe *CPTables*). Dans l'autre sens, le *Network* peut faire plusieurs fois l'inférence à chaque observation. Ainsi que, l'inférence peut être appliquée plusieurs fois par les CPTs à chaque observation.

Les détails de ces classes:

1. **Classe *Network*** : Dans cette classe, on implémente le résultat que nous avons obtenu au chapitre III (Figure 3.9), en nous basant sur Weka.
   Premièrement, nous implémentons la discrétisation de *Fayyad & Irani's MDL* (Figure 4.4) pour les attributs pertinents obtenus par *Wrapper-BestFirst* (17 attributs, Figure 4.5.). Deuxièmement, nous créons le réseau de croyance qui est fourni par la méthode TAN (Figure 4.6). Ces deux étapes sont obtenues lorsqu'on applique la méthode *CreationN*() (Figure 4.3). Le code d'implémentation de la classe *Network* se trouve à l'appendice C.1.

Figure 4.4 : Discrétisation *Fayyad & Irani's MDL* sur les 17 attributs

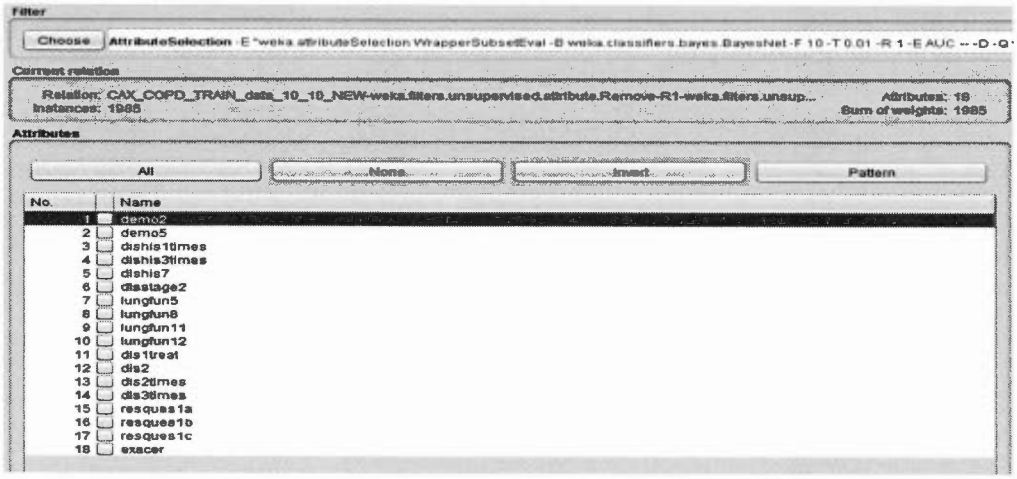


Figure 4.5 : Sélection des attributs pertinents par Weka (*Wrapper-BestFirst*).

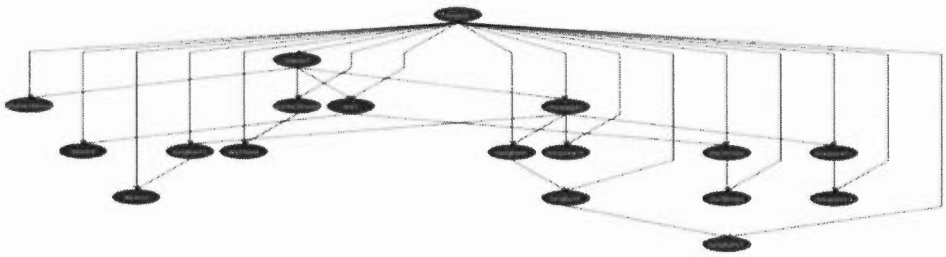


Figure 4.6 : Structure du réseau bayésien, en utilisant la méthode TAN pour détecter l'exacerbation de la MPOC

2. **Classe *CPTables*** : Pour faire l'inférence dans le réseau bayésien, il faut remplir les CPTs de chaque attribut. Le CPT d'un attribut contient les probabilités de toutes les combinaisons possibles entre les états de l'attribut et ses parents.
   D'abord, la classe *CPTables* lit le réseau créé par la classe *Network*. Ensuite, il nettoie le CPT de chaque nœud dans le réseau. À la fin, nous appliquons la méthode *reviseCPTsByCaseFile,* qui est disponible dans Netica-J, pour apprendre les tableaux de probabilités conditionnelles (CPTs) à partir de la base d'apprentissage. Ces tableaux représentent les probabilités d'exacerbation et

leurs symptômes dans la MPOC. Le code d'implémentation de cette classe est donné à l'appendice C.2.

3. **Classe *d'inférence*** : Cette classe permet l'inférence dans le réseau bayésien. En se basant sur les *CPTs,* le réseau créé, et la formule de Bayes. Dans Netica-J, le moteur d'inférence est implémenté par la méthode *getBelief()*, qui retourne une probabilité à chaque observation. Le code de cette classe se trouve à l'appendice C.3.

   En plus, pour bien classifier les patients les plus susceptibles d'avoir une exacerbation et ceux qui ne courent pas de risques, nous devons préciser la meilleure *Cutoff*[16], soit le seuil d'un modèle probabiliste, pour remplir la matrice de confusion. Le meilleur *Cutoff* est un point sur la courbe ROC; ce point doit être à une distance minimale à la coordonnée = (0, 1). On peut trouver ce point (meilleur *Cutoff*), en calculant (1-sensitivité)^2 + (spécificité)^2 sur tous les points de la courbe, pour choisir la valeur minimale. La figure 4.7 en montre un exemple.

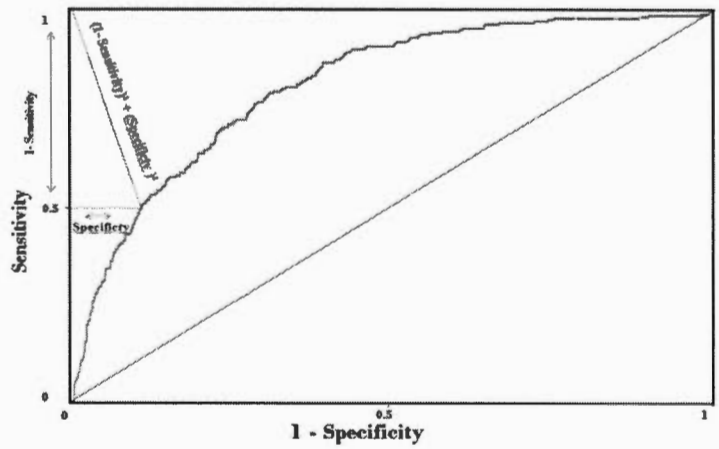


Figure 4.7 : Méthode pour trouver la meilleure *CutOff.*

[16] Une explication autour *Cutoff* est ici : http://www.norsys.com/tutorials/netica/secD/tut_D2.htm

Selon Weka et la courbe ROC de la figure 4.7, nous avons précisé que ce point (meilleur *cutoff*) égal à 0,1 (c'est-à-dire le *Cutoff* = 0.1 a la valeur minimale de [1-sensitivité] ^{2} + [spécificité] ^{2}). Lorsque la probabilité d'exacerbation est > 0.1, on peut déduire que le patient est successible d'avoir une exacerbation. Dans le cas contraire, on affiche que son état est bon.

4.5 Application de l'heuristique Gain-Ratio pour arranger les attributs pertinents

Après l'achèvement des classes nécessaires pour faire l'inférence dans le réseau bayésien, il faut relier les attributs pertinents à une interface graphique, afin que l'application réagisse avec l'utilisateur. Cependant, pendant la construction de l'interface graphique avec NetBeans, nous avons trouvé que le nombre d'attributs pertinents est aussi grand (17 attributs) pour que le patient ou le médecin les observe en cas d'urgence. Or, nous ne pouvons pas supprimer de ces attributs, parce que la suppression diminuera la performance de système à moins que 81.5 %.

Par conséquent, comme solution alternative, nous proposons d'arranger les attributs de façon descendante en utilisant l'heuristique *GainRatio*. Habituellement, le *GainRatio* est utilisé pour la construction de l'arbre de décision C4.5. Avec cette proposition, le patient ou le médecin commence à observer les attributs qui ont une capacité discriminante élevée, et chaque fois qu'il observe un nouvel attribut selon l'ordre, la précision de la prédiction s'améliore. De cette façon, en cas d'urgence, le patient et le médecin observent les attributs les plus pertinents, ce qui garantit une précision minimale. Considérons par exemple seulement 8 attributs arrangés. Dans ce cas, on ne peut obtenir que 78 % de précision. D'autre part, si les attributs ne sont pas arrangés, c'est-à-dire qu'on observe arbitrairement n'importe quel attribut, alors il est difficile de connaître la capacité prédictive de ces attributs. C'est seulement si on observe tous les 17 attributs qu'on pourra atteindre 81.5 % pour la précision de la prédiction.

Pour réaliser cette proposition, nous utilisons l'attribut qui présente la capacité discriminante la plus élevée au début de la fenêtre d'interaction, puis de façon descendant, selon les valeurs de *GainRatio* de chaque attribut, nous ajoutons les autres attributs. L'efficacité de cette méthode est démontrée au tableau 4.1, où nous commençons le test avec les huit premiers attributs les plus pertinents qui sont apparus à la figure 4.8. Ces huit principaux attributs garantissent une précision minimale de 78 % (Tableau 4.1). Cette proposition de réduction des attributs pertinents de 17 à 8 en les arrangeant est une nouvelle contribution en général, et nous l'appliquons au contexte de la MPOC.

```
Ranked attributes:
0.05913   1 dishis3times
0.05105   2 dishis1times
0.04878   3 dishis7
0.04598   4 resques1a
0.03419   5 dis2times
0.03293   6 resques1c
0.03054   7 lungfun8
0.03042   8 resques1b
0.0233    9 disstage2
0.0219   10 dis3times
0.02172  11 lungfun12
0.01535  12 dis2
0.01277  13 dis1treat
0.01193  14 demo2
0.01179  15 lungfun5
0.01078  16 lungfun11
0.00512  17 demo5
```

Figure 4.8 : Arrangement des attributs pertinents, basé sur la mesure *GainRatio*

	Nombre d'attributs									
	8	9	10	11	12	13	14	15	16	17
AUROC	78 %	78.4 %	79.6 %	79.9 %	80.4 %	80.5 %	81 %	80.9 %	81.1 %	81.5 %

Tableau 4.1 : Variation de la précision de prédiction, en fonction du nombre d'attributs utilisé, en se basant sur l'ordre de la figure 4.8

Dans le tableau 4.1, nous ajoutons un nouvel attribut, basé sur l'ordre de la figure 4.8, et on commence par 8 attributs qui représentent l'interface de base de notre application

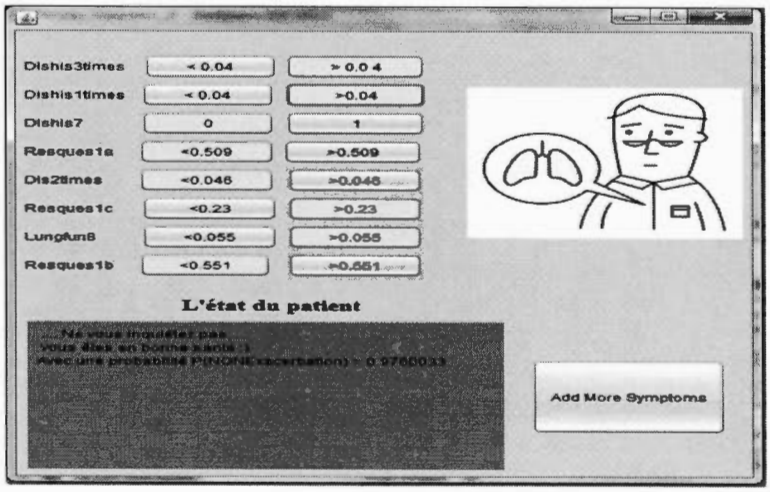


Figure 4.9 : Interface de base de notre application présentant les huit symptômes primaires

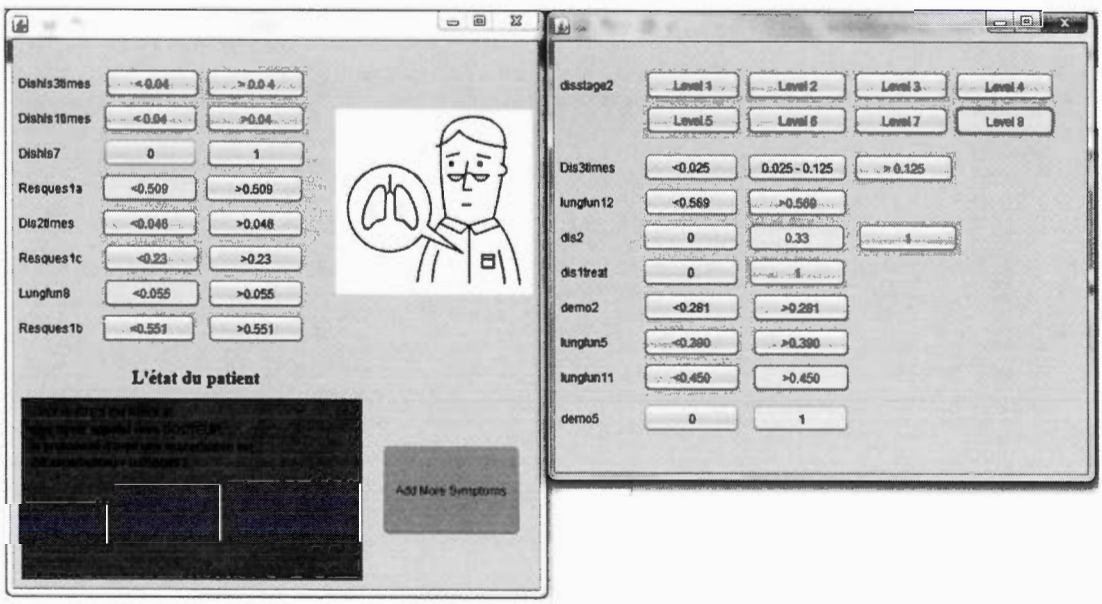


Figure 4.10 : Interface de notre application, avec les symptômes secondaires (partie droite de la figure)

Les figures 4.9 et 4.10 représentent l'interface graphique de notre application. Cette application changera la couleur du bouton à jaune sur clique. Les boutons devant chaque étiquette (*label*) sont les états de chaque attribut. Ainsi, les étiquettes représentent les noms d'attributs dans la base d'apprentissage. Soulignons que les noms d'attributs utilisés dans notre expérimentation sont invisibles et anonymes (Section 3.3); cette application est un prototype rapide pour valider l'intégration des différentes parties de ce travail.

La première interface de cette application contient les huit attributs principaux (Figure 4.9) à considérer pour garantir une précision minimale égale à 78 %. Après, si le patient veut mesurer plus de symptômes, il peut cliquer sur (*Add more symptoms*), puis à chaque nouvelle observation selon l'ordre, la précision de la détection d'exacerbation s'améliore jusqu'à 81.5 %.

4.6 Validation du modèle proposé et extension de la méthode Wrappers : WrappersPlus

Dans ce mémoire, la maladie MPOC est utilisée comme domaine d'application pour divers algorithmes et méthodes: la discrétisation (*Fayyad & Irani's MDL*, etc.), la sélection d'attributs pertinents (*Wrappers*, *Filters*) et le raisonnement (naïf bayésien, réseau bayésien, arbre de décision) dans le domaine de l'apprentissage automatique.

En raison de l'existence de plusieurs méthodes dans ce domaine, la plupart des recherches concernant la sélection d'attributs pertinents se concentrent sur la comparaison entre ces méthodes. En effet, (Porkodi, 2014) a comparé cinq méthodes de type *filters* pour sélectionner les attributs pertinents du cancer, et il a trouvé que la méthode *ReliefF (RF)* est celle qui retourne le meilleur sous ensemble de ces attributs. (Bangsuk et Cheng-Fa, 2014) ont comparé quatre méthodes *Wrappers* et *filters*. Le

résultat a démontré que la méthode *Wrappers* est meilleure que la *filters* pour améliorer la précision de la prédiction. (Yildirim, 2015) ont comparé plusieurs méthodes de *filters* en utilisant *hepatitis dataset* et ont conclu qu'aucune méthode unique de *filters* n'est la meilleure. Même chose pour (Panthong et Srivihok, 2015), qui a travaillé sur les méthodes *Wrappers* en utilisant treize bases d'apprentissage. Le résultat a montré que le *Sequential Forward Selection* et l'arbre de décision obtiennent les meilleurs résultats. (Gutlein *et al.*, 2009) a proposé un nouvel algorithme pour améliorer la vitesse de la méthode *Wrappers* et (Karegowda *et al.*, 2010) a proposé l'algorithme génétique comme une méthode de recherche en utilisant *Wrappers*.

Problème : Au cours de notre expérimentation avec les méthodes *Wrappers* et *Filters* (Chapitre III), nous avons remarqué que la méthode *Wrappers-BestFirst,* avec le classificateur de réseau bayésien et la méthode TAN, retourne le meilleur sous-ensemble d'attributs dans le cas de la MPOC. Cependant, l'utilisation de la méthode *BestFirst,* une heuristique qui chercher un sous-ensemble d'attributs dans la méthode *Wrappers*, implique la situation suivante.

Pendant l'exécution du processus *Wrappers*, la méthode *BestFirst* commence par un ensemble d'attributs vide et génère toutes les expansions possibles d'un sous-ensemble ayant un seul attribut. Le sous-ensemble d'attributs qui a la valeur d'évaluation la plus élevée est choisi, et il s'étend de la même manière. Si l'extension d'un sous-ensemble n'entraîne aucune amélioration, la recherche retombe sur le sous-ensemble non expansé suivant et continue à partir de là (Karagiannopoulos *et al.*, 2007). Dans notre cas, la méthode d'évaluation dans *Wrappers* est basée sur le réseau bayésien et la métrique AUROC.

Alors, selon le processus que nous venons de décrire, *BestFirst* ne retourne pas en arrière sauf s'il ne s'améliore pas. Dans ce contexte, le cas suivant n'est pas pris en compte. Exemple : si vous avez un espace de recherche de trois attributs {a, b, c}, et

que selon la méthode d'évaluation (en unité « U ») utilisée, on a : a= 5 U, b = 4 U et c= 3U.

Pour sélectionner les attributs pertinents parmi les trois attributs, en nous basant sur *Wrapper-BestFirst*, dans la première extension, le sous-ensemble pertinent va être {a}, car il a la valeur d'évaluation la plus élevée. Ensuite, dans la deuxième expansion, si {a, b}=6U, {a, c}=5.5U et {b, c}= 7U on choisit le sous-ensemble {a, b} et on ignore {b, c}, car l'algorithme *BestFirst* ne retourne en arrière que si le score ne s'améliore pas. Maintenant, il nous reste seulement l'attribut {c} dans l'espace, et {a, b, c}= 7U selon le score d'évaluation. Alors, dans ce cas-là, *Wrapper-BestFirst* retourne le sous-ensemble {a, b, c} = 7U, ainsi {b, c}=7U.

Alors, en raison de l'ignorance de ce dernier cas dans *Wrappers*, on propose *WrappersPlus,* qui vise à diminuer le nombre d'attributs pertinents en utilisant la formule de GainRatio.

À cet effet, après qu'on ait sélectionné les attributs pertinents avec la méthode *Wrapper-BestFirst,* on arrange les attributs pertinents avec l'heuristique *GainRatio*, puis on supprime l'attribut présentant le moins de *Rank*. On évalue le nouveau sous-ensemble d'attributs en utilisant le réseau bayésien et la métrique ROC pour nous assurer que le score AUROC n'a pas changé. On continue le processus de suppression et d'évaluation jusqu'à ce que le score AUROC diminue. Dans ce cas-là, on retourne le sous-ensemble d'attributs avec le dernier attribut supprimé. Pour valider cette nouvelle proposition que nous avons appelée *WrappersPlus,* nous utilisons huit bases d'apprentissages. L'effet positif de cette méthode est apparu clairement pour six d'entre elles.

Tableau 4.2 : *Wrappers* par rapport notre proposition *WrappersPlus*, en utilisant les algorithmes obtenus pour fournir un modèle MPAR

Nom db	Sans-MPAR		MPAR		WrappersPlus		
	AUROC	NbA	AUROC	NbA	AUROC	NbA	NbS
Cancer	99.1 %	30	99.6 %	14	99.6 %	13	1
Splice	99.4 %	60	99.5 %	31	99.5 %	23	8
Spectfheart	84 %	44	92.1 %	15	92.1 %	14	1
dermatology	99.9 %	34	100 %	19	100 %	18	1
Inosphere	94.7 %	33	97.8 %	16	-	-	-
DataFars	95.3 %	29	96 %	15	-	-	-
Thyroid	99.8 %	21	99.9 %	6	99.9 %	5	1
Movement Libras	94.9 %	90	98.8 %	31	98.8 %	30	1

Les tirets (-) dans le tableau 4.2 signifient que WrappersPlus n'influe pas sur les attributs pertinents obtenus en utilisant Wrappers.

Sans-MPAR signifie qu'on applique le réseau bayésien directement à la base d'apprentissage originale, sans faire la discrétisation de *Fayyad & Irani's MDL,* sans appliquer *Wrapper-BestFirst* pour sélectionner les attributs pertinents et sans utiliser TAN pour créer le réseau de dépendance, mais en utilisant k2 avec un nombre de parents égal à 1. En d'autres termes, nous utilisons la configuration par défaut de Weka.

MPAR (Modèle de prédiction performant, autonome et raffiné) signifie qu'on applique les algorithmes de la figure 3.9.

WrappersPlus est l'extension de la méthode *Wrappers* avec *BestFirst.* Cette méthode peut minimiser le nombre d'attributs et garder la même capacité prédictive si on applique *Wrapper-BestFirst,* le réseau bayésien et TAN.

NbA désigne le nombre d'attributs dans la base d'apprentissage.

NbS est le nombre d'attributs supprimés lorsqu'on applique *WrappersPlus* tout en gardant la même capacité prédictive que lorsqu'on utilise la méthode *Wrappers*.

Les bases d'apprentissages qui ont été utilisées (Tableau 4.2) sont issues de deux sources différentes: *datazar.com* et *keel.es*.

4.6.1 Validation générale du modèle proposé

Selon le tableau 4.2, nous constatons que l'application du modèle de prédiction performante, autonome et rafiné (MPAR) décrit dans le chapitre III, augmente clairement la précision de prédiction, et diminue le nombre d'attributs pertinents.

Par exemple, en appliquant le MPAR sur la base d'apprentissage concernant le *Single Photon Emission Computed Tomography for heart* (Spectfheart), la précision de prédiction s'augmente 8.10%, ainsi que le MPAR détecte 29 attributs non pertinents.

4.6.2 Résultat de l'application de *WrappersPlus* sur huit différentes bases d'apprentissage

Suite à l'expérimentation sur huit bases d'apprentissage différentes (Tableau 4.2), nous observons que la méthode *WrappersPlus* que nous proposons peut diminuer le nombre d'attributs pertinents jusqu'à huit par rapport au *Wrappers-BestFirst*, comme dans le cas de la base d'apprentissage *Splice* (Tableau 4.2),.

De plus, cette expérimentation a validé les étapes que nous avons proposées dans le cas de la MPOC, pour créer un modèle général (MPAR) indépendant d'une maladie spécifique.

CONCLUSION

L'automatisation de la détection de l'exacerbation chez les patients atteints de maladies pulmonaires obstructives chroniques (MPOC) est récente. Notre revue de littérature a permis d'identifier quatre défis principaux dans le domaine MPOC, portant sur dix-sept systèmes d'intervention informatique, visant à aider le personnel médical à identifier les attributs pertinents de la MPOC et à détecter les risques d'exacerbation chez les patients. En particulier, les systèmes informatiques de prédiction et d'intervention présentent des lacunes en ce qui concerne l'aspect «automatisation» des systèmes de prédiction et i) la faible «capacité prédictive» de ces systèmes, ii) ainsi, le manque d'algorithmes utilisés dans la phase de sélection des attributs pertinents, de discrétisation et de traitement, et iii) l'absence de la tâche d'arrangement de contexte pertinent.

À cet effet, notre étude s'est concentrée sur l'amélioration de l'infrastructure logique des systèmes informatiques existants, afin de développer un modèle de prédiction performant capable d'une précision élevée (AUROC= 81,5), et autonome puisqu'il utilise des algorithmes qui font la discrétisation et la création du réseau de croyance au lieu de référer à un expert médical. De plus, nous avons comparé dans ce mémoire quatre algorithmes de sélection d'attributs pertinents et quatre méthodes de discrétisation en utilisant quatre algorithmes d'apprentissage sur la MPOC, afin de valider le modèle de prédiction proposé (Figure 3.2). En outre, nous avons proposé une ontologie qui décrit un système automatique capable de surveiller les patients atteints de MPOC.

D'ailleurs, nous avons appliqué l'ensemble des algorithmes sélectionnés aux étapes d'apprentissage (discrétisation, sélection, dépendance et raisonnement) et ils sont tous intégrés dans notre application contextuelle qui détecte les risques d'exacerbation. Dans cette application, nous avons proposé l'utilisation de l'heuristique *GainRatio* pour arranger les attributs pertinents, et ce, dans le but de garder une bonne précision de la prédiction en cas d'urgence. Dans le même contexte, nous avons utilisé huit bases d'apprentissage portant sur des maladies différentes, afin de confirmer l'utilité des algorithmes utilisés (Figure 3.9) dans le modèle proposé (Figure 3.2). Nous avons aussi proposé *WrappersPlus,* qui permet de sélectionner les attributs pertinents. Cette méthode permet de minimiser le nombre d'attributs pertinents tout en conservant la même capacité prédictive que si on utilise *WrappersBestFirst*. De plus, grâce au réseau bayésien, cette application contextuelle est capable de faire l'inférence sans nécessiter de connexion internet ou encore de faire le diagnostic d'une situation individuelle et d'offrir une haute précision de détection des exacerbations malgré des données partielles sur l'état du patient.

En conclusion, nous avons proposé, conçu et validé un modèle prédictif performant (AUROC =81.5 %), capable d'évoluer dans le temps facilement en utilisant une collection d'algorithmes d'apprentissage au lieu d'impliquer le recours à des experts. Ce modèle met en pratique plusieurs nouveaux concepts, comme l'arrangement du contexte et l'ordonnancement entre la sélection d'attributs et la discrétisation. Ainsi, l'application de la discrétisation automatique à la MPOC, est une autre contribution claire permettant d'améliorer la performance et la précision de modèle de prédiction proposé.

En nous basant sur cette étude, dans nos travaux futurs, nous espérons réussir à combiner l'arbre de décision avec les attributs pertinents obtenus, pour sélectionner en temps réel les attributs les plus pertinents par rapport à l'état choisi. Aussi, nous voulons:

1. Appliquer notre modèle de prédiction à d'autres études de cas pour pouvoir le généraliser.
2. Adapter notre application contextuelle à l'environnement téléphonique.
3. Intégrer la méthode *GainRatio* avec *Wrappers* intérieurement, dans une solution hybride.

APPENDICE A

EXEMPLES PRATIQUES : LES ALGORITHMES D'APPRENTISSAGES

A.1 Arbre de décision

A.1.1 Exemple de calcul de l'entropie

Exemple : Soit un attribut T qui représente *la classe d'attribut* dans la figure 1.5, a deux états *Yes* et *No*. La probabilité d'état '*Yes*' = P_+, et la probabilité d'état '*No*' = P-, ainsi que $P_+ + P_- = 1$.
Alors, l'entropie de T égale :

$$E(T) = -p_+ Log_2(p_+) - p_- Log_2(p_-)$$

Si P(T) = (0.5, 0.5) c'est-à-dire $P_+ = P_-$, alors E(T) = 1 (mauvais cas).
Si P(T) = (0.67, 0.33) alors E (T) = 0.92.
Si P(T) = (1, 0) alors E(T) = 0 (cas parfait, la classe d'attribut est pure).

A.1.2 Exemple d'ID3

Comment peut-on construire l'arbre de décision en se basant sur l'ID3 ?
Supposons que nous avons la base d'apprentissage de la figure 1.5.

Donc, pour appliquer l'algorithme ID3, on commence à calculer l'entropie de la classe d'attribut, *Play Golf* :

$$Entropy\ (PlayGolf) = -\frac{5}{14}Log_2\left(\frac{5}{14}\right) - \frac{7}{14}Log_2\left(\frac{7}{14}\right) = 0.94$$

Ensuite, On applique l'*Information Gain* sur chaque attribut, comme suit :

- Gain(PlayGolf,Outlook) = 0.94 – $\frac{|T_{Rainy}|}{|T|}\ Entropy\ (T_{Rainy})$ – $\frac{|T_{Overoast}|}{|T|}\ Entropy\ (T_{Overoast})$ – $\frac{|T_{sunny}|}{|T|}\ Entropy\ (T_{sunny})$ = 0.94 – (5/14) (0.97) – (4/14)(0) – (5/14)(0.97)= 0.2471.

Même chose pour calculer :

Gain(PlayGolf,Temp)=0.02, Gain(PlayGolf,Humidity)=0.145, et Gain(PlayGolf,Windy)=0.04.

Nous concluons que l'attribut *Outlook* a le *Gain* le plus grand. C'est-à-dire, les états de l'attribut *Outlook*, ont la capacité la plus grande de faire la classe d'attribut *PlayGolf* pure. Maintenant, sur chaque état dans l'attribut *Outlook*, on calcule l'*Information Gain* de tous les attributs qui ne sont pas existés dans le chemin. Ce dernier représente les attributs de la racine jusqu'à l'attribut actuel. Dans notre cas, l'attribut actuel est *Outlook,* ainsi que le chemin est représenté par Outlook seulement. Donc, on calcule l'*Information Gain* de tous les attributs sauf l'Outlook.

Ce processus est répété jusqu'à ce que nous obtenions des cas ayant la classe d'attribut pure (Hssina *et al.*, 2014).

A.2 Naïf bayésien

A.2.3 Exemple 1

Comment peut-on construire le naïf bayésien par la base d'apprentissage de la figure 1.5 ?

Les probabilités a posteriori $P\ (\ F_i \mid C\ )$ doivent être calculées en premier lieu, en construisant un tableau de fréquence pour chaque attribut. Un exemple du tableau de fréquence pour l'attribut Outlook est comme suit :

Ce tableau est basé sur la base d'apprentissage de la figure 1.5.

Tableau A.1 : Tableau de fréquence.

Le tableau de fréquence d'Outlook		**PlayGolf**		
		Yes	No	
Outlook	Sunny	**3/9**	2/5	**5/14**
	Overcast	4/9	0/5	4/14
	Rainy	2/9	2/5	5/14
		9/14	5/14	

Selon ce tableau nous concluons le suivant :

$P(F_i/C) = P(\text{Outlook} = \text{Sunny} / \text{PalyGolf} = \text{yes}) = 3/9.$

$P(F_i) = P\ (\text{Outlook} = \text{Sunny}\ ) = 5/14\ .$

$P(C) = P\ (\ \text{PlayGolf} = \text{Yes}) = 9/14\ .$

$P(C/F_i)=P(\text{PalyGolf=yes/Outlook=Sunny})=$

$$\frac{P(\text{Outlook} = \text{Sunny}\ /\text{PalyGolf} = \text{yes}).P\ (\ \text{PlayGolf} = \text{Yes})}{P\ (\text{Outlook} = \text{Sunny}\ )} = \frac{(3/9)(9/14)}{(5/14)} = 0.6$$

A.2.4 Exemple 2

Supposons que l'attribut *Humidity* (Figure 1.5) ayant des valeurs continues. Donc, l'algorithme d'apprentissage naïf bayésien utilise la distribution normale pour calculer la probabilité par rapport de la classe d'attribut *PlayGolf*. un exemple est comme suit:

Tableau A.2 : Tableau pour calculer la moyenne et l'écart-type.

	Humidity		**Moyenne**	**Ecart-type**
Play Golf	Yes	86, 96, 80, 65, 70, 80, 70, 90, 75	79.1	10.2
	No	85, 90, 70, 95, 91	86.2	9.7

Alors $P(F_i/C)=P(\text{humidity}=74 /\text{PlayGolf} = \text{Yes}) = f(74) = \frac{1}{\sqrt{2\pi}10.2} e^{\frac{-(74-79.1)^2}{2(10.2)^2}}$

$= 0.0344$.

A.3 Réseau bayésien

A.3.5 Relation d'indépendance conditionnelle

En général, la relation d'indépendance conditionnelle (D-séparation) dans un réseau bayésien est expliquée par l'algorithme de "Bayes Ball" (Shachter, 1998), qui indique dans quel cas les deux nœuds N_1 et N_2 sont conditionnellement indépendants étant donné un nœud N_3. Pour mieux illustrer ces propos, voyons la figure suivante.

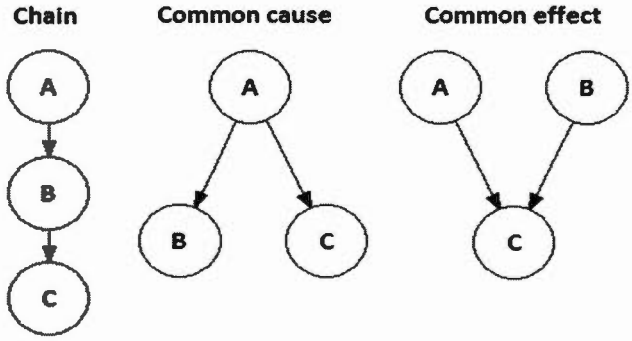


Figure A.1 : Configuration possible de trois variables adjacentes dans un réseau bayésien.

Dans la configuration en série (*chain*) A et C sont dépendant, si on ne connaît pas (n'est pas observé) la valeur de B, une fois B connu, A et C deviennent indépendants. Dans la configuration de cause commune, B et C sont dépendants si on ne connaît pas A, une fois A connu, B et C deviennent indépendants. Dans la configuration d'effet commun, A et B sont indépendant si l'effet commun C n'est pas observé, si l'état de C devient connu, A et B deviennent dépendants.

A.3.6 Exemple du réseau bayésien

Dans cet exemple, nous allons utiliser la méthode d'inférence exacte. En fait, dans le réseau bayésien il y a plusieurs types d'inférences pour réduire la complexité de calcul, comme l'inférence par élimination des variables, l'inférence approximative, etc. (Olivier, 2006).

Supposons que nous avons la base d'apprentissage du tableau A.3, qui concerne les patients atteints *l'asthme*. Selon cette base, nous voulons connaître l'état d'un patient, s'il a l'asthme ou non. Pour répondre à cette question nous utilisons le réseau bayésien.

Tableau A.3 : Historique des patients atteint *l'asthme*.

Patients	**Temperature (T)**	**Smoke (S)**	**NbrOfCigarette (N)**	**Asthma (A)**
P1	37	Yes	20	No
P2	40	Yes	200	Yes
P3	37	No	0	Yes
P4	40	No	200	No
P5	40	No	20	Yes
P6	40	Yes	0	No
P7	37	No	200	No
P8	37	Yes	20	Yes

Le tableau A.3 est créé aléatoirement, l'attribut *patient* contient les identificateurs patients avec les symptômes suivants: Temperature, Smoke, et *NbrOfCigarette*. Ainsi, le résultat détermine si le patient a *l'asthme* ou no.

Normalement, la structure du réseau bayésien peut être modélisée en s'appuyant sur la connaissance d'experts, ou à partir des données. Dans ce simple exemple, nous supposons une dépendance selon notre connaissance (peut être expert).

Les tableaux de la probabilité conditionnelle (CPT), sont remplis de la manière suivante :

Exemple : P(N=200 / S = Yes) = 1/4. Selon le tableau A.3 nous avons 4 patients qui ont S=Yes, parmi ces quatre (Sachant), nous avons un qui a N=200. Alors, la probabilité = 1/4. Comme ça nous remplissons les tableaux de la probabilité dans l'image suivants :

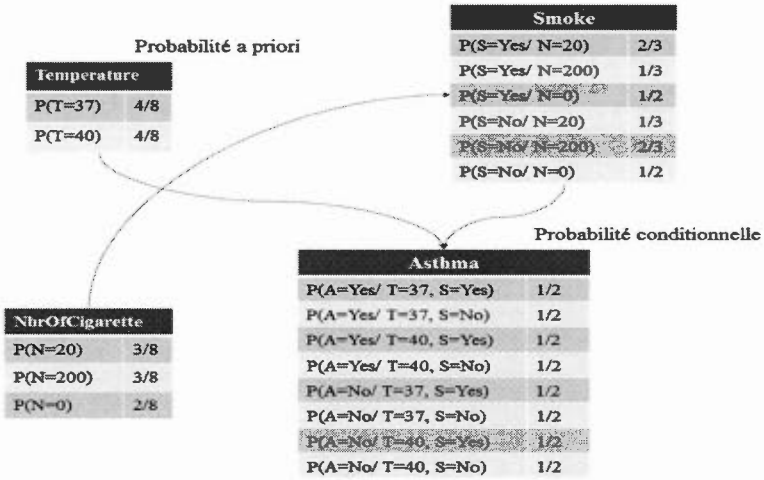


Figure A.2 : Réseau bayésien crée par les données du tableau A.3

Pour pratiquer l'inférence du réseau bayésien, nous voulons calculer le suivant :

A. $\underline{P(A = Yes \mid T = 37,\ S = Yes,\ N = 200) = ??}$

En appliquant la formule de bayes : P(A/B)= P(A,B)/P(B), nous obtenons :

$$P(A = Yes \mid T = 37, S = Yes, N = 200) = \frac{P(A = Yes,\ T = 37,\ S = Yes,\ N = 200)}{P(T = 37,\ S = Yes,\ N = 200)}$$

Pour calculer la probabilité jointe, on utilise la formule (2.11), et le figure A.2

$$= \frac{P(A=Yes \mid T=37,\ S=Yes).\ P(T=37).\ P(S=Yes \mid N=200).\ P(N=200)}{P(T=37)\ .\ P(S=Yes \mid N=200)\ .\ P(N=200)}$$

$= P\ (A=Yes \mid T=37,\ S=Yes) = 1/2.$

Ce résultat est attendu, car, selon la configuration en série de la (figure A.1), les attributs *Asthma* et *NbrOfCigarette* vont être indépendants lorsque la classe *Smoke* est observée. Donc, $P(A = Yes \mid T = 37,\ S = Yes,\ N = 200) = P(A=Yes \mid T=37,\ S=Yes)$

B. $\underline{P(A = Yes, T = 40) = ??}$

Pour calculer cette probabilité, il faut utiliser la formule de marginalisation.

La probabilité marginale =

$$P(A = Yes, T = 40) = \sum_n \sum_s P(A = Yes, T = 40, S, N)$$

$P(A = Yes, T = 40) =$

$$= \sum_n \sum_s P(A = Yes \mid T = 40) . P(T = 40) . P(N) . P(S \mid N)$$

$$= P(T = 40) . \sum_n \sum_s P(A = Yes \mid T = 40) . P(N) . P(S \mid N)$$

$$= P(T=40) . \{ (P(A=Yes \mid T=40, S=Yes) . P(N=0) + P(S=Yes|N=0)$$

$$+ P(A=Yes \mid T=40, S=Yes) . P(N=20) + P(S=Yes|N=20)$$

$$+ P(A=Yes \mid T=40, S=Yes) . P(N=200) + P(S=Yes|N=200))$$

$$+$$

$$(P(A=Yes \mid T=40, S=No) . P(N=0) + P(S= No |N=0)$$

$$+ P(A=Yes \mid T=40, S= No) . P(N=20) + P(S= No |N=20)$$

$$+ P(A=Yes \mid T=40, S= No) . P(N=200) + P(S= No |N=200))$$

$$\}$$

Selon les CPTs a priori et conditionnel, les probabilités sont comme suivantes:

$$=4/8 \{ ((½ . 2/8 + ½) + (½ . 3/8 . 2/3) + (½ . 3/8 . 1/3))$$

$$+ ((\text{½} \,.\, 2/8 + \text{½}) + (\text{½} \,.\, 3/8 \,.\, 1/3) + (\text{½} \,.\, 3/8 \,.\, 2/3))\}$$

$= \text{¼}$.

Pour assurer le résultat, nous pouvons consulter le tableau d'apprentissage (Tableau A.3), là où nous avons trouvé que $P(A = Yes, T = 40) = \frac{2}{8} = \frac{1}{4}$.

En effet, le réseau bayésien n'ajoute pas de plus sur le tableau d'apprentissage (Tableau A.3), qui représente toutes les combinaisons possibles de la probabilité conjointe, telle que leur somme =1. Ainsi que, le réseau bayésien nous permet de calculer n'importe quelle probabilité impliquant les attributs dans le réseau.

Nous trouvons que ces exemples, peuvent expliquer les trois algorithmes (arbre de décision, naïf bayésien, réseau bayésien) d'apprentissage de façon très simple sans avoir besoin de consulter beaucoup de références.

A.4 Différence entre l'apprentissage supervisé et non supervisé

L'explication suivante est basée sur le livre (Murphy, 2012).

A.4.7 Approche d'apprentissage prédictif ou supervisé.

Dans l'apprentissage supervisé, le but est d'apprendre une correspondance à partir d'entrées X aux sorties Y, étant donné un ensemble d'entraînement de paires d'entrées-sorties $D = \{ (X_i, Y_i) \}_{i=1}^{N}$, N est le nombre de cas dans la base d'apprentissage. X_i représente l'entrée, par exemple : l'âge, le gendre, le poids du patient, etc. Ceux-ci sont appelés des caractéristiques (*features*) ou attributs. Dans le cas de classification Y_i est une variable catigoral (ensemble fini des valeurs) et il s'appelle la classe d'attribut, par

exemple : le patient a deux états (Exacerbation Ou Non-Exacerbation). Lorsque Y_i est une valeur réelle, le problème est connu comme une régression.
Généralement, l'apprentissage supervisé est couramment utilisé pour prédire des événements futurs probables.

A.4.8 Approche descriptive ou l'apprentissage non supervisé.

Dans ce type d'apprentissage, nous avons seulement les données d'entrées $D = \{ X_i \}_{i=1}^{N}$ qui ne sont pas marquées (c'est-à-dire sans la classe d'attribut Y_i). L'objectif ici, est de trouver dans les données, des modèles qui décrivent une structure cachée. Cette approche n'a pas une métrique évidente pour l'évaluation du modèle détecté. D'autre part, dans l'apprentissage supervisé, la métrique d'erreur ou d'évaluation est existée, en cachant une partie teste de Y_i, puis on prédit cette partie, et à la fin, on compare ce que nous prédisons à ce que nous avons caché de Y_i.
Pour faire la différence entre ces deux types d'apprentissage, un exemple d'image est intéressant. L'apprentissage non supervisé peut répondre à la question, *combien de différentes classes ou phénotype cellulaire dans cette image?* Ainsi, l'apprentissage supervisé peut répondre à la question, *combien de cellules de la classe A et combien de cellules de la classe B sont dans cette image? (Davies et al., 2014).*

APPENDICE B

L'ÉLARGISSEMENT D'ONTOLOGIE

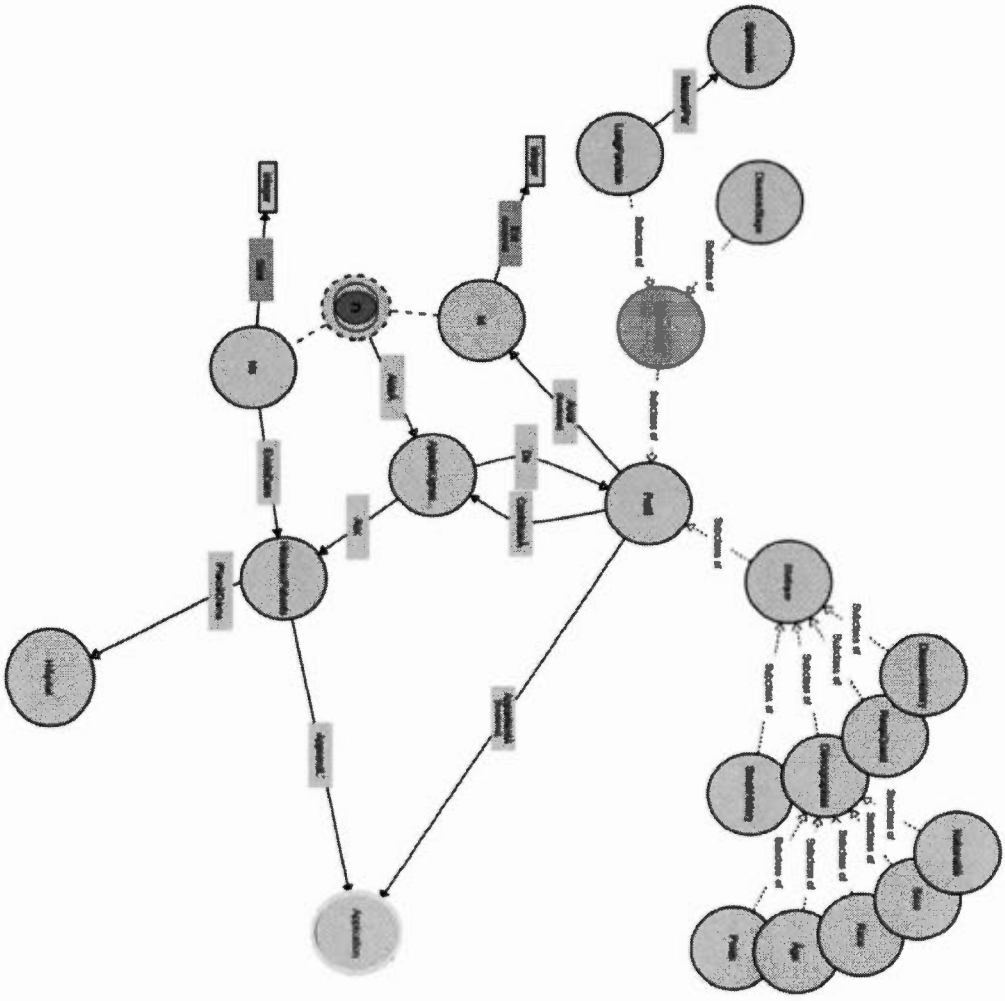


Figure B.1 : Élargissement de la figure 3.3, partie 1.

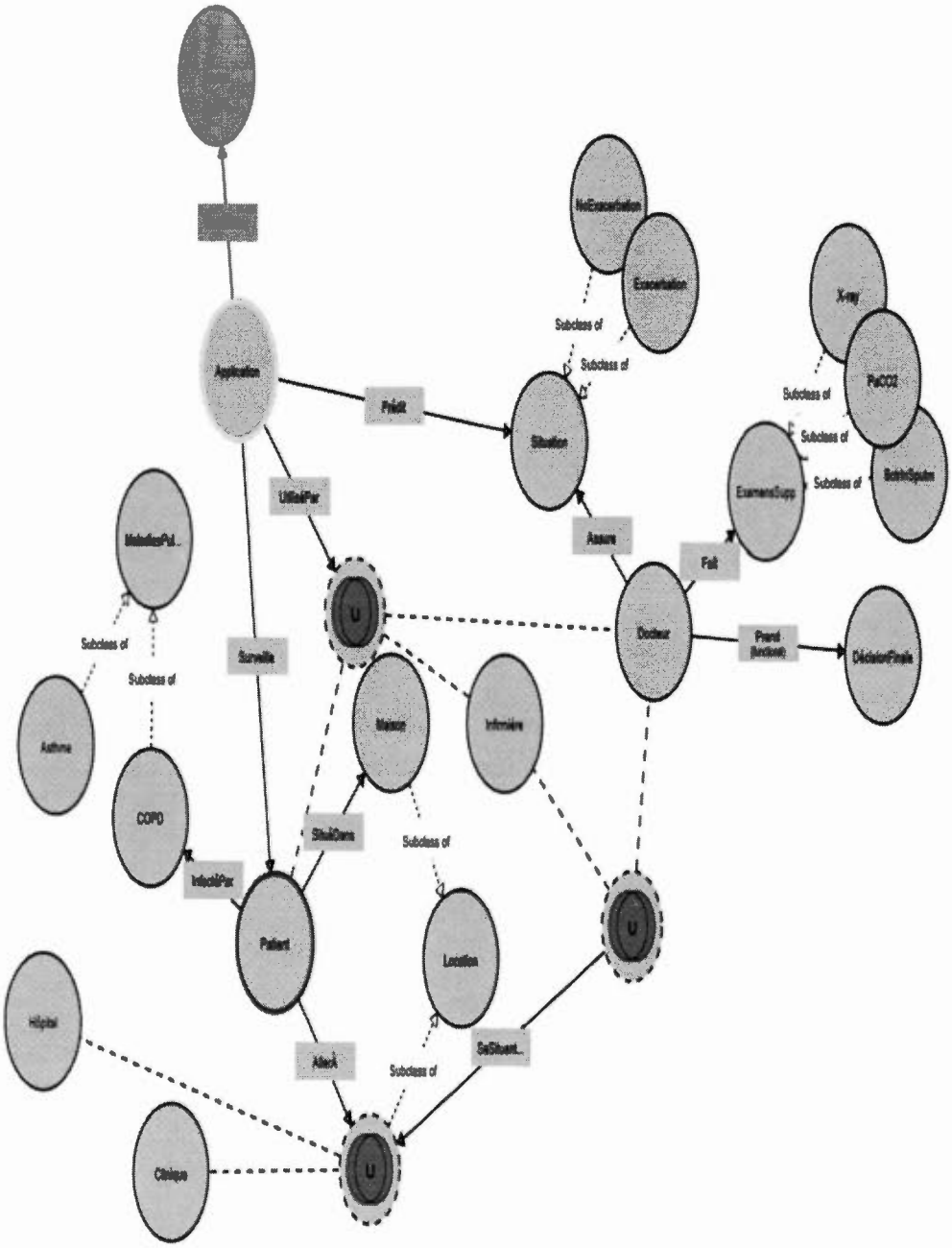


Figure B.2 : Élargissement de la figure 3.3, partie 2.

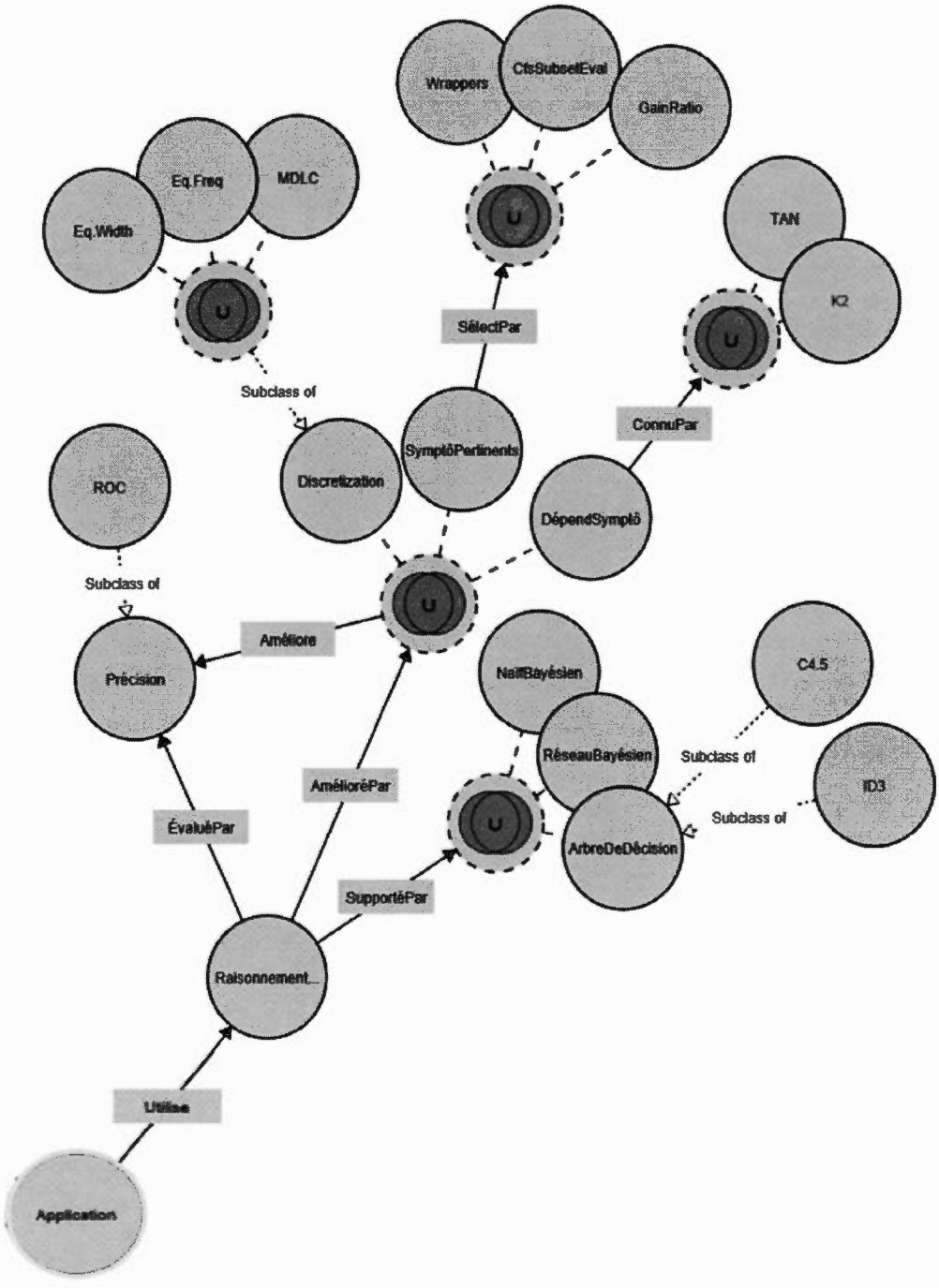


Figure B.3 : Élargissement de la figure 3.3, partie 3.

APPENDICE C

CODE D'IMPLÉMENTATION D'APPLICATION CONTEXTUELLE

C.1 Classe Network

```
package reseau.bayesien;
import norsys.netica.Environ;
import static norsys.netica.Environ.INFINITY;
import norsys.netica.Net;
import norsys.netica.Streamer;
import norsys.neticaEx.aliases.Node;

public class Network {

    public  void CreationN(){
      try {
// Licence a obtenu par netica, pour être capable d'enregistrer le réseau
        Environ env = new Environ("+SalehL/UQAM/Ex17-02-15,120,.....");
        Net net = new Net();
        net.setName("MPOC");
//Exacer   demo2   demo5  dishis1time  dishis3times ... etc.
        /*Le constructeur de la classe «Node» a deux paramètres, le premier est le nom
du nœud, et la deuxième indique si l'attribut est continu ou discret. S'il est continu on
met 0; et s'il est discret, on met le nombre des états. Le troisième paramètre, c'est le
pointeur du réseau */
        Node Exacer = new Node("exacer", 2, net);// Nœud discrète, avec 2 states ou
états
        Node demo2 = new Node("demo2", 0, net); // Nœud continue
        Node demo5 = new Node("demo5", 2, net); // Nœud discrète, avec 2 states....
        Node dishis1time = new Node("dishis1time", 0, net);
        Node dishis3times = new Node("dishis3times", 0, net);
        Node dishis7 = new Node("dishis7", 2, net);
```

```
Node disstage2 = new Node("disstage2", 8, net);
Node lungfun5 = new Node("lungfun5", 0, net);
Node lungfun8 = new Node("lungfun8", 0, net);
Node lungfun11 = new Node("lungfun11", 0, net);
Node lungfun12 = new Node("lungfun12", 0, net);
Node dis1treat = new Node("dis1treat", 2, net);
Node dis2 = new Node("dis2", 3, net);
Node dis2times = new Node("dis2times", 0, net);
Node dis3times = new Node("dis3times", 0, net);
Node resques1a = new Node("resques1a", 0, net);
Node resques1b = new Node("resques1b", 0, net);
Node resques1c = new Node("resques1c", 0, net);
```

/* Dans les instructions suivantes, nous allons préciser les intervalles des discrétisations pour les attributs continus en se basant sur l'algorithme Fayyad & Irani's MDL. En plus, nous donnons un nom pour chaque état */

```
double[] levels1 = new double[2];  // Nœud discret
levels1[0] = 0;    // Le premier état =0
levels1[1] = 1;  // Le deuxième état = 1
Exacer.setLevels(levels1);
Exacer.setStateNames("NonExacer,Exacer");  // Pour nommer les états

double[] levels2 = new double[3];  // Nœud continue.
levels2[0] = 0;  // Le premier état ayant un intervalle entre 0 et 0.258
levels2[1] = 0.2851; // Le deuxième état ayant un intervalle est entre 0.258 à 1
ou infini
levels2[2] = Environ.INFINITY;;
demo2.setLevels(levels2);
demo2.setStateNames("a, b");

double[] levels3 = new double[2];
levels3[0] = 0;
levels3[1] = 1;
demo5.setLevels(levels3);
demo5.setStateNames("a, b");

double[] levels4 = new double[3];
levels4[0] = 0;
levels4[1] = 0.41667;
levels4[2] = Environ.INFINITY;
dishis1time.setLevels(levels4);
dishis1time.setStateNames("a, b");
```

```
double[] levels5 = new double[3];
levels5[0] = 0;
levels5[1] = 0.41667;
levels5[2] = Environ.INFINITY;;
dishis3times.setLevels(levels5);
dishis3times.setStateNames("a, b");

double[] levels6 = new double[2];
levels6[0] = 0;
levels6[1] = 1;
dishis7.setLevels(levels6);
dishis7.setStateNames("a, b");

double[] levels7 = new double[8];
levels7[0] = 0.000;
levels7[1] = 0.1428571;
levels7[2] = 0.285714286;
levels7[3] = 0.428571429;
levels7[4] = 0.571428571;
levels7[5] = 0.714285714;
levels7[6] = 0.857142857;
levels7[7] = 1;
disstage2.setLevels(levels7);
disstage2.setStateNames("a, b, c, d, e, f, g, h");

double[] levels8 = new double[3];
levels8[0] = 0;
levels8[1] = 0.390419;
levels8[2] = Environ.INFINITY;
lungfun5.setLevels(levels8);
lungfun5.setStateNames("a, b");

double[] levels9 = new double[3];
levels9[0] = 0;
levels9[1] = 0.055848;
levels9[2] = Environ.INFINITY;
lungfun8.setLevels(levels9);
lungfun8.setStateNames("a, b");

double[] levels10 = new double[3];
levels10[0] = 0;
levels10[1] = 0.450008;
levels10[2] = Environ.INFINITY;
```

```
lungfun11.setLevels(levels10);
lungfun11.setStateNames("a, b");

double[] levels11 = new double[3];
levels11[0] = 0;
levels11[1] = 0.569149;
levels11[2] = Environ.INFINITY;
lungfun12.setLevels(levels11);
lungfun12.setStateNames("a, b");

double[] levels12 = new double[2];
levels12[0] = 0;
levels12[1] = 1;
dis1treat.setLevels(levels12);
dis1treat.setStateNames("a, b");

double[] levels13 = new double[3];
levels13[0] = 0;
levels13[1] = 0.333333333;
levels13[2] = 1;
dis2.setLevels(levels13);
dis2.setStateNames("a, b, c");

double[] levels14 = new double[3];
levels14[0] = 0;
levels14[1] = 0.046875;
levels14[2] = Environ.INFINITY;
dis2times.setLevels(levels14);
dis2times.setStateNames("a, b");

double[] levels15 = new double[4];
levels15[0] = 0;
levels15[1] = 0.025;
levels15[2] = 0.125;
levels15[3] = Environ.INFINITY;
dis3times.setLevels(levels15);
dis3times.setStateNames("a, b, c");

double[] levels16 = new double[3];
levels16[0] = 0;
levels16[1] = 0.5093;
levels16[2] = Environ.INFINITY;
resques1a.setLevels(levels16);
```

```
resques1a.setStateNames("a, b");

double[] levels17 = new double[3];
levels17[0] = 0;
levels17[1] = 0.55155;
levels17[2] = Environ.INFINITY;
resques1b.setLevels(levels17);
resques1b.setStateNames("a, b");

double[] levels18 = new double[3];
levels18[0] = 0;
levels18[1] = 0.2329;
levels18[2] = Environ.INFINITY;
resques1c.setLevels(levels18);
resques1c.setStateNames("a, b");
```

/* Après la discrétisation des attributs pertinents, nous voulons maintenant de relier les attributs entre eux pour construire le réseau bayésien, en se basant sur l'algorithme TAN */

```
demo2.addLink(Exacer);demo2.addLink(lungfun12); //Lien de l'exacer à demo2 et un autre lien de lungfun12 à demo2

demo5.addLink(Exacer);demo5.addLink(dis2);
        dishis1time.addLink(Exacer);dishis1time.addLink(dishis7);
        dishis3times.addLink(Exacer);dishis3times.addLink(dishis);
dishis7.addLink(Exacer);
disstage2.addLink(Exacer);disstage2.addLink(dishis7);
lungfun5.addLink(Exacer);lungfun5.addLink(lungfun8);
lungfun8.addLink(Exacer);lungfun8.addLink(disstage2);
lungfun11.addLink(Exacer);lungfun11.addLink(lungfun5);
lungfun12.addLink(Exacer); lungfun12.addLink(disstage2);
        dis1treat.addLink(Exacer);dis1treat.addLink(dishis3times;
dis2.addLink(Exacer);dis2.addLink(dishis7);
dis2times.addLink(Exacer);dis2times.addLink(dis2);
dis3times.addLink(Exacer);dis3times.addLink(dis2times);
resques1a.addLink(Exacer);resques1a.addLink(resques1c);
resques1b.addLink(Exacer);resques1b.addLink(disstage2);
resques1c.addLink(Exacer);resques1c.addLink(disstage2);
```

/* À la fin de cette classe on enregistre le réseau dans un fichier qui s'appelle Exacer1.dne. Cet enregistrement est important pour relire ce réseau lors de l'exécution de la classe CPTables */

```
        Streamer stream = new
Streamer("C:\\Users\\user\\Documents\\NetBeansProjects\\Data
Files\\Exacer\\Exacer1.dne");
        net.write(stream);
        System.out.printf("BUILD 1\n");
        net.finalize();

    } catch (Exception e) {

        e.printStackTrace();}}
```

C.2 Classe CPTables

```
package reseau.bayesien;
import java.io.File;
import norsys.netica.*;

public class CPTables {
public  void Apprend() {
    try {

        // Lire le réseau créé par Netowrk.java
        Net net = new Net(new
Streamer("C:\\Users\\user\\Documents\\NetBeansProjects\\Data
Files\\Exacer\\Exacer1.dne"));
        NodeList nodes = net.getNodes();
        int numNodes = nodes.size();

        // Nettoyer les CPTs existantes dans le réseau (net)

        for (int n = 0; n < numNodes; n++) {
           Node node = (Node) nodes.get(n);
           node.deleteTables();
        }

        // Lire la base d'apprentissage, enregistré dans un fichier Exacer.cas
```

```
        Streamer caseFile = new
Streamer("C:\\Users\\user\\Documents\\NetBeansProjects\\Data
Files\\Exacer\\Exacer.cas");

        //Remplir les tableaux de probabilité (CPTs) basée sur la base
d'apprentissage (Exacer.cas)

        net.reviseCPTsByCaseFile(caseFile, nodes, 1.0);

        // Enregistrer le réseau avec les tableaux de probabilité (CPTs)

        net.write(new Streamer("C:\\Users\\user\\Documents\\NetBeansProjects\\Data
Files\\Exacer\\Appris.dne"));
        System.out.printf("Learn 1\n");
        net.finalize();
      } catch (Exception e) {
        e.printStackTrace();
      }
    }
}
```

C.3 Classe d'inférence

```
public void INFER(String attribut, String state) {
      try {
        jTextArea1.setBackground(null);
        net.getNode(attribut).finding().clear(); // Supprimer toutes les informations
relatives à cet attribut observé

        net.getNode(attribut).finding().enterState(state); // Préciser l'état observé à cet
attribut

        float NonExacer = net.getNode("exacer").getBelief("NonExacer");
        float Exacer = net.getNode("exacer").getBelief("Exacer");

        if (Exacer > 0.1) // Le cutoff que nous avons choisi = 0.1
        {
          jTextArea1.setText("....VOUS ÊTES EN RISQUE..... \nvous devez appeler
votre DOCTEUR. "
                + " \n la probabilité d'avoir une exacerbation est \n "
```

```
                + "P(Exacerbation) = " + Exacer

            );
            jTextArea1.setBackground(Color.red);
        } else {

            jTextArea1.setText(" .....Ne vous inquiétez pas..... \n vous êtes en bonne
santé :)"
                + "\nAvec une probabilité P(NONExacerbation) = " + NonExacer);

            jTextArea1.setBackground(Color.GREEN);
        }

    } catch (NeticaException ex) {
        Logger.getLogger(InferenceFrame1.class.getName()).log(Level.SEVERE,
null, ex);
    }

}
```

APPENDICE D

RÉSULTATS DE WEKA

D.1 Méthodes de discrétisation

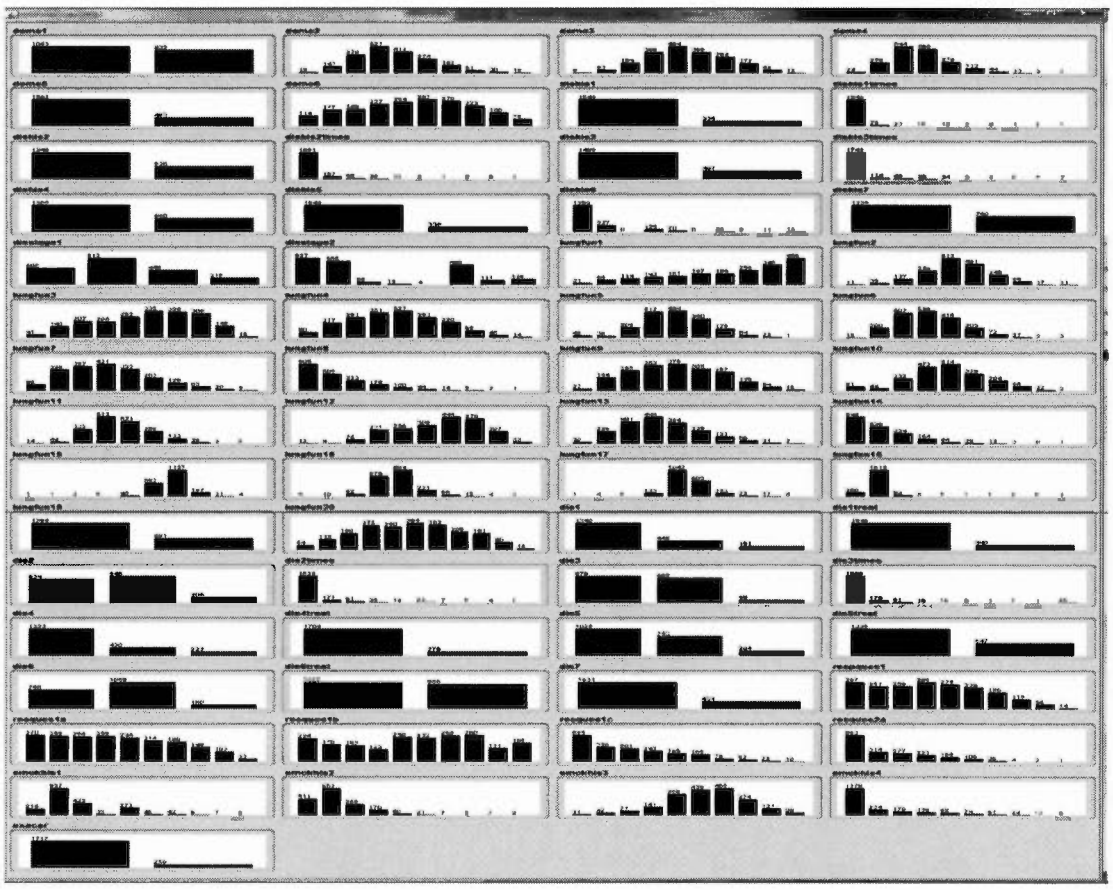


Figure D.1: Equal Width Discretization (EWD) sur 61 attributs, avec 10 bins.

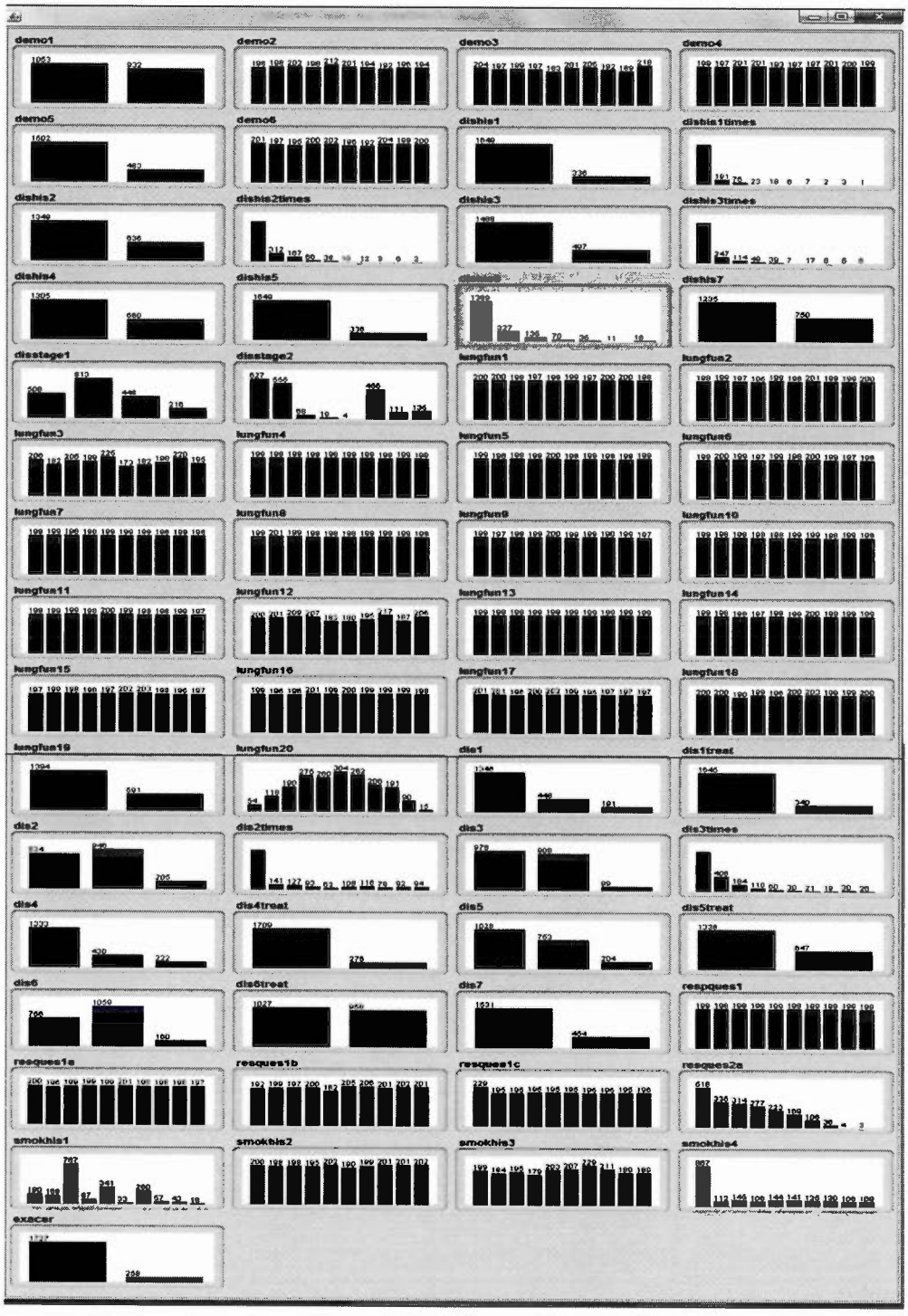


Figure D.2 : Equal Frequency Discretization (EFD) sur 61 attributs, avec 10 bins.

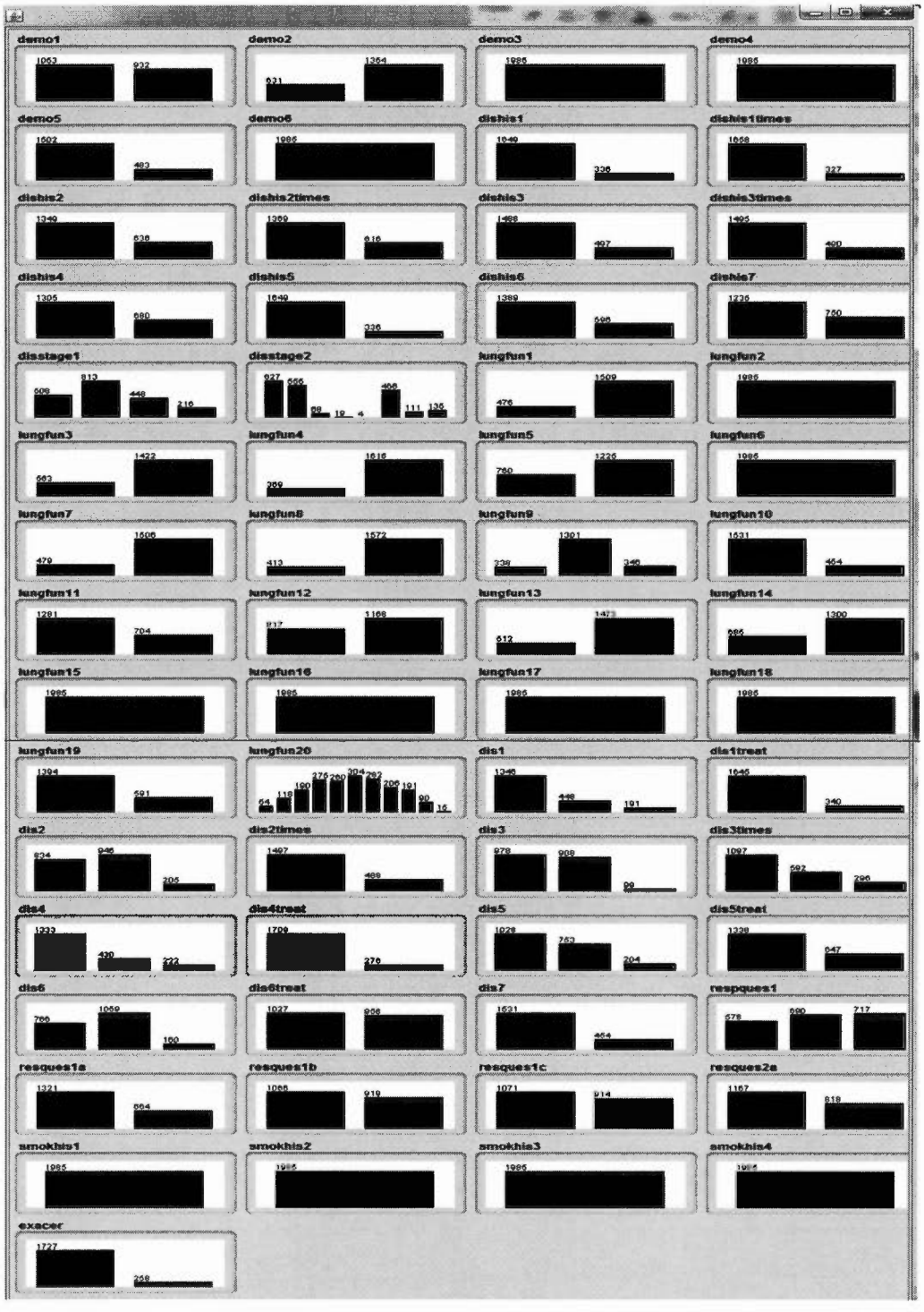


Figure D.3 : Discrétisation Fayyad & Irani's MDL *(Minimum Description Length)*

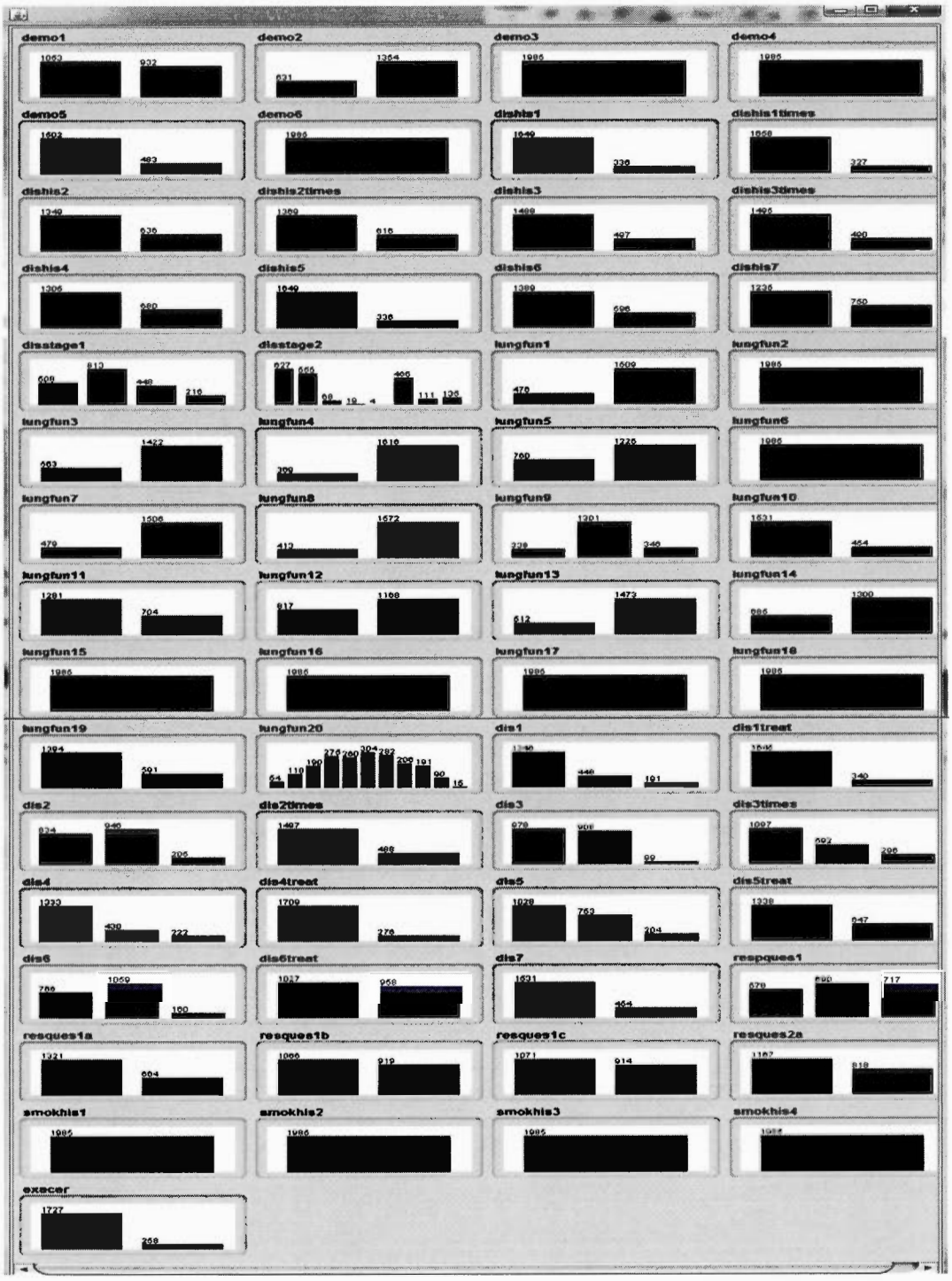


Figure D.4: Discretization Kononenko's MDL *(Minimum Description Length).*

D.2 Echantillon de notre expérimentation (Test B)

Pour éviter la répétition, ainsi que l'immensité de ce mémoire, nous allons montrer dans cette partie juste quelque résultats du teste B (Tableau 3.3).

Figure D.5 : En appliquant *CFSsubsetEval* sur les 61 attributs, nous obtenons 14 attributs pertinents.

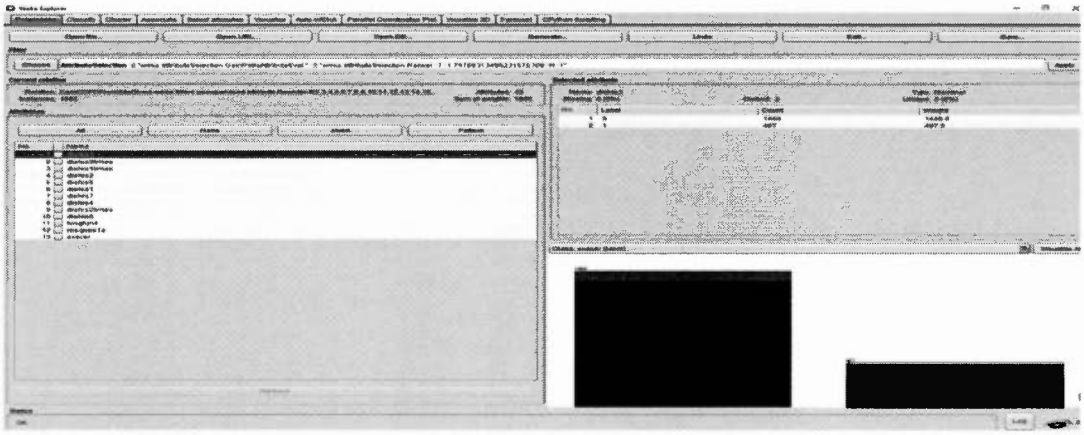


Figure D.6 : En appliquant *CFSsubsetEval* sur les 61 attributs, nous obtenons 12 attributs pertinents.

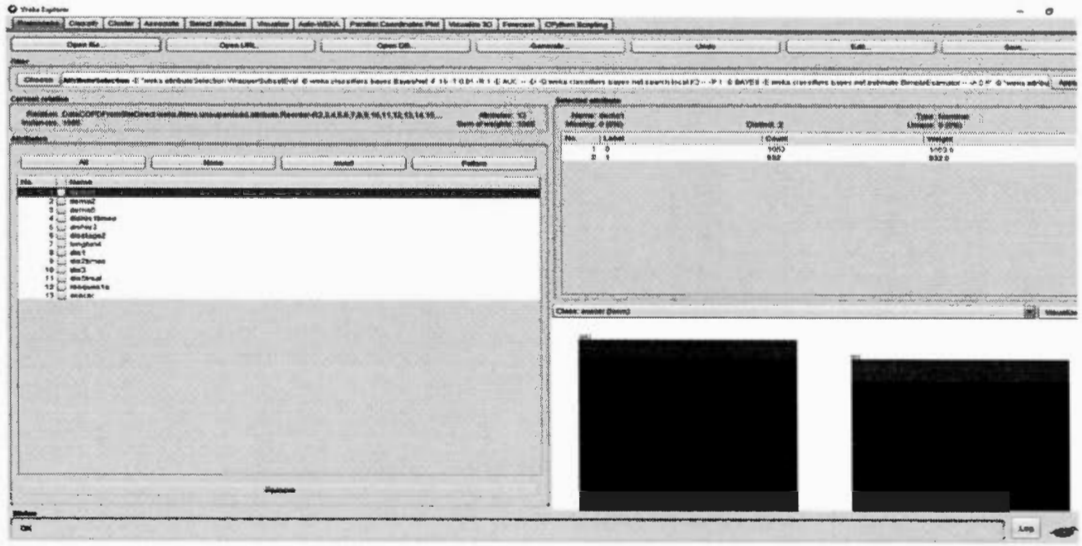


Figure D.7 : En appliquant *Wrapper-BestFirst* sur les 61 attributs, avec réseau bayésien (K2 – 1 parent et la métrique AUROC), nous obtenons 12 attributs pertinents.

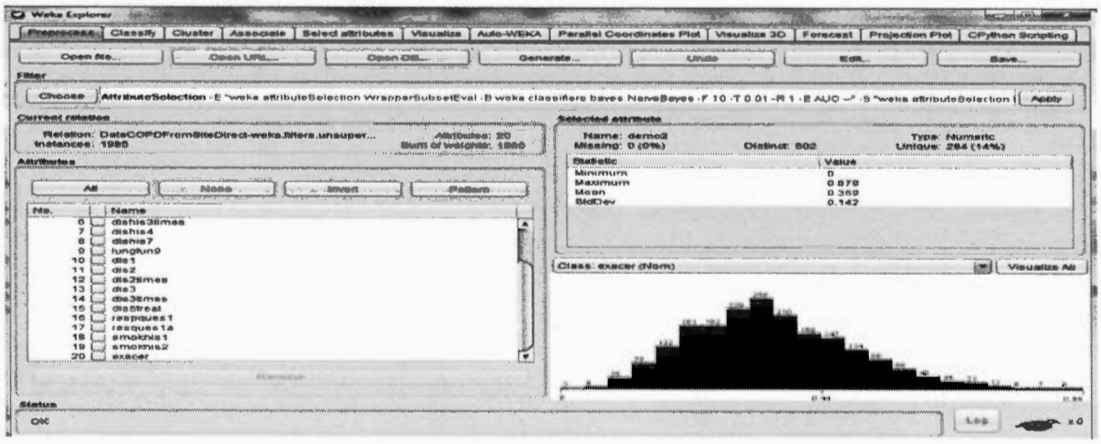


Figure D.8 : En appliquant *Wrapper-BestFirst* sur les 61 attributs, avec le naïf bayésien et la métrique AUROC, nous obtenons 19 attributs pertinents.

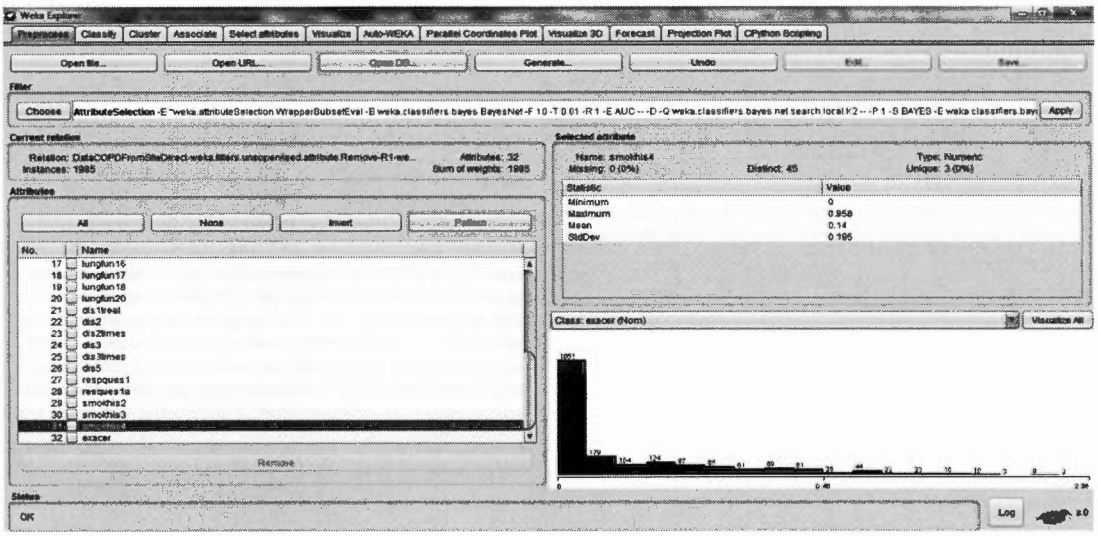


Figure D.9 : En appliquant *Wrapper-BestFirst* sur les 61 attributs, avec réseau bayésien (K2 – 1 parent et la métrique AUROC), nous obtenons 31 attributs pertinents.

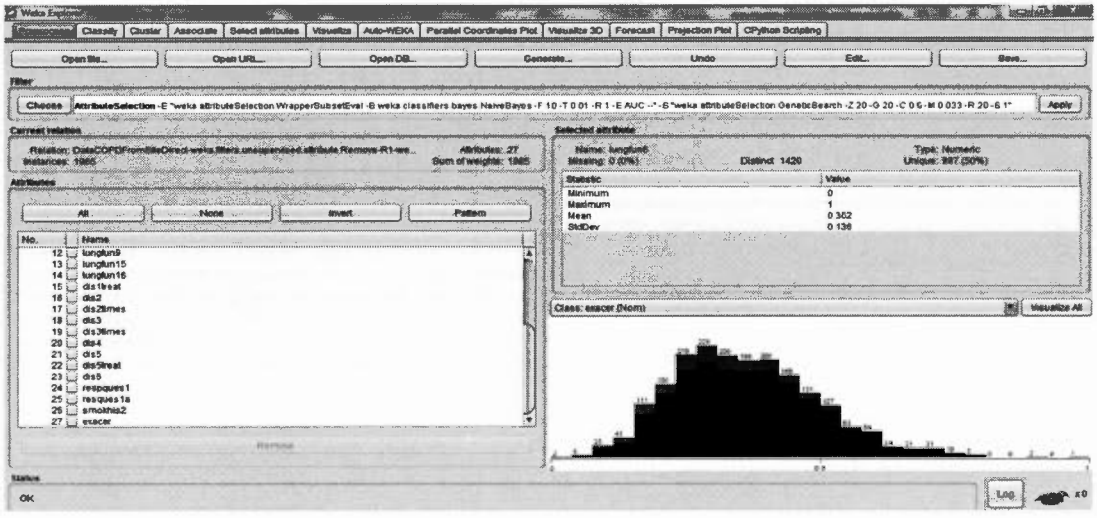


Figure D.10 : En appliquant *Wrapper-Genetique* sur les 61 attributs, avec le naïf bayésien et la métrique AUROC, nous obtenons 26 attributs pertinents.

D.3 Réseaux de croyance en utilisant TAN et k2

Dans cette partie, nous allons montrer les réseaux de croyances du Tableau 3.7.

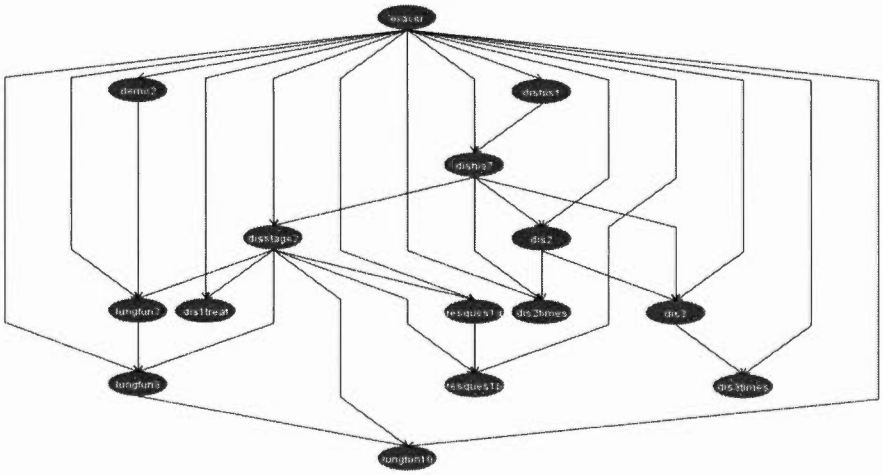


Figure D.11 : La structure du réseau bayésien avec K2 - 3 parents

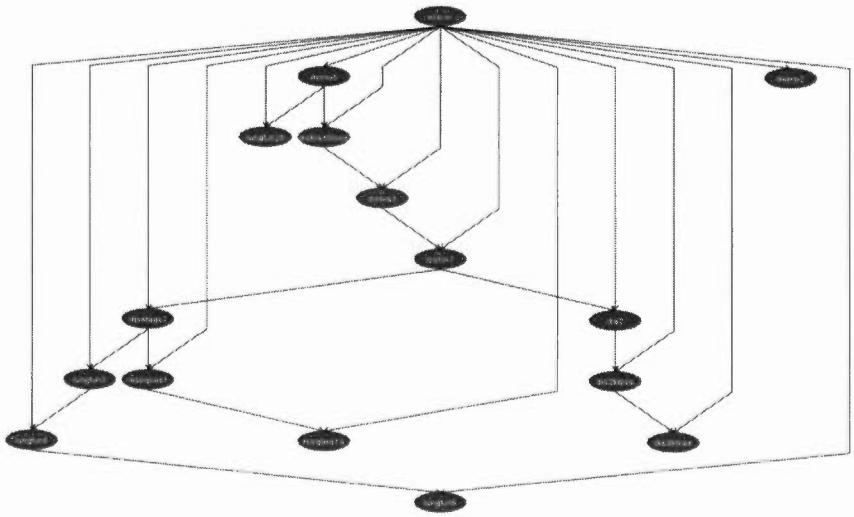


Figure D.12 : La structure du réseau bayésien avec K2 - 2 parents

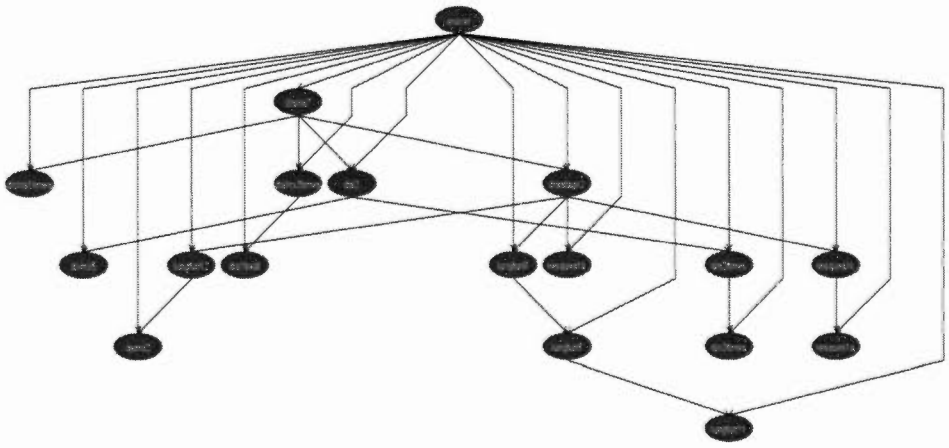


Figure D.13 : La structure du réseau bayésien avec BN(TAN)

BIBLIOGRAPHIE

Aarab, Z., Saidi, R. et Rahmani, M.D. (2016). Context Modeling and Metamodeling: A State of the Art. *Proceedings of the Mediterranean Conference on Information & Communication Technologies 2015*, 287-295.

Amalakuhan, B., Kiljanek, L., Parvathaneni, A., Hester, M., Cheriyath, P. et Fischman, D. (2012). A prediction model for COPD readmissions: catching up, catching our breath, and improving a national problem. *Journal of Community Hospital Internal Medicine Perspectives, 2*(1).

Analytix, C. (2015). Récupéré de https://www.crowdanalytix.com/contests/predict-exacerbation-in-patients-with-copd

Arroyo-Figueroa, G.S., Luis Enrique. (1999). A temporal Bayesian network for diagnosis and prediction. *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, 13-20.

Ats, A.T.S. (1995). Standards for the diagnosis and care of patients with chronic obstructive pulmonary disease. *Am J Respir Crit Care Med*. Récupéré de https://www.thoracic.org/statements/resources/archive/standards-for-the-diagnosis-and-care-of-patients-with-chronic-obstructive-pulmonary-disease-1995.pdf

Baldauf, M., Dustdar, S. et Rosenberg, F. (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing, 2*(4), 263-277.

Bangsuk, J. et Cheng-Fa, T. (2014). A Comparison of Filter and Wrapper Approaches with Data Mining Techniques for Categorical Variables Selection. *International Journal of Innovative Research in Computer and Communication Engineering*.

Berkhof, F.F., Berg, J.W., Uil, S.M. et Kerstjens, H.A. (2015). Telemedicine, the effect of nurse-initiated telephone follow up, on health status and health-care utilization in COPD patients: A randomized trial. *Respirology, 20*(2), 279-285.

Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A. et Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing, 6*(2), 161-180.

Bland, J.M. et Altman, D.G. (1996). Transforming data. *BMJ: British Medical Journal, 312*(7033), 770.

Bouckaert, R.R. (May 2008). Bayesian Network Classifiers in Weka. Récupéré de http://www.cs.waikato.ac.nz/~remco/weka.bn.pdf

Boulle, M. (2005). Optimal bin number for equal frequency discretizations in supervized learning. *Intelligent Data Analysis, 9*(2), 175-188.

Bouzy, B. et Cazenave, T. (1997). Using the object oriented paradigm to model context in computer go. *Proceedings of the first international and interdisciplinary conference on modeling and using context*, 279-289.

Brazeau, J. (2005). Les bronches et les poumons. *Bibliothèque nationale du Québec.*

Brézillon, P. (2005). Task-realization models in contextual graphs. *International and Interdisciplinary Conference on Modeling and Using Context*, 55-68.

Brown, P.J., Bovey, J.D. et Chen, X. (1997). Context-aware applications: from the laboratory to the marketplace. *IEEE personal communications, 4*(5), 58-64.

Burt, L. et Corbridge, S. (2013). COPD exacerbations. *AJN The American Journal of Nursing, 113*(2), 34-43.

Butterworth, R., Simovici, D.A., Santos, G.S. et Ohno-Machado, L. (2004). A greedy algorithm for supervised discretization. *Journal of biomedical informatics, 37*(4), 285-292.

Canada. (2007). La vie et le souffle : Les maladies respiratoires au Canada. *Ottawa: Agence de santé publique du Canada.*

Canada, S. (2010). *Le fardeau humain et financier de la MPOC*. Récupéré de http://www.lignesdirectricesrespiratoires.ca/le-fardeau-humain-et-financier-de-la-mpoc-une-des-principales-causes-d%E2%80%99hospitalisation-au-canada

Chaari, T., Laforest, F. et Flory, A. (2005). Adaptation des applications au contexte en utilisant les services Web. *Proceedings of the 2nd French-speaking conference on Mobility and ubiquity computing*, 111-118.

Chow, C. et Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory, 14*(3), 462-467.

Connors Jr, A.F., Dawson, N.V., Thomas, C., Harrell Jr, F.E., Desbiens, N., Fulkerson, W.J., Kussin, P., Bellamy, P., Goldman, L. et Knaus, W.A. (1996). Outcomes following acute exacerbation of severe chronic obstructive lung disease. The SUPPORT investigators (Study to Understand Prognoses and Preferences for Outcomes and Risks of Treatments). *American journal of respiratory and critical care medicine, 154*(4), 959-967.

Cooper, G.F. et Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine learning, 9*(4), 309-347.

Cornuéjols, A. (2006) *Sélection d'attributs*. de https://www.lri.fr/~antoine/Courses/DEA-I3/Tr-selection-attributs.pdf

Crooks, G. (2016). *Telehealth in Practice - Care Delivery Models from 14 Regions in Europe*. Récupéré de http://united4health.eu/wp-content/uploads/2016/02/U4H-Care-Process-Brochure-FINAL.pdf

Davies, A., Shamu, C., Haney, S.A., Bowman, D. et Chakravarty, A. (2014). *An introduction to high content screening: imaging technology, assay development, and data analysis in biology and drug discovery*. : John Wiley & Sons.

De Bosschere, R.J.K. (2013). Compiler Construction.

De Voogd, J.N., Wempe, J.B., Koëter, G.H., Postema, K., van Sonderen, E., Ranchor, A.V., Coyne, J.C. et Sanderman, R. (2009). Depressive symptoms as predictors of mortality in patients with COPD. *Chest Journal, 135*(3), 619-625.

Dey, A.K. (2000). *Providing architectural support for building context-aware applications*. Georgia Institute of Technology.

Dey, A.K. (2001). Understanding and using context. *Personal and ubiquitous computing, 5*(1), 4-7.

Dey, A.K. et Häkkilä, J. (2008). Context-awareness and mobile devices. *User interface design and evaluation for mobile technology, 1*, 205-217.

Dougherty, J., Kohavi, R. et Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. *Machine learning: proceedings of the twelfth international conference, 12*, 194-202.

El Ferchichi, S. (2013). *Sélection et extraction d'attributs pour les problèmes de classification.* Lille 1.

Emitzá, G.S., Zamarripa; López, Joao; Coelho, Garcia; Nitesh, Narayan; Benoit, Gaudin; Walid, Maalej;. (2010). *State-of-the-art Context Elicitation, Context Modeling and User Modeling.* Récupéré de http://fastfixrsm.sourceforge.net/fastfix-project/sites/default/files/Deliverables/D3.1_final.pdf

Fagon, J.-Y. et Chastre, J. (1996). Severe exacerbations of COPD patients: the role of pulmonary infections. *Seminars in respiratory infections, 11*(2), 109-118.

Fawcett, T. (2004). ROC graphs: Notes and practical considerations for researchers. *Machine learning, 31*(1), 1-38.

Flachaire, E. (2000). Les méthodes du bootstrap dans les modèles de régression. *Économie & prévision, 142*(1), 183-194.

Ford, E.S., Murphy, L.B., Khavjou, O., Giles, W.H., Holt, J.B. et Croft, J.B. (2015). Total and state-specific medical and absenteeism costs of COPD among adults aged≥ 18 years in the United States for 2010 and projections through 2020. *Chest Journal, 147*(1), 31-45.

Friedman, N., Geiger, D. et Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning, 29*(2-3), 131-163.

Gama, J. et Porto, L.-I. (2008). *Bayesian learning: An introduction.".*

Garcia, S., Luengo, J., Sáez, J.A., Lopez, V. et Herrera, F. (2013). A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Transactions on Knowledge and Data Engineering, 25*(4), 734-750.

Geneviève , T.M., Doucet ;. (2013). Les maladies respiratoires obstructives chroniques (la MPOC et l'asthme). *Institut national de santé publique du Québec.*

Goswami, S. et Chakrabarti, A. (2014). Feature selection: A practitioner view. *International Journal of Information Technology and Computer Science (IJITCS), 6*(11), 66.

Gray, P. et Salber, D. (2001). Modelling and using sensed context information in the design of interactive applications. Dans *Engineering for Human-Computer Interaction* (p. 317-335) : Springer.

Greenberg, S. (2001). Context as a dynamic construct. *Human-Computer Interaction, 16*(2), 257-268.

Gu, T., Pung, H.K. et Zhang, D.Q. (2005). A service-oriented middleware for building context-aware services. *Journal of Network and computer applications, 28*(1), 1-18.

Gu, T., Pung, H.K., Zhang, D.Q., Pung, H.K. et Zhang, D.Q. (2004). *A bayesian approach for dealing with uncertain contexts*. : na.

Guérif, M.S.M. (2006). *Réduction de dimension en Apprentissage Numérique Non Supervisé.* Université Paris 13.

Gutlein, M., Frank, E., Hall, M. et Karwath, A. (2009). Large-scale attribute selection using wrappers. *Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on*, 332-339.

Hall, M.A. (1999). *Correlation-based feature selection for machine learning.* The University of Waikato.

Hall, M.A. (2000). Correlation-based feature selection of discrete and numeric class machine learning.

Halpin, D.M., Laing-Morton, T., Spedding, S., Levy, M.L., Coyle, P., Lewis, J., Newbold, P. et Marno, P. (2011). A randomised controlled trial of the effect of automated interactive calling combined with a health risk forecast on frequency and severity of exacerbations of COPD assessed clinically and using EXACT PRO. *Primary Care Respiratory Journal, 20*, 324-331.

Hand, D.J. et Yu, K. (2001). Idiot's Bayes—not so stupid after all? *International statistical review, 69*(3), 385-398.

Hanley, J.A. et McNeil, B.J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology, 143*(1), 29-36.

Healthline.com. (2014) *COPD: Symptoms and Grades*. de http://www.healthline.com/health/copd/stages#Overview1

Héon, M. (2014). *Web sémantique et modélisation ontologique (avec G-OWL).* (Editions ENI éd.).

Himes, B.E., Dai, Y., Kohane, I.S., Weiss, S.T. et Ramoni, M.F. (2009). Prediction of chronic obstructive pulmonary disease (COPD) in asthma patients using electronic medical records. *Journal of the American Medical Informatics Association, 16*(3), 371-379.

Hoot, N.R. et Aronsky, D. (2005). Using Bayesian networks to predict survival of liver transplant patients. *AMIA*.

Hssina, B., Merbouha, A., Ezzikouri, H. et Erritali, M. (2014). A comparative study of decision tree ID3 and C4. 5. *Int. J. Adv. Comput. Sci. Appl, 4*(2).

Hurst, J.R., Donaldson, G.C., Perera, W.R., Wilkinson, T.M., Bilello, J.A., Hagan, G.W., Vessey, R.S. et Wedzicha, J.A. (2006). Use of plasma biomarkers at exacerbation of chronic obstructive pulmonary disease. *American Journal of Respiratory and Critical Care Medicine, 174*(8), 867-874.

Ingargiola, G. (1996). Building classification models: ID3 and C4. 5. *Disponível por WWW em: http://www. cis. temple. edu/~ ingargio/cis587/readings/id3-c45. html*.

Ingram, T. (2012) *Disease vs. Illness*. de http://www.bboyscience.com/disease-vs-illness/

Irani, K.B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning.

Ismail, M. et Ciesielski, V. (2003). An Empirical Investigation of the Impact of Discretization on Common Data Distributions. *HIS*, 692-701.

Jensen, M.H., Cichosz, S.L., Hejlesen, O.K., Toft, E., Nielsen, C., Grann, O. et Dinesen, B.I. (2012). Clinical impact of home telemonitoring on patients with chronic obstructive pulmonary disease. *Telemedicine and e-Health, 18*(9), 674-678.

Juvelekian, G. (2012). Chronic Obstructive Pulmonary Disease. *Cleveland Clinic Journal of Medicine*. Récupéré de http://www.clevelandclinicmeded.com/medicalpubs/diseasemanagement/pulmonary/chronic-obstructive-pulmonary-disease/

Kabir, M.A.H., Jun; Colman, Alan;. (2016). Pervasive Social Computing: Socially-Aware Pervasive Systems and Mobile Applications. [ebook]. *Springer*.

Kamberov, R. (2016). Using social paradigms in smart cities mobile context-aware computing. *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*, 1-5.

Karagiannopoulos, M., Anyfantis, D., Kotsiantis, S. et Pintelas, P. (2007). Feature selection for regression problems. *Proceedings of the 8th Hellenic European Research on Computer Mathematics & its Applications, Athens, Greece, 2022*.

Karegowda, A.G., Jayaram, M. et Manjunath, A. (2010). Feature subset selection problem using wrapper approach in supervised learning. *International journal of Computer applications, 1*(7), 13-17.

Khattak, A.M., Akbar, N., Aazam, M., Ali, T., Khan, A.M., Jeon, S., Hwang, M. et Lee, S. (2014). Context representation and fusion: advancements and opportunities. *Sensors, 14*(6), 9628-9668.

Kirsch Pinheiro, M. (2006). *Adaptation contextuelle et personnalisée de l'information de conscience de groupe au sein des Systèmes d'Information coopératifs.* Grenoble 1.

Kohavi, R. (1995a). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Ijcai, 14*(2), 1137-1145.

Kohavi, R. (1995b). *Wrappers for performance enhancement and oblivious decision graphs.* Citeseer.

Kononenko, I. (1995). On biases in estimating multi-valued attributes. *Ijcai, 95*, 1034-1040.

Kononenko, I. (2001). Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine, 23*(1), 89-109.

Kotsiantis, S. et Kanellopoulos, D. (2006). Discretization techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering, 32*(1), 47-58.

Kumari, A. (2014). Study on Naive Bayesian Classifier and its relation to Information Gain. *international Journal on Recent and Innovation Trends in Computing and Communication, 2*(3), 601 – 603.

Lareau, S., Moseson, E. et Slatore, C.G. (2014). Patient information series. *American journal of respiratory and critical care medicine, 189*(6).

Lerner, B. et Malka, R. (2011). Investigation of the K2 algorithm in learning Bayesian network classifiers. *Applied Artificial Intelligence, 25*(1), 74-96.

Li, X., Eckert, M., Martinez, J.-F. et Rubio, G. (2015). Context Aware Middleware Architectures: Survey and Challenges. *Sensors, 15*(8), 20570-20607.

Liu, W., Li, X. et Huang, D. (2011). A survey on context awareness. *Computer Science and Service System (CSSS), 2011 International Conference on*, 144-147.

Lustgarten, J.L., Gopalakrishnan, V., Grover, H. et Visweswaran, S. (2008). Improving classification performance with discretization on biomedical datasets. *AMIA annual symposium proceedings, 2008*, 445.

Maiolo, C., Mohamed, E.I., Fiorani, C.M. et De Lorenzo, A. (2003). Home telemonitoring for patients with severe respiratory illness: the Italian experience. *Journal of Telemedicine and Telecare, 9*(2), 67-71.

Mannino, D.M. et Buist, A.S. (2007). Global burden of COPD: risk factors, prevalence, and future trends. *The Lancet, 370*(9589), 765-773.

Mansoor, W., Khedr, M., Benslimane, D., Maamar, Z., Hauswirth, M., Aberer, K., Chaari, T., Laforest, F. et Celentano, A. (2008). Adaptation in context-aware pervasive information systems: the SECAS project. *International Journal of Pervasive Computing and Communications, 3*(4), 400-425.

Manual, N.-J. (2012). *Version 4.18 and Higher, Norsys Software Corp.*

McGeachie, M.J., Chang, H.-H. et Weiss, S.T. (2014). CGBayesNets: conditional Gaussian Bayesian Network learning and inference with mixed discrete and continuous data. *PLoS Comput Biol, 10*(6), e1003676.

Mcheick, H., Khreiss, M., Sweidan, H. et Zaarour, I. (2015). PHEN: Parkinson Helper Emergency Notification System Using Bayesian Belief Network. Dans *E-Technologies* (p. 212-223) : Springer.

McKinstry, B., Pinnock, H. et Sheikh, A. (2009). Telemedicine for management of patients with COPD? *The Lancet, 374*(9691), 672-673.

McLean, S., Nurmatov, U., Liu, J.L., Pagliari, C., Car, J. et Sheikh, A. (2012). Telehealthcare for chronic obstructive pulmonary disease: Cochrane Review and meta-analysis. *Br J Gen Pract, 62*(604), e739-e749.

Megari, K. (2013). Quality of life in chronic disease patients. *Health Psychology Research, 1*(3).

Montserrat-Capdevila, J., Godoy, P., Marsal, J., Barbé, F. et Galvan, L. (2016). Risk factors for exacerbation in chronic obstructive pulmonary disease: a prospective study. *The International Journal of Tuberculosis and Lung Disease, 20*(3), 389-395.

Moore, P., Hu, B., Zhu, X., Campbell, W. et Ratcliffe, M. (2007). A survey of context modeling for pervasive cooperative learning. *Proceedings 1st International Symposium on Information Technologies and Applications in Education (ISITAE 2007), pp. K51–K56.*

Mostefaoui, G.K., Pasquier-Rocha, J. et Brezillon, P. (2004). Context-aware computing: a guide for the pervasive computing community. *Pervasive Services, 2004. ICPS 2004. IEEE/ACS International Conference on*, 39-48.

Murphy, K.P. (2012). *Machine learning: a probabilistic perspective.* : MIT press.

Murray, C.J. et Lopez, A.D. (1997). Alternative projections of mortality and disability by cause 1990–2020: Global Burden of Disease Study. *The Lancet, 349*(9064), 1498-1504.

Naïm, P., Wuillemin, P.-H., Leray, P., Pourret, O. et Becker, A. (1999). Les réseaux bayésiens. *Paris: Eyrolles*.

Najar, S. (2014). Adaptation des services sensibles au contexte selon une approche intentionnelle.

Najar, S., Saidani, O., Kirsch-Pinheiro, M., Souveyet, C. et Nurcan, S. (2009). Semantic representation of context models: a framework for analyzing and understanding. *Proceedings of the 1st Workshop on Context, Information and Ontologies*, 6.

Netica. (2000). Application for Belief Networks and influence Diagrams. Récupéré de http://www.norsys.com.

Nizet, T.A., van den Elshout, F.J., Heijdra, Y.F., van de Ven, M.J., Mulder, P.G. et Folgering, H.T.M. (2005). Survival of chronic hypercapnic COPD patients is

predicted by smoking habits, comorbidity, and hypoxemia. *CHEST Journal, 127*(6), 1904-1910.

O'Donnell, D.E., Aaron, S., Bourbeau, J., Hernandez, P., Marciniuk, D., Balter, M., Ford, G., Gervais, A., Goldstein, R. et Hodder, R. (2004). State of the art compendium: Canadian Thoracic Society recommendations for management of chronic obstructive pulmonary disease. *Canadian respiratory journal, 11*(Suppl B), 7B-59B.

Olivier, F. (2006). *De l'identification de structure de réseaux bayésiensa la reconnaissance de formesa partir d'informations completes ou incompletes.* INSA de Rouen.

Paliwal, R. (2012). Can optimal use of spirometry have a positive impact on the progression of chronic obstructive pulmonary disease? *Lung India: official organ of Indian Chest Society, 29*(1), 4.

Panthong, R. et Srivihok, A. (2015). Wrapper Feature Subset Selection for Dimension Reduction Based on Ensemble Learning Algorithm. *Procedia Computer Science, 72*, 162-169.

Parsons, S. et Kubat, M. (1994). A first-order logic for reasoning under uncertainty using rough sets. *Journal of Intelligent Manufacturing, 5*(4), 211-223.

Pascoe, J. (1998). Adding generic contextual capabilities to wearable computers. *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*, 92-99.

Polisena, J., Tran, K., Cimon, K., Hutton, B., McGill, S., Palmer, K. et Scott, R.E. (2010). Home telehealth for chronic obstructive pulmonary disease: a systematic review and meta-analysis. *Journal of telemedicine and telecare, 16*(3), 120-127.

Porkodi, R. (2014). comparison of filter based feature selection algorithms: An overview. *international journal of innovative research in technology& science, 2*(2), 108-113.

Priyadarsini, R.P., Valarmathi, M. et Sivakumari, S. (2011). Gain ratio based feature selection method for privacy preservation. *ICTACT Journal on soft computing, 1*(04), 20011.

Québec, L.a.p. (2014) *C'est quoi, la MPOC?* de http://sct.poumon.ca/diseases-maladies/copd-mpoc/what-quoi/index_f.php

Québec., A.p.d. (2016a). MPOC, bronchite et emphysème. Récupéré de http://www.pq.poumon.ca/diseases-maladies/copd-mpoc/

Québec., A.p.d. (2016b) *Pneumonie.* de http://www.pq.poumon.ca/diseases-maladies/pneumonia-pneumonie/

Quinlan, J.R. (1986). Induction of decision trees. *Machine learning, 1*(1), 81-106.

Quinlan, J.R. (1993). C4. 5: Programming for machine learning. *Morgan Kauffmann*, 38.

Quinlan, J.R. (1996). Improved use of continuous attributes in C4. 5. *Journal of artificial intelligence research, 4*, 77-90.

Raghavan, N., Lam, Y.-M., Webb, K.A., Guenette, J.A., Amornputtisathaporn, N., Raghavan, R., Tan, W.C., Bourbeau, J. et O'Donnell, D.E. (2012). Components of the COPD Assessment Test (CAT) associated with a diagnosis of COPD in a random population sample. *COPD: Journal of Chronic Obstructive Pulmonary Disease, 9*(2), 175-183.

Rajasekaran, S. (2015) *Database about COPD exacerbation.* de https://github.com/sibrajas/data-python/blob/master/CAX_COPD_TRAIN_data.csv

Rakotomalala, R. (2005). Arbres de décision. *Revue Modulad, 33*, 163-187.

Reynolds, G.M., Peet, A.C. et Arvanitis, T.N. (2007). Generating prior probabilities for classifiers of brain tumours using belief networks. *BMC medical informatics and decision making, 7*(1), 1.

Ricco, R. (Mai 2010). *Tanagra Discretization for Supervised Learning*. Récupéré de http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/en_Tanagra_Discretization_for_Supervised_Learning.pdf

Rish, I. (2001). An empirical study of the naive Bayes classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence, 3*(22), 41-46.

Robinson, R.W. (1977). Counting unlabeled acyclic digraphs. Dans *Combinatorial mathematics V* (p. 28-43) : Springer.

Rodriguez-Roisin, R. (2000a). Toward a consensus definition for COPD exacerbations. *CHEST Journal, 117*(5_suppl_2), 398S-401S.

Rodriguez-Roisin, R. (2000b). Toward a consensus definition for copd exacerbations*. *CHEST Journal, 117*(5_suppl_2), 398S-401S.

Ryan, N., Pascoe, J. et Morse, D. (1999). Enhanced reality fieldwork: the context aware archaeological assistant. *Bar International Series, 750*, 269-274.

Ryynänen, O.-P., Soini, E.J., Lindqvist, A., Kilpeläinen, M. et Laitinen, T. (2013). Bayesian predictors of very poor health related quality of life and mortality in patients with COPD. *BMC medical informatics and decision making, 13*(1), 1.

Salzberg, S.L. (1994). C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning, 16*(3), 235-240.

Sandberg, D. et Fleetham, J. (2013). Home oxygen therapy in British Columbia. *BC Med J, 55*(3), 149-152.

Sandelowsky, H., Ställberg, B., Nager, A. et Hasselström, J. (2011). The prevalence of undiagnosed chronic obstructive pulmonary disease in a primary care population with respiratory tract infections-a case finding study. *BMC family practice, 12*(1), 122.

Sanders, D.L. et Aronsky, D. (2006a). Detecting asthma exacerbations in a pediatric emergency department using a Bayesian network. *AMIA Annual Symposium Proceedings, 2006*, 684.

Sanders, D.L. et Aronsky, D. (2006b). Detecting asthma exacerbations in a pediatric emergency department using a Bayesian network. *AMIA*.

Schilit, B., Adams, N. et Want, R. (1994). Context-aware computing applications. *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, 85-90.

Schilit, B.N. et Theimer, M.M. (1994). Disseminating active map information to mobile hosts. *IEEE network, 8*(5), 22-32.

Seemungal, T.a., Donaldson, G.C., BHOWMIK, A., JEFFRIES, D.J. et WEDZICHA, J.A. (2000). Time course and recovery of exacerbations in patients with chronic obstructive pulmonary disease. *American journal of respiratory and critical care medicine, 161*(5), 1608-1613.

Seemungal, T.A., Donaldson, G.C., Paul, E.A., Bestall, J.C., Jeffries, D.J. et Wedzicha, J.A. (1998). Effect of exacerbation on quality of life in patients with chronic obstructive pulmonary disease. *American journal of respiratory and critical care medicine, 157*(5), 1418-1422.

Shachter, R.D. (1998). Bayes-ball: Rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, 480-487.

Shannon, C.E. (1948). A mathematical theory of communication. *Reprinted with corrections from The Bell System Technical Journal, 27*, 379-423.

Sieben, W. et Gather, U. (2007). Classifying alarms in intensive care-analogy to hypothesis testing. *Conference on Artificial Intelligence in Medicine in Europe*, 130-138.

Simões, P.W., Silva, G.D.d., Moretti, G.P., Simon, C.S., Winnikow, E.P., Nassar, S.M., Medeiros, L.R. et Rosa, M.I. (2015). Metanálise do uso de redes bayesianas no diagnóstico de câncer de mama. *Cadernos de Saúde Pública, 31*(1), 26-38.

Sinoquet, C. et Mourad, R. (2014). *Probabilistic Graphical Models for Genetics, Genomics, and Postgenomics*. : OUP Oxford.

Slimani, Y., Essegir, M.A., Samb, M.L., Camara, F. et Ndiaye, S. (2014). Approche de sélection d'attributs pour la classification basée sur l'algorithme RFE-SVM. *Revue Africaine de la Recherche en Informatique et Mathématiques Appliquées, 17*, 197-219.

Stéphane, T. (2012). *Data Mining et statistique décisionnelle: L'intelligence des données*. : Éditions Technip.

Strang, T. et Linnhoff-Popien, C. (2004). A context modeling survey. *Workshop Proceedings*.

Studer, R., Benjamins, V.R. et Fensel, D. (1998). Knowledge engineering: principles and methods. *Data & knowledge engineering, 25*(1), 161-197.

Tantucci, C., Donati, P., Nicosia, F., Bertella, E., Redolfi, S., De Vecchi, M., Corda, L., Grassi, V. et Zulli, R. (2008). Inspiratory capacity predicts mortality in patients with chronic obstructive pulmonary disease. *Respiratory medicine, 102*(4), 613-619.

Thoracologie, S.C.D. (Février 2010) *Le fardeau humain et financier de la MPOC - Une des principales causes d'hospitalisation au Canada.* de http://www.lignesdirectricesrespiratoires.ca/sites/all/files/MPOC_report.pdf

Ting, K.M. (1995). Common Issues in Instance-Based and Naive Bayesian Classifiers. *PhD Thesis, Basser Department of Computer Science, University of Sydney.*

Topcu, F. (2011). Context Modeling and Reasoning Techniques. *SNET Seminar in the ST.*

Trappenburg, J.C., Niesink, A., de Weert-van Oene, G.H., van der Zeijden, H., van Snippenburg, R., Peters, A., Lammers, J.-W.J. et Schrijvers, A.J. (2008). Effects of telemonitoring in patients with chronic obstructive pulmonary disease. *Telemedicine and e-Health, 14*(2), 138-146.

Tryo.labs. (2013) *Why accuracy alone is a bad measure for classification tasks, and what we can do about it.* de https://tryolabs.com/blog/2013/03/25/why-accuracy-alone-bad-measure-classification-tasks-and-what-we-can-do-about-it/

Tsui, F.-C., Espino, J.U., Dato, V.M., Gesteland, P.H., Hutman, J. et Wagner, M.M. (2003). Technical description of RODS: a real-time public health surveillance system. *Journal of the American Medical Informatics Association, 10*(5), 399-408.

van-Mölken, M.P.R. et Feenstra, T.L. (2001). The burden of asthma and chronic obstructive pulmonary disease. *Pharmacoeconomics, 19*(2), 1-6.

Van den Berge, M., Hop, W.C., van der Molen, T., van Noord, J.A., Creemers, J.P., Schreurs, A.J., Wouters, E.F. et Postma, D.S. (2012). Prediction and course of symptoms and lung function around an exacerbation in chronic obstructive pulmonary disease. *Respiratory research, 13*(1), 1.

Van der Heijden, M., Lucas, P.J., Lijnse, B., Heijdra, Y.F. et Schermer, T.R. (2013). An autonomous mobile system for the management of COPD. *Journal of biomedical informatics, 46*(3), 458-469.

Van der Heijden, M., Velikova, M. et Lucas, P.J. (2014). Learning Bayesian networks for clinical time series analysis. *Journal of biomedical informatics, 48*, 94-105.

Van der Molen, T., Willemse, B.W., Schokker, S., Ten Hacken, N.H., Postma, D.S. et Juniper, E.F. (2003). Development, validity and responsiveness of the Clinical COPD Questionnaire. *Health and quality of life outcomes, 1*(1), 1.

Van Nguyen, T., Lim, W., Nguyen, H., Choi, D. et Lee, C. (2010). Context ontology implementation for smart home. *arXiv preprint arXiv:1007.1273*.

Verduijn, M., Rosseel, P.M., Peek, N., de Jonge, E. et de Mol, B.A. (2007). Prognostic bayesian networks: II: An application in the domain of cardiac surgery. *Journal of biomedical informatics, 40*(6), 619-630.

Viegi, G., Pistelli, F., Sherrill, D.L., Maio, S., Baldacci, S. et Carrozzi, L. (2007). Definition, epidemiology and natural history of COPD. *European Respiratory Journal, 30*(5), 993-1013.

Vontetsianos, T., Giovas, P., Katsaras, T., Rigopoulou, A., Mpirmpa, G., Giaboudakis, P., Koyrelea, S., Kontopyrgias, G. et Tsoulkas, B. (2005). Telemedicine-assisted home support for patients with advanced chronic obstructive pulmonary disease: preliminary results after nine-month follow-up. *Journal of telemedicine and telecare, 11*(suppl 1), 86-88.

Wang, J. et Valtorta, M. (2012). Using Relative Classification Probability to Increase Accuracy of Restricted Structure Bayesian Network Classifiers. *2012 IEEE 24th International Conference on Tools with Artificial Intelligence, 1*, 105-113.

Want, R., Hopper, A., Falcao, V. et Gibbons, J. (1992). The active badge location system. *ACM Transactions on Information Systems (TOIS), 10*(1), 91-102.

Weka. (2011). Data Mining Software in Java. Récupéré de http://www.cs.waikato.ac.nz/ml/weka/index.html

Weka.net. Class NaiveBayes. Récupéré de http://weka.sourceforge.net/doc.dev/weka/classifiers/bayes/NaiveBayes.html

Wilkinson, T.M., Donaldson, G.C., Hurst, J.R., Seemungal, T.A. et Wedzicha, J.A. (2004). Early therapy improves outcomes of exacerbations of chronic obstructive pulmonary disease. *American journal of respiratory and critical care medicine, 169*(12), 1298-1303.

Witten, I.H. et Frank, E. (2011). *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition*. (Third Edition éd.).

Wolpert, D.H. (1996). The lack of a priori distinctions between learning algorithms. *Neural computation, 8*(7), 1341-1390.

Yañez, A.M., Guerrero, D., de Alejo, R.P., Garcia-Rio, F., Alvarez-Sala, J.L., Calle-Rubio, M., de Molina, R.M., Falcones, M.V., Ussetti, P. et Sauleda, J. (2012). Monitoring breathing rate at home allows early identification of COPD exacerbations. *CHEST Journal, 142*(6), 1524-1529.

Yang, Y. et Webb, G.I. (2002). A comparative study of discretization methods for naive-bayes classifiers. *Proceedings of PKAW, 2002*.

Yildirim, P. (2015). Filter based feature selection methods for prediction of risks in hepatitis disease. *International Journal of Machine Learning and Computing, 5*(4), 258.

Zhang, H. (2004). The optimality of naive Bayes. *AA, 1*(2), 3.