Titre: Title:	Intégration des approches statistiques et d'apprentissage machine dans l'étude des maladies neurodégénératives
Auteur: Author:	Thomas Ricard
Date:	2024
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Ricard, T. (2024). Intégration des approches statistiques et d'apprentissage machine dans l'étude des maladies neurodégénératives [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/61587/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/61587/
Directeurs de recherche: Advisors:	Richard Labib
Programme: Program:	Maîtrise recherche en mathématiques appliquées

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Intégration des approches statistiques et d'apprentissage machine dans l'étude des maladies neurodégénératives

THOMAS RICARD
Département de mathématiques et de génie industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Mathématiques appliquées

Décembre 2024

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Intégration des approches statistiques et d'apprentissage machine dans l'étude des maladies neurodégénératives

présenté par **Thomas RICARD**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Nadia LAHRICHI, présidente
Richard LABIB, membre et directeur de recherche
Luc **ADJENGUE,** membre

REMERCIEMENTS

Mes remerciements vont à mon directeur de recherche, Monsieur Richard Labib, pour m'avoir offert ce projet de maîtrise, ainsi que pour sa patience, son expertise et son aide tout au long de la rédaction de ce mémoire.

Je tiens également à remercier nos collaborateurs pour la confiance qu'ils nous ont accordée avec ces projets.

Enfin, j'aimerais remercier mes amis et ma famille pour leur soutien tout au long de mon parcours.

RÉSUMÉ

De nos jours, les études cliniques sur les maladies neurodégénératives génèrent des bases de données de plus en plus grandes et complexes à analyser pour atteindre leurs objectifs de recherche. L'identification de biomarqueurs potentiels et la découverte de nouveaux traitements exigent de nouvelles approches mathématiques, en raison de la complexité biologique de ces maladies. Dans le cadre de notre recherche, nous nous intéressons principalement à deux études cliniques portant sur la maladie de Parkinson et sur une autre maladie neurologique de type génétique.

Notre premier projet a pour objectif de valider l'existence de biomarqueurs de la maladie dans les protéines des vésicules extracellulaires dérivées des érythrocytes. Les données à notre disposition sont cependant très complexes, corrélées et incomplètes, avec un nombre de variables bien plus élevé que le nombre de patients impliqués. L'approche par apprentissage machine s'avère alors adéquate. Plus précisément, nous avons optimisé les différentes étapes de l'approche systématique, à savoir la mise à l'échelle, l'imputation, la sélection de variables, l'échantillonnage et la prédiction. Notre méthodologie consiste à classifier les patients parkinsoniens et les patients contrôles, puis à interpréter les prédictions des modèles pour identifier des biomarqueurs potentiels. Pour ce faire, quatre méthodes originales ont été testées dans l'approche systématique : une méthode d'imputation flexible, de sous-échantillonnage par prototypes, de sous-échantillonnage par erreurs de reconstruction et un classificateur utilisant les interactions protéine-protéine. Nous observons, de manière générale, que les performances de classification sur l'ensemble de données demeurent limitées, ce qui suggère une possible absence de biomarqueurs dans celui-ci. Toutefois, nos méthodes originales permettent d'améliorer les performances ou potentiellement l'interprétabilité des modèles.

Notre deuxième projet a pour objectif d'évaluer l'efficacité d'une nouvelle molécule dans le contexte d'une maladie neurologique de type génétique. Pour ce faire, nous analysons des données cliniques recueillies sur 54 mois, avec un groupe placebo différé qui débute le traitement au 18e mois. Notre approche consiste à utiliser des modèles mixtes pour mesures répétées afin de préserver la dépendance temporelle des données pour chaque patient. Ces modèles sont généralement analysés en trois parties distinctes, ce qui complexifie la tâche. Nous avons donc implémenté une approche simplifiée permettant de confirmer l'efficacité du traitement dans un seul modèle, en se fiant à une variable mesurant le temps de traitement chez les patients. De manière générale, l'analyse montre que le traitement n'a pas d'effica-

cité significative sur l'ensemble de la cohorte. Toutefois, en dichotomisant les patients selon certaines caractéristiques, nous avons obtenu des sous-groupes pour lesquels le traitement semble avoir un effet significatif sur leur état de santé. Ces résultats encourageants issus de notre analyse pourraient conduire au lancement de nouvelles études plus approfondies sur l'efficacité de ce traitement.

ABSTRACT

Nowadays, clinical studies on neurodegenerative diseases generate increasingly large and complex datasets, making them challenging to analyze in order to meet research objectives. Identifying potential biomarkers or discovering new treatments requires new mathematical approaches due to the biological complexity of these diseases. In our research, we are primarily focused on two clinical studies concerning Parkinson's disease and another genetically-based neurological disease.

Our first project on Parkinson's disease aims to validate the existence of disease biomarkers in proteins from erythrocyte-derived extracellular vesicles. However, the data at our disposal are very complex, correlated, and incomplete, with a much higher number of variables than the number of patients involved. Machine learning approaches therefore prove to be appropriate. Thus, a machine learning approach proves to be suitable. Specifically, we optimized various stages of a systematic approach, including scaling, imputation, feature selection, sampling, and prediction. Our methodology involves classifying Parkinsonian patients and control patients, followed by interpreting model predictions to identify potential biomarkers. To achieve this, we tested four novel methods within the systematic approach: a flexible imputation method, prototype-based undersampling, reconstruction error-based undersampling, and a classifier utilizing protein-protein interactions. Overall, we observe that classification performance on the dataset remains limited, suggesting a possible lack of biomarkers within it. However, our original methods allow for improvements in performance or potentially in the interpretability of the models.

Our second project aims to evaluate the efficacy of a new molecule in the context of a genetically-based neurological disease. To achieve this, we are analyzing clinical data collected over 54 months, with a delayed placebo group starting treatment at the 18th month. Our approach involves using mixed models for repeated measures to preserve the temporal dependency of the data for each patient. These models are typically analyzed in three distinct parts, which complicates the task. Therefore, we have implemented a simplified approach to confirm the treatment's efficacy in a single model, relying on a variable that measures the treatment duration in patients. In general, the analysis shows that the treatment does not have a significant effect on the entire cohort. However, by dichotomizing patients based on certain characteristics, we identified subgroups for which the treatment seems to have a significant effect on their health status. These encouraging results from our analysis could lead to the launch of more in-depth studies on the efficacy of this treatment.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

AUC	« Area under the curve »
ANOVA	Analyse de la variance
CNN	« Convolutional neural network »
EM	Espérance-maximisation
FN	« False negative »
FP	« False positive »
HSIC	« Hilbert-Schmidt Independence Criterion »
IMC	Indice de masse corporelle
KNN	« K-nearest neighbors »
Lasso	« Least Absolute Shrinkage and Selection Operator »
LC-MS	Chromatographie en phase liquide couplée à la spectrométrie de masse
LDA	« Linear discriminant analysis »
MAR	« Missing at random »
MNAR	« Missing not at random »
MMRM	Modèle mixte pour mesures répétées
mRMR	« Minimum Redundancy Maximum Relevance »
MVS	Machine à vecteurs de support
PCA	« Principal component analysis »
PLS	« Partial least squares »
PPI	Interactions protéine-protéine
PR-AUC	« Precision-Recall area under the curve »
QRILC	« Quantile Regression Imputation of Left-Censored data »
RNA	Réseau de neurones artificiels
SMOTE	« Synthetic Minority Over-sampling Technique »
TN	« True negative »
TP	« True positive »
ODCS	« Overall disease clinical scale »
VAE	« Variational autoencoder »
VEE	Vésicules extracellulaires dérivées des érythrocytes

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

1.1 Problématique des maladies neurodégénératives

Les maladies neurodégénératives sont des maladies principalement présentes chez les personnes âgées, qui se caractérisent généralement par des pertes neuronales et des connexions synaptiques altérées. En particulier, cela se traduit par des tremblements et des raideurs musculaires pour la maladie de Parkinson [1]. Cette maladie nous intéresse particulièrement, puisqu'elle sera au cœur de notre étude. Nous allons également considérer une autre maladie neurologique de type génétique, dont les détails ne peuvent être divulgués en raison d'un accord de confidentialité.

La croissance de la population est un facteur parmi tant d'autres ayant contribué à l'augmentation du nombre de personnes affectées par des maladies neurodégénératives ces dernières années [2]. En 2018, environ 15 % de la population mondiale était affectée par des troubles neurologiques de tout genre, et on s'attend à ce que le nombre de patients atteints de maladies neurodégénératives chroniques double d'ici deux décennies [3]. Selon la *Parkinson Foundation*, plus de 10 millions de personnes seraient atteintes de cette maladie à travers le monde [4]. La gestion des patients atteints de celle-ci engendre également d'énormes coûts pour la société. Aux États-Unis, chaque année, la maladie de Parkinson coûte environ 52 milliards de dollars américains, la maladie d'Alzheimer 360 milliards de dollars américains, et le coût de la démence devrait atteindre 2 000 milliards de dollars d'ici 2030 [2, 5].

Le coût, tant économique que psychologique, rattaché aux maladies neurodégénératives a déclenché une mobilisation de la recherche, incitant un grand nombre de scientifiques à s'attaquer à celles-ci pour améliorer la vie des patients. Parmi l'ensemble des travaux de recherche, deux axes principaux se distinguent : le premier touche au diagnostic précoce des maladies à travers la recherche de biomarqueurs, et le second porte sur l'évaluation de l'efficacité de traitements spécifiques aux maladies neurodégénératives. Les biomarqueurs sont des indicateurs biologiques mesurables qui permettent de diagnostiquer la maladie, c'est-à-dire d'identifier les patients atteints, ou de suivre l'évolution de leur état [6]. Les études portant sur les traitements visent à évaluer l'impact d'une médication sur un indice clinique au fil du temps, reflétant la progression de la maladie chez les patients. Dans le cadre de notre étude, nous explorons l'existence d'un biomarqueur dans le sang pour la maladie de Parkinson et tentons de valider l'efficacité d'un nouveau traitement pour une maladie neurologique de type génétique.

1.2 Problématiques et objectifs de recherche

Dans les dernières décennies, les chercheurs ont remarqué que les maladies neurodégénératives étaient souvent liées à des protéines ayant des propriétés physico-chimiques altérées et, avec l'avènement de nouvelles technologies, on voit de plus en plus d'études consacrées sur des données protéomiques [7, 8]. Toutefois, ces maladies se distinguent par une grande hétérogénéité et une complexité accrues, résultant en des interactions entre facteurs génétiques, environnementaux et du mode de vie des patients, ce qui rend difficile l'identification de biomarqueurs [9]. Nous avons cependant eu l'opportunité de travailler en collaboration avec des chercheurs d'une université canadienne, qui nous ont donné accès à des données d'études cliniques. En effet, ils soupçonnent, d'après leurs études, qu'il pourrait exister des biomarqueurs dans le sang des patients atteints de la maladie de Parkinson. Plus précisément, ils envisagent la possibilité de découvrir des biomarqueurs au sein des protéines des vésicules extracellulaires dérivées des érythrocytes (VEE), également appelées globules rouges. L'objectif du premier projet est donc de valider l'existence d'un biomarqueur de la maladie de Parkinson dans le protéome des VEE, ce qui permettrait de distinguer des patients sains des patients parkinsoniens avant l'apparition de symptômes.

Dans le cadre du deuxième projet, des collaborateurs s'intéressent à une nouvelle molécule pour traiter les symptômes d'une autre maladie neurologique de type génétique. Pour des raisons de confidentialité, nous garderons anonymes l'identité des chercheurs, le nom de la molécule et celui de la maladie. Certains termes cliniques associés à celle-ci seront aussi remplacés par des termes plus génériques. L'objectif de ce deuxième projet est d'évaluer l'efficacité de ce nouveau traitement à base d'une molécule, que nous appellerons ici molécule X, chez une cohorte de patients atteints d'une maladie neurologique de type génétique.

La recherche de biomarqueurs et la vérification de l'efficacité d'un traitement ne sont pas des tâches simples. En effet, dans le cadre du premier projet, nous avons accès à des données protéomiques composées d'un très grand nombre de variables en comparaison au nombre de patients où la majorité de ces protéines n'ont aucun lien direct avec la maladie de Parkinson. Les méthodes statistiques classiques ne conviennent donc pas à ce type de problème. Il s'avère qu'une approche par apprentissage machine est mieux adaptée à la complexité des données, c'est pourquoi nous explorons cette avenue dans notre recherche.

Pour ce qui est de la vérification du traitement pour la maladie neurologique de type génétique, nous analysons différents scores cliniques au fil du temps (études longitudinales). Il est donc crucial de tenir compte de la dépendance des données. En effet, les mesures prises pour un même patient peuvent être affectées par des caractéristiques propres à cet individu. Les

modèles statistiques de régression à effets aléatoires et fixes se prêtent donc bien à ce genre d'étude, et seront utilisés pour le deuxième projet. La figure 1.1 présente un résumé de notre étude avec les objectifs, les méthodes et les contributions originales pour les deux projets.

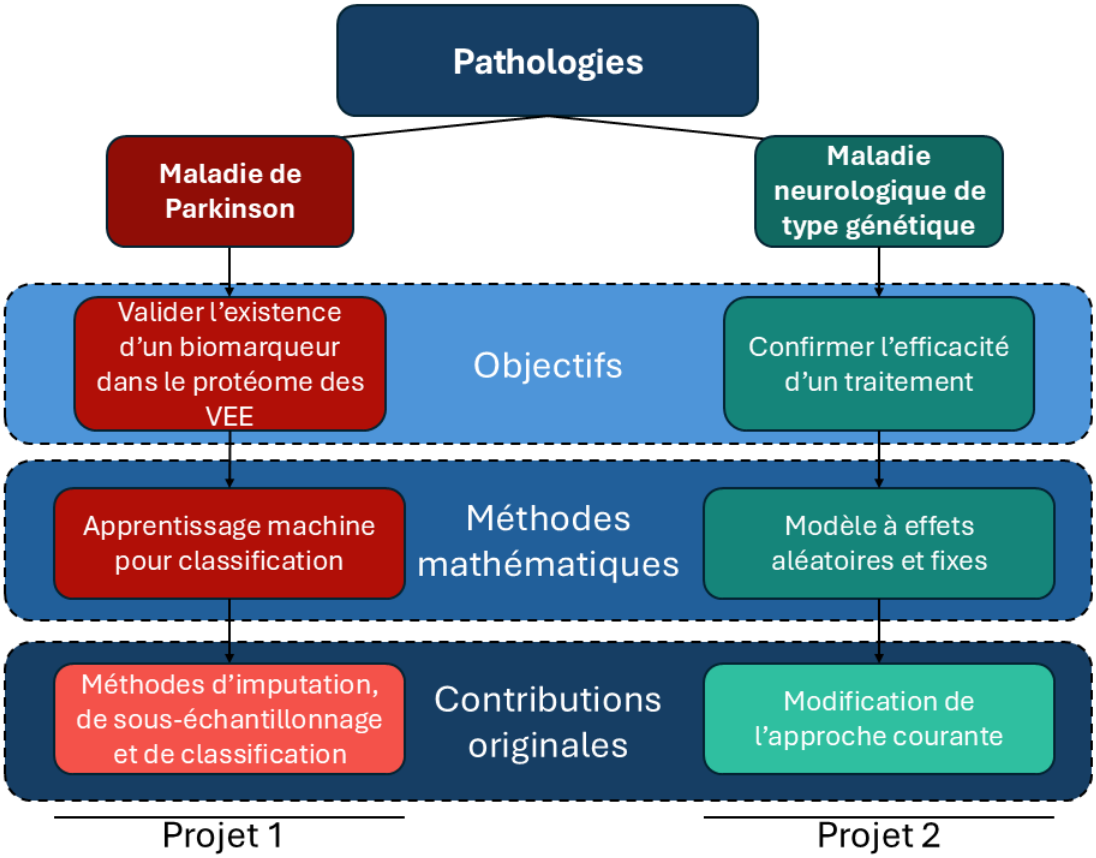

FIGURE 1.1 Diagramme des projets, objectifs, méthodologies et méthodes originales.

1.3 Plan du mémoire

Dans un premier temps, le chapitre 2 portera sur la revue de littérature des méthodes mathématiques utilisées pour aborder des problèmes semblables à nos projets. Nous explorons donc à la fois les méthodes d'apprentissage machine pour cibler des biomarqueurs potentiels et les méthodes statistiques pour valider des traitements d'études médicales. La revue ne se limite pas à la maladie de Parkinson, mais s'étend à toutes les maladies neurodégénératives, puisque l'on peut s'attendre à ce que les processus biologiques concernés se ressemblent, et que les méthodes mathématiques utilisées s'adaptent bien à nos maladies.

Le chapitre 3 traitera du premier projet, portant sur la maladie de Parkinson. Dans cette étude, nous utilisons une approche systématique d'apprentissage machine visant à identifier

des biomarqueurs en classifiant les patients contrôles par rapport à ceux atteints de la maladie. Cette méthode se décompose en différentes étapes de l'apprentissage machine, telles que le prétraitement, le remplacement de données manquantes (imputation), la normalisation, la sélection de variables pertinentes, le rééquilibrage des classes, l'augmentation de données, le sous-échantillonnage, la classification et la validation. Nous verrons notamment que la partie originale de notre recherche porte sur des modifications des méthodes d'imputation, de sous-échantillonnage et de classification pour mieux s'adapter à notre problème.

Le deuxième projet, présenté au chapitre 4, portera sur les analyses mathématiques en lien avec l'efficacité de la molécule X chez des patients souffrant d'une maladie neurologique génétique. Nous verrons comment évaluer l'efficacité d'un traitement dans des études à début différé, où un groupe reçoit d'abord un placebo, puis rejoint le groupe traité après une période déterminée. Plus précisément, nous proposons une approche novatrice pour analyser ce type de données en modifiant l'approche classique utilisant des modèles statistiques de régression avec effets aléatoires et fixes.

Finalement, le chapitre 5 présentera un récapitulatif des méthodes originales abordées, une discussion sur les limites des solutions actuelles et proposera des pistes d'amélioration pour les algorithmes à explorer dans les projets futurs.

CHAPITRE 2 REVUE DE LITTÉRATURE

Ce chapitre débute par la revue de littérature des méthodes d'apprentissage machine liées à la recherche de biomarqueurs de maladies neurodégénératives dans le cadre du projet 1. Elle est suivie par une revue de l'état de l'art des méthodes statistiques utilisées pour analyser des études longitudinales en médecine, en lien avec l'analyse de traitement dans le cadre du projet 2.

2.1 Projet Parkinson

Pour cette étude, nous avons accès à une base de données contenant des intensités relatives de plus de mille protéines dans les VEE, mesurées à l'aide d'une méthode de chromatographie en phase liquide couplée à la spectrométrie de masse (LC-MS). Cependant, cette base de données ne comprend que 355 patients, un nombre relativement faible en comparaison aux mille variables. De plus, les données sont très bruitées en raison des nombreuses manipulations en laboratoire et du manque de développement technique dans le domaine. Elles sont aussi incomplètes, avec une proportion importante de valeurs manquantes pour chaque patient. Ces trois problèmes sont donc les plus importants à résoudre pour identifier des biomarqueurs à travers nos données. Pour ce faire, nous avons dans un premier temps analysé les types d'études utilisant l'apprentissage machine pour identifier des biomarqueurs dans le cadre de diverses maladies neurodégénératives. En effet, étant donné que les processus biologiques sont susceptibles de se ressembler, nous ne nous sommes pas limités à la seule maladie de Parkinson. De plus, nous ne sommes pas non plus limités à des données protéomiques. Nous avons plutôt exploré l'ensemble des études utilisant l'apprentissage machine pour avoir une vue globale des méthodes à notre disposition. Cette section débute par une revue globale des méthodes utilisées dans le domaine, suivie d'études portant sur des données dites omiques, qui représentent l'analyse systématique des biomolécules d'un organisme. Cela inclut la génomique (gènes), la métabolomique (métabolites), la transcriptomique (acide ribonucléique messagers) et la protéomique (protéines). Ces types de données sont souvent très complexes à analyser en raison de leur grand nombre de variables, de l'abondance de valeurs manquantes et de la forte présence de bruit. C'est pourquoi la dernière section porte sur une revue des études portant sur des méthodes d'apprentissage machine adaptées à des données de grandes dimensions ayant peu d'échantillons, sans toutefois se limiter au domaine biologique.

2.1.1 Apprentissage machine pour les maladies neurodégénératives

L'apprentissage automatique a été utilisé dans la littérature pour les maladies neurodégénératives sur divers types de données et pour atteindre plusieurs objectifs. Cette sous-section examine ces différents types de données, leurs raisons d'utilisation, et offre un aperçu des méthodologies d'intelligence artificielle qui leur sont appliquées. Premièrement, un symptôme courant des maladies neurodégénératives est le fait d'avoir des problèmes moteurs de toutes sortes. Il y a alors de nombreuses études portant sur ce type de données que ce soit pour les tremblements causés par la maladie de Parkinson, la démarche des patients ou les problèmes de langage chez les patients atteints de démence. Certaines études portent par exemple sur l'analyse de la voix des patients. En analysant les enregistrements audio et en isolant des variables importantes comme le temps entre chaque mot, les chercheurs sont capables de créer un problème de classification où l'on cherche à dire si un patient est atteint de troubles cognitifs ou s'il est sain [10–13]. La majorité de ces articles utilisent des méthodes classiques comme la machine à vecteurs de support (MVS) ou des réseaux de neurones artificiels (RNA). Ces méthodes très générales s'adaptent bien aux problèmes de classification et sont alors utilisées dans notre recherche. Cependant, on peut voir des méthodes plus poussées comme montrent Vásquez-Correa et al. (2018) qui transforment les enregistrements audio en représentation temps-fréquence pour ensuite les passer aux travers d'un réseau neuronal convolutif (CNN de l'anglais « Convolutional neural network ») et faire la classification [14]. Ces réseaux sont spécialisés dans le traitement d'images grâce à leur architecture, qui leur permet d'apprendre des filtres via des convolutions, contrairement aux simples poids utilisés dans d'autres modèles. Ces modèles ne sont toutefois pas utiles dans notre cas puisque nos données ne possèdent aucune caractéristique spatiale ou temporelle.

De nombreux articles portent aussi sur l'analyse de données motrices provenant de mouvement des patients. Par exemple, certaines études analysent la démarche des patients à l'aide d'accéléromètres et extraient des caractéristiques des données, telles que les fluctuations entre leurs pas. Les auteurs utilisent encore une fois principalement la MVS pour tenter de classifier les patients atteints de la maladie de Parkinson [15] ou de la sclérose latérale amyotrophique [16]. Cependant, étant donné qu'il s'agit d'un signal temporel, certains chercheurs ont également recours à des méthodes plus complexes, telles que les réseaux de neurones récurrents qui utilise la propriété séquentielle du mouvement [17]. Cette méthode plus adaptée a permis aux auteurs de différencier différentes maladies neurodégénératives dans leur cohorte de patients avec une précision semblable à celle des autres articles sur une seule maladie. Les réseaux de neurones récurrents ne sont cependant pas applicables dans notre étude puisque nos données ne possèdent aucune caractéristique séquentielle. Néanmoins, l'article

souligne l'importance de comprendre la structure des données afin d'optimiser les algorithmes en fonction de celle-ci.

Encore du côté des données motrices, des articles portent sur l'écriture des patients. Certains chercheurs ressortent des variables importantes dans l'écriture des patients comme la pression, l'accélération et la vitesse, puis classifient les patients à l'aide de différents algorithmes de classification. Ils observent, en général, que la MVS reste le meilleur algorithme pour ce type de données [18, 19]. D'autres ont amélioré les méthodes de classification en utilisant de la sélection de variables à l'aide du test statistique de Mann-Whitney U qui teste de manière non-paramétrique la probabilité d'obtenir une valeur plus grande dans la distribution des patients contrôles que dans celle des patients malades [20]. On remarque alors que l'extraction et la sélection de variables sont des étapes importantes avant la classification, c'est pourquoi elles sont explorées dans notre étude. Par la suite, on voit encore une fois des chercheurs pousser plus loin en utilisant la structure des données, ici spatiale (image de l'écriture), pour appliquer un ensemble de CNN ce qui permet d'améliorer considérablement les résultats même sur des patients en phase initiale des maladies [21]. Pour terminer sur les données de type motrices, des chercheurs ont aussi tenté une approche sur de plus grandes populations pour être en mesure de réduire les problèmes de surapprentissage avec l'apprentissage profond. Pour ce faire, ils ont développé des applications mobiles permettant aux patients sains et malades de collecter des données en continu sur les gyroscopes et accéléromètres des téléphones cellulaires. Ces grandes quantités de données permettent ensuite d'utiliser des méthodes plus complexes comme les forêts aléatoires et, encore une fois, les CNN [22, 23]. Ces méthodes utilisant les téléphones cellulaires permettent aussi de faire des débuts de diagnostic à distance plus simplement en téléchargeant l'application et en gardant son téléphone cellulaire sur soi. Cependant, les données sont bien plus bruitées en raison du fait qu'elles ne proviennent pas de tests précis pour vérifier les symptômes de la maladie, mais de données récoltées par le téléphone cellulaire durant la journée ce qui réduit l'efficacité de prédiction des modèles en comparaison au reste de la littérature. Dans le contexte de notre étude, l'application des CNN ne s'avère pas pertinente. Toutefois, on voit bien la tendance des études à utiliser les propriétés intrinsèques de leurs données à leur avantage, une approche que nous explorons dans cette recherche.

Puisque les maladies neurodégénératives apportent généralement des pertes neuronales fortes, on voit aussi plusieurs recherches porter sur l'analyse de données obtenues par électroencéphalogramme. Cette méthode permet d'analyser, à l'aide d'électrodes, l'activité cérébrale des patients atteints d'Alzheimer, de la sclérose latérale amyotrophique ou d'Huntington. De nombreux articles portent sur l'analyse directe des données obtenues par une analyse spectrale des signaux temporels. La MVS reste l'algorithme le plus couramment utilisé sans

qu'une sélection de variables préalables ne soit effectuée [24–26]. Toutefois, certains chercheurs vont plus loin en démontrant l'importance de réduire le nombre de variables avant l'application du modèle. Par exemple, Dauwan et al. (2016) utilisent un algorithme de forêt aléatoire et Vanegas et al. (2018) utilisent l'ExtraTrees qui sont des algorithmes plus robustes par rapport aux variables inutiles grâce à l'utilisation d'un grand nombre d'arbres de décision avec une sélection aléatoire des variables [27, 28]. Ces algorithmes plus appropriés leur permettent d'atteindre des exactitudes de classification de plus de 90%. Les forêts aléatoires sont donc des algorithmes pertinents pour notre recherche puisqu'ils permettent de réduire l'impact des protéines inutiles dans nos données. D'autres chercheurs, comme McBride et al. (2015), utilisent des techniques de sélection de variables telles que l'analyse en composantes principales (PCA de l'anglais « Principal component analysis »), et Trambaiolli et al. (2017) analysent huit méthodes différentes de sélection de variables permettant d'améliorer l'exactitude des modèles de près de 20%, montrant encore une fois l'importance de cette étape que nous allons privilégier [29, 30].

Une autre grande catégorie de données utilisée en pair avec l'apprentissage machine est celle de l'imagerie en raison de sa complexité. Ces images proviennent de différents types de technologies comme l'imagerie par résonance magnétique, la tomographie par émission de positons, la tomographie par émission de photons simple ou l'imagerie par tenseur de diffusion. Cependant, la très grande majorité des études porte sur l'imagerie par résonance magnétique. Puisque ces données sont des images simples en 2D, il existe déjà de nombreuses méthodes d'apprentissage profond adaptées, comme les CNNs qui permettent de faire de la classification et de la régression. Malgré ce fait, on voit encore de nombreux articles ne pas prendre avantage de ces avancées dans le domaine et utiliser des méthodes de bases de classification. En effet, on voit par exemple des articles tenter de classifier des patients en santé contre des patients atteints de la maladie d'Huntington en utilisant la MVS, avec ou sans sélection de variables [31, 32]. De même pour la maladie de Parkinson et d'Alzheimer où plusieurs utilisent encore une fois la MVS accompagnée de différentes méthodes de sélection de variables comme la méthode par redondance minimale ou par l'autocorrélation [33–36]. D'autres articles tentent différentes méthodes comme la forêt aléatoire, la régression logistique, les RNA, la méthode de Bayes Naive ou la méthode des K plus proches voisins (KNN de l'anglais « K-nearest neighbors ») et obtiennent des résultats similaires [37–40]. Ce sont toutes des méthodes ne faisant pas ou peu d'hypothèses sur la forme des données et n'utilisent donc pas la propriété spatiale des images (certaines peuvent toutefois utiliser des noyaux pour faire des hypothèses comme des espaces gaussiens). Cependant, certaines de ces méthodes comme les RNA, les forêts aléatoires et la régression logistique peuvent être utilisées dans le cadre de notre recherche puisque nos données ne possèdent pas cette propriété spatiale. Toutefois,

la méthode de Bayes Naive et la méthode des KNN ne sont pas adaptées à nos données puisqu'elles ne performent pas face à des données de grande dimension avec peu d'échantillons. Plus récemment, on commence à voir des articles proposant des méthodes utilisant de l'apprentissage profond pour analyser ces images complexes. Certains articles explorent des architectures de CNN connue comme GoogleNet, ResNet ou LeNet et d'autres vont créer leur propre architecture souvent plus simple. Dans les deux cas, les résultats de classification sont impressionnants avec des exactitudes approchant souvent 99% ou plus [41–46].

Pour conclure cette sous-section, il est très difficile de comparer les résultats numériques des études entre elles. En effet, puisque les études utilisent souvent des banques de données différentes, il est possible que certaines soient plus simples à classifier que d'autres (par exemple, si les patients sont dans des stages plus avancés). Pour ce qui est des méthodes d'apprentissage machine, l'approche qui ressort le plus est celle de la classification des patients en santé contre les patients atteint de maladies neurodégénératives. Les méthodes les plus utilisées pour cette tâche semblent être la MVS suivie de la forêt aléatoire. Dans les dernières années, davantage de recherches se font à l'aide d'apprentissage profond, mais ce dernier est souvent restreint par la faible quantité de données apportant du surapprentissage. La sélection de variables semble aussi être une étape très importante au préalable de la classification pour améliorer les modèles, mais aussi afin de permettre une analyse plus simple des variables importantes pour tenter de trouver des biomarqueurs. Finalement, plusieurs études analysent des données qui nécessitent que le patient ait des symptômes de la maladie, comme les données motrices, ce qui ne permet pas de faire des diagnostics tôt dans le développement de la maladie comme avec les données omiques.

2.1.2 Méthodes appliquées aux données omiques

Dans ce projet, nous avons accès à une banque de données sur l'intensité de différentes protéines dans les EEV de patients contrôles et Parkinsoniens. Les données omiques sont alors celles qui ressemblent le plus aux nôtres (principalement les protéomiques). En effet, comme abordé précédemment, les données omiques proviennent de quatre catégories principales, soient la génomique, la transcriptomique, la métabolomique et la protéomique. Les deux premières sont prises principalement avec la méthode de *next-generation sequencing* qui est déjà bien explorée dans la littérature. Les données métabolomiques quant à elles proviennent de petites molécules apparaissant principalement à la fin de procédés biochimiques complexes. Les données sont extraites avec différentes méthodes comme la chromatographie en phase liquide, l'électrophorèse capillaire ou la chromatographie en phase gazeuse suivie d'une spectrométrie de masse. Finalement, la protéomique touche les protéines, qui sont de

plus grosses molécules bien plus complexes qui peuvent apparaître sous différentes formes et appartenir à différents gènes. Les méthodes de spectrométrie de masse sont à nouveau utilisées, toutefois l'extraction et l'identification des protéines constituent un domaine en pleine évolution, présentant de nombreux défis à surmonter, ce qui génère une quantité significative de bruit dans les données. [47, 48]. Les analyses métabolomiques ou protéomiques obtenues par LC-MS s'avèrent être les plus similaires aux données de ce projet.

Pour ce qui est des données génomiques et transcriptomiques qui viennent souvent en pair, de nombreux articles utilisent des algorithmes de classification similaires comme la MVS, la forêt aléatoire, les RNA et la régression logistique [49–51]. Les meilleurs algorithmes diffèrent pour chaque article en fonction de la banque de données. Cependant, la différence provient souvent de la méthode d'extraction et de sélection des variables. L'extraction des variables est souvent faite à l'aide de banques de données d'interactions gène-gène ou protéine-protéine [52, 53]. Cette méthode d'extraction de variables basée sur les interactions protéiques est très intéressante puisqu'elle permet d'utiliser les propriétés biologiques intrinsèques de nos données dans l'algorithme. Elle est donc explorée davantage dans notre recherche. Pour ce qui est de la sélection des variables, on peut voir des méthodes simples comme la régression *Least Absolute Shrinkage and Selection Operator* (Lasso), des auto-encodeurs variationnels adaptés pour capturer les variations dans les tissus ou une méthode permettant de garder le procédé biologique à travers les variables [54–56]. Lasso est un algorithme classique de sélection de variables qui est employé dans notre recherche, tandis que les deux autres méthodes reposent sur des concepts biologiques absents de nos données. Certains articles exploitent également les propriétés séquentielles des acides aminés formant les gènes pour utiliser des algorithmes appropriés. Par exemple, certains auteurs vont transformer les données pour les rendre adaptées à des CNNs, permettant ainsi d'extraire des dépendances séquentielles. D'autres chercheurs extraient davantage de variables en transformant les données en vecteurs de 400 dimensions (nombre maximal de combinaisons possibles d'acides aminés) [57, 58]. Cette idée est pertinente, mais elle n'est pas utilisée dans notre étude puisqu'il est bien plus complexe de retrouver les chaînes d'acides aminés des protéines que celles des gènes. Après avoir classifié les patients, la majorité des études portent sur les méthodes de sélection de variables pour identifier des biomarqueurs. On voit toutefois certains auteurs pousser plus loin en utilisant la méthode de « SHapley Additive exPlanations » qui permet d'expliquer plus intuitivement les variables importantes dans des modèles complexes comme les RNA ou les méthodes d'ensembles [54, 59]. Cet algorithme n'est cependant pas exploré dans notre étude puisqu'il est encore controversé avec différents chercheurs démontrant plusieurs défauts. Par exemple, l'hypothèse d'indépendance des variables est souvent fausse et la méthode possède des problèmes de stabilité face au bruit [60–63].

Analysons maintenant les données métabolomiques et protéomiques, qui sont celles représentant le mieux les données du projet, principalement lorsqu'elles proviennent de LC-MS. L'intelligence artificielle a déjà été utilisée dans de nombreuses parties de la chaîne de traitement des données, on voit par exemple Zhou et al. (2017) prédire les spectres à partir des séquences de peptides (éléments de base des protéines) avec l'outil *pDeep*. Demichev et al. (2020) augmentent la sensibilité de l'identification des peptides, avec le modèle *DIA-NN*, et Ma et al. (2018) prédisent les temps de rétention chromatographique [64–66]. Toutefois, les données auxquelles nous avons accès ont déjà passé ces étapes et sont maintenant sous forme d'intensités relatives de peptides obtenues à partir des pics de spectrométrie de masse, et ces intensités sont ensuite utilisées pour déterminer celles des protéines correspondantes. Elles ont aussi été normalisées à l'aide de l'algorithme *MaxLFQ* qui permet de quantifier les protéines en réduisant l'erreur systématique posée par les différentes prises de mesures durant les manipulations [67].

Ces données possèdent cependant beaucoup de valeurs manquantes de façon non aléatoire (MNAR de l'anglais « Missing not at random ») dû à des signaux trop faibles pour être détectés, ce qui est un problème connu dans le domaine. En effet, on peut voir plusieurs solutions d'imputation adaptées aux données LC-MS proposées dans la littérature. Plusieurs auteurs mentionnent d'abord la « règle des 80 % », qui stipule qu'une variable présentant plus de 20 % de valeurs manquantes, soit moins de 80 % de données disponibles, doit être exclue de l'ensemble [68, 69]. D'autres chercheurs, tels que Wei et al. (2018), l'emploient comme règle de décision pour choisir si l'imputation de la variable se fera selon une méthode pour données MNAR ou pour des données manquantes dues au hasard (MAR). S'il y a plus de 20% de valeurs manquantes, ils utilisent la méthode « Quantile Regression Imputation of Left-Censored data » (QRILC), sinon ils utilisent une forêt aléatoire [70]. Yuan et al. (2023) utilisent l'optimisation par essaims particulaires pour générer un ensemble de données trouées ressemblant le plus à celui d'origine, puis apprennent à l'aide de différents algorithmes comme la forêt aléatoire à discerner les valeurs manquantes MNAR de MAR [71]. Cette idée de discerner le type de données manquantes pour procéder à l'imputation avec une méthode en conséquence est très prometteuse et est poursuivie dans notre étude. Par exemple, Dubey & Rasool (2020) assument des données MAR pour utiliser des méthodes de regroupement puis ensuite imputent à l'aide d'une méthode de KNN pondéré sur les voisins dans les groupes respectifs [72]. Wei et al. (2018) supposent des données de types MNAR et utilisent donc un algorithme d'imputation basé sur l'échantillonnage de Gibbs en échantillonnant à partir d'une distribution normale tronquée et d'un modèle de régression elastic-net (linéaire L1 et L2). L'échantillonnage force cependant les données imputées à être sous un seuil donné pour respecter le processus de valeurs manquantes, soit MNAR, ce qui restreint la méthode à un

seul mécanisme de valeurs manquantes [73]. D'autres comme Kumar, Hoque, & Sugimoto (2021) ne font aucune hypothèse sur le type de données et utilisent une simple régression linéaire en ajoutant une pondération autour de la médiane permettant ainsi de réduire l'impact des données aberrantes sur l'imputation [74]. Puisque l'on sait que la majorité des valeurs manquantes dans nos données proviennent d'une absence de détection en raison de signaux trop faibles, cette méthode n'est pas adéquate. Toutefois, l'échantillonnage tronqué sous un seuil est pertinent et est exploitée davantage dans notre recherche.

Étant donné la complexité des données protéomiques et métabolomiques et à la grande quantité de variables qui dépasse souvent les milliers, il est fortement recommandé d'appliquer des méthodes de sélection de variables. Il est également conseillé de limiter l'utilisation des méthodes linéaires, car celles-ci peuvent négliger des relations non linéaires importantes présentes dans les données [75]. En effet, dans la littérature, la majorité des recherches se penchent sur des méthodes de sélection de variables non-linéaires comme les forêts aléatoires, l'information mutuelle, l'ExtraTrees, le score VIP ou des réseaux de neurones comme des autoencodeurs [75–79]. L'information mutuelle et le score VIP sont cependant peu efficaces sur des données de grandes dimensions avec peu d'échantillons et ne sont donc pas utilisés dans notre étude. Certains chercheurs développent des modèles plus avancés, notamment Agarwal, Ghanty, & Pal (2020) qui combinent la classification et la sélection de variables en un seul réseau à partir de fonctions de base radiale sur les différents neurones et un poids exponentiel atténuateur des variables pour sélectionner les plus importantes à la tâche [80]. D'autres n'utilisent aucune méthode de sélection de variables, mais utilisent ensuite des réseaux de neurones profonds qui sont connus pour être efficaces pour extraire des relations dans de grands nombres de variables. Ils sont cependant plus propices au surapprentissage lorsque l'on garde énormément de variables avec peu d'échantillons, ce qui est souvent le cas en recherche médicale [81, 82]. Dans le cadre de notre étude, le surapprentissage est effectivement un problème d'ampleur dû aux faible nombre d'échantillons et à la grande quantité de variables. Les méthodes de forêts aléatoires sont donc les plus appropriées dans ce contexte puisqu'elles permettent de modéliser des relations non-linéaires tout en étant moins sensible au surapprentissage.

On peut ensuite examiner quelles méthodes étaient utilisées pour classifier les patients contrôles et les patients atteints d'une maladie à l'aide des données omiques. La majorité des articles tentent différentes méthodes classiques comme les régressions logistiques avec régularisation, les MVS, les forêts aléatoires, les méthodes de boosting comme XGBoost, les discriminants linéaires/quadratiques (LDA/QDA de l'anglais « linear/quadratic discriminant analysis »), les méthodes de Bayes et les RNA. En général, on remarque que les méthodes linéaires comme la régression logistique et le discriminant linéaire performent moins bien que les méthodes

non-linéaires. Le RNA est souvent le meilleur, suivi de XGBoost puis de la forêt aléatoire. Ces méthodes semblent donc être les plus adaptées pour la tâche de classification de notre recherche. Toutefois, elles sont aussi souvent plus difficiles à interpréter sans sélection de variable et nécessitent des étapes supplémentaires, par exemple, la méthode de « SHapley Additive exPlanations » mentionnée plus tôt [76–79, 83–87].

Enfin, parmi ces articles, plusieurs se basent sur des données variées, qu'il s'agisse de protéines ou de métabolites, et proviennent de différents milieux tels que le sang, l'urine ou le liquide cérébrospinal. Il est donc difficile de comparer les méthodes entre les articles. Néanmoins, on voit une tendance chez les méthodes plus complexes comme les forêts aléatoires, la XGBoost et des réseaux de neurones à mieux performer que les méthodes linéaires. On remarque aussi qu'en général, les bases de données métabolomiques performent mieux que celles protéomiques pour la classification de patients. Cette différence de performance peut probablement être attribuée à la complexité accrue des données protéomiques et au développement technologique encore insuffisant dans ce domaine. On voit aussi que les données provenant du liquide cérébrospinal apportent de meilleurs résultats, ce qui est compréhensible puisque les maladies neurodégénératives débutent dans le cerveau.

2.1.3 Approches pour les données de haute dimension avec peu d'échantillons

Cette sous-section porte sur les méthodes plus récentes dans la littérature utilisées pour améliorer les performances des algorithmes classiques sur des bases de données de haute dimension avec peu d'échantillons. Ces méthodes ne sont donc pas spécifiquement conçues pour des données protéomiques, métabolomiques ou médicales. En effet, ce type de données se retrouvent dans de nombreux domaines où il est difficile de prendre des échantillons en grandes quantités et où ces échantillons comportent un grand nombre de variables, ce qui est souvent le cas en médecine. Ces caractéristiques apportent de nombreux problèmes en raison du fléau de la dimensionalité qui stipule que lorsque celle des données augmente, le volume de l'espace de recherche augmente exponentiellement ce qui éparpille les données. Elle est donc peu occupée par les données d'entraînements ce qui rend la généralisation très mauvaise. Un autre problème courant est l'empilement de données où, lorsque l'on fait une projection des données vers un espace latent plus petit, ceux-ci s'empilent et deviennent indiscernables. Des algorithmes tels que LDA peuvent également échouer à converger en raison de la singularité de la matrice de covariance des données d'entraînement [88]. Plusieurs avenues de solutions sont proposées dans la revue de littérature sur différentes étapes de l'approche systématique vers la classification ou la régression. En effet, de nombreuses méthodes vont toucher la sélection de variables pour diminuer l'espace original vers un espace latent comparable au

nombre de données. Certains s'attaquent à des méthodes d'augmentation de données pour améliorer la quantité d'échantillons de manière synthétique tandis que d'autres travaillent directement sur les méthodes de classification pour les adapter aux problèmes.

Commençons par regarder quelques méthodes appliquées pour l'augmentation de données. Chadebec et al. (2023) utilisent un autoencodeur variationnel (VAE de l'anglais « Variational autoencoder »), qui est un modèle de réseaux de neurones autoencodeur qui force une représentation latente probabiliste à l'aide de reparamétrisation permettant de l'échantillonnage par la suite. Les auteurs améliorent ces modèles en transformant l'espace latent vers la géométrie riemannienne et utilisant un échantillonnage de Monte Carlo adapté à ces géométries. Ils montrent que cela permet un échantillonnage plus robuste et riche pour faire de l'augmentation de données ce qui améliore les résultats de classification [89]. Leelarathna et al. (2023) utilisent aussi des VAE, mais avec un ensemble de petit réseaux combinés à l'aide d'un mélange d'experts pour réduire le surapprentissage et obtenir un meilleur espace latent [90]. Les approches par VAE afin d'augmenter le nombre d'échantillons semblent donc pertinentes pour notre recherche. Une autre approche est celle des réseaux antagonistes génératifs, où l'on crée deux réseaux, un génératif qui génère des nouveaux échantillons et un discriminateur qui différencie les vrais échantillons des faux. Les deux s'entraînent en tandem pour tenter d'améliorer leur performance et d'ainsi créer les meilleurs échantillons synthétiques à l'aide du modèle génératif. Nguyen et al. (2024) utilisent ces modèles pour faire de l'augmentation de données dans le contexte de hautes dimensions avec peu d'échantillons. Ils ajoutent aussi un réseau auxiliaire de classification permettant de pousser les réseaux génératifs vers des échantillons représentant mieux leur classes respectives [91]. Pourtant, les réseaux antagonistes génératifs ne sont pas introduits dans notre étude puisqu'ils sont connus comme étant extrêmement sensibles aux données bruitées et aux choix d'hyperparamètres.

Certains articles portent sur l'amélioration des méthodes de sélection de variables. Par exemple, une méthode couramment utilisée est celle de «Hilbert-Schmidt Independence Criterion Lasso» ou HSIC Lasso. Cette méthode est basée sur la régularisation L1 classique à partir des valeurs de dépendance entre chaque variable et la variable de sortie calculée avec le HSIC. Elle permet de retenir les variables les plus pertinentes à la tâche de prédiction et ainsi réduire l'espace de recherche [92]. On voit aussi Mandal et al. (2024) faire de la sélection de variables en plusieurs étapes. En effet, ils commencent par utiliser cinq méthodes différentes, soit Khi-deux, indice GINI, F-score, l'information mutuelle et «Symmetric uncertainty» pour faire un ensemble et choisir les K meilleures variables. Ces variables sont ensuite utilisées dans un algorithme itératif où les auteurs utilisent une méthode d'optimisation, soit l'évaluation différentielle pour trouver le groupe de variables permettant les meilleurs résultats [93]. Toutefois, cette méthode n'est pas robuste aux données bruitées et est donc exclue de notre

étude. En revanche, HSIC Lasso est relativement robuste aux données bruités et permet des interactions non-linéaires, elle est donc explorée dans notre étude.

Il est observé que la majorité des études portant sur des méthodes pour les données de haute dimension avec peu d'échantillons utilisent des modèles intégrés dans lesquels la sélection de variables est effectuée simultanément avec la classification. En effet, on peut voir par exemple, Jiang et al. (2024) utiliser un réseau sous forme d'auto-encodeur, où la dernière couche donne une valeur entre 0 et 1 pour chaque variable, puis utiliser cette nouvelle base pour faire une classification par KNN. Le tout est relié en un seul réseau permettant la propagation du gradient [94]. Cette méthode n'est toutefois pas incluse dans notre étude en raison de sa sensibilité élevée aux hyperparamètres et à la banque de données. On voit aussi plusieurs qui tentent de réduire le problème de surapprentissage en utilisant des plus petits réseaux qui prédisent les poids de plus grands réseaux. Par exemple, Margeloiu et al. (2022) utilisent un petit réseau pour prédire les poids et un autre petit réseau pour la sélection de variables qui sont ensuite combinées dans un plus grand réseau pour la classification [95]. Singh et al. (2020) créent le modèle « Feature Selection Network » qui est sous la forme d'un réseau auto-encodeur supervisé, mais qui utilise deux petits réseaux pour prédire les poids du grand. Ils utilisent aussi une première couche de sélection de variables à l'aide d'une variable aléatoire et du principe de reparamétrisation pour permettre la propagation du gradient puisque la sélection de variables est normalement un problème combinatoire [96]. On voit aussi Liu et al. (2017) tenter une approche itérative avec leur modèle Deep Neural Pursuit, où durant l'entraînement, ils ajoutent progressivement des variables au modèle, puis enlèvent celles considérées comme inutiles par celui-ci. Ils montrent aussi l'efficacité de la méthode de régularisation du « dropout » où l'on enlève aléatoirement des neurones durant l'entraînement [97]. Ces idées pour réduire le nombre de variables dans l'entraînement à l'aide de méthodes itératives ou de couches de sélection sont intéressantes pour réduire les impacts du surapprentissage et sont explorées dans notre étude. Certains chercheurs tentent aussi des approches complètement différentes. Par exemple, Shen et al. (2022) tentent de trouver la meilleure direction de projection qui maximise la dispersion intra-classe tout en gardant les classes dans leur région respective. Cela permet ainsi de réduire l'effet d'empilement de données [98]. De plus, Ziaei et al. (2024) utilisent un processus stochastique gaussien pour projeter les données dans un espace latent pour ensuite inférer avec l'inférence bayésienne. Cavalheiro et al. (2023) combinent la forêt aléatoire avec la MVS en utilisant la matrice de similarité du premier comme noyau pour le second ce qui permet de combiner leurs forces respectives [99, 100]. Cependant, ces méthodes ne sont pas utiles pour faire de l'inférence qui, dans notre cas, est essentiel pour identifier des potentiels biomarqueurs. Finalement, certains chercheurs soulignent l'inefficacité des métriques classiques dans ce domaine, telles que la

distance euclidienne, en raison du grand nombre de dimensions, et proposent des solutions alternatives. Sarkar et al. (2017) proposent l'utilisation du « Model Absolute Density Distance » qui utilise l'écart absolu moyen et la distance dans chaque dimension et montrent que cette métrique apporte de meilleures performances dans les méthodes de regroupement [101]. Modarres (2022) utilise cette métrique pour créer deux indices de dissimilarité et présente quelques manières de les utiliser dans des méthodes de classification comme KNN et LDA montrant de ce fait leur efficacité [88]. Ces indices sont utilisés davantage dans notre étude pour modifier les métriques de distances dans les grandes dimensions.

2.1.4 Points saillants de la revue de littérature pour le projet Parkinson

Pour conclure cette section, les méthodes de classification qui semblent les plus utilisées et adaptées à nos données sont le RNA, la forêt aléatoire, la MVS et le XGBoost. Ces méthodes permettent de modéliser des relations complexes et non linéaires au sein des données tout en étant robustes au surapprentissage avec quelques ajustements. Les méthodes de sélection de variables semblent également très pertinentes pour notre étude, en raison du grand nombre de variables face au nombre restreint d'échantillons de notre ensemble. De plus, puisque les données omiques contiennent souvent un grand nombre de valeurs manquantes, plusieurs méthodes d'imputation ont été présentées pour traiter ce problème. Étant donné que la majorité des valeurs manquantes sont de type MNAR, notre étude apporte une modification des algorithmes d'imputation adaptée à des ratios élevés de données MNAR. On observe également une tendance dans la littérature à utiliser des méthodes tirant avantage des propriétés des données, par exemple, les CNN pour des données temporelles ou spatiales. Nous allons montrer que notre approche originale de classification s'inspire de cette idée en utilisant les interactions protéine-protéine. Deux algorithmes de sous-échantillonnage seront également présentés, permettant de réduire le nombre d'échantillons potentiellement trop bruités et améliorer l'interprétabilité.

2.2 Projet sur une maladie neurologique de type génétique

Dans le cadre de ce projet, nous travaillons avec des données de type longitudinal, plus précisément celles recueillies au fil du temps. Il est donc crucial de déterminer la manière appropriée de les traiter. Ce type de problème, pour lequel un score permettant de suivre l'évolution de la maladie dans le temps est disponible, est bien connu dans le domaine médical. Les études en question sont souvent des essais randomisés en double aveugle, un protocole où deux groupes sont assignés de manière aléatoire. L'un des groupes reçoit un placebo, tandis que l'autre est traité avec un médicament. L'objectif est ensuite d'évaluer l'efficacité

du traitement. Dans notre cas, il s'agit plutôt d'une étude en double aveugle avec démarrage différé, puisque le groupe placebo devient également traité après une certaine période. Nous commencerons donc par présenter les études randomisées en double aveugle, avant de nous pencher sur celles à démarrage différé.

2.2.1 Modèles d'analyse des études randomisées en double aveugle

Deux modèles ressortent principalement de la littérature, soit le modèle mixte pour mesures répétées (MMRM) et le modèle d'analyse de la variance (ANOVA) pour mesures répétées. Le premier est un modèle linéaire très flexible qui permet de poser des effets fixes et des effets aléatoires pour modéliser des variations entre groupes et des variations par sujet. Cela permet, dans le cadre d'une étude longitudinale, de modéliser l'effet du temps comme une variable aléatoire suivant une loi normale, et donc de modéliser la pente dans le temps différemment pour chaque patient. On peut aussi poser une ordonnée à l'origine aléatoire, donc différente pour chaque patient. Dans le cas où l'on veut analyser l'effet d'une médication, donc une étude randomisée en double aveugle, plusieurs approches existent dans la littérature. La plus courante est d'utiliser un modèle classique MMRM où l'on ajoute un effet d'interaction entre le temps et le groupe (placebo ou traité), et l'on regarde si cette variable est significative dans le modèle à l'aide de la valeur-p du test de Wald. Si elle l'est, cela signifie que le traitement a un effet significatif sur l'avancement de la maladie dans le temps. Par exemple, Turner et al. (2015) utilisent un modèle MMRM pour vérifier l'efficacité d'un traitement par resvératrol, un composé naturel avec des propriétés neuroprotectrices. Pour ce faire, ils analysent la variation de différents scores cognitifs et biomarqueurs, en se concentrant sur l'interaction entre la variable catégorique représentant le groupe (traitement ou placebo) et la variable temporelle correspondant au nombre de visites effectuées. Ils utilisent également une structure de covariance autorégressive d'ordre 1, qui repose sur l'hypothèse que les observations plus rapprochées dans le temps sont plus fortement corrélées ; plus précisément, celles-ci diminuent exponentiellement avec l'intervalle de temps [102]. Cependant, ce type de structure est particulièrement adapté lorsque les intervalles de temps sont fixes. Dans notre cas, ceux-ci sont irréguliers, cette approche n'a donc pas été retenue dans notre étude. On peut également mentionner l'étude de Tsuboi et al. (2015), qui ont évalué l'efficacité du zonisamide, un médicament anticonvulsivant, pour traiter la maladie de Parkinson en utilisant l'approche MMRM, bien qu'ils ne précisent pas le type de structure de covariance utilisée [103]. D'autre part, Rascol et al. (2022) expérimentent avec la molécule de foliglurax dans le but de réduire certains symptômes chez les patients parkinsoniens. Ils appliquent cette fois une covariance non structurée, ce qui signifie que chaque paire d'observations dispose de sa propre covariance, permettant ainsi un modèle très flexible qui est utilisé dans

notre étude [104].

D'autres chercheurs vont également plus loin en ajoutant dans le modèle un délai exponentiel causé par l'effet placebo, qui peut parfois améliorer l'état des patients. Cet ajout permet de mieux modéliser les cas où les patients sous placebo voient leur condition s'améliorer durant une courte période au début de l'étude. Cependant, ce délai utilise un coefficient dans l'exponentielle qui doit être justifié par des résultats d'études sur la maladie spécifique ou être approximé [105, 106]. L'ajout de ce terme peut aussi rendre les modèles inutilement complexes et entraîner des résultats instables si la quantité de données est faible. Ces deux raisons nous amènent à ne pas inclure cet effet dans notre recherche. En ce qui concerne les modèles basés sur l'ANOVA, certains chercheurs, comme Navan et al. (2003) ou Devos et al. (2008), utilisent le modèle ANOVA pour mesures répétées afin de tester l'efficacité de leur médication. Ce modèle linéaire évalue les différences de moyennes entre les groupes à différents moments et n'utilise donc pas d'effets aléatoires. Toutefois, il impose une structure de covariance de symétrie composée hétérogène, où la covariance entre les observations est constante. Le modèle repose également sur l'hypothèse de sphéricité, stipulant que la variabilité des scores pour chaque groupe doit être homogène et requiert que les intervalles de temps soient fixes [107,108]. En raison de ces contraintes, il n'est pas utilisé dans notre étude. D'autres chercheurs utilisent plutôt le modèle ANOVA mixte, qui permet l'utilisation d'effets aléatoires sans toutefois modéliser explicitement les corrélations entre les mesures répétées. Ce modèle se concentre davantage sur les effets globaux en utilisant les moyennes des groupes, comme dans le cas du modèle ANOVA pour mesures répétées. Il nécessite également l'hypothèse de sphéricité, bien que celle-ci soit moins critique que dans le modèle ANOVA pour mesures répétées, et requiert des données complètes [109, 110]. Pour ces raisons, ce modèle n'est également pas exploré dans notre recherche.

2.2.2 Modèles d'analyse des études avec démarrage différé

Pour ce qui est des études avec démarrage différé, les modèles MMRM restent ceux privilégiés ; cependant, ils sont utilisés d'une manière différente. En effet, l'analyse des résultats doit être adaptée en raison des deux périodes distinctes de l'étude. Pour vérifier l'efficacité de la médication, les chercheurs testent deux ou trois hypothèses. Par exemple, Olanow et al. (2008) ont testé l'efficacité d'une médication par rasagiline, une molécule permettant de diminuer le niveau de dopamine dans le cerveau et ainsi de réduire les symptômes de la maladie de Parkinson. Pour ce faire, ils ont utilisé un MMRM sur le score clinique pour vérifier trois hypothèses. D'abord, la pente dans le temps du score clinique doit être supérieure pour le groupe traité durant la première période. Il est ensuite nécessaire que, pendant la deuxième

période, la variation du score du groupe ayant reçu le traitement en premier ne soit pas inférieure à celle des nouveaux traités. Cela permet de vérifier que l'effet de la médication persiste bien chez le premier groupe. Enfin, la variation du score entre le début et la fin de l'étude doit être plus grande pour le groupe traité au départ. Pour conclure que la médication a un impact significatif sur l'avancement de la maladie, il est alors nécessaire de vérifier toutes ces hypothèses à l'aide des modèles [111,112]. Verschuur et al. (2019) appliquent le même type de modèle pour vérifier l'efficacité d'une médication par lévodopa, un précurseur de dopamine pour la maladie de Parkinson [113]. Wang et al. (2019) utilisent le même type d'approche, mais modifient les trois hypothèses. En effet, pour réduire l'effet des possibles pentes non linéaires dans les données, ils analysent seulement les différences des scores cliniques entre les deux groupes aux points finaux des périodes. Les deux premières hypothèses sont donc modifiées pour des tests de comparaison des moyennes à la fin des périodes [114]. D'autres chercheurs, comme Chen et al. (2022) utilisent seulement des t-tests à la fin des deux périodes pour vérifier l'efficacité de leur traitement [115]. Cependant, cette méthode ignore la dépendance temporelle des mesures répétées sur des patients, ce qui peut amener à des résultats biaisés. Les méthodes MMRM sont donc plus appropriées pour notre étude.

2.2.3 Points saillants de la revue de littérature pour le second projet

À la lumière de tous ces travaux, on remarque que la méthode la plus appropriée pour analyser des études de mesures répétées semble être le MMRM. Ces modèles sont les plus flexibles, car ils utilisent des effets fixes et aléatoires, permettant de modéliser des pentes temporelles variables pour chaque patient. Notre étude porte sur deux groupes avec un démarrage différé, et la flexibilité du modèle MMRM est donc nécessaire pour bien modéliser les données. Nous allons voir que la contribution de notre projet portera sur la réalisation d'une méthode utilisant le MMRM pour vérifier l'efficacité de la médication en un seul test couvrant toute la période de l'étude.

CHAPITRE 3 RECHERCHE DE BIOMARQUEURS DE LA MALADIE DE PARKINSON

3.1 Introduction au projet

L'objectif de ce projet consiste à valider l'existence de biomarqueurs à l'intérieur du protéome des VEE dans le sang. Pour ce faire, nous avons accès à une base de données de 355 patients, dont 201 sont atteints de différents stades de la maladie de Parkinson et 154 sont des patients contrôles. Plus précisément, nous avons 44 patients au stade léger, 102 au stade modéré, 25 au stade modéré-sévère et 30 au stade sévère. Pour chacun de ces patients, nous avons accès aux intensités relatives obtenues par LC-MS de 1042 protéines. Ces données présentent cependant plusieurs problèmes. Les données sont à la fois très bruitées, très corrélées (en moyenne 25% de corrélation) et contiennent beaucoup de valeurs manquantes (en moyenne 18 % par patient), ce qui rend leur analyse difficile. De plus, le nombre de variables est bien plus élevé que le nombre d'échantillons, et la majorité des protéines n'ont aucun lien avec la maladie de Parkinson, ce qui complexifie la recherche de variables significatives.

Ces problèmes nous conduisent à recourir à des méthodes d'apprentissage machine pour atteindre notre objectif. Pour ce faire, il est important de savoir qu'il existe quatre grandes catégories d'intelligence artificielle, à savoir : l'apprentissage supervisé, non supervisé, semi-supervisé et par renforcement. La première catégorie consiste à avoir un ensemble de données où la valeur de sortie est connue et donc utilisée dans l'entraînement. Dans le deuxième cas, on ne possède pas la valeur de sortie dans la base de données ; l'entraînement doit alors se faire de manière différente. La troisième catégorie est tout simplement une combinaison des deux premières, où certains échantillons ont une valeur de sortie et d'autres non. Finalement, l'apprentissage par renforcement consiste à modéliser un agent qui agit dans un environnement et tente d'optimiser sa politique pour obtenir le plus de récompenses. Dans le cadre de ce projet, l'apprentissage supervisé est mis de l'avant puisque l'on connaît la variable de sortie, soit l'état des patients. Dans certains cas, il peut être pertinent d'utiliser l'apprentissage non supervisé pour optimiser certaines méthodes, en négligeant la valeur de sortie. Trois types de tâches classiques se distinguent parmi ces catégories : la classification, où l'on tente de prédire une catégorie à laquelle l'échantillon appartient (valeur discrète), la régression, où l'on tente de prédire une valeur continue, et finalement le regroupement, où l'on fait de l'apprentissage non supervisé pour tenter de grouper des échantillons selon des caractéristiques communes. Des modèles peuvent aussi combiner différentes tâches. Par exemple, il est possible d'avoir un réseau de neurones avec de nombreuses variables de sortie, certaines de régression et d'autres

de classification. Pour atteindre notre objectif, nous utilisons alors des méthodes supervisées et non supervisées pour classifier des patients contrôles et parkinsoniens. Si les classificateurs sont performants, il devient possible d'inférer quelles protéines ou groupes de protéines sont les plus pertinents pour les modèles et ainsi d'identifier des biomarqueurs potentiels.

Ce chapitre portera d'abord sur l'approche systématique d'apprentissage machine que nous utilisons tout au long du projet. Nous allons brièvement exposer les développements mathématiques de ces méthodes et de nos méthodes originales, les raisons derrière nos choix et discuter de leur pertinence pour notre étude. Ce chapitre sera conclu par un exposé des résultats en comparant les différents modèles pour chaque étape de l'approche systématique ainsi que d'une brève discussion.

3.2 Approche systématique d'apprentissage machine

L'idée derrière l'apprentissage machine est d'utiliser une base de données préexistante, généralement avec une grande quantité de données, puis, à l'aide de modèles mathématiques, d'apprendre des relations entre ces variables qui permettront par la suite au modèle de faire des prédictions sur de nouveaux échantillons jamais présentés. Un modèle général peut être représenté comme ceci :

$$\hat{\mathbf{y}} = f(\mathbf{X}; \boldsymbol{\theta}) \tag{3.1}$$

où $\hat{\mathbf{y}}$ sont les valeurs ou les vecteurs de prédiction, $\mathbf{X}$ est la matrice de variables d'entrée, f est la fonction représentant le modèle et $\boldsymbol{\theta}$ est le vecteur des paramètres du modèle appris à l'aide des données d'entraînement. Ces paramètres peuvent être appris de multiples manières ; la méthode générale est la minimisation d'une fonction de perte L telle que :

$$\boldsymbol{\theta}^* = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L(f(\mathbf{x}_i; \boldsymbol{\theta}), y_i) \tag{3.2}$$

où $f(\mathbf{x}_i; \boldsymbol{\theta})$ est la prédiction de l'i-ème échantillon, y_i la vraie réponse, N est le nombre total d'échantillons et $\boldsymbol{\theta}^*$ le vecteur des paramètres optimisés. On utilise généralement la variable N pour le nombre d'échantillons et M pour le nombre de variables. La matrice $\mathbf{X}$ contenant toutes les données est donc de taille $N \times M$.

L'objectif est de développer un modèle qui optimise la précision des prédictions sur de nouvelles données, ce qu'on appelle la généralisation. Un modèle qui ne parvient pas à généraliser correctement présente des performances insuffisantes et une utilité limitée. Pour certaines

tâches, il est aussi important de pouvoir faire de l'inférence, c'est-à-dire de comprendre comment les variables influencent les décisions du modèle. En général, plus le modèle devient complexe, plus il est difficile de faire de l'inférence.

Un modèle de classification est généralement précédé de nombreuses étapes de prétraitement des données visant à améliorer ses performances, ce qu'on appelle ici l'approche systématique d'apprentissage machine. Dans cette section, nous explorons ces différentes étapes, leurs utilités, leurs méthodes et nos approches originales. Plus précisément, cette section décrira les méthodes de validation croisée, de normalisation, d'imputation, de sélection de variables, d'équilibrage des classes, d'échantillonnage et de prédiction que nous utilisons pour analyser nos données. Ces différentes étapes sont présentées à la figure 3.1, où l'on peut voir les types de méthodes pour chaque étape ainsi que nos méthodes originales.

Tel que mentionné plus tôt, nos données possèdent cependant de nombreux problèmes importants qui reviennent souvent dans la littérature de l'apprentissage machine. Nos approches novatrices ont pour objectif de limiter l'impact de ces problématiques. Dans un premier temps, notre ensemble de données contient une grande quantité de valeurs manquantes, nous implémentons alors une nouvelle méthode d'imputation adaptée à notre problème. Nos données sont également potentiellement très bruitées en raison du manque de développement technique dans la méthodologie de prise de mesure. Pour atténuer ce problème, nous avons ajouté une méthode de sous-échantillonnage à l'approche systématique pour réduire le nombre d'échantillons potentiellement aberrants. Par la suite, pour améliorer l'interprétabilité de notre approche et ainsi faciliter l'identification de biomarqueurs potentiels, nous avons utilisé une méthode originale de sous-échantillonnage pour identifier des patients « prototypes ». Enfin, afin de réduire les problèmes de surapprentissage dus au faible nombre d'échantillons par rapport au grand nombre de variables, nous avons implémenté un réseau de neurones original exploitant les interactions protéine-protéine.

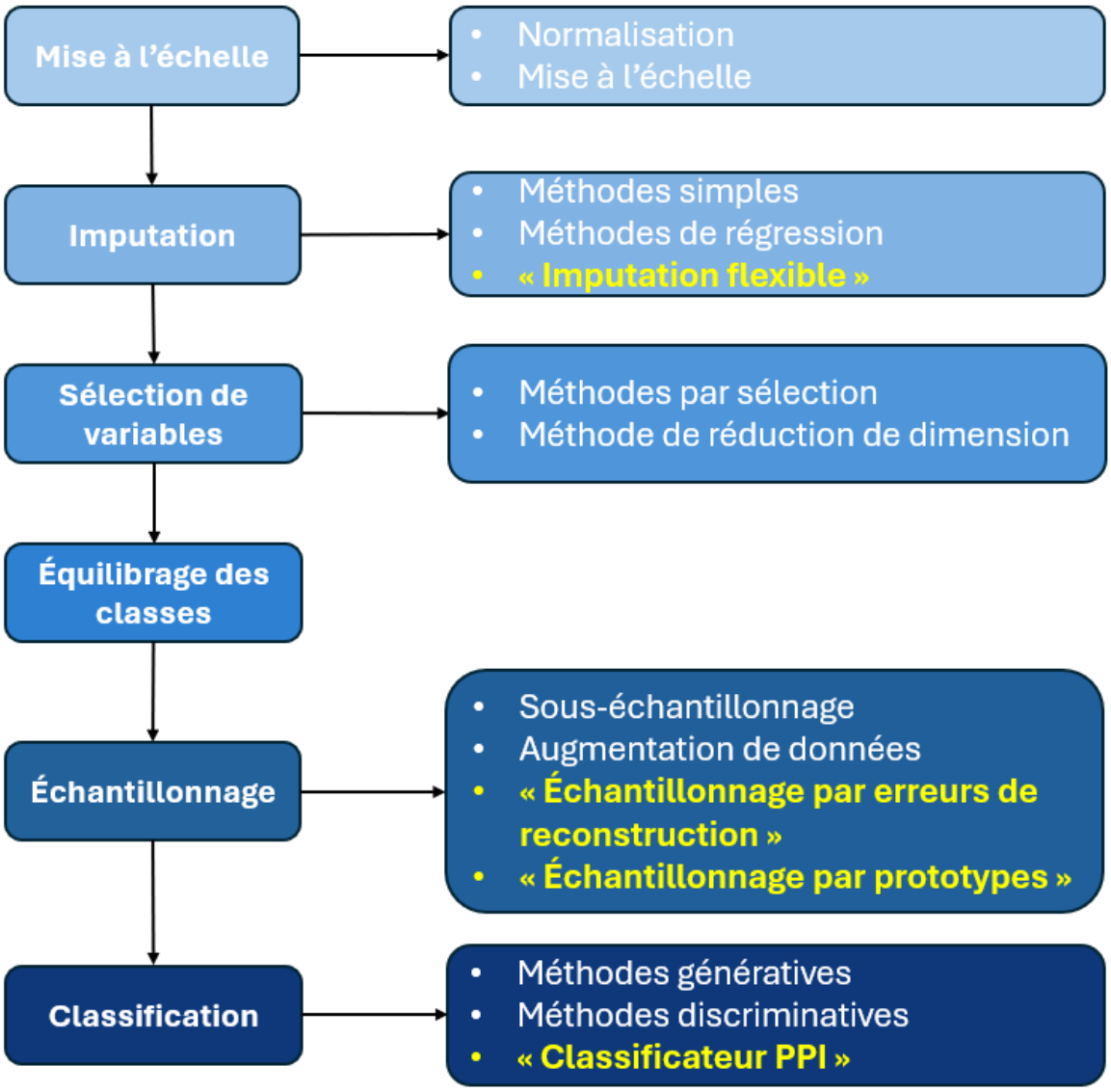

FIGURE 3.1 Diagramme de l'approche systématique avec nos méthodes originales en jaune.

3.2.1 Validation croisée

Pour commencer, il est essentiel de comprendre comment nous évaluons nos modèles. En effet, étant donné que nous disposons d'une banque de données restreinte, nous ne pouvons pas l'utiliser intégralement pour l'entraînement, car cela nous laisserait sans échantillons pour vérifier l'efficacité sur des données jamais vues. Pour ce faire, l'approche naïve consiste à séparer les échantillons. En général, on réserve 80 % de la base de données pour l'entraînement et 20 % pour les tests. Cependant, cette méthode présente une très grande variance, car les 20 % d'échantillons gardés peuvent ne pas bien représenter la capacité de généralisation du

modèle à des données externes. Une approche plus robuste consiste donc à effectuer cette séparation plusieurs fois, de manière aléatoire, et à tester chaque modèle sur les données de test. Cette approche est appelée validation croisée. La méthode la plus couramment utilisée est la validation croisée « K-Fold », où l'on divise la base de données en K groupes, généralement de 5 à 10. Chaque groupe devient, à son tour, le groupe de test, tandis que les autres servent à l'entraînement. Cela permet de tester nos modèles K fois et d'ainsi réduire la variance des résultats.

La majorité des méthodes d'apprentissage automatique utilisent également des hyperparamètres. Ceux-ci ne sont pas calculés par le modèle durant l'entraînement, mais sont choisis au préalable par l'utilisateur. Ils peuvent avoir un impact important sur les performances des modèles, ce qui rend leur optimisation cruciale dans certains cas. Pour ce faire, nous effectuons une autre séparation de la base de données afin d'obtenir un jeu de données pour l'entraînement, un pour les tests et un pour la validation. Ce dernier est utilisé pour sélectionner les meilleurs hyperparamètres sans avoir accès aux données de test, afin de ne pas surestimer le pouvoir de généralisation des modèles. Dans le cadre de notre étude, nous utilisons la validation croisée imbriquée, où une première boucle « K-Fold » est implémentée pour générer des données de test, puis une deuxième est mise en place pour séparer les données d'entraînement et de validation.

Nous observons cependant une instabilité dans les résultats de nos modèles. En raison du faible nombre d'échantillons dans les sous-ensembles de tests et des performances limitées des modèles, les résultats varient considérablement d'un test à l'autre. Afin de réduire cette variance et avoir une meilleure estimation des performances, nous avons répété plusieurs fois ces deux boucles, en modifiant les séparations « K-Fold » de manière aléatoire. L'approche utilisée est donc une validation croisée imbriquée répétée.

3.2.2 Prétraitement

Avant d'utiliser les données dans des calculs, il est important de les prétraiter pour les ajuster à nos futurs modèles. Les données catégoriques doivent être encodées de manière adéquate, et les valeurs continues doivent être normalisées. En effet, cette étape est cruciale afin que toutes les variables soient traitées de manière uniforme lors des entraînements des modèles. Sans cette étape, les variables possédant des valeurs intrinsèquement élevées recevraient plus d'attention par le modèle durant l'entraînement que les variables avec des valeurs faibles. Cela permet également aux algorithmes de converger plus rapidement. Certaines méthodes, comme les MVS et les RNA, performent aussi mieux lorsque les distributions des données suivent des lois normales. La mise à l'échelle est aussi utilisée pour réduire l'asymétrie dans

certaines distributions, ce qui est fréquent dans nos données. Différentes méthodes existent ; parmi celles-ci, quatre ont été testées durant notre recherche. Cette étape peut, en théorie, être réalisée avant ou après l'étape d'imputation, où l'on estime les valeurs manquantes. Les deux choix présentent des avantages et des inconvénients. Pour notre étude, nous avons opté pour effectuer le prétraitement en premier, puisque nous avons une très grande quantité de valeurs manquantes. Nous allons donc utiliser des méthodes de régression pour l'imputation, ce qui peut créer des problèmes si les données ne sont pas mises à l'échelle.

Encodage

Il est nécessaire d'encoder les données de manière appropriée. Cette étape est importante principalement pour les variables catégoriques. Dans le cas d'une classification binaire comme la nôtre, on peut simplement avoir un scalaire représentant la probabilité d'appartenir à la classe 1 (ici représentant un patient parkinsonien), et soustraire cette valeur à 1 pour obtenir la probabilité d'appartenir à la classe 0. Notre vecteur de prédiction est donc composé de « 1 » et de « 0 », représentant chaque classe, et les modèles vont prédire une probabilité d'appartenir à la classe 1. Puisque le reste de notre ensemble est constitué de variables continues, aucun encodage supplémentaire n'est nécessaire.

Standardisation

La première méthode est la standardisation, qui permet de transformer la distribution des données pour avoir une moyenne nulle et un écart type égal à un. La matrice des variables mises à l'échelle $\mathbf{X}_{échelle}$ est donnée par :

$$\mathbf{X}_{échelle} = \frac{\mathbf{X} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \tag{3.3}$$

où $\boldsymbol{\mu}$ est le vecteur des moyennes des variables et $\boldsymbol{\sigma}$ celui de leur écart type. Cette approche est principalement utilisée pour les modèles qui nécessitent des données distribuées de cette manière, comme la MVS, le RNA ou la régression logistique.

Mise à l'échelle robuste

Dans le cas où l'on sait que nos données sont asymétriques ou possèdent des valeurs aberrantes, on peut utiliser une mise à l'échelle robuste définie par :

$$\mathbf{X}_{échelle} = \frac{\mathbf{X} - \mathbf{Q1}}{\mathbf{Q3} - \mathbf{Q1}}. \tag{3.4}$$

Celle-ci utilise **Q1**, le vecteur de médiane du premier quartile, et **Q3**, le vecteur de médiane du troisième quartile, pour réduire l'effet des données aberrantes sur la mise à l'échelle.

Mise à l'échelle min-max

Si l'on cherche à forcer les données à être à l'intérieur d'un intervalle donné, on peut utiliser la mise à l'échelle min-max calculée comme suit :

$$\mathbf{X}_{échelle} = \frac{\mathbf{X} - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}} \tag{3.5}$$

où $\mathbf{x}_{min}$ est le vecteur des valeurs minimales des variables et $\mathbf{x}_{max}$ celui des valeurs maximales. Cette méthode est utile lorsque l'on cherche à avoir des données dans un intervalle, généralement entre 0 et 1. Elle n'est cependant pas robuste aux données aberrantes et asymétriques.

Mise à l'échelle maximum absolue

Finalement, si la banque de données est creuse, c'est-à-dire qu'elle contient beaucoup de zéros, on peut utiliser la mise à l'échelle maximum absolue pour forcer les données dans l'intervalle $[-1, 1]$ tout en gardant les valeurs à 0. On a alors :

$$\mathbf{X}_{échelle} = \frac{\mathbf{X}}{|\mathbf{x}_{max}|}. \tag{3.6}$$

Pour notre étude, nous nous attendons à ce que la standardisation et la mise à l'échelle robuste offrent de meilleures performances en tant que première normalisation. Toutefois, les deux autres méthodes sont nécessaires dans différents cas pour forcer les données dans un intervalle connu.

3.2.3 Imputation

Notre ensemble de données protéomiques est composé d'une grande partie de données manquantes. En effet, chaque échantillon possède en moyenne 18 % de valeurs manquantes, avec des patients en possédant jusqu'à 38 %. Il est donc essentiel de tester différents types de méthodes d'imputation pour bien compléter notre jeu de données. Pour cela, il est important de noter que, selon les experts ayant effectué les mesures, la majorité des données manquantes sont de type MNAR (« Missing not at random »). Ces valeurs sont donc manquantes en raison d'un seuil de détection trop élevé, ce qui entraîne l'absence de valeurs d'intensité faibles. En effet, lors de la prise de mesures effectuée par nos collaborateurs, les instruments utilisés

n'étaient pas en mesure de détecter toutes les protéines pour chaque patient si elles étaient en faible quantité, ce qui entraînait des valeurs manquantes. Néanmoins, il n'est pas certain que toutes les valeurs manquantes proviennent de ce processus ; certaines d'entre elles pourraient aussi être dues au simple hasard, c'est-à-dire MAR (« Missing at random »). C'est pourquoi nous testons, dans cette étude, différentes méthodes applicables aux deux types de valeurs manquantes.

Méthodes simples

Les méthodes d'imputation simples permettent de rapidement compléter une base de données ; elles sont cependant peu efficaces lorsque l'on fait face à des données complexes. Dans le cas de données MAR, on peut utiliser les moyennes, médianes ou modes du reste des données pour remplacer les valeurs manquantes. Dans notre cas, avec des données MNAR en raison d'un seuil de détection, on peut imputer en utilisant la valeur minimale ou la moitié de la valeur minimale de l'intensité de la protéine en question.

Méthodes par régression

Pour améliorer les modèles d'imputation, on peut utiliser le reste des valeurs connues pour comprendre et exploiter les relations entre les variables. Cela revient à un problème de régression que l'apprentissage automatique peut résoudre. Toutefois, la majorité des méthodes nécessitent une base de données complète, ce qui restreint le choix des méthodes. La forêt aléatoire est souvent privilégiée pour ce type de tâche, puisqu'elle est suffisamment puissante pour modéliser des relations non linéaires tout en utilisant des échantillons comportant des valeurs manquantes. Dans notre cas, ces méthodes consistent, pour chaque protéine, à utiliser les intensités des autres protéines afin de prédire les intensités manquantes. Une autre approche consiste à faire une première imputation à l'aide d'une des méthodes simples, puis à appliquer d'autres types de régression sur les données complétées. Cette approche en deux étapes est utilisée dans notre méthode originale.

QRILC

Cette méthode modélise les quantiles des données connues pour ensuite échantillonner dans une loi normale tronquée et imputer des données MNAR. Plus précisément, elle estime la moyenne et la variance des quantiles des données en supposant qu'elles suivent des lois normales. Cette estimation permet d'obtenir des prédictions plus représentatives des données dans leurs quantiles respectifs, puisque les relations peuvent être différentes entre chacun

d'entre eux. Par la suite, l'imputation se fait en générant des échantillons à partir d'une loi normale tronquée (troncature au quantile correspondant au pourcentage de valeurs manquantes) en utilisant un modèle de régression linéaire entre les quantiles observés et un quantile normal. Ce modèle est donc utile dans le cadre de notre recherche puisqu'il s'applique à des données manquantes de type MNAR tronquées vers la gauche, exactement comme nos intensités de protéines. Il est cependant restreint à ce type de problème et n'est pas très flexible puisqu'il suppose que toutes les valeurs manquantes sont de type MNAR.

GSimp

GSimp est un modèle itératif appliqué aux données MNAR tronquées à gauche. Pour ce faire, il utilise un échantillonneur de Gibbs et une régression linéaire « Elastic Net » en tandem. Cette méthode de régression linéaire pénalisée est présentée brièvement dans la section des modèles de prédiction. L'algorithme commence par une première imputation à l'aide de la méthode QRILC pour obtenir un jeu de données complet. Par la suite, de manière séquentielle, on entraîne un modèle pour chaque variable en fonction du reste des données, et on calcule l'écart quadratique moyen entre les prédictions et les valeurs initiales. Pour mettre à jour les valeurs manquantes, on utilise une loi normale tronquée sous la valeur minimale, dont la moyenne suit celle des données, et l'écart type est donné par l'écart quadratique moyen entre les prédictions et les vraies valeurs. Ces étapes sont répétées de manière itérative après avoir mis à jour le jeu de données complet initial. Cependant, ces itérations peuvent augmenter le surapprentissage et le biais apporté par le modèle de régression [73]. La méthode reste pertinente pour notre étude en raison de son approche pour les valeurs manquantes MNAR tronquées à gauche et de sa complexité, qui peut bien s'adapter à nos données.

Méthode originale d'imputation flexible

Comme nous l'avons vu dans la revue de littérature, différentes méthodes d'imputation s'appliquent aux valeurs manquantes de type MAR, tandis que d'autres s'appliquent à celles de type MNAR. Les chercheurs ont souvent tendance à faire des hypothèses sur les types de valeurs manquantes, puis à imputer à l'aide d'une méthode appropriée. Certains ont également tenté des méthodes visant à classifier le type de variable manquante pour ensuite réaliser l'imputation correctement. Cependant, cette approche est très sensible à la première étape de classification. En effet, si des erreurs surviennent à cette étape, l'imputation qui en découle sera automatiquement erronée. Pour atténuer ce problème, notre approche ne cherche pas à classifier les variables, mais à imputer les données en formulant une hypothèse souple selon laquelle la majorité des valeurs manquantes résulte d'un seuil de détection trop élevé. Cette

hypothèse s'appuie également sur l'expertise des chercheurs qui nous ont fourni les données.

Nous supposons d'abord que toutes les valeurs manquantes proviennent d'un seuil de détection trop élevé. Cela implique que les valeurs manquantes doivent nécessairement être inférieures à l'intensité minimale de leur protéine respective. Toutefois, cela ne signifie pas que le seuil de détection correspond exactement à la valeur minimale, car il doit être plus petit ou égale à celle-ci. Nous cherchons alors à estimer ce seuil de détection pour permettre une imputation plus précise. Pour mieux représenter la réalité, nous n'utilisons pas directement la valeur du seuil estimé comme imputation, mais plutôt comme valeur maximale d'une distribution. Plus précisément, nous posons comme hypothèse que les données proviennent de distributions normales et que les valeurs manquantes sont tronquées au seuil estimé.

Notre approche consiste donc de trois étapes d'imputation différentes qu'on peut voir dans l'algorithme 1. Dans un premier temps, nous cherchons à créer un ensemble de données complet pour faire l'estimation du seuil. Pour chaque variable, nous estimons les paramètres d'une distribution gaussienne et échantillonnons la distribution tronquée à la valeur minimale des données. Nous posons alors l'hypothèse simple, dans un premier temps, que le seuil de détection est égal à la valeur minimale de nos données. Cela nous permet de créer un nouvel ensemble de données complet, où les valeurs manquantes sont toutes inférieures aux valeurs minimales de leur variable respective. On a alors la première imputation $\mathbf{X}_{\text{imputé}}$ donnée par :

$$\mathbf{X}_{\text{imputé}} \sim \mathcal{N}_{(-\infty,\mathbf{x}_{\min}]}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2) \tag{3.7}$$

où $\mathcal{N}_{(-\infty,\mathbf{x}_{\min}]}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ représente une loi normale tronquée entre $-\infty$ et $\mathbf{x}_{\min}$ avec le vecteur des moyennes $\boldsymbol{\mu}$ et le vecteur des écarts types $\boldsymbol{\sigma}$. La deuxième étape consiste à optimiser cette première imputation simple en tentant de trouver la vraie valeur du seuil de détection. Pour ce faire, nous avons décidé d'utiliser une approche par vraisemblance maximale, qui est souvent utilisée pour estimer des paramètres de distribution. Nous cherchons alors à maximiser la vraisemblance $\mathcal{L}$ des données calculée comme suit :

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{t}) = \prod_{i \in \text{observé}} \left(\frac{1}{\sqrt{2\pi\boldsymbol{\sigma}^2}} \exp\left(-\frac{(\mathbf{x}_i - \boldsymbol{\mu})^2}{2\boldsymbol{\sigma}^2} \right) \right) \cdot \prod_{i \in \text{manquante}} \left(\frac{\phi\left(\frac{\mathbf{x}_i - \boldsymbol{\mu}}{\boldsymbol{\sigma}}\right)}{\Phi\left(\frac{\mathbf{t} - \boldsymbol{\mu}}{\boldsymbol{\sigma}}\right)} \right) \tag{3.8}$$

où $\phi(\mathbf{x})$ est la fonction de densité d'une loi normale, $\Phi(\mathbf{x})$ est la fonction de répartition d'une loi normale, et $\mathbf{t}$ est le vecteur des valeurs du seuil. Le premier terme de l'équation représente donc la vraisemblance des données connues, en supposant encore une fois des distributions normales. Le deuxième terme, quant à lui, représente la vraisemblance des données imputées en supposant la loi normale tronquée vers la gauche au seuil $\mathbf{t}$. En prenant le logarithme de

cette vraisemblance on a :

$$\ell(\boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{t}) = \sum_{i \in \text{observé}} \log\left(\frac{1}{\sqrt{2\pi\boldsymbol{\sigma}^2}} \exp\left(-\frac{(\mathbf{x}_i - \boldsymbol{\mu})^2}{2\boldsymbol{\sigma}^2}\right)\right) + \sum_{i \in \text{manquante}} \log\left(\frac{\phi\left(\frac{\mathbf{x}_i - \boldsymbol{\mu}}{\boldsymbol{\sigma}}\right)}{\Phi\left(\frac{\mathbf{t} - \boldsymbol{\mu}}{\boldsymbol{\sigma}}\right)}\right). \quad (3.9)$$

On cherche alors à optimiser les paramètres de la distribution qui maximisent cette fonction ℓ. Pour ce faire, on utilise l'algorithme d'espérance-maximisation (EM). Cet algorithme se déroule en deux étapes. L'étape d'espérance consiste à imputer les données manquantes à partir des paramètres estimés, pour ensuite calculer la vraisemblance. Par la suite, l'étape de maximisation consiste à optimiser les paramètres de la distribution afin de maximiser la vraisemblance. Cette étape peut utiliser différentes méthodes d'optimisation, comme la descente de gradient, quasi-Newton, Newton-Raphson, etc. Dans le cadre de notre recherche, nous utilisons l'algorithme de Broyden-Fletcher-Goldfarb-Shanno, qui est facilement implémentable dans le langage Python. Après avoir optimisé les paramètres, ceux-ci sont finalement utilisés pour créer la distribution normale tronquée, qui permet la deuxième imputation donnée par :

$$\mathbf{X}_{\text{imputé}} \sim \mathcal{N}_{]-\infty, \mathbf{t}]}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2). \quad (3.10)$$

Nous avons ainsi un ensemble de données complet, avec comme hypothèse que les variables manquantes proviennent du processus MNAR en raison d'un seuil de détection trop élevé. Cependant, cette hypothèse n'est pas nécessairement vraie pour toutes les variables de notre ensemble de données. En effet, selon les experts, la majorité des valeurs manquantes proviennent de ce processus, mais pas nécessairement toutes. Certaines peuvent être dues à d'autres problèmes survenant durant la prise de mesures qui ne sont pas expliqués par le seuil de détection, donc aléatoire. Pour corriger l'imputation des variables qui sont en fait de type MAR, nous terminons par une dernière imputation en appliquant une régression par apprentissage machine sur chaque variable en fonction des autres. En théorie, sachant que nous utilisons un ensemble de données maintenant complet, n'importe quel algorithme de régression pourrait être utilisé. Malgré tout, comme nous sommes dans un problème de très grande dimension, cette régression va faire face à du surapprentissage. Puisque nos données sont très corrélées, nous profitons de cette propriété pour utiliser une simple régression linéaire sur les K variables les plus corrélées afin de réduire le surapprentissage. La régression linéaire tente de modéliser l'espérance conditionnelle des variables d'intérêt sachant les valeurs observées, soit $\mathbb{E}[X_{\text{vrai}}|X_{\text{obs}}]$. On peut alors s'attendre à ce que la régression améliore la prédiction des variables de type MAR, mais qu'elle n'affecte que peu les variables de type

MNAR puisqu'elles suivent déjà la vraie distribution des données. Les variables mal imputées dans un premier temps auront alors un biais dû à cette erreur, qui est tel que :

$$\mathbb{E}[\hat{\mathbf{X}}^{\text{MAR}}|\mathbf{X}_{\text{obs}}] \neq \mathbb{E}[\mathbf{X}_{\text{vrai}}^{\text{MAR}}|\mathbf{X}_{\text{obs}}] \tag{3.11}$$

et donc,

$$\mathbb{E}[\hat{\mathbf{X}}^{\text{MAR}}|\mathbf{X}_{\text{obs}}] = \mathbb{E}[\mathbf{X}_{\text{vrai}}^{\text{MAR}}|\mathbf{X}_{\text{obs}}] + \mathbf{b}_{\text{MAR}} \tag{3.12}$$

où $\hat{\mathbf{X}}^{\text{MAR}}$ désigne les variables manquantes de type MAR imputées, $\mathbf{X}_{\text{vrai}}^{\text{MAR}}$ les vraies valeurs de ces variables et $\mathbf{b}_{\text{MAR}}$ les biais créés par la première imputation sous l'hypothèse MNAR. La régression linéaire permettrait alors de réduire ce biais puisque les distributions ne seraient pas alignées. On a alors :

$$\mathbb{E}[\hat{\mathbf{X}}_{\text{corrigé}}^{\text{MAR}}|\mathbf{X}_{\text{obs}}] = \mathbb{E}[\mathbf{X}_{\text{vrai}}^{\text{MAR}}|\mathbf{X}_{\text{obs}}] + \mathbf{b}_{\text{corrigé MAR}} \tag{3.13}$$

avec, $\mathbf{b}_{\text{corrigé MAR}} < \mathbf{b}_{\text{MAR}}$ où $\hat{\mathbf{X}}_{\text{corrigé}}^{\text{MAR}}$ sont les variables imputées après la régression linéaire et $\mathbf{b}_{\text{corrigé MAR}}$ les biais après la correction. Les variables bien imputées, quant à elles, suivront la distribution des données et seront alors peu affectées par la régression linéaire, comme suit :

$$\mathbb{E}[\hat{\mathbf{X}}^{\text{MNAR}}|\mathbf{X}_{\text{obs}}] \approx \mathbb{E}[\mathbf{X}_{\text{vrai}}^{\text{MNAR}}|\mathbf{X}_{\text{obs}}] \tag{3.14}$$

où $\hat{\mathbf{X}}^{\text{MNAR}}$ sont les variables manquantes imputées de type MNAR et $\mathbf{X}_{\text{vrai}}^{\text{MNAR}}$ leurs vraies valeurs. Cette régression devrait donc avoir peu d'effet sur les variables qui ont déjà été correctement imputées, mais modifierait drastiquement l'imputation de celles de type MAR. Il est également possible d'utiliser des régressions avec régularisation L1 ou L2, mais nous avons remarqué expérimentalement avec nos données que celles-ci ne convergeaient pas.

Finalement, puisque les deux premières imputations utilisent de l'échantillonnage aléatoire à partir de la distribution tronquée, nous répétons la première partie de l'imputation P fois pour faire P prédictions et faire leur moyenne pour obtenir la valeur finale. Cette étape permet de réduire la variance apportée par l'échantillonnage aléatoire sur les imputations finales.

Algorithme 1 : Algorithme d'imputation originale

Entrée :

- Ensemble de données $\mathbf{X}$ de taille $N \times M$
- Nombre K de variables corrélées
- Nombre P de répétitions
- Nombre I d'itérations maximales de la méthode EM
- Tolérance Tol de l'algorithme EM

Sortie : $\mathbf{X}_{imputé}$

$Masque = \textbf{IndexValeursManquantes}(\mathbf{X})$

pour $j \leftarrow 1$ *à* M **faire**

$\mu_j = \frac{1}{N} \sum_{i=1}^{N} x_{i,j}$

$\sigma_j = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_{i,j} - \mu_j)^2}$

$t_j = min(\mathbf{x_j})$

$\mathbf{X}[Masque, j] \sim \mathcal{N}_{]-\infty, t_j)]}(\mu_j, \sigma_j^2)$

fin

pour *itération* $\leftarrow 1$ *à* I **faire**

pour $j \leftarrow 1$ *à* M **faire**

$\mathbf{X}[Masque, j] \sim \mathcal{N}_{]-\infty, t_j]}(\mu_j, \sigma_j^2)$

$\ell(\mu_j, \sigma_j, t_j) =$

$\sum_{i \in \text{observé}} \log\left(\frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_{i,j}-\mu_j)^2}{2\sigma_j^2}\right)\right) + \sum_{i \in \text{manquante}} \log\left(\frac{\phi\left(\frac{x_{i,j}-\mu_j}{\sigma_j}\right)}{\Phi\left(\frac{t_j-\mu_j}{\sigma_j}\right)}\right)$

$\mu_j, \sigma_j, t_j \leftarrow \arg\max_{\mu_j, \sigma_j, t_j}(\ell(\mu_j, \sigma_j, t_j))$

fin

si $\ell_{itération-1} - \ell_{itération} < Tol$ **alors**

Sortir de la boucle

fin

fin

pour $j \leftarrow 1$ *à* M **faire**

$\mathbf{X}_{entraînement} = \textbf{PlusCorrélées}(K, \mathbf{x}_j, \mathbf{X}_{\sim j})$

$\mathbf{y}_{entraînement} = \mathbf{x}_j$

$\textbf{Modèle}_j = \textbf{RégressionLinéaire}(\mathbf{X}_{entraînement}, \mathbf{y}_{entraînement})$

pour $p \leftarrow 1$ *à* P **faire**

$\mathbf{X}_{prédiction} \sim \mathcal{N}_{]-\infty, t_j]}(\mu_j, \sigma_j^2)$

$Prédiction_p = \textbf{Modèle}_j(\mathbf{X}_{prédiction}[Masque])$

fin

$\mathbf{X}[Masque, j] \leftarrow \frac{1}{P} \sum_{p=1}^{P} Prédiction_p$

fin

$\mathbf{X}_{imputé} = \mathbf{X}$

3.2.4 Sélection de variables

L'étape de la sélection de variables peut être séparée en deux grandes catégories : la sélection et la réduction de dimension. Dans le premier cas, on cherche les meilleures variables parmi l'ensemble de données selon différents critères. Dans le deuxième cas, on crée un espace latent qui condense l'information à partir des relations entre les variables initiales. Cette étape est très importante dans notre recherche pour réduire l'effet de l'une des problématiques de nos données, à savoir le grand nombre de dimensions. Elle aide aussi à l'interprétabilité des modèles, puisque le nombre de variables utilisées pour la prédiction est réduit, ce qui permet d'identifier plus facilement des biomarqueurs potentiels.

Sélection

Deux catégories de méthodes seront utilisées dans notre étude pour la sélection : les méthodes de filtre et les méthodes intégrées. Les méthodes de filtre organisent les variables en fonction de leurs propriétés intrinsèques ou de tests statistiques. Par exemple, on peut utiliser le coefficient de corrélation de Pearson $r(\mathbf{x}_1;\mathbf{x}_2)$ ou l'information mutuelle $I(\mathbf{x}_1;\mathbf{x}_2)$ entre les vecteurs de variables $\mathbf{x}_1$ et $\mathbf{x}_2$ donnés par :

$$r(\mathbf{x}_1;\mathbf{x}_2) = \frac{\sum_{i=1}^{N}(x_{1,i}-\bar{x_1})(x_{2,i}-\bar{x_2})}{\sqrt{\sum_{i=1}^{N}(x_{1,i}-\bar{x_1})^2}\sqrt{\sum_{i=1}^{N}(x_{2,i}-\bar{x_2})^2}} \tag{3.15}$$

et

$$I(\mathbf{x}_1;\mathbf{x}_2) = \int\int p(\mathbf{x}_1,\mathbf{x}_2)\log\left(\frac{p(\mathbf{x}_1,\mathbf{x}_2)}{p(\mathbf{x}_1)p(\mathbf{x}_2)}\right) d\mathbf{x}_1\, d\mathbf{x}_2 \tag{3.16}$$

pour retirer celles trop corrélées.

Sachant que nos données sont très corrélées, ces méthodes peuvent être utiles pour réduire le nombre de variables en éliminant celles apportant de l'information redondante. Ces valeurs ne prennent cependant pas en compte la tâche de prédiction ; ce sont des méthodes non supervisées. On peut combiner l'information mutuelle entre les variables ainsi qu'entre chaque variable et la variable cible afin de créer un modèle optimisant une redondance minimale et une pertinence maximale (mRMR). Ce critère est donné par :

$$\text{Critère mRMR :} \quad \max_{\mathbf{x}_i\in\mathbf{F}-\mathbf{S}}\left(I(\mathbf{x}_i;y) - \frac{1}{|\mathbf{S}|}\sum_{\mathbf{x}_j\in\mathbf{S}} I(\mathbf{x}_i;\mathbf{x}_j)\right) \tag{3.17}$$

où $\mathbf{F}$ est l'ensemble de toutes les variables et $\mathbf{S}$ l'ensemble des variables déjà choisies.

On peut aussi utiliser des tests statistiques pour trouver des variables ayant des différences

significatives de moyennes ou de distributions entre les classes, comme le test de Mann-Whitney U. Ce test non paramétrique permet de vérifier si l'on peut rejeter l'hypothèse nulle H_0, selon laquelle les distributions sont identiques. L'hypothèse alternative H_1 est que $p(\mathbf{x}_0 > \mathbf{x}_1) \neq p(\mathbf{x}_1 > \mathbf{x}_0)$ où $\mathbf{x}_0$ appartient à la classe 0 et $\mathbf{x}_1$ à la classe 1.

Les méthodes intégrées, quant à elles utilisent des algorithmes d'intelligence artificielle pour s'entraîner sur la tâche de prédiction et identifier les variables les plus importantes à la prédiction. Dans cette catégorie, on peut par exemple voir la méthode de régularisation Lasso qui sera présentée dans la section sur les méthodes de prédiction. En effet, celle-ci a tendance à faire tendre les poids liés aux variables inutiles vers 0, ce qui permet de sélectionner le reste de l'ensemble comme pertinent.

Dans le cadre de notre recherche, nous utilisons une variante de cette méthode adaptée aux données de haute dimension avec peu d'échantillons. La méthode HSIC Lasso utilise le critère d'indépendance d'Hilbert-Schmidt pour cibler les variables pertinentes individuellement. Elle permet également, à l'aide de fonctions de noyau, de créer des relations non linéaires entre les variables.

On peut aussi utiliser différents modèles d'apprentissage automatique qui offrent des valeurs d'importance aux variables à la fin de l'entraînement. Dans le cadre de notre étude, nous avons testé la forêt aléatoire pour ses performances en haute dimension et en présence de relations non linéaires. Cette méthode permet de calculer un score d'importance pour chaque variable en fonction de la diminution d'impureté qu'elle apporte aux arbres. Cette notion sera présentée davantage dans la section des méthodes de prédiction.

Réduction de dimension

Les méthodes de réduction de dimension ont pour but de trouver de nouvelles variables qui permettent de garder seulement l'information pertinente dans un espace latent plus petit en combinant de différentes manières les variables d'entrée. Les méthodes de réduction de dimension permettent aussi de réduire, dans certains cas, la quantité de bruit dans nos données en condensant l'information pertinente dans l'espace latent. Cependant, certaines méthodes plus complexes, comme les réseaux de neurones autoencodeurs, rendent l'inférence plus difficile par la suite.

La méthode la plus couramment utilisée est celle de l'analyse en composantes principales, dans laquelle on tente de trouver les combinaisons linéaires de nos variables permettant d'expliquer la majorité de la variance. Pour ce faire, on utilise les vecteurs propres de la matrice de covariance de $\mathbf{X}$ donnés par :

$$\mathbf{\Sigma w} = \lambda \mathbf{w}. \tag{3.18}$$

Les composantes principales sont donc les vecteurs propres $\mathbf{w}$ de la matrice de covariance $\mathbf{\Sigma}$, et λ sont les valeurs propres représentant la variance. Il ne reste plus qu'à choisir le nombre de variables que l'on veut parmi les nouvelles, en commençant par celles expliquant le plus la variance à l'aide de :

$$\mathbf{X_{transformé}} = \mathbf{XW} \tag{3.19}$$

où $\mathbf{W}$ est la matrice des vecteurs propres et $\mathbf{X_{transformé}}$ notre nouvelle matrice de données. On doit généralement centrer la matrice $\mathbf{X}$ avant de calculer la matrice de covariance pour bien capturer la direction de la variance.

Cette méthode ne prend aucunement en compte les valeurs à prédire, c'est donc une méthode non supervisée. Si l'on cherche à utiliser ces valeurs, la méthode des moindres carrés partiels (PLS de l'anglais « Partial least squares ») s'y prête bien. Celle-ci cherche à optimiser les projections dans un espace latent de $\mathbf{X}$ permettant d'expliquer le plus possible la variance d'un espace latent de $\mathbf{y}$. Pour ce faire, il existe de nombreux algorithmes qui utilisent des approches différentes ; celui utilisé dans notre étude se nomme *PLS1*. Cette méthode permet en théorie de trouver itérativement de nouvelles variables expliquant la variation des intensités des protéines et leur relation avec la maladie de Parkinson. Cependant, elle reste linéaire ce qui n'est pas optimal pour la complexité de nos données.

3.2.5 Équilibrage des classes

Il arrive souvent, comme dans notre cas, que le nombre d'échantillons par classe ne soit pas équilibré. Dans ces cas, les algorithmes auront souvent tendance à négliger la classe minoritaire durant l'entraînement, puisque cela leur permet d'avoir un meilleur taux de classification sur la classe majoritaire et donc une perte plus faible en moyenne. Différentes méthodes peuvent être appliquées à l'intérieur des algorithmes même pour réduire cet effet, toutefois la manière la plus simple reste d'équilibrer les classes. Pour ce faire, on peut tout simplement échantillonner dans l'ensemble de données de la classe minoritaire jusqu'à ce que les classes soient équilibrées. Cependant, cela ne fait que créer des doublons durant l'entraînement. Une méthode plus appropriée est la «Synthetic Minority Over-sampling Technique» (SMOTE). Dans cette méthode, on choisit au hasard un échantillon de la classe minoritaire, puis on utilise un des K plus proches voisins de la même classe au hasard pour créer un nouvel échantillon synthétique en interpolant entre ces deux échantillons. On peut alors calculer le

nouvel échantillon $\mathbf{x}_{\text{nouveau}}$ donné par :

$$\mathbf{x}_{\text{nouveau}} = \mathbf{x}_i + \lambda(\mathbf{x}_j - \mathbf{x}_i), \quad \lambda \in [0, 1] \tag{3.20}$$

Où $\mathbf{x}_i$ est le vecteur échantillon pris au hasard, $\mathbf{x}_j$ le vecteur de l'un des K plus proches voisins et λ une valeur aléatoire entre 0 et 1 d'une distribution uniforme. On répète ces étapes jusqu'à ce que les classes soient équilibrées. Cette approche est utilisée dans notre étude pour faire un premier équilibrage des classes afin d'avoir autant d'échantillons de la classe contrôle que de la classe parkinsonienne.

3.2.6 Sous-échantillonnage et augmentation de données

Cette section porte sur les méthodes permettant de réduire le nombre d'échantillons d'entraînement, ce qu'on appelle le sous-échantillonnage, ainsi que sur les méthodes d'augmentation de données pour créer de nouveaux échantillons synthétiques. Ces méthodes sont généralement utilisées pour améliorer le pouvoir de généralisation des modèles de prédiction en retirant les mauvais échantillons et en augmentant la diversité des échantillons dans les données d'entraînement. Nous allons ici présenter deux méthodes originales : le sous-échantillonnage par erreurs de reconstruction et le sous-échantillonnage par prototypes.

Sous-échantillonnage par erreurs de reconstruction

Comme nous l'avons mentionné plus tôt, nos données sont très bruitées, ce qui crée des échantillons que l'on pourrait considérer comme aberrants. Ces échantillons pourraient nuire aux performances de nos modèles s'ils sont pris en compte durant l'entraînement. En effet, le bruit devient encore plus problématique dans notre cas en raison du grand nombre de dimensions des données et du faible nombre d'échantillons. Les modèles peuvent donc plus facilement faire face au surapprentissage et apprendre de fausses relations entre les intensités de protéines et la maladie de Parkinson qui seraient apparues par hasard au travers du bruit. La méthode présentée a pour objectif d'identifier ces échantillons aberrants afin de les retirer de l'ensemble de données d'entraînement. Plus précisément, nous profitons du fait que nous pouvons exclure les échantillons potentiellement aberrants pour effectuer une augmentation des données. Cette étape devrait permettre de réduire le surapprentissage, puisqu'elle fournit un plus grand nombre d'échantillons et que notre méthode de sous-échantillonnage permet de retirer les échantillons synthétiques de mauvaise qualité.

L'approche consiste à utiliser un premier modèle pour faire de l'augmentation de données, suivi d'un second pour retirer les échantillons aberrants. Nous commençons d'abord par créer

un mélange d'experts de VAE, comme présenté dans l'article de Leelarathna et al. (2023) [90]. Ces modèles sont décrits dans la section des modèles de prédiction et permettent de modéliser la distribution des données dans un espace latent de plus petite dimension à l'aide d'un réseau de neurones autoencodeur modifié. Dans le cas du mélange d'experts, l'architecture permet de diviser l'entraînement en plusieurs petits modèles de VAE et de combiner leurs prédictions dans l'espace latent. Cette division permet de réduire le nombre total de poids du modèle, ce qui peut atténuer le surapprentissage. On peut ensuite échantillonner dans l'espace latent et reconstruire les données complètes grâce à la partie décodeur du réseau. Notre méthode originale consiste ensuite à utiliser des autoencodeurs supervisés pour trouver des échantillons aberrants dans la nouvelle base de données augmentée. Ces réseaux ont la forme classique d'autoencodeur, qui sont présentés brièvement dans la section des modèles de prédictions, mais tentent aussi de prédire la variable de sortie, ce qui permet un espace latent plus riche pour la prédiction.

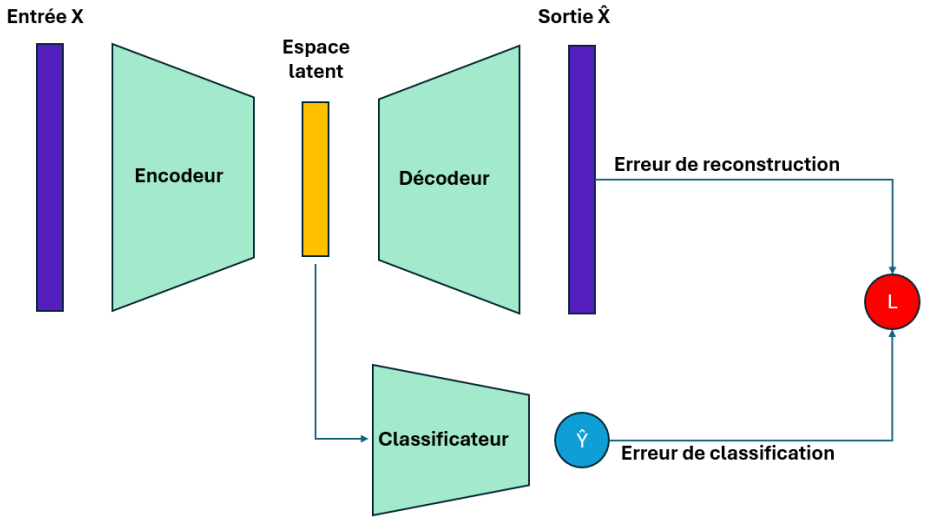

FIGURE 3.2 Schéma d'un autoencodeur supervisé.

Comme on le voit à la figure 3.2, les autoencodeurs supervisés sont séparés en deux parties : une pour la reconstruction, où l'on tente de reproduire les données d'entrée, et une pour la classification. La perte complète du modèle est donc donnée par l'addition de la perte de reconstruction et de classification. Cependant, pour que les deux soient bien équilibrées, on peut transformer les données d'entrée à l'aide d'une mise à l'échelle min-max pour permettre l'utilisation d'une fonction de perte d'entropie croisée au lieu d'une erreur quadratique moyenne. Cette perte est aussi généralement plus stable qu'une perte quadratique qui peut diverger plus facilement. On peut également utiliser un hyperparamètre β pour mettre plus

ou moins d'emphase sur l'erreur de classification, de sorte que l'erreur totale $\mathcal{L}$ soit donnée par :

$$\mathcal{L} = \left(-\sum_{i=1}^{N}\sum_{j=1}^{M} X_{ij}\log(\hat{X}_{i,j}) \right) + \beta \left(-\frac{1}{N}\sum_{i=1}^{N} \left(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i) \right) \right) \tag{3.21}$$

où $\hat{\mathbf{X}}$ et $\hat{\mathbf{y}}$ sont la matrice et le vecteur de prédictions du modèle. Pour détecter des échantillons potentiellement aberrants, nous utilisons la partie de gauche de l'équation portant sur la perte de reconstruction du modèle. En effet, si celle-ci est trop élevée en comparaison avec le reste des données d'entraînement, nous pouvons assumer que l'échantillon diverge des autres et qu'il est donc trop bruité. Cependant, lorsqu'on utilise des données complexes et bruitées comme les nôtres, un réseau peut apprendre différemment entre chaque entraînement simplement en raison de l'initialisation aléatoire des poids ou de son architecture. Il n'est donc pas fiable d'utiliser un seul réseau et un seul entraînement pour déterminer les échantillons aberrants.

Nous entraînons alors plutôt un ensemble de K autoencodeurs supervisés, initialisés avec des architectures différentes en modifiant aléatoirement la taille des couches cachées et celle de l'espace latent. Nous obtenons ainsi un vecteur de longueur K pour chaque échantillon de l'ensemble d'entraînement, contenant les valeurs d'erreur de reconstruction des modèles. Nous pouvons ensuite appliquer différentes méthodes de détection d'échantillons aberrants adaptées aux vecteurs. Dans notre cas, nous avons décidé d'utiliser la méthode basée sur le score Z, initialement prévue pour des scalaires, mais ici modifiée pour des vecteurs. Pour ce faire, nous calculons la distance de Mahalanobis pour chaque échantillon. Celle-ci permet de comparer les vecteurs d'erreur de reconstruction au reste de la distribution en utilisant les moyennes et les covariances. On peut ensuite comparer ces distances à une distribution χ^2, ce qui permet d'identifier les valeurs aberrantes en fonction d'un niveau de signification données par :

$$D_M(\mathbf{L}) = \sqrt{(\mathbf{L}-\boldsymbol{\mu}_L)^T \boldsymbol{\Sigma}_L^{-1} (\mathbf{L}-\boldsymbol{\mu}_L)} \tag{3.22}$$

et

$$Seuil = \chi^2_{\alpha,K} \tag{3.23}$$

où $\mathbf{L}$ est le vecteur de dimension K des erreurs de reconstruction, $\boldsymbol{\mu}_L$ est le vecteur de moyenne des erreurs par modèle, $\boldsymbol{\Sigma}_L$ est la matrice de covariance des erreur et α est un hyperparamètre pour notre modèle donnant le niveau de signification (autour de 0.05). Les

échantillons dont les distances de Mahalanobis sont supérieures au seuil sont donc retirés du jeu de données d'entraînement pour le reste des étapes. L'algorithme 2 détaille l'ensemble des étapes de notre approche. Cette méthode possède plusieurs hyperparamètres à optimiser par validation croisée, à savoir le nombre de modèles, les hyperparamètres d'entraînement d'un réseau de neurones classique, ainsi que la taille minimale et maximale de l'espace latent et de la couche cachée. Les structures des modèles sont ensuite générées aléatoirement à l'aide d'une distribution uniforme entre ces tailles minimales et maximales.

Algorithme 2 : Algorithme de sous-échantillonnage par erreur de reconstruction

Entrée :

- Ensemble de données $\mathbf{X}$ de taille $N \times M$
- Vecteur de classe à prédire $\mathbf{y}$ de longueur N
- Nombre K de modèles autoencodeurs supervisés
- Hyperparamètre β pour équilibrer la perte
- Niveau de signification α
- Dimension maximale de la couche cachée $DimMax_{cachée}$
- Dimension minimale de la couche cachée $DimMin_{cachée}$
- Dimension maximale de l'espace latent $DimMax_{latent}$
- Dimension minimale de l'espace latent $DimMin_{latent}$

Sortie : $\mathbf{X}_{épuré}$, $\mathbf{y}_{épuré}$

Pose $\mathbf{e}$, le vecteur d'erreurs de reconstruction

$\mathbf{X}_{normalisé} = \textbf{ÉchelleMinMax}(\mathbf{X})$

pour $k \leftarrow 1$ *à* K **faire**

$Dimension_{cachée} \sim \mathcal{U}(DimMin_{cachée}, DimMax_{cachée})$

$Dimension_{latent} \sim \mathcal{U}(DimMin_{latent}, DimMax_{latent})$

$\textbf{Modèle}_k =$

$\textbf{AutoEncodeurSupervisé}(\mathbf{X}_{normalisé}, \mathbf{y}, Dimension_{cachée}, Dimension_{latent}, \beta)$

$\hat{\mathbf{X}} = Modèle_k(\mathbf{X}_{normalisé})$

$\mathbf{e} \leftarrow \textbf{EntropieCroisée}(\mathbf{X}_{normalisé}, \hat{\mathbf{X}})$

fin

$\mathbf{d} = \textbf{Mahalanobis}(\mathbf{e})$

$seuil = \chi^2_{\alpha,K}$

$index_{correct} = \mathbf{d} < seuil$

$\mathbf{X}_{épuré}, \mathbf{y}_{épuré} = \mathbf{X}[index_{correct}], \mathbf{y}[index_{correct}]$

Sous-échantillonnage par prototypes

Notre deuxième méthode de sous-échantillonnage adopte une approche totalement différente pour traiter un autre type de problème. En effet, dans la méthode basée sur l'erreur de reconstruction, nous cherchions à éliminer les échantillons potentiellement aberrants, qui seraient trop bruités et pourraient alors nuire aux performances des modèles durant l'entraînement. Dans cette nouvelle méthode, nous visons plutôt à améliorer l'interprétabilité de nos modèles afin de mieux comprendre l'effet des protéines sur la maladie et d'identifier des biomarqueurs potentiels.

Au lieu de chercher des échantillons aberrants, nous cherchons plutôt à identifier des échantillons « prototypes ». Cette approche n'est pas nouvelle dans la littérature, elle consiste à trouver des échantillons ou des combinaisons d'échantillons qui permettent de représenter les distributions de leur classes respectives sans avoir à utiliser l'ensemble de données complet. Il devient alors plus facile de comprendre les décisions des modèles en analysant individuellement les échantillons utilisés. Ces échantillons seraient ceux qui représentent le mieux leur classe respective tout en maximisant leur différence avec la classe opposée.

Cette approche offrirait une bien meilleure interprétabilité, notamment en médecine, car on pourrait étudier les patients prototypes en profondeur pour mieux comprendre pourquoi ils représentent bien leur classe. Nous tentons ici une méthode différente pour obtenir deux sous-ensembles d'échantillons : un groupe pour la classe contrôle et un pour la classe parkinsonienne. Pour ce faire, nous cherchons à optimiser dans l'espace $\mathbf{X}$ de nos variables la distance inter-classe tout en minimisant la distance intra-classe des deux sous-groupes définie par :

$$\max_{\mathcal{S}_1,\mathcal{S}_2} \left(- \left(\frac{1}{|\mathcal{S}_1|} \sum_{\mathbf{x}_i,\mathbf{x}_j \in \mathcal{S}_1} d(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{|\mathcal{S}_2|} \sum_{\mathbf{x}_i,\mathbf{x}_j \in \mathcal{S}_2} d(\mathbf{x}_i, \mathbf{x}_j) \right) + \beta \cdot \frac{1}{|\mathcal{S}_1||\mathcal{S}_2|} \sum_{\mathbf{x}_i \in \mathcal{S}_1, \mathbf{x}_j \in \mathcal{S}_2} d(\mathbf{x}_i, \mathbf{x}_j) \right) \tag{3.24}$$

où $\mathcal{S}_1$ et $\mathcal{S}_2$ sont les sous-groupes des deux classes, $d(\mathbf{x}_i, \mathbf{x}_j)$ est la distance entre les vecteurs échantillons $\mathbf{x}_i$ et $\mathbf{x}_j$, et β est un hyperparamètre permettant d'équilibrer les distances intra-classe et inter-classe. En théorie, maximiser cette fonction objectif nous permettrait d'obtenir deux groupes bien distincts, chacun étant constitué uniquement d'échantillons représentant bien sa classe respective. En effet, maximiser la distance inter-classe permettrait de choisir les échantillons expliquant les différences entre les deux classes, tandis que minimiser la distance intra-classe garderait les échantillons qui représentent bien leur classe respective.

La métrique de distance est aussi un paramètre important à choisir. En effet, il s'avère que la

métrique euclidienne classique devient peu efficace lorsque l'espace est de grande dimension, comme dans notre étude. Deux approches peuvent donc s'appliquer : on peut réduire l'espace davantage à l'aide de méthodes de sélection de variables et de réduction de dimension, ou on peut utiliser une métrique différente. La première approche peut cependant nous faire perdre de l'information et rendre l'inférence difficile si l'on utilise des méthodes de réduction de dimension, ce qui contredit le but initial de cette méthode. Dans le cadre de notre étude, nous avons alors décidé d'utiliser une métrique différente, plus précisément un indice de dissimilarité utilisé par Modarres (2022) [88]. Cet indice se trouve à être plus efficace sur les tâches de classification utilisant des méthodes avec des métriques de distance comme les KNN lorsque les données sont de haute dimension. L'indice $\rho_0(\mathbf{x_1}, \mathbf{x_2})$ permet de quantifier la dissimilarité entre deux échantillons en utilisant la base de données complète et la différence absolue moyenne des distances. Il est donné par :

$$\rho_0(\mathbf{x_1}, \mathbf{x_2}) = \frac{1}{N-2}\frac{1}{\sqrt{M}} \sum_{\mathbf{w} \in \mathbb{W}_N \setminus \{\mathbf{x_1}, \mathbf{x_2}\}} |\|\mathbf{x_1} - \mathbf{w}\| - \|\mathbf{x_2} - \mathbf{w}\||. \tag{3.25}$$

où $\mathbb{W}_N$ est l'ensemble des données, $\mathbf{w}$ sont les vecteurs cet ensemble, et $\mathbf{x_1}$ et $\mathbf{x_2}$ les vecteurs d'intérêt. Cet indice est alors utilisé dans notre fonction objectif comme valeur de distance entre les échantillons.

Identifier les meilleurs sous-ensembles permettant de maximiser notre fonction objectif est un problème combinatoire, qui peut être résolu en théorie en calculant la fonction objectif pour toutes les combinaisons de nos échantillons. Cependant, cela est bien trop long, et il nous faut alors utiliser un algorithme d'optimisation. Plusieurs peuvent être mis en œuvre, comme l'essaim de particules, le recuit simulé ou l'algorithme génétique. Nous avons plutôt opté pour ce dernier en raison de notre grand nombre de combinaisons d'échantillons possibles. En effet, l'algorithme génétique n'est pas le plus rapide des trois, mais il a une capacité d'exploration de l'espace objectif plus grande que les autres, ce qui s'adapte bien à notre problème. Il utilise aussi plusieurs hyperparamètres permettant de jouer avec la convergence du modèle, comme le taux de mutation et la taille des populations, ce qui nous permet d'optimiser ses performances plus facilement.

L'algorithme génétique 3 se base sur la création de populations composées de différentes solutions possibles au problème. Dans notre cas, ces solutions sont composées de deux ensembles d'échantillons représentant leurs classes respectives. Ces populations sont ensuite testées sur la fonction objectif, et seulement les meilleures sont gardées pour la prochaine itération. Les autres sont remplacés par des enfants de la population en combinant aléatoirement des parents. Une mutation aléatoire est ensuite effectuée pour modifier les solutions et espérer sortir

des minimums locaux. On obtient alors une nouvelle population et on répète ces étapes pour le nombre de générations choisi. À la fin de ces générations, on choisit la solution optimale de la dernière population selon l'équation 3.24. Dans notre cas, le but est d'obtenir les deux groupes $\mathcal{S}_1$ et $\mathcal{S}_2$ contenant les K échantillons prototypes pour chaque classe maximisant la fonction objectif. Ces sous-groupes peuvent ensuite être utilisés pour l'entraînement d'un modèle quelconque et permettraient une interprétation bien plus simple.

Algorithme 3 : Algorithme Génétique pour Maximiser la Fonction Objectif

Entrée :

- Ensemble de données $\mathcal{D}$
- Taille α de la population
- Taux ϵ d'élitisme
- Taux γ de mutation
- Nombre δ de génération
- Hyperparamètre d'équilibrage β
- Nombre K d'échantillons par groupe

Sortie : Solution ensemble $\mathcal{S}$

pour $i \leftarrow 1$ *à* δ **faire**

  si *i = 1* **alors**

    $Pop = \textbf{GénérerSolutionAléatoire}(\mathcal{D}, \alpha, K)$

  fin

  pour $\mathcal{S}$ *dans Pop* **faire**

    $score \leftarrow \textbf{objectif}(\mathcal{D}, \mathcal{S}, \beta)$

  fin

  Nombre d'élitisme $ne = \alpha \cdot \epsilon$;

  $Pop_1 = \textbf{MeilleuresSolutions}(score, Pop, ne)$

  Nombre de croisements $nc = (\alpha - ne)/2$;

  pour $j \leftarrow 1$ *à* nc **faire**

    $\mathcal{S}_A, \mathcal{S}_B = \textbf{SélectionAléatoire}(Pop)$

    $\mathcal{S}_C, \mathcal{S}_D = \textbf{Croisement}(\mathcal{S}_C, \mathcal{S}_D)$

    $Pop_2 \leftarrow \mathcal{S}_C, \mathcal{S}_D$

  fin

  pour $\mathcal{S}$ *dans Pop* **faire**

    $\mathcal{S}' = \textbf{Mutation}(\mathcal{S}, \gamma)$

    $Pop_2 \leftarrow \mathcal{S}'$

  fin

  $Pop = Pop_1 + Pop_2$;

fin

pour $\mathcal{S}$ *dans Pop* **faire**

  $score \leftarrow \textbf{objectif}(\mathcal{D}, \mathcal{S}, \beta)$

fin

$\mathcal{S} = \textbf{MeilleureSolution}(score, Pop)$

3.2.7 Modèles de prédictions

Cette sous-section présente brièvement les différentes méthodes de classification utilisées dans notre étude. Certaines de ces méthodes peuvent également s'appliquer à la régression en adaptant les modèles, et elles seront brièvement abordées. Toutefois, notre objectif est de distinguer les patients contrôles des patients parkinsoniens, c'est pourquoi nous utilisons principalement des méthodes de classification. Plus précisément, nous utilisons ces méthodes dans un cas de classification binaire ; nous cherchons donc à prédire un scalaire représentant la probabilité d'appartenir à la classe 1, soit la classe parkinsonienne.

Dans cette recherche, nous comparons sept modèles de classification : la régression logistique, la MVS, la LDA, la forêt aléatoire, le XGBoost, le RNA et notre approche originale, qui seront tous détaillés dans cette section. Nous aborderons également brièvement les méthodes KNN et VAE, car elles sont utilisées dans d'autres parties de notre approche. Tous ces modèles, à l'exception de la LDA et du VAE, sont des modèles discriminants, où l'on modélise $P(y|\mathbf{x})$ en posant une forme à la distribution de probabilité. Cependant, LDA et VAE sont des modèles génératifs, car ils modélisent la génération des données en calculant la probabilité conjointe $P(y, \mathbf{x})$ en faisant des hypothèses sur la forme de $P(\mathbf{x}|y)$ et de $P(y)$. Cette propriété a été exploitée pour générer de nouveaux échantillons avant le sous-échantillonnage dans nos méthodes originales, afin d'éliminer les échantillons aberrants.

Régression logistique

La régression logistique est un modèle simple basé sur la régression linéaire où l'on force la valeur prédite à être dans l'intervalle [0,1]. Cela permet d'utiliser la prédiction comme une probabilité d'appartenir à la classe 1. On utilise donc une transformation logit pour obtenir la prédiction $p(\mathbf{x})$ donnée par :

$$p(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{x}^\top \boldsymbol{\beta}}}. \tag{3.26}$$

Les paramètres $\boldsymbol{\beta}$ sont optimisés en utilisant la méthode du maximum de vraisemblance $\mathcal{L}(\boldsymbol{\beta})$ donnée par :

$$\mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^{N} \left[y_i \log p(\mathbf{x}_i) + (1 - y_i) \log(1 - p(\mathbf{x}_i)) \right], \tag{3.27}$$

où y_i est la classe de l'échantillon i et $p(\mathbf{x}_i)$ la probabilité, prédite par le modèle, que l'échantillon i appartienne à la classe 1. Les gradients sont ensuite définis par :

$$\nabla \mathcal{L}(\boldsymbol{\beta}) = \mathbf{X}^\top (\mathbf{y} - p(\mathbf{X})) \tag{3.28}$$

où $\mathbf{X}$ est la matrice d'entrée de tous les échantillons et $\mathbf{y}$ le vecteur des classes de tous les échantillons. La maximisation de la vraisemblance en fonction des paramètres peut être résolue en utilisant des méthodes d'optimisation comme Newton, quasi-Newton ou la descente de gradient.

Sachant que nous possédons un très grand nombre d'intensités de protéines comparé au nombre d'échantillons, le modèle peut faire face à du surapprentissage et avoir une grande variance. Nous ajoutons donc une régularisation pour réduire ces problèmes. La régularisation Ridge (L2) consiste à ajouter une pénalité proportionnelle au carré des poids, ce qui réduit les valeurs trop élevées sans les forcer à être nulles. On peut ainsi formuler l'équation de la vraisemblance comme suit :

$$\mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^{N} [y_i \log p(\mathbf{x}_i; \boldsymbol{\beta}) + (1 - y_i) \log(1 - p(\mathbf{x}_i; \boldsymbol{\beta}))] - \frac{\lambda}{2} \sum_{j=1}^{M} \beta_j^2 \tag{3.29}$$

où λ est un hyperparamètre à optimiser pour ajuster la force de la régularisation. Il existe également la régularisation Lasso (L1), qui est plus efficace pour réduire le nombre de variables. En effet, elle permet de tendre les poids des variables non significatives vers 0, grâce à l'utilisation de la valeur absolue des poids. Le terme de pénalisation proportionnel à la valeur absolue des poids pousse ces derniers plus facilement vers 0 que la pénalisation L2. Cette vraisemblance est donnée par :

$$\mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^{N} [y_i \log p(\mathbf{x}_i; \boldsymbol{\beta}) + (1 - y_i) \log(1 - p(\mathbf{x}_i; \boldsymbol{\beta}))] - \lambda \sum_{j=1}^{M} |\beta_j| \tag{3.30}$$

Cette méthode est souvent utilisée pour la sélection de variables, car elle permet d'éliminer les variables dont les poids sont nuls, produisant ainsi un jeu de données plus épuré. Il est également possible de combiner les régularisations L1 et L2, ce que l'on appelle la régression logistique «Elastic Net», qui cumule leurs avantages respectifs. La vraisemblance est écrite sous la forme de :

$$\mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^{N} [y_i \log p(\mathbf{x}_i; \boldsymbol{\beta}) + (1 - y_i) \log(1 - p(\mathbf{x}_i; \boldsymbol{\beta}))] - \lambda_1 \sum_{j=1}^{M} |\beta_j| - \frac{\lambda_2}{2} \sum_{j=1}^{M} \beta_j^2 \tag{3.31}$$

où λ_1 et λ_2 sont des hyperparamètres pour modifier la force des deux types de régularisation.

Pour notre recherche, la régression logistique permet de créer un modèle très simple, mais souvent très efficace. Elle offre des solutions rapides et facilement interprétables, ce qui est intéressant pour identifier des biomarqueurs potentiels. La régularisation permet également de complexifier légèrement le modèle et de réduire le surapprentissage qui pourrait apparaître en raison du grand nombre de dimensions. Cependant, elle reste un modèle linéaire et ne permet donc pas de modéliser des relations complexes entre les variables. De plus, elle est assez sensible aux valeurs aberrantes et nécessite donc un prétraitement.

Machine à vecteurs de support (MVS)

L'idée derrière la MVS est de trouver un hyperplan permettant de séparer les échantillons des deux classes du mieux possible, tout en maximisant la distance entre l'hyperplan et les échantillons les plus proches, ceux qu'on appelle les vecteurs de support. Le modèle cherche alors à optimiser le vecteur de poids $\mathbf{w}$ et le biais b de l'hyperplan donné par :

$$\mathbf{w}^T\mathbf{x} + b = 0. \tag{3.32}$$

Ce qui revient à minimiser les poids donnant l'hyperplan tel que :

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2, \tag{3.33}$$

tout en étant soumis à la contrainte de classification qui est donnée par :

$$y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 \quad \forall i. \tag{3.34}$$

Cependant, ce problème d'optimisation présente un inconvénient : il repose sur l'hypothèse que les données sont linéairement séparables, c'est-à-dire qu'il existe un hyperplan permettant de parfaitement séparer les échantillons des deux classes. Cette hypothèse est rarement vraie lorsque l'on travaille avec des données complexes comme les nôtres, puisqu'il y a de la confusion entre les classes, ce qui rend le problème non linéaire.

Deux solutions s'offrent à nous : utiliser l'astuce du noyau et modifier la marge pour être souple. Dans le premier cas, on applique une transformation permettant de représenter les données dans un espace de dimension plus grande, où il existe un hyperplan permettant de séparer les échantillons linéairement. Un exemple souvent utilisé de noyau entre deux échantillons $K(\mathbf{x}_i, \mathbf{x}_j)$ est la fonction de base radiale ci-dessous :

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma|\mathbf{x}_i - \mathbf{x}_j|^2\right) \tag{3.35}$$

où γ est un hyperparamètre déterminant la largeur du noyau.

Dans le deuxième cas, la marge souple permet d'avoir des points entre l'hyperplan et la marge qui ne sont pas du bon côté de l'hyperplan et donc mal classifiés. Cependant, une erreur est ajoutée plus la distance entre l'hyperplan et l'échantillon mal classifié est grande. La nouvelle équation à optimiser est définie comme suit :

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}|\mathbf{w}|^2 + C\sum_{i=1}^{N}\xi_i \quad \text{sujet à} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i \tag{3.36}$$

où C est un hyperparamètre permettant d'équilibrer l'erreur de classification et la marge, et $\xi_i = 0$ si le point est bien classé, sinon $\xi_i = |y_i - \mathbf{w}^T\mathbf{x}_i + b|$.

On peut combiner ces deux méthodes et obtenir la solution à l'aide du problème dual d'une formulation de Lagrange $\mathcal{L}(\boldsymbol{\alpha})$ donnée par :

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad \text{sujet à} \quad 0 \leq \alpha_i \leq C \quad \text{et} \quad \sum_{i=1}^{N}\alpha_i y_i = 0 \tag{3.37}$$

où $\boldsymbol{\alpha}$ est le vecteur des multiplicateurs de Lagrange qui se trouve en posant que les dérivées sont égales à 0. On peut finalement faire une prédiction $\hat{y}(\mathbf{x})$ en utilisant les solutions trouvées pour les $\boldsymbol{\alpha}$ et b du problème dual tel que :

$$\hat{y}(\mathbf{x}) = \text{signe}\left(\sum_{i=1}^{N}\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right). \tag{3.38}$$

Comme nous avons pu le voir dans la revue de la littérature, la MVS est une méthode très utilisée dans le domaine. En effet, celle-ci est très efficace pour des espaces de grande dimension puisqu'elle permet de trouver un hyperplan séparant les classes avec une grande marge de manière plus aisée. Elle nécessite également peu d'échantillons, car on cherche uniquement les vecteurs de support pour faire la prédiction. Cette méthode est donc très adéquate pour les données de haute dimension avec peu d'échantillons. Elle s'applique donc bien à notre recherche. En utilisant des noyaux, on peut également modifier l'espace pour créer des frontières de décision non linéaires, ce qui permet de résoudre des problèmes plus complexes comme le nôtre. Cependant, le choix de ces noyaux et des hyperparamètres est très sensible et peut énormément affecter les résultats. Il peut donc être difficile d'optimiser ces

choix. C'est pourquoi, nous testons plusieurs hyperparamètres pour cette méthode à l'aide de la validation croisée. La MVS est aussi sensible aux données où les classes sont difficilement distinguables, car il est alors difficile de trouver un hyperplan avec une marge suffisamment large.

Analyse par discriminant linéaire

L'analyse par discriminant est une méthode générative, comme mentionné plus tôt. Elle utilise donc des hypothèses sur la distribution des données. On doit alors supposer que les données d'intensités sont distribuées selon des lois normales et que les deux classes possèdent toutes la même matrice de covariance. Pour faire la prédiction, on calcule des fonctions discriminantes à partir des hypothèses et de la loi de Bayes. La probabilité de $\mathbf{x}$ conditionnelle à la classe k est donnée par :

$$p(\mathbf{x} \mid y = k) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^\top \mathbf{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right) \tag{3.39}$$

où $\mathbf{\Sigma}$ est la matrice de covariance des données $\mathbf{X}$ et $\boldsymbol{\mu}_k$ le vecteur des moyennes pour la classe k. À l'aide de la loi de Bayes, on calcule la probabilité conditionnelle à $\mathbf{x}$ d'être dans la classe k donnée par :

$$P(y = k \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid y = k)\pi_k}{p(\mathbf{x})} \tag{3.40}$$

où π_k est la distribution a priori de la classe k. Il s'avère que l'on peut ignorer $p(\mathbf{x})$ puisqu'elle est constante pour toutes les classes et en utilisant le logarithme faire de même pour deux autres termes. On obtient finalement la fonction discriminante $\delta_k(\mathbf{x})$ définie par :

$$\log P(y = k \mid \mathbf{x}) = \delta_k(\mathbf{x}) = \mathbf{x}^\top \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^\top \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_k + \log \pi_k. \tag{3.41}$$

On peut ensuite faire la prédiction de la classe pour l'échantillon en choisissant la classe ayant la valeur discriminante la plus élevée telle que :

$$\hat{y} = \arg\max_k \delta_k(\mathbf{x}). \tag{3.42}$$

Ces modèles sont donc très simples, ce qui leur permet d'être interprétables. Ils ont également une faible variance, ce qui réduit le surapprentissage. Cependant, leur simplicité pose aussi problème, car ils sont très sensibles aux données aberrantes et instables face aux données corrélées. Les hypothèses de distribution sont également très restreintes, ce qui ajoute un biais

aux prédictions et peut fausser les résultats. Pour notre étude, cette méthode est utilisée pour illustrer les problèmes qui surviennent lorsqu'on applique des modèles simples à des données complexes. Nous ne nous attendons donc pas à obtenir de bons résultats en comparaison aux autres méthodes.

KNN

La méthode des KNN est très simple, et elle peut être utilisée pour la régression ou la classification. Pour ce faire, on doit choisir une métrique de distance, généralement la distance euclidienne, ainsi qu'un entier K. Ces deux choix sont des hyperparamètres. Par la suite, on prédit la classe d'un nouvel échantillon en comptant le nombre d'échantillons de chaque classe dans les K plus proches voisins. La classe possédant le plus d'échantillons est ainsi celle prédite par le modèle. Il n'y a donc aucun entraînement pour ce modèle ; on calcule directement la prédiction pour chaque échantillon donné en utilisant la base de données. Pour la classification, la prédiction est donnée par :

$$\hat{y} = \arg\max_{y} \sum_{i \in \mathcal{N}_K} \mathbf{1}(y_i = y) \tag{3.43}$$

où $\mathcal{N}_K$ est l'ensemble des K plus proches voisins et $\mathbf{1}(y_i = y)$ est une fonction indicatrice qui égale 1 si $y_i = y$ et 0 sinon. Pour faire de la régression, on fait la moyenne de y pour les plus proches voisins. La prédiction est donc calculée comme suit :

$$\hat{y} = \frac{1}{k} \sum_{i \in \mathcal{N}_K} y_i. \tag{3.44}$$

La méthode des KNN est rapide à utiliser puisqu'elle ne nécessite aucun entraînement. Elle ne fait aucune hypothèse sur les données et est non paramétrique, ce qui lui permet d'être flexible dans différents types de problèmes. Cependant, elle devient peu efficace lorsque les données sont en haute dimension, car les distances deviennent alors peu discriminantes. Cette méthode n'est donc pas utilisée dans notre recherche pour effectuer la classification des patients, mais elle est employée dans différentes étapes comme l'imputation et l'équilibrage des classes, ce qui justifie sa présentation.

Forêt aléatoire

La forêt aléatoire fait partie des méthodes d'ensembles, un type de modèle qui combine la puissance de plusieurs modèles simples pour améliorer la qualité des prédictions. Cette ap-

proche est très efficace pour notre recherche puisqu'elle permet de réduire le surapprentissage et l'effet des données bruitées. En effet, puisqu'à la base elle utilise des modèles faibles, les grandes dimensions de nos données impactent moins le surapprentissage. De plus, le mécanisme de moyennage sur plusieurs modèles permet de réduire la variance et donc l'impact du bruit dans les données.

Pour les forêts aléatoires, les modèles simples utilisés sont les arbres de décision. Ces arbres sont des organigrammes formés d'une suite de règles de décision qui permettent de séparer consécutivement les échantillons selon leurs caractéristiques jusqu'à obtenir des sous-groupes composés de classes uniques. On peut, par exemple, séparer les échantillons en fonction de l'intensité d'une protéine potentiellement liée à la maladie de Parkinson, si elle possède une valeur supérieure à une constante donnée.

Pour choisir la variable et la condition pour le test de séparation, il existe différentes métriques d'information qui sont choisies comme hyperparamètres. La première métrique testée pour notre étude est celle de l'entropie, qui mesure la quantité d'incertitude dans le nœud, donnée par :

$$\text{Entropie}(t) = -\sum_{k=1}^{K} p_k \log(p_k) \tag{3.45}$$

où p_k est la proportion des échantillons appartenant à la classe k dans le nœud en question. La deuxième métrique est le coefficient de Gini d'impureté, qui mesure l'homogénéité des échantillons dans le nœud et est calculé comme suit :

$$\text{Gini}(t) = 1 - \sum_{k=1}^{K} p_k^2. \tag{3.46}$$

On peut aussi mesurer le gain d'information apporté par la séparation à l'aide de l'entropie ou du coefficient de Gini comme mesure d'impureté donné par :

$$\text{Gain}(\mathcal{S}) = \text{Impureté}(\mathcal{S}) - \sum_{v \in V} \frac{|\mathcal{S}_v|}{|\mathcal{S}|} \text{Impureté}(\mathcal{S}_v) \tag{3.47}$$

où $\mathcal{S}$ est l'ensemble avant la séparation et $\mathcal{S}_v$ sont les sous-ensembles après la séparation. Pour notre étude, le gain d'impureté est aussi utilisé pour quantifier l'impact de différentes variables sur la prédiction dans le modèle. Cette étape permet de faire de la sélection de variables comme nous l'avons vu auparavant.

Il faut ensuite décider quand arrêter d'effectuer des séparations, car il est possible de séparer

jusqu'à ce que chaque feuille ne possède qu'un seul échantillon, ce qui est inutile pour la tâche de généralisation. Le modèle peut donc facilement surapprendre sur le bruit ou des coïncidences dans les données d'entraînement. Deux méthodes s'offrent à nous pour résoudre ce problème : on peut arrêter la croissance d'un arbre à un moment donné ou on peut le laisser croître et, par la suite, l'élaguer. La première méthode consiste à utiliser des hyperparamètres comme la profondeur maximale de l'arbre, le nombre minimum d'échantillons par nœud ou le nombre minimum d'échantillons pour diviser un nœud. Cette méthode est cependant instable en raison du choix des hyperparamètres, qui peut s'avérer complexe.

La méthode par élagage consiste à laisser l'arbre grandir, puis à couper les branches qui ne réduisent pas les performances de l'arbre sur un sous-ensemble de validation. Une alternative est l'utilisation du coût-complexité, qui combine une erreur en fonction de la taille de l'arbre et l'erreur de classification sur les données d'entraînement.

À l'aide des arbres de décision, il est possible de créer des forêts aléatoires en combinant les résultats de nombreux arbres. Ces arbres ne sont toutefois pas tous créés de la même manière. En effet, pour les rendre uniques, on utilise le principe de « bagging », qui consiste à entraîner chaque arbre sur un nouvel ensemble de données formé en échantillonnant au hasard avec remise dans l'ensemble original, ce qui crée des doublons. Cette étape n'est cependant pas suffisante pour rendre les arbres différents et réduire leur corrélation. L'algorithme comporte donc une autre étape de « bagging », cette fois-ci sur les variables, en en choisissant m parmi les M de l'ensemble initial. Cette étape se fait à chaque nœud, ce qui augmente significativement la variance entre les arbres. Le nombre exact m est un hyperparamètre qui est souvent fixé à $\sqrt{M}$ ou $\log_2(M)$. Après l'entraînement des arbres individuels, la prédiction est faite en appliquant une moyenne pour la régression ou une majorité pour la classification sur les B arbres. Elle est alors donnée par :

$$\hat{y}(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} T_b(\mathbf{x}) \tag{3.48}$$

où $T_b(\mathbf{x})$ est la prédiction de l'arbre b.

Les modèles de forêts aléatoires s'adaptent très bien à notre problème. En effet, ils sont robustes aux données aberrantes et au bruit en raison du grand nombre d'arbres, ce qui permet de réduire la variance totale. Ils peuvent aussi être résistants au surapprentissage, mais cela peut impliquer une optimisation complexe des hyperparamètres. Ce sont des modèles non paramétriques permettant de modéliser des relations complexes entre les variables, ce qui est indispensable dans un problème comme le nôtre. Ils s'appliquent alors bien à la complexité de nos données pour cette étude. Les forêts aléatoires peuvent également fonctionner

avec des valeurs manquantes, ce qui explique leur utilisation fréquente dans les méthodes d'imputation. Néanmoins, en combinant un grand nombre d'arbres de décision, interpréter le modèle complet devient plus complexe. Par la suite, il est possible de calculer une métrique d'importance pour chaque variable d'entrée, permettant ainsi de faire de la sélection de variables.

XGBoost

XGBoost est un modèle de type « boosting » qui est un modèle d'ensemble, comme la forêt aléatoire. Toutefois, au lieu de simplement combiner la prédiction de nombreux modèles simples, ceux-ci s'entraînent de manière séquentielle en tentant d'améliorer la prédiction en fonction de l'erreur du modèle précédent. On appelle ces petits modèles des modèles faibles puisque leur performance individuelle est légèrement meilleure qu'une prédiction aléatoire. Ils possèdent donc une faible variance, ce qui leur permet d'être résistants au surapprentissage, aux bruits et aux données aberrantes.

Le modèle XGBoost utilise ce principe à l'aide d'arbres de décision. Plus précisément, la fonction objectif $\mathcal{L}$ cherche à réduire l'erreur de classification tout en ayant une erreur de régularisation pour réduire le surapprentissage. Elle est définie comme suit :

$$\mathcal{L} = \sum_{i=1}^{N}(y_i \log(p_i) + (1 - y_i) \log(1 - p_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{3.49}$$

où

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda ||\mathbf{w}_k||^2, \tag{3.50}$$

K est le nombre d'arbres, γ est un hyperparamètre contrôlant la complexité des arbres, T est le nombre de feuilles dans l'arbre, λ est un hyperparamètre pour contrôler la régularisation, $\mathbf{w}_k$ sont les vecteurs de poids, y_i les valeurs à prédire et $\log(p_i)$ les prédictions. Dans cet algorithme, les poids représentent les valeurs prédites par les feuilles de l'arbre. L'objectif est donc d'utiliser un premier arbre pour prédire les classes de nos échantillons, de calculer la fonction de perte pour obtenir les résidus, puis d'entraîner un nouvel arbre sur ces résidus. La prédiction au moment t dans l'entraînement est donc donnée par :

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{3.51}$$

où

$$f_t(x) = w_{q(x)}, \tag{3.52}$$

$f_t(x)$ est la nouvelle prédiction de l'arbre au temps t sur les résidus de l'arbre au temps $t-1$, w sont les poids des feuilles et $q(x)$ est la fonction qui assigne les échantillons à la feuille correspondante. En répétant ce processus pour T arbres, on obtient des modèles qui ne sont pas efficaces seuls, mais qui, lorsqu'ils sont combinés, deviennent très puissants. La prédiction finale est alors donnée par :

$$\hat{y}(x) = \sum_{t=1}^{T} f_t(x). \tag{3.53}$$

Cette méthode s'adapte bien à notre problème principalement grâce à sa structure de type « boosting ». En effet, elle est reconnue comme étant très performante grâce à son apprentissage séquentiel sur les résidus, ce qui permet d'accorder davantage de poids aux échantillons mal classés. Cela permet au modèle d'obtenir de très bonnes performances sur des tâches complexes, et l'utilisation d'arbres comme modèles faibles s'adapte bien à nos données de haute dimension. Cependant, la méthode est très sensible aux hyperparamètres et peut facilement entraîner un surapprentissage si la régularisation n'est pas suffisamment forte.

Réseau de neurones artificiels

Les RNA sont à la base de la grande majorité des modèles d'intelligence artificielle les plus complexes. Ces réseaux utilisent des couches de neurones artificiels connectés par des poids pour traiter l'information contenue dans les données jusqu'à une couche de sortie qui, dans notre cas, prédit la probabilité de l'échantillon d'être dans la classe 1. Le réseau complet est composé d'une multitude de combinaisons linéaires et de fonctions d'activation qui sont toutes dérivables, ce qui permet d'optimiser les poids en appliquant une descente de gradient à travers le réseau à partir de la fonction de perte. Ces réseaux apprennent à projeter les données dans de nouveaux espaces qui rendent la tâche de classification plus simple, ce qui est très utile pour le cas complexe de notre étude. Pour cela, l'entraînement se déroule en deux étapes : la propagation avant, où l'on calcule les valeurs de sortie, et la propagation arrière, où l'on calcule le gradient des paramètres et on les optimise avec une descente de gradient. La figure 3.3 présente un schéma d'un réseau de neurones artificiels général.

La propagation avant est l'étape où l'on transforme le vecteur d'entrée à travers le réseau pour obtenir le vecteur de prédiction en sortie. On peut calculer les vecteur de sortie de la couche l donnés par :

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)T}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)} \tag{3.54}$$

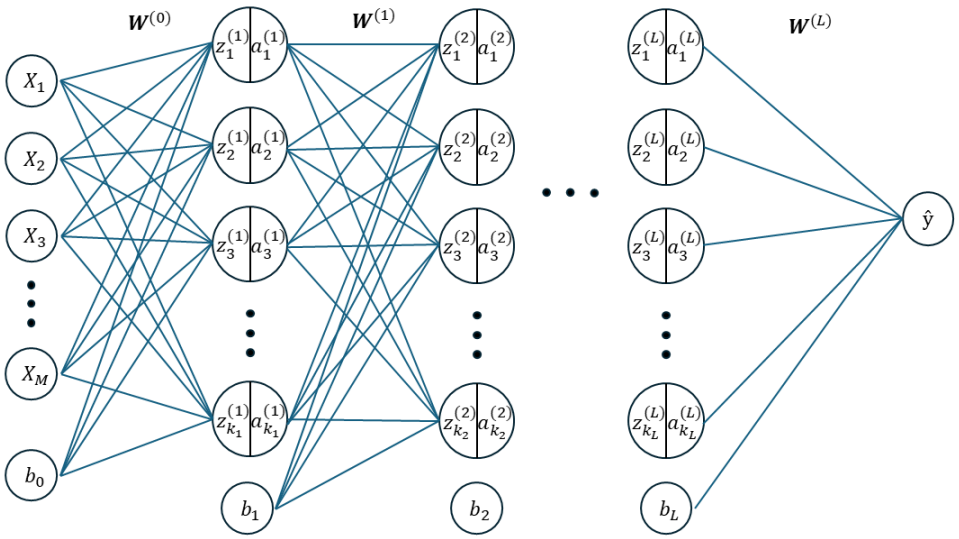

FIGURE 3.3 Schéma d'un réseau de neurones artificiels.

et

$$\mathbf{a}^{(l)} = g^{(l)}\left(\mathbf{z}^{(l)}\right) \tag{3.55}$$

où $\mathbf{z}^{(l)}$ et $\mathbf{a}^{(l)}$ sont les vecteurs de sortie avant et après l'activation de la couche l de dimension $k_l \times 1$, g est la fonction d'activation, $\mathbf{W}^{(l)}$ est la matrice de poids de dimension $k_{l-1} \times k_l$ entre le vecteur de la couche $l-1$ et la couche l, $\mathbf{b}^{(l)}$ est le vecteur de biais de dimension $k_l \times 1$ et k_l est le nombre de neurones de la couche l. L'équation complète du modèle peut alors s'écrire sous la forme :

$$\hat{\mathbf{y}} = g^{(L)}\left(\mathbf{W}^{(L)T}\left(g^{(L-1)}\left(\mathbf{W}^{(L-1)T}\left(\ldots g^{(1)}\left(\mathbf{W}^{(1)T}\mathbf{x} + \mathbf{b}^{(1)}\right)\ldots\right) + \mathbf{b}^{(L-1)}\right)\right) + \mathbf{b}^{(L)}\right). \tag{3.56}$$

Plusieurs fonctions d'activation sont utilisées dans la littérature. Les plus importantes utilisées dans notre recherche sont présentées dans le tableau 3.1.

Les valeurs de sortie étant maintenant calculées, il est possible de calculer l'erreur par rapport aux vraies valeurs de nos données et de procéder à la propagation arrière. Différentes fonctions de perte existent ; dans une tâche de classification binaire comme la nôtre, on utilise l'erreur d'entropie croisée binaire $\mathcal{L}$ donnée par :

TABLEAU 3.1 Différentes fonctions d'activation et leurs utilités

Fonction d'activation	**Équation**	**Utilité pour notre étude**
Sigmoïde	$\sigma(x) = \dfrac{1}{1+e^{-x}}$	Utilisée principalement pour la classification binaire en forçant la valeur entre 0 et 1 pour représenter une probabilité
Softmax	$f(\mathbf{x}) = \dfrac{e^{\mathbf{x}}}{\sum_k e^{x_k}}$	Utilisée dans la méthode de classification originale pour créer le réseau de porte nécessaire au mélange d'experts
«Leaky» ReLU	$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases}$ Où $\alpha < 1$	Utilisée dans les couches cachées pour réduire le problèmes d'évanescence du gradient dans les réseaux profonds

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right]. \tag{3.57}$$

En utilisant la règle de dérivation en chaîne, il est possible de calculer le gradient des poids et des biais d'un réseau à L couches très efficacement. En effet, on peut calculer ces gradients donnés par :

$$\frac{\partial \mathcal{L}}{\partial \mathbf{z}^{(L)}} = \frac{\partial \mathcal{L}}{\partial \mathbf{a}^{(L)}}\frac{\partial \mathbf{a}^{(\mathcal{L})}}{\partial \mathbf{z}^{(L)}} = \frac{\partial \mathcal{L}}{\partial \mathbf{a}^{(L)}} \odot g'^{(L)}(\mathbf{z}^{(L)}), \tag{3.58}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(L)}} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}^{(L)}}\frac{\partial \mathbf{z}^{(L)}}{\partial \mathbf{W}^{(L)}} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}^{(L)}}\mathbf{a}^{(L-1)T} \tag{3.59}$$

et

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(L)}} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}^{(L)}} \tag{3.60}$$

où $g'^{(L)}(\mathbf{z}^{(L)})$ est la dérivée de la fonction d'activation de la couche L et $\frac{\partial \mathcal{L}}{\partial \mathbf{a}^{(L)}}$ est la dérivée de la fonction d'erreur par rapport à la sortie du modèle ; elle dépend donc de la fonction de perte utilisée. Dans notre cas de classification avec la fonction de perte d'entropie croisée, cette dérivée est donnée par :

$$\frac{\partial \mathcal{L}}{\partial \mathbf{a}^{(L)}} = -\frac{\mathbf{y}}{\hat{\mathbf{y}}} + \frac{1-\mathbf{y}}{1-\hat{\mathbf{y}}}. \tag{3.61}$$

On peut faire de même pour une couche l quelconque dans le réseau à l'aide de la dérivation en chaîne.

Après avoir calculé les gradients, il ne reste plus qu'à mettre à jour les poids et les biais. Différentes méthodes de descente de gradient existent ; celle la plus couramment utilisée et qui est appliquée dans nos modèles est la méthode de « Adaptive Moment Estimation » [116]. Elle se base sur le calcul du premier et du second moment pour optimiser la descente de gradient en ralentissant ou en accélérant la descente en fonction des anciens gradients.

Les réseaux de neurones sont utilisés dans notre étude pour leur aptitude à générer des relations complexes entre les données grâce à leur capacité à apprendre des projections dans des espaces différents. Ils peuvent cependant être très complexes à optimiser en raison du grand nombre d'hyperparamètres, que ce soit par le choix de l'architecture ou des hyperparamètres des méthodes d'optimisation. Les réseaux de neurones partagent aussi la même caractéristique que les autres modèles complexes, où leur grand nombre de paramètres à optimiser crée facilement du surapprentissage, ce qui est très problématique pour nous en raison de notre faible nombre d'échantillons. Plusieurs méthodes de régularisation seront utilisées dans nos approches, comme Ridge, Lasso ou l'abandon (« dropout »), une méthode où l'on ignore certains neurones au hasard durant l'entraînement. Ces méthodes ne sont toutefois pas infaillibles, et les réseaux peuvent continuer à faire face au surapprentissage. Ils sont aussi considérés comme des boîtes noires ; il est donc très complexe de les interpréter pour comprendre le processus de décision. Pour notre étude, nous exploitons la polyvalence de leur architecture pour créer notre méthode de classification originale, basée sur les interactions protéine-protéine, afin de réduire le nombre total de poids du modèle et d'augmenter son interprétabilité.

Autoencodeur variationnel

Les réseaux de neurones peuvent permettre de créer toute sorte d'architectures permettant de s'attaquer à différentes tâches. Une de ces structures sont les autoencodeurs, des réseaux séparés en deux parties : un encodeur ayant une forme d'entonnoir et un décodeur ayant la forme inverse. L'entrée et la sortie de ces réseaux sont les mêmes, soit $\mathbf{X}$, ce sont donc des réseaux non supervisés qui visent à reconstruire les données en entrée. Pour ce faire, la première partie encode les données dans un espace latent plus petit que l'espace original et la deuxième partie décode cet espace pour retrouver les données en entrée. En appliquant ces transformations, les réseaux apprennent à retenir seulement les relations les plus importantes dans les données, ce qui permet de réduire le bruit et les variables inutiles. Cette approche peut donc être très efficace dans notre étude, sachant que nos données sont bruitées et

contiennent de nombreuses intensités de protéines sans lien avec la maladie.

Pour optimiser ces modèles, on utilise une perte de reconstruction, qui est généralement une perte d'erreur quadratique moyenne, ou une perte d'entropie croisée lorsque les valeurs sont entre 0 et 1. Cependant, il existe un modèle plus poussé qui permet de modéliser la distribution des données à travers l'espace latent. Cette architecture est l'autoencodeur variationnel. Le but de ce modèle est donc de trouver la distribution $p(\mathbf{x})$ donnée par :

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \tag{3.62}$$

où $\mathbf{z}$ est le vecteur dans l'espace latent. Cette intégrale est toutefois trop complexe à calculer directement. Pour régler ce problème, on utilise une méthode variationnelle pour approximer la probabilité a posteriori $p(\mathbf{x}|\mathbf{z})$ à l'aide de notre modèle $q(\mathbf{x}|\mathbf{z})$. Afin d'atteindre la meilleure approximation, la méthode se base sur la maximisation de la fonction borne inférieure de l'évidence et utilise la fonction de perte définie comme suit :

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[-\log p(\mathbf{x}|\mathbf{z})\right] + \mathrm{KL}\left(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\right) \tag{3.63}$$

où KL représente la divergence de Kullback–Leibler, un terme de régularisation qui assure que la distribution latente du modèle reste près de la distribution a priori de $\mathbf{z}$, et d est la taille de l'espace latent. Le premier terme représente la perte de reconstruction. On pose généralement que la distribution a priori de $p(\mathbf{z})$ et a posteriori $q(\mathbf{x}|\mathbf{z})$ suivent des lois gaussiennes.

La propagation arrière ne peut cependant pas être effectuée en échantillonnant directement l'espace latent, car cela empêche la propagation du gradient. Pour contourner cette limitation, les VAE utilisent la technique de reparamétrisation. Cette méthode exprime $\mathbf{z}$ selon une fonction déterministe avec un ajout de bruit gaussien donné par :

$$\mathbf{z} = \mu(\mathbf{x}) + \sigma(\mathbf{x}) \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}) \tag{3.64}$$

où $\mu(\mathbf{x})$ et $\sigma(\mathbf{x})$ sont les paramètres maintenant différentiables que le modèle va calculer dans l'espace latent, $\boldsymbol{\epsilon}$ est le vecteur de bruit normal ajouté et $\mathbf{I}$ la matrice identité. Cette reparamétrisation permet la propagation du gradient à travers le modèle complet. On peut finalement, après avoir entraîné le modèle, générer de nouveaux exemples en échantillonnant $\mathbf{z}$ dans l'espace latent et en passant le vecteur dans le décodeur du réseau.

Ces modèles possèdent une meilleure capacité de représentation dans l'espace latent que les autoencodeurs classiques grâce à leur structure probabiliste. Ils sont cependant plus difficiles

à entraîner en raison de leur complexité et de l'ajout de la perte par divergence de Kullback–Leibler. En effet, les deux termes de perte doivent être judicieusement équilibrés pour créer une bonne représentation dans l'espace latent tout en gardant une bonne reconstruction des données. Les VAE sont alors utilisés dans notre étude comme modèles génératifs pour créer de nouveaux échantillons avant notre étape de sous-échantillonnage, afin de retirer les échantillons synthétiques de moindre qualité.

Classification avec interactions protéine-protéine

Notre dernière méthode innovatrice porte directement sur la tâche de classification. Le but de cette méthode est de réduire l'impact des hautes dimensions de nos données et de créer un modèle plus interprétable dans un contexte médical. Pour ce faire, nous implémentons les interactions protéine-protéine (PPI) dans un réseau afin d'améliorer l'entraînement du modèle. Pour commencer, il faut savoir qu'il est possible d'utiliser la base de données en ligne STRING, qui permet de créer des graphes d'interactions entre différentes protéines et gènes à partir d'une liste donnée. Nous avons donc converti nos protéines pour qu'elles soient interprétables par STRING à l'aide d'UniProt, une base de données contenant des informations sur les différentes protéines et gènes des organismes. Nous obtenons ainsi un graphe représentant les interactions entre nos 1042 protéines. Cependant, comme illustré à la figure 3.4, où les cercles représentent les protéines et les lignes les interactions, il est impossible d'utiliser ce graphe de manière visuelle.

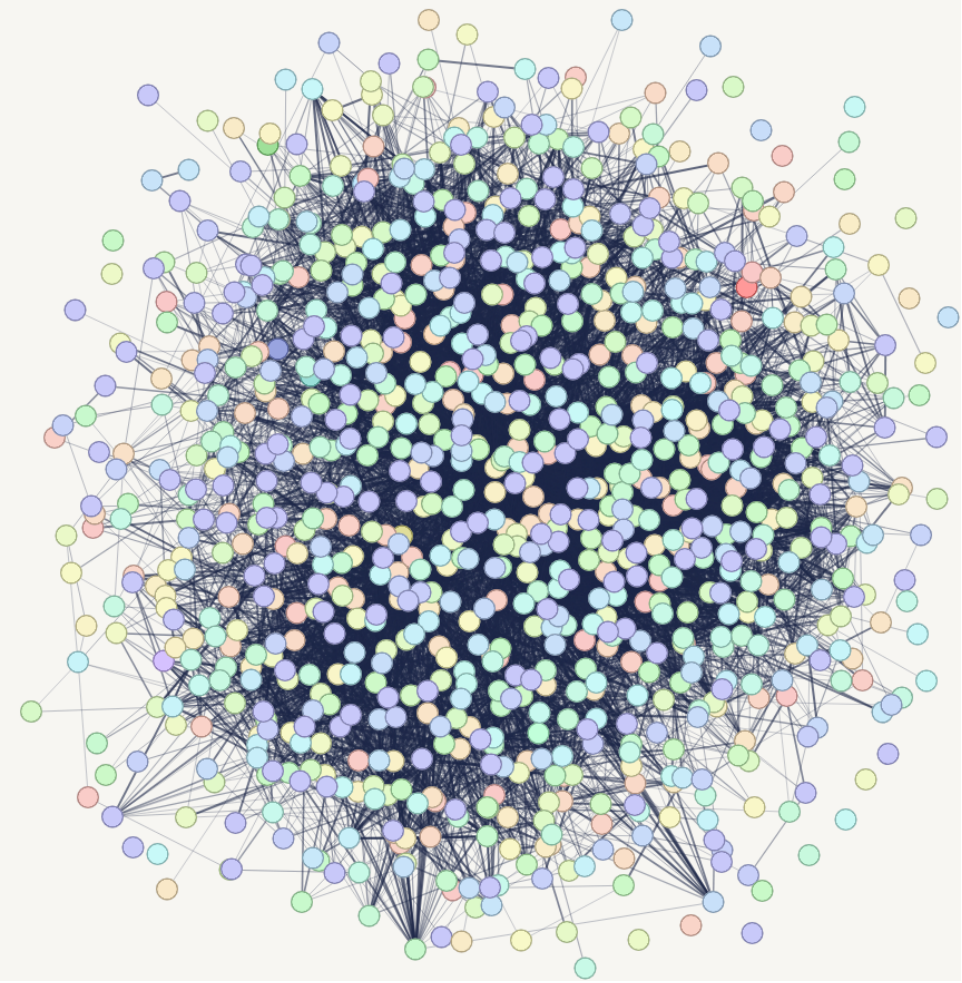

FIGURE 3.4 Graphe des interactions protéine-protéine.

On le traduit alors avec la théorie des graphes par une matrice d'adjacence. Ce type de matrice de taille $M \times M$ (1042x1042) contient les interactions entre chaque combinaison de variables du graphe. Dans notre cas, ces interactions sur STRING sont quantifiées par un score d'évidence, qui combine différents types d'évidence comme des données expérimentales, des études connexes, des co-expressions et de la co-occurrence dans le génome. Il est techniquement possible d'ajouter nos intensités de protéines comme variables sur chaque nœud du graphe pour ensuite utiliser des théories de graphes en apprentissage machine. En effet, il existe des méthodes permettant de classifier des graphes comme pour l'apprentissage machine classique. Cependant, le grand nombre d'interactions (environ 30 000) vient brouiller l'entraînement sur les intensités, et les modèles (principalement des réseaux de neurones adaptés aux graphes) ne sont pas capables d'apprendre à discerner les classes.

Notre approche porte donc sur une adaptation de la matrice d'adjacence vers des données sim-

plifiées permettant leur utilisation dans une méthode d'apprentissage machine classique. Pour ce faire, nous utilisons des méthodes de regroupement adaptées aux graphes pour identifier des sous-groupes de protéines reliées entre elles. Nous adaptons ensuite un réseau de neurones en le transformant en un mélange d'experts, où chaque expert s'entraîne sur son groupe de protéines. Cette méthode permet de réduire considérablement le nombre de poids dans notre réseau comparativement à un réseau traditionnel prenant toutes les données en entrée. Cela devrait réduire le surapprentissage et potentiellement améliorer la généralisation du modèle. La séparation en mélange d'experts permet également une meilleure interprétabilité, puisqu'il est possible de savoir quels experts ont été sollicités pour la prédiction d'un échantillon. En outre, le regroupement est réalisé directement à partir de STRING, qui fournit également une explication biologique pour chaque groupe. En combinant cette explication avec les experts utilisés pour la prédiction, il est possible de mieux comprendre biologiquement comment la prédiction a été faite. Cela améliore ainsi considérablement l'interprétabilité de notre réseau de neurones, qui est normalement très complexe à interpréter.

Pour trouver les sous-groupes, on utilise sur STRING l'algorithme de regroupement de Markov, qui utilise une marche aléatoire pour propager les liens en élevant la matrice d'adjacence à une puissance (généralement de 2). L'algorithme se déroule donc en deux étapes : l'expansion, où l'on élève la matrice, et l'inflation, où l'on renforce ou affaiblit les liens et normalise la matrice pour la rendre stochastique. Ces deux étapes sont définies comme suit :

$$\mathbf{A}^{(t+1)} = \mathbf{A}^{(t)} \circ \mathbf{A}^{(t)} \tag{3.65}$$

et

$$\mathbf{A}_{ij}^{(t+1)} = \frac{(\mathbf{A}_{ij}^{(t+1)})^r}{\sum_k (\mathbf{A}_{ik}^{(t+1)})^r} \tag{3.66}$$

où $\mathbf{A}^{(t)}$ est la matrice d'adjacence stochastique au temps t et r est le taux d'inflation qui régule la granularité des groupes. On répète ces deux étapes jusqu'à ce que l'algorithme converge lorsque la variation de la matrice entre le temps t et $t+1$ est négligeable. La nouvelle matrice est naturellement composée de régions denses et de régions faibles, ce qui permet de créer des groupes en posant des liens comme étant des attracteurs qui attirent les liens sur la même ligne de la matrice.

On obtient finalement une séparation de nos protéines en différents groupes pour nos experts. Cependant, on aimerait aussi pouvoir utiliser les scores d'évidence du graphe pour quantifier les interactions entre les protéines sans avoir à utiliser les quelques 30 000 valeurs directement dans l'entraînement. Pour ce faire, il existe différentes métriques de centralité permettant de quantifier l'importance d'un nœud dans un graphe. Puisque, dans notre cas, les nœuds

représentent les protéines, il est possible de quantifier l'importance de chaque protéine dans le graphe. Pour notre étude, nous utilisons la centralité des vecteurs propres pondérée, qui permet de mesurer l'influence d'un nœud en fonction de son influence et de l'influence de ses nœuds voisins. Pour ce faire, on calcule le vecteur propre possédant la plus grande valeur propre donné par :

$$\mathbf{Av} = \lambda \mathbf{v} \tag{3.67}$$

où λ est la valeur propre et $\mathbf{v}$ le vecteur propre. On obtient alors un vecteur contenant des scores d'importance pour chaque protéine, qui seront utilisés pour l'entraînement de notre modèle. Pour ce faire, nous ajoutons au début du réseau une couche linéaire $\mathbf{S}(\mathbf{x})$ de même dimension que l'entrée, avec un poids par variable. On ajoute ensuite une perte Lasso proportionnelle à l'inverse du vecteur d'importance des protéines dans notre fonction de perte totale. La perte $\mathbf{L}_{sélection}(\mathbf{w}_{sélection})$ est donc donnée par :

$$\mathbf{L}_{sélection}(\mathbf{w}_{sélection}) = \sum_{j=1}^{M} \frac{|w_j|}{v_j} \tag{3.68}$$

où $\mathbf{w}_{sélection}$ sont les poids de la couche de sélection $\mathbf{S}(\mathbf{x})$. Le but est de créer une couche de sélection au début du modèle, ce qui nous permettrait de faciliter l'interprétabilité du réseau tout en dirigeant l'entraînement du modèle vers les protéines importantes dans le graphe. Par la suite, le réseau se divise en différents experts, où chaque expert s'entraîne sur un des groupes de protéines. Les sorties de ces experts sont ensuite utilisées dans une dernière couche pour prédire la classe des échantillons. On peut voir l'architecture du réseau à la figure 3.5.

Le système de portes que nous avons utilisé est une sélection par les K-meilleurs bruités présentée par Shazeer et al. (2017) [117]. Cette approche impose un nombre fixe d'experts activés à chaque prédiction, ce qui permet une meilleure interprétabilité, puisqu'il est alors possible de lier les experts utilisés à l'explication fournie par STRING. Elle contribue également à réduire le surapprentissage, car le modèle n'est pas entièrement sollicité, créant ainsi une forme de régularisation. Le bruit ajouté garantit que tous les experts sont sollicités au cours de l'entraînement, empêchant le modèle de converger vers des minimums locaux où seuls certains experts seraient toujours activés. La couche du système de porte $\mathbf{G}(\mathbf{x})$ est donnée par :

$$\mathbf{G}(\mathbf{x}) = \text{Softmax}(\text{KeepTopK}(\mathbf{h}(\mathbf{x}), K)) \tag{3.69}$$

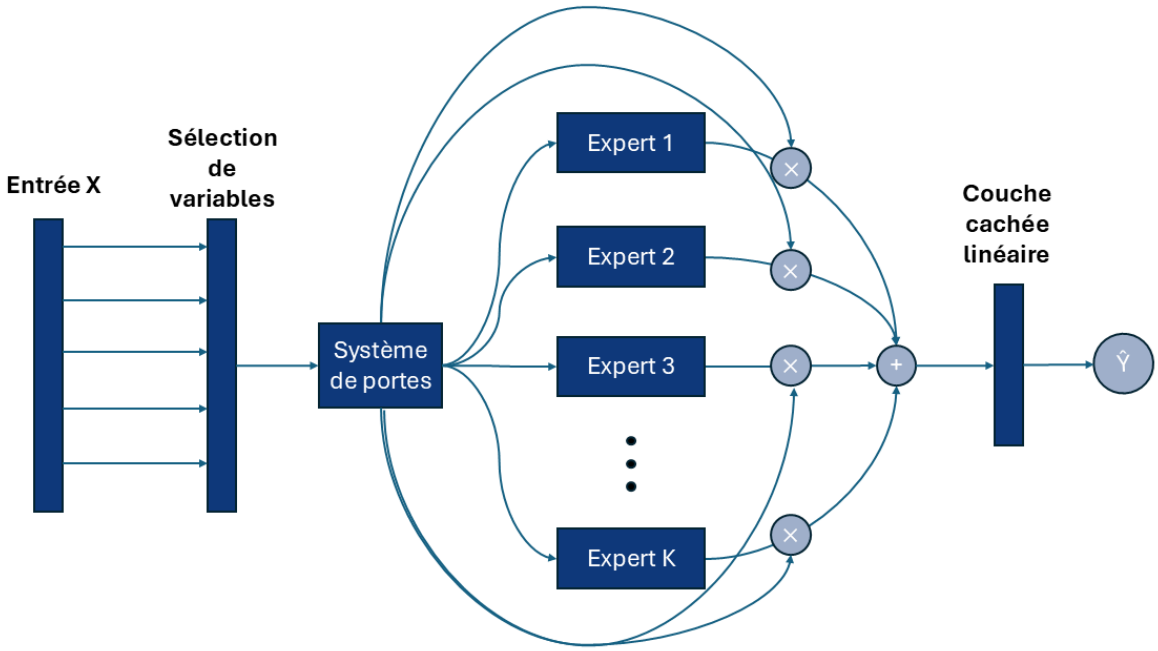

FIGURE 3.5 Schéma du modèle de classification avec interaction protéine-protéine.

où

$$\mathbf{h}(\mathbf{x})_i = (\mathbf{W_g}^T\mathbf{x})_i + \mathcal{N}(0,1) \circ \text{Softplus}((\mathbf{W_{\text{noise}}}^T\mathbf{x})_i), \tag{3.70}$$

$$\text{Softplus}(x) = \log\left(1 + e^x\right), \tag{3.71}$$

$$\text{KeepTopK}(\mathbf{v}, k)_i = \begin{cases} v_i & \text{si } v_i \text{ est dans le top } k \text{ éléments de } \mathbf{v} \\ -\infty & \text{sinon} \end{cases}, \tag{3.72}$$

$\mathbf{W_g}$ et $\mathbf{W}_{\text{noise}}$ sont les matrices de poids du système de porte et K est un hyperparamètre contrôlant le nombre d'experts utilisés. Ces valeurs sont ensuite multipliées par les sorties respectives des experts, ce qui permet de sélectionner celles qui seront utilisées pour la prédiction. Cependant, l'algorithme de regroupement ne permet pas de choisir le nombre de groupes ni le nombre de variables par groupe. Les experts n'auront donc pas le même nombre de variables en entrée. Pour compenser cela, nous utilisons un nombre fixe de neurones en sortie des experts, que nous distribuons équitablement en fonction du nombre de variables en entrée. Par exemple, si nous avons deux experts avec respectivement 70 et 30 variables en entrée, 70 % des neurones en sortie iront au premier expert et 30 % au deuxième. On peut représenter

le modèle final par quatre couches distinctes comme suit :

$$\hat{\mathbf{y}} = \mathbf{H}(\mathbf{E}(\mathbf{G}(\mathbf{S}(\mathbf{x})))) \tag{3.73}$$

où $\mathbf{H}$ est la couche cachée linéaire de classification, $\mathbf{E}$ est la couche d'experts, $\mathbf{G}$ est le système de porte et $\mathbf{S}$ est la couche de sélection de variables. Ce modèle devrait alors permettre d'obtenir un réseau de neurones interprétable sous deux angles : le premier étant la couche de sélection de variables, et le deuxième les experts utilisés pour la prédiction. Le nombre de poids est aussi très réduit en comparaison à un réseau normal, grâce à la séparation de l'entrée vers chaque expert, ce qui permet de réduire le surapprentissage. L'algorithme 4 présente les étapes pour la génération du classificateur PPI.

Algorithme 4 : Algorithme de classification avec PPI

Entrée :

- Ensemble de données $\mathbf{X}$ de taille $N \times M$
- Vecteur de classe à prédire $\mathbf{y}$ de taille N
- Taux d'inflation r
- Matrice d'adjacence $\mathbf{A}$
- Nombre K d'experts utilisés par prédiction
- Critère de convergence ϵ
- Hyperparamètre α pour équilibrer la perte de sélection de variable
- Taille D de la couche linéaire cachée

Sortie : Prédictions $\hat{\mathbf{y}}$

$\mathbf{M}_{i,j} = \frac{\mathbf{A}_{i,j}}{\sum_k \mathbf{A}_{k,j}}$

tant que $||\mathbf{M}^{(t+1)} - \mathbf{M}^{(t)}|| < \epsilon$ **faire**

$\mathbf{M} \leftarrow \mathbf{M} \circ \mathbf{M}$

$\mathbf{M}_{i,j} \leftarrow \mathbf{M}_{i,j}^{r}$

$\mathbf{M}_{i,j} \leftarrow \frac{\mathbf{M}_{i,j}}{\sum_k \mathbf{M}_{k,j}}$

fin

Calcul du vecteur d'importance $\mathbf{v}$ des protéines à l'aide des vecteurs propres de la matrice d'adjacence : $\mathbf{Av} = \lambda\mathbf{v}$.

Séparer les protéines en groupes à l'aide des régions denses de la matrice $\mathbf{M}$ pour obtenir le vecteur $\mathbf{g}$ des numéros de groupe pour chaque protéine.

$\mathbf{Modèle} = \mathbf{RéseauPPI}(\mathbf{X}, \mathbf{y}, \mathbf{g}, \alpha, K, D, \mathbf{v})$

$\hat{\mathbf{y}} = \mathbf{Modèle}(\mathbf{X})$

Vérifier la couche $\mathbf{G}(\mathbf{X})$ du système de porte pour voir les experts utilisés pour les prédictions et les protéines importantes avec les poids de la couche de sélection $\mathbf{S}(\mathbf{x})$.

Métriques de performances

Il existe de nombreuses métriques permettant d'analyser les performances des modèles d'apprentissage machine. Pour notre étude, on se concentre sur les métriques pour les tâches de classification. Plus précisément, nous nous intéressons aux métriques utilisées pour la classification binaire, puisque nous avons seulement deux classes : contrôle et parkinsonien. Dans un premier temps, nous pouvons calculer la précision globale, qui mesure tout simplement le pourcentage d'échantillons bien classés et qui est donnée par :

$$\text{Précision globale} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.74}$$

où TP est le nombre de vrais positifs, TN le nombre de vrais négatifs, FP le nombre de faux positifs et FN le nombre de faux négatifs. Cette métrique est la plus intuitive et simple ; cependant, elle n'est pas fiable lorsque le nombre d'échantillons par classe n'est pas équilibré. En effet, dans notre cas, nous avons plus de patients parkinsoniens que de patients contrôles, le modèle va alors principalement apprendre à classifier le premier groupe. Il aura donc plus de vrais positifs, mais aussi beaucoup de faux positifs. Toutefois, puisque les classes ne sont pas équilibrées, la précision globale est toujours élevée et donne la fausse impression que le modèle est performant.

Les deux prochaines métriques sont souvent utilisées en paire : la précision permet de vérifier les performances sur les prédictions de cas positifs, tandis que le rappel permet de mesurer la capacité du modèle à détecter toutes les instances positives. La précision est obtenue par :

$$\text{Précision} = \frac{TP}{TP + FP} \tag{3.75}$$

et le rappel par :

$$\text{Rappel} = \frac{TP}{TP + FN}. \tag{3.76}$$

Une précision élevée indique que le modèle prédit bien les cas positifs (peu de faux positifs), tandis qu'un rappel élevé montre la capacité du modèle à identifier la majorité des cas positifs. Ces deux métriques sont combinées pour créer le score F1, qui offre une évaluation équilibrée définie par :

$$F1 = 2 \cdot \frac{\text{Précision} \cdot \text{Rappel}}{\text{Précision} + \text{Rappel}} = 2 \cdot \frac{TP}{2TP + FP + FN}. \tag{3.77}$$

On peut également utiliser le score F2, qui met davantage l'emphase sur le rappel pour les

cas où il est plus important de minimiser les faux négatifs que les faux positifs. Ce score est donné par :

$$F2 = \frac{5 \cdot \text{Précision} \cdot \text{Rappel}}{4 \cdot \text{Précision} + \text{Rappel}} = \frac{5 \cdot TP}{5TP + 4FP + FN}. \tag{3.78}$$

Il est également possible de créer des courbes de caractéristiques de fonctionnement du récepteur pour avoir une meilleure vue d'ensemble des performances du modèle. Cette courbe met en relation le taux de vrais positifs (axe Y) et le taux de faux positifs (axe X) en modifiant le seuil de probabilité de prédiction du modèle. Plus la courbe se rapproche du coin supérieur gauche, plus le modèle maximise les vrais positifs tout en minimisant les faux positifs. On peut ensuite calculer l'aire sous la courbe (AUC), qui donne une mesure unique de la performance du modèle, avec une valeur comprise entre 0 (mauvaise performance) et 1 (performance parfaite). Elle se calcule comme suit :

$$\text{AUC} = \int_0^1 \text{TPR}(t)\, d\text{FPR}(t) \tag{3.79}$$

où TPR(t) est le taux de vrais positifs à un seuil t et FPR(t) est le taux de faux positifs à un seuil t. Cette méthode demeure cependant biaisée lorsque le nombre d'échantillons par classe est déséquilibré. Pour atténuer ce biais, nous pouvons utiliser la courbe Précision-Rappel, qui met l'accent sur la performance de classification de la classe positive, c'est-à-dire celle qui nous intéresse pour diagnostiquer les patients atteints de la maladie. En effet, notre objectif est d'identifier des biomarqueurs potentiels de la maladie de Parkinson. Nous cherchons donc principalement à identifier les patients atteints de la maladie. Il est ainsi plus important pour nous de bien diagnostiquer les patients parkinsoniens, même si cela entraîne davantage de faux positifs. La PR-AUC nous permet de bien équilibrer ces deux aspects en tenant compte du rappel et de la précision. Elle est donnée par :

$$\text{PR-AUC} = \int_0^1 \text{Précision}\, d(\text{Rappel}). \tag{3.80}$$

Pour notre recherche, nous allons présenter ces différentes métriques afin de comparer nos modèles. La métrique la plus importante pour nous est cependant la PR-AUC, que nous utiliserons lors de la recherche par grille pour optimiser les hyperparamètres de nos modèles.

3.2.8 Résumé de la méthodologie

En résumé, nous avons testé différentes méthodes s'appliquant à diverses étapes de l'approche systématique pour l'apprentissage machine. Nous avons d'abord, étape par étape,

vérifié quelles méthodes sont les plus efficaces sur nos données afin d'obtenir la combinaison de méthodes la plus performante. Pour ce faire, nous avons d'abord testé les différentes méthodes de mise à l'échelle, suivies par les algorithmes d'imputation et les modèles de sélection de variables. Cependant, on constate que les performances des modèles ne sont pas satisfaisantes et très variables. Cela nous oblige donc à utiliser la validation croisée imbriquée avec deux boucles, une pour la validation et une pour le test. Les deux boucles utilisent $K = 5$ séparations et sont répétées 10 fois pour obtenir un bon échantillonnage des performances. Après avoir choisi l'algorithme complet, nous avons testé nos deux méthodes de sous-échantillonnage, suivies de la méthode de classification par interaction protéine-protéine. Les méthodes que nous avons testées sont présentées à la figure 3.6. Nous avons finalement vérifié, en fonction des performances des modèles, si les protéines dérivées des EEV peuvent contenir des biomarqueurs potentiels.

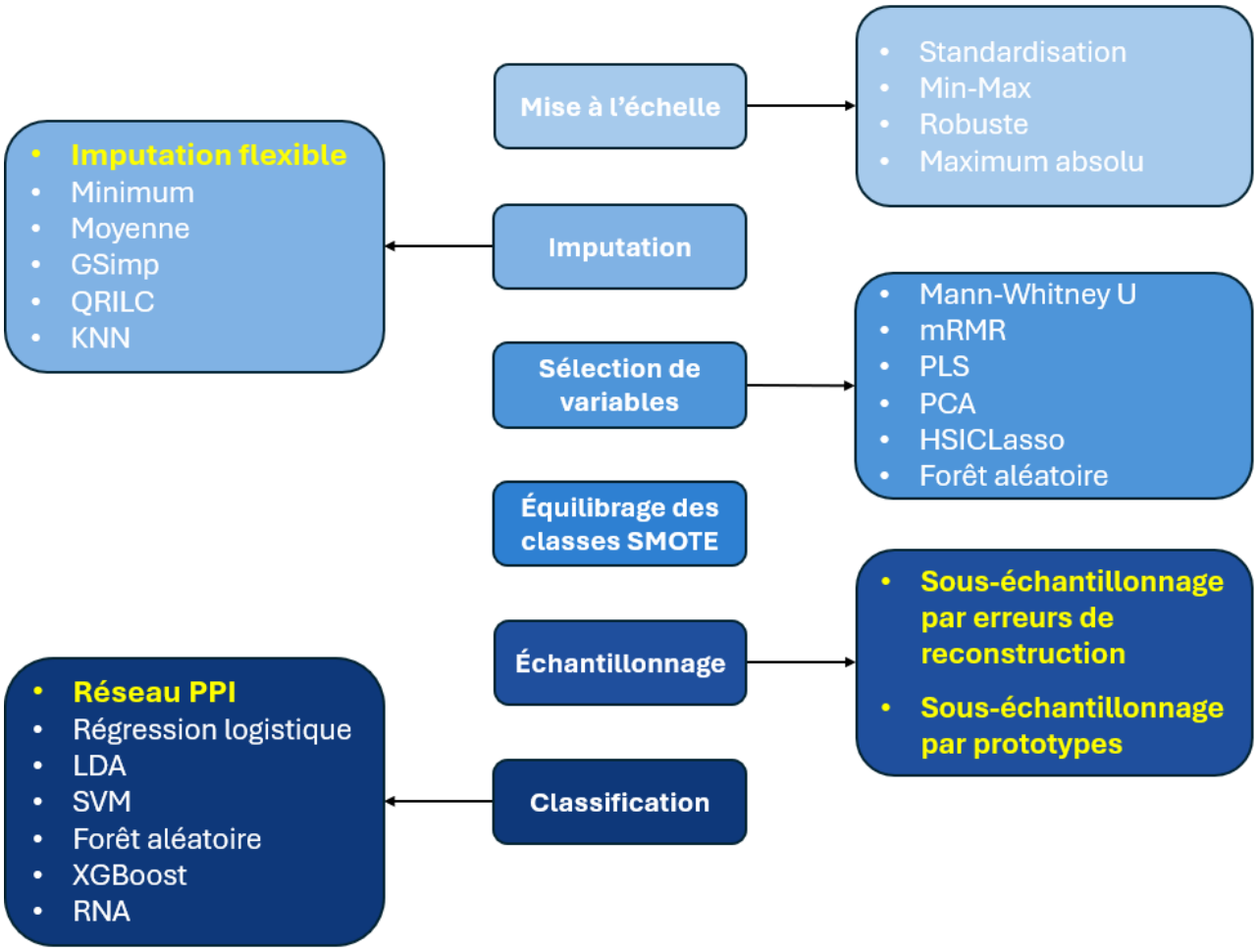

FIGURE 3.6 Diagramme de la méthodologie, avec les méthodes originales en jaune.

3.3 Résultats

Dans cette section, nous présentons les résultats des différentes méthodes utilisées pour la classification des patients contrôles et parkinsoniens. Pour ce faire, nous analysons la moyenne des différentes métriques sur les 50 tests obtenus par validation croisée imbriquée répétée 10 fois. Encore une fois, la métrique la plus importante pour nous est la PR-AUC, utilisée pour effectuer le choix des hyperparamètres. Sachant que nos données comportent 201 échantillons de classe 1 (parkinsonien) et 155 de classe 0 (contrôle), notre PR-AUC de référence, celle que devrait obtenir un modèle qui fait des prédictions aléatoires, est de 0,5646. Dans cette section, nous nous concentrons donc sur cette métrique, bien que les scores F1 et F2 soient également présentés en annexe. Il est aussi important de noter que nous avons éliminé les protéines ayant plus de 80 % de valeurs manquantes, car le nombre d'échantillons n'est plus assez significatif pour procéder à une imputation. Le jeu de données contient donc 956 variables au lieu de 1042. Les données ont également été transformées une première fois à l'aide du logarithme naturel afin de réduire l'asymétrie et les valeurs potentiellement aberrantes. Tout au long des tests, nous avons utilisé les mêmes grilles d'hyperparamètres pour les méthodes de classification et elles sont présentées en annexe. Au total, nous avons 121 modèles à entraîner à chaque boucle de validation sans tenir compte des autres étapes de l'approche systématique.

3.3.1 Normalisation

Dans un premier temps, nous avons testé les différentes méthodes de normalisation. Pour ce faire, aucune méthode de sélection de variables n'a été utilisée, et les données ont été imputées par la moitié des valeurs minimales.

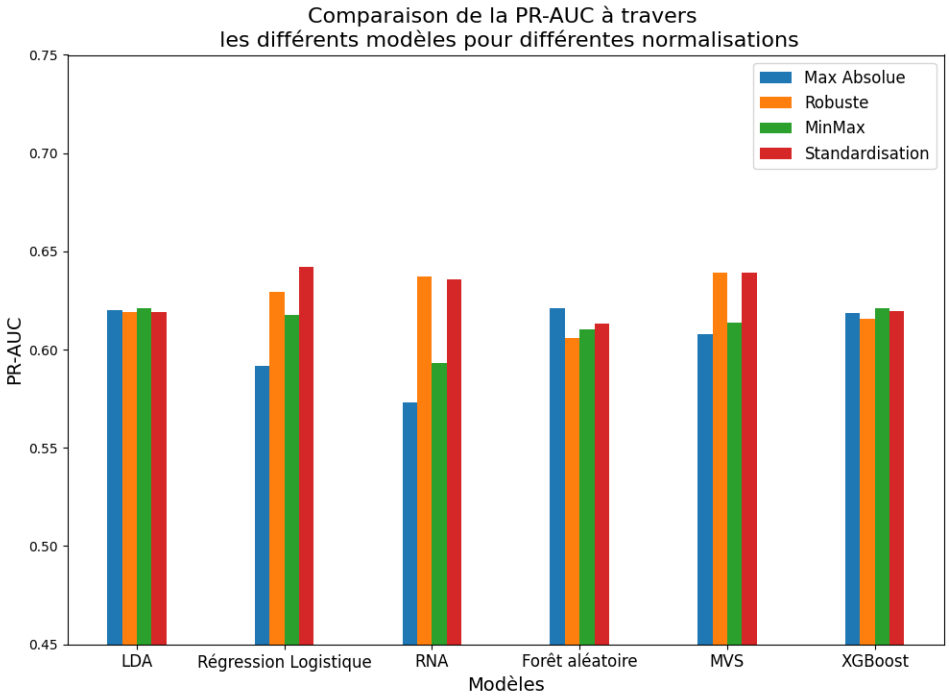

FIGURE 3.7 PR-AUC des différents modèles avec différentes normalisations.

On observe donc, à l'aide de la figure 3.7, qu'en moyenne, la standardisation et la mise à l'échelle robuste semblent être les plus efficaces. La différence est particulièrement visible sur la régression logistique, la MVS et les RNA, car ce sont des modèles qui utilisent des descentes de gradient ou des distances dans l'espace des $\mathbf{X}$. Les deux méthodes offrent donc des performances très similaires, cependant, la standardisation semble être légèrement meilleure. Ces résultats ne sont pas surprenants, puisque les méthodes min-max et maximum absolu sont principalement utilisées dans des cas spécifiques pour contraindre les données dans des intervalles connus. La standardisation et la mise à l'échelle robuste sont donc plus appropriées pour notre jeu de données. Pour la suite, nous utilisons la standardisation, car cette méthode est reconnue comme étant une étape importante pour les réseaux de neurones, et ceux-ci seront utilisés davantage dans nos méthodes originales.

On remarque toutefois que, de manière générale, les performances des modèles sont très similaires, mais limitées. En effet, comme mentionné plus tôt, la PR-AUC de référence est de 0,5646, tandis que notre meilleur modèle ici atteint une PR-AUC d'environ 0,6422 avec la régression logistique. On peut voir les performances des modèles sur la PR-AUC avec la standardisation dans le tableau 3.2.

Tableau 3.2 Résultats de la PR-AUC test, d'entraînement et de validation croisée avec erreur type pour les différents modèles avec la standardisation.

Classificateur	Test PR-AUC	VC PR-AUC	Entraînement PR-AUC
LDA	0.6193(0.0078)	0.6191(0.0044)	0.897(0.0028)
Régression logistique	**0.6422(0.0079)**	0.655(0.0045)	0.9855(0.0037)
RNA	0.6357(0.0081)	**0.668(0.0039)**	0.9943(0.0033)
Forêt aléatoire	0.6135(0.008)	0.6432(0.0035)	**1(0)**
MVS	0.6389(0.0084)	0.6589(0.0043)	0.9903(0.0034)
XGBoost	0.6195(0.0078)	0.6455(0.0041)	0.9997(0.0001)

Ces premiers modèles semblent montrer une faible corrélation entre nos données protéomiques et la maladie de Parkinson. On observe cependant, avec la PR-AUC d'entraînement, que tous les modèles font face à un surapprentissage sévère, même le modèle d'analyse discriminante linéaire. Il est important de noter ici que nous avons tenté d'augmenter la régularisation dans tous ces modèles, mais les performances en test ne faisaient qu'empirer. Néanmoins, en ajoutant les différentes étapes de l'approche systématique, les résultats devraient s'améliorer. Il est également important de noter que l'écart type de ces moyennes de PR-AUC sur les 50 tests tourne autour de 0,05. La variance des résultats est donc très élevée entre chaque test, principalement en raison du faible nombre d'échantillons.

3.3.2 Imputation

La prochaine étape de l'approche est celle de l'imputation pour compléter notre ensemble de données en remplaçant les valeurs manquantes. Nous avons ici utilisé la standardisation comme méthode de normalisation pour tous les tests, et encore une fois, nous n'avons appliqué aucune méthode de sélection de variables. Notre première méthode originale, portant sur l'imputation flexible, a été comparée ici aux autres méthodes d'imputation classiques.

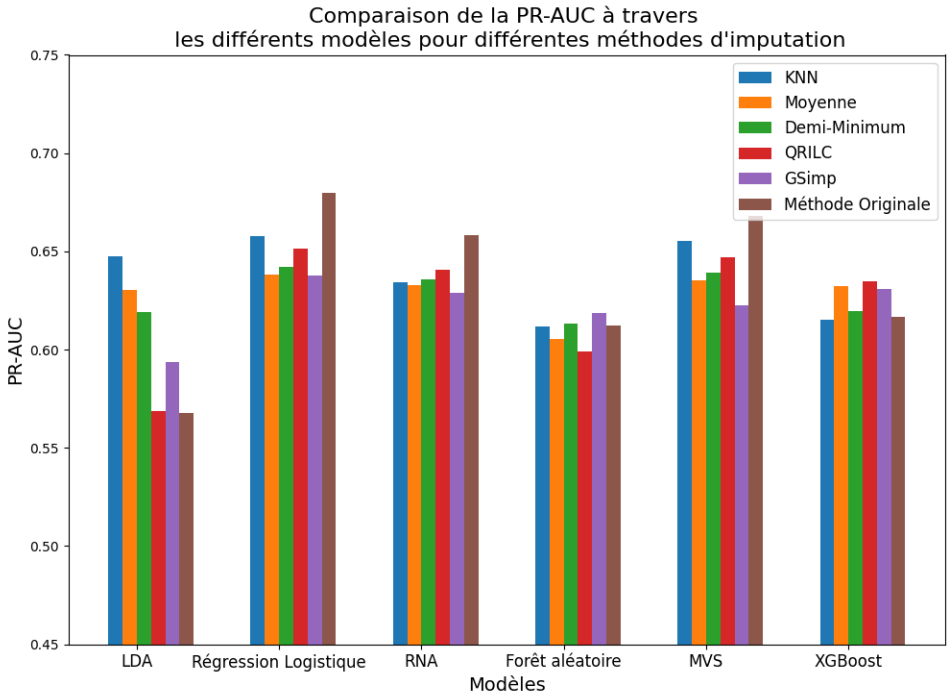

FIGURE 3.8 PR-AUC des différents modèles avec différentes méthodes d'imputation.

La figure 3.8 permet de comparer les performances de nos six modèles de prédiction avec les différentes méthodes d'imputation sur la PR-AUC. On constate notamment que notre méthode originale permet d'atteindre les meilleures performances globales. Plus précisément, les trois modèles les plus efficaces, soit la régression logistique, les RNA, et la MVS, atteignent des PR-AUC respectives de 0,6797, 0,6582 et 0,6680 avec notre méthode d'imputation. Les méthodes de LDA, de forêt aléatoire et de XGBoost ne semblent pas être améliorées par notre imputation, mais les différences de performances pour XGBoost et la forêt aléatoire sont très faibles. On remarque aussi que les performances avec la LDA sont moins bonnes lorsqu'on utilise des méthodes complexes par rapport à des méthodes simples. Cela est plutôt normal, sachant que la LDA est une méthode très simple et linéaire, peu adaptée à des tâches complexes. Notre méthode reste cependant celle qui permet la plus grande amélioration des performances en général et est donc utilisée pour la suite des tests. Le tableau 3.3 montre les résultats des modèles avec notre méthode d'imputation durant la validation croisée et l'entraînement.

Tableau 3.3 Résultats de la PR-AUC test, d'entraînement et de validation croisée avec erreur type pour les différents modèles avec notre méthode d'imputation.

Classificateur	Test PR-AUC	VC PR-AUC	Entraînement PR-AUC
LDA	0.5677(0.0046)	0.6534(0.0045)	0.9848(0.0037)
Régression logistique	**0.6798(0.0092)**	0.6814(0.0042)	0.9927(0.0024)
RNA	0.6582(0.0087)	**0.6843(0.0039)**	0.9928(0.0038)
Forêt aléatoire	0.6123(0.0062)	0.6434(0.0038)	**1(0)**
MVS	0.6680(0.0092)	0.6774(0.0043)	0.9976(0.0012)
XGBoost	0.6169(0.0078)	0.6477(0.0039)	0.9999(0.0001)

On observe que notre méthode a permis d'améliorer significativement les performances des trois meilleurs modèles.

3.3.3 Sélection de variables

Comme nous l'avons expliqué dans l'introduction de ce projet, les intensités des protéines sont très corrélées entre elles. Cette propriété peut complexifier la convergence des méthodes et leur interprétabilité. Le grand nombre de protéines par rapport au petit nombre d'échantillons crée aussi du surapprentissage, comme nous l'avons vu dans les résultats précédents sur la PR-AUC d'entraînement. La sélection de variables pourrait permettre de réduire l'impact de ces deux caractéristiques de nos données. La figure 3.9 présente les performances sur la métrique de PR-AUC des méthodes de prédiction avec une standardisation, notre méthode d'imputation originale, et différentes sélections de variables.

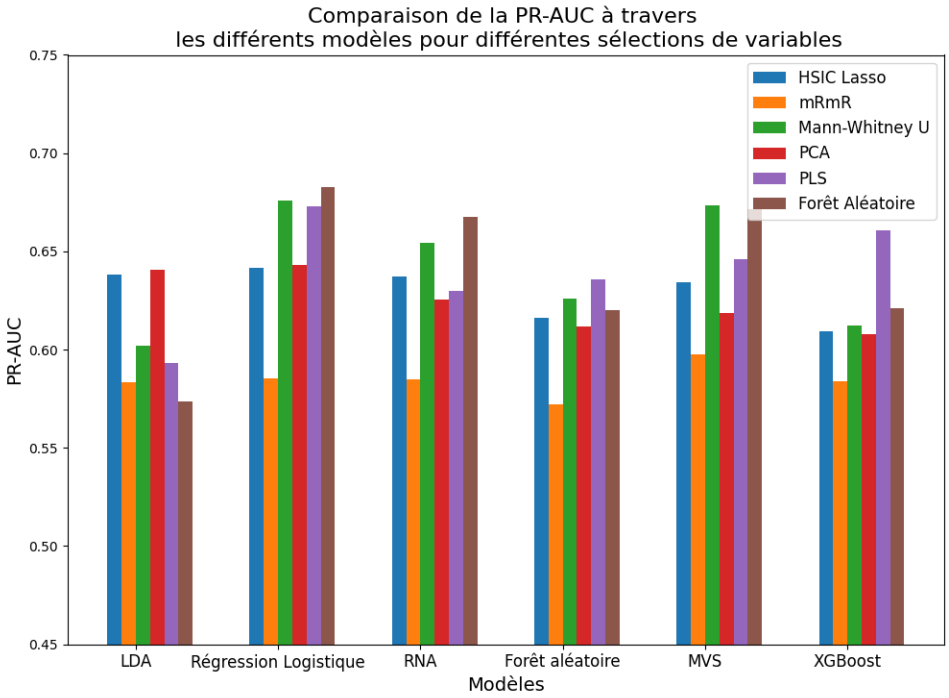

FIGURE 3.9 PR-AUC des différents modèles avec différentes méthodes de sélection de variables.

On observe d'abord que, en général, les deux meilleures méthodes semblent être la forêt aléatoire et la sélection à l'aide du test de Mann–Whitney U. En effet, les trois méthodes les plus performantes sont, encore une fois, la régression logistique, le SVM et les réseaux de neurones artificiels (RNA) en utilisant ces deux méthodes de sélection de variables. La PLS semble également donner de bons résultats, cependant elle rend l'interprétabilité des modèles plus complexe et n'est donc pas envisagée pour la suite. La LDA obtient de meilleures performances lorsque l'on utilise des méthodes simples comme le HSIC Lasso et la PCA, ce qui concorde avec les résultats d'imputation. Le tableau 3.4 présente les résultats des modèles avec la forêt aléatoire et le test de Mann–Whitney U pour mieux comparer les meilleures méthodes.

Tableau 3.4 Résultats de la PR-AUC test, d'entraînement et de validation croisée avec erreur type pour les différents modèles avec sélection de variables par Mann-Whitney U et forêt aléatoire.

Classificateur avec forêt aléatoire			
Classificateur	Test PR-AUC	VC PR-AUC	Entraînement PR-AUC
LDA	0.5737(0.005)	0.6549(0.0045)	0.9798(0.0044)
Régression logistique	**0.6829(0.0091)**	0.6825(0.0042)	0.9927(0.0024)
RNA	0.66779(0.0086)	**0.6964(0.0038)**	0.9885(0.0063)
Forêt aléatoire	0.6202(0.0082)	0.6560(0.0033)	**1(0)**
MVS	0.6714(0.0099)	0.6787(0.0043)	0.9973(0.0013)
XGBoost	0.6212(0.0071)	0.6483(0.0039)	0.9999(0.0001)
Classificateur avec Mann-Whitney U			
LDA	0.6021(0.0098)	0.6551(0.0039)	0.9649(0.0051)
Régression logistique	0.6759(0.008)	0.6795(0.0041)	0.9809(0.0036)
RNA	0.6544(0.0079)	0.6940(0.0035)	0.9909(0.0035)
Forêt aléatoire	0.6260(0.0072)	0.6597(0.0037)	**1(0)**
MVS	0.6735(0.0088)	0.6808(0.004)	0.981(0.0042)
XGBoost	0.6121(0.0072)	0.6605(0.0041)	0.9993(0.0003)

Les résultats sont très semblables et quasiment indiscernables si l'on prend en compte l'erreur-type. Toutefois, les deux modèles les plus performants pour l'instant, soit les RNA et la régression logistique, obtiennent les meilleures performances avec la forêt aléatoire. De plus, la forêt aléatoire est une méthode qui permet de capturer des relations non linéaires entre les variables, ce qui est très important dans notre étude, tandis que le test statistique n'utilise qu'une protéine à la fois et les considère indépendantes. Pour ces deux raisons, nous avons conclu que la forêt aléatoire serait utilisée comme méthode de sélection de variables pour le reste des calculs. La sélection de variables ne semble cependant pas avoir eu l'impact escompté sur le surapprentissage. En effet, on remarque que les PR-AUC d'entraînement sont toujours proches de 1 tandis que les valeurs de validation croisée et de test restent limitées.

3.3.4 Échantillonnage

Cette prochaine section porte sur les résultats de nos deux méthodes d'échantillonnage après la sélection de variables et l'équilibrage des classes. Pour ce faire, nous avons comparé les résultats de nos deux méthodes à notre meilleur modèle obtenu jusqu'à présent. Plus précisément, nous avons testé deux variations des deux méthodes. Pour le sous-échantillonnage

avec l'erreur de reconstruction, nous avons testé avec et sans une première augmentation de données à l'aide du mélange d'experts de VAE. Dans le cas du sous-échantillonnage par prototypes, nous avons constaté que les résultats ne sont malheureusement pas comparables à ceux de notre meilleur modèle. Dans un premier temps, nous avons testé avec $K = 5, 10, 15$ ou 20, puis avec $K = 70, 80, 90$ ou 100. Le premier test visait à évaluer notre première idée pour cette méthode, où l'on utilise une méthode de classification simple avec un faible nombre d'échantillons pour permettre une bonne interprétabilité. Le deuxième cas ne permet plus cette grande interprétabilité, mais pourrait aider à sélectionner de meilleurs groupes d'échantillons parmi l'ensemble complet. On peut voir les résultats de la PR-AUC de ces quatre méthodes comparés à ceux de notre meilleur modèle jusqu'à présent à la figure 3.10.

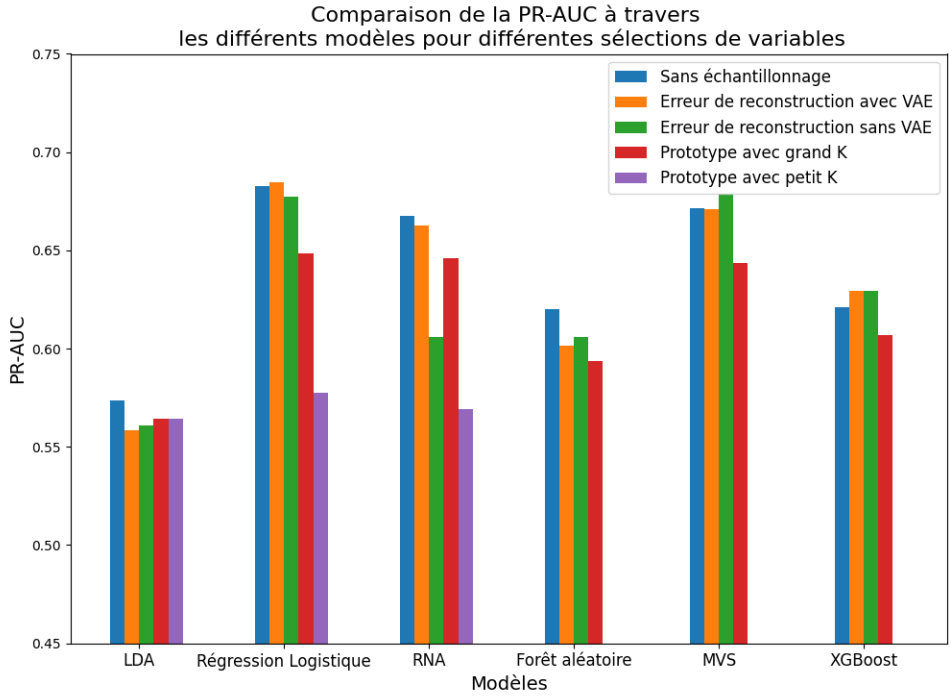

FIGURE 3.10 PR-AUC des différents modèles avec différentes méthodes d'échantillonnage.

On observe donc, premièrement, que la méthode par prototype ne semble pas être efficace. En effet, que ce soit avec des valeurs de l'hyperparamètre K élevées ou faibles, les performances des modèles sont inférieures à celles des autres méthodes d'échantillonnage ou à l'absence d'échantillonnage. On constate cependant qu'augmenter le nombre d'échantillons par groupe rapproche les performances des modèles de celles obtenues sans échantillonnage. Il est donc envisageable d'optimiser davantage cet hyperparamètre pour obtenir un nombre proche de la

quantité totale d'échantillons, mais plus faible, permettant ainsi d'éliminer ceux qui réduisent les performances des modèles.

Ensuite, si l'on observe les résultats des modèles avec sous-échantillonnage par erreur de reconstruction, on constate que certains d'entre eux obtiennent de meilleures performances que notre meilleur modèle actuel. En effet, on peut voir, par exemple, que la régression logistique, la MVS et le XGBoost performent mieux selon la PR-AUC lorsqu'on applique dans un premier temps notre méthode de sous-échantillonnage. Cependant, on remarque aussi que l'ajout préalable de l'augmentation de données par VAE n'est pas nécessairement bénéfique. Cela peut s'expliquer par le fait que le modèle était probablement confronté à du surapprentissage, comme dans toutes nos méthodes, ce qui a conduit à des échantillons synthétiques peu variés. De plus, nous ne nous sommes pas attardés à l'optimisation des hyperparamètres pour ces VAE, ce qui aurait pu améliorer la qualité des échantillons générés. Le tableau 3.5 présente les performances du modèle avec VAE et sous-échantillonnage et du meilleur modèle pour l'instant. Ce tableau permet de conclure que notre méthode de sous-échantillonnage par erreur de reconstruction, avec l'aide d'autoencodeurs supervisés, a permis une amélioration des performances. Cependant, ces gains sont relativement faibles.

Tableau 3.5 Résultats de la PR-AUC test, d'entraînement et de validation croisée avec erreur type pour les différents modèles avec échantillonnage par erreur de reconstruction et VAE et sans échantillonnage.

Classificateur sans échantillonnage			
Classificateur	Test PR-AUC	VC PR-AUC	Entraînement PR-AUC
LDA	0.5737(0.005)	0.6549(0.0045)	0.9798(0.0044)
Régression logistique	0.6829(0.0091)	0.6825(0.0042)	0.9927(0.0024)
RNA	0.66779(0.0086)	0.6964(0.0038)	0.9885(0.0063)
Forêt aléatoire	0.6202(0.0082)	0.6560(0.0033)	**1(0)**
MVS	0.6714(0.0099)	0.6787(0.0043)	0.9973(0.0013)
XGBoost	0.6212(0.0071)	0.6483(0.0039)	0.9999(0.0001)
Classificateur avec échantillonnage			
LDA	0.5582(0.0054)	0.6549(0.0045)	0.9729(0.0054)
Régression logistique	**0.6847(0.0088)**	0.6823(0.0042)	0.9919(0.0024)
RNA	0.6627(0.0072)	**0.6967(0.0038)**	0.9887(0.004)
Forêt aléatoire	0.6016(0.0067)	0.6526(0.0032)	**1(0)**
MVS	0.6712(0.0095)	0.6782(0.0043)	0.9925(0.0022)
XGBoost	0.6296(0.0092)	0.6461(0.0039)	0.9997(0.0002)

3.3.5 Classificateur PPI

Maintenant que nous avons optimisé les différentes étapes de l'approche systématique, nous testons notre dernière méthode originale, soit le classificateur PPI. Pour comparer les résultats de classification de notre méthode à ceux des autres méthodes, nous utilisons les mêmes étapes préalables choisies précédemment. La figure 3.11 présente les performances des modèles sur les différentes métriques de classification. Celles qui nous intéressent le plus sont le score F1, le score F2 et la PR-AUC.

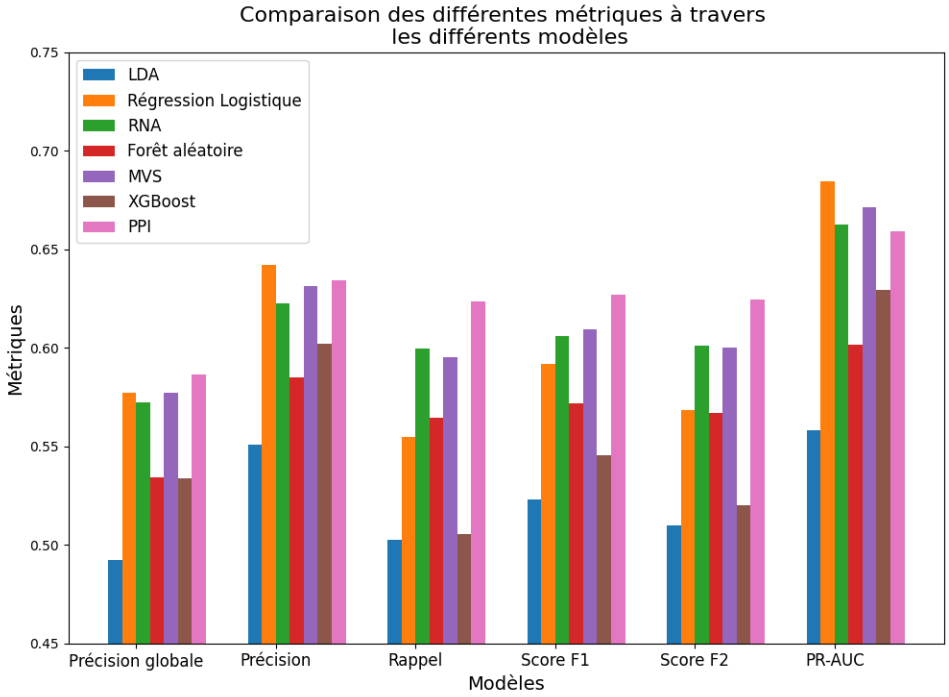

FIGURE 3.11 Métriques sur les différents modèles dont notre méthode originale avec PPI.

On observe que notre méthode n'est pas la plus efficace sur la métrique de PR-AUC, étant en dessous de la régression logistique, de la MVS et du RNA. Cependant, notre modèle performe mieux sur les scores F1 et F2, montrant qu'au seuil de base du modèle de 0.5, il est plus performant que les autres. Il est également meilleur sur la métrique de rappel, ce qui indique qu'il détecte mieux les cas positifs, mais avec une précision légèrement inférieure à celle de la régression logistique. Le tableau 3.6 présente les résultats numériques des différentes métriques.

Tableau 3.6 Résultats des différentes métriques sur les classificateurs avec les étapes préalable optimisées.

Classificateur	Test Accuracy	Test Precision	Test Recall	Test F1-Score
Classificateur PPI	**0.5863(0.0068)**	0.6342(0.0063)	**0.6234(0.0089)**	**0.627(0.0062)**
LDA	0.4922(0.0094)	0.5509(0.0095)	0.5026(0.0134)	0.5227(0.0105)
Régression logistique	0.577(0.0082)	**0.6419(0.0083)**	0.5548(0.0130)	0.592(0.0096)
RNA	0.5721(0.0072)	0.6224(0.0065	0.5996(0.0154)	0.6061(0.0098)
Forêt aléatoire	0.5342(0.0091)	0.5851(0.0079)	0.5646(0.0149)	0.5719(0.0109)
MVS	0.5773(0.0079)	0.6312(0.0080)	0.595(0.0128)	0.6093(0.0086)
XGBoost	0.5337(0.0086)	0.602(0.0095)	0.5052(0.0124)	0.5455(0.0096)
Classificateur	Test F2-Score	Test PR-AUC	CV PR-AUC	Entraînement PR-AUC
Classificateur PPI	**0.6244(.0077)**	0.6592(0.0075)	**0.7006(0.0042)**	**1(0)**
LDA	0.5099(0.0122)	0.5582(0.0054)	0.6549(0.0045)	0.9729(0.0054)
Régression logistique	0.5685(0.0116)	**0.6847(0.0088)**	0.6823(0.0042)	0.9919(0.0024)
RNA	0.6013(0.0131)	0.6627(0.0072)	0.6967(0.0038)	0.9887(0.004)
Forêt aléatoire	0.5669(0.0133)	0.6016(0.0067)	0.6526(0.0032)	**1(0)**
MVS	0.6(0.0109)	0.6712(0.0095)	0.6782(0.0043	0.9925(0.0022)
XGBoost	0.5199(0.0112)	0.6296(0.0092)	0.6461(0.0039)	0.9997(0.0002)

On remarque que notre classificateur PPI est le plus performant sur la majorité des métriques, sauf la précision (de très peu) et la PR-AUC. Toutefois, la PR-AUC de validation est la plus élevée, avec autant d'hyperparamètres que le RNA, ce qui pourrait indiquer que les performances en test sont légèrement sous-estimées. La méthode est également encore sujette à du surapprentissage, comme en témoigne la PR-AUC d'entraînement, bien que le nombre de passages soit significativement plus faible que pour le RNA classique.

3.4 Discussion des résultats

Les résultats de nos modèles suggèrent l'absence de biomarqueurs significatifs dans notre ensemble de données protéomiques. En effet, les performances limitées compliquent l'identification des biomarqueurs, puisque chaque prédiction utilise des protéines complètement différentes. Il semble donc y avoir très peu de corrélation entre le protéome des VEE dans le

sang et la maladie de Parkinson. Bien que les résultats des modèles ne soient pas suffisamment concluants pour poursuivre l'analyse médicale et identifier des biomarqueurs, il demeure pertinent d'examiner l'impact de nos méthodes sur les performances de classification dans le cadre de futures études.

3.4.1 Mise à l'échelle

Dans un premier temps, nous avons testé l'efficacité de différentes méthodes de mise à l'échelle des données pour le prétraitement. Nous avons retenu, parmi quatre méthodes, que la standardisation semblait être la plus efficace pour la majorité des modèles. En effet, les autres méthodes, comme min-max et maximum absolu, sont plutôt employées pour contraindre les variables dans un intervalle donné. Par exemple, le min-max a été utilisé dans nos réseaux autoencodeurs supervisés afin d'employer une perte d'entropie croisée pour la reconstruction avec les variables entre 0 et 1.

3.4.2 Imputation flexible

Nous avons ensuite testé les différentes méthodes d'imputation des valeurs manquantes. C'est ici que notre première méthode originale a été implémentée. Nous avons observé que cette étape permettait la plus grande augmentation de performances parmi toutes les étapes. Plus précisément, notre méthode originale s'est avérée être la meilleure parmi un ensemble de six méthodes, allant des plus simples aux plus complexes et adaptées aux domaines. En effet, l'imputation flexible a permis une augmentation d'environ 0.03 de la PR-AUC des meilleures méthodes de classification par rapport à l'imputation avec le demi-minimum. L'approche permet aussi d'estimer la valeur du seuil de détection, ce qui pourrait être potentiellement pertinent pour les chercheurs du domaine. Elle était cependant plus longue en termes de calcul en comparaison aux méthodes simples, en raison de son approche par espérance-maximisation qui peut nécessiter de nombreuses itérations si les données sont complexes.

3.4.3 Sélection de variables

Par la suite, nous avons implémenté différentes méthodes de sélection de variables pour tenter de réduire la quantité de données en entrée des modèles de prédiction afin de limiter le surapprentissage. En effet, nous avons observé que les performances en entraînement étaient toujours autour de 99 % de précision globale, ce qui représente du surapprentissage, sachant que les performances test et validation étaient bien plus faibles. Nous avons constaté que la sélection de variables par forêt aléatoire semblait être la plus efficace pour les performances

finales. Cependant, elle ne permettait pas de réduire le surapprentissage comme nous l'aurions souhaité.

3.4.4 Échantillonnage

Ensuite, nous avons testé l'ajout de deux méthodes de sous-échantillonnage originales. La première avait pour but de réduire le nombre d'échantillons aberrants, car nos données sont bruitées, tandis que la deuxième servait à réduire significativement le nombre d'échantillons pour permettre une meilleure interprétabilité. Dans un premier temps, la méthode par erreur de reconstruction a permis des améliorations des performances de nos modèles en général ; toutefois, elles étaient plutôt faibles. Ces améliorations étaient un peu plus grandes lorsque l'on utilisait une méthode d'augmentation de données avec VAE avant le sous-échantillonnage. Notre méthode a permis de générer des échantillons synthétiques afin d'élargir la base de données tout en éliminant ceux de mauvaise qualité. Cependant, la deuxième méthode n'a pas permis d'obtenir des résultats aussi concluants. En effet, nous avons testé la méthode avec un nombre faible d'échantillons par groupe, ce pour quoi nous l'avions initialement conçue. Toutefois, les performances des modèles simples devenaient nettement moins bonnes, malgré le fait que les échantillons choisis devraient être bien séparés dans l'espace des $\mathbf{X}$. Nous avons ensuite tenté d'augmenter le nombre d'échantillons pour adapter la méthode vers une approche ressemblant davantage à notre première méthode, où l'on retire les échantillons aberrants. On observait cependant, encore une fois, des performances inférieures à celles obtenues sans l'ajout de cette étape. Le but de cette méthode n'était pas d'améliorer les performances, mais plutôt d'augmenter l'interprétabilité des modèles. Néanmoins, les réductions de performances sont trop importantes pour conclure que la méthode est efficace. Nous cherchions plutôt à obtenir des performances stables ou faiblement réduites, ce qui aurait permis un modèle puissant avec une grande interprétabilité.

3.4.5 Classificateur PPI

En dernier, nous avons créé un algorithme de classification intégrant les interactions protéine-protéine, le regroupement et un réseau de neurones modifié pour réduire le nombre total de poids et améliorer l'interprétabilité. Ce classificateur PPI a permis des augmentations de performances sur certaines métriques comme le rappel, le score F1 et le score F2, mais pas sur la PR-AUC. Cela semble donc montrer que le modèle, avec un seuil de 0.5, est plus performant que les autres classificateurs. Cependant, il est très sensible à la valeur de ce seuil, puisque si l'on change celle-ci, ses performances deviennent moins bonnes, ce qui est montré par la PR-AUC. On peut néanmoins conclure que notre méthode originale de classification

rivalise avec les performances des méthodes classiques. De plus, le modèle permet un niveau d'interprétabilité bien plus élevé que les autres méthodes. En effet, en observant quels experts ont été utilisés pour faire les prédictions, il est possible de mieux comprendre biologiquement quel groupe de protéines est relié à la maladie. On peut aussi regarder les poids de la couche de sélection pour retirer les protéines non pertinentes ayant des poids proches de 0. Dans notre cas, les modèles semblent indiquer qu'il n'y a pas de biomarqueurs, car l'ensemble des données est utilisé, ce qui entraîne du surapprentissage. Néanmoins, on peut conclure que le classificateur PPI a du potentiel tant au niveau des performances de classification que de son interprétabilité.

CHAPITRE 4 ÉTUDE EXPLORATOIRE D'UN NOUVEAU TRAITEMENT POUR UNE MALADIE NEUROLOGIQUE DE TYPE GÉNÉTIQUE

4.1 Introduction au projet

Ce chapitre porte sur l'analyse de données provenant d'une étude clinique où la molécule X a été administrée à des patients souffrant d'une maladie neurologique de type génétique. Toutefois, les données récoltées ne respectent pas de nombreuses conditions qui sont de nos jours obligatoires, notamment l'homogénéité des groupes de patients choisis. Ils présentent effectivement de nombreuses caractéristiques très variables, ce qui rend les analyses plus complexes et moins fiables. Notre étude est alors de type exploratoire. Le but n'est pas de prouver l'efficacité du médicament pour mener à sa commercialisation, mais plutôt de vérifier s'il serait pertinent de mener de nouvelles études avec des groupes de patients plus homogènes ou qui répondent à des critères bien spécifiques.

Les données d'étude auxquelles nous avons accès portent sur 95 patients, qui sont séparés en deux groupes égaux de manière aléatoire. Durant les 18 premiers mois de l'étude, le premier groupe a été traité à l'aide de la médication, tandis que le second groupe recevait un placebo. Les deux groupes et les chercheurs ne sont pas au courant du type d'intervention que les patients reçoivent, pour éviter tout biais cognitif. Cette première partie de l'étude permet de séparer les deux groupes pour vérifier si la médication améliore l'état des patients dans le temps. Ensuite, après le 18e mois, le second groupe commence aussi à prendre la médication jusqu'à la fin de l'étude. Cette deuxième partie permet de vérifier si l'efficacité de la médication sur le premier groupe perdure dans le temps et si l'état des patients du second groupe se voit aussi amélioré. Dans notre cas, la deuxième partie de l'étude s'est étendue jusqu'à 72 mois pour certains patients. Cependant, nous avons analysé les données à un maximum de 54 mois, puisqu'il n'y avait plus assez de patients après. Nous avons donc accès à 8 visites différentes, soit les mois 0, 12, 18, 24, 36, 45, 54, et la référence un mois avant le début de l'étude.

Pour vérifier l'amélioration de l'état des patients, différents indices permettant de mesurer leurs capacités ont été utilisés durant l'étude. Ces indices proviennent en majorité du « overall disease clinical scale » (ODCS), l'échelle la plus utilisée pour surveiller les patients souffrant de cette maladie génétique. L'indice principalement utilisé dans notre étude est le score sur des évaluations motrices que nous nommerons « partie motrice ». Celle-ci est de plus composée de 4 types de sous-évaluations que nous nommerons ici, toujours pour des

besoins de confidentialité, symptôme moteur 1, symptôme moteur 2, symptôme moteur 3 et symptôme moteur 4. Pour notre étude, cet indice nous permettra principalement d'analyser la variation de l'état des patients dans le temps. Cependant, nous avons également considéré quelques autres scores, comme l'échelle de démence de Mattis pour évaluer les fonctions cognitives, l'échelle d'activités journalières, qui permet d'évaluer l'autonomie du patient et une autre mesure faisant référence au niveau de fonctionnalité. Nous avons également eu accès à différentes variables démographiques des patients, comme l'âge, le sexe, la taille, le poids et l'indice de masse corporelle (IMC). D'autres variables, plus en lien avec l'étude, nous ont également été fournies, telles que la sévérité de la mutation génétique associée à la maladie. Finalement, nous avons aussi accès aux concentrations du médicament dans le sang des patients et mesurées durant l'étude. Toutefois, nous avons vite remarqué que cette donnée ne semblait pas être fiable, car les patients ne prenaient pas la médication aux mêmes heures ni de manière régulière. Cette variable a été analysée durant notre étude en fonction des intérêts des collaborateurs, mais les résultats n'ont pas été concluants et ne seront pas présentés.

4.2 Méthodologie

Comme nous l'avons vu dans la revue de littérature, le modèle le plus pertinent pour analyser des études longitudinales est celui du MMRM. Cette méthode statistique permet de prendre en compte la corrélation entre les mesures provenant d'un même sujet, à l'aide d'effets aléatoires ajoutés aux modèles. On peut donc à la fois modéliser les effets intra-sujets et inter-sujets par le biais des effets fixes et des effets aléatoires. Le modèle étant statistique, il permet aussi de faire des tests de signification sur les effets pour vérifier leur importance dans le modèle. C'est avec cette propriété que nous testons l'efficacité de la médication sur la variation de l'état des patients dans le temps. Ce modèle est aussi très flexible comparé au modèle ANOVA pour mesures répétées. En effet, le MMRM ne force pas les mesures répétées à être dans des intervalles fixes ; il permet d'analyser des données possédant des valeurs manquantes, de modéliser différents types de structures de covariance et d'ajouter des effets aléatoires pour modéliser des pentes différentes pour chaque sujet. Nous allons d'abord présenter l'analyse de la variation du ODCS - partie motrice dans le temps sur la cohorte complète pour voir si la médication est efficace. Par la suite, nous avons dichotomisé la cohorte selon différentes caractéristiques que les collaborateurs estiment pertinentes et testé l'efficacité de la médication sur ces sous-groupes. Nous avons finalement répété ces étapes sur les différents indices pour déterminer si la médication avait des impacts plus significatifs sur des symptômes précis de la maladie.

4.2.1 Modèle mixte pour mesure répétée

Le MMRM est un modèle linéaire permettant de modéliser une variable de réponse $\mathbf{y}$ en fonction de différents effets fixes et effets aléatoires. On peut modéliser la réponse comme suit :

$$\mathbf{y}_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{Z}_i\mathbf{b}_i + \boldsymbol{\epsilon}_i \tag{4.1}$$

où $\mathbf{y}_i$ est le vecteur des réponses dans le temps du i-ème sujet, de longueur n_i, $\mathbf{X}_i$ est la matrice des effets fixes, de taille $n_i \times p$, $\boldsymbol{\beta}$ est le vecteur de coefficients des effets fixes, de longueur p, $\mathbf{Z}_i$ est la matrice des effets aléatoires, de taille $n_i \times q$, $\mathbf{b}_i$ est le vecteur de coefficients des effets aléatoires, de longueur q et $\boldsymbol{\epsilon}_i$ est le vecteur d'erreurs résiduelles, de longueur n_i. n_i représente donc le nombre de mesures pour le i-ème sujet, p le nombre de variable à effet fixe et q le nombre de variable à effet aléatoire. Ces derniers suivent des lois normales données par :

$$\mathbf{b}_i \sim \mathcal{N}(0, \mathbf{G}) \tag{4.2}$$

et

$$\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, \mathbf{R}_i) \tag{4.3}$$

où $\mathbf{G}$ est la matrice de covariance des effets aléatoires et $\mathbf{R}_i$ la matrice de covariance des résidus. Ces deux matrices peuvent avoir différentes formes comme la structure de symétrie composée hétérogène, l'autorégressive, la non-structurée et bien d'autres. La première permet de forcer les variances σ^2 et les corrélations ρ entre deux mesures répétées à être constantes et est définie par :

$$\mathbf{R}_i = \sigma^2(\mathbf{I} + \rho\mathbf{J}) \tag{4.4}$$

où $\mathbf{I}$ est la matrice identité et $\mathbf{J}$ est une matrice dont tous les éléments sont égaux à 1. Dans la structure autorégressive, on pose que les corrélations se détériorent dans le temps plus les mesures sont espacées. Elle est donc exprimée comme suit :

$$\mathbf{R}_i = \sigma^2 \begin{bmatrix} 1 & \rho & \rho^2 & \dots & \rho^{m-1} \\ \rho & 1 & \rho & \dots & \rho^{m-2} \\ \rho^2 & \rho & 1 & \dots & \rho^{m-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{m-1} & \rho^{m-2} & \rho^{m-3} & \dots & 1 \end{bmatrix}. \tag{4.5}$$

Finalement, la covariance non structurée permet d'avoir un modèle totalement flexible et est donnée par :

$$\mathbf{R}_i = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \dots & \sigma_{1m}^2 \\ \sigma_{12}^2 & \sigma_{22}^2 & \dots & \sigma_{2m}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1m}^2 & \sigma_{2m}^2 & \dots & \sigma_{mm}^2 \end{bmatrix}. \tag{4.6}$$

Plusieurs articles dans la littérature utilisent cette structure pour sa flexibilité sans forcer d'hypothèse sur les données répétées. Nous allons voir que les variations dans le temps des scores de nos données sont très variables et complexes, c'est pourquoi nous utilisons aussi cette structure dans notre étude. On obtient alors la réponse totale :

$$\mathbf{y}_i \sim \mathcal{N}(\mathbf{X_i}\boldsymbol{\beta}, \mathbf{V}_i) \tag{4.7}$$

où

$$\mathbf{V}_i = \mathbf{Z}_i \mathbf{G} \mathbf{Z}_i^T + \mathbf{R}_i. \tag{4.8}$$

Après avoir spécifié le modèle, on utilise une méthode de vraisemblance pour optimiser l'ensemble des paramètres $\boldsymbol{\theta}$ en maximisant la vraisemblance $\mathcal{L}(\boldsymbol{\theta})$ suivante :

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{(2\pi)^{n/2}|\mathbf{V}|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{r}^\top \mathbf{V}^{-1}\mathbf{r}\right) \tag{4.9}$$

où $\mathbf{r} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$. On utilise généralement plutôt le logarithme de la vraisemblance, ce qui simplifie l'équation :

$$\log \mathcal{L}(\boldsymbol{\theta}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{V}| - \frac{1}{2}\mathbf{r}^\top \mathbf{V}^{-1}\mathbf{r}. \tag{4.10}$$

Cependant, cette méthode devient biaisée lorsque l'on ajoute des effets aléatoires. En effet, l'estimation de la variance se voit sous-estimée puisque son calcul repose sur la banque de données complète, ce qui inclut l'information provenant des effets fixes. Pour remédier à ce problème, nous utilisons la maximisation de la vraisemblance restreinte (REML). Cette méthode enlève l'influence des effets fixes sur la variable de réponse en la projetant sur l'espace orthogonal aux effets fixes (régression). On utilise ensuite les résidus de la variable de réponse, soit le reste de la variation, pour expliquer la variance du modèle. Ces deux étapes se font de manière itérative jusqu'à optimiser les paramètres. La vraisemblance à maximiser devient :

$$\log \mathcal{L}_{\text{REML}}(\boldsymbol{\theta}) = -\frac{n-p}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{V}| - \frac{1}{2}\log|\mathbf{X}^\top\mathbf{V}^{-1}\mathbf{X}| - \frac{1}{2}\mathbf{r}^\top\mathbf{V}^{-1}\mathbf{r} \tag{4.11}$$

où n est le nombre d'observations, p est le nombre d'effets fixes, $\mathbf{r}$ sont les résidus et les paramètres fixes $\hat{\boldsymbol{\beta}}$ sont donnés par :

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{V}^{-1}\mathbf{y}. \tag{4.12}$$

Pour optimiser les paramètres, on peut utiliser tout type de méthodes d'optimisation, comme celles de Newton avec l'estimation du gradient et de la matrice hessienne. On peut finalement utiliser un test de Wald pour vérifier la pertinence d'un paramètre dans le modèle avec les hypothèses H_0 et H_1 données par :

$$H_0 : \beta_j = \beta_0 \tag{4.13}$$

et

$$H_1 : \beta_j \neq \beta_0. \tag{4.14}$$

Pour ce faire, on calcule la statistique W définie comme suit :

$$W = \frac{\hat{\beta}_j^2}{\text{Var}(\hat{\beta}_j)}. \tag{4.15}$$

Sachant que W suit une distribution χ_1^2, on peut trouver la valeur-p pour l'hypothèse et ainsi valider si l'effet est pertinent au modèle.

Dans le cadre de notre étude, la matrice d'effets fixes $\mathbf{X}_i$ est composée de différentes variables et covariables d'intérêt pour les collaborateurs. Plus précisément, elle comprend le score de référence avant le début de l'étude, l'âge, le sexe, l'IMC, le groupe ainsi que la sévérité de la mutation génétique pour chaque patient. Comme indiqué dans la littérature, on considère généralement que le temps a un effet aléatoire, ce qui permet à chaque patient d'avoir sa propre pente dans le temps et son ordonnée à l'origine dans le modèle. Ensuite, une nouvelle variable mesurant l'interaction entre le groupe du patient et le temps est utilisé pour vérifier l'efficacité de la médication à l'aide du test de Wald.

Dans notre cas, où l'étude est réalisée en deux phases, dont une durant laquelle les deux groupes sont médicamentés, la vérification doit être effectuée différemment. Notre approche consiste à analyser l'ensemble des données de l'étude dans un seul MMRM. Nous conservons l'effet aléatoire lié à la variable temps pour que chaque patient ait sa propre pente dans le temps. Cependant, pour évaluer l'efficacité de la médication, nous ajoutons au modèle une variable indiquant le nombre de mois depuis le début du traitement pour chaque patient.

Cette variable, « temps de traitement », nous permet de réaliser un seul test de Wald pour évaluer l'efficacité de la médication sur toute la durée de l'étude. En effet, si le test donne une valeur-p inférieure à 0,05, nous pourrons conclure que le temps de traitement est significatif pour la variation du score, et donc que le traitement a eu un impact sur l'état des patients. Cette méthode nous permet d'utiliser l'ensemble des données de l'étude dans un seul modèle, sans avoir à effectuer trois tests distincts comme mentionné dans la littérature. La réalisation d'un unique test statistique pour évaluer l'efficacité réduit également le nombre total de tests requis lorsque nous procéderons à la dichotomisation et à l'évaluation des autres scores médicaux. Cela permet de limiter le risque de faux positifs lié au biais des multiples tests statistiques sur les données observées.

Nous avons également décidé de ne pas imputer les données manquantes. En effet, les données médicales étant souvent complexes, les modèles d'imputation produisent fréquemment des prédictions trop simplifiées et erronées, pouvant ainsi biaiser les résultats de l'analyse du modèle MMRM. Nous exploitons ainsi les capacités du MMRM à gérer les données en prenant en compte les valeurs manquantes sans les imputer. Nous avons aussi choisi d'arrêter l'analyse de nos données au mois 54, en conservant uniquement les patients présents jusqu'à ce point, car le nombre de patients diminue significativement après.

4.3 Résultats

Dans cette section, nous présenterons les résultats des analyses statistiques visant à vérifier l'efficacité du traitement de la molécule X. Pour ce faire, nous avons d'abord analysé l'ODCS - partie motrice, qui était le score le plus significatif pour notre étude. Par la suite, nous avons examiné l'impact de la médication sur les autres scores auxquels nous avions accès. Pour tous les scores, nous avons également effectué une dichotomisation afin de vérifier si certains sous-groupes répondaient mieux au traitement.

4.3.1 Analyse du ODCS - partie motrice

Dans un premier temps, nous avons vérifié l'efficacité du traitement sur le groupe complet en analysant la variation de la partie motrice. Plus ce score est élevé, plus le patient à un stade avancé de la maladie. La figure 4.1 permet de visualiser cette variation moyenne du ODCS - partie motrice dans le temps pour les deux groupes.

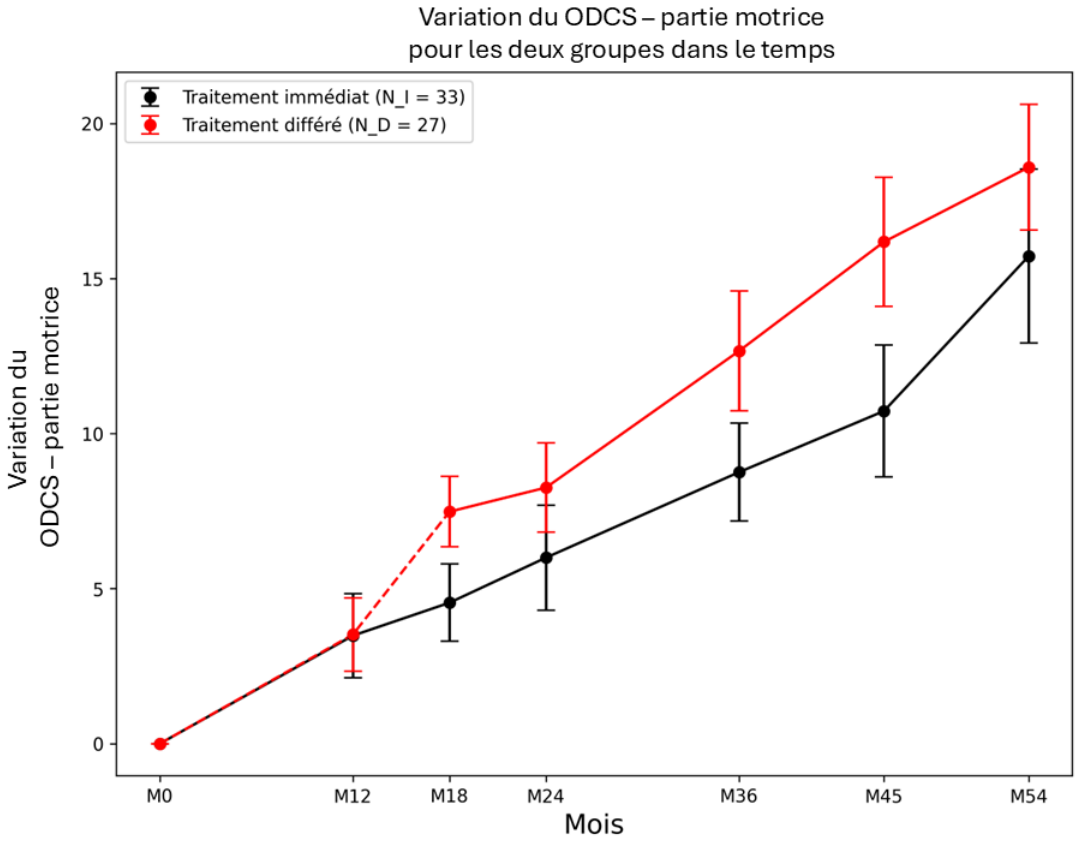

FIGURE 4.1 Variation du ODCS - partie motrice dans le temps pour les deux groupes, avec des barres d'erreur représentant l'erreur-type, et N_I et N_D indiquant le nombre de patients à chaque mois.

Dans un premier temps, si l'on observe la première partie de l'étude où les deux groupes sont distincts, une légère variation semble apparaître entre les mois 12 et 18. Cependant, cette différence est faible lorsque l'on prend en compte les grandes variances. Lorsque le deuxième groupe commence le traitement, cette différence semble s'accentuer au fil du temps, ce qui n'est pas tout à fait attendu puisque l'état moyen des patients devrait s'améliorer avec le traitement. Il est toutefois possible que l'effet du traitement prenne du temps à se manifester dans l'état des patients, ce qui pourrait expliquer ces variations. Le tableau 4.1 présente les tests de différences de moyennes pour chaque mois. Pour ce faire, nous avons d'abord vérifié l'égalité des variances à l'aide du test de Levene, puis utilisé le test approprié entre le T-test et le test de Mann–Whitney U. L'hypothèse alternative utilisée est que la moyenne du groupe différé est plus élevée que celle du groupe traité immédiatement.

Tableau 4.1 Résultats des tests statistiques simples de différence de distribution entre les deux groupes pour chaque mois.

Mois	Test	Valeur-p
M0	Mann-Whitney U	0.1127
M12	T-test	0.2298
M18	T-test	0.0889
M24	Mann-Whitney U	0.0850
M36	T-test	**0.0308**
M45	T-test	**0.0424**
M54	T-test	0.1387

Le tableau confirme ce que l'on observe dans le graphique : les mois 36 et 45 semblent bien présenter une différence significative de moyenne entre les deux groupes, avec des valeurs-p inférieures à 0,05, mais pas les autres mois. Cependant, cette approche par tests statistiques mois par mois n'est pas suffisante pour conclure sur l'efficacité de la médication. Nous devons maintenant utiliser notre approche par MMRM pour analyser chaque patient individuellement en intégrant les effets aléatoires et la corrélation temporelle. Le tableau 4.2 présente les résultats de notre modèle sur ces données.

Tableau 4.2 Résultats du modèle MMRM sur l'ODCS - partie motrice avec le groupe complet.

Variable	Valeur-p
Groupe traitement	0.53
Temps de traitement	0.372
Temps	**0**
IMC	0.957
Mutation	0.139
Âge	0.299
Référence	**0**
Sexe	0.955

Le modèle comprend différentes variables tel que mentionné précédemment, soit le groupe du patient, son IMC, la sévérité de la mutation génétique, son âge, son sexe, sa valeur du ODCS - partie motrice à la référence, le temps et le temps écoulé depuis le début de son traitement. Cette dernière variable est celle qui nous intéresse pour vérifier l'efficacité du traitement. On

constate dans un premier temps qu'elle n'est pas significative, avec une valeur-p de 0,372 ce qui voudrait dire que le traitement n'est pas efficace sur la cohorte. On remarque également que les deux seules variables significatives sont le temps et la valeur du ODCS - partie motrice à la référence, ce qui est attendu. Les autres covariables ne sont pas significatives dans le modèle, avec la sévérité de la mutation génétique ayant la plus petite valeur-p, soit 0,139.

À partir de ces résultats, on peut conclure que le traitement n'a pas d'effet significatif sur la population complète de notre étude. Nous nous sommes alors penchés sur des sous-groupes de patients pour voir si certaines caractéristiques pourraient les rendre plus sensibles au traitement. La première dichotomisation consiste à séparer nos patients en deux groupes selon la médiane du ODCS - partie motrice à la référence. Nous analysons ainsi les patients dans des stades peu avancés de la maladie d'une part, et ceux dans des stades avancés d'autre part. Les figures 4.2 et 4.3 présentent la variation du ODCS - partie motrice pour ces deux sous-groupes.

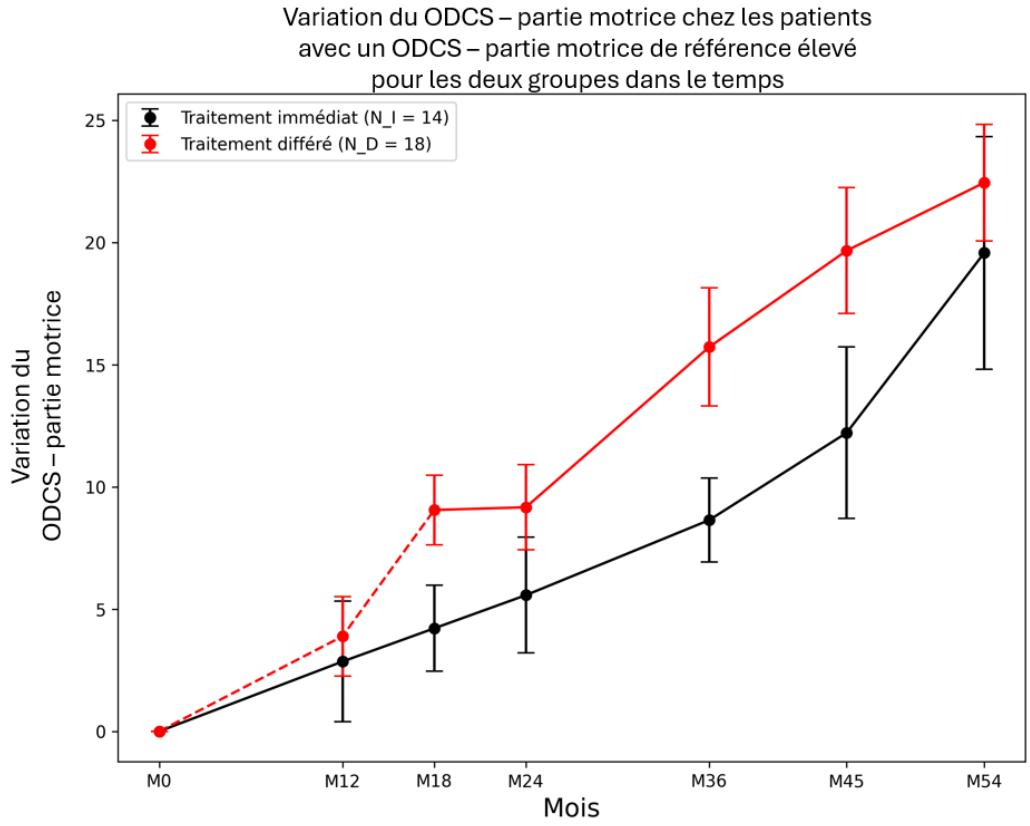

FIGURE 4.2 Variation du ODCS - partie motrice dans le temps pour les patients avec un ODCS - partie motrice de référence élevé, avec des barres d'erreur représentant l'erreur-type, et N_I et N_D indiquant le nombre de patients à chaque mois.

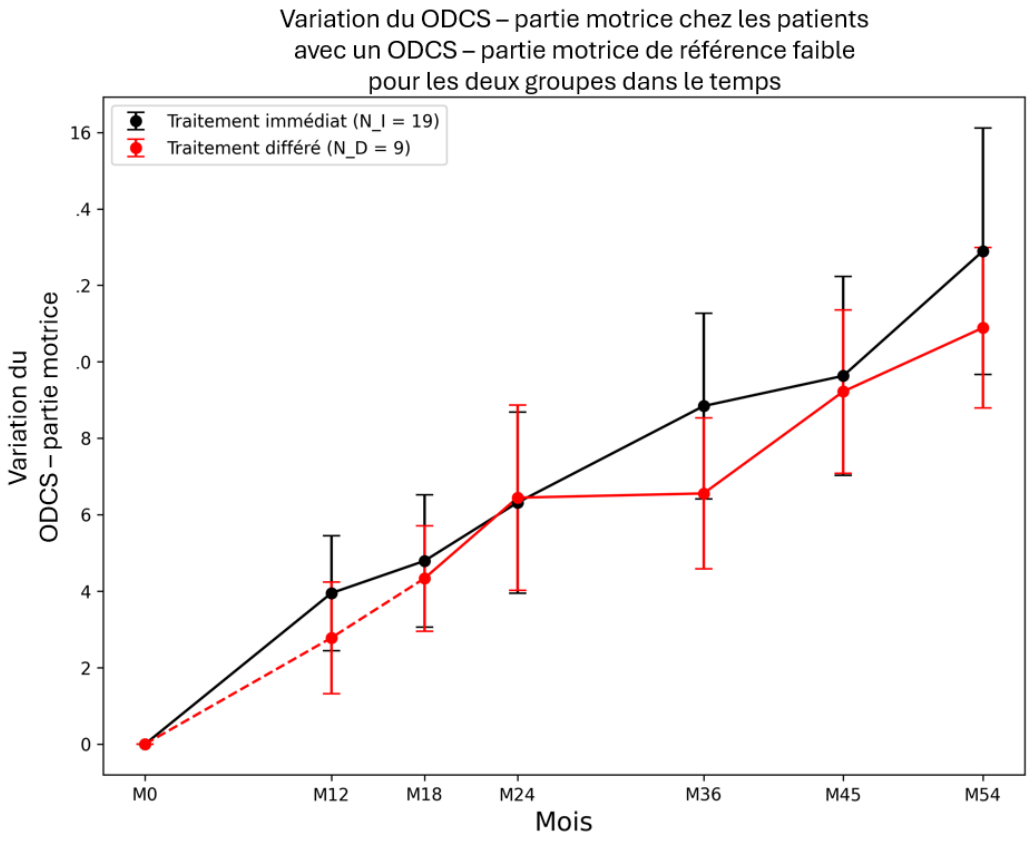

FIGURE 4.3 Variation du ODCS - partie motrice dans le temps pour les patients avec un ODCS - partie motrice de référence faible, avec des barres d'erreur représentant l'erreur-type, et N_I et N_D indiquant le nombre de patients à chaque mois.

On observe que le traitement ne semble avoir aucun effet sur les patients ayant un ODCS - partie motrice de référence faible, puisque la variation du score pour les deux groupes est similaire tout au long de l'étude. Cependant, pour les patients aux stades plus avancés de la maladie, une différence semble apparaître entre les deux groupes. En effet, à partir du mois 18, la moyenne des deux groupes est bien distincte pour le reste de l'étude. Toutefois, la variation du score semble s'accentuer de plus en plus rapidement dans le temps pour les patients traités depuis le début. Aussi, à l'exception du mois 24, la variation dans le temps ne semble pas s'améliorer pour les patients du deuxième groupe lorsqu'ils commencent le traitement. Pour vérifier ces observations, on peut voir les résultats des modèles MMRM pour ces deux sous-groupes dans le tableau 4.3.

Tableau 4.3 Résultats du modèle MMRM sur l'ODCS - partie motrice avec la dichotomisation sur l'ODCS - partie motrice.

Variable	Valeur-p avec l'ODCS - partie motrice élevé	Valeur-p avec l'ODCS - partie motrice faible
Groupe traitement	0.593	0.992
Temps de traitement	0.587	0.624
Temps	**0**	**0.023**
IMC	0.981	0.826
Mutation	**0.039**	0.482
Âge	0.21	0.761
Référence	**0**	**0**
Sexe	0.889	0.586

Ces résultats montrent que le traitement n'est effectivement pas significatif pour l'état des patients, même lorsque l'on dichotomise selon l'ODCS - partie motrice à la référence. Cependant, on remarque que la sévérité de la mutation génétique devient significative dans les modèles pour le sous-groupe avec un ODCS - partie motrice élevé, avec une valeur-p de 0,039. Pour cette raison, et puisque la sévérité de la mutation génétique est connue comme une variable importante liée à la maladie, notre prochaine dichotomisation utilise ce facteur. En effet, on sait que plus la mutation génétique est sévère, plus les symptômes se manifestent rapidement chez l'individu et plus ils sont prononcés. Les figures 4.4 et 4.5 présentent donc la variation du ODCS - partie motrice lorsque l'on dichotomise selon cette variable importante.

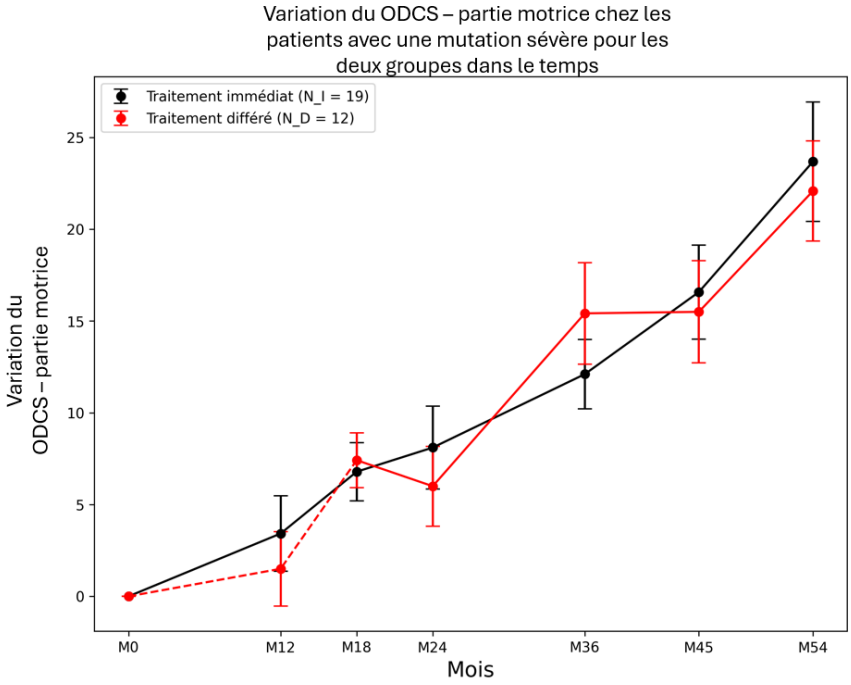

FIGURE 4.4 Variation du ODCS - partie motrice dans le temps pour les patients avec une mutation sévère, avec des barres d'erreur représentant l'erreur-type, et N_I et N_D indiquant le nombre de patients à chaque mois.

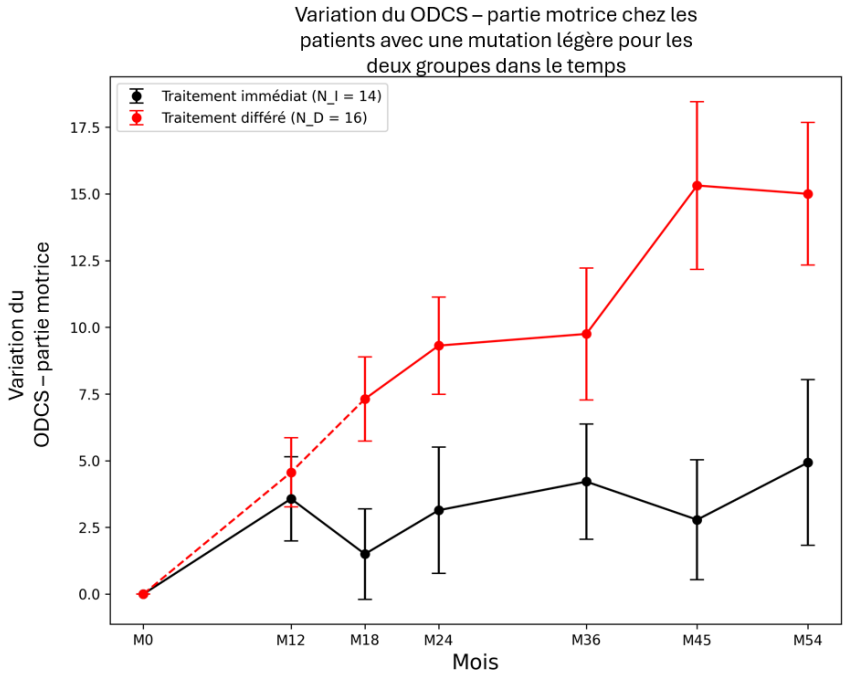

FIGURE 4.5 Variation du ODCS - partie motrice dans le temps pour les patients avec une mutation faible, avec des barres d'erreur représentant l'erreur-type, et N_I et N_D indiquant le nombre de patients à chaque mois.

On observe que les variations du ODCS - partie motrice dans le temps ne sont pas distinguables entre les deux groupes lorsque l'on prend seulement les patients ayant une mutation sévère. Cela semble donc montrer que le traitement n'est pas efficace pour les patients avec cette caractéristique. Cependant, si l'on regarde les patients avec une mutation plus légère, il y a une différence frappante entre les deux groupes. En effet, la variation moyenne du ODCS - partie motrice dans le temps semble stable pour le groupe traité depuis le début, tandis qu'elle ralentit pour le second groupe lorsque le traitement débute. Le traitement semble donc être efficace pour ce sous-groupe de patients. Le tableau 4.4 présente les résultats des modèles MMRM avec cette dichotomisation.

Tableau 4.4 Résultats du modèle MMRM sur l'ODCS - partie motrice avec la dichotomisation sur la mutation.

Variable	Valeur-p avec mutation sévère	Valeur-p avec mutation légère
Groupe traitement	0.345	0.746
Temps de traitement	0.451	**0.009**
Temps	**0.016**	**0**
IMC	0.991	0.947
Mutation	0.206	0.229
Âge	0.137	0.857
Référence	**0**	**0**
Sexe	0.83	0.918

Les résultats des modèles MMRM confirment ce que l'on observait sur les graphiques. En effet, on voit que le traitement n'est pas significatif dans le modèle lorsque l'on prend les patients avec une mutation sévère de la maladie. Cependant, lorsqu'elle est faible, le temps de traitement devient significatif pour la variation du ODCS - partie motrice dans le temps. On a donc trouvé un potentiel sous-groupe dont le traitement semble avoir un impact bénéfique non négligeable sur l'état des patients.

Par la suite, après avoir trouvé ce sous-groupe où le traitement est efficace, nous avons aussi testé d'autres dichotomisations qui sont intéressantes pour les collaborateurs. Nous avons donc dichotomisé en fonction du sexe, de l'âge, de l'IMC, l'échelle d'activités journalières à la référence, du poids et du niveau de fonctionnalité. Toutefois, aucun des sous-groupes créés n'obtenait des résultats où le traitement était significatif à la variation du ODCS - partie motrice dans le temps. La mutation génétique était quant à elle souvent significative, illustrant son importance pour la maladie.

4.3.2 Analyse des autres scores

Nous avons ensuite analysé la variation des autres scores dans le temps. Parmi ceux-ci, nous avons le sous-score des symptômes moteurs 1, le sous-score des symptômes moteurs 2, le sous-score des symptômes moteurs 3, le sous-score des symptômes moteurs 4, l'échelle d'activités journalières, le score cognitif, l'échelle de démence de Mattis et le niveau de fonctionnalité. Pour tous ces scores, nous avons aussi testé les mêmes dichotomisations qu'avec l'ODCS - partie motrice. Étant donné que nous disposons de huit scores différents et de plusieurs dichotomisations, nous présenterons uniquement ici les résultats qui se sont révélés concluants. Cependant, ce grand nombre de tests peut avoir créé des faux positifs parmi les résultats, ce problème sera discuté davantage dans le prochain chapitre.

Dans un premier temps, nous avons remarqué que le traitement était efficace sur la variation de la majorité des sous-scores formant l'ODCS - partie motrice lorsque l'on prenait les patients avec une mutation génétique légère. En effet, le sous-score des symptômes moteurs 1, le sous-score des symptômes moteurs 2 et le sous-score des symptômes moteurs 3 sont affectés significativement par le traitement lorsque l'on regarde ce sous-groupe. Ce résultat n'est cependant pas surprenant sachant que l'addition de ces sous-scores (avec le sous-score symptômes moteurs 4) forment l'ODCS - partie motrice.

Par la suite, en séparant les patients selon leur genre, le traitement devient aussi significatif sur certains scores. La figure 4.6 présente la variation du sous-score des symptômes moteurs 2 chez les femmes où l'on observe un potentiel effet du traitement.

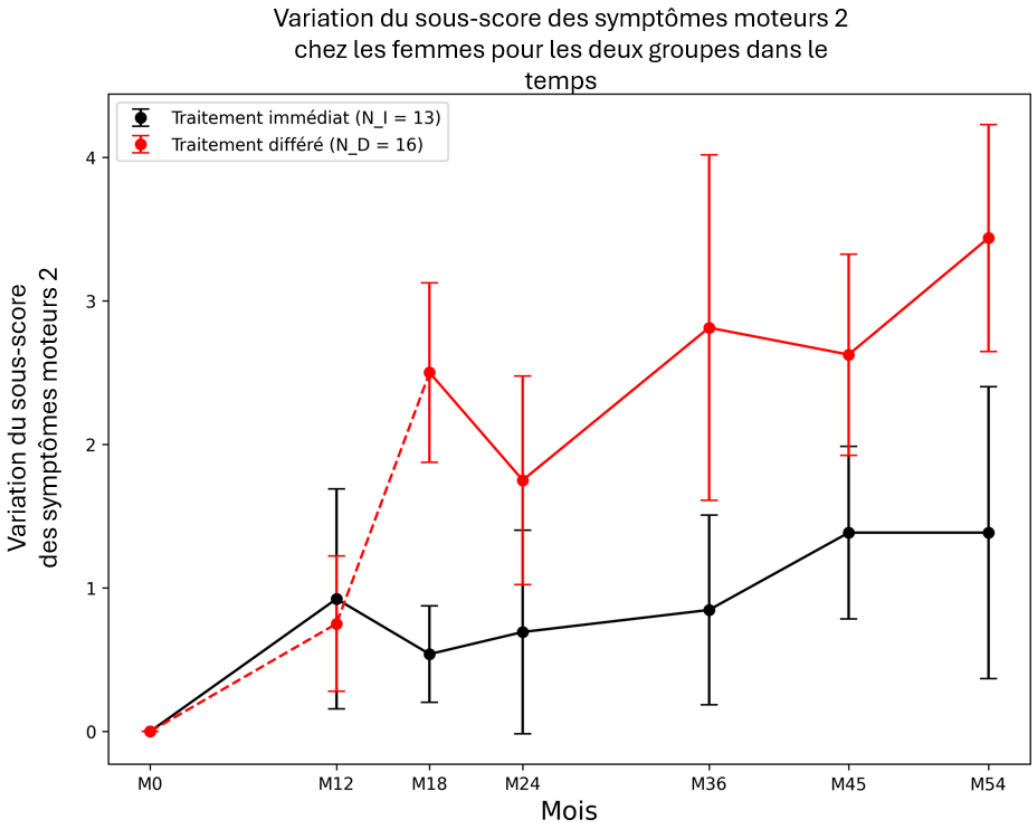

FIGURE 4.6 Variation du sous-score des symptômes moteurs 2 dans le temps chez les femmes, avec des barres d'erreur représentant l'erreur-type, et N_I et N_D indiquant le nombre de patients à chaque mois.

On peut confirmer cette conclusion à l'aide des résultats du modèle MMRM du tableau 4.5 où, effectivement, la valeur-p reliée à la variable du temps de traitement est de 0.038. On peut faire de même pour les hommes, mais cette fois-ci en utilisant le sous-score des symptômes moteurs 3 qui se voit affecté par le traitement. La figure 4.7 présente la variation de ce sous-score chez les hommes, cependant, il est difficile de conclure seulement visuellement pour ce score. Le tableau 4.6 présente les résultats du modèle en détail et permet de conclure que le traitement est efficace pour ce type de symptôme chez les hommes avec une valeur-p de 0.026.

TABLEAU 4.5 Résultats du modèle MMRM sur le sous-score des symptômes moteurs 2 chez les femmes.

Variable	Valeur-p
Groupe traitement	0.965
Temps de traitement	**0.038**
Temps	**0.002**
IMC	0.781
Mutation	0.176
Âge	0.734
Référence	**0**

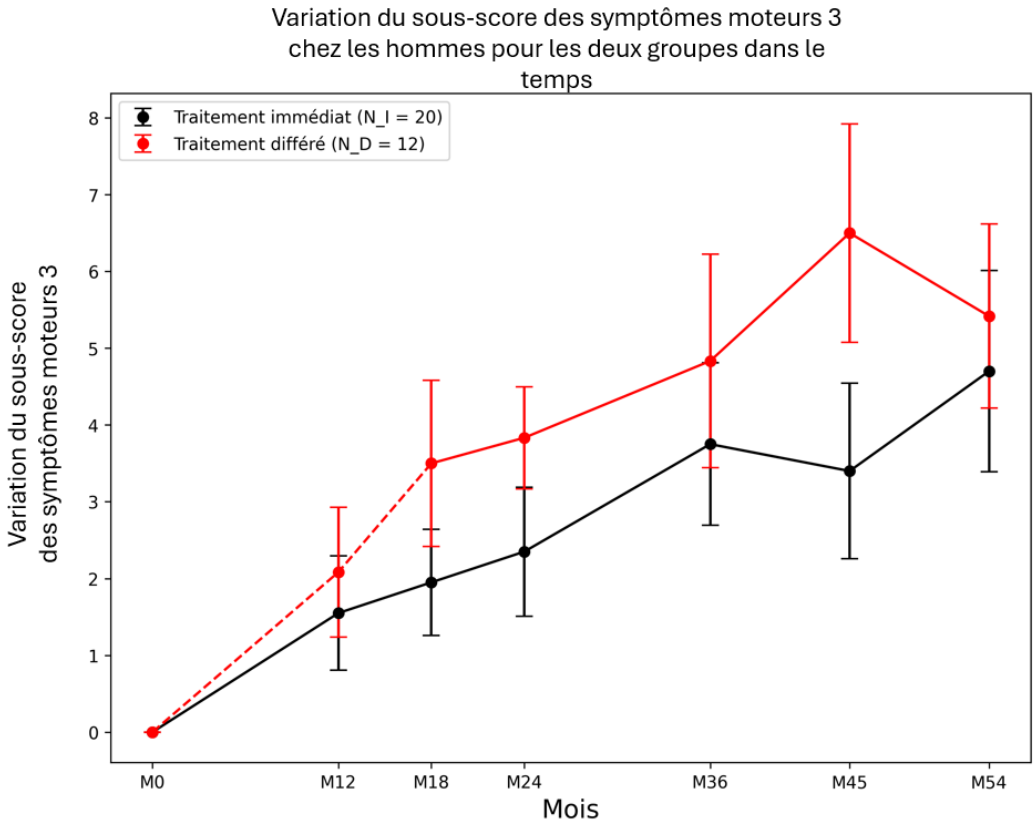

FIGURE 4.7 Variation du sous-score des symptômes moteurs 3 dans le temps chez les hommes, avec des barres d'erreur représentant l'erreur-type, et N_I et N_D indiquant le nombre de patients à chaque mois.

Tableau 4.6 Résultats du modèle MMRM sur le sous-score des symptômes moteurs 3 chez les hommes.

Variable	Valeur-p
Groupe traitement	0.65
Temps de traitement	**0.026**
Temps	**0**
IMC	0.944
Mutation	**0.047**
Âge	0.084
Référence	**0**

Par la suite, nous avons aussi remarqué que l'IMC des patients pouvait avoir un impact sur l'efficacité du traitement, puisque ce facteur peut empirer l'état de santé général des patients. En effet, si l'on isole les patients avec un IMC élevé, on peut voir à la figure 4.8 que le traitement semble être significatif sur la variation du niveau de fonctionnalité dans le temps. On voit le même genre de comportement sur ce sous-groupe lorsque l'on regarde la variation de l'échelle d'activités journalières dans le temps. Celle-ci est présentée à la figure 4.9. Une diminution de ces scores représente une détérioration de l'état du patient. Ces observations sont confirmées par nos modèles MMRM avec les résultats présentés au tableau 4.7, où l'on peut voir une valeur-p de 0.02 pour le score du niveau de fonctionnalité et de 0.028 pour l'échelle d'activités journalières.

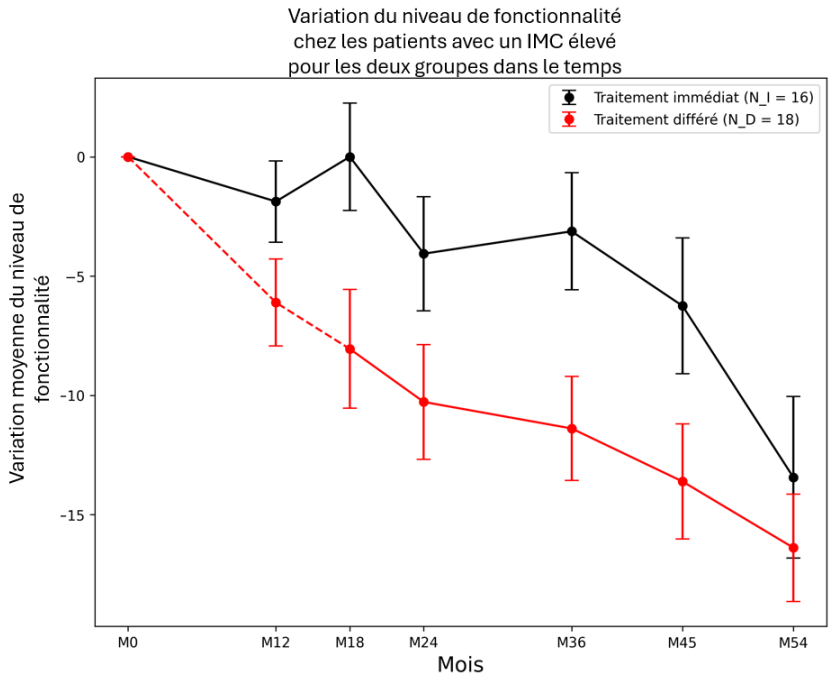

FIGURE 4.8 Variation du niveau de fonctionnalité chez les patients avec un IMC élevé, avec des barres d'erreur représentant l'erreur-type, et N_I et N_D indiquant le nombre de patients à chaque mois.

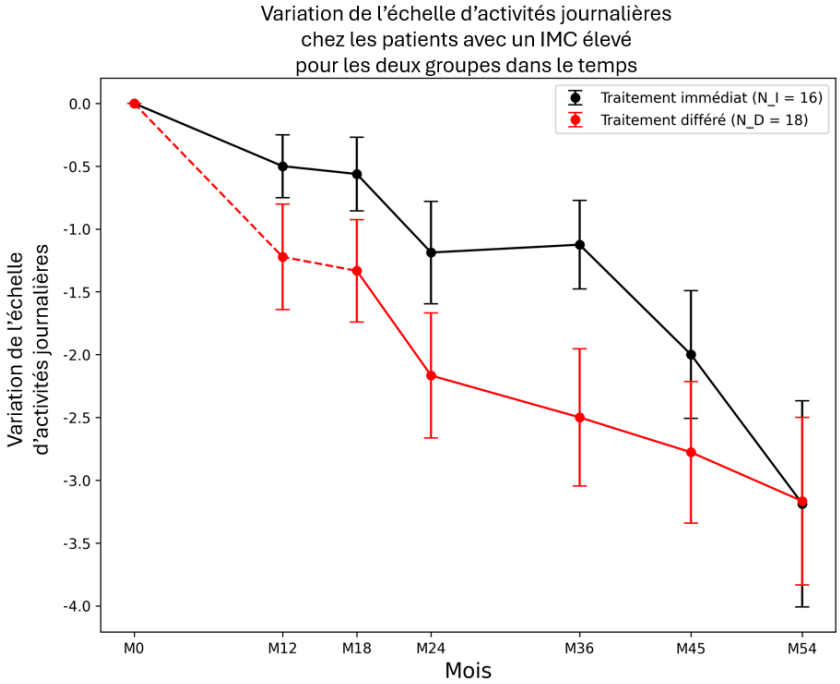

FIGURE 4.9 Variation de l'échelle d'activités journalières chez les patients avec un IMC élevé, avec des barres d'erreur représentant l'erreur-type, et N_I et N_D indiquant le nombre de patients à chaque mois.

Tableau 4.7 Résultats du modèle MMRM sur le niveau de fonctionnalité et l'échelle d'activités journalières chez les patients avec un IMC élevé.

Variable	Valeur-p pour le niveau de fonctionnalité	Valeur-p pour l'échelle d'activités journalières
Groupe traitement	0.304	0.401
Temps de traitement	**0.02**	**0.028**
Temps	**0**	**0**
IMC	0.888	0.201
Mutation	0.47	0.543
Âge	0.078	0.711
Référence	-	**0**
Sexe	0.502	0.304

D'autres scores étaient affectés significativement par le traitement sous certaines dichotomisations. Cependant, ils ne sont pas présentés ici puisqu'ils ne possédaient pas assez d'échantillons dans les sous-groupes utilisés pour être considérés dans l'étude.

4.4 Discussion des résultats

Dans un premier temps, nous avons utilisé le modèle pour vérifier l'efficacité du traitement sur l'indice clinique ODCS - partie motrice. Nous avons remarqué que le traitement ne pouvait pas être considéré comme significatif, car la valeur-p de la variable d'intérêt était de 0.372, ce qui est plus élevé que le seuil de 0.05 recherché. Nous avons ensuite procédé à des dichotomisations selon différents facteurs cliniques pour tester l'effet de la médication sur différents sous-groupes. Cela nous a permis de découvrir que les patients avec une mutation légère semblaient être affectés positivement et de manière significative par le traitement avec une valeur-p de 0.009.

Par la suite, nous avons appliqué ces mêmes étapes sur d'autres indices cliniques moins importants de cette maladie neurologique de type génétique. On peut alors observer que le sexe semble avoir un impact sur différents sous-scores. Par exemple, les hommes sont affectés significativement par le traitement selon le score des symptômes moteurs 3, tandis que chez les femmes, il est significatif sur le score des symptômes moteurs 2. On remarque aussi que l'IMC peut être un critère de dichotomisation important au traitement. En effet, on a déterminé que la médication avait un effet significatif sur les patients ayant un IMC élevé selon le niveau de fonctionnalité et l'échelle d'activités journalières.

Il est toutefois important d'interpréter ces résultats avec prudence pour plusieurs raisons. Premièrement, pour identifier les sous-groupes et sous-scores où la médication est significative, nous avons dû tester près de cent combinaisons différentes, soit environ cent modèles avec des tests statistiques. Comme nous avons abordé cette étude de manière exploratoire, nous n'avons pas appliqué de correction pour tenir compte de l'inflation du risque d'erreur de type I. Cependant, il serait nécessaire d'appliquer des corrections dans le cadre d'une étude plus approfondie visant à prouver définitivement l'efficacité de la médication. Nous avons également un faible nombre de patients, car nous avons examiné l'étude sur une longue période de 54 mois et dichotomisé le groupe, ce qui réduit la puissance statistique de nos modèles. Enfin, ce faible nombre de patients, combiné à la grande variabilité des scores de chaque patient dans le temps, compliquait la convergence des modèles avec certains d'entre eux incapables de converger. Les valeurs-p pouvaient alors varier en fonction du type de méthode d'optimisation utilisé pour estimer les paramètres. Néanmoins, les résultats sont concluants et semblent indiquer qu'il serait pertinent de poursuivre une étude plus rigoureuse sur l'efficacité de ce traitement en utilisant des groupes plus homogènes et avec des caractéristiques spécifiques.

CHAPITRE 5 CONCLUSION

5.1 Synthèse des travaux

Le but de notre premier projet était de valider l'existence de biomarqueurs de la maladie de Parkinson au sein du protéome des VEE dans le sang. Pour y parvenir, nous avons opté pour une approche de classification par apprentissage automatique, sachant que la quantité de données était trop importante pour recourir à de simples analyses statistiques. Cependant, cette tâche présentait aussi son lot de défis : le grand nombre de variables et le faible nombre d'échantillons, les données bruitées et corrélées, les valeurs manquantes, ainsi que la nécessité d'avoir des modèles interprétables. Pour atteindre notre objectif tout en atténuant l'impact de ces problématiques, nous avons tenté d'optimiser l'ensemble des étapes de l'approche systématique de l'apprentissage automatique. Nous avons également créé quatre méthodes originales : l'imputation flexible, le sous-échantillonnage par erreurs de reconstruction et par prototypes, et le classificateur PPI. La méthode d'imputation repose sur l'hypothèse souple que la majorité des valeurs manquantes sont de type MNAR en raison d'un seuil de détection. La première méthode de sous-échantillonnage permet de réduire le nombre d'échantillons trop bruités, tandis que la seconde cherche à identifier des sous-groupes d'échantillons représentatifs de leur classe pour accroître l'interprétabilité. Enfin, le modèle de classification intègre les interactions protéine-protéine et le regroupement pour réduire le nombre de poids du réseau et accroître l'interprétabilité grâce à une mixture d'experts. Nous obtenons une PR-AUC maximale de 0.6847, alors que celle de référence est d'environ 0.5646, et des précisions globales maximales d'environ 58 %. Nos modèles semblent alors indiquer qu'il n'existe pas de biomarqueurs dans le protéome des VEE dans le sang pour la maladie de Parkinson. En effet, nos tentatives d'inférence n'ont pas été fructueuses, car chaque prédiction effectuée par les modèles utilisait l'ensemble des protéines en raison du surapprentissage. Nous avons néanmoins observé que trois de nos méthodes originales, à savoir l'imputation flexible, le sous-échantillonnage par erreurs de reconstruction et le classificateur PPI, amélioraient les performances de classification et potentiellement l'interprétabilité des modèles.

Plus récemment, nous avons eu la chance de recevoir de nouvelles données sur la même cohorte de patients, avec lesquelles nous obtenons de meilleures performances de classification. Cela semble donc indiquer que les données utilisées dans cette étude étaient trop bruitées ou qu'il y a potentiellement eu des problèmes lors de la prise des mesures. Ce ne sont donc pas nos modèles qui limitaient les performances, mais plutôt les données.

Pour le second projet, notre objectif était de vérifier l'efficacité d'un nouveau traitement pour

une maladie neurologique de type génétique dans une étude à démarrage différé. Un groupe a pris un placebo durant les 18 premiers mois, puis la médication, tandis que l'autre groupe a été traité durant toute la durée de l'étude. Il s'agissait d'une étude exploratoire visant à déterminer s'il serait pertinent pour nos collaborateurs de poursuivre vers des études plus approfondies. Pour ce faire, nous avons utilisé une approche basée sur les MMRM, qui permet de poser des effets fixes et aléatoires et de prendre en compte la dépendance temporelle des données longitudinales pour chaque patient. Nous avons opté pour une approche originale, plus simple que celle décrite dans la littérature, en intégrant une variable mesurant le temps de traitement au modèle afin d'évaluer l'impact de celui-ci à l'aide d'un unique test statistique. Cette méthode nous a permis d'examiner de manière plus ciblée les effets du traitement sur de nombreux indices cliniques associés à cette maladie neurologique. Nous avons ainsi identifié des sous-groupes de patients présentant des caractéristiques distinctes pour lesquels le traitement semble efficace sur différents symptômes de la maladie. La caractéristique la plus pertinente semble être la sévérité de la mutation génétique. En effet, lorsqu'elle est légère, le traitement devient significatif sur plusieurs indices cliniques importants de la maladie. Notre approche a permis de mettre en évidence ces cas intéressants au sein d'un seul modèle statistique, sans avoir à diviser l'étude en différentes sections en raison du démarrage différé. Ces résultats prometteurs pourraient également permettre le début d'une nouvelle étude clinique plus rigoureuse sur la médication.

5.2 Limitations des solutions proposées

Comme les résultats du projet sur la maladie de Parkinson l'ont révélé, l'approche par apprentissage automatique présente des limites. En dépit de nos efforts d'optimisation, les résultats sont restés peu concluants, suggérant un manque de corrélation entre les protéines et la maladie. Nos méthodes originales présentent également des aspects à améliorer. La méthode d'imputation flexible est spécifiquement adaptée aux cas où la majorité des valeurs manquantes sont de type MNAR et proviennent d'un seuil de détection. Quant à la méthode de sous-échantillonnage par prototypes, elle semble peu adaptée à cette tâche, nécessitant des problèmes de moindre complexité ou un nombre de variables comparables au nombre d'échantillons restants pour préserver les performances. En revanche, le sous-échantillonnage par erreurs de reconstruction peut s'appliquer à des problématiques variées puisqu'on utilise des réseaux de neurones. Cette approche est toutefois très sensible aux hyperparamètres. En effet, une architecture trop complexe conduit le modèle à surapprendre les données, ce qui réduit toutes les erreurs de reconstruction et empêche l'identification d'échantillons aberrants. À l'inverse, une architecture trop simple conduit à une erreur élevée, retirant un nombre ex-

cessif d'échantillons. De plus, cette méthode est coûteuse en temps de calcul, car elle nécessite l'entraînement de K modèles pour un gain limité en performance.

Enfin, le classificateur PPI est particulièrement spécifique, s'appliquant uniquement à des ensembles de données d'intensité de protéines. La couche de sélection, dans notre cas, s'est révélée trop sensible à la complexité du problème, ne permettant pas de sélection adéquate : toutes les protéines ont été utilisées pour la prédiction. De même, la méthode de regroupement MCL est restrictive, car elle n'offre pas la possibilité de définir le nombre de protéines par groupe. Notre modèle tend alors à privilégier les experts avec plus de variables dans leurs groupes, bien que certains petits groupes puissent présenter un intérêt biologique particulier.

Pour le projet sur la maladie neurologique de type génétique, notre approche basée sur le temps de traitement présente des limitations. Cette variable est corrélée avec la variable du temps, ce qui peut nuire à la stabilité de l'estimation des coefficients et à l'interprétabilité. Dans certains cas, il est possible que la variable du temps perde sa signification statistique, tandis que celle du temps de traitement demeure pertinente, ce qui ne reflète pas la réalité. Une attention particulière est donc nécessaire lors de l'interprétation des résultats, bien que cette corrélation existe également dans les approches examinées dans la revue de littérature.

5.3 Améliorations futures

Dans un premier temps, il serait intéressant de tester nos approches originales dans le cadre d'une approche systématique d'apprentissage machine sur une base de données répondant à nos hypothèses, mais présentant de meilleures performances. Cela permettrait une analyse des résultats plus simplifiée, évitant ainsi d'effectuer de multiples boucles de validation. Nous pourrions alors approfondir l'analyse de l'impact des hyperparamètres sur les performances de nos méthodes. Cette étape est particulièrement importante pour la méthode de sous-échantillonnage par erreurs de reconstruction, car elle est très sensible aux hyperparamètres. Une optimisation adéquate pourrait réduire le temps de calcul tout en maximisant le nombre d'échantillons aberrants identifiés avec précision. Concernant la méthode d'imputation flexible, il serait pertinent d'évaluer différentes distributions probabilistes pour l'algorithme EM ainsi que diverses méthodes de régression après l'imputation par EM. Pour améliorer le sous-échantillonnage par prototypes, il pourrait être envisagé d'explorer d'autres méthodes d'optimisation que l'algorithme génétique, telles que l'optimisation par essaims particulaires. Une réduction de dimension pourrait également être appliquée pour atténuer le problème lié à la forte dimensionnalité dans les calculs de distances tout en maintenant un niveau d'interprétabilité adéquat. En ce qui concerne le classificateur PPI, sa principale faiblesse semble résider actuellement dans la couche de sélection. Une alternative pourrait

consister à la modifier pour intégrer une couche avec une variable probabiliste qui sélectionne les variables pertinentes de manière aléatoire, pondérée selon les scores d'importance des protéines. En outre, recourir à une méthode de regroupement adaptée au graphe, permettant de spécifier le nombre de variables par groupe, éliminerait le biais mentionné précédemment dans le mélange d'experts.

En conclusion, la création de données synthétiques s'avérerait judicieuse pour évaluer les limites de notre méthode d'évaluation de la médication à l'aide de la variable de temps de traitement. Cela permettrait d'étudier l'effet de la corrélation entre le temps et le temps de traitement, de comparer les résultats avec les méthodes en trois parties identifiées dans la littérature, et d'évaluer l'impact de l'ajout d'un effet aléatoire sur le temps de traitement. Dans le cas de dichotomisations, intégrer une correction des valeurs-p obtenues via les tests de Wald pourrait limiter l'erreur de type I due à la multiplicité des tests.

RÉFÉRENCES

[1] R. N. Lamptey *et al.*, "A review of the common neurodegenerative disorders : current therapeutic approaches and the potential role of nanotherapeutics," *International journal of molecular sciences*, vol. 23, n°. 3, p. 1851, 2022.

[2] J. W. Krellman et G. Mercuri, "Cognitive interventions for neurodegenerative disease," *Current Neurology and Neuroscience Reports*, vol. 23, n°. 9, p. 461–468, 2023.

[3] J. Van Schependom et M. D'haeseleer, "Advances in neurodegenerative diseases," p. 1709, 2023.

[4] C. Marras *et al.*, "Prevalence of parkinson's disease across north america," *NPJ Parkinson's disease*, vol. 4, n°. 1, p. 21, 2018.

[5] M. A. Better, "Alzheimer's disease facts and figures," *Alzheimer's Dement*, vol. 20, p. 3708–3821, 2024.

[6] D. Koníčková *et al.*, "Biomarkers of neurodegenerative diseases : biology, taxonomy, clinical relevance, and current research status," *Biomedicines*, vol. 10, n°. 7, p. 1760, 2022.

[7] G. G. Kovacs, "Current concepts of neurodegenerative diseases," *Emj Neurol*, vol. 1, n°. 1, p. 10–11, 2014.

[8] A. Schumacher-Schuh *et al.*, "Advances in proteomic and metabolomic profiling of neurodegenerative diseases," *Frontiers in Neurology*, vol. 12, p. 792227, 2022.

[9] N. Alexander *et al.*, "Using unsupervised learning to identify clinical subtypes of alzheimer's disease in electronic health records," *Studies in health technology and informatics*, vol. 270, p. 499–503, 2020.

[10] A. Satt *et al.*, "Speech-based automatic and robust detection of very early dementia," dans *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[11] K. C. Fraser *et al.*, "An analysis of eye-movements during reading for the detection of mild cognitive impairment," dans *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, p. 1016–1026.

[12] J. Wang *et al.*, "Towards automatic detection of amyotrophic lateral sclerosis from speech acoustic and articulatory samples." dans *Interspeech*, 2016, p. 1195–1199.

[13] I. Bhattacharya et M. P. S. Bhatia, "Svm classification to distinguish parkinson disease patients," dans *Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India*, 2010, p. 1–6.

[14] J. C. Vásquez-Correa *et al.*, "Multimodal assessment of parkinson's disease : a deep learning approach," *IEEE journal of biomedical and health informatics*, vol. 23, n°. 4, p. 1618–1630, 2018.

[15] A. Samà *et al.*, "Estimating bradykinesia severity in parkinson's disease by analysing gait through a waist-worn sensor," *Computers in biology and medicine*, vol. 84, p. 114–123, 2017.

[16] Y. Xia *et al.*, "A novel approach for analysis of altered gait variability in amyotrophic lateral sclerosis," *Medical & biological engineering & computing*, vol. 54, p. 1399–1408, 2016.

[17] S. Dutta, A. Chatterjee et S. Munshi, "An automated hierarchical gait pattern identification tool employing cross-correlation-based feature extraction and recurrent neural network based classification," *Expert systems*, vol. 26, n°. 2, p. 202–217, 2009.

[18] P. Drotár *et al.*, "Evaluation of handwriting kinematics and pressure for differential diagnosis of parkinson's disease," *Artificial intelligence in Medicine*, vol. 67, p. 39–46, 2016.

[19] S. Xu et Z. Pan, "A novel ensemble of random forest for assisting diagnosis of parkinson's disease on small handwritten dynamics dataset," *International Journal of Medical Informatics*, vol. 144, p. 104283, 2020.

[20] C. Taleb *et al.*, "Feature selection for an improved parkinson's disease identification based on handwriting," dans *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*. IEEE, 2017, p. 52–56.

[21] C. R. Pereira *et al.*, "Handwritten dynamics assessment through convolutional neural networks : An application to parkinson's disease identification," *Artificial intelligence in medicine*, vol. 87, p. 67–77, 2018.

[22] H. Zhang *et al.*, "Deep learning identifies digital biomarkers for self-reported parkinson's disease," *Patterns*, vol. 1, n°. 3, 2020.

[23] A. Zhan *et al.*, "High frequency remote monitoring of parkinson's disease via smartphone : Platform overview and medication response detection," *arXiv preprint arXiv :1601.00960*, 2016.

[24] R. Cassani *et al.*, "Towards automated electroencephalography-based alzheimer's disease diagnosis using portable low-density devices," *Biomedical Signal Processing and Control*, vol. 33, p. 261–271, 2017.

[25] O. F. Odish *et al.*, "Eeg may serve as a biomarker in huntington's disease using machine learning automatic classification," *Scientific Reports*, vol. 8, n°. 1, p. 16090, 2018.

[26] B. F. O. Coelho *et al.*, "Parkinson's disease effective biomarkers based on hjorth features improved by machine learning," *Expert Systems with Applications*, vol. 212, p. 118772, 2023.

[27] M. Dauwan *et al.*, "Random forest to differentiate dementia with lewy bodies from alzheimer's disease," *Alzheimer's & Dementia : Diagnosis, Assessment & Disease Monitoring*, vol. 4, p. 99–106, 2016.

[28] M. I. Vanegas *et al.*, "Machine learning for eeg-based biomarkers in parkinson's disease," dans *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2018, p. 2661–2665.

[29] J. C. McBride *et al.*, "Sugihara causality analysis of scalp eeg for detection of early alzheimer's disease," *NeuroImage : Clinical*, vol. 7, p. 258–265, 2015.

[30] L. Trambaiolli *et al.*, "Feature selection before eeg classification supports the diagnosis of alzheimer's disease," *Clinical Neurophysiology*, vol. 128, nº. 10, p. 2058–2067, 2017.

[31] S. L. Mason *et al.*, "Predicting clinical diagnosis in huntington's disease : An imaging polymarker," *Annals of neurology*, vol. 83, nº. 3, p. 532–543, 2018.

[32] A. Rizk-Jackson *et al.*, "Evaluating imaging biomarkers for neurodegeneration in presymptomatic huntington's disease using machine learning techniques," *Neuroimage*, vol. 56, nº. 2, p. 788–796, 2011.

[33] G. Singh et L. Samavedham, "Unsupervised learning based feature extraction for differential diagnosis of neurodegenerative diseases : a case study on early-stage diagnosis of parkinson disease," *Journal of neuroscience methods*, vol. 256, p. 30–40, 2015.

[34] Z.-Y. Shu *et al.*, "Predicting the progression of parkinson's disease using conventional mri and machine learning : An application of radiomic biomarkers in whole-brain white matter," *Magnetic resonance in medicine*, vol. 85, nº. 3, p. 1611–1624, 2021.

[35] Y. Wu *et al.*, "Use of radiomic features and support vector machine to distinguish parkinson's disease cases from normal controls," *Annals of translational medicine*, vol. 7, nº. 23, 2019.

[36] R. U. Khan *et al.*, "A novel method for the classification of alzheimer's disease from normal controls using magnetic resonance imaging," *Expert Systems*, vol. 38, nº. 1, p. e12566, 2021.

[37] E. Moradi *et al.*, "Machine learning framework for early mri-based alzheimer's conversion prediction in mci subjects," *Neuroimage*, vol. 104, p. 398–412, 2015.

[38] G. Solana-Lavalle et R. Rosas-Romero, "Classification of ppmi mri scans with voxel-based morphometry and machine learning to assist in the diagnosis of parkinson's disease," *Computer Methods and Programs in Biomedicine*, vol. 198, p. 105793, 2021.

[39] G. Battineni *et al.*, "A comprehensive machine-learning model applied to magnetic resonance imaging (mri) to predict alzheimer's disease (ad) in older subjects," *Journal of Clinical Medicine*, vol. 9, n°. 7, p. 2146, 2020.

[40] B. Peng *et al.*, "A multilevel-roi-features-based machine learning method for detection of morphometric biomarkers in parkinson's disease," *Neuroscience letters*, vol. 651, p. 88–94, 2017.

[41] A. Mozhdehfarahbakhsh *et al.*, "An mri-based deep learning model to predict parkinson's disease stages," *medRxiv*, p. 2021–02, 2021.

[42] H. Zhao *et al.*, "Deep learning based diagnosis of parkinson's disease using diffusion magnetic resonance imaging," *Brain imaging and behavior*, vol. 16, n°. 4, p. 1749–1760, 2022.

[43] A. W. Salehi *et al.*, "A cnn model : earlier diagnosis and classification of alzheimer disease using mri," dans *2020 International Conference on Smart Electronics and Communication (ICOSEC)*. IEEE, 2020, p. 156–161.

[44] A. Farooq *et al.*, "A deep cnn based multi-class classification of alzheimer's disease using mri," dans *2017 IEEE International Conference on Imaging systems and techniques (IST)*. IEEE, 2017, p. 1–6.

[45] S. Sarraf et G. Tofighi, "Classification of alzheimer's disease structural mri data by deep learning convolutional neural networks," *arXiv preprint arXiv :1607.06583*, 2016.

[46] J. Islam et Y. Zhang, "A novel deep learning based multi-class classification method for alzheimer's disease detection using brain mri data," dans *Brain Informatics : International Conference, BI 2017, Beijing, China, November 16-18, 2017, Proceedings*. Springer, 2017, p. 213–222.

[47] B. B. Misra *et al.*, "Integrated omics : tools, advances and future approaches," *Journal of molecular endocrinology*, vol. 62, n°. 1, p. R21–R45, 2019.

[48] Z. Ahmed *et al.*, "Artificial intelligence for omics data analysis," *BMC Methods*, vol. 1, n°. 1, p. 4, 2024.

[49] W. Zhang *et al.*, "Blood ssr1 : A possible biomarker for early prediction of parkinson's disease," *Frontiers in Molecular Neuroscience*, vol. 15, p. 762544, 2022.

[50] E. Pantaleo *et al.*, "A machine learning approach to parkinson's disease blood transcriptomics," *Genes*, vol. 13, n°. 5, p. 727, 2022.

[51] C. J. Huseby *et al.*, "Blood transcript biomarkers selected by machine learning algorithm classify neurodegenerative diseases including alzheimer's disease," *Biomolecules*, vol. 12, n°. 11, p. 1592, 2022.

[52] M. A. Myszczynska *et al.*, "Applications of machine learning to diagnosis and treatment of neurodegenerative diseases," *Nature reviews neurology*, vol. 16, n°. 8, p. 440–456, 2020.

[53] T. Lee et H. Lee, "Prediction of alzheimer's disease using blood gene expression data," *Scientific reports*, vol. 10, n°. 1, p. 3485, 2020.

[54] N. Bhandari *et al.*, "Integrative gene expression analysis for the diagnosis of parkinson's disease using machine learning and explainable ai," *Computers in Biology and Medicine*, vol. 163, p. 107140, 2023.

[55] C. Park, J. Ha et S. Park, "Prediction of alzheimer's disease based on deep neural network by integrating gene expression and dna methylation dataset," *Expert Systems with Applications*, vol. 140, p. 112873, 2020.

[56] C. Maj *et al.*, "Integration of machine learning methods to dissect genetically imputed transcriptomic profiles in alzheimer's disease," *Frontiers in genetics*, vol. 10, p. 726, 2019.

[57] L. Xu *et al.*, "An efficient classifier for alzheimer's disease genes identification," *Molecules*, vol. 23, n°. 12, p. 3140, 2018.

[58] A. Y. Lan et M. R. Corces, "Deep learning approaches for noncoding variant prioritization in neurodegenerative diseases," *Frontiers in Aging Neuroscience*, vol. 14, p. 1027224, 2022.

[59] S. Lundberg, "A unified approach to interpreting model predictions," *arXiv preprint arXiv :1705.07874*, 2017.

[60] X. Huang et J. Marques-Silva, "On the failings of shapley values for explainability," *International Journal of Approximate Reasoning*, p. 109112, 2024.

[61] O. Letoffe, X. Huang et J. Marques-Silva, "On correcting shap scores," *arXiv preprint arXiv :2405.00076*, 2024.

[62] A. Salih *et al.*, "Commentary on explainable artificial intelligence methods : Shap and lime," *arXiv preprint arXiv :2305.02012*, 2023.

[63] I. E. Kumar *et al.*, "Problems with shapley-value-based explanations as feature importance measures," dans *International conference on machine learning*. PMLR, 2020, p. 5491–5500.

[64] X.-X. Zhou *et al.*, "pdeep : predicting ms/ms spectra of peptides with deep learning," *Analytical chemistry*, vol. 89, n°. 23, p. 12 690–12 697, 2017.

[65] V. Demichev *et al.*, "Dia-nn : neural networks and interference correction enable deep proteome coverage in high throughput," *Nature methods*, vol. 17, n°. 1, p. 41–44, 2020.

[66] K. Liu *et al.*, "Full-spectrum prediction of peptides tandem mass spectra using deep neural network," *Analytical chemistry*, vol. 92, n°. 6, p. 4275–4283, 2020.

[67] J. Cox *et al.*, "Accurate proteome-wide label-free quantification by delayed normalization and maximal peptide ratio extraction, termed maxlfq," *Molecular & cellular proteomics*, vol. 13, n°. 9, p. 2513–2526, 2014.

[68] A. K. Smilde *et al.*, "Fusion of mass spectrometry-based metabolomics data," *Analytical chemistry*, vol. 77, n°. 20, p. 6729–6736, 2005.

[69] S. Bijlsma *et al.*, "Large-scale human metabolomics studies : a strategy for data (pre-) processing and validation," *Analytical chemistry*, vol. 78, n°. 2, p. 567–574, 2006.

[70] R. Wei *et al.*, "Missing value imputation approach for mass spectrometry-based metabolomics data," *Scientific reports*, vol. 8, n°. 1, p. 663, 2018.

[71] Y. Yuan *et al.*, "Discrimination of missing data types in metabolomics data based on particle swarm optimization algorithm and xgboost model," *Scientific Reports*, vol. 14, n°. 1, p. 152, 2024.

[72] A. Dubey et A. Rasool, "Efficient technique of microarray missing data imputation using clustering and weighted nearest neighbour. sci. rep. 11,(2021)."

[73] R. Wei *et al.*, "Gsimp : A gibbs sampler based left-censored missing value imputation approach for metabolomics studies," *PLoS computational biology*, vol. 14, n°. 1, p. e1005973, 2018.

[74] N. Kumar, M. A. Hoque et M. Sugimoto, "Kernel weighted least square approach for imputing missing values of metabolomics data," *Scientific reports*, vol. 11, n°. 1, p. 11108, 2021.

[75] U. W. Liebal *et al.*, "Machine learning applications for mass spectrometry-based metabolomics," *Metabolites*, vol. 10, n°. 6, p. 243, 2020.

[76] S. Virreira Winter *et al.*, "Urinary proteome profiling for stratifying patients with familial parkinson's disease," *EMBO molecular medicine*, vol. 13, n°. 3, p. e13257, 2021.

[77] O. Karayel *et al.*, "Proteome profiling of cerebrospinal fluid reveals biomarker candidates for parkinson's disease," *Cell Reports Medicine*, vol. 3, n°. 6, 2022.

[78] H.-m. Park *et al.*, "Discovering biomarker proteins and peptides for parkinson's disease prognosis prediction with machine learning and interpretability methods," *bioRxiv*, p. 2023–05, 2023.

[79] X. Wang *et al.*, "Urine biomarkers discovery by metabolomics and machine learning for parkinson's disease diagnoses," *Chinese Chemical Letters*, vol. 34, n°. 10, p. 108230, 2023.

[80] S. Agarwal, P. Ghanty et N. R. Pal, "Identification of a small set of plasma signalling proteins using neural network for prediction of alzheimer's disease," *Bioinformatics*, vol. 31, n°. 15, p. 2505–2513, 2015.

[81] J. D. Zhang *et al.*, "Interpretable machine learning on metabolomics data reveals biomarkers for parkinson's disease," *ACS Central Science*, vol. 9, n°. 5, p. 1035–1045, 2023.

[82] W. Wang *et al.*, "Early detection of parkinson's disease using deep learning and machine learning," *IEEE Access*, vol. 8, p. 147 635–147 646, 2020.

[83] K. Tsukita *et al.*, "High-throughput csf proteomics and machine learning to identify proteomic signatures for parkinson disease development and progression," *Neurology*, vol. 101, n°. 14, p. e1434–e1447, 2023.

[84] K. Wang *et al.*, "Deep learning analysis of uplc-ms/ms-based metabolomics data to predict alzheimer's disease," *Journal of the neurological sciences*, vol. 453, p. 120812, 2023.

[85] L. Gaetani *et al.*, "Neuroinflammation and alzheimer's disease : a machine learning approach to csf proteomics," *Cells*, vol. 10, n°. 8, p. 1930, 2021.

[86] P. A. LeWitt *et al.*, "Diagnostic metabolomic profiling of parkinson's disease biospecimens," *Neurobiology of disease*, vol. 177, p. 105962, 2023.

[87] D. Stamate *et al.*, "A metabolite-based machine learning approach to diagnose alzheimer-type dementia in blood : Results from the european medical information framework for alzheimer disease biomarker discovery cohort," *Alzheimer's & Dementia : Translational Research & Clinical Interventions*, vol. 5, n°. 1, p. 933–938, 2019.

[88] R. Modarres, "Nonparametric classification of high dimensional observations," *Statistical Papers*, vol. 64, n°. 6, p. 1833–1859, 2023.

[89] C. Chadebec *et al.*, "Data augmentation in high dimensional low sample size setting using a geometry-based variational autoencoder," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, n°. 3, p. 2879–2896, 2022.

[90] N. Leelarathna *et al.*, "Enhancing representation learning on high-dimensional, small-size tabular data : A divide and conquer method with ensembled vaes," *arXiv preprint arXiv :2306.15661*, 2023.

[91] N. Thach *et al.*, "A novel gan approach to augment limited tabular data for short-term substance use prediction," *arXiv preprint arXiv :2407.13047*, 2024.

[92] M. Yamada *et al.*, "High-dimensional feature selection by feature-wise kernelized lasso," *Neural computation*, vol. 26, n°. 1, p. 185–207, 2014.

[93] A. K. Mandal *et al.*, "Feature subset selection for high-dimensional, low sampling size data classification using ensemble feature selection with a wrapper-based search," *IEEE Access*, 2024.

[94] X. Jiang *et al.*, "Protogate : Prototype-based neural networks with global-to-local feature selection for tabular biomedical data," dans *Forty-first International Conference on Machine Learning*, 2024.

[95] A. Margeloiu *et al.*, "Weight predictor network with feature selection for small sample tabular biomedical data," dans *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, n°. 8, 2023, p. 9081–9089.

[96] D. Singh *et al.*, "Fsnet : Feature selection network on high-dimensional biological data," dans *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, p. 1–9.

[97] B. Liu *et al.*, "Deep neural networks for high dimension, low sample size data." dans *IJCAI*, vol. 2017, 2017, p. 2287–2293.

[98] L. Shen, M. J. Er et Q. Yin, "Classification for high-dimension low-sample size data," *Pattern Recognition*, vol. 130, p. 108828, 2022.

[99] N. Ziaei *et al.*, "A bayesian gaussian process-based latent discriminative generative decoder (ldgd) model for high-dimensional data," *IEEE Access*, 2024.

[100] L. P. Cavalheiro *et al.*, "Random forest kernel for high-dimension low sample size classification," *Statistics and Computing*, vol. 34, n°. 1, p. 9, 2024.

[101] S. Sarkar et A. K. Ghosh, "On perfect clustering of high dimension, low sample size data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, n°. 9, p. 2257–2272, 2019.

[102] R. S. Turner *et al.*, "A randomized, double-blind, placebo-controlled trial of resveratrol for alzheimer disease," *Neurology*, vol. 85, n°. 16, p. 1383–1391, 2015.

[103] M. Murata *et al.*, "Zonisamide improves wearing-off in parkinson's disease : A randomized, double-blind study," *Movement Disorders*, vol. 30, n°. 10, p. 1343–1350, 2015.

[104] O. Rascol *et al.*, "A randomized, double-blind, controlled phase ii study of foliglurax in parkinson's disease," *Movement Disorders*, vol. 37, n°. 5, p. 1088–1093, 2022.

[105] C. Chen *et al.*, "Detecting placebo and drug effects on parkinson's disease symptoms by longitudinal item-score models," *CPT : Pharmacometrics & Systems Pharmacology*, vol. 10, n°. 4, p. 309–317, 2021.

[106] V. A. Bhattaram *et al.*, "Endpoints and analyses to discern disease-modifying drug effects in early parkinson's disease," *The AAPS journal*, vol. 11, p. 456–464, 2009.

[107] P. Navan *et al.*, "Double-blind, single-dose, cross-over study of the effects of pramipexole, pergolide, and placebo on rest tremor and updrs part iii in parkinson's disease," *Movement disorders : official journal of the Movement Disorder Society*, vol. 18, n°. 2, p. 176–180, 2003.

[108] D. Devos *et al.*, "Comparison of desipramine and citalopram treatments for depression in parkinson's disease : a double-blind, randomized, placebo-controlled study," *Movement Disorders*, vol. 23, n°. 6, p. 850–857, 2008.

[109] S. L. Marrinan *et al.*, "A randomized, double-blind, placebo-controlled trial of camicinal in parkinson's disease," *Movement Disorders*, vol. 33, n°. 2, p. 329–332, 2018.

[110] S. Murrar et M. Brauer, "Mixed model analysis of variance," *The SAGE encyclopedia of educational research, measurement, and evaluation*, vol. 1, p. 1075–1078, 2018.

[111] C. W. Olanow *et al.*, "A randomized, double-blind, placebo-controlled, delayed start study to assess rasagiline as a disease modifying therapy in parkinson's disease (the adagio study) : rationale, design, and baseline characteristics," *Movement disorders : official journal of the Movement Disorder Society*, vol. 23, n°. 15, p. 2194–2201, 2008.

[112] C. W. Olanow *et al.*, "A double-blind, delayed-start trial of rasagiline in parkinson's disease," *New England Journal of Medicine*, vol. 361, n°. 13, p. 1268–1278, 2009.

[113] C. V. Verschuur *et al.*, "Randomized delayed-start trial of levodopa in parkinson's disease," *New England Journal of Medicine*, vol. 380, n°. 4, p. 315–324, 2019.

[114] D. Wang *et al.*, "Statistical considerations in a delayed-start design to demonstrate disease modification effect in neurodegenerative disorders," *Pharmaceutical Statistics*, vol. 18, n°. 4, p. 407–419, 2019.

[115] C. L. Chen *et al.*, "Alzheimer's disease therapy with neuroaid (athene) : A randomized double-blind delayed-start trial," *Journal of the American Medical Directors Association*, vol. 23, n°. 3, p. 379–386, 2022.

[116] D. P. Kingma, "Adam : A method for stochastic optimization," *arXiv preprint arXiv :1412.6980*, 2014.

[117] N. Shazeer *et al.*, "Outrageously large neural networks : The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv :1701.06538*, 2017.

ANNEXE A HYPERPARAMÈTRES POUR LES MÉTHODES DE CLASSIFICATION

- **Forêt aléatoire**
  - critère : ['gini', 'entropy']
  - max_variables : ['sqrt', 'log2']
  - n_arbres : [50, 100, 200]
  - ccp_alpha (régularisation) : [10, 1, 0.01, 0.001]
- **Classificateur PPI**
  - dimension_cachée : [20, 30]
  - époque : [20, 30]
  - taux_abandon : [0.2, 0.3]
  - dimension_couche_experts : [n_variables/14,n_variables/12, n_variables/10]
  - k : [3, 5, 7]
- **Régression logistique**
  - C : [0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03]
  - pénalité : ['l1', 'l2']
- **MVS**
  - C : [0.0001, 0.001, 0.01, 0.1, 1, 10]
  - noyau : ['linear', 'rbf']
- **XGBoost**
  - max_profondeur : [None, 3, 5]
  - n_arbres : [100, 200]
  - taux_apprentissage : [0.1, 0.01]
  - alpha (régularisation) : [1, 0.1]
- **RNA**
  - dimension_couches_cachées : [(100,), (50,), (25, 25)]
  - activation : ['relu']
  - optimiseur : ['adam']
  - alpha : [0.1, 0.01]
  - taille_lot : [16, 32]
  - taux_apprentissage_type : ['adaptive']
  - taux_aprentissage : [0.01, 0.1]

ANNEXE B FIGURES DES RÉSULTATS DES SCORES F1 ET F2

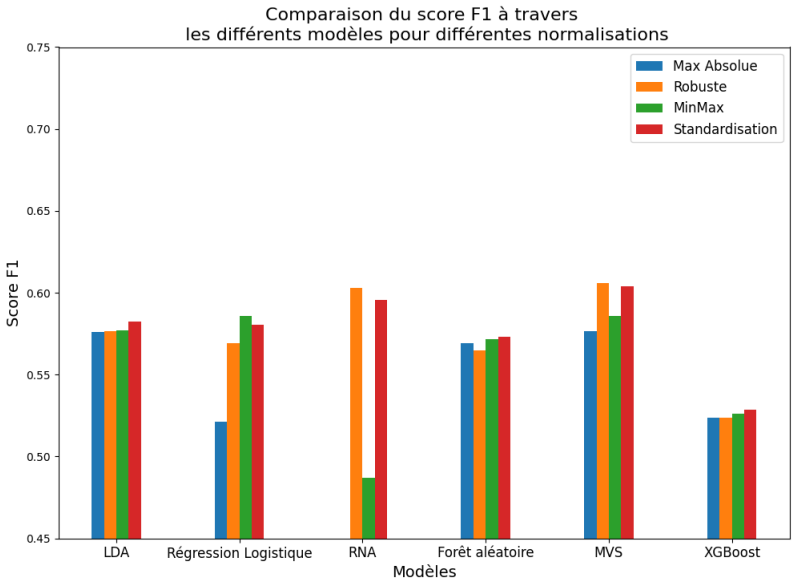

FIGURE B.1 Score F1 des différents modèles avec différentes normalisations.

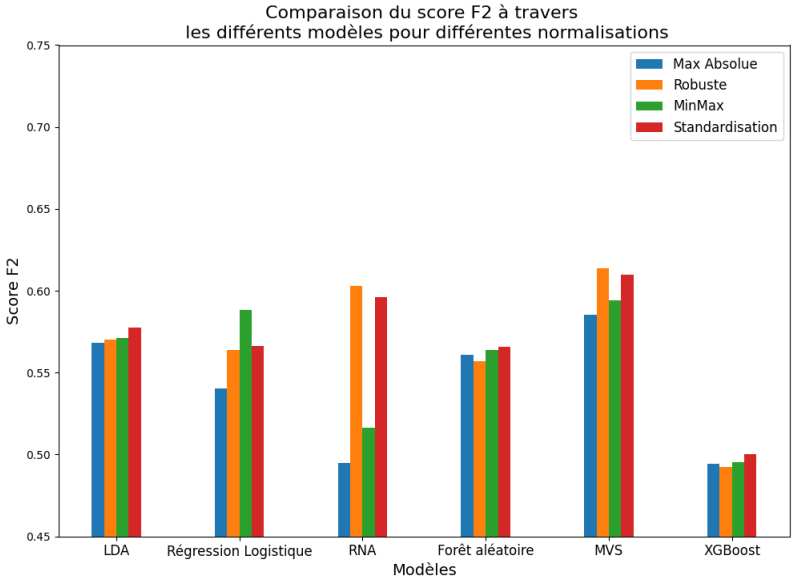

FIGURE B.2 Score F2 des différents modèles avec différentes normalisations.

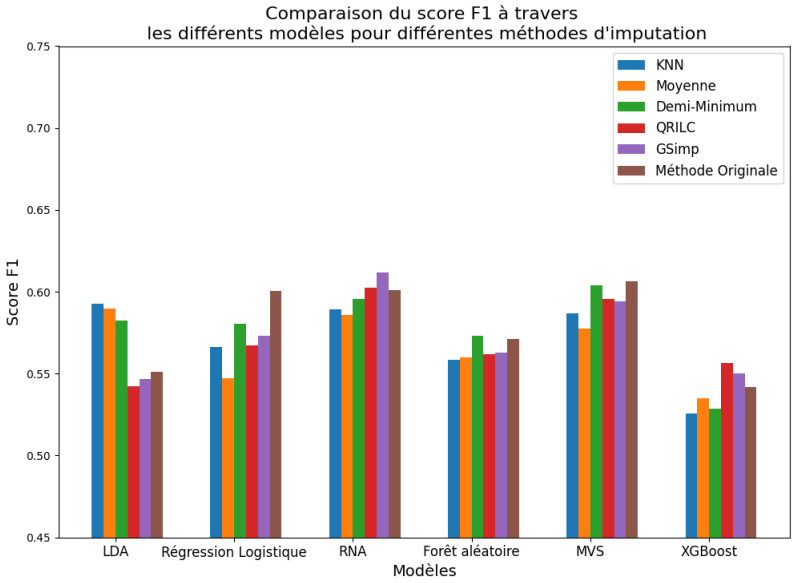

FIGURE B.3 Score F1 des différents modèles avec différentes méthodes d'imputation.

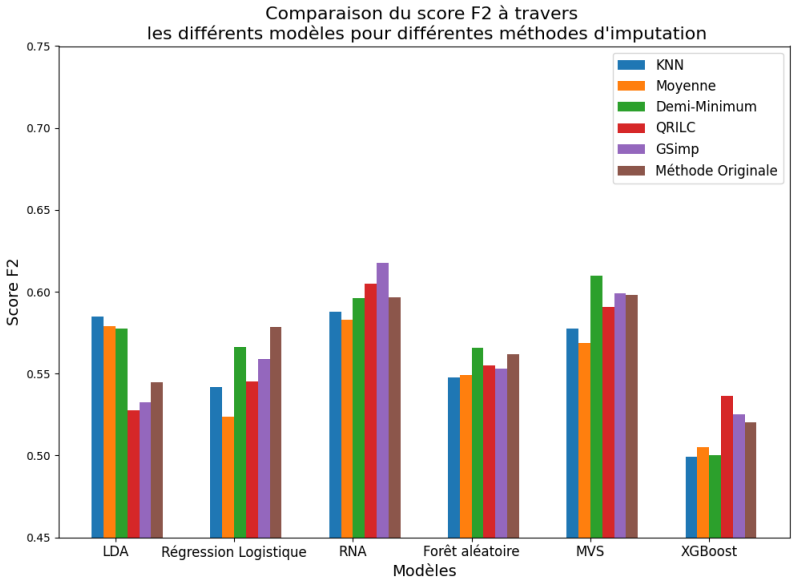

FIGURE B.4 Score F2 des différents modèles avec différentes méthodes d'imputation.

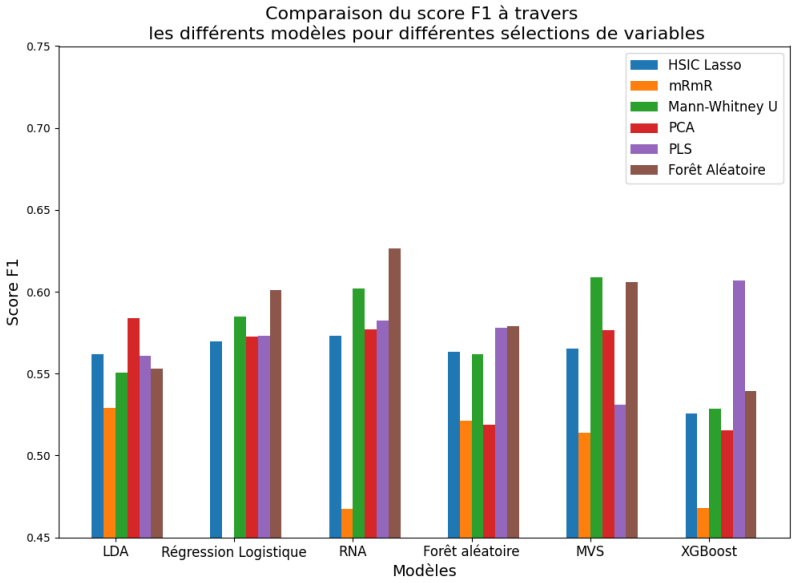

FIGURE B.5 Score F1 des différents modèles avec différentes méthodes de sélection de variables.

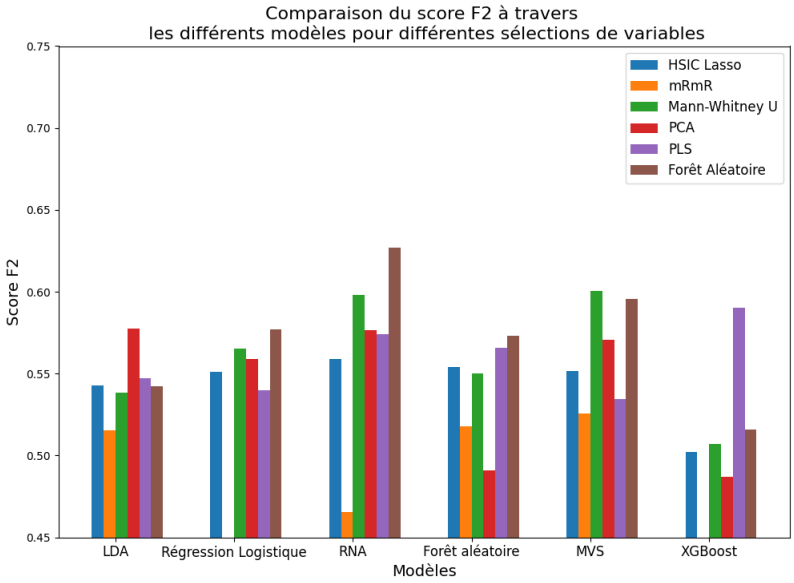

FIGURE B.6 Score F2 des différents modèles avec différentes méthodes de sélection de variables.

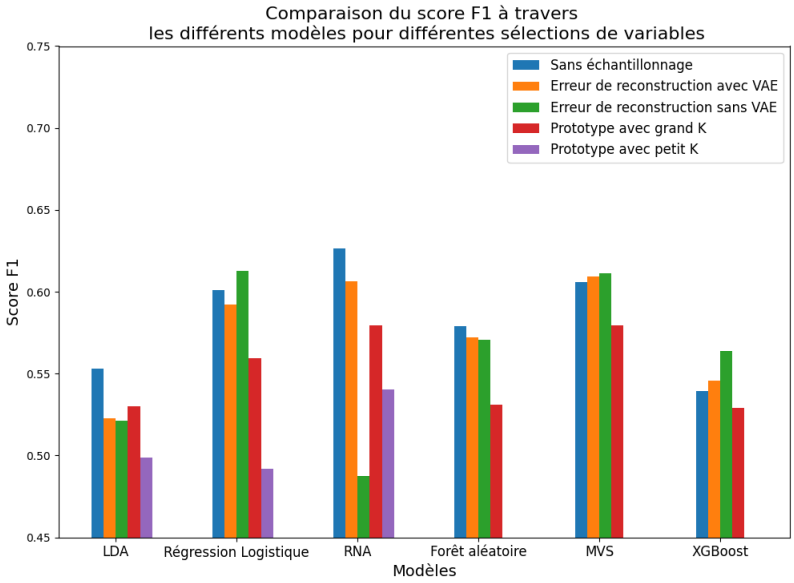

FIGURE B.7 Score F1 des différents modèles avec différentes méthodes d'échantillonnage.

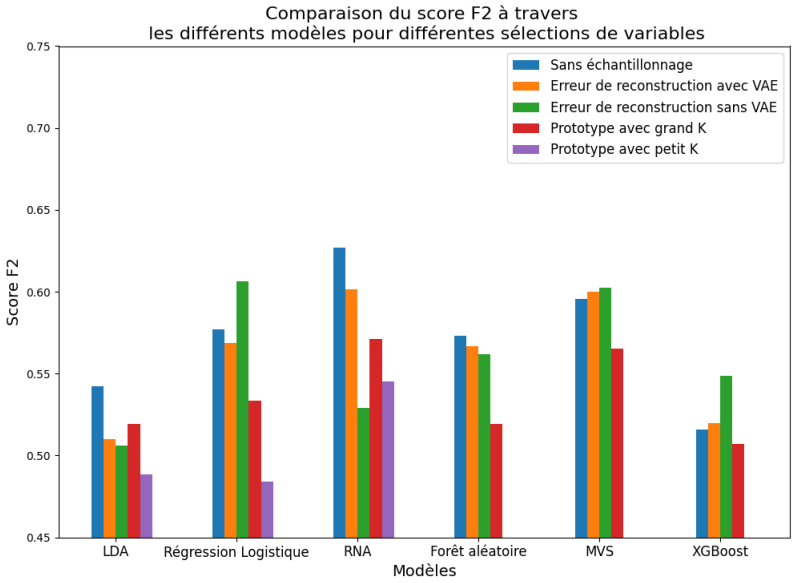

FIGURE B.8 Score F2 des différents modèles avec différentes méthodes d'échantillonnage.