المملكة المغربية
ROYAUME DU MAROC

INSTITUT AGRONOMIQUE
ET VETERINAIRE HASSAN II

معهد الحسن الثاني للزراعة
والبيطرة

Projet de Fin d'Etudes présenté pour l'obtention du diplôme d'Ingénieur d'Etat en Agronomie

Option : Ingénierie Data Science en Agriculture

Évaluation des performances de modèles d'apprentissage profond pour prédire la composition chimique de fourrage à partir de données de spectroscopie en proche infrarouge.

Présenté et soutenu publiquement par

DSSAM Abdelali

Devant le Jury composé de :

Pr. HAMOUDA Allal	**Président**	**IAV HASSAN II**
Pr. BENSIALI Saloua	**Rapporteuse**	**IAV HASSAN II**
Dr. LESNOFF Matthieu	**Co-Rapporteur**	**CIRAD**
Pr. ELAAYADI Soufiane	**Examinateur**	**IAV HASSAN II**

Juillet 2025

DEDICACE

Je dédie ce travail, fruit de longues années d'efforts et de persévérance,
À mes très chers parents,
À mon père, **Abdelmajid**, pour son soutien infaillible et ses précieux conseils.
À ma mère, **Fatiha**, pour sa tendresse, son amour inconditionnel et ses prières qui m'ont accompagné tout au long de mon parcours.
À mes frères, **Mohammed, Oussama et Abdelilah**,
Et à ma sœur, **Ibtisam**,
Pour leur fraternité, leur encouragement et les moments de joie partagés.

Une pensée pieuse et affectueuse à la mémoire de mes chères grands-pères, **Abderrahmane et Hassan**. Son souvenir reste une source d'inspiration.

À toute ma famille, pour les liens qui nous unissent et le soutien que vous m'avez toujours apporté.
À tous mes amis, ceux de l'IAV et d'ailleurs, pour leur amitié sincère et les souvenirs inoubliables qui ont enrichi mes années d'études.
Puisse ce travail être le reflet de la gratitude que je vous porte.

DSSAM Abdelali

REMERCIEMENT

Tout d’abord, louange à Allah, qui m’a guidé sur le droit chemin tout au long de ce projet, m’a inspiré les bonnes décisions et les justes réflexes. Sans Sa miséricorde et Son soutien, ce travail n’aurait pu aboutir. *Al-ḥamdu li-l-lāh*.

Ce n'est pas seulement par tradition que cette page trouve sa place ici, mais bien parce que les personnes à qui s'adressent ces remerciements les méritent sincèrement.

Je tiens à exprimer ma plus profonde reconnaissance à mes encadrants, dont la contribution a été inestimable. En premier lieu, je remercie chaleureusement **Madame BENSIALI Saloua**, Professeure à l’Institut Agronomique et Vétérinaire Hassan II au sein du Département de Statistique et Informatique Appliquées. Sa rigueur scientifique, sa grande disponibilité et ses précieux conseils ont été un guide essentiel tout au long de ce projet. Mes vifs remerciements vont également à **Monsieur LESNOFF Matthieu**, mon encadrant au sein du CIRAD. Son expertise du domaine, Ses conseils avisés et ses observations rigoureuses ont contribué de manière significative à l’avancement de ce mémoire. Au-delà de ce travail, ils représentent pour moi un socle de réflexion et d’orientation sur lequel je m’appuierai dans mes démarches scientifiques et professionnelles à venir.

J'adresse également ma reconnaissance aux membres du jury pour l'honneur qu'ils me font en acceptant d'évaluer ce mémoire. Je remercie tout particulièrement **Monsieur HAMOUDA Allal**, Professeur au Département de Statistique et Informatique Appliquées, d'avoir accepté de présider ce jury. Je remercie également **Monsieur ELAAYADI Soufiane**, Professeur au Département de Production et Biotechnologies Animales, d'avoir pris le soin d'examiner attentivement mon travail et pour le temps précieux qu'il y a consacré.

Mes remerciements s'étendent à l'ensemble du corps pédagogique et administratif de l'IAV Hassan II. Je remercie tout particulièrement, un par un, les professeurs et enseignants de l'option **Data Science en Agriculture (IDSA)**, pour la qualité de l’enseignement dispensé, la rigueur académique et leur engagement constant durant l’ensemble de notre formation.

Enfin, je présente mes excuses à toute personne que j’aurais omis de citer, et à qui j’adresse également mes remerciements les plus sincères.

Merci à toutes et à tous.

RESUME

L'évaluation de la qualité des fourrages par spectroscopie en proche infrarouge (SPIR) est un enjeu essentiel, notamment pour sa nature non-destructive et la rapidité d'analyse de la composition des fourrages. Ce projet aborde la problématique de la modélisation des données spectrales, caractérisées par une haute dimensionnalité, une forte hétérogénéité et des relations non-linéaires complexes. Étant donné la taille relativement limitée de ces jeux de données, la question centrale est de déterminer si les nouvelles architectures d'apprentissage profond peuvent être efficaces et comment elles se positionnent par rapport à des méthodes chimiométriques performantes de pointe.

En s'appuyant sur un jeu de données privé du CIRAD-Selmet, ce travail propose une évaluation comparative entre le kNN-LWPLSR, un modèle local non-linéaire de référence, et trois stratégies de Deep Learning : Une approche convolutive supervisée directe (1D-CNN), Une architecture multi-échelles plus complexe inspirée d'Inception (IPA), et Une extraction de caractéristiques non-supervisée via un autoencodeur convolutif (1D-CAE). Six modèles distincts ont été développés pour prédire six variables chimiques clés (*cp*, *ndf*, *adf*, *adl*, *cf*, *dmdcell*).

Sur ce jeu de données, Les résultats ont montré que le modèle kNN-LWPLSR s'est avéré le plus performant pour quatre des six variables, soulignant l'efficacité de sa stratégie de modélisation locale pour gérer l'hétérogénéité des données. À l'inverse, un modèle de Deep Learning, le CNN-R_v1E, a montré une performance supérieure pour 2 variables, confirmant le potentiel de l'extraction de caractéristiques par convolution. Tous les modèles les plus performants ont atteint un niveau de prédiction jugé "satisfaisant", avec un rapport de performance à l'écart-type (RPD) supérieur à 3.

En conclusion, ce travail valide que les méthodes chimiométriques locales spécialisées comme le kNN-LWPLSR conservent leur robustesse et ne sont pas encore systématiquement surpassées par les approches d'apprentissage profond pour ce type de problématique. Bien que l'apprentissage profond soit prometteur, ses performances actuelles dans cette étude suggèrent que, pour ce jeu de données, il ne justifie pas systématiquement son coût computationnel et sa complexité. Ainsi, le kNN-LWPLSR représente un compromis pratique supérieur entre performance et efficience pour ce type de données.

Mots-clés : Spectrométrie en Proche Infrarouge (SPIR), Chimiométrie, Apprentissage Profond, Apprentissage Automatique, Qualité des Fourrages, Régression, kNN-LWPLSR, Réseaux de Neurones Convolutifs (CNN), Autoencodeur (AE), CIRAD.

ABSTRACT

The evaluation of forage quality by near-infrared spectroscopy (NIRS) is a key challenge, particularly due to its non-destructive nature and the speed of analyzing forage composition. This project addresses the problem of modeling spectral data, which is characterized by high dimensionality, significant heterogeneity, and complex non-linear relationships. Given the relatively limited size of these datasets, the central question is to determine whether new deep learning architectures can be effective and how they position themselves against state-of-the-art, high-performing chemometric methods.

Using a private dataset from CIRAD-Selmet, this work presents a comparative evaluation between kNN-LWPLSR, a benchmark local non-linear model, and three Deep Learning strategies. The first is a direct supervised convolutional approach (1D-CNN), the second is a more complex multi-scale architecture inspired by Inception (IPA), and the third relies on unsupervised feature extraction via a convolutional autoencoder (1D-CAE). Six distinct models were developed to predict six key chemical variables (cp, ndf, adf, adl, cf, dmdcell).

The results show that the kNN-LWPLSR model proved to be the top performer for four of the six variables (adf, cf, cp, and ndf), highlighting the effectiveness of its local modeling strategy in handling data heterogeneity. Conversely, a Deep Learning model, CNN-R_v1E, showed superior performance for the adl and dmdcell variables, confirming the potential of feature extraction by convolution. All top-performing models achieved a prediction level deemed "excellent," with a Ratio of Performance to Deviation (RPD) greater than 3.

In conclusion, this work validates that specialized local chemometric methods like kNN-LWPLSR maintain their robustness and are not yet systematically surpassed by deep learning approaches for this type of problem. Although deep learning is promising, its current performance in this study suggests that, for this dataset, it does not systematically justify its increased cost and complexity without methodological refinement and deeper exploration. For now, kNN-LWPLSR represents a superior practical compromise between performance and efficiency for this type of data.

Keywords: Near-Infrared Spectroscopy (NIRS), Chemometrics, Deep Learning, Machine Learning, Forage Quality, Regression, kNN-LWPLSR, Convolutional Neural Networks (CNN), Autoencoder (AE), CIRAD.

Table des matières

Table des figures

Liste des tableaux

Abréviations

ADF Acid Detergent Fiber (Méthode Van Soest)

ADL Acid Detergent Lignin (Méthode Van Soest)

AE Autoencodeur

ALS Asymmetric Least Squares / Moindres carrés asymétriques

ANN Artificial Neural Networks / Réseaux de Neurones Artificiels

CAE Convolutional Autoencoder / Autoencodeur convolutif

CF Cellulose brute de Weende

CIRAD L'organisme français de recherche agronomique pour le développement

CNN Convolutional Neural Networks / Réseaux de neurones convolutifs

CP Matière azotée totale (Nx6.25)

CV Cross-validation / Validation croisée

DL Deep Learning / Apprentissage Profond

DMDCELL Digestibilité enzymatique in-vitro de la matière sèche

ELU Exponential Linear Unit

EMSC Extended Multiplicative Signal Correction / Correction Multiplicative Étendue du Signal

GNNs Graph Neural Networks / Réseaux de neurones graphiques

HPO Hyperparameter Optimization / Optimisation des hyperparamètres

IPA Inception for Petroleum Analysis

IR Infrarouge

KNN-LWPLSR K-Nearest Neighbors Locally Weighted Partial Least Squares Regression

Lasso Least Absolute Shrinkage and Selection Operator

LIME Local Interpretable Model-agnostic Explanations

LVs Latent Variables / Variables latentes

MAE Mean Absolute Error / Erreur absolue moyenne

MIR Infrarouge moyen

MLP Perceptron multicouche

MLR Multiple Linear Regression / Régression Linéaire Multiple

MSC Multiplicative Signal Correction / Correction Multiplicative du Signal

MSE Mean Squared Error / Erreur quadratique moyenne

NAS Neural Architecture Search / Recherche d'architecture neuronale

NDF Neutral Detergent Fiber (Méthode Van Soest)

NIRS Near Infrared Spectroscopy / Spectroscopie Proche Infrarouge

OLS Ordinary Least Squares / Moindres carrés ordinaires

OPLEC Optical Path-Length Estimation and Correction / Estimation et Correction de la Longueur du Chemin Optique

OSC Orthogonal Signal Correction / Correction du Signal Orthogonal

PCA Principal Component Analysis / Analyse en Composantes Principales

PCR Principal Component Regression / Régression sur Composantes Principales

PIR Proche Infrarouge

PLSR Partial Least Squares Regression / Régression par les Moindres Carrés Partiels

RBF Radial Basis Function / Fonction de base radiale

RE Erreur relative

ReLU Rectified Linear Unit

RMSE Root Mean Square Error / Erreur quadratique moyenne

RMSEP Root Mean Square Error of Prediction / Erreur quadratique moyenne de prédiction

RPD Ratio of Performance to Deviation / Rapport de performance à l'écart-type

SBN Scale-Based Normalization / Normalisation Basée sur l'Échelle

SHAP SHapley Additive exPlanations

SNV Standard Normal Variate / Variable Normale Standard

SPIR Spectrométrie proche infrarouge

SVM Support Vector Machines / Machines à vecteurs de support

SVMR Support Vector Machine Regression / Régression par Machines à Vecteurs de Support

XAI Explainable AI / Intelligence Artificielle Explicable

Chapitre 1

Introduction

1.1 Contexte

La chimiométrie, discipline à l'intersection de la chimie et des méthodes statistiques, s'appuie sur un éventail de techniques de régression pour extraire des informations quantitatives et qualitatives des données chimiques complexes, notamment les spectres. Ces méthodes permettent de modéliser la relation entre les variables explicatives (par exemple, les spectres) et les variables de réponse (par exemple, la composition chimique ou les propriétés physiques). Le choix de l'algorithme de régression est crucial et dépend de la nature des données, de la présence de colinéarité, de la nécessité de réduire la dimensionnalité et de la nature linéaire ou non linéaire de la relation sous-jacente.

1.1.1 SPIR et régression : les grands problèmes

L'application de la régression aux données de spectrométrie proche infrarouge (SPIR) se heurte à deux difficultés majeures. Premièrement, les données spectrales sont de haute dimensionnalité, générant un grand nombre de variables (longueurs d'onde) qui sont fortement corrélées entre elles. Cela produit des matrices de données mal conditionnées où les modèles de régression linéaire classiques sont généralement inapplicables. Deuxièmement, les bases de données agronomiques sont souvent caractérisées par une forte hétérogénéité, car elles regroupent des échantillons de natures diverses (espèces et variétés de plantes, parties de la plante, zones géographiques, etc.). Cette hétérogénéité engendre des relations non linéaires complexes (clusters) entre les spectres et les variables chimiques, rendant les modèles linéaires sous-optimaux.

1.1.2 Le CIRAD et l'unité Selmet

Le **CIRAD**, l'organisme français de recherche agronomique pour le développement, a pour mission de construire des solutions durables pour les agricultures du Sud. Cette mission s'incarne au sein de l'unité de recherche **Selmet**, qui se consacre à l'étude des systèmes d'élevage en zones méditerranéennes et tropicales.

Ces systèmes pastoraux, patrimoines culturels et écologiques, font face à des menaces critiques : raréfaction des ressources fourragères due au dérèglement climatique, dégrada-

tion des terres et concurrence économique. Garantir une alimentation animale stable est donc un enjeu majeur. L'objectif principal de Selmet est ainsi d'analyser les dynamiques de ces élevages pour renforcer leur résilience. L'évaluation précise de la qualité des fourrages, au cœur de ce stage, est une composante essentielle de cette stratégie.

1.2 Problématique du stage

Face aux défis de non-linéarité et d'hétérogénéité des données spectrales, des algorithmes sophistiquées comme le **kNN-LWPLSR** (*k-Nearest Neighbors Locally Weighted Partial Least Squares Regression*) ont été développés et se sont avérés très performants et compétitifs pour l'analyse de données agronomiques. Parallèlement, le domaine de l'apprentissage profond (*Deep Learning*) connaît une croissance rapide, avec l'émergence de nouvelles architectures neuronales (CNNs, CAE) qui montrent des performances prometteuses en chimiométrie.

Dès lors, une question centrale et actuelle se pose : il s'agit de positionner ces nouveaux modèles d'apprentissage profond par rapport à des pipelines déjà existants et très performants comme le kNN-LWPLSR. La problématique de ce stage est donc de réaliser une évaluation comparative entre la performance du kNN-LWPLSR et celle de différentes architectures de Deep Learning pour prédire la composition chimique (protéines brutes, fibres, etc.) d'échantillons des plantes fourragères à partir de leurs spectres proche infrarouge, dans un contexte de jeu de données privé de taille relativement réduite, fourni par le CIRAD-Selmet et composé d'environ 1 100 échantillons.

1.3 Méthodologie envisagée

Pour répondre à cette problématique, la démarche méthodologique s'est articulée autour de plusieurs étapes clés. Dans un premier temps, une revue de la littérature a été conduite pour identifier les architectures d'apprentissage profond les plus pertinentes pour l'analyse de données spectrales, en tenant compte des contraintes d'un jeu de données de taille limitée.

Ensuite, une comparaison a été mise en œuvre entre le modèle de référence chimiométrique, le **kNN-LWPLSR**, et une sélection de trois stratégies de deep learning distinctes pour l'extraction de caractéristiques. Celles-ci incluaient : une approche convolutive supervisée directe via des **réseaux de neurones convolutifs (1D-CNN)** simples , une architecture multi-échelles plus complexe inspirée d'Inception, le modèle **IPA** , et une approche d'extraction non-supervisée utilisant un **autoencodeur convolutif (1D-CAE)** en amont d'une régression.

Un protocole expérimental commun a été défini pour assurer une comparaison équitable. Les données ont été partitionnées en un ensemble d'entraînement (75%) et un ensemble de test (25%). Les hyperparamètres de chaque modèle ont ensuite été optimisés par une validation croisée à 5 plis (5-fold cross-validation) menée exclusivement sur l'ensemble d'entraînement. Finalement, les performances prédictives des modèles finaux optimisés ont été évaluées sur l'ensemble de test, qui avait été tenu à l'écart durant toute la phase d'entraînement, à l'aide de métriques statistiques standard comme l'erreur quadratique

moyenne de prédiction (RMSEP), le rapport de performance à l'écart-type (RPD) et l'erreur relative (RE).

1.4 Annonce du plan du rapport

Ce rapport est structuré de la manière suivante. Le chapitre 2 présentera le socle théorique de la spectrométrie infrarouge ainsi que les fondements des différentes méthodes de régression employées, des approches chimiométriques classiques aux modèles d'apprentissage profond. Le chapitre 3 détaillera le matériel et les méthodes, incluant la description du jeu de données et le protocole expérimental mis en œuvre pour le dévelopement, l'entrainement et la comparaison des modèles. Le chapitre 4 expose et analyse les résultats obtenus, en comparant les performances des différents modèles. Enfin, le chapitre 5 conclura ce travail en discutant les implications des résultats et en proposant des perspectives pour de futurs travaux.

Chapitre 2

Fondements et aspects théorique

2.1 Spectrométrie IR

2.1.1 Principe général de la spectrométrie infrarouge

La spectroscopie, en tant que discipline scientifique, se définit comme l'étude de l'interaction entre la lumière et la matière (YUFENG et al., 2024). La spectroscopie infrarouge (IR) se distingue par son fondement sur l'interaction spécifique entre le rayonnement électromagnétique et les molécules, conduisant à des transitions au niveau de leurs états vibrationnels.

Le rayonnement électromagnétique présente une dualité onde-corpuscule, se manifestant à la fois comme une onde et comme un flux de particules élémentaires appelées photons ou quanta. L'énergie transportée par un photon est directement proportionnelle à sa fréquence (ν) et inversement proportionnelle à sa longueur d'onde (λ), une relation fondamentale décrite par l'équation de Planck-Einstein :

$$E = h\nu = \frac{hc}{\lambda}$$

où h représente la constante de Planck et c la célérité de la lumière dans le vide.

L'absorption d'un rayonnement électromagnétique par une molécule n'est possible que si l'énergie du photon incident correspond précisément à la différence d'énergie (ΔE) entre deux de ces niveaux énergétiques de la molécule (DONATO et al., 2015) :

$$\Delta E = h\nu$$

Ce principe de résonance est universel à travers diverses techniques spectroscopiques. Par exemple, les transitions électroniques sont observées dans le domaine UV-Visible, tandis que les transitions vibrationnelles caractérisent le domaine infrarouge, et les transitions rotationnelles sont associées aux micro-ondes. Cette connexion fondamentale de la spectroscopie IR à un cadre plus large d'interaction lumière-matière confère une base théorique solide aux données utilisées dans les études ultérieures. La gamme d'énergie spécifique du rayonnement infrarouge détermine directement son interaction privilégiée avec les vibrations moléculaires, plutôt qu'avec les transitions électroniques. Cette relation de cause à effet explique pourquoi la spectroscopie IR est une source d'information précieuse sur la

structure moléculaire et les groupes fonctionnels, un aspect essentiel pour les tâches de régression visant à prédire des propriétés chimiques.

Le domaine infrarouge est conventionnellement subdivisé en plusieurs régions. Le proche infrarouge (PIR), également connu sous l'acronyme *NIRS* (Near Infrared Spectroscopy), utilise des longueurs d'onde allant de 700 à 2500 nm. Dans le cadre de ce travail, les données disponibles couvrent une portion de cette plage, allant d'environ 1100 à 2498 nm. L'infrarouge moyen (MIR) se situe généralement dans une gamme de nombres d'onde comprise entre 4000 et $400\,\mathrm{cm}^{-1}$.

TABLE 2.1 – longueurs d'onde (μm) et nombres d'onde (cm^{-1}) pour les régions du proche, moyen et lointain infrarouge.

	longueurs d'onde (μm)	**nombres d'onde (cm^{-1})**
proche infrarouge	0,7-2,5 μm	13300-4000 cm^{-1}
moyen infrarouge	2,5-25 μm	4000-400 cm^{-1}
lointain infrarouge	25-1000 μm	400-10 cm^{-1}

(a) Vibrations moléculaires et absorption du rayonnement infrarouge

En spectroscopie infrarouge, les liaisons chimiques au sein d'une molécule peuvent être conceptualisées comme des systèmes de ressorts reliant des atomes, assimilés à des masses. Ces liaisons sont capables d'osciller à des fréquences intrinsèques, désignées sous le terme de fréquences de résonance ou fréquences propres (BHAGWAT et al., 2024).

Lorsqu'une molécule est exposée à un rayonnement infrarouge, De manière simplifiée, elle peut absorber cette énergie et transiter d'un niveau de vibration de plus basse énergie vers un niveau de vibration supérieur. Ce phénomène se produit spécifiquement lorsque la fréquence du rayonnement infrarouge incident correspond à la fréquence naturelle de vibration de la molécule. C'est pourquoi la spectroscopie infrarouge est souvent qualifiée de « spectroscopie de vibration-rotation », car les transitions vibrationnelles sont intrinsèquement couplées à des changements de niveaux rotationnels.

(b) Représentation et interprétation des spectres infrarouges

Un spectre infrarouge est typiquement présenté sous la forme d'un graphique où l'intensité d'absorption (ou la transmittance) est tracée en fonction du nombre d'ondes (en cm^{-1}) ou de la longueur d'onde (en nm). L'axe des abscisses, représentant le nombre d'ondes, est souvent orienté de manière décroissante, allant des nombres d'ondes élevés vers les plus faibles.

Les diminutions significatives et localisées de la transmittance, ou les augmentations correspondantes de l'absorption, à des nombres d'ondes spécifiques, sont le résultat de l'absorption sélective par les liaisons chimiques et sont appelées bandes d'absorption (YUFENG et al., 2024). L'interprétation de ces spectres implique l'identification de ces bandes caractéristiques, qui sont associées à des groupes fonctionnels spécifiques, en se référant à des tables de données établies.

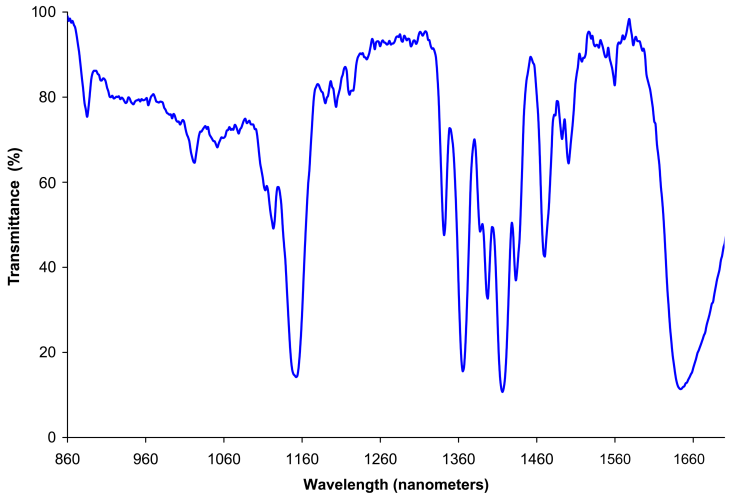


FIGURE 2.1 – Exemple de spectre Proche Infrarouge (PIR), illustrant la relation entre la transmittance et la longueur d'onde.
Source : Wikimedia Commons

Un spectre IR est souvent décrit comme une "carte d'identité" moléculaire en raison de sa richesse informationnelle, bien que l'attribution structurale de toutes les bandes puisse être complexe en raison de phénomènes tels que les bandes harmoniques ou les combinaisons de fréquences, le spectre global fournit une signature unique de la molécule (EL FALEH, 2019).

Cette signature moléculaire unique est précisément ce que les modèles d'apprentissage automatique sont capables d'exploiter. Alors que l'interprétation traditionnelle se concentre sur l'identification de groupes fonctionnels spécifiques, les modèles d'apprentissage automatique peuvent traiter l'ensemble du spectre, y compris la région complexe et chevauchante de "l'empreinte digitale" et les légers décalages induits par l'environnement moléculaire. Cette approche holistique, allant au-delà de la simple attribution de pics, est une force majeure de l'apprentissage automatique, ce qui rend les spectres IR des candidats idéaux pour les applications d'apprentissage automatique en régression.

2.1.2 Revue des prétraitements en spectrométrie

(a) Nécessité et objectifs du prétraitement des données spectrales

Les données brutes acquises par spectroscopie infrarouge sont fréquemment sujettes à des altérations dues à divers phénomènes indésirables. Ces perturbations, qui incluent le bruit instrumental, les variations de la ligne de base, les effets de diffusion (scattering) et les variations d'épaisseur de l'échantillon, peuvent masquer les informations analytiques pertinentes et compromettre la précision des modèles prédictifs (WAN et al., 2017).

C'est pour cela que le prétraitement des données spectrales constitue une étape indispensable en chimiométrie, dont l'objectif est de raffiner le signal et de réduire la complexité des données. L'objectif principal de cette phase est d'améliorer le rapport signal/bruit, de corriger les artefacts induits par l'instrumentation ou liés aux propriétés physiques de l'échantillon, et d'assurer la comparabilité des spectres entre eux.

Si les variations de la ligne de base sont importantes et complexes, le modèle d'apprentissage profond pourrait consacrer sa capacité d'apprentissage à modéliser ces variations non pertinentes plutôt que les relations chimiques sous-jacentes. Un prétraitement permet au modèle de concentrer ses ressources sur l'extraction de caractéristiques chimiquement significatives, ce qui devrait conduire à une régression plus efficace et plus précise.

Les techniques de prétraitement sont largement catégorisées en fonction du type de variation qu'elles visent à corriger (WITTEVEEN et al., 2022). Ces catégories incluent :

(b) Normalisation et mise à l'échelle

Méthodes qui ajustent l'intensité ou l'amplitude globale des spectres pour assurer la comparabilité. Cela inclut des techniques comme la Mise à l'échelle Constante (Constant Shift) pour ajuster un décalage vertical constant, la Mise à l'échelle Générale (Scaling) pour redimensionner les amplitudes, la Variable Normale Standard (SNV) (Standard Normal Variate) qui centre et met à l'échelle chaque spectre individuellement pour corriger les variations de ligne de base et de diffusion, et la Normalisation Basée sur l'Échelle (SBN) (Scale-Based Normalization) qui utilise la décomposition par ondelettes pour une normalisation plus sophistiquée.

(c) Correction de la ligne de base

Techniques conçues pour supprimer les signaux de fond additifs ou à basse fréquence qui décalent verticalement l'ensemble du spectre, indépendamment des pics d'intérêt. La méthode des moindres carrés asymétriques (ALS) (Asymmetric Least Squares) est une technique populaire pour estimer et soustraire la ligne de base, tandis que l'ajustement polynomial (Polynomial Fitting) peut également être utilisé pour modéliser et retirer les dérives de ligne de base.

(d) Correction de la diffusion

Approches qui traitent les variations causées par la diffusion de la lumière due aux propriétés physiques de l'échantillon (taille des particules, densité, etc.). La Correction Multiplicative du Signal (MSC) (Multiplicative Signal Correction) modélise ces effets en considérant que le spectre de l'échantillon est une version décalée et mise à l'échelle d'un spectre idéal. La Correction Multiplicative Étendue du Signal (EMSC) (Extended Multiplicative Signal Correction) est une extension de la MSC qui peut également corriger d'autres interférences. L'Estimation et Correction de la Longueur du Chemin Optique (OPLEC) (Optical Path-Length Estimation and Correction) est une autre méthode ciblant spécifiquement les variations liées à la longueur du chemin optique.

(e) Suppression du bruit/Lissage

Méthodes qui réduisent les fluctuations aléatoires à haute fréquence (bruit) pour améliorer la clarté du signal sans déformer les caractéristiques spectrales importantes. La méthode de Savitzky-Golay (Savitzky-Golay) est couramment utilisée pour le lissage en ajustant un polynôme de faible degré à une fenêtre glissante de points de données.

(f) Traitement dérivé

Techniques qui calculent le taux de changement du signal spectral pour améliorer la résolution des pics, séparer les pics superposés, et supprimer les effets de ligne de base et les dérives linéaires. La dérivation Savitzky-Golay (Savitzky-Golay Derivatisation) est souvent employée pour calculer les première ou seconde dérivées des spectres tout en les lissant.

(g) redressement (*Detrending*)

Procédures pour éliminer les dérives à basse fréquence au fil du temps ou de la longueur d'onde, qui peuvent apparaître sous forme de lignes de base courbées ou d'autres variations lentes non liées à la composition chimique.

(h) Centrage de la moyenne

Une technique simple mais fondamentale qui soustrait la moyenne de chaque variable (longueur d'onde) à toutes les observations. Cela recentre les données autour de zéro, éliminant un décalage global et rendant les variables plus comparables, ce qui est crucial pour les méthodes d'analyse multivariée comme l'Analyse en Composantes Principales (PCA) ou les Moindres Carrés Partiels (PLS).

(i) Correction du signal orthogonal (OSC)

Une méthode qui vise à supprimer les informations non corrélées à la variable de réponse (propriété chimique) des données spectrales. Elle projette les données sur un espace orthogonal à la variable de réponse, ne conservant que les variations utiles pour le modèle prédictif. Bien que catégorisées de manière distincte, de nombreuses techniques de prétraitement ne sont pas mutuellement exclusives et sont souvent appliquées en séquence dans le cadre d'un pipeline. Par exemple, dans (LESNOFF, 2023), une normalisation par la méthode SNV est suivie d'un lissage selon la méthode de Savitzky-Golay, avec un polynôme d'ordre 3. Il s'agit de la même méthodologie utilisée pour les données employées dans ce travail (voir section 3.1).

Cependant, une tendance émergente dans le domaine consiste à utiliser des modèles d'apprentissage profond pour contourner l'étape de prétraitement. les réseaux de neurones convolutifs (ZHANG, XU et al., 2018) et les transformeurs (FU et al., 2022) sont capables de traiter directement les données spectrales brutes. Ces approches de bout en bout (end-to-end) visent non seulement à simplifier le flux de travail analytique, mais aussi à potentiellement améliorer la performance des modèles en apprenant à extraire les caractéristiques pertinentes de manière autonome.

2.2 Méthodes de régression en chimiométrie

L'analyse des données de spectroscopie proche infrarouge (NIRS) est souvent confrontée à des défis inhérents à leur haute dimensionalité. En effet, les spectres sont fréquemment fortement colinéaires et sujettes au bruit instrumental. Ces problèmes peuvent gravement affecter la stabilité et la prédictivité des modèles de régression traditionnels. Pour pallier ces limitations et extraire l'information pertinente tout en réduisant la complexité du modèle, les méthodes de régression basées sur les variables latentes sont largement privilégiées. Ces approches consistent à projeter les données originales dans un espace de dimension inférieure, défini par un ensemble de nouvelles variables (les variables latentes), qui capturent la majeure partie de la variance et de la covariance des données tout en minimisant l'impact du bruit.

2.2.1 Méthodes de régression sur variables latentes

(a) Régression sur composantes principales (PCR)

La Régression sur Composantes Principales (PCR) est une technique qui combine deux étapes distinctes : l'Analyse en Composantes Principales (ACP) et la Régression Linéaire Multiple (MLR). Le processus débute par l'application d'une ACP sur la matrice des variables explicatives (X), pour extraire un ensemble de composantes principales (ou scores), qui capturent la variance maximale des données X. Ces composantes sont intrinsèquement orthogonales et non corrélées. Une fois ces composantes obtenues, une MLR est ensuite effectuée sur les composantes principales retenues afin de prédire la variable de réponse Y. Cependant, la PCR présente une limitation significative. Les composantes extraites par l'ACP sont calculées dans l'unique but de maximiser la variance des variables explicatives (X), sans tenir compte de leur relation avec la variable de réponse (Y). Cela implique que les composantes qui expliquent la plus grande part de variance dans X ne sont pas nécessairement celles qui sont les plus pertinentes pour la prédiction de Y.

(b) Régression par les moindres carrés partiels (PLSR)

La Régression par les Moindres Carrés Partiels (PLSR) est une méthode de référence en chimiométrie, particulièrement adaptée pour gérer les données spectrales caractérisées par une forte colinéarité et une dimensionnalité élevée (LESNOFF, 2023). Le principe de la PLSR repose sur la réduction de la dimensionnalité de la matrice de données X. Pour ce faire, on calcule un nombre limité de vecteurs orthogonaux, appelés scores ou variables latentes (LVs), qui sont construits de manière à maximiser la covariance carrée avec la variable de réponse Y. Ces scores sont ensuite utilisés comme régresseurs dans un modèle MLR pour prédire Y.

L'avantage majeur de la PLSR par rapport à la PCR réside dans sa capacité à intégrer la variable de réponse Y directement dans le processus de construction de ses composantes. Cette approche permet à la PLSR de générer des modèles qui peuvent prédire Y avec un nombre réduit de composantes et d'être moins susceptible d'inclure des facteurs non pertinents pour la prédiction. La PLSR est particulièrement efficace lorsque la relation entre X et Y est linéaire.

2.2.2 Régression régularisée : Ridge et Lasso

Pour améliorer la robustesse et la prédictibilité des modèles de régression linéaires avec les données colinéaires et nombreuses, la régression Ridge et Lasso intègrent des termes de pénalisation. Cette approche est spécifiquement conçue pour améliorer la gestion de la colinéarité et prévenir le surajustement, des problèmes courants dans les données à haute dimensionnalité.

La Régression Ridge (HOERL, 2020), également désignée sous le nom de régularisation L2, est une technique de régularisation statistique qui modifie la fonction de coût des moindres carrés ordinaires (OLS) en y ajoutant un terme de pénalité. Ce terme est proportionnel à la somme des carrés des coefficients du modèle, ce qui correspond à la norme L2. L'incorporation de cette pénalité a pour effet de contraindre les coefficients du modèle à se contracter (phénomène de coefficient shrinkage) et à se rapprocher de zéro. Il est important de noter que, contrairement à d'autres méthodes, la Régression Ridge ne réduit jamais les coefficients à zéro absolu. Cette contraction des coefficients contribue à réduire la complexité du modèle, à gérer efficacement la multicolinéarité présente dans les données et à prévenir le surajustement. Toutefois, une limitation inhérente à la Régression Ridge est son incapacité à réaliser une sélection de variables, puisque aucun coefficient n'est complètement éliminé du modèle.

La Régression Lasso (Least Absolute Shrinkage and Selection Operator) (TIBSHIRANI, 1996) est une autre technique de régularisation qui, comme la Ridge, ajoute un terme de pénalité à la fonction de coût OLS. La distinction fondamentale réside dans la nature de ce terme de pénalité : pour Lasso, il est proportionnel à la somme des valeurs absolues des coefficients (norme L1). Cette pénalité L1 confère à Lasso une propriété unique : elle peut réduire certains coefficients à exactement zéro. Par conséquent, Lasso réalise de manière intrinsèque une sélection de variables, en éliminant les caractéristiques moins importantes du modèle. Au-delà de sa capacité à réduire le surajustement et à gérer la multicolinéarité, Lasso offre l'avantage de produire un modèle plus parcimonieux et plus facilement interprétable, en identifiant explicitement les variables les plus influentes.

2.2.3 Algorithmes non linéaires

Dans la plupart des cas, les relations entre les données spectrales et les propriétés chimiques ne se manifestent pas par une simple linéarité, les algorithmes non linéaires deviennent des outils essentiels, Voici les algorithmes les plus utilisées :

(a) Forêts aléatoires

Les Forêts Aléatoires (Random Forests) sont des algorithmes d'apprentissage par ensemble qui combinent les prédictions de multiples arbres de décision pour former un modèle plus robuste. Pour les tâches de régression, la prédiction finale est obtenue en calculant la moyenne des prédictions générées par chaque arbre individuel au sein de la forêt. Le fonctionnement des Forêts Aléatoires repose sur une technique appelée "bagging" (bootstrap aggregating). Cette méthode implique que chaque arbre de la forêt est entraîné sur un échantillon bootstrap, c'est-à-dire un sous-échantillon des données originales obtenu par tirage avec remise. De plus, à chaque nœud de l'arbre, seule une sous-sélection aléatoire

de variables est prise en compte pour la division. Cette double source de hasard réduit significativement la corrélation entre les arbres individuels, ce qui améliore la robustesse et la capacité de généralisation du modèle global. Les Forêts Aléatoires sont largement reconnues pour leur grande précision et leur robustesse (RELANDER et al., 2022). Elles sont particulièrement efficaces pour gérer les relations non linéaires et sont moins sujettes au surajustement qu'un seul arbre de décision. Leur architecture leur permet de traiter de grands ensembles de données complexes et d'intégrer des caractéristiques catégorielles et numériques sans nécessiter de prétraitement extensif.

(b) Régression par machines à vecteurs de support (SVMR)

La Régression par Machines à Vecteurs de Support (SVR) est une extension des machines à vecteurs de support (SVM) spécifiquement conçue pour les problèmes de régression. L'objectif fondamental de la SVR est de trouver une fonction qui s'écarte des valeurs cibles d'une marge ϵ (epsilon) au maximum pour toutes les données d'entraînement, tout en étant la plus "plate" possible. Pour ce faire, la SVR cherche à construire un hyperplan dans un espace de haute dimension qui maximise la largeur d'une marge autour des données d'entraînement (MA et al., 2018). La SVR est capable de modéliser des relations non linéaires grâce à l'utilisation de fonctions de noyau (kernels). Ces fonctions transforment implicitement les données originales dans un espace de caractéristiques de dimension supérieure où une régression linéaire peut alors être appliquée. Parmi les noyaux couramment utilisés figurent le noyau linéaire, polynomial et la fonction de base radiale (RBF). Parmi ses avantages, la SVR est reconnue pour sa robustesse aux valeurs aberrantes (outliers), car seules les observations situées en dehors ou sur la marge (appelées vecteurs de support) influencent directement la construction du modèle. Elle offre une excellente capacité de généralisation et une grande précision de prédiction. De plus, elle est particulièrement apte à modéliser des relations non linéaires complexes.

(c) PLS local (kNN-LWPLSR)

La PLS locale est une approche qui vise à gérer la non-linéarité et l'hétérogénéité des données en adaptant le modèle aux caractéristiques locales. Le kNN-LWPLSR est un exemple avancé de cette catégorie, qui sera détaillé dans la section 2.3.

2.3 Algorithmes retenus pour le stage

Au vu des caractéristiques de notre jeu de données d'échantillons de fourrages, qui présente une taille limitée et de fortes présomptions de non-linéarité, nous avons sélectionné un portefeuille de modèles variés pour comparer leurs performances. Nous avons retenu le kNN-LWPLSR, une méthode locale reconnue pour sa robustesse face aux non-linéarités, qui servira de modèle de référence (*baseline*) pour l'analyse chimiométrique classique.

Pour l'approche par apprentissage profond, nous avons choisi d'explorer trois stratégies d'extraction de caractéristiques distinctes :

— **Un 1D-CNN simple**, représentant une approche convolutive supervisée directe pour établir une performance de base.
— **Le modèle IPA**, pour tester une architecture multi-échelles inspirée d'Inception, capable de capturer des informations spectrales à différentes résolutions simultanément.
— **Un 1D-CAE**, afin d'évaluer une approche par extraction de caractéristiques non supervisée, où le modèle apprend une représentation pertinente du spectre avant de réaliser la régression.

cette section présente en détail les différents modèles retenus.

2.3.1 kNN-LWPLSR

L'algorithme kNN-LWPLSR (k-nearest neighbors locally weighted PLSR) est une méthode avancée qui combine la sélection des plus proches voisins avec une régression PLS pondérée localement (LESNOFF, 2023). Cette approche est particulièrement bien adaptée pour gérer la non-linéarité et l'hétérogénéité fréquemment rencontrées dans les données chimiométriques.

La Régression par les Moindres Carrés Partiels (PLSR) est une méthode efficace lorsque les relations entre les données spectrales et les variables de réponse sont linéaires. Cependant, les bases de données agronomiques, par exemple, agrègent souvent des échantillons hétérogènes (mélanges d'espèces, origines géographiques diverses) qui introduisent des non-linéarités, se manifestant par des courbures ou des regroupements dans les données. La PLSR locale a été développée comme une solution pour gérer cette non-linéarité.

Le principe de la PLSR locale est le suivant : pour chaque nouvelle observation (x_{new}) dont la valeur doit être prédite, une pré-sélection de k plus proches voisins est effectuée. Une fois ce voisinage défini, une PLSR est appliquée uniquement à ce sous-ensemble de k voisins. Le kNN-LWPLSR est une variante qui affine cette approche en appliquant une PLSR pondérée localement (LWPLSR) sur le voisinage. Dans la LWPLSR, un vecteur de poids (δ) est intégré à l'algorithme PLSR. Ces poids sont calculés à partir d'une fonction décroissante de la distance entre les observations d'entraînement et x_{new}. Cela signifie que les observations plus proches de x_{new} se voient attribuer un poids plus élevé, leur conférant ainsi une plus grande importance dans la prédiction locale.

Le pipeline kNN-LWPLSR se déroule en plusieurs étapes séquentielles :

1. **Espace de scores PLSR global** : Une PLSR globale est initialement ajustée sur l'ensemble des données d'entraînement pour établir un espace de scores global.

2. **Calcul des distances** : Pour chaque nouvelle observation x_{new}, les distances de Mahalanobis sont calculées dans cet espace de scores global entre x_{new} et les observations d'entraînement.
3. **Sélection kNN et pondération** : Ces distances servent à sélectionner les k plus proches voisins et à calculer leurs poids via une fonction de pondération (souvent de forme exponentielle négative), dont la "netteté" est contrôlée par un paramètre h.
4. **LWPLSR sur le voisinage** : Une LWPLSR est ensuite appliquée sur ce voisinage pondéré pour générer la prédiction.

TABLE 2.2 – Paramètres du KNN-LWPLSR

Paramètre	**Description**	**Rôle dans l'algorithme**
nlv_{dis}	Nombre de scores PLS globaux	Définit l'espace dans lequel les distances de Mahalanobis sont calculées pour la sélection kNN.
k	Nombre de voisins	Détermine la taille du voisinage local sur lequel la LWPLSR est appliquée.
h	Facteur de forme de la fonction de poids	Contrôle la "netteté" de la fonction de poids, influençant l'importance des voisins les plus proches.
nlv	Nombre de variables latentes (LVs)	Spécifie la dimensionnalité du modèle LWPLSR appliqué sur le voisinage local.

2.3.2 Modèles de Deep learning

L'intégration du deep learning (DL) en chimiométrie a considérablement transformé l'analyse des données chimiques complexes, marquant une évolution notable par rapport aux méthodes chimiométriques traditionnelles. Les réseaux de neurones profonds, notamment les réseaux neuronaux convolutifs (CNNs) (MISHRA, PASSOS et al., 2022) et les réseaux de neurones graphiques (GNNs) (TRAN et al., 2021), ont démontré une capacité supérieure à modéliser les relations non linéaires et à extraire des caractéristiques pertinentes des données spectrales, chromatographiques et moléculaires.

Plusieurs études comparatives récentes soulignent la performance accrue des architectures de deep learning par rapport aux approches conventionnelles comme la régression par les moindres carrés partiels (PLSR) ou la régression sur composantes principales (PCR), particulièrement dans la gestion des données de haute dimension et bruitées. Par exemple, des travaux ont mis en évidence la robustesse des CNNs pour l'évaluation de la qualité des fruits par spectroscopie proche infrarouge (WALSH et al., 2023) ou pour la prédiction des propriétés du pétrole brut à partir de données FTIR (SOUVIK et al., 2023). Bien que des défis persistent, notamment en matière de disponibilité des données, d'interprétabilité des modèles et de standardisation, le deep learning est désormais reconnu comme un outil essentiel pour améliorer la précision prédictive et la compréhension des processus chimiques.

Pour Ce travail, et avec un jeu de données de taille limitée, nous avons exploré la littérature afin d'identifier des architectures pertinentes et spécifiquement adaptées à ces contraintes, lesquelles seront détaillées dans la ci-après.

2.3.2.1 Modèle 1D-CNN (PASSOS et MISHRA, 2023)

L'article (PASSOS et MISHRA, 2023) explore des variantes de l'architecture standard des réseaux convolutifs 1D-CNN (ANDERSON et al., 2021), c'est-à-dire une seule couche convolutive suivie d'une ou plusieurs couches denses. Le préfixe *1D* indique simplement que l'entrée du modèle est un vecteur unidimensionnel des absorbances qui représentant un individus.

principe générale du modèle
L'objectif de cette architecture est d'extraire automatiquement les caractéristiques pertinentes du signal spectral. Le modèle 1D-CNN utilise des filtres de convolution qui parcourent le spectre unidimensionnel afin d'y détecter des motifs locaux significatifs, comme des bandes d'absorption ou des variations de pente. Les sorties issues de ces filtres sont ensuite combinées au fil des couches pour permettre au réseau d'apprendre des représentations de plus en plus complexes et abstraites. Cette approche permet au modèle de construire une représentation du signal directement guidée par la structure des données, et de réaliser, à travers les couches denses finales, une régression vers la variable chimique cible.

Le modèle de base est structuré comme suit : les caractéristiques spectrales sont extraites par le bloc de convolution, puis concaténées et transmises au bloc dense, qui est composé de paires de couches denses + *dropout*. Enfin, les données atteignent une couche de sortie dense avec une seule unité et une fonction d'activation linéaire, utilisée pour prédire les variables cibles.

La couche convolutive utilise un `padding="same"` et un déplacement le long du spectre (`stride=1`), ce qui signifie que des zéros sont ajoutés aux extrémités du vecteur d'entrée pour conserver sa taille originale après la convolution. Le nombre de filtres, la largeur des filtres, le nombre de couches denses, le nombre d'unités par couche, le taux de dropout, ainsi que la force de la régularisation L2 sont tous traités comme des hyperparamètres à optimiser dans le cadre du processus d'optimisation des hyperparamètres (HPO).

Il convient de noter que les paires *dense* + *dropout* ne sont utilisées que si le modèle comporte plus d'une couche dense. Si une seule couche dense précède la sortie, aucun dropout n'est appliqué. Même si le nombre de couches denses et la présence de dropout relèvent en principe de la recherche d'architecture neuronale (Neural Architecture Search, NAS), l'article (ZELA et al., 2018) propose de les inclure dans la procédure HPO pour des raisons de simplicité et d'efficacité.

Toutes les couches cachées utilisent la fonction d'activation ELU (Exponential Linear Unit) et sont régularisées par une pénalisation L2 uniforme afin de limiter le surapprentissage.

Deux variantes de cette architecture sont étudiées : **CNN-R_v1D** et **CNN-R_v1E**.
La version **CNN-R_v1D** suit la structure standard décrite plus haut, avec l'utilisation de plusieurs filtres de convolution de largeurs différentes, un `padding="same"`.

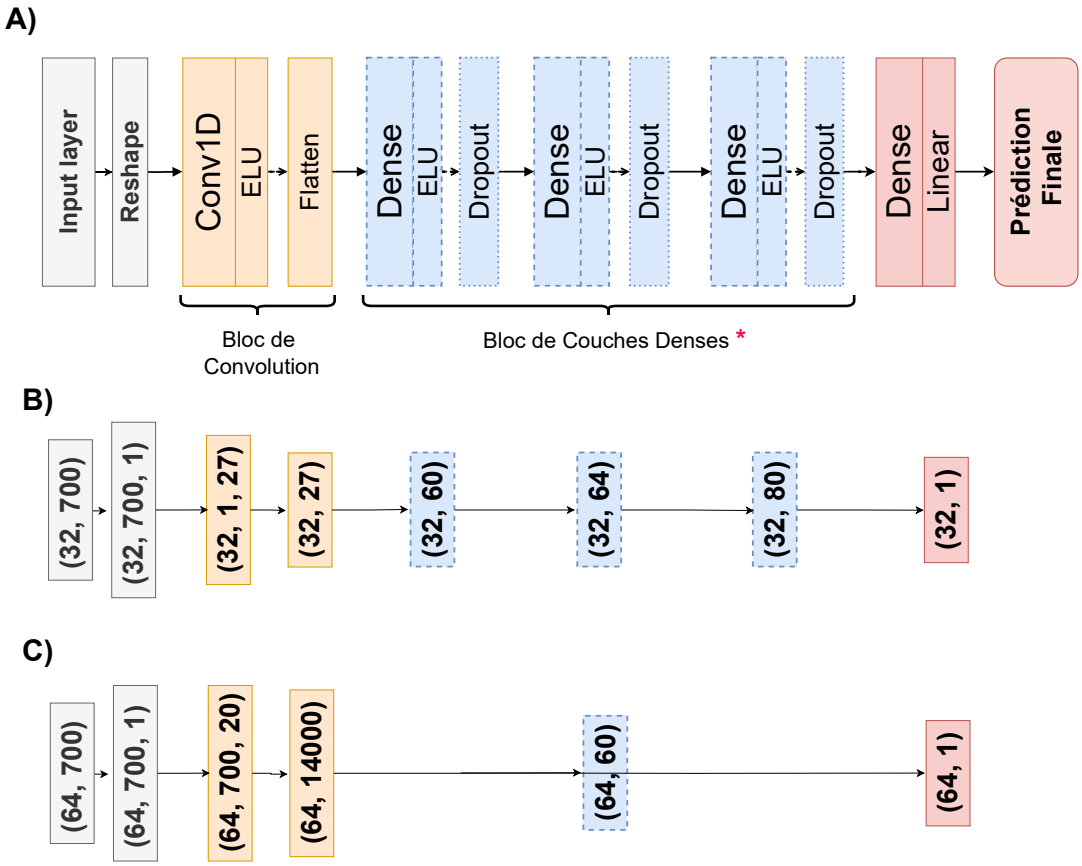


FIGURE 2.2 – Modèle 1D-CNN, utilisé comme squelette pour l'optimisation des prédictions de chaque variable cible. **(A)** Structure générale du modèle, (*) Le nombre de couches denses et l'utilisation ou non de *dropout* sont des hyperparamètres à optimiser. **(B)** Exemple de propagation des dimensions des données à travers les couches du meilleur modèle **CNN-R_v1E** obtenu pour la variable **adf** (*batch size* = 32 et 27 filtres). **(C)** Exemple de propagation des dimensions à travers les couches du meilleur modèle **CNN-R_v1D** obtenu pour la variable **adf** (batch size = 64 et 20 filtres).

La version **CNN-R_v1E**, quant à elle, utilise un seul filtre de convolution de largeur fixe égale à 700 — soit la taille totale du vecteur d'entrée — avec un `padding="valid"`. Cela signifie que la convolution ne se déplace pas le long du spectre, mais s'applique en une seule opération sur l'ensemble du vecteur, et on obtient une seule valeur.

Il convient de noter que l'article explore également la possibilité d'utiliser jusqu'à trois couches convolutives empilées. Afin de concentrer notre étude sur les architectures les plus prometteuses pour notre contexte, et considérant que les configurations à plusieurs couches convolutives ont montré des performances moindres dans l'étude de référence, nous avons choisi de nous limiter à une seule couche convolutive. Ce choix est également motivé par le fait que notre ensemble de données est plus petit que celui utilisé dans l'étude, et qu'il est donc pertinent de vérifier d'abord si une architecture plus simple peut suffire à obtenir de bonnes performances.

2.3.2.2 Modèle IPA (HAFFNER et al., 2025)

Le modèle IPA (Inception for Petroleum Analysis) proposé par (HAFFNER et al., 2025), est un réseau neuronal convolutif (CNN) profond, qui s'inspire directement de l'architecture *DeepSpectra*, un modèle proposé en 2019 (ZHANG, LIN et al., 2019), présenté comme un modèle *end-to-end* capable d'apprendre automatiquement à partir des données brutes, sans dépendre de techniques de prétraitement ou de sélection de variables. Mais pour ce travail, nous avons fait le choix d'utiliser les données prétraitées afin de maintenir une homogénéité de démarche entre tous les modèles.

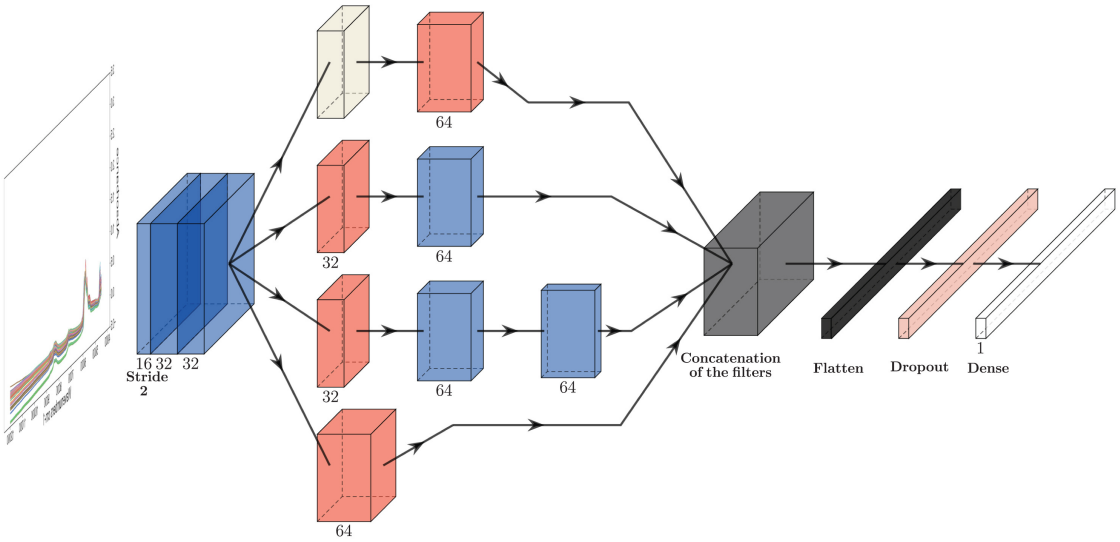


FIGURE 2.3 – Structure du modèle IPA. Les convolutions utilisant un noyau (*kernel*) de taille 1 sont représentées par un rectangle rouge et utilisent un pas (*stride*) de 2. Les convolutions avec un noyau de taille 3 sont représentées par un rectangle bleu. La couche de *max-pooling* est représentée par un rectangle orange clair.

Principe générale du modèle

Le modèle IPA s'inspire des architectures de type "Inception" pour réaliser une analyse multi-échelles du spectre. Plutôt que d'analyser le signal à une seule résolution (comme le 1D-CNN), son architecture se compose de plusieurs branches de convolution qui opèrent en parallèle avec des tailles de filtres différentes. Cette conception permet au modèle de capturer simultanément des informations à diverses échelles : des détails fins et localisés (via les petits filtres) ainsi que des tendances plus larges et globales (via les filtres plus grands). Les caractéristiques extraites de chaque échelle sont ensuite concaténées, fournissant une représentation complète du spectre qui est ensuite utilisée pour la tâche de régression.

Le modèle est spécifiquement conçu pour l'analyse de données spectroscopiques Proche Infrarouge dans le domaine pétrolier (utilisé pour prédire l'indice de cétane dans l'article. Son architecture, illustrée à la figure 2.3, se compose principalement de trois parties : un « tronc » (*stem*) initial, un module multi-branches inspiré des architectures de type *'Inception'*, et des couches finales pour la régression.

Le réseau débute par le tronc (*stem*), inspiré d'une version réduite d'un module Inception-v4 (SZEGEDY, LIU et al., 2015).

Un module Inception typique est composé de plusieurs branches de couches de convolution opérant en parallèle sur la même entrée. Ces branches utilisent des filtres de tailles différentes (par exemple, 1×1, 3×3, 5×5) et des opérations de *pooling*, permettant au réseau d'apprendre des caractéristiques à diverses échelles simultanément. Les sorties de ces branches parallèles sont ensuite concaténées, formant la sortie du module Inception. La version Inception-v4, par exemple, représente une de ces architectures Inception plus profondes et optimisées.

Le tronc du modèle IPA est constitué d'une séquence de trois couches convolutives 1D de base ('Conv1D'). Chaque couche 'Conv1D' comprend :

1. Une couche 'Conv1D' avec des noyaux de taille 3. La première couche 'Conv1D' a 16 filtres et un pas (stride) de 2. Les deux couches 'Conv1D' suivantes ont également 16 filtres mais une pas de 1.
2. Une fonction d'activation **LeakyReLU**.

Aucune opération de *padding* n'est utilisée dans les convolutions, ce qui signifie que la taille de la sortie diminue après chaque convolution (Voir le tableau 2.4).

Après le tronc, les données passent par un module multi-branches, nommé 'Module_35x35'. Ce module s'inspire du module 35×35 d'Inception-V2 (SZEGEDY, VANHOUCKE et al., 2015) qui vise à capturer des caractéristiques à différentes échelles en parallèle. Il prend en entrée 16 canaux. Le module se compose de quatre branches parallèles :

Branche	**Couches**
Branche 1	`MaxPooling1D (pool_size=2, strides=2)` `Conv1D (filters=64, kernel_size=1, strides=2)`
Branche 2	`Conv1D (filters=32, kernel_size=1, strides=2)` `Conv1D (filters=64, kernel_size=3, strides=1)`

Branche 3	`Conv1D (filters=32, kernel_size=1, strides=2)`
	`Conv1D (filters=64, kernel_size=3, strides=2)`
	`Conv1D (filters=64, kernel_size=3, strides=2)`

Branche 4	`Conv1D (filters=64, kernel_size=1, strides=2)`

TABLE 2.3 – Structure du module multi-branche utilisé dans l'architecture IPA.

Chaque 'Conv1D' dans ce module utilise également l'activation LeakyReLU. La concaténation des sorties des quatre branches va permettre d'agréger des représentations diverses et riches du signal (ZHANG, LIN et al., 2019).

Après le module multi-branches, les caractéristiques extraites passent par :

1. Une couche *'Flatten()'* qui transforme la matrice de données 2D en un vecteur 1D.
2. Une couche *'Dropout'* avec un taux d'exclusion de 20%.
3. Une couche Dense (entièrement connectée) avec un seul neurone et une activation linéaire, pour prédire la valeur finale.

Au total, le modèle intègre dix couches de convolution. La taille maximale des noyaux est de 3, et le nombre total de paramètres du modèle est d'environ 59 000.

Layer / Block	**Output Shape**
Input Data	(16, 700, 1)
Stem Block	
Layer 1 : (filters=16, ks=3, s=2)	(16, 349, 16)
Layer 2 : (filters=16, ks=3, s=1)	(16, 347, 16)
Layer 3 : (filters=16, ks=3, s=1)	(16, 345, 16)
Output du tronc	*(16, 345, 16)*
Module_35x35	
Input en Module_35x35	*(16, 345, 16)*
Branch 1 Output	(16, 86, 64)
Branch 2 Output	(16, 171, 64)
Branch 3 Output	(16, 42, 64)
Branch 4 Output	(16, 173, 64)
Output après Concaténation (axis=1)	*(16, 472, 64)*
Couche Flatten	(16, 30208)
Couche Dropout	(16, 30208)
Couche Regressor	(16, 1)
Output Final	**(16, 1)**

TABLE 2.4 – propagation des dimensions des données à travers le modèle. ks = kernel_size, s = strides.

Fonction de Perte Personnalisée

Pour ce modèle, ils ont utilisés une fonction de perte personnalisée. Cette fonction combine deux composantes principales : l'erreur absolue moyenne (MAE) et le terme de régularisation L2.

Mathématiquement, la fonction de perte s'écrit comme suit :

$$\text{Loss}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| + \lambda \cdot \|\mathbf{w}\|_2$$

où w représente les poids du modèle, y_i les valeurs cibles, $\hat{y}_i$ les prédictions, et λ le coefficient de régularisation qui est ajusté expérimentalement.

La première partie de la fonction est l'**Erreur Absolue Moyenne (MAE)**. il a été choisi car elle est moins sensible aux valeurs aberrantes que la plus commune Erreur Quadratique Moyenne (MSE). Comme elle calcule la différence absolue directe entre la valeur mesurée (y_i) et la valeur prédite ($\hat{y}_i$), elle empêche quelques points de données anormaux de fausser de manière disproportionnée le processus d'entraînement.

La seconde partie est un terme de **régularisation L2**, qui pénalise les poids élevés ($\mathbf{w}$) dans les couches convolutionnelles du réseau. Son rôle principal est de lutter contre le surajustement (overfitting). Il sert à empêcher le modèle de dépendre trop fortement d'un petit nombre de caractéristiques potentiellement bruitées. En effet, il force le réseau à

accorder moins d'importance aux caractéristiques spectrales non pertinentes en réduisant leurs poids correspondants.

Le coefficient **lambda** (λ) est l'hyperparamètre qui contrôle l'équilibre entre ces deux objectifs.

2.3.2.3 Modèle 1D-CAE (CATALTAS et TUTUNCU, 2023)

Les autoencodeurs (AE) sont des algorithmes de réseaux de neurones génératifs et non supervisés dont l'objectif principal est de produire des valeurs de sortie égales aux valeurs d'entrée. Ils sont classifiés comme des méthodes d'apprentissage non supervisé en raison de cette propriété. Un cadre d'autoencodeur se compose de deux blocs principaux : un encodeur et un décodeur. Le bloc encodeur compresse les données d'entrée en une représentation de faible dimension appelée variables latentes, qui contient des informations pertinentes sur les données d'entrée. Le bloc décodeur prend ces variables latentes en entrée et tente de reconstruire les données originales. Dans le contexte de la spectroscopie proche infrarouge (NIRS), les autoencodeurs sont souvent utilisés pour l'extraction de caractéristiques, le débruitage ou la détection d'anomalies.

Le modèle proposé dans cette étude est un autoencodeur convolutif unidimensionnel (1D-CAE). La première phase consiste en un entraînement non-supervisé de l'architecture complète, composée d'un encodeur et d'un décodeur. L'encodeur compresse le spectre d'entrée en un vecteur de "variables latentes", et le décodeur est entraîné à reconstruire le spectre original à partir de ce même vecteur. Cette tâche de compression-reconstruction contraint l'encodeur à apprendre une représentation optimisée qui distille l'information du signal d'entré dans l'espace latent. Pour la seconde phase, qui est la tâche de régression supervisée, seul l'encodeur entraîné est conservé, il est alors utilisé pour transformer les spectres en leurs variables latentes correspondantes

L'architecture du modèle 1D-CAE (voir Figure 2.4) se divise en deux sous-modèles : l'encodeur et le décodeur, avec un goulot d'étranglement (*bottleneck*) entre les deux.

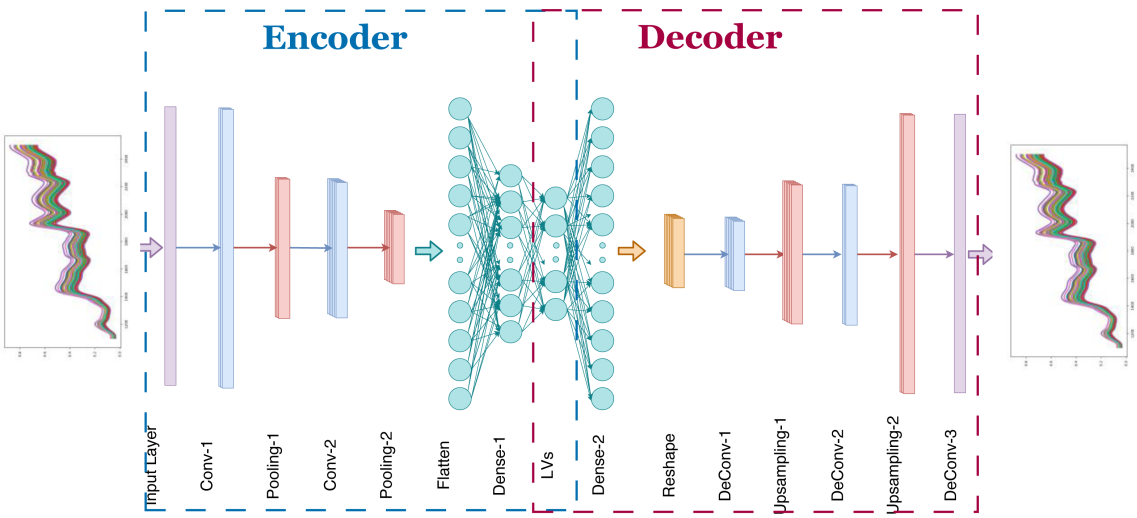


FIGURE 2.4 – Architecture du modèle 1D-Convolutional Autoencoder proposé. L'encodeur (gauche) compresse le spectre NIRS en variables latentes, et le décodeur (droite) tente de reconstruire le spectre original à partir de ces variables.

Sous-modèle Encodeur

Le sous-modèle encodeur est conçu pour compresser le spectre NIRS d'entrée (700 caractéristiques) en une représentation de dimension inférieure. Il est composé des couches suivantes (détaillées dans le Tableau 2.5) :

— Deux couches `Conv1D` : La première avec 16 filtres et la seconde avec 32 filtres, toutes deux utilisant une taille de noyau de (5). Ces couches sont responsables de l'extraction de caractéristiques hiérarchiques du spectre.
— Deux couches `MaxPooling1D` : Suivent chaque couche convolutive, avec une taille de pool de (2). Elles réduisent la dimensionnalité des caractéristiques extraites sans modifier le nombre de canaux.
— Une couche `Flatten` : Convertit la sortie des couches convolutives et de pooling en un vecteur unidimensionnel pour l'entrée des couches denses.
— Deux couches `Dense` : La première avec 64 unités, et la seconde, qui représente le goulot d'étranglement, avec 32 unités. Ces couches compressent davantage l'information en variables latentes. Le choix de 32 neurones pour les variables latentes a été fait après avoir constaté que moins de neurones n'apportaient pas de succès notable.

Le choix de deux couches convolutives empilées est motivé par des travaux antérieurs qui ont montré que deux ou trois couches sont suffisantes pour les applications NIRS basées sur les CNN.

Sous-modèle Décodeur

Le sous-modèle décodeur prend les variables latentes du goulot d'étranglement et tente de reconstruire le spectre NIRS original. Il est composé des couches suivantes (détaillées dans le Tableau 2.6) :

— Une couche `Dense` suivie d'une couche `Reshape` pour remodeler la sortie en une forme adaptée aux couches de déconvolution.
— Trois couches `Conv1D_Transpose` : Avec respectivement 32, 16, et 1 filtre(s) et une taille de noyau de (3). Ces couches effectuent une opération inverse de la convolution pour reconstruire les caractéristiques.
— Deux couches `UpSampling1D` : Intercalées entre les couches de déconvolution, avec une taille de (2). Elles augmentent la dimensionnalité spatiale des caractéristiques, inversant l'effet du pooling. La couche finale reconstruit le spectre original à 700 caractéristiques.

Toutes les couches convolutives et denses utilisent la fonction d'activation tangente hyperbolique (`tanh`) pour introduire de la non-linéarité.

Fonction de Perte et Optimisation La fonction de perte utilisée pour évaluer la progression de l'apprentissage du réseau est l'erreur quadratique moyenne (MSE). L'optimisation est réalisée à l'aide de l'optimiseur Adam. Les valeurs sélectionnées pour le taux d'apprentissage (η), β_1 et β_2 sont respectivement 0.001, 0.9 et 0.999.

Variables Latentes et Régression Linéaire Multiple (MLR) Après l'entraînement, les données sont passées à travers le sous-modèle encodeur pour obtenir les variables latentes (dont le nombre est optimisé dans la phase de HPO), et on effectue une régression linéaire

multiple pour faire la prédiction.

TABLE 2.5 – Description du sous-modèle encodeur de l'architecture du 1D-CAE.

No	Type	Filtres	Noyau/Pool	Stride	Sortie	Paramètres
1	Input	-	-	-	(700,1)	0
2	`Conv1D`	16	(5)	1	(700, 16)	96
3	`MaxPooling1D`	-	(2)	2	(350, 16)	0
4	`Conv1D`	32	(5)	1	(350, 32)	2,592
5	`MaxPooling1D`	-	(2)	2	(175, 32)	0
6	`Flatten`	-	-	-	(5600)	0
7	`Dense`	-	-	-	(64)	358,464
8	`Dense`	-	-	-	(32)	2,080

TABLE 2.6 – Description du sous-modèle décodeur de l'architecture du 1D-CAE.

No	Type	Filtres	Noyau/Pool	Stride	Sortie	Paramètres
1	Input	-	-	-	(32)	0
2	`Dense`	-	-	-	(5600)	184,800
3	`Reshape`	-	-	-	(175, 32)	0
4	`Conv1D_Transpose`	32	(3)	1	(175, 32)	3,104
5	`UpSampling1D`	-	(2)	-	(350, 32)	0
6	`Conv1D_Transpose`	16	(3)	1	(350, 16)	1,552
7	`UpSampling1D`	-	(2)	-	(700, 16)	0
8	`Conv1D_Transpose`	1	(3)	1	(700,1)	49

2.4 Stratégies d'optimisation des hyperparamètres en DL

À partir de la littérature, nous avons identifié et regroupé les stratégies les plus couramment utilisées pour l'optimisation des hyperparamètres et le choix du modèle en DL. Voici les principales approches rencontrées dans les travaux récents :

a) Méthode Classique de *machine learning*

C'est la stratégie classique la plus utilisée ((MISHRA et PASSOS, 2021), (PASSOS et MISHRA, 2022), (PRECHELT, 2012)). Elle consiste à séparer les données en trois sous-ensembles : calibration, validation (ou *tuning*) et test. Les ensembles de calibration et de validation sont utilisés pour l'optimisation des hyperparamètres, et l'ensemble de test est réservé exclusivement pour évaluer les performances du modèle final. Après avoir optimisé les hyperparamètres, une bonne pratique consiste à concaténer les ensembles de calibration et de validation en un seul ensemble d'entraînement, afin de ré-entraîner le modèle optimisé en utilisant toutes les données disponibles, puis procéder à l'évaluation finale sur l'ensemble de test.

b) Méthode de *cross-validation*

Cette stratégie est davantage employée pour les petits jeux de données. Elle consiste à diviser les données en deux parties : un ensemble d'entraînement et un ensemble de test, selon des proportions telles que 80% / 20% (MARTINS et al., 2022), 75% / 25% (CATALTAS et TUTUNCU, 2023) ou encore 70% / 30% (HAFFNER et al., 2025).

L'optimisation des hyperparamètres se fait ensuite par validation croisée *k-fold* sur l'ensemble d'entraînement uniquement. Cette méthode revient globalement à la stratégie classique, mais elle est considérée comme plus robuste, car elle permet d'entraîner et de valider le modèle sur l'ensemble complet des données d'entraînement.

c) Méthode d'entraînement sur le jeu de calibration

Cette méthode débute comme l'approche classique, consiste à découper en jeux de calibration (Cal), de validation (Val) et de test, et optimiser les hyperparamètres sur Cal et Val. La différence réside dans le fait que, pour le modèle final, l'entraînement se fait uniquement sur Cal, puis l'évaluation est sur le jeu de test. L'hypothèse est que les hyperparamètres optimaux trouvés sur Cal ne s'appliquent pas nécessairement à l'ensemble d'entrainement. En conséquence, on conserve seulement Cal pour l'entraînement final, au détriment de Val qui n'est pas réutilisé pour l'évaluation finale, ce qui peut entraîner une perte d'information si le jeu de donnée est n'est pas assez grand.

Pour ce travail, nous allons adopter la stratégie de cross-validation, adaptée aux petits jeux de données, afin d'homogénéiser la méthodologie pour l'ensemble des modèles présentés dans ce rapport. Pour plus de détails sur l'implémentation de la cross validation pour chaque modèle, voir la section 3.3.

Chapitre 3

Matériel et méthodes

3.1 Présentation du jeu de données

3.1.1 Source et description des données

Les données utilisées dans cette étude ont été fournies par le CIRAD. Les échantillons de plantes — qui incluent des graminées, des herbacées, des légumineuses et des arbustes, souvent sous forme de mélanges de fourrages — proviennent principalement des régions méditerranéennes, de l'île de la Réunion et de zones sahéliennes (Burkina Faso, Tchad, Mali, Sénégal). En parallèle, les données sur les aliments ont une portée plus globale, ayant été collectées dans plus de 150 pays et englobant des aliments complets, des matières premières, ainsi que des sous-produits d'origine agro-industrielle et animale.

La base de notre étude est un jeu de données avec une structure et une partition spécifiques. Les données sont organisées en trois fichiers distincts, comme illustré par la Figure 3.1 :

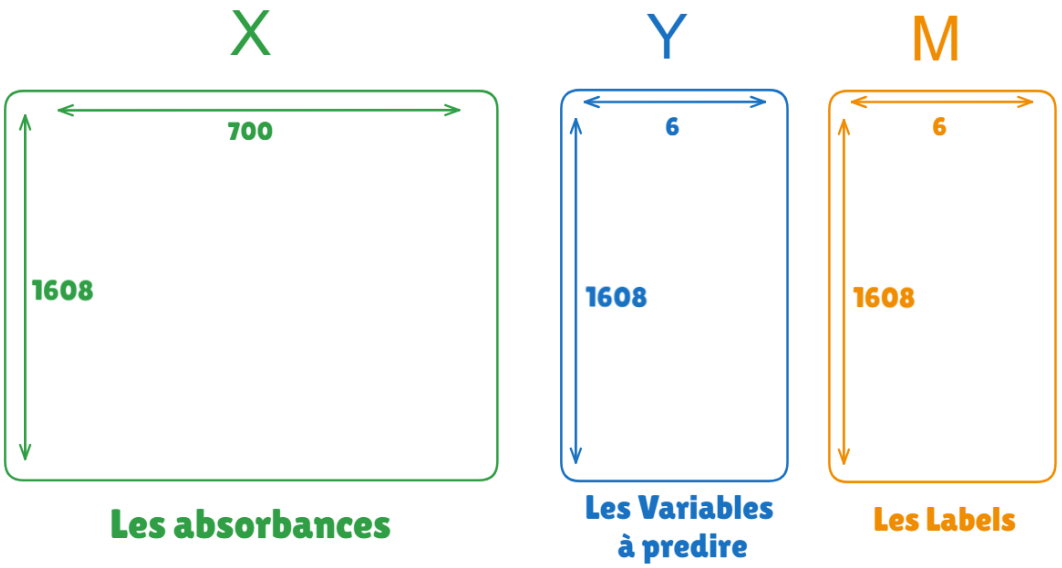


FIGURE 3.1 – Schéma des Matrices de Données

Fichier X.csv

Contient les valeurs des absorbances de 1608 échantillons mesurées à 700 longueurs d'onde, sur une plage de 1100 nm à 2498 nm avec une résolution spectrale de 2 nm. Ils ne contient pas de valeurs manquantes. (Voir section 3.2 Pour l'analyse exploratoire)

Prétraitements appliqués

Comme déjà discuté dans la section 2.1.2, le prétraitement des données spectrales est une étape indispensable en chimiométrie, dont l'objectif est de raffiner le signal et de corriger les artefacts induits par l'instrumentation ou liés aux propriétés physiques de l'échantillon.

Pour notre cas, les données ont été fournies déjà prétraitées, avec une normalisation par la méthode SNV, suivie d'un lissage selon la méthode de Savitzky-Golay.

(a) Standard Normal Variate (SNV)

La transformation **Standard Normal Variate (SNV)** est une méthode de prétraitement utilisée pour corriger les effets de diffusion de la lumière dans les données spectrales.

La méthode SNV normalise chaque spectre, en soustrayant sa propre moyenne puis en le divisant par son propre écart-type. Ce processus **centre et réduit** chaque spectre, pour que les variations observées entre les spectres sont davantage liées à la composition chimique qu'aux effets physiques.

Formule

Pour un spectre unique x composé de p longueurs d'onde ($x = \{x_1, x_2, ..., x_p\}$), la transformation SNV est appliquée comme suit :

$$x_{SNV} = \frac{x - \bar{x}}{\sigma_x}$$

Où

— $\bar{x}$ est la moyenne des x_i

— σ_x est l'écart-type.

Le spectre transformé donc possède une moyenne de 0 et un écart-type de 1.

(b) Dérivation de Savitzky-Golay

Le filtre de **Savitzky-Golay** est une technique de filtrage numérique utilisée pour le lissage de données et le calcul de dérivées. Dans le cadre de cette étude, il est utilisé pour calculer la **dérivée seconde** des spectres.

L'application d'une dérivée à un spectre présente plusieurs avantages :

— **Résolution de pics superposés :** Elle améliore la résolution des pics qui sont proches les uns des autres.

— **Élimination des dérives de la ligne de base :** Elle aide à corriger les décalages de ligne de base constants et linéaires dans les données.

Le filtre fonctionne par l'ajustement d'une fonction polynomiale sur une petite fenêtre mobile de points de données. La valeur du point central de la fenêtre est ensuite remplacée

par la valeur de la dérivée de ce polynôme ajusté. Pour cette analyse, un **polynôme de second ordre**.

Formule

La procédure de Savitzky-Golay est essentiellement une somme pondérée (ou une convolution). Pour une fenêtre de taille $2m + 1$ points, la valeur filtrée x_j^* à la position j est calculée comme suit :

$$x_j^* = \sum_{i=-m}^{m} c_i x_{j+i}$$

Où :

— x_j est le point de donnée original.
— c_i sont les coefficients du filtre.

Ces coefficients sont pré-calculés en fonction de l'ordre du polynôme et de l'ordre de la dérivation. Pour une **dérivée seconde**, les coefficients sont choisis de manière à fournir la valeur de la dérivée seconde du polynôme ajusté à la fenêtre de données.

Fichier Y.csv

Contient les valeurs des 6 composantes chimiques à prédire pour les mêmes échantillons dans le fichier X.csv. Une brève explication de chaque variable est présentée ci-dessous :

— **cp (Matières Azotées Totales) :** Cet indicateur, obtenu en multipliant la teneur en azote par 6,25, est une estimation de la teneur totale en protéines du fourrage. C'est un critère essentiel pour évaluer la valeur nutritive.
— **ndf (Neutral Detergent Fiber) :** Cette fraction représente l'ensemble des parois cellulaires végétales (hémicellulose, cellulose et lignine), qui sont lentement ou partiellement digestibles. La teneur en NDF est inversement corrélée à la capacité d'ingestion du fourrage par l'animal.
— **adf (Acid Detergent Fiber) :** Il s'agit de la fraction la moins digestible des parois cellulaires, composée principalement de cellulose et de lignine. L'ADF est un indicateur clé de la digestibilité de l'énergie du fourrage : plus sa teneur est élevée, moins le fourrage est digestible.
— **adl (Acid Detergent Lignin) :** La lignine est un composé phénolique qui rend les parois cellulaires rigides et indigestibles. En tant que principal facteur limitant la digestibilité des fibres, une teneur élevée en ADL réduit significativement la valeur énergétique du fourrage.
— **cf (Cellulose Brute) :** Mesurée par la méthode de Weende, cette fraction estime la teneur en cellulose, bien qu'elle puisse inclure une partie de la lignine et de l'hémicellulose. Elle est souvent utilisée comme un indicateur de la teneur en fibres totales.
— **dmdcell (Digestibilité de la Matière Sèche) :** Cette mesure, obtenue in vitro par une méthode enzymatique (pepsine-cellulase), simule la digestion dans le rumen. Elle représente la proportion de la matière sèche totale qu'un animal est capable de digérer et d'utiliser, ce qui en fait un excellent indicateur de la valeur énergétique globale du fourrage.

Variable	Unité	Label
cp	%MS	Matière azotée (Nx6.25)
ndf	%MS	Neutral Detergent Fiber (Méthode Van Soest)
adf	%MS	Acid Detergent Fiber (Méthode Van Soest)
adl	%MS	Acid Detergent Lignin (Méthode Van Soest)
cf	%MS	Cellulose brute de Weende
dmdcell	%MS	Digestibilité enzymatique in-vitro de la matière sèche (Méthode pepsine-cellulase Aufrère)

TABLE 3.1 – Liste des variables avec unités et labels

Fichier M.csv

Ce fichier définit la répartition des échantillons pour les six variables cibles. Pour chaque variable, un échantillon est assigné à l'un des quatre statuts suivants :

— **'cal'** (Calibration) : pour l'entraînement du modèle.
— **'val'** (Validation) : pour l'optimisation des hyperparamètres.
— **'test'** (Test) : pour l'évaluation finale de la performance du modèle.
— **'missing'** : pour les valeurs de référence manquantes, qui sont exclues de l'analyse.

Les données nous ont été fournies déjà partitionnées selon ce fichier, comme illustré par le Tableau 3.2. La composition des ensembles de calibration, validation et test varie d'une variable chimique à l'autre. Autrement dit, un même échantillon spectral peut être utilisé pour la calibration d'une variable, mais pour le test ou être manquant pour une autre. Cette structure impose donc la construction de six modèles indépendants, un par variable cible, afin de respecter les répartitions spécifiques à chacune et pour exclure les variables manquantes.

TABLE 3.2 – Extrait du fichier de partitionnement M.csv.

adf	adl	cf	cp	dmdcell	ndf
cal	cal	missing	test	test	val
test	test	missing	val	cal	test
cal	val	missing	val	cal	val
test	test	test	cal	cal	val
cal	test	cal	test	test	cal
.	.	.	.	.	.

La répartition des échantillons via le fichier M.csv permet de définir, pour chaque variable cible, deux ensembles de données finaux : un ensemble d'entraînement (Train) et un ensemble de test (Test). L'ensemble d'entraînement est formé en combinant tous les échantillons étiquetés *cal* et *val*. C'est sur cet ensemble que les modèles sont entraînés et que leurs hyperparamètres sont optimisés via une validation croisée. L'ensemble de test, mis de côté durant tout ce processus, sert uniquement à l'évaluation finale et non biaisée des modèles optimisés. Le Tableau 3.3 détaille les effectifs de ces partitions, qui varient

pour chaque variable.

TABLE 3.3 – Effectifs des partitions finales d'entraînement (75%) et de test (25%) pour chaque variable cible. Le total initial est de 1608 échantillons.

Partition	**adf**	**adl**	**cf**	**cp**	**dmdcell**	**ndf**
Entraînement ('Train')	1148	1067	666	1173	1094	1147
Test ('Test')	382	356	222	391	365	382
Total disponible	**1530**	**1423**	**888**	**1564**	**1459**	**1529**

3.2 Analyse exploratoire

Dans cette section, nous procédons à l'analyse exploratoire de notre jeu de données. On commence d'abord par les caractéristiques des données spectrales (matrice X) dans la sous-section 3.2.1, puis sur celles des variables chimiques à prédire (matrice Y) dans la sous-section (a).

3.2.1 Analyse des données spectrales (Matrice X)

L'analyse exploratoire débute par l'étude de la matrice des spectres X. Les données spectrales fournies ayant déjà été prétraitées, cette section vise à visualiser leur aspect général et à explorer leur structure multidimensionnelle. Pour ce faire, une Analyse en Composantes Principales (ACP) est employée afin de réduire la dimensionnalité des données et LES visualiser.

On observe que les spectres de l'ensemble de test se situent entièrement dans l'amplitude et la distribution de variation de ceux de l'ensemble d'entraînement. Cette similarité visuelle est un bon indicateur que l'ensemble de test est bien représentatif de la population des données d'entraînement

Une Analyse en Composantes Principales (ACP) a été réalisée sur l'ensemble des données spectrales d'entraînement pour visualiser la structure des échantillons dans un espace de dimension réduite et de valider la partition des données. Pour illustrer la méthodologie et les résultats, les graphiques correspondant à la partition de la variable *adf* sont présentés ci-dessous. Une tendance de répartition similaire est observée pour les cinq autres variables, et leurs graphiques respectifs sont disponibles en Annexe B.

D'après la figure 3.3, on observe que les trois premières composantes principales expliquent environ 72% de la variance, rendant leur projection particulièrement pertinente pour l'analyse exploratoire des données.

Les projections des échantillons sur les plans PC1-PC2 et PC2-PC3 (figure 3.4) confirment la bonne répartition des données. L'observation la plus importante est que les échantillons de l'ensemble du test (en rouge) sont distribués de manière homogène au sein du nuage de points formé par l'ensemble d'entraînement (en bleu). Aucun échantillon de test ne semble être une aberration. Cette analyse confirme la représentativité de l'échantillonnage du test.

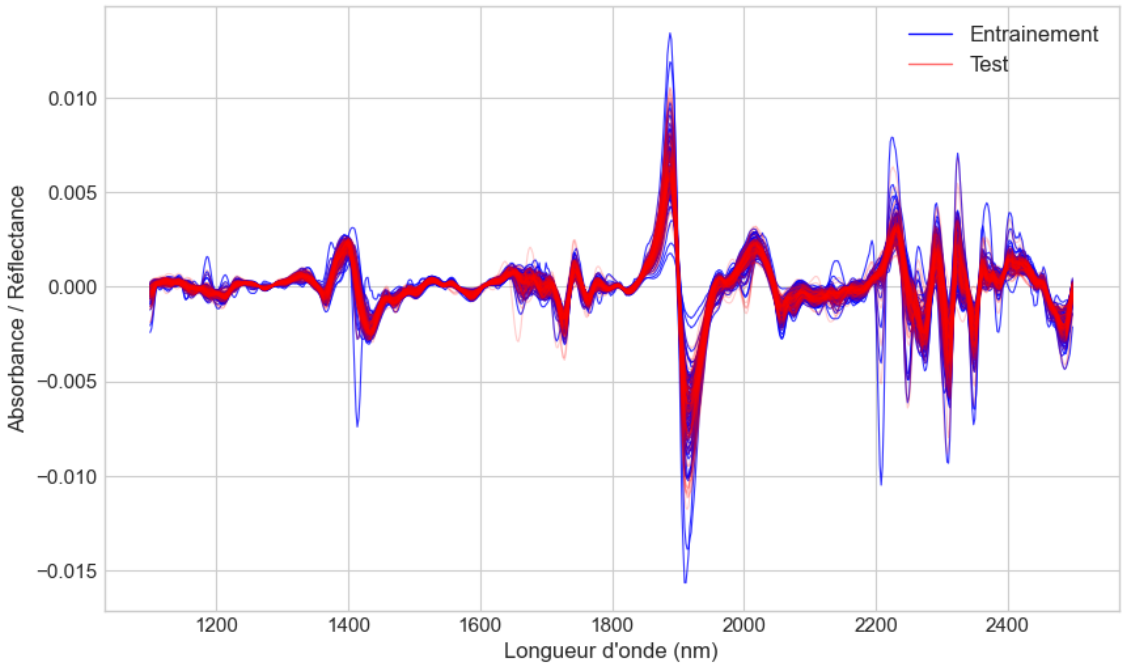


FIGURE 3.2 – Superposition des spectres proche infrarouge (PIR) prétraités pour un échantillon de 50 observations, pour illustrer la similarité entre les ensembles d'entraînement et de test.

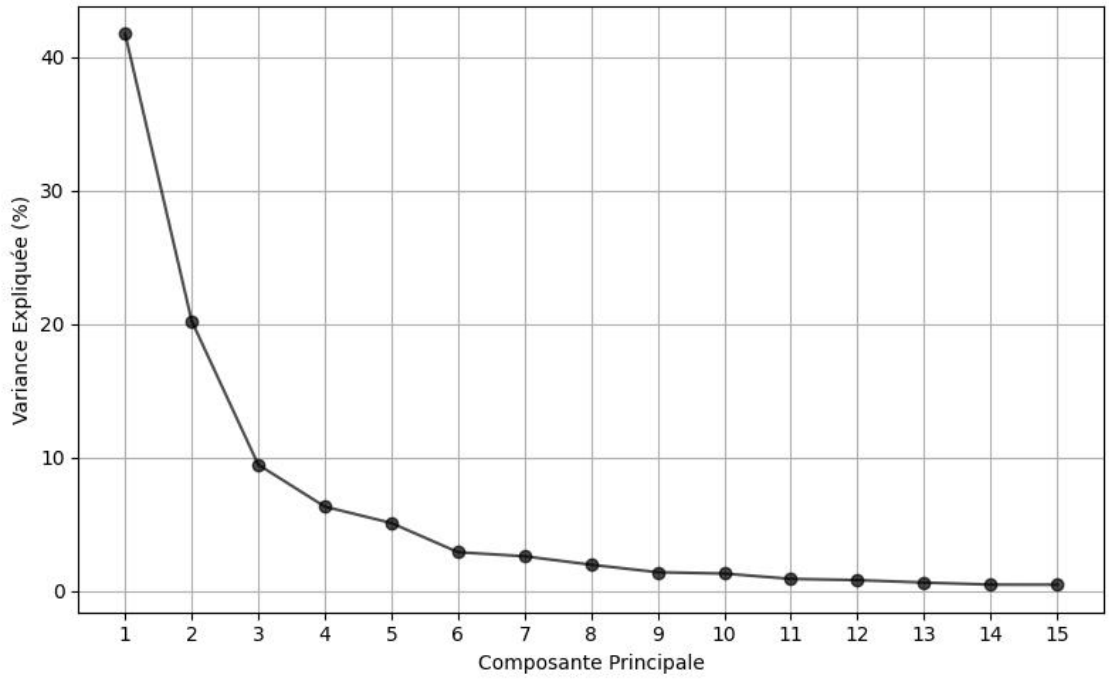
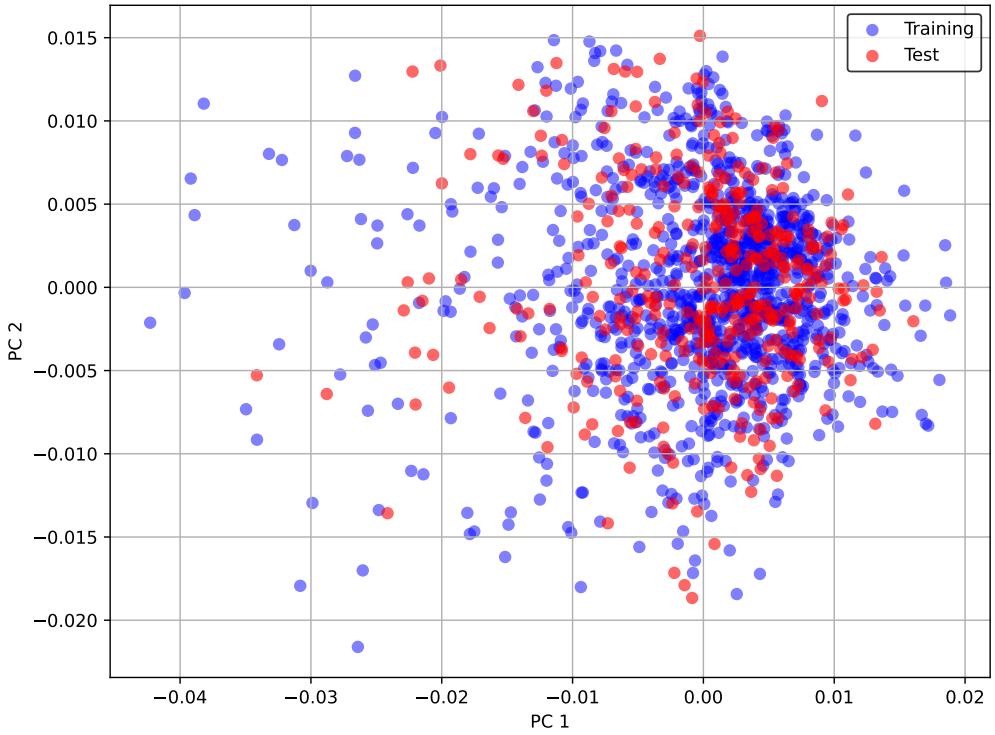
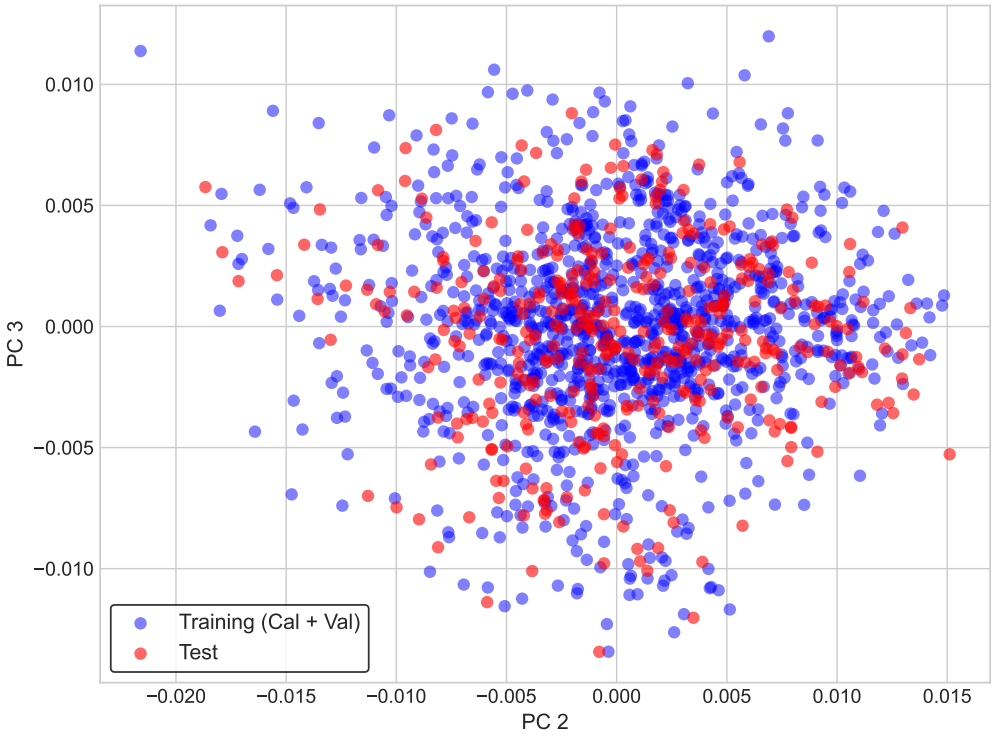


FIGURE 3.3 – Distribution de la variance expliquée par chacune des composantes principales extraites de l'ensemble d'entraînement.



(a) Projection sur le plan principale (PC1 et PC2)



(b) Projection sur le plan 2-3 (PC2 et PC3)

FIGURE 3.4 – Projections des ensembles d'entraînement et de test sur différents plans des composantes principales, pour la variable `adf`.

3.2.2 Analyse des variables chimiques cibles (Matrice Y)

(a) Présentation du tableau des statistiques descriptives

TABLE 3.4 – Statistiques descriptives des variables cibles.

Variable	Statistique	Entraînement	Test
adf	**Moyenne**	**33.221**	**33.232**
	Médiane	32.945	32.940
	Écart-type	9.884	9.998
	Min	9.200	8.770
	Max	65.980	66.910
adl	**Moyenne**	**10.950**	**10.983**
	Médiane	8.760	8.780
	Écart-type	7.678	7.778
	Min	0.930	0.730
	Max	42.820	43.070
cf	**Moyenne**	**28.803**	**28.808**
	Médiane	28.850	28.840
	Écart-type	9.048	9.172
	Min	7.930	7.810
	Max	52.140	52.210
cp	**Moyenne**	**11.375**	**11.393**
	Médiane	11.010	10.990
	Écart-type	4.575	4.658
	Min	1.980	1.590
	Max	31.260	32.340
dmdcell	**Moyenne**	**53.300**	**53.299**
	Médiane	53.485	53.490
	Écart-type	16.438	16.582
	Min	10.230	9.900
	Max	94.210	95.000
ndf	**Moyenne**	**52.680**	**52.670**
	Médiane	52.510	52.510
	Écart-type	13.459	13.589
	Min	17.480	15.990
	Max	84.890	85.720

Le Tableau 3.4 présente, pour chaque variable, des valeurs de moyenne, de médiane, d'écart-type, de minimum et de maximum très proches entre l'ensemble d'entraînement et l'ensemble de test. Cette cohérence est un premier indicateur qui confirme la qualité et la représentativité de la partition des données.

(b) Analyse visuelle des distributions

Pour compléter cette analyse, une inspection visuelle des distributions est réalisée. La Figure 3.6 (les diagrammes en boîtes) confirme les observations du tableau : les médianes et les dispersions sont quasiment identiques entre les ensembles d'entraînement et de test pour toutes les variables.

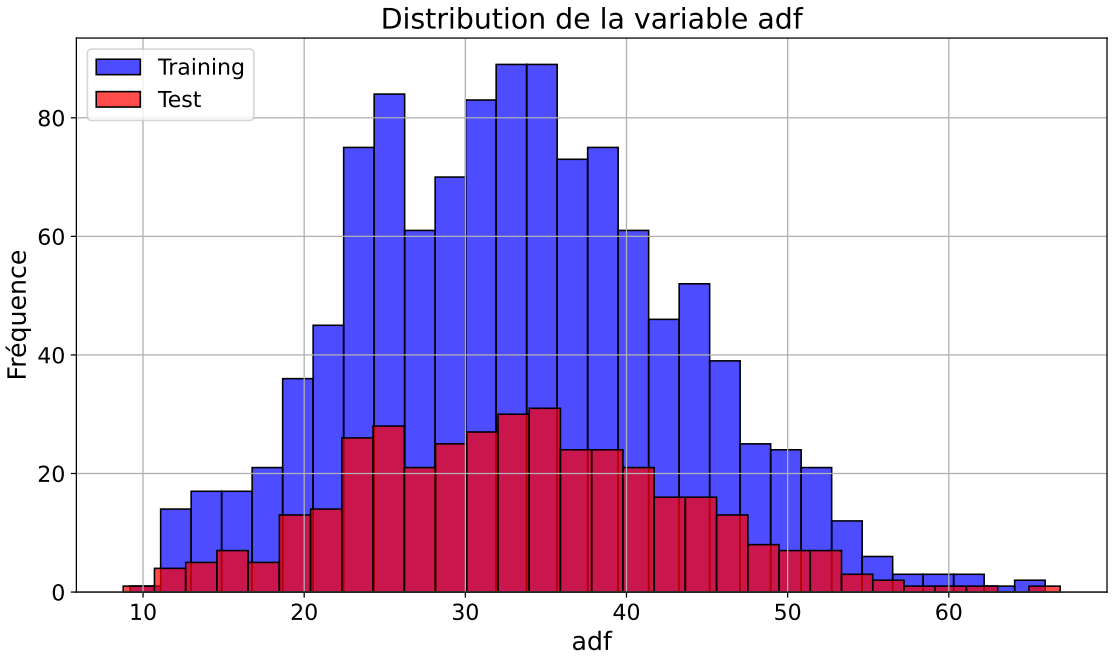


FIGURE 3.5 – Comparaison des histogrammes de fréquence pour la variable adf entre les ensembles d'entraînement et de test

Afin d'examiner la forme de ces distributions plus en détail, l'histogramme de la variable *adf* est présenté à titre d'exemple sur la Figure 3.5. Le graphique confirme que la distribution de l'ensemble de test épouse parfaitement celle de l'ensemble d'entraînement. Les histogrammes pour les autres variables, présentant des profils similaires, sont disponibles en Annexe B.

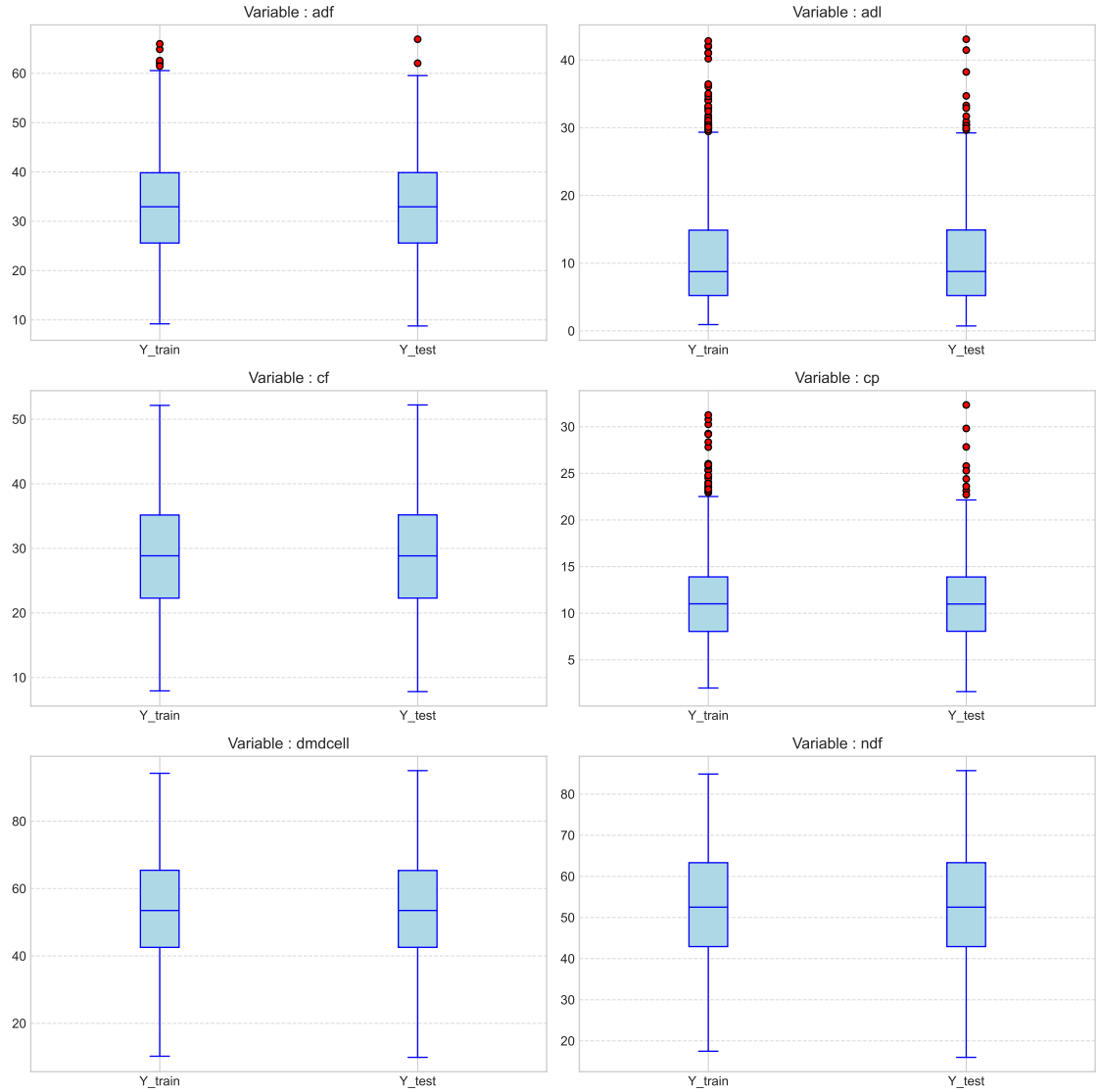


FIGURE 3.6 – Comparaison des distributions des variables cibles entre les ensembles d'entraînement et de test

3.3 Stratégie de modélisation et d'évaluation

3.3.1 Cadre expérimental général

Afin d'assurer la cohérence et la reproductibilité de notre étude, un protocole expérimental commun a été défini et appliqué à l'ensemble des modèles. Cette section détaille le flux de travail complet pour le développement et l'évaluation des modèles. La stratégie globale, illustrée à la 3.7, consiste à partitionner les données en un ensemble d'entraînement (75 %) et un ensemble de test (25 %). Les hyperparamètres de chaque modèle sont ensuite optimisés par une validation croisée à 5 plis (5-fold cross-validation) sur l'ensemble d'entraînement. Finalement, le modèle final, entraîné avec les meilleurs hyperparamètres sur la totalité des données d'entraînement, est évalué sur l'ensemble de test mis de côté.

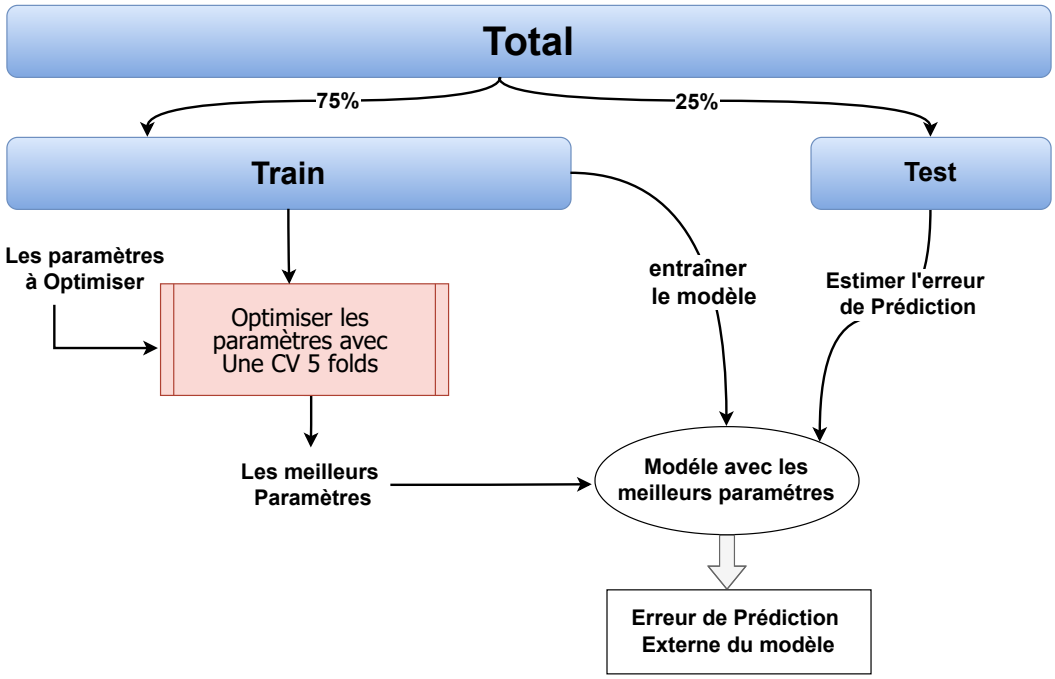


FIGURE 3.7 – Stratégie de partitionnement des données et d'optimisation des hyperparamètres pour l'entraînement des modèles dans ce rapport

Pour les modèles *Deep learning*, on commence par standardiser les matrices des X d'entrainement et de test en retranchant la moyenne et divisant par l'écart-type de chaque variable spectrale de X_train, ce qui est le pratique le plus utilisé ((MISHRA et PASSOS, 2021) ; (HAFFNER et al., 2025) ; (PASSOS et MISHRA, 2023)) pour accélérer la convergence des algorithmes DL et améliorer la stabilité numérique.

3.3.2 Les métriques d'évaluation des performances utilisées

Les métriques suivantes ont été sélectionnées pour évaluer et comparer les performances des modèles :

(a) Erreur quadratique moyenne de prédiction (RMSEP)

RMSEP est l'un des indicateurs les plus utilisés dans la littérature (PLEVRIS et al., 2022) pour évaluer la performance des modèles de régression sur un jeu de données de test. Elle mesure l'écart moyen quadratique entre les valeurs prédites par le modèle et les valeurs observées, selon la formule suivante :

$$\text{RMSEP} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

où y_i représente la valeur réelle, $\hat{y}_i$ la valeur prédite, et n le nombre total d'échantillons dans l'ensemble de test.

(b) Erreur relative (RE)

Erreur relative (ou *RMSEP* relative ou *RMSEP* normalisée), exprimée en pourcentage de la valeur moyenne observée, est une autre métrique utilisée dans ce travail, calculé en divisant le *RMSEP* par la moyenne des valeurs prédites :

$$\text{RE} = \frac{\text{RMSEP}}{\mu}$$

ou RMSEP est l'érreur quadratique moyenne de prédiction et μ est la moyenne des valeurs observées y_i.

(c) Rapport de performance de la prédiction (RPD)

Le rapport de performance à l'écart-type RPD (*Ratio of Performance to Deviation*) est une métrique fréquemment utilisée pour évaluer la qualité des prédictions d'un modèle de régression, en particulier dans le domaine de la spectroscopie. Il est défini comme le rapport entre l'écart-type des valeurs observées sur l'ensemble de test et la RMSEP :

$$\text{RPD} = \frac{\sigma_y}{\text{RMSEP}}$$

où σ_y est l'écart-type des valeurs observées (Y_{test}) et RMSEP est l'erreur quadratique moyenne de prédiction. Une valeur élevée de RPD indique une meilleure capacité prédictive du modèle. En général, un RPD supérieur à 2 est considéré comme satisfaisant, et au-delà de 3 comme excellent.

3.3.3 Implémentation et optimisation des modèles

(a) Modèle de Référence : kNN-LWPLSR

Ce modèle sert de baseline chimiométrique avancée, retenu pour sa capacité démontrée à modéliser des relations non-linéaires locales dans les données spectrales. L'optimisation par validation croisée a porté sur les quatre hyperparamètres clés du modèle (voir section 2.3), dont les espaces de recherche sont définis dans le Tableau 3.5.

TABLE 3.5 – Espace de recherche des hyperparamètres pour le modèle kNN-LWPLSR.

Hyperparamètre	**Valeurs Testées**
nlv_{dis} (scores PLS globaux)	5, 15, 25
k (nombre de voisins)	150, 300, 500, 600, 750, 1000
h (facteur de forme)	1, 2, 3, 5
nlv (LVs locales)	Entiers de 0 à 20

À l'issue de cette procédure d'optimisation, une combinaison d'hyperparamètres optimale a été sélectionnée pour chaque variable cible en minimisant la RMSECV. Le Tableau 3.6 présente les configurations finales retenues pour l'entraînement des modèles de

prédiction. On remarque une tendance à la sélection de valeurs élevées pour le nombre de voisins (k), suggérant que les modèles bénéficient d'un voisinage local large pour établir leurs prédictions.

TABLE 3.6 – Hyperparamètres optimaux du modèle kNN-LWPLSR pour chaque variable cible.

Variable Cible	nlv	nlv_{dis}	h	k
adf	16	15	1.0	1000
adl	18	25	1.0	1000
cf	14	15	1.0	600
cp	18	25	1.0	1000
dmdcell	14	25	1.0	1000
ndf	20	15	1.0	750

(b) Modèle 1D-CNN

L'objectif du modèle 1D-CNN est d'évaluer une approche de *deep learning* supervisée directe, basée sur l'architecture 1D-CNN simple mais efficace explorée par (PASSOS et MISHRA, 2023). Afin de sonder l'importance de l'architecture de la couche de convolution, deux variantes ont été systématiquement testées :

— **CNN-R_v1D** : Utilise une couche de convolution classique avec un *padding* de type 'same' et une largeur de filtre (*kernel size*) variable. Cette variante est suivie d'une unique couche dense.
— **CNN-R_v1E** : Utilise un *padding* de type 'valid' et un filtre dont la largeur est fixe et égale à la taille totale du spectre (700). Cette approche transforme la convolution en une projection unique du spectre. L'architecture permet l'ajout de plusieurs couches denses successives.

Étant donné la tendance des réseaux de neurones au surapprentissage, une stratégie d'arrêt précoce (*early stopping*) a été intégrée au sein de chaque pli de la validation croisée. Pour ce faire, 10% des données de chaque pli d'entraînement ont été isolées pour servir de jeu de validation interne, permettant de surveiller la performance du modèle et d'arrêter l'entraînement lorsque l'erreur sur ce jeu ne diminue pas après une patience de 15 époques.

Pour ces deux variantes, une optimisation des hyperparamètres a été menée via Optuna. Le Tableau 3.7 détaille l'espace de recherche complet qui a été exploré.

Les Tableaux 3.8 et 3.9 présentent les combinaisons d'hyperparamètres optimales identifiées pour les variantes v1D et v1E respectivement.

TABLE 3.7 – Espace de recherche des hyperparamètres pour les variantes du modèle 1D-CNN.

Hyperparamètre	**Variante CNN-R_v1D**	**Variante CNN-R_v1E**
Largeur du filtre	Entier [5, 200], pas de 20	700 (fixe)
Nombre de filtres	Entier [1, 32]	Entier [1, 32]
Nombre de couches denses	1 (fixe)	Entier [1, 3]
Unités par couche dense	Entier [8, 128], pas de 4	Entier [8, 128], pas de 4 (par couche)
Taux de *dropout*	Non applicable	Flottant [0.0, 0.6], pas de 0.005
Taux d'apprentissage	Log [1e-4, 0.03]	Log [1e-4, 0.03]
Régularisation L2	Flottant [0.0, 0.1], pas de 5e-4	Flottant [0.0, 0.1], pas de 5e-4
Taille du lot (*batch size*)	Entier [32, 256], pas de 16	Entier [32, 256], pas de 16

TABLE 3.8 – Hyperparamètres optimaux pour le modèle **CNN-R_v1D** pour chaque variable.

Variable	**Largeur Filtre**	**Nb. Filtres**	**Unités Denses**	**Taux d'Apprent.**	**Rég. L2**	**Taille du Lot**
adf	5	20	96	7.62e-4	0.0	64
adl	5	15	120	2.47e-4	0.0385	32
cf	165	25	116	4.03e-4	0.0290	32
cp	45	23	128	1.38e-4	0.0405	32
dmdcell	25	28	128	6.29e-4	0.0370	32
ndf	5	9	64	1.24e-4	0.0425	32

TABLE 3.9 – Hyperparamètres optimaux pour le modèle **CNN-R_v1E** pour chaque variable.

Variable	**Nb. Filtres**	**Nb. Couches Denses**	**Unités Couches Denses**	**Taux Dropout**	**Taux d'Apprent.**	**Rég. L2**	**Taille du Lot**
adf	27	3	60, 64, 80	0.025	4.65e-4	0.0170	32
adl	22	3	108, 120, 128	0.125	3.19e-4	0.0610	32
cf	22	1	120	0.045	2.58e-3	0.0145	32
cp	29	1	100	0.350	5.32e-3	0.0	48
dmdcell	30	3	56, 80, 108	0.220	1.58e-3	0.0595	32
ndf	30	1	124	0.050	2.17e-2	0.0325	32

(c) Modèle IPA

Ce modèle teste une architecture profonde plus complexe, inspirée d'Inception (HAFFNER et al., 2025), conçue pour extraire des caractéristiques spectrales à de multiples échelles simultanément.

En nous alignant sur la démarche de l'étude originale, nous avons fait le choix de conserver l'architecture du modèle fixe et de concentrer l'optimisation des hyperparamètres sur les deux facteurs les plus influents pour la convergence et la régularisation :

— **Le taux d'apprentissage** (`lr`) de l'optimiseur Adam.
— **Le coefficient de régularisation L2** (`l2`), qui pondère la pénalité sur les poids du modèle.

Ces deux paramètres ont été optimisés via Optuna sur une échelle logarithmique, dans un intervalle de $[10^{-5}, 10^{-2}]$. La même stratégie d'arrêt précoce (*early stopping*) que pour les modèles CNN a été employée. Le Tableau 3.10 présente les valeurs optimales retenues pour chaque variable cible à l'issue de cette recherche.

TABLE 3.10 – Hyperparamètres optimaux du modèle IPA pour chaque variable cible.

Variable Cible	Taux d'Apprentissage (`lr`)	Régularisation L2 (`l2`)
adf	9.24×10^{-4}	1.66×10^{-4}
adl	5.48×10^{-4}	3.55×10^{-5}
cf	1.04×10^{-3}	2.48×10^{-3}
cp	3.18×10^{-4}	2.28×10^{-4}
dmdcell	1.65×10^{-3}	1.60×10^{-3}
ndf	5.10×10^{-3}	3.25×10^{-4}

(d) Modèle 1D-CAE (Autoencoder Convolutionnel)

Cette approche vise à évaluer l'efficacité d'une extraction de caractéristiques non-supervisée. Pour ce modèle, nous avons mis en place une stratégie d'optimisation de bout en bout (*end-to-end*). Cela signifie que pour chaque essai de la recherche d'hyperparamètres, le processus complet — incluant l'entraînement de l'autoencodeur et la régression qui s'ensuit — est exécuté, et la performance de la régression finale est directement utilisée comme la métrique à optimiser.

Le processus d'optimisation pour chaque pli de la validation croisée se déroule donc comme suit :

1. **Entraînement de l'Autoencodeur :** Pour une combinaison d'hyperparamètres donnée, telle que définie dans le Tableau 3.11, un modèle 1D-CAE est entraîné sur les données spectrales pour apprendre à les reconstruire.
2. **Extraction de Caractéristiques :** L'encodeur du modèle fraîchement entraîné est utilisé pour transformer les spectres en une représentation dans l'espace latent.
3. **Régression Supervisée :** Une Régression Linéaire Multiple (MLR) est entraînée sur ces nouvelles caractéristiques latentes pour prédire la variable cible.

4. **Évaluation pour Optimisation :** L'erreur quadratique moyenne (RMSE) de cette régression est calculée. C'est cette valeur de RMSE finale qui est retournée à l'optimiseur Optuna, qui cherche à la minimiser sur l'ensemble des essais.

Cette approche intégrée garantit que les hyperparamètres de l'autoencodeur, sont optimisés non pas pour la qualité de la reconstruction, mais bien pour leur utilité directe dans la tâche de régression finale. Le Tableau 3.11 résume l'espace de recherche exploré.

TABLE 3.11 – Espace de recherche des hyperparamètres pour l'approche 1D-CAE + MLR.

Hyperparamètre	**Valeurs ou Intervalle Exploré**
Dimension de l'espace latent	32, 64, 128
Taux d'apprentissage (pour l'AE)	Échelle Logarithmique $[10^{-4}, 10^{-2}]$
Taille du lot (*batch size*)	16, 32

À l'issue de cette recherche, les meilleures configurations ont été identifiées pour chaque variable. Le Tableau 3.12 présente les hyperparamètres finaux retenus. On note une préférence pour une dimension latente élevée (128) pour la majorité des variables, suggérant qu'une représentation riche en information est bénéfique pour la régression linéaire subséquente.

TABLE 3.12 – Hyperparamètres optimaux pour l'approche 1D-CAE + MLR pour chaque variable cible.

Variable Cible	**Dimension Latente**	**Taux d'Apprentissage**	**Taille du Lot**
adf	128	6.80×10^{-3}	64
adl	64	8.30×10^{-3}	64
cf	128	5.08×10^{-4}	32
cp	128	3.60×10^{-3}	64
dmdcell	128	6.30×10^{-4}	64
ndf	64	2.30×10^{-3}	64

3.3.4 Environnement de travail

(a) Modèles d'apprentissage profond (Python)

L'ensemble des expérimentations relatives aux modèles d'apprentissage profond a été conduit dans un environnement basé sur des **Jupyter Notebooks** et le langage de programmation **Python (version 3.9.0)**. L'implémentation et l'optimisation de ces modèles ont reposé sur plusieurs bibliothèques :

— **TensorFlow (version 2.10.0)** (MARTÍN ABADI et al., 2015) : Un framework pour le développement et l'entraînement des réseaux de neurones.
— **Scikit-Learn (version 1.6.1)** (PEDREGOSA et al., 2011) : Une bibliothèque de référence pour les métriques d'évaluation et les tâches de pré-traitement.
— **Optuna (version 4.2.1)** (AKIBA et al., 2019) : Un framework pour l'optimisation systématique des hyperparamètres.

(b) Modèles chimiométriques (Julia)

Pour les modèles chimiométriques de référence (PLS et kNN-LWPLSR), ont été développés en utilisant le langage de programmation **Julia (version 1.11.4)**, reconnu pour ses hautes performances dans le calcul scientifique.
Pour l'implémentation de ces approches, la bibliothèque **Jchemo (0.8.6)** (LESNOFF, 2021) a été spécifiquement utilisée. Il s'agit d'une boîte à outils performante dédiée à l'analyse de données chimiométriques dans l'écosystème Julia.

(c) Matériel de calcul

Les calculs et l'entraînement de l'ensemble des modèles ont été effectués sur un ordinateur portable (Laptop) équipé d'un processeur **Intel Core i7-9750H**, de **16 Go de RAM** et d'une carte graphique **NVIDIA GeForce RTX 2060** exploitant l'architecture **CUDA (version 11.2)** pour accélérer les opérations de calcul.

L'intégralité du code développé pour ce travail est disponible publiquement sur le dépôt GitHub (GitHub_PFE).

Chapitre 4

Résultats

Ce chapitre présente les résultats finaux de notre étude comparative. Nous commencerons par présenter les performances des différents modèles optimisés afin d'identifier les approches les plus efficaces pour notre problématique (section 4.1). Ensuite, nous procéderons à une analyse des architectures et des stratégies pour comprendre les raisons de leurs succès ou de leurs échecs relatifs (section 4.2). Enfin, nous conclurons en validant la robustesse de notre démarche d'optimisation des hyperparamètres (section 4.3).

4.1 Performance prédictive finale des modèles

Pour obtenir une évaluation complète et nuancée des modèles, leur performance a été mesurée à l'aide de trois métriques complémentaires : l'erreur quadratique moyenne de prédiction (RMSEP) qui quantifie l'erreur absolue, le rapport de performance à l'écart-type (RPD) qui évalue l'utilité pratique en chimiométrie, et l'erreur relative (RE) qui permet de comparer les performances entre des variables d'échelles différentes. Les Figures 4.1, 4.3 et 4.2 présentent une vue consolidée des résultats pour ces trois indicateurs.

4.1.1 Analyse des résultats

(a) Erreur quadratique moyenne de prédiction RMSEP (figure 4.1)

Le modèle de régression par les moindres carrés partiels (PLSR), qui sert de référence linéaire dans cette étude, affiche systématiquement le RMSEP le plus élevé pour l'ensemble des six variables cibles. Cette performance suggère la présence de relations non-linéaires significatives entre les données spectrales et les propriétés chimiques des échantillons. Un modèle purement linéaire s'avère donc insuffisant pour capturer toute la complexité des informations contenues dans les spectres.

Le modèle kNN-LWPLSR, une approche chimiométrique reconnue pour sa capacité à gérer les non-linéarités et la nature locale des relations, se positionne comme un compétiteur extrêmement robuste. Il obtient le RMSEP le plus faible, et donc la meilleure performance, pour quatre des six variables : *adf*, *cf*, *cp* et *ndf*. Cela démontre qu'une modélisation locale et non-linéaire est particulièrement bien adaptée pour la majorité des variables étudiées. Le succès de ce modèle suggère que les relations pour ces variables sont mieux décrites au

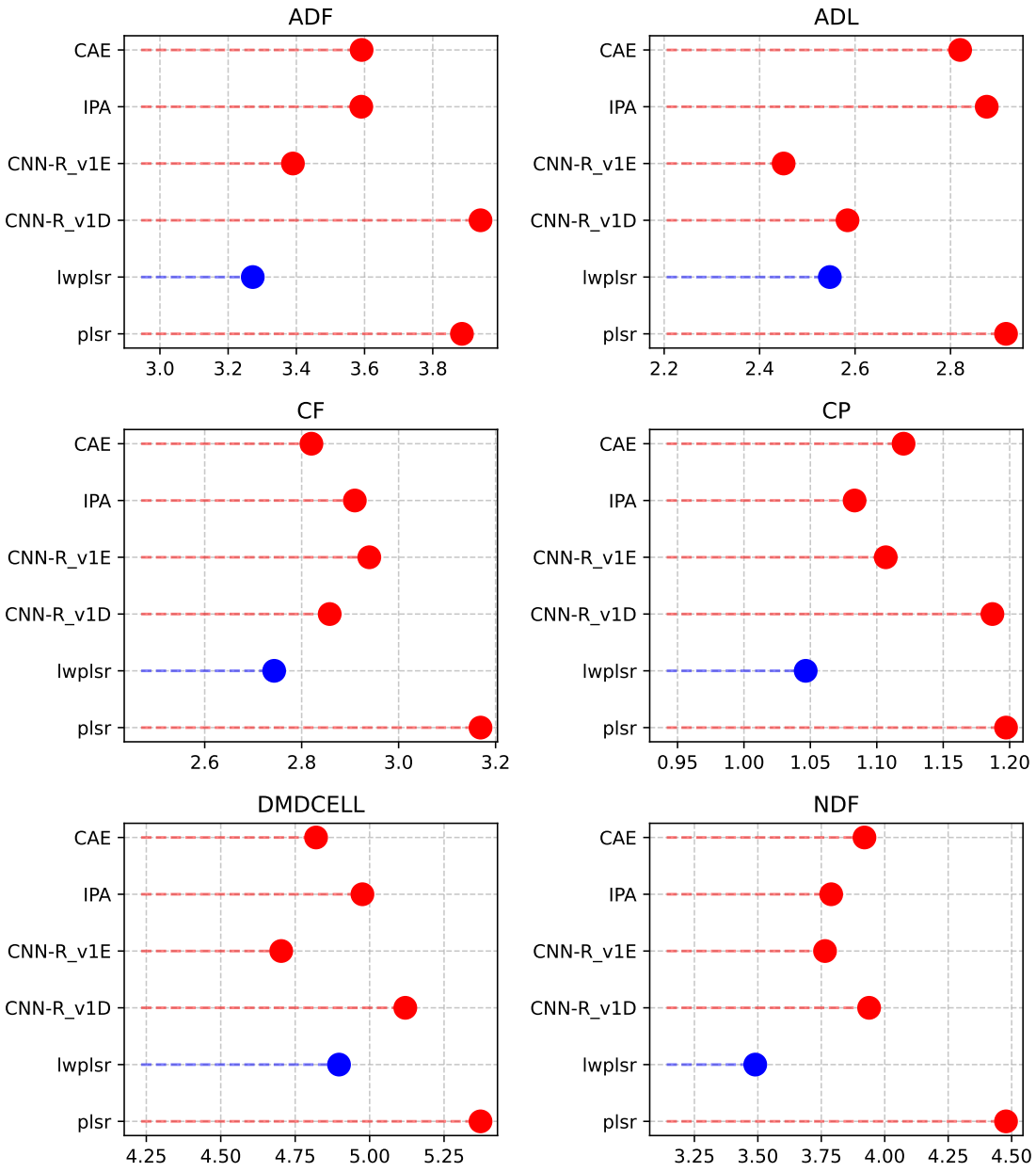


FIGURE 4.1 – Comparaison basée sur l'erreur quadratique moyenne de prédiction (RMSEP). Une valeur plus faible est meilleure.

sein de sous-ensembles locaux de données, ce qui pourrait être dû à l'hétérogénéité des échantillons, qui sont constitués de différentes espèces ou types de plantes.

Les modèles de DL présentent un tableau de performances plus hétérogène, mais tout aussi instructif. Ils surpassent systématiquement et largement le modèle linéaire PLS, confirmant leur capacité à modéliser des relations non linéaires. Leur positionnement par rapport au kNN-LWPLSR est cependant dépendant de la variable cible.

Pour les variables *adl* et *dmdcell*, certains modéles surpassent le kNN-LWPLSR. Le modèle CNN-R_v1E, en particulier, se classe premier pour ces deux cas, suivi de près par d'autres modèles de DL. Cette observation suggère que pour des relations potentiellement plus globales ou hiérarchiques, une extraction de caractéristiques réalisée par un réseau

de neurones convolutif peut être plus performante qu'une approche locale, même si cette dernière est très sophistiquée.

Il est toutefois important de noter qu'il n'existe pas une unique architecture DL qui domine universellement. La performance dépend fortement de l'adéquation entre le modèle et la nature de la variable, ce qui met en lumière l'importance du processus de sélection d'architecture et d'optimisation des hyperparamètres (Section 3.2).

(b) rapport de performance à l'écart-type RPD (figure 4.2)

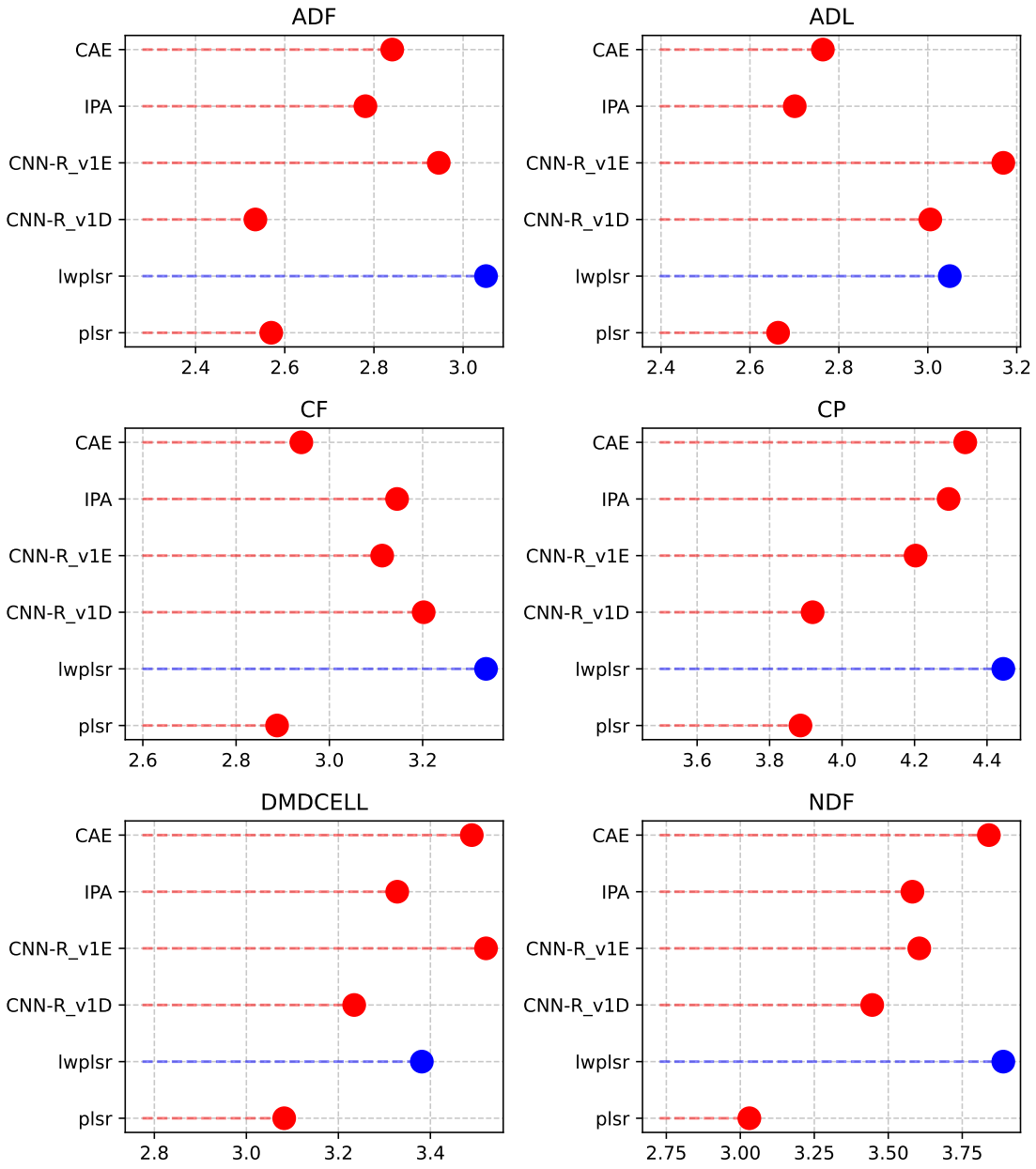


FIGURE 4.2 – Comparaison basée sur le rapport de performance à l'écart-type (RPD). Une valeur plus élevée est meilleure.

En complément de l'analyse du RMSEP, le rapport de performance à l'écart-type (RPD) offre une évaluation standardisée de la qualité et de la robustesse des modèles de

prédiction. Puisque le RPD est inversement proportionnel au RMSEP pour une variable donnée, le classement des modèles reste le même. L'intérêt principal de cette métrique réside dans l'application de seuils qualitatifs reconnus, où un RPD supérieur à 2 est jugé satisfaisant, et un RPD supérieur à 3 indique un modèle d'excellente qualité.

Le graphique de RPD montre que : pour chaque variable cible, le modèle le plus performant atteint un RPD supérieur à 3, se qualifiant ainsi comme excellent. C'est le cas du modèle kNN-LWPLSR pour les variables *adf*, *cf*, *cp* et *ndf*, avec une performance particulièrement exceptionnelle pour la variable *cp* (RPD > 4.4). De même, le modèle CNN-R_v1E se distingue pour les variables *adl* et *dmdcell*, pour lesquelles il est le plus performant.

En plus, plusieurs modèles d'apprentissage profond, bien que n'étant pas classés premiers, dépassent également le seuil de RPD de 3, confirmant leur viabilité pour des applications quantitatives. À l'opposé, le modèle PLS, avec des valeurs de RPD se situant majoritairement entre 2.5 et 3.0, se classe comme simplement satisfaisant.

(c) Erreur relative RE (figure 4.3)

L'analyse de l'erreur relative (RE), qui normalise l'erreur de prédiction par la moyenne de la variable observée, offre une perspective complémentaire. La présentation des résultats sur un axe unifié (figure 4.3) met en évidence un résultat particulièrement intéressant : l'erreur relative pour la variable *adl* est substantiellement plus élevée que pour toutes les autres variables.

Cette observation ne signifie pas nécessairement que les modèles sont intrinsèquement moins performants pour cette variable. L'explication réside dans la définition même de l'erreur relative ($RE = RMSEP/\bar{y}$). Une valeur de RE élevée peut résulter soit d'un RMSEP élevé, soit d'une valeur moyenne de la variable ($\bar{y}$) faible.

En nous référant aux statistiques descriptives du jeu de données (Tableau 3.4), la variable *adl* (Lignine au Détergent Acide) possède la plus faible valeur moyenne de toutes les variables cibles (environ 11 %MS). Par conséquent, même une erreur de prédiction absolue (RMSEP) modérée est amplifiée lorsqu'elle est divisée par cette moyenne, ce qui conduit à une erreur relative élevée.

Concernant les autres variables, les erreurs relatives sont nettement plus faibles et relativement homogènes, se situant majoritairement entre 6 % à 12 %. La variable **ndf** présente systématiquement l'erreur relative la plus faible pour l'ensemble des modèles. Ce performance en termes relatifs peut être aussi due à la valeur moyenne élevée de la ndf (parceque c'est la fraction total des fibes, hemicellulose, cellulose et lignin).

Cette analyse souligne l'importance d'interpréter plusieurs métriques de performance. La combinaison de ces trois métriques fournit une évaluation complète et équilibrée des modèles développés.

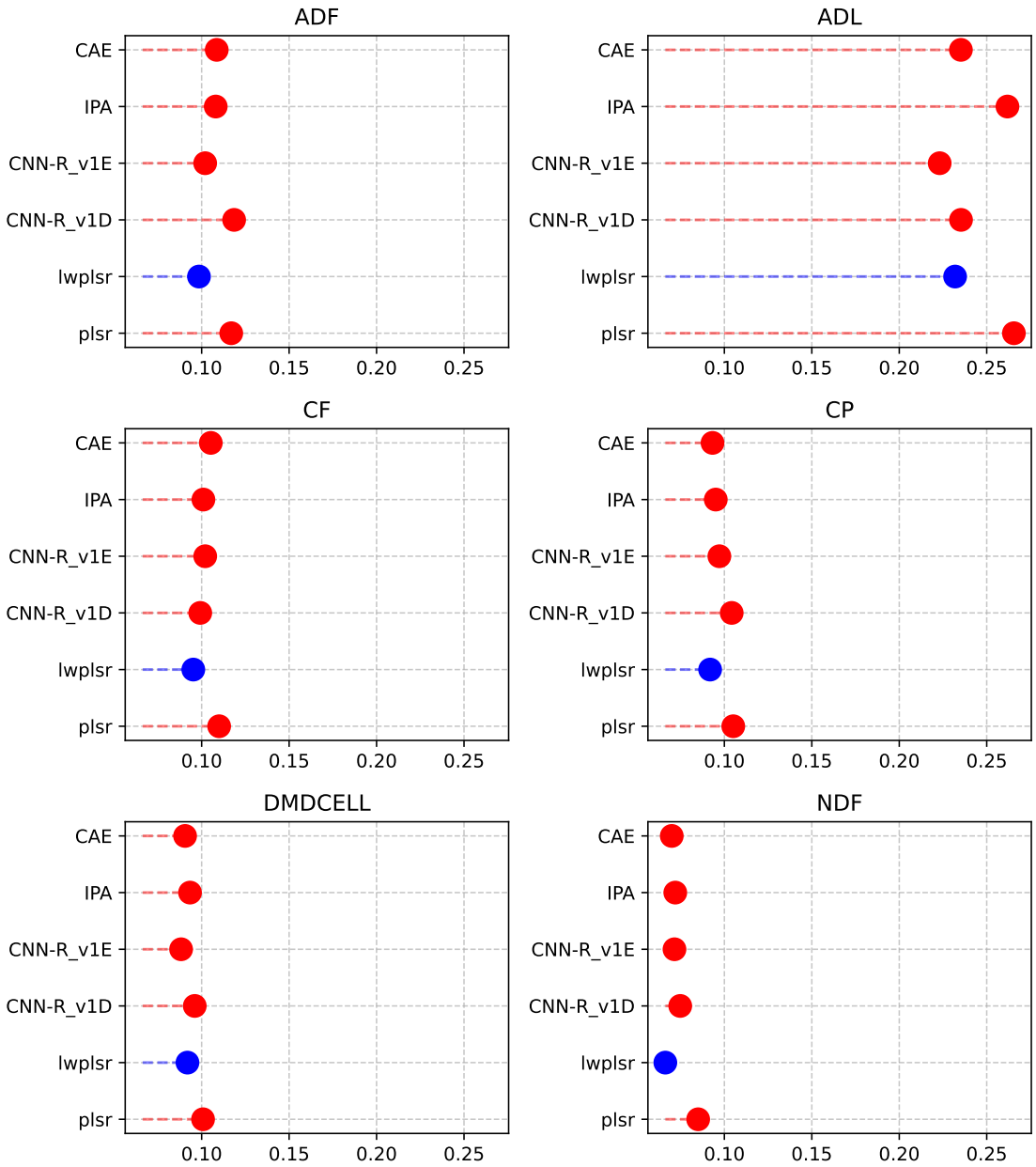


FIGURE 4.3 – Comparaison basée sur l'erreur relative (RE). Une valeur plus faible est meilleure. Les axes des abscisses ont été unifiés Afin de comparer les performances entre les modèles.

4.2 Discussion des performances des modèles

Au-delà du classement des performances, il est essentiel de comprendre les facteurs qui les déterminent. Cette section discute les raisons probables du succès ou de l'échec des différentes stratégies de modélisation.

Le modèle kNN-LWPLSR s'est imposé comme l'approche la plus performante pour la majorité des variables cibles, surpassant souvent les architectures d'apprentissage profond plus complexes. Son succès peut être directement attribué à sa nature de "modélisation locale", particulièrement bien adaptée aux caractéristiques de ce jeu de données.

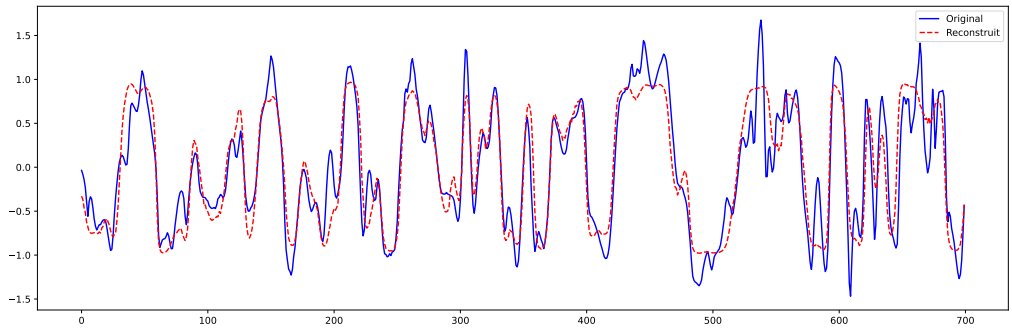
Les données spectrales proviennent d'échantillons de fourrages hétérogènes, incluant diverses espèces (graminées, légumineuses, arbustes) et origines géographiques (régions méditerranéennes, sahéliennes, etc.). Cette diversité induit très probablement des relations non-linéaires complexes et variables au sein de l'espace des données. Un modèle global, comme un réseau de neurones convolutif (CNN), tente d'apprendre une fonction unique pour l'ensemble du jeu de données. Il peut peiner à capturer des relations qui changent subtilement d'une région à l'autre de l'espace spectral.

À l'inverse, le kNN-LWPLSR ne construit pas un seul modèle global. Pour chaque nouvelle prédiction, il identifie un sous-ensemble d'échantillons similaires (les k plus proches voisins) et construit un modèle PLS simple, pondéré et spécifiquement adapté à ce voisinage local. Cette flexibilité lui permet de s'adapter aux particularités locales des données, offrant une robustesse supérieure lorsque le jeu de données est hétérogène.

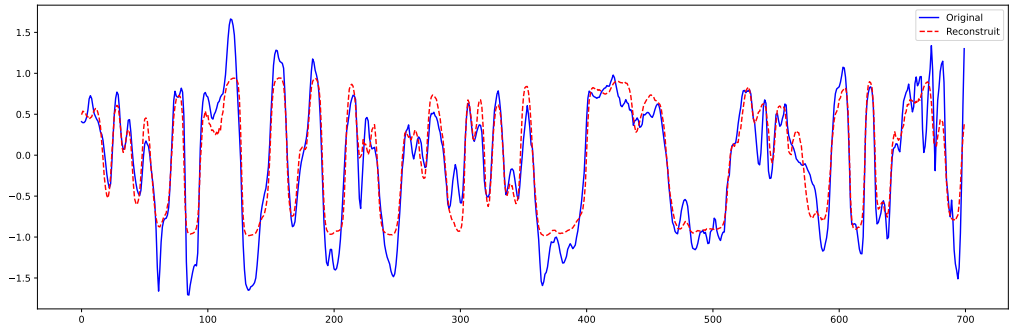
L'analyse des hyperparamètres optimaux (Tableau 3.5) renforce cette hypothèse. On y observe une tendance marquée à la sélection d'un grand nombre de voisins (k), allant jusqu'à 1000 pour plusieurs variables. Cela suggère que, bien que la modélisation soit "locale", elle bénéficie de "larges régions" de voisinage pour établir des prédictions stables.

Concernant les autres approches, l'architecture profonde et multi-branches du modèle IPA, bien que puissante, est peut-être surdimensionnée pour la taille et la nature de ce jeu de données, la rendant plus difficile à optimiser. Une recherche d'hyperparamètres plus exhaustive serait probablement nécessaire.

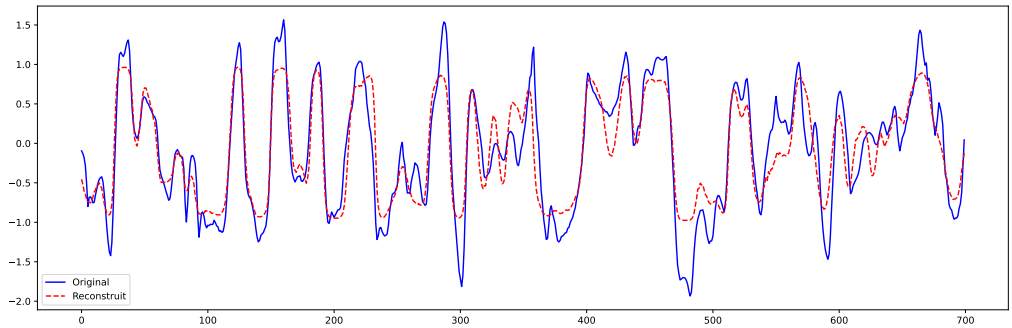
Enfin, pour l'autoencodeur convolutif (1D-CAE), la stratégie consistant à appliquer une simple régression linéaire multiple (MLR) sur les caractéristiques extraites est probablement le principal facteur limitant. L'encodeur, par sa nature non-linéaire, transforme les spectres en un espace latent qui n'a aucune raison de présenter une relation purement linéaire avec les variables chimiques. Cette hypothèse est d'ailleurs renforcée par la qualité de la reconstruction spectrale. Comme l'illustre la Figure 4.4, le signal reconstruit par le décodeur suit le spectre original, ce qui prouve que l'encodage vers l'espace latent a été réalisé avec succès. Le point faible de la méthode réside donc très probablement dans l'étape de régression, qui est incapable d'exploiter la richesse non-linéaire de ces caractéristiques.



(a) Échantillon 220.



(b) Échantillon 249.



(c) Échantillon 137.

FIGURE 4.4 – Comparaison du spectre original (bleu) et reconstruit (tiret rouge) pour 3 échantillons de l'ensemble du test.

4.3 Validation de la stratégie d'optimisation (HPO)

L'optimisation des hyperparamètres est une étape fondamentale mais coûteuse en ressources pour les modèles d'apprentissage profond. Il est fréquent que la méthodologie exacte de cette optimisation ne soit pas explicitement détaillée dans la littérature. Cependant, l'article de (PASSOS et MISHRA, 2023), qui a introduit les architectures 1D-CNN que nous explorons, où un budget allant jusqu'à 1000 essais est mentionné. Mais pour ce travail et a contrainte du temps, notre budget initial a été fixé à 100 essais par modèle.

En effet, des travaux comme ceux de (WEERTS et al., 2020) ont mis en évidence le "risque de tuning" : une recherche avec un budget limité peut paradoxalement conduire à des performances inférieures à celles obtenues avec des paramètres par défaut bien choisis. Il était donc impératif de vérifier l'efficacité de notre processus d'optimisation.

La Figure 4.5 compare les performances (RMSEP) des modèles d'apprentissage profond avec les valeurs Défaut (meilleurs valeurs choisis des articles respectives) et après une optimisation avec un budget de 100 essais.

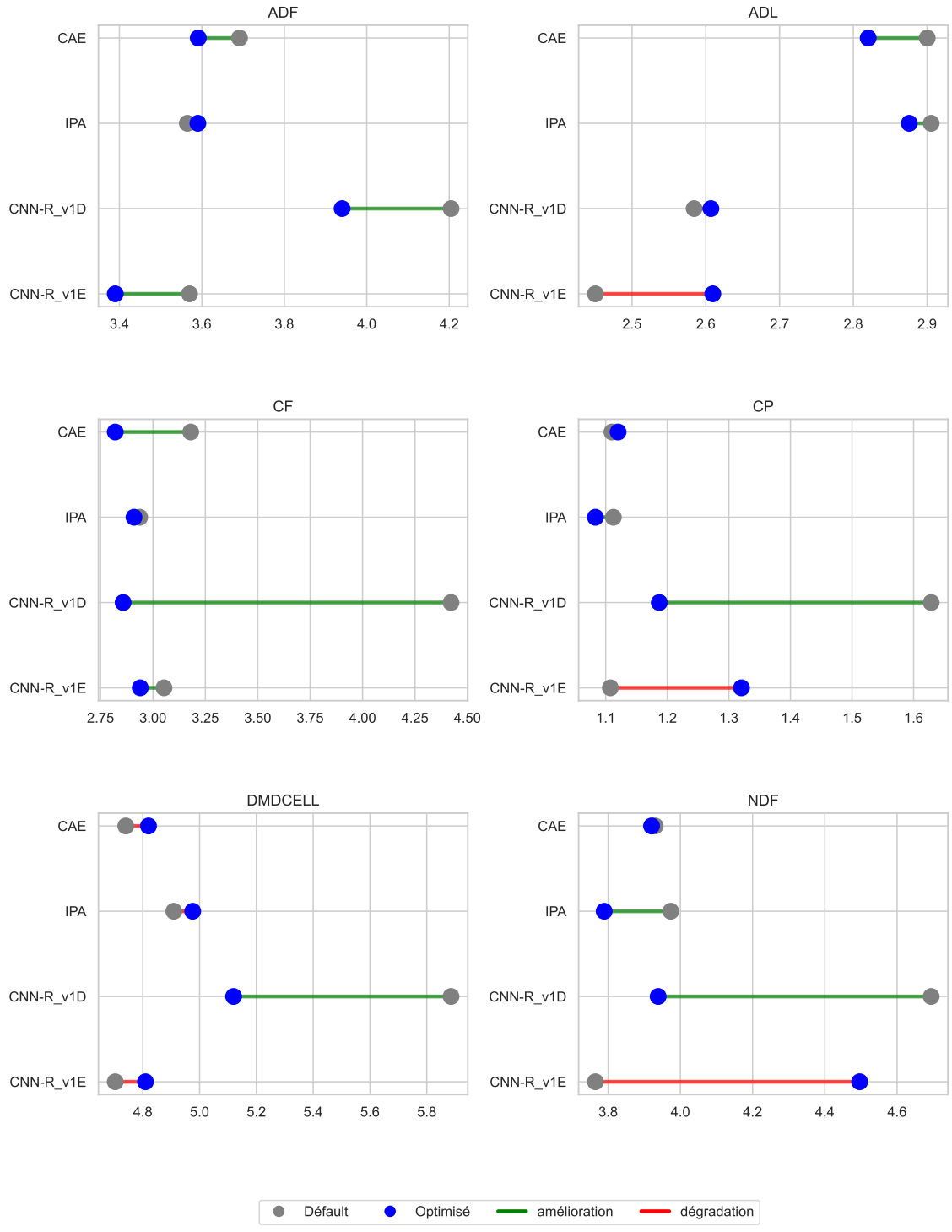


FIGURE 4.5 – Comparaison de la performance (RMSEP) des modèles avant (Défaut, point gris) et après une optimisation de 100 essais (Optimisé, point bleu). Une ligne verte indique une amélioration, et une ligne rouge indique une dégradation.

Comme l'illustrent les lignes vertes sur la Figure 4.5, le processus d'optimisation a été globalement bénéfique ou neutre pour la majorité des modèles.

Une exception est observée pour le modèle **CNN-R_v1E** sur les variables *adl*, *cp* et *ndf*, où on constate claire dégradation. L'hypothèse est que cette contre-performance est directement liée à la taille et complexité de l'espace de recherche pour ce modèle.

En effet, comme défini dans le Tableau 3.7 dans la section 3.3, l'architecture du CNN-R_v1E permet un nombre variable de couches denses, et l'utilisation ou non de *Dropout* sont des paramètres à optimiser, ce qui rend son espace d'hyperparamètres à explorer bien plus vaste que celui des modèles IPA ou 1D-CAE et même CNN-R_v1D (tous ces modèles ayant un nombre de couches denses fixe). Un budget de 100 essais était donc potentiellement insuffisant pour naviguer efficacement dans cet espace complexe.

Pour vérifier notre hypothèse, une seconde phase d'optimisation a été menée spécifiquement pour le modèle **CNN-R_v1E**. Grâce à des ressources de calcul supplémentaires, le budget d'essais a été porté de 100 à 300. La Figure 4.6 présente les résultats de cette expérience.

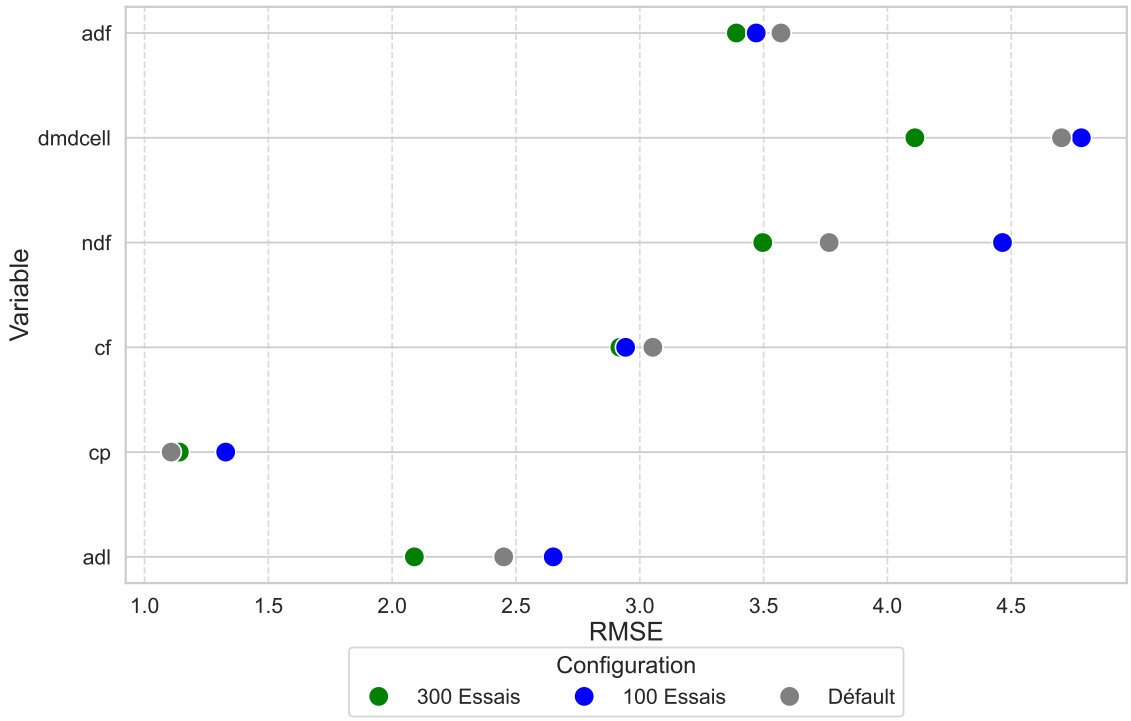


FIGURE 4.6 – Comparaison des performances (RMSEP) pour le modèle CNN-R_v1E. Les performances sont montrées pour les hyperparamètres par défaut (en gris), après 100 essais d'optimisation (en bleu), et après 300 essais (en vert).

L'analyse de la Figure 4.6 est sans équivoque. Pour chaque variable cible, le budget étendu à 300 essais a non seulement inversé la dégradation observée après 100 essais, mais il a également permis d'obtenir une performance finale meilleure que celle des hyperparamètres par défaut.

Les performances prédictives finales rapportées dans la section 4.1 sont basées sur les résultats des modèles après optimisation des hyperparamètres. Étant donné que l'espace de recherche du modèle CNN-R_v1E est significativement plus vaste que celui des autres modèles, nous présentons les résultats issus de 300 essais pour ce dernier, contre 100 essais

pour les autres modèles, en raison de contraintes de temps.

4.4 Conclusion

En synthèse, les résultats de cette étude comparative ont mis en lumière l'absence d'un modèle universellement supérieur. L'approche de modélisation locale kNN-LWPLSR s'est avérée la plus performante pour la majorité des variables (*adf, cf, cp, ndf*), soulignant l'importance de la gestion de l'hétérogénéité des données. Parallèlement, le modèle d'apprentissage profond CNN-R_v1E a montré sa supériorité pour les variables *adl* et *dmdcell*, confirmant le potentiel de l'extraction de caractéristiques par convolution.

Il est à noter que tous les modèles non-linéaires ont systématiquement surpassé la référence linéaire PLS. Les modèles les plus performants pour chaque variable ont atteint un niveau de performance jugé excellent, avec des rapports de performance à l'écart-type (RPD) supérieurs à 3.

Chapitre 5

Conclusions et perspectives

5.1 Introduction

En synthèse, ce travail a démontré qu'il n'existe pas de modèle universellement supérieur pour la prédiction des propriétés chimiques des fourrages par spectroscopie PIR. La performance s'est avérée fortement dépendante de la variable cible, l'approche locale kNN-LWPLSR excellant pour la majorité des variables, tandis que les modèles convolutifs se sont distingués sur des variables spécifiques. Ce chapitre vise à prendre du recul sur notre démarche, et identifier les limitations de l'étude et proposer des pistes de recherche futures pour améliorer la robustesse, la performance et l'interprétabilité des modèles.

5.2 Pistes d'amélioration méthodologique

La performance de tout modèle prédictif est intrinsèquement liée aux choix méthodologiques effectués en amont. Cette section vise à explorer plusieurs améliorations qui pourraient être apportées à notre protocole expérimental, et définir définir une feuille de route pour de futurs travaux susceptibles d'accroître la performance des modèles d'apprentissage profond qui, dans notre cas, n'ont pas systématiquement démontré la supériorité souvent rapportée dans la littérature. Nous aborderons ces pistes en deux volets : d'abord les approches centrées sur les données (section 5.2.1), puis celles relatives à l'optimisation et à l'architecture des modèles (section 5.2.2).

5.2.1 Gestion et augmentation des données

Une première série d'améliorations concerne la manière dont les données sont structurées et utilisées pour l'entraînement.

(a) Vers une approche multivariée

Notre étude a adopté une approche de modélisation où six modèles indépendants ont été construits, un pour chaque variable chimique à prédire. Si cette méthode est directe et robuste, elle ignore les corrélations qui existent entre les différentes propriétés chimiques

(voir la figure 5.1). Par exemple, les fractions de fibres (ADF, NDF) et de lignine (ADL), qui sont bien évidement liées au sein de la matière végétale, présente une grande corrélation entre eux.

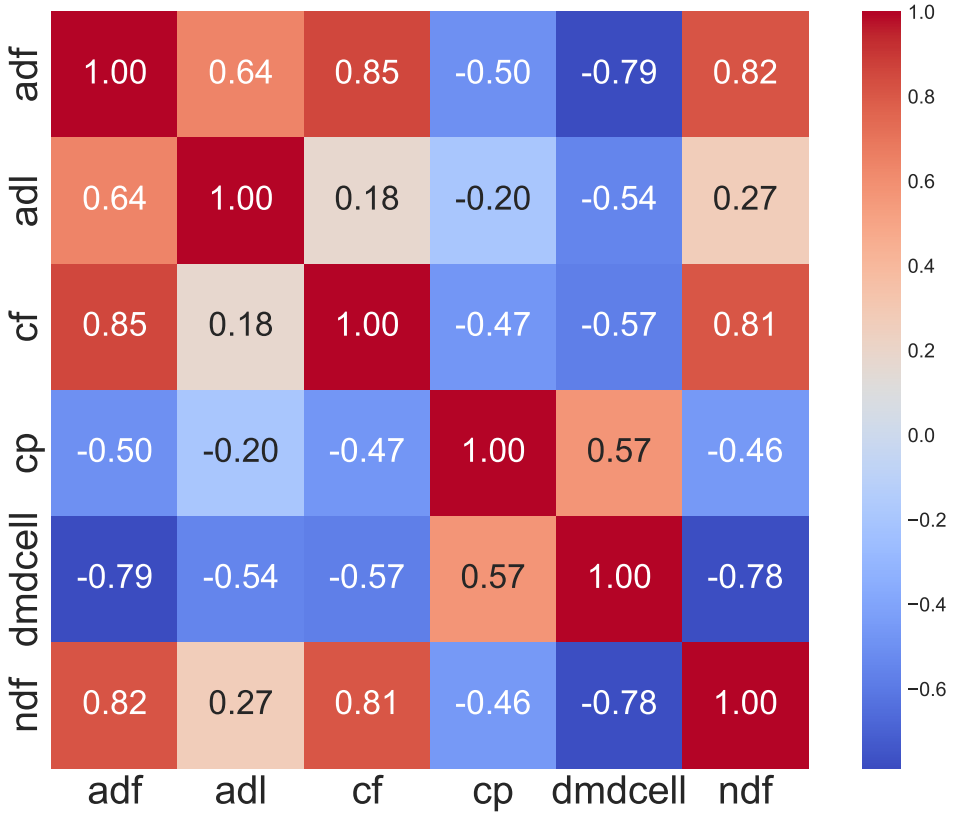


FIGURE 5.1 – Matrice de corrélation entre les variables chimiques. Plus la corrélation est proche de 1 ou de -1, plus la relation linéaire entre les deux variables est forte.

Une piste de recherche prometteuse serait donc de développer un modèle de régression unique, dit multivariée, ou multi-tâches (ou *multi-output*), capable de prédire l'ensemble des six variables simultanément. Un tel modèle pourrait apprendre une représentation, et capturer les relations sous-jacentes entre les variables cibles et potentiellement aboutir à des prédictions plus cohérentes et plus précises. en effet, des travaux comme ceux de (MARTINS et al., 2022) ont montré que les CNN obtiennent des résultats améliorés lorsqu'ils sont entraînés à prédire simultanément plusieurs sorties au lieu d'une seule.

La mise en œuvre d'une telle approche sur notre jeu de données actuel présente cependant un défi, dans la mesure où le nombre d'échantillons disponibles varie fortement d'une variable à l'autre — allant de 888 pour la variable cf à 1564 pour la variable cp — ce qui peut potentiellement introduire un biais dans le modèle.

(b) Augmentation des données pour les modèles profonds

Les modèles d'apprentissage profond sont réputés pour leur besoin important en données. La taille de notre jeu de données, bien que conséquente pour la chimiométrie classique, reste limitée pour ces architectures (environ 1100 échantillons en entraînement

par variable en moyenne). Il est donc probable que la performance des modèles CNN et CAE ait été contrainte par la quantité de données.

Pour pallier cette limitation, l'application de techniques d'augmentation de données est une voie incontournable. Plutôt que de simplement dupliquer les données, des altérations réalistes peuvent être appliquées aux spectres originaux pour en générer de nouveaux. Parmi les techniques pertinentes pour les données spectrales, on pourrait envisager :

— L'ajout d'un bruit gaussien contrôlé afin de simuler les variations instrumentales.
— L'application de légers décalages sur l'axe des longueurs d'onde (*spectral jitter*).
— L'utilisation de méthodes génératives plus avancées, telles que les réseaux antagonistes génératifs (GANs) (WU et al., 2021), pour créer des spectres synthétiques mais plausibles. Les GANs peuvent être utilisés également pour corriger le déséquilibre entre les classes (CHUNG et al., 2024).

L'objectif serait d'exposer les modèles à une plus grande diversité de signaux, ce qui améliore leur capacité de généralisation et leur robustesse au surajustement.

5.2.2 Optimisation et architecture des modèles

La seconde série d'améliorations concerne directement la conception et l'entraînement des modèles.

(a) Repenser l'usage des données pré-traitées

Dans un souci de cohérence, nous avons appliqué une chaîne de pré-traitement unique (SNV et dérivation Savitzky-Golay) à toutes les données en entrée des modèles. Or, cette décision entre en conflit avec la philosophie des architectures dites de bout en bout (end-to-end), comme le modèle IPA, qui sont spécifiquement conçues pour extraire l'information pertinente directement à partir des données brutes. Il est possible que le pré-traitement ait involontairement supprimé des informations subtiles ou des artefacts (comme des décalages de ligne de base) que le réseau de neurones aurait pu apprendre à ignorer ou même à utiliser. Une expérience future cruciale consisterait à entraîner les modèles d'apprentissage profond sur les données spectrales brutes afin de quantifier rigoureusement l'impact de cette étape de pré-traitement.

(b) Améliorer la chaîne de prédiction de l'autoencodeur

L'approche basée sur l'autoencodeur convolutif (1D-CAE) a démontré une excellente capacité à compresser et reconstruire les spectres, prouvant la création d'un espace latent informationnel de qualité. Cependant, sa performance prédictive finale s'est avérée décevante. Le facteur limitant est très probablement la seconde étape de la chaîne : l'application d'une simple régression linéaire multiple (MLR) sur les variables latentes.

Nos résultats ont montré que la relation entre spectres et propriétés chimiques est fondamentalement non-linéaire, ce qui explique la faible performance du modèle PLSR. Appliquer une MLR sur des caractéristiques extraites de manière non-linéaire par le CAE est donc une approche conceptuellement faible. Une amélioration évidente serait de remplacer la MLR par un **régresseur non-linéaire** plus puissant, tel qu'un modèle à base

d'arbres de décision (par exemple, XGBoost ou LightGBM) ou même un petit perceptron multicouche (MLP), qui serait plus à même de modéliser la complexité de l'espace latent.

(c) L'Agrégation des modèles et robustesse par l'optimisation

L'entraînement d'un réseau de neurones comporte une part d'aléa, et même après une optimisation des hyperparamètres, le "meilleur" modèle trouvé n'est pas garanti d'être le plus stable. Une stratégie classique pour améliorer la robustesse et souvent la performance est l'agrégation de modèles (*ensembling*). Comme suggéré dans le travail de (PASSOS et MISHRA, 2023), une approche simple consisterait à conserver non pas un, mais les 10 meilleurs modèles issus de la recherche d'hyperparamètres, et d'utiliser la moyenne de leurs prédictions comme prédiction finale.

Enfin, notre propre analyse a confirmé que l'effort d'optimisation est directement corrélé à la performance. Il est donc impératif que de futurs travaux allouent un "budget d'optimisation" plus conséquent à l'ensemble des espaces de recherches complexes pour s'assurer qu'elles sont évaluées à leur plein potentiel.

5.3 Interprétabilité

Au-delà de la seule performance prédictive, deux autres critères sont fondamentaux pour évaluer la pertinence d'un modèle dans un contexte scientifique et pratique : son interprétabilité, c'est-à-dire notre capacité à comprendre son fonctionnement, et son efficience, qui englobe les ressources nécessaires à son développement et à son déploiement.

5.3.1 Le défi de l'interprétabilité des modèles profonds

Un des reproches les plus courants faits aux modèles d'apprentissage profond est leur nature de boîte noire (black box). En raison de leurs millions de paramètres organisés en couches non-linéaires, il est extrêmement difficile de comprendre comment une décision est prise. Dans notre étude, lorsqu'un modèle CNN fournit une prédiction précise, nous ne pouvons pas savoir simplement quelles régions spectrales ont le plus influencé son résultat. Le modèle s'appuie-t-il sur des bandes d'absorption connues, ou a-t-il découvert des relations nouvelles et complexes que l'œil humain ne peut déceler ? Cette opacité constitue un frein majeur à la découverte scientifique et à l'adoption de ces modèles en toute confiance.

Pour répondre à ce défi, le domaine de l'intelligence artificielle explicable *(Explainable AI, ou XAI)* a développé des outils permettant de sonder ces boîtes noires. Des méthodes comme **LIME** *(Local Interpretable Model-agnostic Explanations)* ou **SHAP** *(SHapley Additive exPlanations)* sont particulièrement intéressantes . Elles permettent d'estimer, pour une prédiction donnée, l'importance de chaque variable d'entrée (ici, chaque longueur d'onde) (PASSOS, 2025).

L'application de ces techniques dans le cadre de nos travaux aurait permis de générer des « cartes d'importance » spectrales pour chaque échantillon, qui vont mettre en évidence les longueurs d'onde jugées cruciales par les modèles profonds. Une telle analyse permettrait de valider si les modèles apprennent des relations chimiquement pertinentes et d'accroître

ainsi la confiance en leurs prédictions, transformant un simple outil de prédiction en un potentiel instrument de découverte.

5.3.2 Coût-bénéfice et pertinence pratique

L'évaluation d'un modèle ne peut se limiter à sa seule erreur de prédiction. Il est impératif de considérer l'efficience globale de la démarche, qui inclut le "coût computationnel", le temps de développement et l'effort humain requis pour l'optimisation des hyperparamètres. Dans cette optique, l'investissement en coût et en complexité d'un réseau de neurones pourrait se justifier pleinement s'il parvenait à offrir une performance de pointe directement à partir des données spectrales brutes. En opérant de manière véritablement de bout en bout, un tel modèle élimine la nécessité de développer et valider une chaîne de pré-traitement, une étape qui requiert une expertise significative et peut introduire des biais.

Cependant, lorsque ce n'est pas le cas, cela soulève une question pragmatique essentielle : le gain de performance marginal justifie-t-il une augmentation drastique du coût et de la complexité ? Pour les variables *adl* et *dmdcell*, les modèles CNN ont offert les meilleures performances, mais avec une avance modérée sur le kNN-LWPLSR. Pour une application pratique, un modèle légèrement moins performant mais plus rapide, plus simple à déployer et surtout plus interprétable, pourrait donc s'avérer être un meilleur choix stratégique. Le « meilleur » modèle n'est donc pas systématiquement celui avec le RMSEP le plus bas, mais celui qui offre le meilleur compromis entre performance, coût et confiance pour un besoin spécifique.

5.4 Conclusion générale et perspectives

Cette étude a comparé une méthode chimiométrique locale établie, le kNN-LWPLSR, à un ensemble d'architectures d'apprentissage profond pour l'analyse quantitative par spectroscopie PIR. Les résultats démontrent la robustesse et la pertinence continue de l'approche locale, qui surpasse les modèles profonds sur la majorité des variables cibles. Néanmoins, les réseaux de neurones convolutifs ont montré une performance supérieure pour deux variables spécifiques, indiquant un potentiel certain.

Ce bilan établit que, pour cette problématique, les architectures d'apprentissage profond, dans leur état actuel, ne remplacent pas encore une méthode spécialisée et éprouvée. Pour que ces nouvelles approches deviennent une alternative véritablement compétitive, les recherches futures devront s'orienter vers plusieurs axes d'amélioration :

1. **Une gestion des données plus sophistiquée**, par l'apprentissage multivariée et l'augmentation de données pour pallier la taille limitée des jeux de données.
2. **Une utilisation plus fidèle de leur philosophie**, en testant des architectures de bout en bout sur des données brutes pour en justifier la complexité.
3. **Une évaluation qui intègre l'interprétabilité et l'efficience** comme critères fondamentaux, au même titre que la performance prédictive.

En l'état actuel, le kNN-LWPLSR représente donc un meilleur compromis entre performance et efficience pour ce type de données. L'adoption à plus grande échelle de l'apprentissage profond dans ce domaine dépendra de sa capacité à relever ces défis, notamment en offrant des solutions interprétables et un avantage clair qui justifie son coût.

Annexe A

Annexes : Notions de base sur l'apprentissage profond

Cette annexe présente des notions fondamentales sur les réseaux de neurones artificiels (ANN) dans la section A.1, les réseaux de neurones convolutifs (CNN) dans la section A.2, les autoencodeurs (AE) dans la section A.3, et d'autres concepts clés du deep learning, pour faciliter la compréhension des méthodes utilisées dans ce travail.

A.1 Réseaux de neurones artificiels (ANN)

Les Réseaux de Neurones Artificiels (ANN, pour *Artificial Neural Networks*) sont des modèles computationnels inspirés par la structure interconnectée des neurones biologiques ((RUMELHART et al., 1986)). Un ANN typique est composé de couches de neurones artificiels (Nœuds), où chaque nœud calcule une somme pondérée des entrées suivie d'une **fonction d'activation** non linéaire (comme illustré dans A.1). L'architecture standard comprend une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie.

Il existe plusieurs fonction d'activations, qui peuvent être utilisé selon le but

Fonction d'activation	**Formule de sortie**	**Bornes**
Sigmoïde	$\frac{1}{1+e^{-x}}$	(0, 1)
Softmax	$\frac{e^{x_i}}{\sum_j e^{x_j}}$	(0, 1)
ReLU	$\max(0, x)$	[0, ∞)
Leaky-ReLU	$\begin{cases} \alpha x & \text{si } x < 0 \\ x & \text{sinon} \end{cases}$	$(-\infty, \infty)$
ELU	$\begin{cases} \alpha(e^x - 1) & \text{si } x < 0 \\ x & \text{sinon} \end{cases}$	$(-\alpha, \infty)$

TABLE A.1 – Fonctions d'activation courantes et leurs propriétés

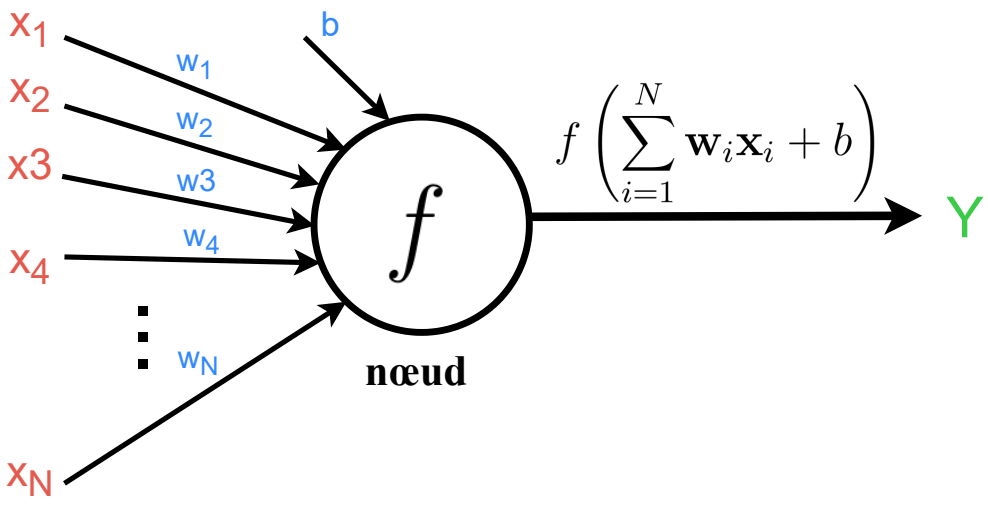


FIGURE A.1 – Structure simple d'un nœud. Les $\mathbf{x}_i$ sont les entrées externes ou les sorties d'autres nœuds. b est le biais, et les $\mathbf{w}_i$ sont les poids (b et $\mathbf{w}_i$ sont des paramètres entraînables). f est la fonction d'activation utilisée.

A.2 Réseaux de neurones convolutifs (CNN)

Les CNN sont particulièrement efficaces pour l'analyse de données structurées comme les images ou les séries temporelles, telles que les spectres. Leur architecture est spécifiquement conçue pour capturer les hiérarchies de caractéristiques locales. Contrairement aux réseaux de neurones denses où chaque neurone est connecté à toutes les entrées, les CNN utilisent des opérations de convolution pour extraire des motifs de manière plus efficace.

— **Couche de Convolution** : C'est le bloc de construction fondamental. Elle applique un ensemble de filtres (ou noyaux) sur les données d'entrée. Chaque filtre est une petite matrice de poids qui glisse sur l'ensemble des données pour détecter des caractéristiques spécifiques (par exemple, des pics, des pentes ou d'autres formes dans un spectre). Cette opération produit une *carte de caractéristiques* (feature map). L'avantage majeur est le **partage de poids** : le même filtre est utilisé sur toute la longueur du spectre, ce qui réduit considérablement le nombre de paramètres à entraîner et permet au modèle de détecter un motif, quelle que soit sa position.

— **Couche de Pooling (ou Sous-échantillonnage)** : Souvent placée après une couche de convolution, elle a pour but de réduire la dimension des cartes de caractéristiques. L'opération la plus courante, le *Max Pooling*, consiste à ne conserver que la valeur maximale d'une petite fenêtre de données. Cela rend la représentation plus compacte, moins coûteuse en calcul et plus robuste aux petites variations dans les données.

— **Couches Entièrement Connectées** : Après plusieurs couches de convolution et de pooling qui extraient des caractéristiques de plus en plus complexes, les données sont généralement aplaties en un vecteur unidimensionnel et traitées par une ou plusieurs couches denses (entièrement connectées) pour effectuer la régression ou la classification finale.

Dans le cadre de cette étude, des CNN unidimensionnels (1D-CNN) ont été utilisés

pour traiter directement les données spectrales, en apprenant à extraire les caractéristiques pertinentes pour prédire les propriétés chimiques.

A.3 Autoencodeurs

Les autoencodeurs (AE) sont des réseaux de neurones non supervisés destinés à apprendre des représentations compressées des données, souvent dans un but de réduction de dimensionnalité ou d'extraction de caractéristiques. Un autoencodeur est toujours composé de deux sous-réseaux :

— **L'Encodeur** : Cette partie du réseau prend les données d'entrée (par exemple, un spectre complet) et les compresse en une représentation de dimension inférieure, appelée *espace latent* ou *goulot d'étranglement* (bottleneck). Ce processus force le modèle à n'apprendre que les caractéristiques les plus essentielles des données.
— **Le Décodeur** : Cette partie prend la représentation compressée de l'espace latent et tente de *reconstruire* les données d'entrée originales avec le moins d'erreur possible.

L'autoencodeur est entraîné en minimisant une fonction de perte qui mesure la différence entre l'entrée originale et la sortie reconstruite (l'erreur de reconstruction). Une fois l'entraînement terminé, le décodeur est généralement abandonné. L'encodeur peut alors être utilisé comme un puissant outil d'extraction de caractéristiques : il transforme les données brutes de haute dimension en un vecteur de caractéristiques de faible dimension, qui peut ensuite être utilisé comme entrée pour un autre algorithme d'apprentissage supervisé, comme une régression linéaire multiple (MLR).

Dans ce travail, un **Autoencodeur Convolutif (CAE)** a été utilisé, où les couches denses de l'encodeur et du décodeur sont remplacées par des couches convolutives, ce qui est particulièrement adapté pour capturer les caractéristiques pertinentes des données spectrales.

A.4 Stratégies de gestion de surapprentissage

A.4.1 Régularisation L_2

La régularisation L_2 (aussi appelée *weight decay* ou régularisation de Tikhonov) consiste à ajouter au coût empirique initial $J_0(\theta)$ un terme de pénalité proportionnel au carré de la norme euclidienne des paramètres. Formellement, on définit

$$J(\theta) = J_0(\theta) + \frac{\lambda}{2}\|\theta\|_2^2,$$

où $\|\theta\|_2^2 = \sum_i \theta_i^2$ et $\lambda > 0$ est le coefficient de régularisation. Cette pénalité encourage les poids θ à rester petits, ce qui réduit la complexité effective du modèle et prévient le sur-apprentissage (GOODFELLOW et al., 2016); (MURPHY, 2012). Du point de vue bayésien, imposer un tel terme quadratique équivaut à placer une loi a priori gaussienne centrée sur zéro pour chaque poids (MURPHY, 2012). Dans la descente de gradient, ce terme se traduit par une mise à jour du type $\theta \leftarrow \theta - \eta\big(\nabla J_0(\theta) + \lambda\theta\big)$, introduisant

un amortissement (décroissance) linéaire des paramètres (GOODFELLOW et al., 2016). Comparée à la régularisation L_1 ou à des méthodes comme le *dropout*, la pénalité L_2 présente l'avantage de préserver la différentiabilité du critère de coût et conduit à des solutions analytiquement interprétables (par exemple la régression ridge linéaire) (BISHOP, 2006).

A.4.2 Arrêt précoce (early stopping)

La stratégie d'arrêt précoce (*early stopping*) consiste à interrompre l'entraînement d'un modèle dès que l'erreur sur un ensemble de validation cesse de diminuer. On réserve une portion des données pour l'ensemble de validation et on arrête l'optimisation dès qu'aucune amélioration n'est observée sur cette métrique. Cette méthode empêche les poids de s'ajuster indéfiniment aux données d'apprentissage et agit comme une régularisation implicite (GOODFELLOW et al., 2016). Il a été démontré que, dans certains modèles (par exemple la régression linéaire simple ou un réseau de neurones de petite taille), l'arrêt précoce est équivalent à une pénalisation L_2 appropriée (GOODFELLOW et al., 2016) ; (BISHOP, 2006). En pratique, on introduit souvent un paramètre de *patience* indiquant le nombre d'époques consécutives sans amélioration avant d'arrêter l'entraînement (PRECHELT, 1998). Cette stratégie, simple à implémenter, est l'une des premières formes de régularisation utilisées en apprentissage profond.

A.4.3 Réduction du taux d'apprentissage sur plateau

La stratégie *ReduceLROnPlateau* est une technique adaptative de planification du taux d'apprentissage. Elle consiste à réduire le pas d'apprentissage lorsqu'une métrique de performance (généralement la perte de validation) cesse de s'améliorer pendant plusieurs époques (fixées par un paramètre de *patience*). Concrètement, si la validation stagne, on multiplie le taux d'apprentissage η par un facteur $\gamma < 1$ (souvent $\gamma = 0.1$), ce qui revient à appliquer $\eta \leftarrow \gamma\eta$. L'idée sous-jacente est qu'après une phase de descente rapide initiale, diminuer le pas d'apprentissage permet de converger plus finement vers un optimum local. Cette méthode heuristique, implémentée dans des frameworks tels que PyTorch ou Keras, peut améliorer la convergence finale en adaptant automatiquement le calendrier d'apprentissage au comportement effectif de la perte (RUDER, 2016). Parmi les variantes, on peut citer des schémas tels que le *cosine annealing* ou les *warm restarts*, qui reposent eux aussi sur l'idée d'une réduction dynamique du taux d'apprentissage.

A.4.4 Décroissance exponentielle du taux d'apprentissage

La méthode de décroissance exponentielle du taux d'apprentissage prévoit de diminuer le pas d'apprentissage de façon continue au fil des itérations. Par exemple, on peut utiliser

$$\eta_t = \eta_0 e^{-\alpha t},$$

où η_0 est le taux initial et $\alpha > 0$ est un hyperparamètre de décroissance. Ce schéma, inspiré du recuit simulé, assure une réduction progressive du pas afin de raffiner la convergence vers l'optimum final. D'un point de vue théorique, un taux décroissant est souvent nécessaire

pour garantir la convergence stochastique : ainsi, les conditions classiques de Robbins-Monro imposent généralement $\sum_{t=1}^{\infty} \eta_t = \infty$ et $\sum_{t=1}^{\infty} \eta_t^2 < \infty$ (GOODFELLOW et al., 2016). En pratique, la décroissance exponentielle offre un compromis raisonnable : si α est trop faible, la convergence sera lente, tandis qu'un α trop élevé peut arrêter prématurément l'entraînement. Parmi les variantes, on trouve les schémas de *step decay* (où le taux est divisé par un facteur constant tous les K itérations) ou de décroissance en temps (par exemple $\eta_t = \eta_0/(1 + \beta t)$), qui reposent sur des logiques analogues.

Annexe B

Annexes : Résultats et tableaux complémentaires

B.1 analyse exploratoire des données

B.1.1 Analyse en Composantes Principales (ACP)

Variable adl

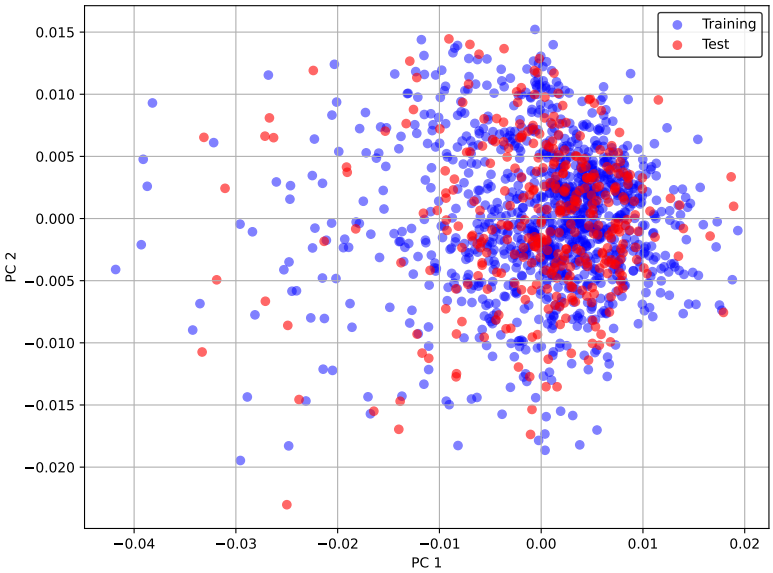


FIGURE B.1 – Projection des ensembles d'entraînement et de test sur le plan principal (PC1 et PC2), pour la variable `adl`.

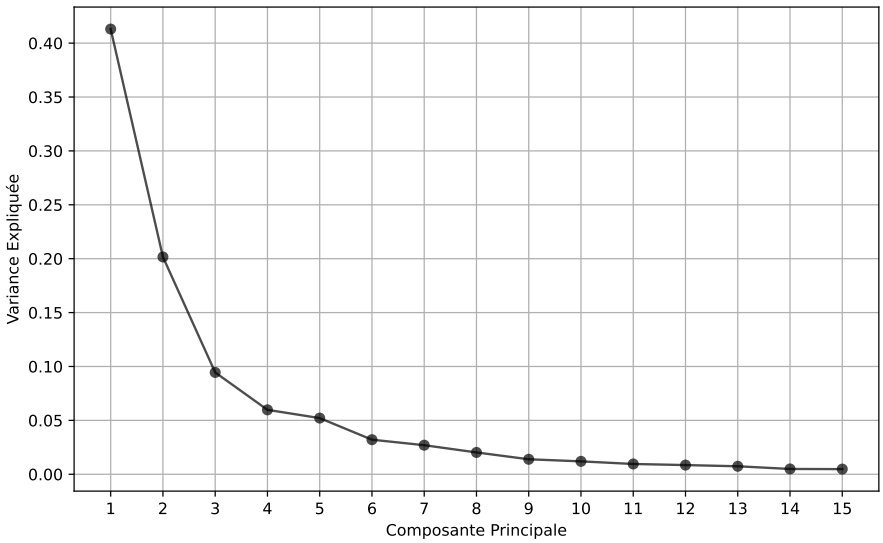


FIGURE B.2 – Distribution de la variance expliquée par chacune des composantes principales extraites de l'ensemble d'entraînement, pour la variable `adl`.

Variable cf

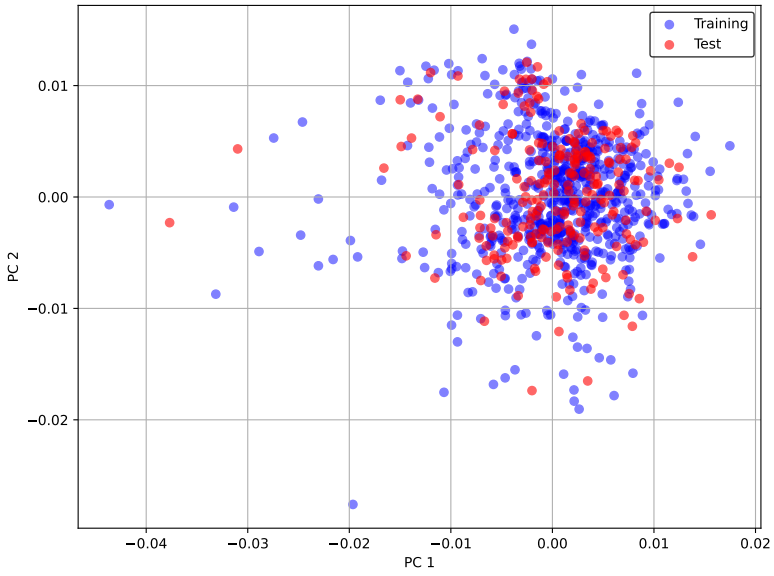


FIGURE B.3 – Projection des ensembles d'entraînement et de test sur le plan principal (PC1 et PC2), pour la variable `cf`.

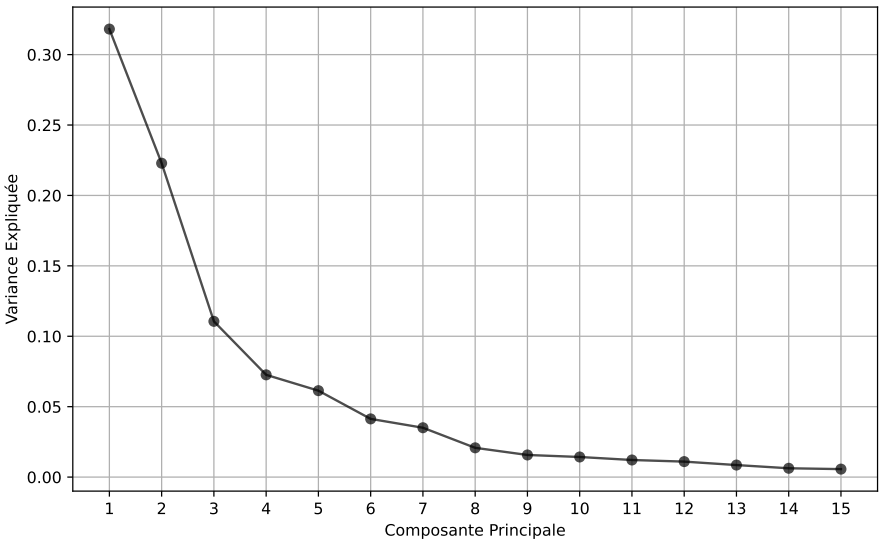


FIGURE B.4 – Distribution de la variance expliquée par chacune des composantes principales extraites de l'ensemble d'entraînement, pour la variable `cf`.

Variable cp

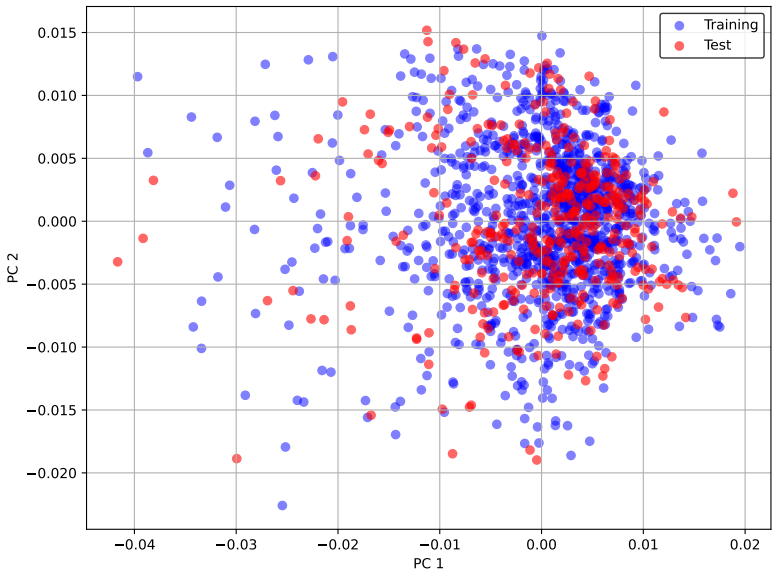


FIGURE B.5 – Projection des ensembles d'entraînement et de test sur le plan principal (PC1 et PC2), pour la variable `cp`.

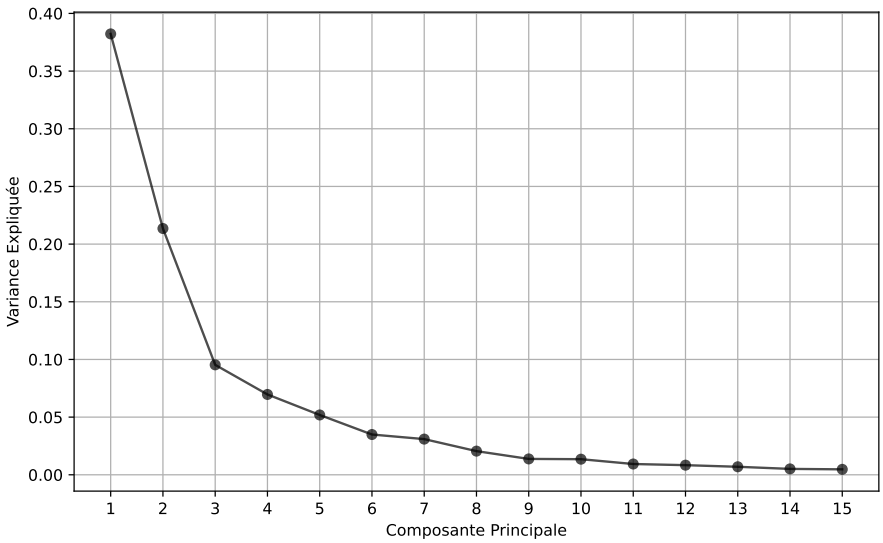


FIGURE B.6 – Distribution de la variance expliquée par chacune des composantes principales extraites de l'ensemble d'entraînement, pour la variable `cp`.

Variable dmdcell

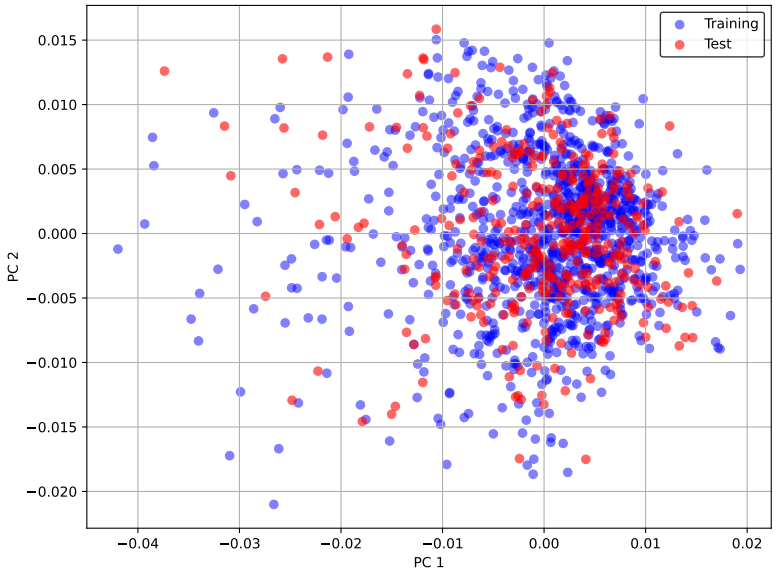


FIGURE B.7 – Projection des ensembles d'entraînement et de test sur le plan principal (PC1 et PC2), pour la variable `dmdcell`.

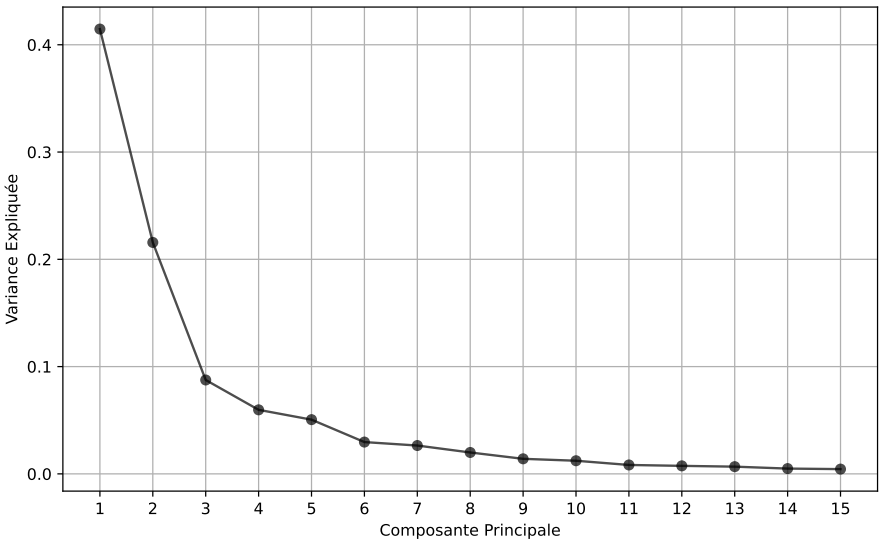


FIGURE B.8 – Distribution de la variance expliquée par chacune des composantes principales extraites de l'ensemble d'entraînement, pour la variable `dmdcell`.

Variable ndf

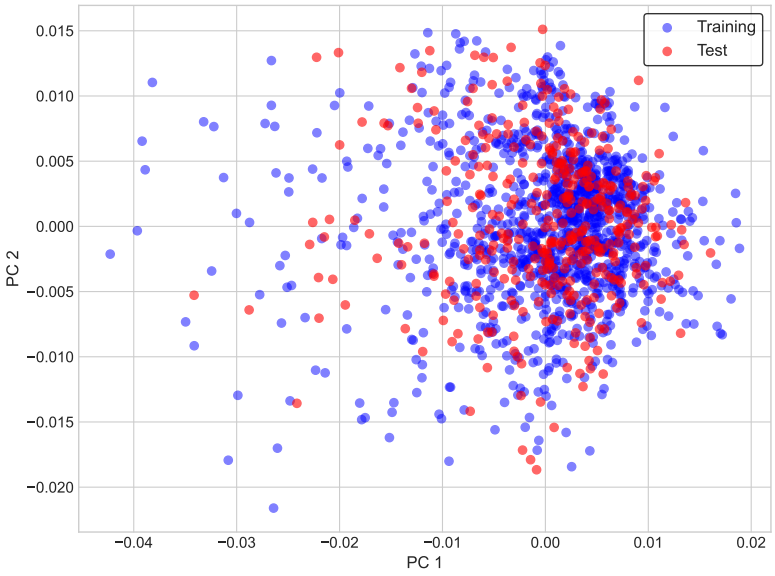


FIGURE B.9 – Projection des ensembles d'entraînement et de test sur le plan principal (PC1 et PC2), pour la variable `ndf`.

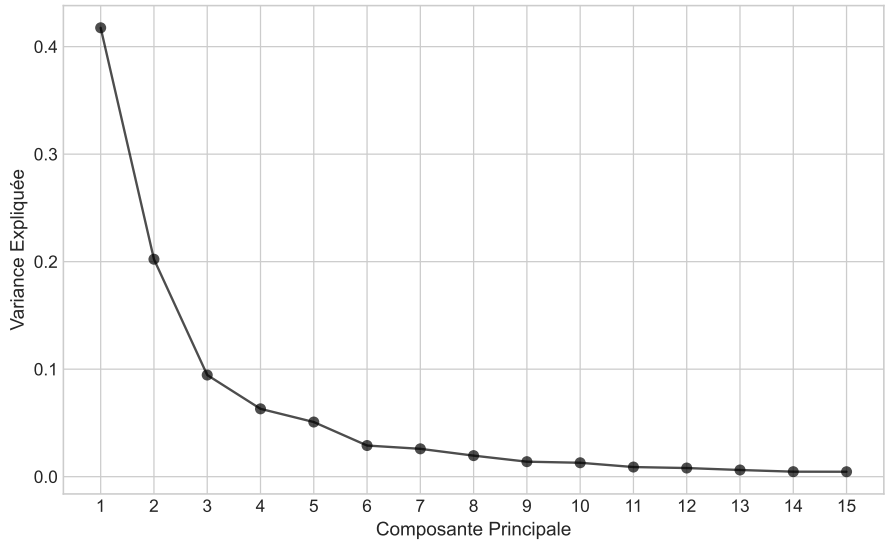


FIGURE B.10 – Distribution de la variance expliquée par chacune des composantes principales extraites de l'ensemble d'entraînement, pour la variable `ndf`.

B.1.2 Analyse visuelle des distributions

Variable adl

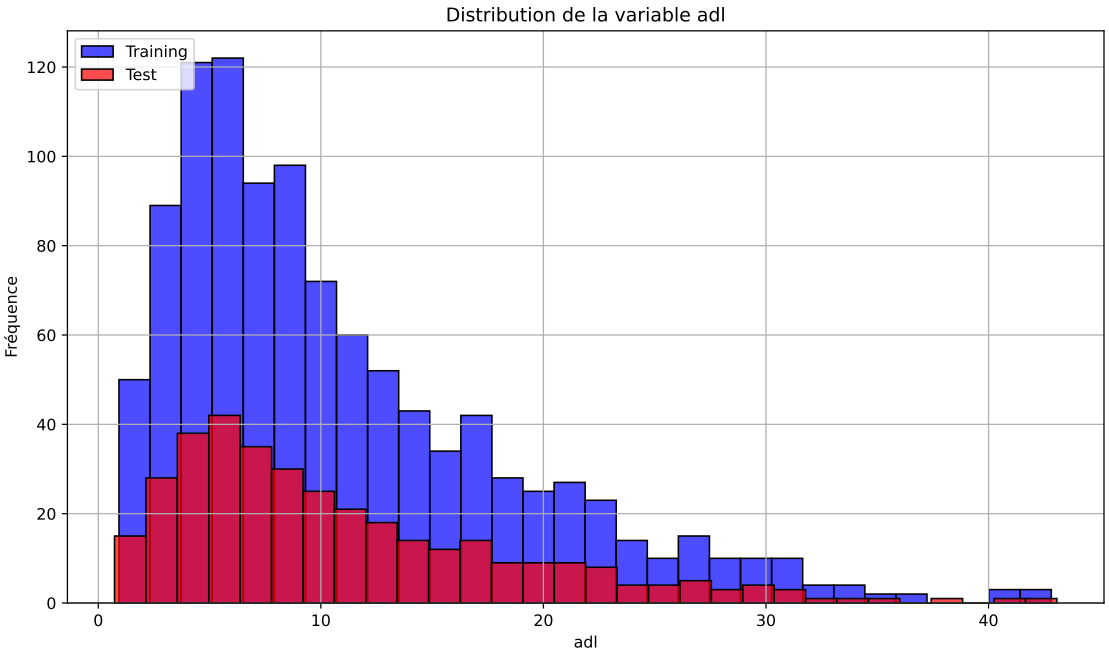


FIGURE B.11 – Comparaison des histogrammes de fréquence pour la variable adl entre les ensembles d'entraînement et de test

Variable cf

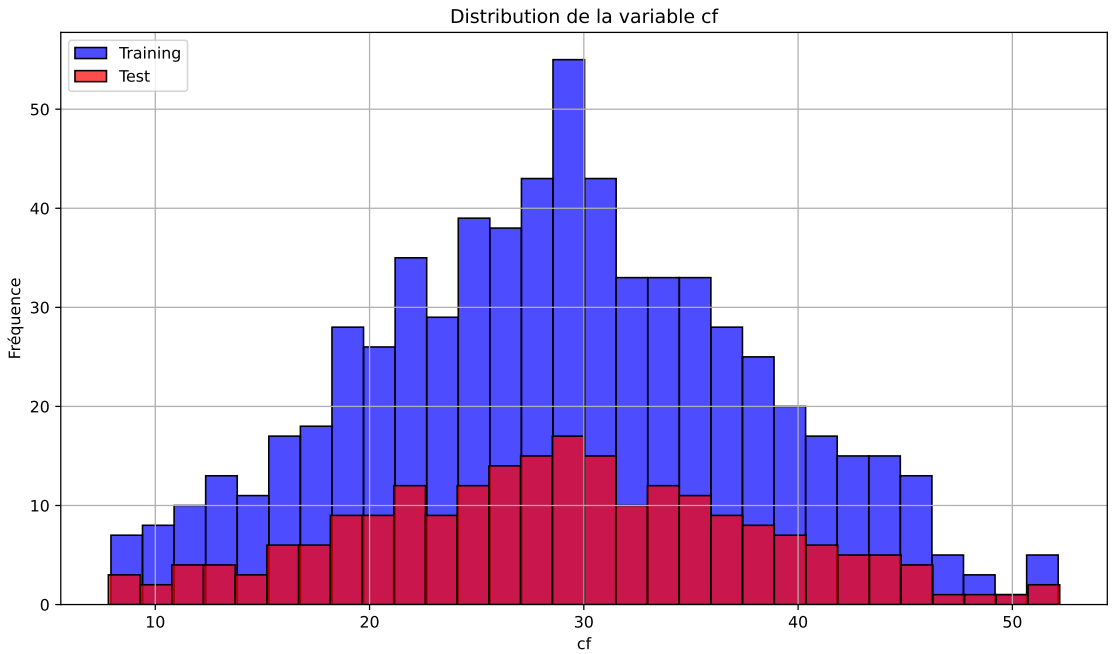


FIGURE B.12 – Comparaison des histogrammes de fréquence pour la variable cf entre les ensembles d'entraînement et de test

Variable cp

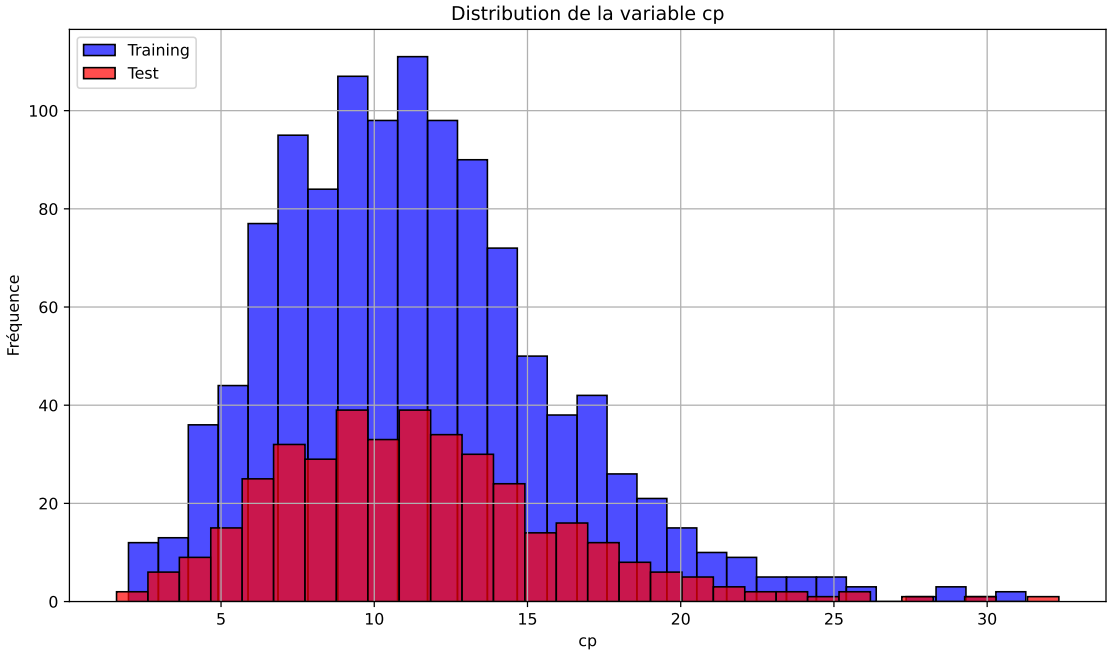


FIGURE B.13 – Comparaison des histogrammes de fréquence pour la variable cp entre les ensembles d'entraînement et de test

Variable dmdcell

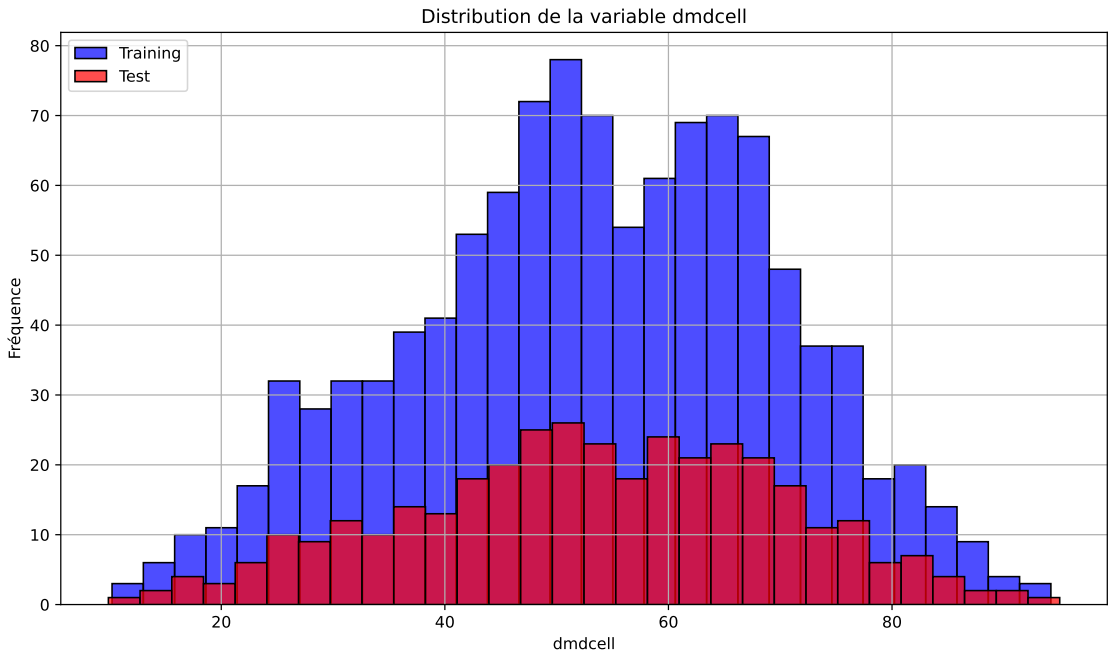


FIGURE B.14 – Comparaison des histogrammes de fréquence pour la variable dmdcell entre les ensembles d'entraînement et de test

Variable ndf

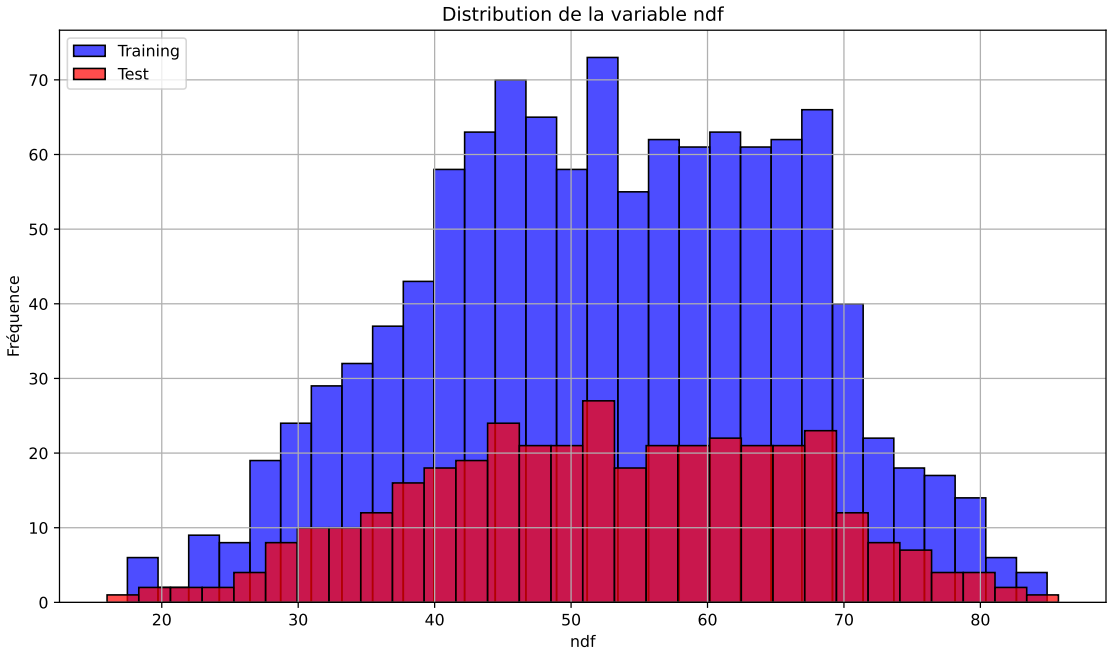


FIGURE B.15 – Comparaison des histogrammes de fréquence pour la variable ndf entre les ensembles d'entraînement et de test

Bibliographie

AKIBA, Takuya et al. (2019). « Optuna: A Next-generation Hyperparameter Optimization Framework ». In : *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

ANDERSON, N.T. et al. (jan. 2021). « Achieving robustness across season, location and cultivar for a NIRS model for intact mango fruit dry matter content. II. Local PLS and nonlinear models ». In : *Postharvest Biology and Technology* 171, p. 111358. DOI : 10.1016/j.postharvbio.2020.111358. (Visité le 05/06/2022).

BHAGWAT, S et al. (mars 2024). « A Review on IR spectroscopy ». In : *International Research Journal of Modernization in Engineering Technology and Science International Research Journal of Modernization in Engineering* 6, p. 2582-5208. URL : https://www.irjmets.com/uploadedfiles/paper//issue_3_march_2024/51042/final/fin_irjmets1711723552.pdf.

BISHOP, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer.

CATALTAS, Ozcan et Kemal TUTUNCU (mars 2023). « Detection of protein, starch, oil, and moisture content of corn kernels using one-dimensional convolutional autoencoder and near-infrared spectroscopy ». In : *PeerJ Computer Science* 9, e1266-e1266. DOI : 10.7717/peerj-cs.1266. (Visité le 13/04/2025).

CHUNG, Jihoon et al. (sept. 2024). « Imbalanced spectral data analysis using data augmentation based on the generative adversarial network ». In : *Scientific reports* 14, p. 13230. DOI : 10.1038/s41598-024-63285-4. URL : https://pubmed.ncbi.nlm.nih.gov/38853181/.

DONATO, Jesus et al. (juin 2015). « Utilisation de la spectroscopie dans le proche infrarouge et de la spectroscopie de fluorescence pour estimer la qualité et la traçabilité de la viande ». In : *INRAE Productions Animales* 28. DOI : 10.20870/productions-animales.2015.28.2.3025. URL : https://www.researchgate.net/publication/282864085_Utilisation_de_la_spectroscopie_dans_le_proche_infrarouge_et_de_la_spectroscopie_de_fluorescence_pour_estimer_la_qualite_et_la_tracabilite_de_la_viande.

EL FALEH, E.M (2019). *Faculté des Sciences Département de Géologie*. URL : https://fad.umi.ac.ma/pluginfile.php/17690/mod_folder/content/0/MGSA_S2_IRTF_EL%20FALEH.pdf?forcedownload=1 (visité le 16/07/2025).

FU, Pengyou et al. (avr. 2022). « SpectraTr: A novel deep learning model for qualitative analysis of drug spectroscopy based on transformer structure ». In : *Journal of Innovative Optical Health Sciences* 15. DOI : 10.1142/s1793545822500213.

GOODFELLOW, Ian et al. (2016). *Deep Learning*. MIT Press.

HAFFNER, F. et al. (jan. 2025). « IPA: A deep CNN based on Inception for Petroleum Analysis ». In : *Fuel* 379, p. 133016. DOI : 10.1016/j.fuel.2024.133016.

HOERL, Roger W. (oct. 2020). « Ridge Regression: A Historical Context ». In : *Technometrics* 62, p. 420-425. DOI : 10.1080/00401706.2020.1742207.

LESNOFF, Matthieu (2021). *Jchemo: Chemometrics and machine learning on high-dimensional data with Julia*. UMR SELMET, Univ Montpellier, CIRAD, INRA, Institut Agro. URL : https://github.com/mlesnoff/Jchemo.jl (visité le 28/07/2025).

— (nov. 2023). « Averaging a local PLSR pipeline to predict chemical compositions and nutritive values of forages and feed from spectral near infrared data ». In : *Chemometrics and Intelligent Laboratory Systems* 244, p. 105031-105031. DOI : 10.1016/j.chemolab.2023.105031. (Visité le 17/07/2025).

MA, Xiaoyan et al. (jan. 2018). « Nonlinear Regression with High-Dimensional Space Mapping for Blood Component Spectral Quantitative Analysis ». In : *Journal of spectroscopy* 2018, p. 1-8. DOI : 10.1155/2018/2689750. (Visité le 01/07/2024).

MARTÍN ABADI et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. URL : https://www.tensorflow.org/.

MARTINS, J.A. et al. (juin 2022). « SpectraNet–53: A deep residual learning architecture for predicting soluble solids content with VIS–NIR spectroscopy ». In : *Computers and Electronics in Agriculture* 197, p. 106945. DOI : 10.1016/j.compag.2022.106945.

MISHRA, Puneet et Dário PASSOS (mai 2021). « A synergistic use of chemometrics and deep learning improved the predictive performance of near-infrared spectroscopy models for dry matter prediction in mango fruit ». In : *Chemometrics and Intelligent Laboratory Systems* 212, p. 104287. DOI : 10.1016/j.chemolab.2021.104287. (Visité le 16/10/2021).

MISHRA, Puneet, Dário PASSOS et al. (déc. 2022). « Deep learning for near-infrared spectral data modelling: Hypes and benefits ». In : *TrAC Trends in Analytical Chemistry* 157, p. 116804. DOI : 10.1016/j.trac.2022.116804. (Visité le 21/01/2023).

MURPHY, Kevin P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

NAME, Your (2025). *Source Code for Master's Thesis: Deep Learning Models for Predicting Forage Chemical Composition*. URL : https://github.com/AliD47/FPE.

PASSOS, Dário (mars 2025). « Deep tutti-frutti II: Explainability of CNN architectures for fruit dry matter predictions ». In : *Spectrochimica Acta Part A Molecular and Biomolecular Spectroscopy*, p. 126068-126068. DOI : 10.1016/j.saa.2025.126068. (Visité le 23/07/2025).

PASSOS, Dário et Puneet MISHRA (avr. 2022). « A tutorial on automatic hyperparameter tuning of deep spectral modelling for regression and classification tasks ». In : *Chemometrics and Intelligent Laboratory Systems* 223, p. 104520. DOI : 10.1016/j.chemolab.2022.104520.

— (déc. 2023). « Deep Tutti Frutti: Exploring CNN architectures for dry matter prediction in fruit from multi-fruit near-infrared spectra ». In : *Chemometrics and Intelligent Laboratory Systems* 243, p. 105023. DOI : 10.1016/j.chemolab.2023.105023. (Visité le 13/12/2024).

PEDREGOSA, Fabian et al. (2011). « Scikit-learn: Machine Learning in Python ». In : *Journal of Machine Learning Research* 12.85, p. 2825-2830. URL : http://jmlr.org/papers/v12/pedregosa11a.html.

PLEVRIS, V. et al. (jan. 2022). « Investigation of performance metrics in regression analysis and machine learning-based prediction models ». In : *8th European Congress on Computational Methods in Applied Sciences and Engineering*. DOI : 10.23967/eccomas.2022.155.

PRECHELT, Lutz (1998). « Early Stopping—But When? » In : *Neural Networks: Tricks of the Trade*, p. 55-69.

— (2012). « Early Stopping — But When? » In : *Lecture Notes in Computer Science* 7700, p. 53-67. DOI : 10.1007/978-3-642-35289-8_5.

RELANDER, Filip A. J. et al. (2022). « Using near-infrared spectroscopy and a random forest regressor to estimate intracranial pressure ». In : *Neurophotonics* 9.4, p. 045001. DOI : 10.1117/1.NPh.9.4.045001. URL : https://doi.org/10.1117/1.NPh.9.4.045001.

RUDER, Sebastian (2016). « An overview of gradient descent optimization algorithms ». In : *arXiv preprint arXiv:1609.04747*.

RUMELHART, David E et al. (1986). « Learning representations by back-propagating errors ». In : *Nature* 323.6088, p. 533-536.

SOUVIK et al. (sept. 2023). « Deep chemometrics using one-dimensional convolutional neural networks for predicting crude oil properties from FTIR spectral data ». In : *The Canadian Journal of Chemical Engineering* 101, p. 6688-6700. DOI : 10.1002/cjce.25076. (Visité le 18/07/2025).

SZEGEDY, Christian, Wei LIU et al. (2015). « Going deeper with convolutions ». In : *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 1-9. DOI : 10.1109/cvpr.2015.7298594.

SZEGEDY, Christian, Vincent VANHOUCKE et al. (2015). *Rethinking the Inception Architecture for Computer Vision*. arXiv.org. URL : https://arxiv.org/abs/1512.00567.

TIBSHIRANI, Robert (1996). « Regression Shrinkage and Selection via the Lasso ». In : *Journal of the Royal Statistical Society. Series B (Methodological)* 58, p. 267-288. URL : https://www.jstor.org/stable/2346178.

TRAN, Tran et al. (jan. 2021). « Graph Neural Networks in Cheminformatics ». In : *Advances in intelligent systems and computing*, p. 823-837. DOI : 10.1007/978-3-030-68154-8_71. (Visité le 18/07/2025).

WALSH, Jeremy et al. (mai 2023). « Review: The evolution of chemometrics coupled with near infrared spectroscopy for fruit quality evaluation. II. The rise of convolutional neural networks ». In : *Journal of Near Infrared Spectroscopy* 31, p. 109-125. DOI : 10.1177/09670335231173140. (Visité le 16/12/2024).

WAN, Jian et al. (mars 2017). « A Comparative Investigation of the Combined Effects of Pre-Processing, Wavelength Selection, and Regression Methods on Near-Infrared Calibration Model Performance ». In : *Applied Spectroscopy* 71, p. 1432-1446. DOI : 10.1177/0003702817694623. (Visité le 12/05/2025).

WEERTS, Hilde J. P. et al. (juill. 2020). *Importance of Tuning Hyperparameters of Machine Learning Algorithms*. arXiv.org. DOI : 10.48550/arXiv.2007.07588. URL : https://arxiv.org/abs/2007.07588.

WITTEVEEN, Mark et al. (oct. 2022). « Comparison of preprocessing techniques to reduce nontissue-related variations in hyperspectral reflectance imaging ». In : *Journal of Biomedical Optics* 27. DOI : 10.1117/1.jbo.27.10.106003. (Visité le 24/10/2024).

WU, Man et al. (déc. 2021). « Deep learning data augmentation for Raman spectroscopy cancer tissue classification ». In : *Scientific Reports* 11. DOI : 10.1038/s41598-021-02687-0. (Visité le 31/05/2022).

YUFENG, Yufeng et al. (fév. 2024). « Une introduction à l'analyse des sols par spectroscopie dans le visible et le proche infrarouge (vis-NIR) ainsi que dans le moyen infrarouge (MIR) ». In : DOI : 10.4060/cb9005fr. (Visité le 16/07/2025).

ZELA, Arber et al. (jan. 2018). « Towards Automated Deep Learning: Efficient Joint Neural Architecture and Hyperparameter Search ». In : *arXiv (Cornell University)*. DOI : 10.48550/arxiv.1807.06906. (Visité le 23/04/2025).

ZHANG, Xiaolei, Tao LIN et al. (juin 2019). « DeepSpectra: An end-to-end deep learning approach for quantitative spectral analysis ». In : *Analytica Chimica Acta* 1058, p. 48-57. DOI : 10.1016/j.aca.2019.01.002. URL : https://www.sciencedirect.com/science/article/pii/S0003267019300169 (visité le 05/01/2023).

ZHANG, Xiaolei, Jinfan XU et al. (juill. 2018). « Convolutional neural network based classification analysis for near infrared spectroscopic sensing ». In : *2015 ASABE International Meeting*. DOI : 10.13031/aim.201800346. (Visité le 17/07/2025).

ملخص

يُعد تقييم جودة الأعلاف باستخدام التحليل الطيفي بالأشعة تحت الحمراء القريبة (NIRS) قضية رئيسيَّة، خاصةً نظرًا لطبيعته غير الإتلافية وسرعة تحليل مكونات الأعلاف. يتناول هذا المشروع إشكالية نمذجة البيانات الطيفية، التي تتميز بأبعادها العالية، وتباينها الكبير، وعلاقاتها غير الخطية المعقدة. بالنظر إلى الحجم المحدود نسبيًا لمجموعات البيانات هذه، تتمثل المسألة المحورية في تحديد ما إذا كانت البنى الجديدة للتعلم العميق يمكن أن تكون فعالة، وكيفية تموضعها مقارنة بالطرق الكيمومترية المتطورة وعالية الأداء.

بالاعتماد على مجموعة بيانات خاصة من مركز "سيراد-سيلميت" (CIRAD-Selmet)، يقدم هذا العمل تقييمًا مقارنًا بين نموذج kNN-LWPLSR، وهو نموذج مرجعي محلي وغير خطي، وثلاث استراتيجيات للتعلم العميق: الأولى هي مقاربة التفافية موجّهة ومباشرة (D-CNN1)، والثانية هي بنية متعددة المقاييس أكثر تعقيدًا مستوحاة من معمارية "إنسيبشن" (Inception)، والثالثة تعتمد على استخلاص الميزات بشكل غير موجّه عبر مرمّز تلقائي التفافي (D-CAE1). وقد تم تطوير ستة نماذج متمايزة للتنبؤ بستة متغيرات كيميائية رئيسية (cp, ndf, adf, adl, cf, dmdcell).

تُظهر النتائج أن نموذج kNN-LWPLSR كان الأفضل أداءً لأربعة من المتغيرات الستة (adf, cf, cp, and ndf)، مما يبرز فعالية استراتيجيته في النمذجة المحلية للتعامل مع تباين البيانات. في المقابل، أظهر أحد نماذج التعلم العميق (CNN-R_v1E) أداءً متفوقًا للمتغيرين adl و dmdcell، مما يؤكد إمكانات استخلاص الميزات عبر عملية الالتفاف. وقد حققت جميع النماذج الأفضل أداءً مستوى تنبؤ يُعتبر " جيدا"، حيث تجاوزمعامل الأداء إلى نسبة الإنحراف المعياري قيمة 3.

ختامًا، يؤكد هذا العمل أن الطرق الكيمومترية المحلية المتخصصة مثل kNN-LWPLSR تحتفظ بمتانتها، وأن مقاربات التعلم العميق لم تتفوق عليها بعد بشكل منهجي في معالجة هذا النوع من الإشكاليات. على الرغم من أن التعلم العميق يُعد واعدًا، إلا أن أداءه الحالي في هذه الدراسة يشير إلى أنه، بالنسبة لمجموعة البيانات هذه، لا يبرر بشكل منهجي تكلفته الحسابية وتعقيده المتزايدين دون تحسينات منهجية واستكشاف أعمق. حاليًا، يمثل نموذج kNN-LWPLSR تسوية عملية متفوقة تجمع بين الأداء والكفاءة لهذا النوع من البيانات.

كلمات مفتاحية: التحليل الطيفي بالأشعة تحت الحمراء القريبة (NIRS)، الكيمومترية، التعلم العميق، تعلم الآلة، جودة الأعلاف، الانحدار، kNN-LWPLSR، الشبكات العصبونية الالتفافية (CNN)، المرمّز التلقائي (AE)، سيراد (CIRAD).

المملكة المغربية
ROYAUME DU MAROC

INSTITUT AGRONOMIQUE
ET VETERINAIRE HASSAN II

معهد الحسن الثاني للزراعة
والبيطرة

مشروع نهاية الدراسات لنيل دبلوم مهندس دولة في الزراعة

تخصص: علم البيانات في الفلاحة

تقييم أداء نماذج التعلم العميق للتنبؤ بالتركيبالكيميائي للكلأ من بيانات مطيافية الأشعة تحت الحمراء القريبة

قدم للعموم ونوقش من طرف

عبد العلي دسم

أمام اللجنة المكونة من:

الأستاذ حمودة علال	رئيس اللجنة	**معهد الحسن الثاني للزراعة والبيطرة**
الأستاذة بنسعلي سلوى	مقررة	**معهد الحسن الثاني للزراعة والبيطرة**
الدكتور ماثيو ليسنوف	مقرر	**المركز الفرنسي للبحوث الزراعية من أجل التنمية**
الأستاذ العيادي سفيان	ممتحن	**معهد الحسن الثاني للزراعة والبيطرة**

يوليوز 2025