



HAL
open science

Stratégies d'optimisation des hyper-paramètres de réseaux de neurones appliqués aux signaux temporels biomédicaux

Souhir Khessiba

► To cite this version:

Souhir Khessiba. Stratégies d'optimisation des hyper-paramètres de réseaux de neurones appliqués aux signaux temporels biomédicaux. Imagerie médicale. Institut Polytechnique de Paris; Université de Sousse (Tunisie), 2024. Français. NNT: 2024IPPAE003 . tel-04848907

HAL Id: tel-04848907

<https://theses.hal.science/tel-04848907v1>

Submitted on 19 Dec 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

NNT : 2024IPPAAE003

Thèse de doctorat



Stratégies d'optimisation des hyperparamètres de réseaux de neurones appliqués aux signaux temporels biomédicaux

Thèse de doctorat de l'Institut Polytechnique de Paris
Préparée à l'École nationale supérieure de techniques avancées

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat : Signal, Images, Automatique et Robotique

Thèse présentée et soutenue à Sousse, Tunisie, le 21/11/2024 par

Souhir KHESSIBA

Composition du Jury :

Mohamed Ali Mahjoub Professeur, Université de Sousse (– Laboratoire LATIS)	Président
Catherine Achard Professeure, Sorbonne Université (–Piros)	Rapportrice
Yassine Ben Ayed Professeur, Université de Sfax (– Laboratoire MIRACL)	Rapporteur
Antoine Manzanera Professeur, ENSTA Paris (–U2IS)	Directeur de thèse
Asma Ben Abdallah Professeure, Université de Monastir (– LABTIM)	Co-Directrice de thèse

Remerciements

Avant de rentrer dans le vif du sujet, je tiens à réserver ces quelques lignes pour adresser mes remerciements à tous ceux qui ont contribué directement ou indirectement à la réalisation et à l'accomplissement de cette thèse.

Je tiens à exprimer mes plus sincères remerciements aux *membres du jury* qui ont accepté de consacrer leur temps et leur expertise à l'évaluation de cette thèse. Votre engagement et vos retours sont inestimables, et je suis reconnaissante de bénéficier de vos connaissances.

Je remercie tout particulièrement le président du jury, *Pr. Mohamed Ali Mahjoub*, pour sa précieuse contribution et son soutien tout au long de ce processus. Votre expertise et votre dévouement ont été essentiels à la qualité de cette thèse.

Je tiens à exprimer ma profonde gratitude à la *Pr. Catherine Achard*, rapporteuse de cette thèse. Ses commentaires détaillés et ses suggestions ont apporté une valeur inestimable à cette recherche. Sa perspicacité et ses conseils ont contribué de manière significative à l'amélioration de ce travail. Je suis sincèrement reconnaissant pour sa rigueur scientifique et son expertise qui ont été des piliers tout au long de ce processus.

Je tiens à exprimer ma reconnaissance tout particulièrement au *Pr. Yassine Ben Ayed*, rapporteur de cette thèse. Ses commentaires détaillés et constructifs ont enrichi ce travail. Je suis profondément reconnaissant pour sa rigueur scientifique et son expertise qui ont été des atouts précieux tout au long de ce processus.

Je remercie chaleureusement *Pr. Raoudha Ben Jemaa* d'avoir accepté d'être examinatrice de cette thèse. Sa disponibilité et son engagement tout au long du processus ont été très appréciés.

J'adresse mes sincères remerciements et ma gratitude à mes directeurs de thèse **Pr. Asma BEN ABDALLAH** et **Pr. Antoine MANZANERA**, et à mon co-directeur **Dr. Ahmed Ghazi BLAEICH** pour leur soutien continu tout au long de la thèse, leurs commentaires, leurs discussions et leurs retours. Leurs conseils m'ont été précieux tout au long de mes recherches durant ces années passées à la préparation de ma thèse de doctorat.

Je tiens également à remercier **Pr. Mohamed Hedi BEDOUI**, pour son encouragement et tous les conseils qu'il m'a prodigués durant ma thèse.

Je tiens à exprimer ma gratitude envers tous les membres du laboratoire **LTIM** et du laboratoire **U2IS** qui ont grandement contribué à créer un environnement de travail exceptionnel. Leur bienveillance et leur enthousiasme constant ont été des facteurs déterminants dans mon parcours, me permettant de progresser tout en conservant une motivation constante. Les années passées en leur compagnie resteront inoubliables, notamment les moments de convivialité que nous avons partagés.

Je tiens également à remercier tous les membres administratifs et techniques du **LTIM** et de **l'U2IS** pour leurs aides et leurs disponibilités au quotidien. En particulier, un grand merci à **Yamina AMMAR** et **Axelle SEGUIN**.

Je tiens à remercier du fond du cœur les deux personnes sans lesquelles rien n'aurait été envisageable, **mes parents**. Je le rends également hommage pour m'avoir donné les clés pour m'adapter aux situations radicalement nouvelles que j'ai rencontrées, grâce à leur éducation et aux valeurs qu'ils ont réussi à me transmettre.

Mes remerciements reviennent également à **mes chères sœurs** qui m'ont toujours soutenu et cru en moi, ainsi qu'à tous les membres de ma **famille**. Cette thèse vous est dédiée.

Je n'oublierai pas de témoigner ma gratitude à ma chère amie **Ines**, merci de m'avoir soutenue pendant ma thèse et les moments difficiles que j'ai traversés et de m'avoir toujours encouragée

Je souhaite également remercier sincèrement mon ami **Mohamed Amine**, qui m'a apporté une aide précieuse tout au long de ma thèse et qui a toujours été présent dans les moments difficiles.

Enfin, je tiens à exprimer ma reconnaissance à **Adam** pour son soutien et ses encouragements tout au long de cette période.

Dédicaces

Merci à Dieu de m'avoir donné le pouvoir d'achever ce travail.

Je dédie de tout mon cœur ce travail :

À ma famille qui était toujours présente pour croire en moi,

Pour m'avoir permis de mener à bien ce rêve

Et qui à chaque fois m'a fait confiance quand il m'est arrivé de douter.

Pour leur soutien inconditionnel,

Leur présence discrète dans ma vie

Et aussi pour avoir fait de moi ce que je suis,

Quoi que vous en pensiez,

Je vous assure que si je suis arrivée là, c'est grâce à vous :

Mon père « Neji »

Ma mère « Houda »

Mes chères sœurs Sirine & Sahar

Mon cher frère « Salem »

Mon grand-père « Ahmad » pour son grand amour.

À l'âme de ma grand-mère « Ftouma »

Ma chère amie-sœur « Chaima »

Avec les plus sincères sentiments je dédie ce travail à mon fiancé « Adam » et toute sa famille.

À tous ceux qui me sont chers...

Merci

Résumé

Les progrès récents en apprentissage automatique ouvrent de nouvelles perspectives pour les systèmes d'aide à la décision médicale. Les modèles d'apprentissage profond, en particulier les réseaux de neurones à convolution (CNN), ont démontré des performances exceptionnelles dans les contextes médicaux, offrant des capacités de détection, de classification et de prédiction de diverses pathologies. Cependant, l'optimisation des réseaux CNN pose un défi significatif en raison de nombreux hyperparamètres gouvernant leur performance. Cette thèse aborde ce défi en proposant des méthodologies innovantes pour l'optimisation efficace des hyperparamètres des réseaux CNN dans le contexte des systèmes d'aide à la décision médicale. Deux principales contributions sont présentées : (1) Le développement d'une méthode d'optimisation hybride, GS-TPE pour régler efficacement les hyperparamètres des modèles de réseaux profonds utilisés dans la classification des états de vigilance à partir de signaux physiologiques tels que l'EEG. GS-TPE a montré des améliorations significatives des performances par rapport aux méthodes d'optimisation classiques. (2) L'introduction d'une nouvelle architecture de réseau profond, STGCN, pour classer les gestes liés à l'arthrose du genou et à la maladie de Parkinson à partir de l'analyse vidéo de la marche. STGCN capture les caractéristiques spatio-temporelles des gestes, démontrant son efficacité dans la discrimination entre différentes pathologies, impactant potentiellement les pratiques cliniques.

Abstract

Recent advancements in machine learning have opened up new possibilities for medical decision support systems. Deep learning models, particularly convolutional neural networks (CNNs), have demonstrated outstanding performance in medical contexts, providing capabilities for detection, classification, and prediction of various pathologies. However, optimizing CNNs poses a significant challenge due to numerous hyperparameters governing their performance. This thesis addresses this challenge by proposing innovative methodologies for efficiently optimizing the hyperparameters of CNNs in the context of medical decision support systems. Two main contributions are presented : (1) The development of a hybrid optimization method GS-TPE to effectively optimize the hyperparameters of deep neural network models used in classifying vigilance states from physiological signals such as EEG. GS-TPE has shown significant performance improvements compared to classical optimization methods. (2) The introduction of a new deep network architecture, STGCN, for classifying gestures related to Knee Osteoarthritis and Parkinson's disease from video gait analysis. STGCN captures the Spatio-Temporal characteristics of gestures, demonstrating its effectiveness in discriminating between different pathologies, potentially impacting clinical practices.

Sommaire

Introduction générale.....	1
1 Contexte scientifique et problématique.....	1
2 Objectifs	2
3 Publications	3
4 Structure de la thèse	4
Chapitre 1.....	5
<i>Revue de la littérature sur l'optimisation des hyperparamètres des réseaux de neurones profonds</i>	<i>5</i>
1 Introduction	5
2 Apprentissage automatique	6
3 Types d'apprentissage.....	6
3.1 Apprentissage supervisé	6
3.2 Apprentissage non-supervisé	6
3.3 Apprentissage semi-supervisé	6
3.4 Apprentissage par renforcement	7
4 Réseaux de neurones artificiels.....	7
4.1 Principe de base	7
4.2 Perceptron multicouches (Multi Layer Perceptron MLP)	8
4.3 Auto-encodeurs.....	10
4.4 Réseaux de neurones récurrentes (RNN).....	11
4.5 Réseaux long-short term memory (LSTM)	12
4.6 Réseaux de neurones à convolution.....	14
5 Optimisation des hyperparamètres.....	16
5.1 Hyperparamètres d'un réseau de neurone.....	16
5.2 Problème d'optimisation.....	16
5.2.1 Introduction	16
5.2.2 Définition et notations	17
5.3 Techniques d'optimisation des hyperparamètres.....	17
5.3.1 Méthodes de recherche systématiques	18

5.3.1.1	Recherche par grille	18
5.3.1.2	Recherche aléatoire.....	19
5.3.2	Optimisation bayésienne (BO)	19
5.3.2.1	Processus gaussiens	20
5.3.2.2	Arbres Estimateurs de Parzen (TPE)	21
5.3.3	Optimisation par essaim de particules (PSO).....	23
5.3.4	Algorithme génétique (GA).....	23
5.4	Outils d'optimisation	24
5.4.1	SKlearn.....	24
5.4.2	HyperOpt.....	25
5.4.3	Optuna	25
5.4.4	Sherpa.....	25
5.4.5	Nevergrad.....	25
5.4.6	Talos	25
6	Étude bibliographique sur l'optimisation des hyperparamètres pour les DNN	26
6.1	Méthodes de recherche exhaustive	26
6.2	Algorithmes génétiques-évolutionnaires	27
6.3	Méthodes méta-heuristiques	29
6.4	Méthodes bayésiennes	32
6.5	Méthodes hybrides.....	33
7	Discussion	35
8	Conclusion	39
	Chapitre 2.....	40
	<i>Traitement de Signaux EEG via les Réseaux de Deep Learning : Etat de l'art</i>	40
1	Introduction.....	40
2	Contexte et performance dans la conduite automobile	40
3	L'électroencéphalogramme (EEG)	41
3.1	Historique	41
3.2	Principe et définition.....	42
3.3	Les ondes d'EEG et leurs significations en pratique	42
4	Traitement du signal EEG.....	44
4.1	Acquisition des signaux EEG	44
4.2	Prétraitement des données	45

4.2.1	Échantillonnage	45
4.2.2	Filtrage.....	45
4.2.2.1	Filtrage dans le domaine fréquentiel.....	45
4.2.2.2	Filtrage dans le domaine spatial	46
4.2.3	Fenêtrage	46
4.3	Extraction des caractéristiques	47
4.3.1	Types d’analyses des signaux EEG.....	47
4.3.1.1	Analyse apériodique (Temporelle)	47
4.3.1.2	Analyse spectrale	47
4.3.2	Techniques d'extraction de caractéristiques	48
4.3.2.1	Transformée en ondelette	48
4.3.2.2	Transformée de Fourier	48
4.4	Classification pour la détection de la baisse de vigilance.....	48
5	Étude bibliographique sur l’exploitation des Réseaux Deep Learning dans le Traitement de l'EEG	49
6	Comparaison des différentes méthodes existantes.....	52
7	Description des données EEG et corpus de travail	53
7.1	Base de données.....	53
7.2	Prétraitement.....	55
8	Conclusion	56
	Chapitre 3.....	57
	<i>Optimisation des hyperparamètres des Réseaux de Neurones Profonds pour la classification des états de vigilance</i>	<i>57</i>
1	Introduction.....	57
2	Motivation.....	57
3	Méthodologie : Nouvelle approche d’optimisation des HP des réseaux DL « GS-TPE ».....	58
4	Architectures DL utilisées pour la classification des états de la vigilance	60
4.1	Architecture MLP	61
4.2	61	
4.3	Architecture LSTM.....	61
4.4	Architecture 1D-CNN.....	63
4.5	Architecture 1D-CNN-LSTM.....	64
4.6	Architecture AE-LSTM	65
5	HPs à optimiser	66

6	Implémentation et résultats	70
6.1	Environnement de travail.....	70
6.2	Processus de Préparation des Données EEG pour l'Entraînement des Modèles	71
6.1	Paramétrage des expériences pour l'optimisation des HP	72
6.2	Métriques de performance	73
6.3	Résultats expérimentaux et discussion	73
6.3.1	Résultats sans processus d'optimisation.....	73
6.3.2	Résultats avec optimisation des hyperparamètres	80
7	Conclusion	96
	Chapitre 4.....	97
	<i>Application des DNN pour la classification des pathologies affectant le mouvement humain ..</i>	<i>97</i>
1	Introduction.....	97
2	Pathologies du mouvement et des gestes humains.....	98
2.1	Arthrose	98
2.2	Arthrose de Genou : Gonarthrose	99
2.2.1	Définition.....	99
2.2.2	Facteurs de risques	100
2.3	Maladie de Parkinson	100
3	Classification et Prédiction des Pathologies liées aux Gestes Humains par algorithmes : Revue de littérature.....	101
4	Modélisation du corps humain.....	104
4.1	Modèle cinématique.....	104
4.2	Modèle planaire	104
4.3	Modèle volumétrique.....	105
5	Estimation de pose humaine	105
5.1	Estimation de la pose d'une seule personne	106
5.2	Métriques d'évaluation	107
6	Étude Bibliographique sur les techniques d'estimation de la pose humaine basées sur les DNN	107
7	Implémentation des architectures DNN à base de graphes pour la classification de l'arthrose du Genou	111
7.1	Base de données relative à la marche	111
7.2	Méthodologie.....	112

7.2.1	Pipeline de BlazePose-STGCN : 3D Keypoints Tracking pour la classification de l'arthrose du genou	113
7.2.1.1	Extraction de squelette via BlazePose : Détection des articulations sans marqueurs	113
7.2.1.2	STGCN: Spatial-Temporal Graph Convolutional Network	116
7.3	Expérimentation.....	120
7.3.1	Environnement de travail	122
7.3.2	Métriques de performance.....	122
7.3.3	HPs et modalités d'apprentissage dans les expériences sans processus d'optimisation	123
7.3.3.1	Résultats et discussion avec l'ensemble des articulations BlazePose (33 points, analyse globale).....	123
7.3.3.2	Résultats et discussion avec un nombre limité d'articulations BlazePose (analyse locale à 8 points).....	126
7.3.4	Optimisation des hyperparamètres de l'architecture ST-GCN.....	127
7.3.4.1	Résultats expérimentaux et discussion	128
7.3.4.2	Synthèse et détails relatifs aux meilleures configurations d'HPs pour l'architecture STGCN	134
7.3.5	Comparaison avec une méthode existante de détection de la KOA.....	136
8	Conclusion	138
	Conclusion générale et perspectives.....	139
	Références bibliographiques	142

Liste des figures

Figure 1. 1 : Un neurone artificiel	8
Figure 1. 2 : Architecture de perceptron multicouches	8
Figure 1. 3 : Fonctions d'activation.....	10
Figure 1. 4 : Auto-encodeur	11
Figure 1. 5: Comparaison entre un RNN (à gauche) et un réseau classique (à droite)	11
Figure 1. 6: Cellule LSTM	12
Figure 1. 7 : Opération de convolution.....	14
Figure 1. 8: Opération de Max-Pooling.....	15
Figure 1. 9: Architecture CNN	15
Figure 1. 10: Recherche par grille	18
Figure 1. 11: Algorithme de la recherche par grille [9].....	19
Figure 1. 12 : Recherche aléatoire.....	19
Figure 1. 13: Optimisation monovariante par processus gaussien avec une fonction d'acquisition.	20
Figure 1. 14: Exemple de distribution (monovariante) modélisée par TPE	21
Figure 1. 15 : Pseudo-code de TPE [13]	23
Figure 1. 16: Les étapes génériques d'un algorithme génétique.....	24
Figure 2. 1: L'inventeur de l'électroencéphalographie (EEG) : Hans Berger.....	41
Figure 2. 2: Exemple de tracé EEG (première ligne), et sa décomposition en bandes de fréquences pertinentes (lignes suivantes). L'axe vertical représente la différence de potentiel, en μV [70]. ...	42
Figure 2. 3: Les rythmes cérébrales de l'EEG.....	43
Figure 2. 4: Traitement du signal EEG.....	44
Figure 2. 5: Acquisition du signal EEG	44
Figure 2. 6: Extraction et analyse des signaux EEG	56
Figure 3. 1: Organigramme de l'approche d'optimisation proposée « GS-TPE.....	59
Figure 3. 2 : Architecture MLP	61
Figure 3. 3 : Architecture LSTM détaillé	63
Figure 3. 4 : Architecture 1D-CNN.....	64
Figure 3. 5 : Architecture 1D-CNN-LSTM.....	65
Figure 3. 6: Architecture AE-LSTM	66

Figure 3. 7 : Analyse monosujet des performances en terme d'accuracy pour la classification des états de vigilance	75
Figure 3. 8 : Matrice de confusion du modèle AE-LSTM-patient-6.....	75
Figure 3. 9 : Évolution de l'accuracy et de la Loss pour le modèle AE-LSTM sur le patient P6....	76
Figure 3. 10 : Variation de l'accuracy pour chaque patient avec le mode LOO pour les différents architectures (a) Architecture MLP, (b) Architecture LSTM.....	77
Figure 3. 11 : Variation de l'accuracy pour chaque patient avec le mode LOO pour les différents architectures (a) Architecture 1D-CNN, (b) Architecture CNN- LSTM	78
Figure 3. 12: Analyse comparative des performances des différentes architectures étudié.....	79
Figure 3. 13: Évolution de l'Accuracy et de la Loss pour le Modèle AE-LSTM_v2.....	79
Figure 3. 14 : Évolution de l'accuracy du modèle 1D-CNN du patient-5 en fonction du nombre d'itérations TPE	81
Figure 3. 15 : Performance en terme d'Accuracy de l'approche GSTPE pour l'architecture 1D-CNN selon le nombre d'itérations : Partie TPE	82
Figure 3. 16 : (a) Evolution de l'accuracy en fonction du nombre des combinaisons RS pour l'architecture CNN-LSTM (b) Evolution de l'accuracy en fonction du nombre des combinaisons GS pour l'architecture 1D-CNN-LSTM.....	86
Figure 3. 17 : (a) Evolution de l'Accuracy en fonction du nombre des combinaisons RS pour l'architecture LSTM (b) Evolution de l'Accuracy en fonction du nombre des combinaisons GS pour l'architecture LSTM.....	86
Figure 3. 18 : Evolution de l'Accuracy en fonction du nombre d'itérations TPE pour l'architecture LSTM (b) Evolution de l'Accuracy en fonction du nombre d'itérations TPE pour l'architecture CNN-LSTM.....	88
Figure 3. 19 : Comparaison des Performances de Modèles pour les deux différents approches GS-TPE et RS-TPE.....	89
Figure 3. 20: : Analyse de l'évolution de la meilleureAccuracy des 5 configurations pour l'approche GSTPE en fonction du nombre d'itérations TPE pour (a) l'architecture MLP (b) l'architecture LSTM (c) l'architecture 1D-CNN (d) l'architecture CNN-LSTM	90
Figure 3. 21 : Analyse de l'évolution de la meilleure Accuracy des 5 configurations pour l'approche GSTPE de l'architecture AE-LSTM en fonction du nombre d'itérations TPE.....	91
Figure 3. 22 : Analyse de l'évolution de la meilleureAccuracy des 5 configurations pour l'approche RSTPE en fonction du nombre d'itérations TPE pour (a) l'architecture MLP (b) l'architecture LSTM (c) l'architecture 1D-CNN (d) l'architecture CNN-LSTM	92
Figure 3. 23: Analyse de l'évolution de la meilleureAccuracy des 5 configurations pour l'approche RSTPE de l'architecture AE-LSTM en fonction du nombre d'itérations TPE	93
Figure 3. 24 : Matrice de confusion de l'architecture LSTM avec GS-TPE.....	93
Figure 3. 25 : Matrice de confusion de l'architecture LSTM avec RS-TPE.....	93
Figure 3. 26 : Matrice de confusion de l'architecture CNN-LSTM avec GS-TPE.....	93
Figure 3. 27 : Matrice de confusion de l'architecture CNN-LSTM avec RS-TPE	93
Figure 4. 1 : Les principales articulations touchées par l'arthrose diagnostiquée en radiologie [113]	98
Figure 4. 2 : Arthrose de Genou	99

Figure 4. 3 : La maladie de Parkinson.....	101
Figure 4. 4: Modélisation du corps humain [135].....	105
Figure 4. 5 : Estimation de la pose humaine	106
Figure 4. 6 : Taxonomie de l'estimation de la pose 2D	106
Figure 4. 7 : Système Codeur-Décodeur pour l'estimation de la pose humaine	107
Figure 4. 8 : Processus de BlazePose-STGCN.....	113
Figure 4. 9 : BlazePose.....	114
Figure 4. 10 : Séquence illustrative à base de squelette à l'aide de BlazePose	115
Figure 4. 11 : Articulations spécifiques de BlazePose	116
Figure 4. 12 : (a) Séquence de graphes de squelettes (b) Représentation des différentes articulations dans le graphe selon leur position	117
Figure 4. 13: Graphe non orienté et sa matrice d'adjacence	117
Figure 4. 14: (a) Séquence de graphes de squelettes, représentant le mouvement humain dans le temps et l'espace. (b) Stratégie d'échantillonnage d'une couche de convolution pour une seule image. (c) Stratégie de partitionnement de la configuration spatiale.....	118
Figure 4. 15: Architecture STGCN	119
Figure 4. 16 : Bloc STGC.....	120
Figure 4. 17 : Matrice de confusion pour l'expérience 1	124
Figure 4. 18 : Matrice de confusion pour l'expérience 2	125
Figure 4. 19: Matrice de confusion pour l'expérience 3	125
Figure 4. 20 : Matrice de confusion pour l'expérience 4	125
Figure 4. 21: : Matrice de confusion pour l'expérience 5	125
Figure 4. 22: Matrice de confusion pour l'expérience 6	126
Figure 4. 23 : Matrice de confusion pour l'expérience 8	127
Figure 4. 24: Matrice de confusion pour l'expérience 10	127
Figure 4. 25: Matrice de confusion pour l'expérience 12	127
Figure 4. 26 : Matrice de confusion pour l'expérience 14	127
Figure 4. 27 : Évolution de la performance du modèle STGCN en fonction du nombre d'itérations TPE pour chaque configuration explorée (Expérience-7).....	131
Figure 4. 28: Évolution de la performance du modèle STGCN en fonction du nombre d'itérations TPE pour chaque configuration explorée (Expérience-9).....	131
Figure 4. 29: Évolution de la performance du modèle STGCN en fonction du nombre d'itérations TPE pour chaque configuration explorée (Expérience-11).....	131
Figure 4. 30: Évolution de la performance du modèle STGCN en fonction du nombre d'itérations TPE pour chaque configuration explorée (Expérience-13).....	132
Figure 4. 31: Évolution de la performance du modèle STGCN en fonction du nombre d'itérations TPE pour chaque configuration explorée (Expérience-15).....	132
Figure 4. 32 : Matrices de confusion des différentes expériences après l'optimisation des HPs avec GS-TPE : (a) Expérience 7, (b) Expérience 9, (c) Expérience 11, et (d) Expérience 13	133
Figure 4. 33 : Matrice de confusion après l'optimisation des hyperparamètres avec GS-TPE pour l'expérience 15	133

Liste des tableaux

Tableau 1. 1 : Tableau comparatif des frameworks d'optimisation d'HP pour les réseaux de neurones.....	26
Tableau 1. 2: Tableau comparatif des algorithmes d'optimisation des HP utilisés pour réseaux de neurones en termes de type, algorithmes d'optimisation, architectures concernées, hyperparamètres et métriques d'évaluation	37
Tableau 2. 1: Méthodes existantes pour l'exploration des réseaux de neurones profonds via l'EEG	52
Tableau 2. 2 : Détails de la Base de Données EEG pour la Classification des États de Vigilance.	55
Tableau 3. 1 : Les HP d'une architecture DL pour la classification des états de vigilance	66
Tableau 3. 2 : HPs clés à optimiser dans l'architecture MLP et leurs espaces de recherche pour les deux algorithmes GS et TPE	68
Tableau 3. 3 : HPs clés à optimiser dans l'architecture LSTM et leurs espaces de recherche pour les deux algorithmes GS et TPE	68
Tableau 3. 4 : HPs clés à optimiser dans l'architecture 1D-CNN et leurs espaces de recherche pour les deux algorithmes GS et TPE	68
Tableau 3. 5: HPs clés à optimiser dans l'architecture 1D-CNN-LSTM et leurs espaces de recherche pour les deux algorithmes GS et TPE	69
Tableau 3. 6 : HPs clés à optimiser dans l'architecture AE-LSTM et leurs espaces de recherche pour les deux algorithmes GS et TPE	69
Tableau 3. 7 : HPs et plages d'exploration pour l'approche d'optimisation GS-TPE	70
Tableau 3. 8 : Création et répartition des séquences par patient (mode monosujet)	71
Tableau 3. 9 : : Création et répartition des séquences pour le mode multi sujet	72
Tableau 3. 10 : Paramétrage des expérimentations effectuées	72
Tableau 3. 11: Résultats en terme d'Accuracy de classification monosujet pour les états de vigilance	74
Tableau 3. 12 : Résultats en terme d'Accuracy de classification pour le mode Leave One Patient Out	76
Tableau 3. 13 : Résultats obtenus pour la classification des états de vigilance avec différentes architectures selon le mode multi-sujets "Train/Val/Test"	78
Tableau 3. 14 : Résultats obtenus sur les données de test pour l'architecture 1D-CNN en utilisant l'approche GS-TPE	80
Tableau 3. 15 : Performance des configurations GS-TPE en termes d'accuracy sur les données de validation du patient-5 (1D-CNN)	82
Tableau 3. 16 : Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'architecture 1D-CNN	83

Tableau 3. 17 : Résultats multi-sujets obtenus avec optimisation pour la classification des états de vigilance	85
Tableau 3. 18 : Performance des configurations GS-TPE en termes d'accuracy sur les données de validation des architectures LSTM, CNN-LSTM et AE-LSTM	88
Tableau 3. 19: Performance des configurations RS-TPE en termes d'accuracy sur les données de validation des architectures LSTM, CNN-LSTM et AE-LSTM	89
Tableau 3. 20 : Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'architecture LSTM	94
Tableau 3. 21: Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'architecture CNN-LSTM	95
Tableau 3. 22: Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'architecture AE-LSTM	95
Tableau 4. 1 : Facteurs de risque liés à l'arthrose de Genou	100
Tableau 4. 2: Récapitulatif sur les modèles DNN relatifs à la classification des pathologies liées aux gestes humains	103
Tableau 4. 3: Métriques d'évaluation des modèles d'estimation de la pose du corps humain	107
Tableau 4. 4: Récapitulatif sur les modèles relatifs à l'estimation de la pose 2D ou 3D	110
Tableau 4. 5: Description détaillée de la base de données KOA-PD-NM en indiquant les sujets concernés et leurs données biologiques.	112
Tableau 4. 6 : Description de la base de données à durée normalisée	120
Tableau 4. 7 : Description des configurations de validation de notre pipeline	121
Tableau 4. 8 : Présentation du paramétrage d'une étude globale à 33 points (expériences de 1 à 10)	122
Tableau 4. 9 : Présentation du paramétrage d'une étude globale à 8 points (expériences de 11 à 20)	122
Tableau 4. 10: HPs utilisés dans les expériences sans processus d'optimisation	123
Tableau 4. 11 : Résultats expérimentaux de l'analyse globale à 33 points pour la classification de l'arthrose de genou et de la maladie de Parkinson	124
Tableau 4. 12 : Résultats expérimentaux de l'analyse locale à 8 points pour la classification de l'arthrose du genou et de la maladie de Parkinson	126
Tableau 4. 13 : HPs clés à optimiser pour l'architecture STGCN	128
Tableau 4. 14 : Résultats expérimentaux obtenus avec optimisation des HP de STGCN pour l'analyse locale à 8 points pour la classification de l'arthrose de genou et de la maladie de Parkinson	129
Tableau 4. 15 : Performance des configurations GS-TPE en termes d'accuracy sur les données de validation en utilisant STGCN	130
Tableau 4. 16: Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'expérience 7	134
Tableau 4. 17: Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'expérience 9	135
Tableau 4. 18: Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'expérience 11	135

Tableau 4. 19: Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'expérience 13	135
Tableau 4. 20: Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'expérience 15	135
Tableau 4. 21: Bilan comparatif entre l'approche proposé et les méthodes existantes	136

Abréviations

DL	Deep Learning
CNN	Convolution Neural Network
SVM	Support Vector Machine
WOA	Whale Optimizer Algorithm
EGO	Efficient Global Optimization
TS	Tabu Search
BO	Bayesian Optimization
TPE	Tree Parzen Estimator
RS	Random Search
RNA	Réseaux de neurones artificiels
SSA	Sparrow Search Algorithm
STFT	Short Time Fourier Transform
GP	Gaussien Process
PSO	Particle Swarm Optimization
GA	Genetic Algorithm
DE	Differential Evolution
OLPSO	Orthogonal Learning Particle Swarm Optimization
APSO	Adaptive Particle Swarm Optimization
WOA	Whale optimizer Algorithm
CWT	Transformée en ondelettes continue
EEG	Electroencéphalographie
HOUSES	Hyperparameter Optimization with sUrrogate-aSsisted Evolutionary Strategy
GWO	Gray Wolf Optimizer
CLSTM	Convolutional Long Short-Term Memory
ANN	Artificial Neural Network
TMS	Troubles Musculo-Squelettiques

RF	Random Forest
RNN	Reccurent Neural Network
GS	Grid Search
MLP	Multi Layer Perceptron
HP	Hyperparameter
DNN	Deep Neural Network
LSTM	Long Short Term Memory
OMS	Organisation Mondiale de la Santé
SGD	Stochastic Gradient Descent
KOA	Knee Osteoarthritis
PD	Parkinson Disease
STGCN	Spatial Temporal Graph Convolution Network
GCN	Graph Convolution network
TCN	Temporal Convolution Network

Introduction générale

1 Contexte scientifique et problématique

Les progrès récents de l'apprentissage automatique ouvrent de nouvelles perspectives pour les systèmes d'aide à la décision médicale. Ces systèmes offrent la possibilité d'améliorer le diagnostic, d'optimiser l'orientation thérapeutique et de prédire l'évolution des maladies [1] [2]. En particulier, les réseaux de neurones profonds, connus sous le nom de "Deep Learning" (DL), ont démontré des performances exceptionnelles dans un contexte médical, permettant la détection, la classification et la prédiction de diverses pathologies. Cette approche gagne rapidement en popularité dans la communauté connexionniste [3]. L'un des aspects les plus prometteurs de l'apprentissage profond dans le domaine de la santé est sa capacité à automatiser les tâches traditionnellement manuelles d'extraction et de traitement des caractéristiques. Dans le domaine des modèles d'apprentissage profond, les réseaux neuronaux convolutifs occupent une place particulièrement importante. Ces réseaux se composent par des couches convolutives pour extraire des caractéristiques et d'autres couches totalement connectées pour assurer la tâche en question (Classification, segmentation, prédiction dans le temps, ...).

Cependant, l'utilisation des CNN (Convolution Neural Network) présente un défi majeur. Ils sont régis par de nombreux hyperparamètres, tels que le nombre de filtres, la taille du filtre, la foulée (stride), le type de padding, la taille de lot (batch size), le nombre d'époques d'entraînement (training epochs), le choix des fonctions d'activation, etc. La détermination empirique des valeurs optimales de ces hyperparamètres constitue un véritable obstacle, car cela implique un temps d'apprentissage considérable et a un impact significatif sur les performances du modèle. Face à la complexité des DNN et à la multitude d'hyperparamètres à considérer, la question principale qui se pose est la suivante : Comment optimiser efficacement les hyperparamètres des réseaux de neurones à convolution pour garantir des performances maximales dans le cadre des systèmes d'aide à la décision médicale ? L'enjeu consiste à trouver une approche d'optimisation qui permettra d'identifier rapidement les meilleures valeurs des hyperparamètres, tout en évitant les inefficiences liées à une exploration empirique fastidieuse de l'espace des hyperparamètres. Une solution adéquate contribuerait ainsi à accélérer le processus de développement et de déploiement de tels systèmes, rendant ainsi leur utilisation plus accessible et pertinente dans le domaine médical.

2 Objectifs

Notre objectif principal est de proposer une méthodologie novatrice visant à optimiser la configuration des hyperparamètres, qui sont généralement fixés de manière empirique. Nous chercherons spécifiquement à améliorer les performances des réseaux neuronaux en recherchant les hyperparamètres qui maximisent la qualité du résultat (optimisation mono-critère). En outre, notre méthodologie aura pour objectif de guider le choix des hyperparamètres du modèle DNN en tenant compte des spécificités de l'application visée et des données d'entrée afin de générer une architecture de réseau optimale.

Notre travail se concentrera principalement sur l'analyse de données biomédicales qui se présentent sous forme de séries temporelles. Ce choix est justifié par l'intérêt porté aux deux applications spécifiques que nous visons, à savoir la détection des états de vigilance et l'analyse de gestes humains à des fins thérapeutiques à savoir arthrose de genou et maladie de Parkinson. Dans les deux domaines, une analyse des données en fonction du temps est nécessaire, étant donné que les données d'entrée, telles que les spectres de puissance et les graphes spatio-temporels de squelettes, sont de tailles raisonnables. Ce type de signaux se prête bien à une étude sur l'exploration des espaces d'hyperparamètres, qui exige un très grand nombre d'entraînements de modèles.

Présentation des contributions

Les contributions majeures de cette thèse sont les suivantes :

- *Optimisation des hyperparamètres pour une meilleure performance des modèles de réseaux de neurones pour la classification de la vigilance :*

Nous avons conçu des modèles de réseaux de neurones profonds pour la classification des états de vigilance à partir de signaux physiologiques (électro-encéphalogrammes, ou EEG) et nous avons proposé une nouvelle méthode d'optimisation de leurs hyperparamètres. Cette méthode hybride, nommée GS-TPE (Grid Search-Tree Parzen Estimator) a permis de régler efficacement les hyperparamètres des modèles, ce qui a entraîné une amélioration significative des performances.

- *La proposition d'une architecture DNN conçue pour la classification des gestes relatifs à l'arthrose du genou et à la maladie de Parkinson (STGCN) :*

Nous avons développé une nouvelle architecture DNN dédiée à l'analyse de graphes extraits de vidéos de marche pour l'analyse de mouvements sans marqueurs. Cette architecture a été élaborée pour capturer les caractéristiques spatio-temporelles des gestes et a prouvé son efficacité

dans la discrimination entre différentes pathologies affectant la marche, offrant ainsi des opportunités significatives pour le diagnostic et le suivi médical. Comme domaine de validation, nous avons choisi la classification des mouvements pathologiques liés à l'arthrose du genou et à la maladie de Parkinson.

Dans ce cadre les contributions suivantes ont été réalisées :

- *Sélection d'une technique d'estimation de pose pour l'extraction des points clés relatifs aux articulations pertinentes pour le mouvement étudié*
- *Proposition d'une architecture d'analyse des graphes caractérisant le geste par les variations de la pose au cours du temps,*
- *Étude comparative de deux approches d'analyse de gestes : globale (33 points sur l'ensemble du corps humain) et locale (8 points autour de la région d'intérêt),*
- *Optimisation des hyperparamètres du réseau ST-GCN pour les configurations réalisées.*

Nous avons montré l'amélioration significative des performances, d'abord par rapport à une optimisation empirique des hyperparamètres, puis par rapport à ce qui existe dans la bibliographie en ce qui concerne la classification des pathologies liées aux gestes. Ces contributions marquent un progrès significatif qui pourrait avoir un impact réel sur les pratiques cliniques et améliorer la qualité de vie des patients.

3 Publications

Les publications suivantes ont été réalisées dans le cadre de cette thèse :

- a) Khessiba, S., Blaeich, A. G., Abdallah, A. B., Grassa, R., Manzanera, A., & Bedoui, M. H. (2023, Septembre). "Improving Knee Osteoarthritis Classification with Markerless Pose Estimation and STGCN Model". *In 2023 IEEE 25th International Workshop on MultiMedia Signal Processing (MMSP 2023) (pp. 1-7) (Classe B).*
- b) Khessiba, S., Blaeich, A. G., Manzanera, A., Ben Khalifa, K., Ben Abdallah, A., & Bedoui, M. H. (2022, Septembre). "Hyperparameters Optimization for Deep Learning Models for EEG Based Vigilance Detection". *In International Conference on Computational Collective Intelligence (CCIS) (pp. 200-210). (Classe B).*
- c) Khessiba, S., Blaeich, A. G., Ben Khalifa, K., Ben Abdallah, A., & Bedoui, M. H. (2021). "Innovative Deep Learning Models for EEG-based vigilance detection". *Neural Computing and Applications, 33, 6921-6937. (Q1, IF : 4.77).*

4 Structure de la thèse

Ce manuscrit de thèse est structuré de la manière suivante :

Dans le premier chapitre, nous établissons le fondement de notre recherche en analysant en détail la littérature relative à l'optimisation des hyperparamètres des réseaux de neurones profonds. Cela nous permet de mieux comprendre les méthodologies, les enjeux et les nouvelles technologies de ce domaine, posant ainsi les fondements conceptuels de notre recherche.

Dans le deuxième chapitre, nous abordons le cœur du traitement des signaux EEG par les DNN. Il est consacré à une étude approfondie sur les signaux EEG, leurs caractéristiques spécifiques et les méthodes et techniques utilisées pour les traiter. Nous effectuons également une étude bibliographique dans laquelle figurent les travaux les plus pertinents relatifs au traitement des signaux EEG à l'aide des DNN.

Le troisième chapitre représente la première contribution de notre thèse. Nous introduisons une nouvelle approche hybride d'optimisation hiérarchique conçue pour la classification des états de vigilance. Outre la méthodologie détaillée, nous présentons les résultats obtenus par cette approche, démontrant son impact significatif sur la performance des modèles. Ce chapitre présente ainsi la portée pratique de nos recherches dans le domaine de l'optimisation.

Le chapitre 4 est consacré à l'amélioration de la gestion et de la prévention de l'arthrose du genou et de la maladie de Parkinson. Nous étudions les approches innovantes qui utilisent les DNN pour évaluer la pose humaine, en particulier dans le contexte de l'arthrose du genou, en explorant des solutions non invasives, i.e. sans marqueurs. Dans la deuxième partie de ce chapitre, nous présentons une approche basée sur les réseaux de neurones à graphes pour la classification de l'arthrose du genou et de la maladie de Parkinson à partir de vidéos de marche. Dans ce cadre une extraction des points clés a été réalisée, à partir d'une analyse dite globale à 33 points clés et d'une autre dite locale à 8 points clés. Les différentes configurations expérimentées ont été optimisées avec notre approche GS-TPE afin d'ajuster efficacement les hyperparamètres.

Dans la conclusion, nous rappelons nos motivations, les choix retenus et nous soulignons les résultats obtenus. Nous discutons aussi notre apport et nous présentons les perspectives générées par ce travail.

Chapitre 1

Revue de la littérature sur l'optimisation des hyperparamètres des réseaux de neurones profonds

1 Introduction

L'intelligence artificielle (IA) a connu une expansion remarquable ces dernières années et a captivé l'attention des chercheurs, des ingénieurs et du grand public. Au cœur de cette vague transformatrice se trouvent deux domaines essentiels de l'IA : l'apprentissage automatique et l'apprentissage profond. Ces domaines ont dévoilé des perspectives passionnantes et prometteuses pour résoudre une multitude de problèmes complexes dans plusieurs applications telles que l'analyse du signal et la vision par ordinateur.

Une des étapes clés de la conception d'un modèle de Machine Learning ou Deep Learning est le choix des hyperparamètres, dont les valeurs doivent être fixées avant le processus d'apprentissage, et ne changent pas au cours d'un entraînement du modèle. Ces hyperparamètres sont nombreux et de diverses natures, par exemple (i) les hyperparamètres de couches (Nombre et taille des filtres, valeur de dropout, fonctions d'activation, ...), (ii) les hyperparamètres d'optimisation du modèle (optimizer, loss, Learning rate, ...) et (iii) les hyperparamètres d'entraînement du modèle (batch size, nombre d'epochs, ...). L'ajustement des valeurs des hyperparamètres est une tâche redoutable, gourmande en ressources et nécessitant l'expérimentation de nombreuses combinaisons pour obtenir des performances optimales. Eu égard à leur rôle fondamental dans la configuration et l'amélioration des performances des réseaux de neurones profonds, l'optimisation automatique des hyperparamètres est donc un facteur clé de performance de ces modèles.

Afin de mieux comprendre le fonctionnement des réseaux profonds et la nécessité de l'optimisation de leurs hyperparamètres, nous commençons dans ce premier chapitre par une exploration des concepts essentiels sous-jacents à l'apprentissage automatique, et plus particulièrement, à l'apprentissage profond (DL). Ensuite, nous présentons une revue de la littérature des méthodes d'optimisation des hyperparamètres des DNN.

2 Apprentissage automatique

L'apprentissage automatique est une approche de modélisation qui se fonde sur les données. Son principe fondamental consiste à découvrir des motifs et des relations dans un ensemble de données afin d'apprendre comment fonctionne le phénomène sous-jacent. Une fois entraîné sur ces données, le modèle obtenu peut être généralisé pour résoudre des problèmes similaires sur de nouvelles données. Contrairement aux méthodes traditionnelles basées sur des équations mathématiques ou des lois physiques, l'apprentissage automatique est utilisé lorsque les problèmes sont complexes et ne peuvent pas être facilement formalisés par des modèles analytiques. Au lieu de programmer explicitement les règles, la machine apprend sur un grand nombre d'exemples et de caractéristiques pertinentes, et les modèles créés permettront de prédire des résultats à partir de nouveaux exemples

3 Types d'apprentissage

Selon la nature des données et le résultat attendu, nous pouvons distinguer 4 types d'apprentissage.

3.1 Apprentissage supervisé

Connu aussi sous le nom d'apprentissage associatif, celui-ci implique de présenter des paires (entrée, réponse attendue) au réseau. Les paramètres du modèle seront modifiés à chaque pas d'apprentissage suivant un algorithme précis afin de diminuer une fonction d'erreur.

3.2 Apprentissage non-supervisé

On parle aussi d'auto-organisation, où seuls les vecteurs d'entrée sont fournis au réseau. Au cours du processus d'apprentissage, les vecteurs sont regroupés en classes en extrayant des propriétés statistiques à partir de l'ensemble des vecteurs d'apprentissage. A l'opposé du cas supervisé, aucun ensemble préalable de classes n'est utilisé pour la classification des modèles. Au lieu de cela, le système est conduit à construire sa propre représentation des données d'entrée. Les algorithmes connexionnistes connus pour ce genre d'apprentissage sont ceux de Kohonen et de Hopfield [4].

3.3 Apprentissage semi-supervisé

L'apprentissage semi-supervisé consiste à entraîner un modèle d'apprentissage sur un jeu de données partiellement annoté. L'idée est d'attribuer les annotations en utilisant la similarité entre les données annotées et les données non annotées. Il se situe ainsi entre l'apprentissage supervisé qui n'utilise que des données annotées et l'apprentissage non supervisé qui n'emploie que des données non annotées. La combinaison de ces deux ensembles de données permet d'améliorer les

résultats sans avoir recours à l'intervention fastidieuse (coûteuse et chronophage) de l'annotation manuelle.

3.4 Apprentissage par renforcement

L'apprentissage par renforcement forme une discipline propre de l'intelligence artificielle centrée sur l'autonomie des agents dans leur processus d'apprentissage. Dans ce cadre, un agent interagit avec un environnement dynamique, prenant des décisions et recevant des récompenses ou des sanctions en retour. Distinct des approches supervisées ou non supervisées, l'apprentissage par renforcement se base sur le principe de l'essai et de l'erreur, où l'agent ajuste continuellement ses stratégies pour maximiser la somme des récompenses cumulées au fil du temps. Le processus itératif implique l'observation de l'environnement, la prise d'action, la réception de récompense, et l'adaptation des stratégies, conduisant à l'émergence de politiques optimales. Ce paradigme dynamique offre une flexibilité essentielle, éliminant le besoin d'ensembles de données étiquetés et permettant à l'agent de s'adapter de manière autonome à des contextes complexes et changeants.

4 Réseaux de neurones artificiels

4.1 Principe de base

Un réseau de neurones artificiels peut être décrit comme un système composé d'au moins deux couches de neurones, une couche d'entrée et une couche de sortie, et comportant généralement des couches intermédiaires ou couches cachées « hidden layers ». Plus le problème à résoudre est complexe, plus le réseau de neurones artificiels doit comporter de couches. Chaque couche contient un grand nombre de neurones artificiels spécialisés. Dans les réseaux de neurones artificiels, le neurone est considéré comme l'unité fondamentale de calcul. Le fonctionnement d'un neurone artificiel implique qu'il reçoit des entrées provenant d'autres neurones ou d'une source externe. Ces entrées, notées E_1, E_2, \dots, E_n , sont des valeurs numériques et sont reliées au neurone par le biais de synapses, chaque synapse est associée à un poids w_i qui détermine son influence relative. Pour obtenir l'activation E du neurone, on effectue la somme pondérée des entrées $E_1w_1, E_2w_2, \dots, E_nw_n$, en y ajoutant un terme d'entrée supplémentaire ayant un poids b appelé biais. Cette somme est fournie comme argument d'une fonction d'activation, qui représente l'opération réalisée par le neurone. Cette procédure illustrée dans la Figure 1. 1. La sortie du neurone S, appelée activation de sortie, est donc calculée selon l'équation (1. 1) :

$$y = f(\sum_{i=1}^n w_i E_i + b) \quad (1.1)$$

Où f est la fonction d'activation.

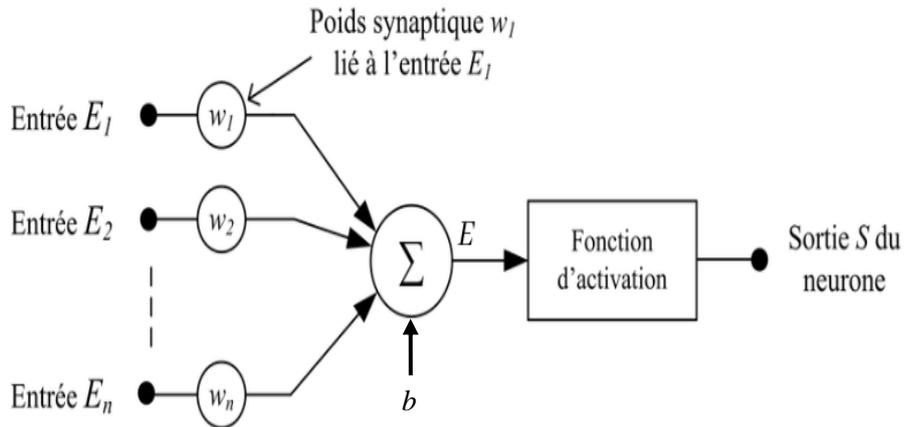


Figure 1. 1 : Un neurone artificiel

4.2 Perceptron multicouches (Multi Layer Perceptron MLP)

Les perceptrons multicouches (MLP) constituent une architecture de réseau de neurones caractérisée par la présence de couches d'entrée, de couches cachées et d'une couche de sortie. L'information circule de manière unidirectionnelle, de la couche d'entrée à la couche de sortie. Chaque neurone utilise une fonction d'activation, pour introduire des non-linéarités. Les MLP possèdent une grande capacité de calcul, grâce à leurs architectures multicouches et à la flexibilité introduite par les poids et les fonctions d'activation (Figure 1. 2).

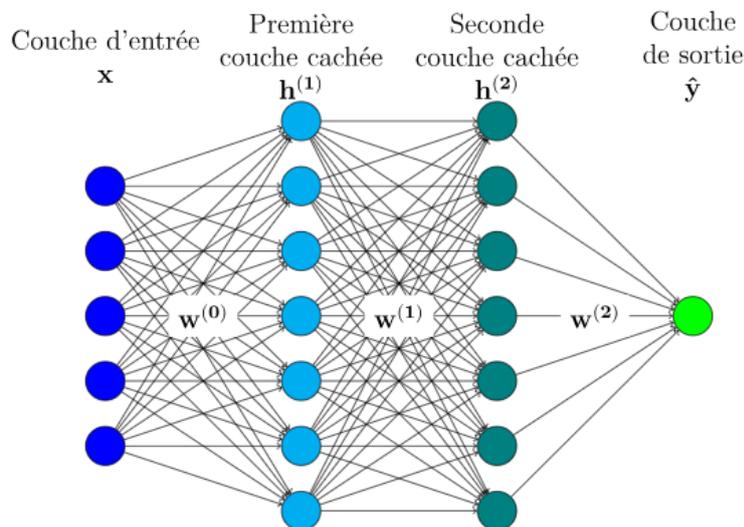


Figure 1. 2 : Architecture de perceptron multicouches

La représentation graphique ci-dessus correspond aux équations suivantes :

- Pour la couche de sortie :

$$y = \varphi_{out}(\sum_i w_i^{(2)} h_i^{(2)} + b^{(2)}) \quad (1.2)$$

- Pour la couche cachée :

$$\forall i, h_i^{(2)} = \varphi(\sum_j w_{ij}^{(1)} h_j^{(1)} + b_i^{(1)}) \quad (1.3)$$

- Pour la couche d'entrée :

$$\forall i, h_i^{(1)} = \varphi(\sum_j w_{ij}^{(0)} x_j + b_i^{(0)}) \quad (1.4)$$

- i représente l'indice des neurones de la couche courante.
- j représente l'indice des neurones de la couche précédente.

Ainsi, dans l'équation (1.2), i parcourt les neurones de la couche de sortie, tandis que dans l'équation (1.3), i parcourt les neurones de la seconde couche cachée et j parcourt les neurones de la première couche cachée. De même, dans l'équation (1.4), i parcourt les neurones de la première couche cachée et j parcourt les différentes entrées x_j .

Pour des raisons de lisibilité, les termes de biais $b_i^{(l)}$ ne sont pas représentés dans la Figure 1. 2.

En général, les neurones du perceptron multicouche possèdent une fonction d'activation non linéaire (au moins dans une des couches) (Figure 1. 3). Les choix classiques pour cette fonction sont :

- La fonction tangente hyperbolique

C'est une fonction d'activation qui produit des valeurs dans l'intervalle $[-1,1]$. Elle est similaire à la fonction sigmoïde mais a une sortie centrée sur zéro, ce qui peut faciliter l'apprentissage dans certaines situations.

$$- \quad \text{Tanh}(x) = \frac{2}{1+e^{-2x}} - 1$$

- La fonction sigmoïde

La fonction sigmoïde produit des valeurs dans l'intervalle $[0,1]$. Elle est couramment utilisée dans la couche de sortie d'un réseau de neurones pour des tâches de classification binaire, où la sortie doit être interprétée comme une probabilité.

$$- \quad \text{Sigmoid}(x) = \frac{1}{1+e^{-x}}$$

- La fonction ReLU (Rectified Linear Unit)

La fonction ReLU est définie comme la fonction identité pour les valeurs d'entrée positives, et comme zéro pour les valeurs d'entrée négatives. Elle est également non linéaire, mais à valeurs

non bornées, ce qui permet de résoudre certains problèmes tels que l'évanouissement du gradient (gradient vanishing) lors de la rétropropagation dans les réseaux très profonds.

$$\text{ReLU}(x) = \begin{cases} x & \text{si } x > 0 \\ 0 & \text{Sinon} \end{cases}$$

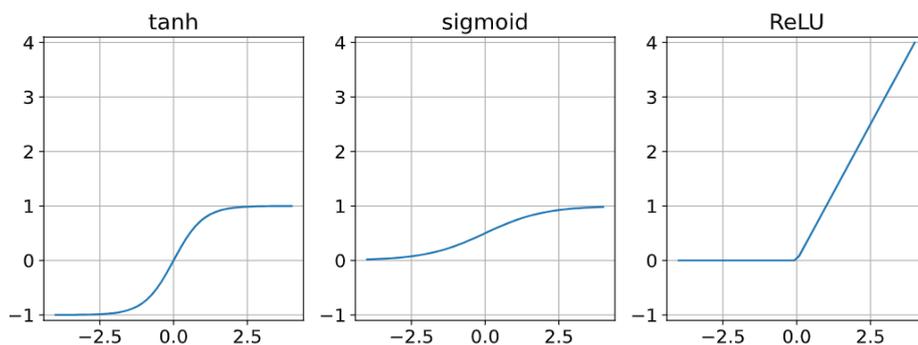


Figure 1. 3 : Fonctions d'activation

4.3 Auto-encodeurs

L'auto-encodeur (AE) fait partie de la famille des réseaux de neurones artificiels. C'est un modèle d'apprentissage auto-supervisé. Les auto-encodeurs ont été utilisés dans plusieurs tâches dont le débruitage. Ils sont conçus pour acquérir la capacité de compresser les données d'entrée dans un espace latent de faible dimension, et de reconstruire ensuite les données initiales à partir de cette représentation latente. L'objectif de l'AE est de minimiser la dissimilarité entre les données d'entrée et leur contrepartie reconstruite.

D'un point de vue architectural, la couche d'entrée et la couche de sortie d'un AE sont constituées par le même nombre de neurones et ce dans l'objectif de reconstruire une sortie la plus proche possible de l'entrée. Les couches cachées d'un AE sont plus petites que la couche d'entrée. Ce type d'architecture « Bottleneck » confère aux AE la capacité de compresser les données d'entrée.

Ainsi, ce réseau comprend deux composants : l'encodeur et le décodeur. Le premier réalise une transformation non linéaire des données d'entrée visant à réduire la dimensionnalité des données d'entrée, pour aboutir à la génération du code latent souhaité. Ses poids sont ajustés lors de l'entraînement pour minimiser la différence entre les données d'entrée et leur reconstruction, ce qui revient à maximiser la capacité du code latent à capturer les informations importantes des données d'entrée. Une fois l'apprentissage terminé, l'encodeur seul peut être utilisé pour compresser de nouvelles données d'entrée dans une représentation latente à faible dimension. Cette représentation peut ensuite être utilisée pour diverses tâches telles que la classification ou la détection d'anomalies (Figure 1. 4).

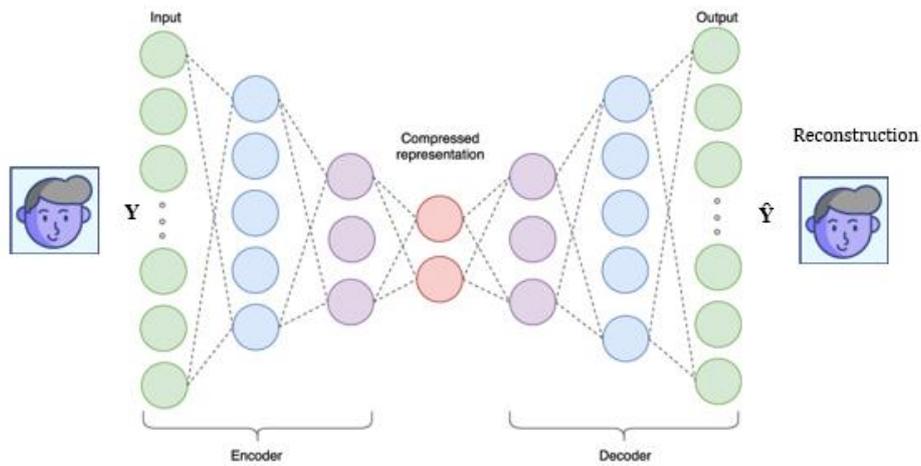


Figure 1. 4 : Auto-encodeur

4.4 Réseaux de neurones récurrents (RNN)

Les réseaux de neurones récurrents (RNN) utilisent généralement des données séquentielles ou des séries temporelles en entrée. Ils sont particulièrement efficaces pour traiter les problèmes impliquant des données ordonnées dans le temps. À l'instar des réseaux neuronaux feed-forward et des réseaux neuronaux convolutifs (CNN), les RNN apprennent également à partir de données d'entraînement. Toutefois, ils se distinguent par leur capacité à intégrer la "mémoire" en tirant parti des entrées passées pour influencer les entrées et sorties courantes. Contrairement aux réseaux neuronaux classiques, qui traitent les entrées et les sorties comme des entités indépendantes, la sortie d'un RNN est influencée par les éléments précédents de la séquence. Les réseaux de neurones récurrents utilisent des boucles de rétroaction, permettant aux signaux de circuler dans les deux sens, ce qui leur confère une capacité de mémorisation (comme le montre la Figure 1. 5) et donc ils sont bien adaptés au traitement de données séquentielles telles que les séries temporelles, car ils prennent en compte les informations contextuelles des étapes temporelles précédentes lorsqu'ils font des prédictions.

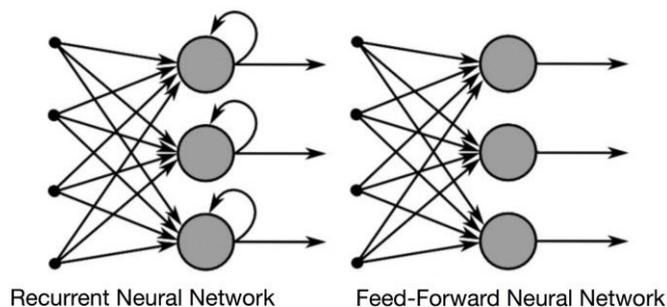


Figure 1. 5: Comparaison entre un RNN (à gauche) et un réseau classique (à droite)

4.5 Réseaux long-short term memory (LSTM)

Les RNN présentent l'avantage de prendre en compte le contexte antérieur lors du traitement d'informations séquentielles. Cependant, ils rencontrent des difficultés lorsqu'ils traitent des séquences longues, typiquement celles qui comprennent plus de 10 événements. En effet, avec des calculs cumulés sur le long terme, l'erreur obtenue avec la rétropropagation du gradient décroît ou, à l'inverse, augmente exagérément au cours du temps. Ces deux problèmes sont nommés "dissipation du gradient" (vanishing gradient) et "explosion du gradient" (exploding gradient) respectivement. Pour pouvoir représenter efficacement les relations de dépendance à plus long terme, les modèles de réseaux de neurones récurrents doivent être capables de maintenir un état sur de longues périodes. Ce rôle est précisément celui des cellules de mémoire à long terme (LSTM), qui disposent d'une mémoire intégrée appelée "cellule". Celle-ci permet de maintenir un état durablement au besoin. Il s'agit d'une valeur codée sur laquelle le réseau peut agir en fonction de différents scénarios. Comme le montre la Figure 1. 6, le flux de données passe par trois portes de contrôle, qui peuvent être comparées à des vannes :

1. La porte d'entrée joue un rôle essentiel pour déterminer si des ajustements doivent être apportés au contenu des cellules.
2. La porte d'oubli sert à décider si le contenu de la cellule doit être remis à zéro.
3. La porte de sortie détermine l'influence de la cellule sur le résultat final du neurone.

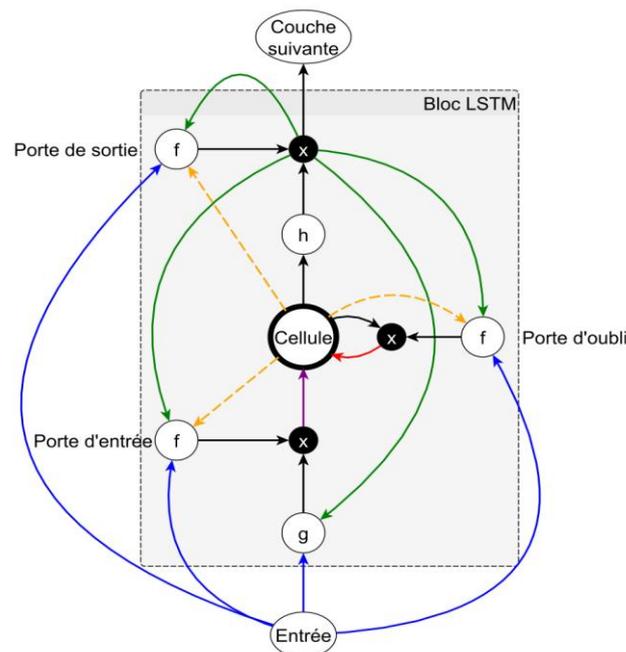


Figure 1. 6: Cellule LSTM

Le fonctionnement de ces portes est représenté mathématiquement par une fonction f , qui prend généralement la forme d'une fonction sigmoïde. Celle-ci est utilisée pour pondérer la somme des

entrées (en bleu), des sorties (en vert) et de la cellule (en orange), par des poids W_{ab} spécifiques à chaque connexion entre les signaux a et b .

Pour x_i et z_h respectivement les entrées et les sorties de la cellule ; S_c représente la valeur de la cellule et l, w, c sont les indices qui décrivent les signaux provenant de la porte d'entrée, de l'oubli et de la cellule. W_{il} est la matrice des poids qui relie les éléments d'entrée à la cellule d'entrée, $W_{h\phi}$ celle des poids associant les sorties vers la porte d'oubli, etc.

Ainsi, les formules qui régissent ces trois portes de contrôle se présentent comme suit : il s'agit d'une somme pondérée suivie d'une activation f , typiquement la sigmoïde :

Porte d'entrée :

$$\begin{aligned} a_i^t &= \sum_{i=1}^l w_{il} x_i^t + \sum_{h=1}^H w_{hl} z_h^{t-1} + \sum_{c=1}^C w_{cl} S_c^{t-1} \\ b_i^t &= f(a_i^t) \end{aligned} \quad (1.5)$$

Porte d'oubli :

$$\begin{aligned} a_\phi^t &= \sum_{i=1}^l w_{i\phi} x_i^t + \sum_{h=1}^H w_{h\phi} z_h^{t-1} + \sum_{c=1}^C w_{c\phi} S_c^{t-1} \\ b_\phi^t &= f(a_\phi^t) \end{aligned} \quad (1.6)$$

Porte de sortie :

$$\begin{aligned} a_w^t &= \sum_{i=1}^l w_{iw} x_i^t + \sum_{h=1}^H w_{hw} z_h^{t-1} + \sum_{c=1}^C w_{cw} S_c^t \\ b_w^t &= f(a_w^t) \end{aligned} \quad (1.7)$$

Pour la cellule mémoire, on peut mettre à jour son contenu S_c ou le réinitialiser à 0. Dans un premier temps, une somme pondérée des entrées et des sorties est appliquée :

$$a_c^t = \sum_{i=1}^l w_{ic} x_i^t + \sum_{h=1}^H w_{hc} z_h^{t-1} \quad (1.8)$$

On calcule alors à nouveau le contenu de la cellule à un instant t en additionnant les deux termes suivants :

$$S_c^t = b_\phi^t S_c^{t-1} + b_i^t g(a_c^t) \quad (1.9)$$

La valeur précédente de la cellule S_c^{t-1} pouvant être annulée avec la valeur de la porte d'oubli, constitue le premier terme b_ϕ^t .

Le deuxième terme reflète la somme pondérée, déterminée en fonction de la porte d'entrée b_i^t . Pour finir, le résultat du neurone LSTM est basé sur une activation qui a été appliquée à la cellule de mémoire S_c^t , contrôlé par la valeur de la porte de sortie b_w^t :

$$b_c^t = b_w^t h(s_c^t) \quad (1.10)$$

4.6 Réseaux de neurones à convolution

Les réseaux de neurones convolutifs représentent une classe spécifique de réseaux neuronaux conçus pour le traitement des données caractérisés par une topologie régulière telle qu'une grille (**Figure 1. 9**). Ils ont ainsi remporté des succès considérables en analyse de signaux ou d'images. Le terme "réseau de neurones à convolution" provient du fait que ces réseaux utilisent avant l'activation, des opérations linéaires invariantes par translation. Dans les CNN, cette opération remplace la multiplication matricielle traditionnelle dans au moins une couche. Les couches typiques d'un CNN sont les suivantes :

- **Couche de convolution (CONV)**

Un exemple de calculs impliqués dans une opération de convolution 1D est illustré dans la Figure 1. 7. Dans ce dernier, la taille du noyau est de 3, ce qui signifie que les poids (w_1, w_2, w_3) sont partagés par chaque décalage de la couche d'entrée (i_1, i_2, \dots, i_6), et les valeurs de sortie sont c_1, c_2, \dots, c_6 . Les valeurs d'entrée dans la fenêtre du noyau sont multipliées par les poids (w_1, w_2, w_3). Ces valeurs sont ensuite résumées et utilisées comme valeurs de carte de caractéristiques. Dans l'exemple présenté, la caractéristique c_3 est obtenue par $c_3 = w_1 * i_2 + w_2 * i_3 + w_3 * i_4$. Ce calcul est généralement suivi par la fonction d'activation ReLU (Unité linéaire rectifiée), qui force les neurones à retourner des valeurs positives.

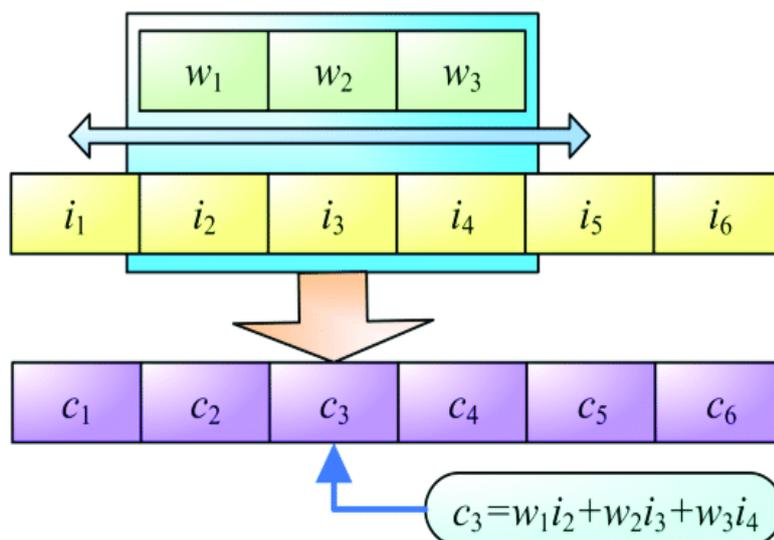


Figure 1. 7 : Opération de convolution

- **Couche de Pooling (Pool)**

Cette couche joue un rôle essentiel dans la compression des données en réduisant la taille des données intermédiaires par un processus de sous-échantillonnage ou regroupement. L'opération de maxpooling est illustrée à la Figure 1. 8. La longueur de la carte des caractéristiques avant le Pooling est de 6, et la taille de Pooling est de 2. Par conséquent, la valeur de sortie est réduite à 3.

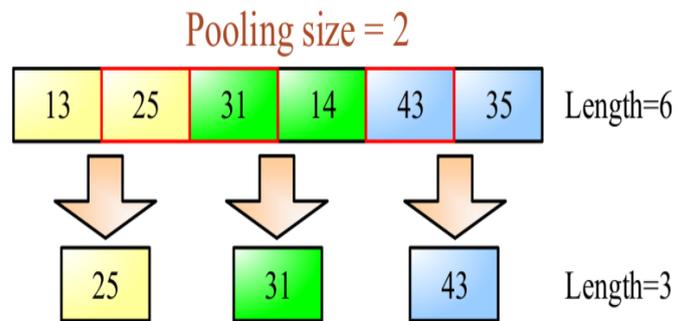


Figure 1. 8: Opération de Max-Pooling

- **Couche "entièrement connectée" (FC)**

Cette couche correspond à un perceptron et se compose de neurones reliés à toutes les sorties de la couche précédente.

- **Couche de normalisation (Softmax)**

Cette "couche" ne fait pas l'objet d'apprentissage (pas de paramètre), mais permet de normaliser la dernière couche apprise du réseau pour obtenir un vecteur de valeurs positives et de somme égale à 1, qui peut donc être interprété comme une distribution de probabilités.

- **Couche de perte (LOSS)**

Cette "couche" correspond à l'application de la fonction de perte qui définit la façon dont l'entraînement du réseau sanctionne les écarts entre le signal anticipé et le signal réel.

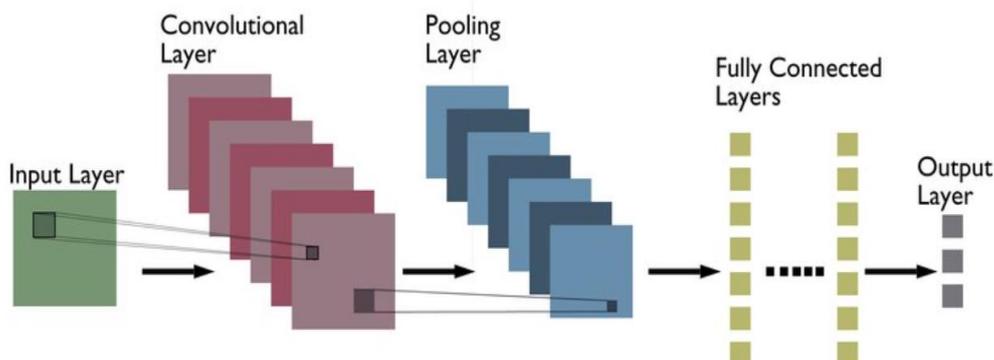


Figure 1. 9: Architecture CNN

5 Optimisation des hyperparamètres

Dans la section précédente, nous avons exploré les réseaux neuronaux profonds, et plus spécifiquement les CNN. Pour atteindre des performances maximales avec ces réseaux, il faut s'intéresser à une facette critique de leur configuration : l'optimisation des hyperparamètres (HP). Cet aspect est crucial pour maximiser l'efficacité et la précision des réseaux de neurones profonds, lors de l'accomplissement de tâches complexes telles que la classification des signaux spatio-temporels, la reconnaissance d'images, etc.

Dans la littérature, plusieurs méthodes ont été proposées pour aborder le problème d'optimisation des hyperparamètres [5] [6] [7]. Dans cette section, nous élaborons une étude comparative de ces méthodes et des principaux outils sous-jacents.

5.1 Hyperparamètres d'un réseau de neurone

Avant d'aborder le problème d'optimisation par hyperparamètres (HP), il convient de rappeler brièvement la différence entre les paramètres et les hyperparamètres. Les premiers font référence aux poids et aux biais associés aux neurones du réseau. Ils sont ajustés et appris au cours du processus d'entraînement à partir des données d'apprentissage. En revanche, les hyperparamètres sont fixés par l'utilisateur avant chaque entraînement et restent constants tout au long de l'exécution. Les réseaux de neurones profonds sont intrinsèquement complexes, car ils comportent des millions de paramètres. La configuration du réseau implique donc des décisions critiques concernant les valeurs d'hyperparamètres de couches (nombre de filtres, fonction d'activation, etc.), d'optimisation (loss, optimiseur, etc.) et d'entraînement (nombre d'itérations, taille du batch) du modèle. Par conséquent, une grande flexibilité est offerte sur la gamme de modèles potentiellement générés. Il est à noter que l'utilisation accrue de réseaux de plus en plus profonds rend la tâche de détermination des hyperparamètres encore plus complexe.

5.2 Problème d'optimisation

5.2.1 Introduction

Un problème d'optimisation est un problème mathématique ou informatique visant à trouver la meilleure solution dans un espace de configurations possibles. Le problème d'optimisation comprend plusieurs éléments clés :

- *La fonction objectif* : Il s'agit de la fonction mathématique à minimiser ou à maximiser, généralement désignée par $f(x)$, où x est un vecteur de variables de décision.
- *Variables de décision* : Ce sont les variables pour lesquelles on recherche les meilleures valeurs de la fonction objectif (les hyperparamètres).

- *Domaine de définition* : C'est l'ensemble des valeurs possibles pour les variables de décision. Il précise les limites dans lesquelles les variables de décision doivent être explorées.
- *Contraintes* : Ce sont des inégalités ou des égalités qui réduisent l'ensemble des valeurs possibles des variables de décision.

Dans notre contexte d'optimisation des hyperparamètres de réseaux de neurones :

- *La fonction objectif* est utilisée pour mesurer les performances du modèle entraîné, pour chaque ensemble d'hyperparamètres. On cherche à maximiser ou minimiser cette fonction selon l'objectif particulier (précision, erreur moyenne, ...).
- *Les variables de décision* sont les hyperparamètres : taux d'apprentissage, nombre de couches, etc.
- *Le domaine de définition* correspond aux plages de valeurs autorisées pour chaque hyperparamètre ; on le spécifie sous forme de bornes d'intervalles et de niveaux de quantification.
- *Les contraintes* peuvent prendre différentes formes, telles que des limitations sur l'utilisation des ressources matérielles. Par exemple, si le GPU dispose de 8 Go de mémoire, il est possible d'imposer une empreinte mémoire limite du modèle, qui impliquera une limitation liant le nombre de poids et la taille des lots (batch size). Pour la contrainte de temps d'entraînement, on peut aussi fixer une limite de temps maximale pour l'entraînement du modèle afin de respecter des contraintes de délais.

5.2.2 Définition et notations

On notera chaque point de l'espace d'hyperparamètres par un vecteur $x \in X$. Chaque vecteur correspond à une configuration particulière du modèle, qui va être évaluée par f , la fonction de coût. Il s'agit donc de trouver le vecteur d'hyperparamètres x_* qui maximise cette fonction (équation(1. 11)) :

$$x_* = \operatorname{argmax}_{x \in X} f(x) \quad (1. 11)$$

5.3 Techniques d'optimisation des hyperparamètres

Les techniques d'optimisation se répartissent principalement en : techniques de recherche systématiques telle que la recherche aléatoire « Random Search » et la recherche par grille « Grid Search », apprentissage par renforcement « Reinforcement Learning », algorithmes de Gradient, algorithmes bayésiens et des algorithmes génétiques. Certaines méthodes se basent sur des

heuristiques telles que Tree Parzen Estimator « TPE » et le Particle Swarm Optimization « PSO ». Nous détaillons dans la suite ces principales catégories.

5.3.1 Méthodes de recherche systématiques

5.3.1.1 Recherche par grille

La recherche par grille (Grid Search « GS») est sans doute la méthode la plus ancienne [8]. Son principe consiste à définir un intervalle discret de valeurs pour chaque hyperparamètre du modèle, la “grille” étant formée par le produit cartésien de ces intervalles. Ensuite, toutes les combinaisons possibles d'hyperparamètres sont testées en entraînant et en évaluant le modèle (**Figure 1. 10**). La principale limitation est le temps de calcul, lorsque la grille est de grande taille, les avantages étant bien entendu la simplicité et le caractère exhaustif (à la quantification près) pour trouver les meilleurs hyperparamètres pour un modèle donné.

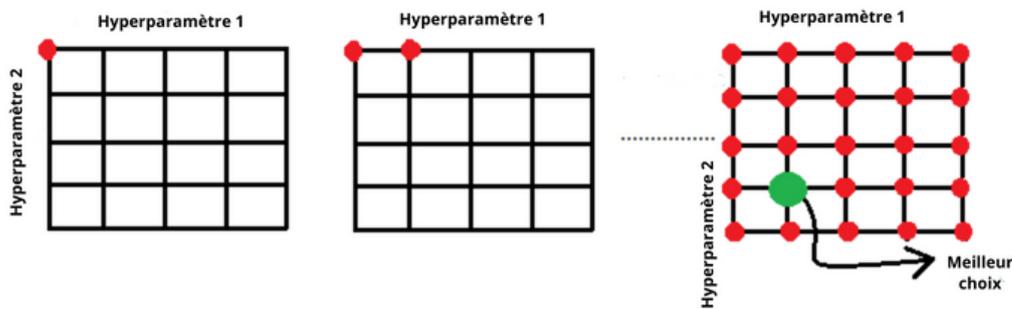


Figure 1. 10: Recherche par grille

L'algorithme ci-dessous décrit est une procédure d'évaluation pour la recherche de grille qui est utilisé pour l'optimisation des hyperparamètres. Il prend en entrée une base de données (D), un algorithme d'apprentissage automatique ou profond (A), l'espace des hyperparamètres à explorer (Θ), et la méthode de recherche par grille (GS) à utiliser. L'objectif est d'obtenir les hyperparamètres optimisés en sortie. Dans cette procédure, un ensemble vide (res) est initialisé pour stocker les résultats de l'évaluation. Ensuite, une boucle principale est exécutée, allant de 1 à N, où N représente le nombre total d'itérations spécifié (ici on parle du nombre de combinaisons des hyperparamètre à tester). À chaque itération, un ensemble d'hyperparamètres est sélectionné à partir de l'espace des hyperparamètres Θ en utilisant la méthode GS et l'algorithme A. Ensuite, le modèle est entraîné avec ces hyperparamètres sur l'ensemble d'entraînement. Par la suite, le modèle est évalué sur l'ensemble de test et le résultat de l'évaluation est enregistré dans res. Une fois que toutes les

itérations sont terminées, l'espace des hyperparamètres Θ est ajusté en fonction des résultats obtenus dans res [9] .

```

Algorithm 1: Evaluation Procedure of Grid search
Optimization


---


Input: Dataset (D); Algorithm (A);  $\Theta$  (hyper parameter
      Space), GS (Grid Search)
Output: Optimized Parameters
res  $\leftarrow$  { };
for i to N do
     $\theta \leftarrow$  select hyperparameter(GS, A,  $\Theta$ )
    Model  $\leftarrow$  train(A, $\theta$ ,D train)
    res  $\leftarrow$  eval(Model,D test)
end
 $\Theta \leftarrow$  Ad just( $\Theta$ , eval(res))

```

Figure 1. 11: Algorithme de la recherche par grille [9]

5.3.1.2 Recherche aléatoire

La recherche aléatoire ou "Random Search (RS)" [7], consiste à sélectionner aléatoirement des configurations de paramètres à tester parmi un ensemble donné, sans suivre de schéma préétabli (Figure 1. 12) afin de trouver la meilleure combinaison pour maximiser les performances d'un modèle.

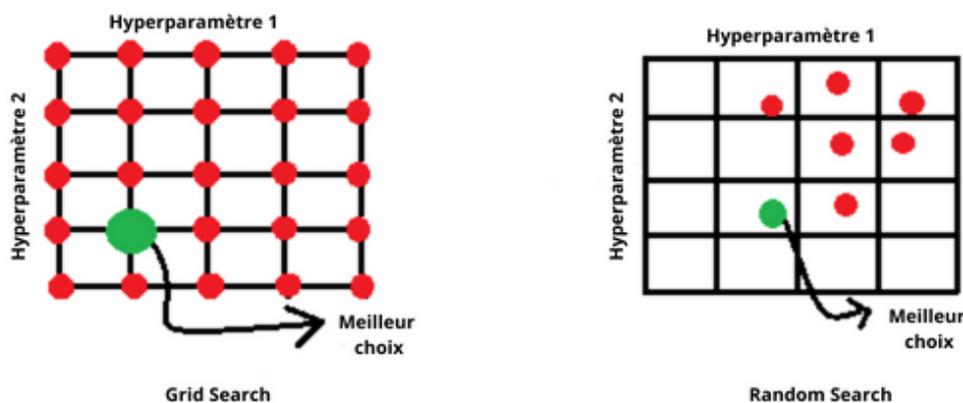


Figure 1. 12 : Recherche aléatoire

L'utilisation conjointe de ces deux algorithmes peut être considérée comme un socle pour l'optimisation des hyperparamètres (HP) des réseaux profonds. Néanmoins leur simplicité limite leur capacité à exploiter efficacement les évaluations antérieures pour guider l'optimisation.

5.3.2 Optimisation bayésienne (BO)

L'optimisation bayésienne « Bayesian Optimisation (BO) » est un concept général conçu pour des fonctions "boîte noire". Son objectif principal consiste à recueillir des informations détaillées

sur la fonction, en se concentrant en particulier sur la localisation du point optimal. L'approche vise à trouver un équilibre entre l'*exploration* des régions d'hyperparamètres aux résultats incertains, et l'*exploitation*, ciblant les régions les plus prometteuses. Dans cette section, nous présentons deux méthodes bayésiennes : les processus gaussiens (GP) et les algorithmes d'estimation de Parzen (TPE).

5.3.2.1 Processus gaussiens

Les processus gaussiens (GP) [10], sont fondés sur l'estimation de distributions gaussiennes multivariées. Ils utilisent une fonction pour estimer la moyenne (μ) et une autre pour estimer la covariance (Σ) du vecteur d'hyperparamètre optimal x^* . Dans le processus de métamodèle (MP), le modèle est constamment mis à jour après chaque évaluation, ce qui permet au MP de créer un modèle plus précis. Ce dernier guide ensuite les choix d'évaluation pour les configurations ultérieures. Au lieu d'utiliser simplement des prévisions de moyenne et de variance, les experts privilégient souvent des fonctions d'acquisition telles que la probabilité d'amélioration (PI) ou l'amélioration attendue (EI). Ces fonctions optimisent directement les modèles construits par les MP. Le processus d'optimisation associe les estimations du processus gaussien à la fonction d'acquisition (EI). À chaque itération, la fonction d'acquisition est améliorée afin de sélectionner le candidat ayant le plus grand potentiel pour améliorer la performance ciblée. Une fois le candidat évalué, les fonctions GP et PI/EI sont mises à jour en fonction des nouvelles données, ce qui déclenche un nouveau cycle d'optimisation (Figure 1. 13).

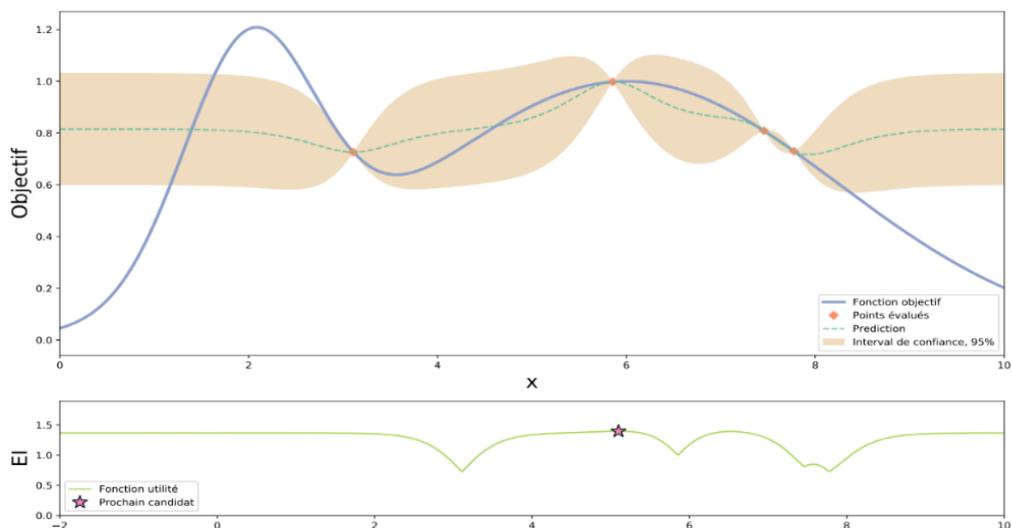


Figure 1. 13: Optimisation monovariante par processus gaussien avec une fonction d'acquisition.

Pour le HPO, les GP sont fréquemment utilisés dans les cas d'espaces de HP de faible dimension, montrant des performances supérieures aux configurations manuelles faites par les experts [11] [12].

5.3.2.2 Arbres Estimateurs de Parzen (TPE)

Les TPE adoptent une approche différente de l'optimisation des HP : ils divisent les points de l'espace des HP en deux catégories distinctes lors de l'optimisation, en supposant que ces catégories proviennent de deux distributions différentes. La première catégorie comprend les modèles les plus prometteurs observés jusqu'à présent, la deuxième catégorie comprend tous les autres modèles. L'objectif de l'algorithme TPE est de simuler les distributions de ces deux types de sous-ensembles d'HP, afin de guider le choix du prochain ensemble de points d'évaluation. Les principaux enjeux tournent autour de la répartition des meilleures configurations. L'idée de base est qu'en se concentrant sur les meilleures configurations, TPE peut explorer l'espace des hyperparamètres plus efficacement en choisissant des valeurs ayant un potentiel plus élevé pour améliorer les performances du modèle. La Figure 1. 14 montre comment utiliser cette stratégie.

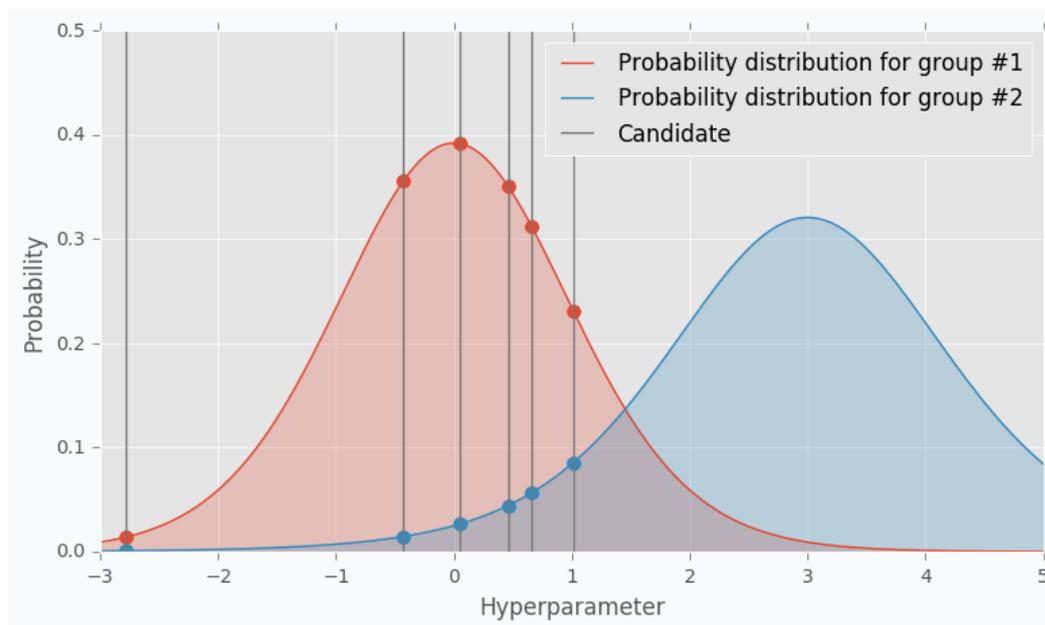


Figure 1. 14: Exemple de distribution (monovariante) modélisée par TPE

Détaillons à présent le fonctionnement du TPE pour la maximisation monoobjective dans l'algorithme 1 [13].

Soit X un espace de recherche des hyperparamètres et $f: X \rightarrow \mathbb{R}$ une fonction objective. Nous considérons que le problème d'optimisation consiste à trouver $x^* \in \operatorname{argmax}_{x \in f(x)}$. Prenons un ensemble d'observations $D = \{(x^{(1)}, y^{(1)}), \dots, (x^{(k)}, y^{(k)})\}$ trouvées après quelques essais initiaux,

généralement sélectionnés de manière aléatoire. TPE modélise $p(x_i|y)$ pour chaque hyperparamètre x_i ($\in X_i$) à l'aide des deux fonctions de densité de probabilité suivantes (équation (1.12)) :

$$P(x_i|y) = \begin{cases} l(x_i) & \text{si } y < y^* \\ g(x_i) & \text{si } y \geq y^* \end{cases} \quad (1.12)$$

Où $l(x_i)$ est construit en utilisant le sous-ensemble des valeurs x_i observées $\{x_i^{(j)} \in X_i | y^{(j)} (= f(x^{(j)})) \geq y^*, j = 1, \dots, k\}$ et $g(x_i)$ est construit en utilisant l'ensemble des valeurs x_i observées restantes. La valeur y^* est choisie comme étant un quantile $\gamma \in (0,1)$ de la valeur y des observations qui satisfait $p(y < y^*) = \gamma$.

Dans les lignes 2-3 de l'algorithme 1, les observations sont divisées en D_l et D_g pour construire les modèles. Cette opération peut être facilement réalisée en triant les observations sur la base des valeurs y . En bref, $l(x_i)$ modélise la densité des "bonnes" valeurs x_i , tandis que $g(x_i)$ modélise la densité des "mauvaises" solutions des x_i . Ces densités sont ensuite estimées à partir des données observées.

Pour obtenir un candidat à évaluer, TPE utilise l'amélioration attendue « Expected Improvement » (EI) suivante comme fonction d'acquisition pour y^* et l'hyperparamètre x_i comme fonction d'acquisition.

$$EI_{y^*}(x_i) = \int_{-\infty}^{y^*} (y^* - y) p(y|x_i) dy = \int_{-\infty}^{y^*} (y^* - y) \frac{p(x_i|y)p(y)}{p(x)} dy \quad (1.13)$$

Par construction, $\gamma = p(y < y^*)$, et $p(x_i) = \int_{-\infty}^{+\infty} p(x_i|y)p(y) dy = \gamma l(x_i) + (1 - \gamma)g(x_i)$

$$\int_{-\infty}^{y^*} (y^* - y) p(x_i|y)p(y)dy = l(x_i) \int_{-\infty}^{y^*} (y^* - y)p(y)dy = \gamma y^* l(x_i) - l(x_i) \int_{-\infty}^{y^*} p(y)dy \quad (1.14)$$

$$EI_{y^*}(x_i) = \frac{\gamma y^* l(x_i) - l(x_i) \int_{-\infty}^{y^*} p(y)dy}{\gamma l(x_i) + (1 - \gamma)g(x_i)} \propto \left(\gamma + \frac{g(x_i)}{l(x_i)}(1 - \gamma) \right)^{-1} \quad (1.15)$$

Nous visons à déterminer les "meilleurs" hyperparamètres à tester en maximisant $\frac{g(x_i)}{l(x_i)}$, sélectionnant ainsi ceux qui sont probables sous la distribution "bonne" $l(x_i)$ et peu probables dans la "mauvaise" distribution $g(x_i)$. Ainsi, à chaque itération, pour chaque hyperparamètre actif, le meilleur candidat x_i^* avec la plus grande valeur d'EI est sélectionné pour être évalué (lignes 2-9 de l'algorithme 1) jusqu'à un nombre prédéterminé d'itérations.

La méthode TPE est souvent intégrée en tant qu'heuristique dans les bibliothèques d'optimisation des HP pour les DNN [14] [15], de par sa meilleure adaptation aux problèmes de grande dimension, comparée aux GPs.

Algorithm 1: Tree-structured Parzen Estimator

Require:
 $D = \{(x^{(1)}, y^{(1)}), \dots, (x^{(k)}, y^{(k)})\}$: Observations
 $n_t \in \mathbb{N}$: number of iterations
 $n_c \in \mathbb{N}$: number of candidates
 $\gamma \in (0, 1)$: quantile

- 1: **for** $t \leftarrow 1, \dots, n_t$ **do**
- 2: $D_l \leftarrow \{(x, y) \in D \mid y \text{ is included in the best } [\gamma|D|] \text{ objective values in } D\}$
- 3: $D_g \leftarrow D \setminus D_l$
- 4: **Repeat**
- 5: $i \leftarrow i \in [1, n]$ such that x_i is active and x_i^* has not been sampled
- 6: construct $l(x_i)$ with $\{x_i \mid (x, y) \in D_l\}$ and $g(x_i)$ with $\{x_i \mid (x, y) \in D_g\}$
- 7: $C_i \leftarrow \{x_i^{(j)} \sim l(x_i) \mid j = 1, \dots, n_c\}$ \rightarrow Sample n_c candidates for x_i^*
- 8: $x_i^* \leftarrow \operatorname{argmax}_{x_i \in C_i} l(x_i) / g(x_i)$ \rightarrow Select the best candidate
- 9: **Until** all active hyperparameters have been sampled
- 10: $D \leftarrow D \cup \{(x^*, f(x^*))\}$ $\rightarrow x^*$ is the vector composed of all sampled x_i^*
- 11: **end for**
- 12: **return** x with the maximum y value in D

Figure 1. 15 : Pseudo-code de TPE [13]

5.3.3 Optimisation par essaim de particules (PSO)

La PSO a été proposée par Eberhart et Kennedy en 1995 [16]. Plusieurs chercheurs ont appliqué la PSO dans un contexte d'optimisation [17] [18] [19]. Dans cette méthode, une particule (un individu) représente un point *qui se déplace* dans l'espace des hyperparamètres, et voit donc son aptitude (fitness, liée à la fonction de coût correspondante) changer au cours des itérations en fonction de diverses politiques de déplacement de l'ensemble de la population (essaim), selon divers mécanismes collectifs et individuels. Le processus est le suivant :

- i. Initialiser une population aléatoirement (positions et vitesses).
- ii. Evaluation de l'aptitude des particules de la population.
- iii. Actualisation de la position et de la vitesse pour chaque particule tout en recherchant son optimum local. La particule la plus apte *pbest* antérieure est remplacée par la nouvelle plus apte de l'essaim. Le processus de sélection se poursuit jusqu'à convergence, ou lorsqu'un critère de terminaison est satisfait.

5.3.4 Algorithme génétique (GA)

Les algorithmes génétiques [20] figurent dans la catégorie des méthodes dites "boîte noire". Ce sont des algorithmes itératifs qui créent et maintiennent un ensemble de configurations d'hyperparamètres, afin d'améliorer progressivement la qualité de cette population. En général, une

itération d'un algorithme génétique se compose de 4 phases : la sélection, le croisement, la mutation et la mise à jour (Figure 1. 16).

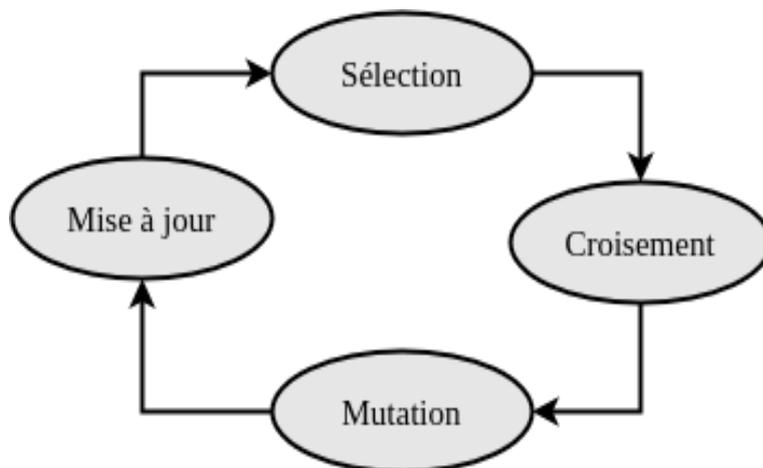


Figure 1. 16: Les étapes génériques d'un algorithme génétique

L'étape de *sélection* consiste à identifier les éléments les plus performants dans la population. Dans notre cas ce sont les configurations de réseaux de neurones ayant obtenu les scores de validation les plus élevés, qui sont désignés comme les "parents". Ces parents sont ensuite soumis à un processus de *croisement*, qui constitue la deuxième étape. L'objectif est ici d'amalgamer certaines caractéristiques de chaque parent pour générer une nouvelle progéniture, héritant possiblement des caractéristiques favorables des différentes configurations de la génération précédente. L'étape de *mutation* introduit des perturbations aléatoires dans les configurations issues des croisements. Ce caractère aléatoire a deux objectifs : éviter la convergence précoce vers un optimum local et ajouter des variations qui n'existaient pas dans la population initiale. Enfin, après avoir évalué la nouvelle génération de descendants, dans l'étape de *mise à jour*, les configurations les moins performantes de la population actuelle sont éliminées et remplacées par les descendants les plus performants. Ce cycle continu garantit une amélioration progressive de la qualité globale au fil du temps.

5.4 Outils d'optimisation

Dans cette section, nous présentons brièvement un certain nombre de bibliothèques ou Framework existants d'optimisation des hyperparamètres.

5.4.1 SKlearn

Dans SKlearn [21], "GridSearchCV" peut être mis en œuvre pour identifier les hyperparamètres optimaux grâce à l'algorithme GS. Le programme évalue chaque valeur d'HP dans l'espace de configuration définie manuellement, et évalue sa performance par validation croisée.

RandomizedSearchCV est également disponible dans SKlearn pour appliquer la méthode RS. Celle-ci évalue en parallèle un nombre prédéfini de valeurs d'HP sélectionnées de manière aléatoire. Une validation croisée est menée pour évaluer efficacement les performances de chaque configuration.

5.4.2 HyperOpt

HyperOpt [14] est un framework d'optimisation des hyperparamètres qui utilise RS et BO-TPE en tant qu'algorithmes d'optimisation. Contrairement à d'autres bibliothèques qui ne supportent qu'un seul modèle, HyperOpt permet d'utiliser plusieurs modèles pour modéliser des hyperparamètres hiérarchiques. HyperOpt SKlearn [22] et Hyperas sont les deux bibliothèques spécifiques qui permettent d'appliquer HyperOpt à scikit-learn et Keras respectivement.

5.4.3 Optuna

Optuna [23] figure parmi les frameworks les plus connus pour l'optimisation automatique des hyperparamètres. Il dispose d'outils de monitoring intéressants, comme un enregistrement de données historiques, des tracés « trial » afin de déterminer la zone intéressante de l'espace des hyperparamètres à explorer et ainsi, de trouver les valeurs optimales des hyperparamètres dans un délai minimum.

5.4.4 Sherpa

Sherpa [24] est un outil d'optimisation des hyperparamètres destiné à être utilisé pour les modèles d'apprentissage automatique. Il est spécialement conçu pour les problèmes d'évaluation de fonctions itératives et coûteuses en calcul, comme le réglage des HP des DNN. Les concepteurs ont la possibilité d'optimiser facilement les hyperparamètres à l'aide d'une variété d'algorithmes puissants et évolutifs tels que les algorithmes BO-GP et GA.

5.4.5 Nevergrad

Nevergrad [25] est une librairie Python open-source qui contient un large choix d'optimiseurs, comme l'algorithme génétique (GA) et PSO. En apprentissage automatique, Nevergrad peut être utilisé pour régler tous les types d'hyperparamètres, tels que les hyperparamètres discrets, continus et catégoriels, en choisissant différents algorithmes.

5.4.6 Talos

Talos [26] est utilisé pour traiter des modèles complexes des architectures DL et choisir la bonne configuration des hyperparamètres. Il fonctionne de la même manière que GridSearchCV. Il teste toutes les combinaisons possibles des paramètres introduits et choisit le modèle le plus approprié, en fonction du critère à optimiser.

Dans le Tableau 1. 1, nous récapitulons les divers algorithmes d'optimisation disponibles dans les frameworks mentionnés, et soulignons leurs points forts.

Tableau 1. 1 : Tableau comparatif des frameworks d'optimisation d'HP pour les réseaux de neurones

<i>Outils d'optimisation</i>	<i>Facilité d'utilisation</i>	<i>Algorithmes utilisés</i>				<i>Points forts des algorithmes</i>
		<i>RS</i>	<i>GS</i>	<i>BO-TPE</i>	<i>BO-GP</i>	
<i>SKlearn</i>	+++	X	X			<i>GS</i> : simple <i>RS</i> : plus efficace que GS en termes de vitesse d'exploration de l'espace des HP. <i>BO-TPE</i> : performant avec tous types des HP. <i>BO-GP</i> : performant avec tout type de HP.
<i>Optuna</i>	++-	X	X	X		
<i>Hyperopt</i>	++	X		X		
<i>Sherpa</i>	+-	X			X	
<i>Talos</i>	+-	X	X			

6 Étude bibliographique sur l'optimisation des hyperparamètres pour les DNN

Plusieurs travaux dans la littérature ont été proposés pour tenter d'optimiser les HP d'un DNN. On distingue plusieurs méthodes d'optimisation dont les performances dépendent de plusieurs HP. Selon le contexte, ces méthodes peuvent être utilisées de manière isolée ou combinée pour trouver la meilleure configuration du réseau. Lorsque l'objectif est de simplifier le processus de définition des hyperparamètres, une première démarche consiste à utiliser les méthodes d'optimisation individuellement pour effectuer une recherche ciblée dans l'espace des hyperparamètres. Cette méthode donne des résultats satisfaisants, mais elle nécessite plus d'essais pour atteindre la configuration optimale. La seconde démarche préconise de combiner de différentes méthodes d'optimisation. En utilisant une approche hybride, on peut bénéficier des points forts de chaque méthode et ainsi compenser leurs limitations.

6.1 Méthodes de recherche exhaustive

Dans [27], les auteurs proposent une méthode de détection des maladies cardiaques basée sur des réseaux de neurones avec réglage des hyperparamètres. Ce dernier est effectué en utilisant la recherche par grille, la recherche aléatoire et l'optimisation bayésienne séparément. Les hyperparamètres ciblés sont les suivants : le nombre de couches cachées, le nombre d'unités cachées, le taux de dropout, le nombre d'époques et la taille du lot. En termes de temps de calcul, la recherche aléatoire nécessite moins de temps que l'optimisation bayésienne et la recherche par grille. Dans le cas plus précis de classification, l'optimisation bayésienne produit une précision plus élevée que celle des recherches par grille et aléatoire. Les résultats obtenus avec l'optimisation bayésienne donnent une exactitude de 91,67%, une sensibilité de 95,83%, une spécificité de 88,89%,

une précision de 85,19% et un score F1 de 90,20%. Ces résultats révèlent, que dans le cas du DNN, l'adoption de l'optimisation bayésienne est préférable pour la détection des maladies cardiaques.

6.2 Algorithmes génétiques-évolutionnaires

Les algorithmes génétiques [28] ont été beaucoup utilisés pour l'optimisation des HP. Par exemple dans [29], un système pour l'optimisation des HP de CNN a été effectuée en utilisant l'algorithme génétique (GA). Les résultats obtenus sont plus performants que ceux de nombreuses architectures CNN modernes. Dans [30], une méthode fondée sur les algorithmes génétiques est utilisée afin d'optimiser les hyperparamètres des réseaux de neurones artificiels en vue de minimiser l'erreur du réseau dans les problèmes liés à l'approximation des signaux. Les hyperparamètres mis en jeu sont : le nombre d'unités cachées dans une seule couche cachée, la fonction de transfert dans la couche cachée et la fonction de transfert à utiliser au niveau de la couche de sortie. Les résultats montrent que cette méthode est prometteuse. Les auteurs dans [31] ont proposé une méthode permettant d'optimiser de manière automatique les hyperparamètres du réseau LSTM en utilisant l'évolution différentielle (DE) dans le contexte de la classification des émotions. Les hyperparamètres à optimiser sont la taille du lot et le nombre de neurones cachés. A partir des valeurs initiales des HP générées de manière aléatoire, l'algorithme DE tente d'améliorer la précision du modèle de classification des émotions de manière itérative jusqu'à ce que le critère d'arrêt soit atteint. Les résultats ont été comparés avec ceux obtenus avec d'autres méthodes d'optimisation d'hyperparamètres (optimisation par essaims de particules, Recuit Simulé, recherche aléatoire.). Les résultats d'expérimentations ont confirmé la capacité de l'algorithme DE qui atteint une précision de 77,68%. Dans [32], les auteurs ont suggéré une nouvelle méthode d'optimisation des réseaux de neurones des perceptrons multicouches nommée « GOAMLP ». Cette dernière a été appliquée sur cinq ensembles de données relatives aux pathologies : Parkinson, diabète, cancer du sein, et aux pathologies coronariennes et orthopédiques. Dans [33], les auteurs ont présenté un travail portant sur l'optimisation des hyperparamètres des CNNs et le développement d'une architecture optimisée pour la classification et la détection des visages. Les hyperparamètres sélectionnés sont la taille du kernel, la fonction d'activation, la taille du filtre, la fonction de perte, le taux d'apprentissage et l'optimiseur. Les résultats ont démontré qu'un taux d'apprentissage plus bas est requis pour avoir une meilleure performance et a révélé aussi que la configuration développée a permis une amélioration remarquable des performances en termes de précision du réseau qui a dépassé 94%.

Les algorithmes évolutionnaires (AE) sont des techniques stochastiques d'optimisation fondées sur une simulation grossière de l'évolution naturelle des populations. Elles s'inspirent du principe de l'évolution des espèces décrit par Darwin [34]. L'algorithme AE exploite des opérations stochastiques de variation sur un "pool" d'individus (population). Chaque individu de la population représente une version codée (solution) du problème. Initialement, cette population est générée de manière aléatoire. À chaque génération/itération de l'algorithme, les solutions sont retenues, appariées et recombinaées pour produire de nouvelles solutions qui remplacent les "moins bonnes" en fonction d'un certain critère, et ainsi de suite. La fonction d'évaluation affecte une valeur de fitness à chaque individu afin de juger de sa pertinence vis-à-vis du problème (critère de sélection). Dans [35], la méthode MENNDL (Multi-Node Evolutionary Neural Networks for Deep Learning) est proposée afin d'automatiser la sélection des réseaux sur les clusters de calcul par une optimisation des hyperparamètres réalisée par des algorithmes génétiques afin de minimiser le temps de convergence. Les hyperparamètres utilisés sont la taille du noyau et le nombre de filtres pour chaque couche de convolution. Cette méthode peut être un outil plus puissant que la recherche aléatoire par sa capacité à utiliser les résultats antérieurs afin de déterminer quelles zones de l'espace des hyperparamètres à parcourir. Dans [36], les auteurs ont mis l'accent sur l'apprentissage d'un réseau de feed-forward par un algorithme évolutif modifié basé sur l'optimisation de la reproduction asexuée (ARO) afin de résoudre les problèmes de classification des données. L'objectif de l'utilisation de cet algorithme est de minimiser la fonction de coût. La précision moyenne mesurée sur l'ensemble des données pour la méthode proposée est de 83,8%, et la performance de la méthode proposée comparée à d'autres algorithmes évolutionnaires, notamment les algorithmes génétiques, le recuit simulé et l'optimisation des essaims de particules, a permis d'obtenir les meilleurs résultats en termes de rapidité et de précision. Dans [37], une optimisation des hyperparamètres avec une stratégie évolutionnaire appelée HOUSES "Hyperparameter Optimization with sURrogate-aSsisted Evolutionary Strategy" a été proposée pour l'optimisation des hyperparamètres d'un réseau convolutif multi-niveaux ML-CNN, dans le cadre de classification de la malignité des nodules pulmonaires. Cette stratégie a été testée sur trois types de réseaux de neurones (MLP, LeNet et AlexNet). Les résultats expérimentaux sur ces différents réseaux ont permis d'obtenir une meilleure configuration des hyperparamètres que ceux réalisées par d'autres approches telles que la recherche par grille, la recherche aléatoire, TPE et l'optimisation bayésienne. Les auteurs dans [38] ont détaillé une approche qui consiste à définir une combinaison optimale d'hyperparamètres dans le but d'améliorer les performances de la détection de la respiration avec un réseau 1D-CNN. Les hyperparamètres à optimiser sont la taille du noyau de chaque couche de convolution, le nombre de noyaux de chaque couche de convolution et le nombre de neurones de chaque couche dense. Une

plage de valeurs spécifique est déterminée pour chaque hyperparamètre puis une combinaison des valeurs de la plage est construite. Les résultats expérimentaux ont permis d'obtenir un taux de reconnaissance de cinq modèles de respiration d'environ 96,7% en moyenne, ce qui représente une amélioration d'environ 2,8% par rapport à une méthode existante. Dans [39], les auteurs ont décrit une méthode de recherche pour trouver le modèle optimal grâce à un réglage des hyperparamètres pour un réseau de neurones en utilisant la méthode uDEAS afin de minimiser les coûts. La méthode d'optimisation des hyperparamètres proposée a été appliquée sur deux modèles de réseaux de neurones : l'auto-encodeur (AE) et le CNN. Les hyperparamètres mis en jeu sont la taille des lots, le nombre de couches cachées, le taux d'apprentissage pour le modèle AE, le taux d'apprentissage, le nombre de filtres, et la première et la deuxième couche de convolution pour le CNN. La recherche par grille a également été appliquée et la méthode proposée a permis de trouver le réglage optimal des hyperparamètres avec un taux de convergence plus rapide et une complexité de calcul inférieure à celle de la recherche aléatoire classique. Afin de montrer l'efficacité de la méthode proposée, les résultats ont été comparés avec d'autres algorithmes d'optimisation étudiés (à savoir, le recuit simulé, l'algorithme génétique et l'algorithme PSO).

6.3 Méthodes méta-heuristiques

Les métaheuristiques désignent généralement les méthodes visant à optimiser l'utilisation des ressources et permettant de trouver une valeur approximative de la solution optimale en un temps limité. Leur objectif est de résoudre un ensemble de problèmes dans différents domaines sans avoir à modifier le principe de base de l'algorithme de la méthode. La méthode d'optimisation approximative PSO est particulièrement intéressante. Elle est basée sur des "interactions sociales" entre des "agents" appelés "particules", dans le but d'atteindre un objectif donné dans un espace de recherche partagé, dans lequel les particules disposent d'une certaine aptitude à stocker et à traiter l'information. L'article [40] décrit la classification des signaux EEG basée sur les réseaux de neurones profonds pour identifier la nature de la crise (épileptique ou normale). La technique d'optimisation PSO a été appliquée afin de régler les hyperparamètres du réseau utilisé. Le classifieur a été évalué avec différents nombre d'itérations et de particules lors de son apprentissage. La précision a atteint 94% avec une base de données publique. Dans [41] l'optimisation par essaims de particules (PSO) a été utilisée pour réduire la complexité globale de l'algorithme. Les résultats expérimentaux ont prouvé l'efficacité de la méthode PSO non seulement à réduire le temps d'apprentissage, mais aussi à augmenter la précision de manière significative.

L'algorithme Efficient Global Optimization (EGO) [42] a été adapté pour optimiser et configurer automatiquement les architectures de réseaux de neurones à convolution afin d'augmenter la précision finale. Dans l'algorithme EGO, le critère de remplissage 'infill' basé sur la fonction génératrice de moment (MGF) est adopté. Ce critère de remplissage permet d'équilibrer explicitement l'exploitation et l'exploration. La Stratégie d'évolution à intégration mixte (MIES) est également utilisée pour maximiser le critère de remplissage. L'approche d'optimisation proposée est capable de générer des réseaux performants avec un nombre limité d'itérations. Le résultat de l'expérience montre que de telles performances du réseau optimisé ont été obtenues avec un très petit nombre d'époques (10 pour le MNIST, et 50 pour le CIFAR-10) durant l'apprentissage. En outre, un taux de dropout de 0,2 est appliqué pour éviter le sur-apprentissage. Une autre approche basée sur les PSO utilisée pour la configuration de l'architecture CNN a été présentée dans [43]. Comme exemple d'hyperparamètres à optimiser, nous pouvons citer la taille des kernels, le padding, le nombre de feature maps, et les types de Pooling. En utilisant l'adaptation PSO, les auteurs ont réussi à obtenir de meilleurs paramètres que ceux trouvés par AlexNet et ont obtenu une haute précision.

Dans [44], la classification de l'épilepsie a été réalisée en deux étapes, tout d'abord une analyse discriminante linéaire bayésienne (BLDA), puis une optimisation et une classification à l'aide de la méthode hybride « Colonie d'abeilles Artificielle - Optimisation des essaims de particules » (ABC-PSO) a été menée. Les résultats ont montré qu'une précision de classification de 94% a été obtenue avec le système BLDA et qu'après une optimisation et une classification supplémentaire, une précision de 97,91% a été atteinte avec le système ABC-PSO, ce qui montre l'efficacité de la méthode utilisée. Dans [45], l'algorithme OLPSO (Orthogonal Learning Particle Swarm Optimization) qui optimise les valeurs des hyperparamètres pour les réseaux CNN VGG16 et VGG19 développés pour le diagnostic des maladies des plantes a été présenté. Les hyperparamètres utilisés sont la taille des lots (Batch size) et le taux de dropout. Les auteurs ont prouvé, par des expérimentations pratiques, que leur approche a permis d'obtenir des performances supérieures avec un haut niveau de précision comparées à celles obtenues par d'autres méthodes testées pour les mêmes données. Un autre type de réseaux CNN (ConvNet) a été exploité dans [19] pour classifier les chiffres manuscrits. L'algorithme d'optimisation PSO a été adapté pour faire évoluer les paramètres internes des couches de traitement dans le but d'améliorer les résultats en termes d'Accuracy. Les résultats ont montré que l'utilisation de l'algorithme PSO permet de diminuer significativement la fonction de perte, ce qui entraîne un meilleur taux d'Accuracy durant les différentes époques.

Dans [46], les auteurs ont présenté un nouvel algorithme pour générer des architectures de réseaux de neurones avec des caractéristiques optimales grâce à l'optimisation par essaims de particules (PSO) afin d'augmenter la précision. Toutes les particules de l'algorithme ont pu trouver un réseau de neurones optimal et certaines particules ont pu réduire l'ensemble des caractéristiques. L'algorithme PSO a été également comparé à un réseau de neurones standard MLP avec une couche cachée de 256 neurones, et un nombre de 32 époques et a abouti à une précision élevée. Dans [47], deux versions d'algorithmes améliorés des méta-heuristiques de développement de plantes TGA et firefly FA swarm intelligent ont été rapportées pour l'optimisation des hyperparamètres de CNN (e.g. nombre de couches de convolution, nombre de filtre, taille des filtres, nombre de couches FC et le nombre d'unités cachées dans les couches FC). Les résultats de ces deux méthodes ont été comparés avec ceux obtenus par les algorithmes originaux en termes de précision et de temps de convergence du réseau. L'article [48] a été consacré au problème d'optimisation des hyperparamètres des réseaux CNN. Deux algorithmes d'optimisation basés sur les méta-heuristiques (un algorithme évolutionnaire et la recherche locale) ont été proposés. Ces algorithmes démontrent leur capacité à produire des CNNs avec des performances comparables à ceux rapportés dans l'état de l'art. L'algorithme des Lucioles (Firefly Algorithm « FA ») est la plus récente méta-heuristique employée dans [49]. Parmi les hyperparamètres pris en considération dans cette recherche figurent le nombre de couches convolutionnelles, le nombre de noyaux par couche et la taille des noyaux. Les auteurs ont constaté que la configuration la plus efficace du réseau est celle qui donne une erreur de test de 0,23% en utilisant FA.

Dans [50], une nouvelle version de l'algorithme PSO, appelée cPSO-CNN, a été proposée pour déterminer la configuration de l'architecture CNN. La méthode cPSO-CNN a amélioré la capacité d'exploration de la PSO classique. Les résultats ont montré que le cPSO-CNN est plus performant en termes de précision et de calcul des coûts globaux, en comparaison avec les résultats obtenus par sept autres méthodes de pointe. Une autre approche a été présentée dans [18], où les auteurs ont proposé une méthode d'optimisation d'essaim de particules à gradient de priorité (GPSO) avec des concepts liés à la priorité, basée sur l'algorithme PSO classique et l'algorithme de descente de gradient pour l'apprentissage du CNN. La méthode proposée a été testée en utilisant trois types de niveaux d'émotion pour chaque sujet et en recueillant simultanément les signaux EEG correspondant à chaque catégorie d'émotion. Les résultats expérimentaux ont prouvé que le GPSO a permis d'obtenir de hautes performances de classification sur l'ensemble des données. Dans [51], une version parallèle de l'algorithme PSO a été proposée pour l'optimisation des hyperparamètres des DNN pour résoudre les deux problèmes relatifs à : (i) l'espace de recherche qui est généralement de grande dimension et (ii) au temps d'exécution élevé. Les expériences ont révélé que la PSO

bénéficie largement de la rapidité offerte par la parallélisation du calcul et que les architectures DNN optimisées à l'aide du PSO parallèle assurent des performances de classification supérieures à celles des architectures dont les hyperparamètres sont fixées manuellement. La précision de classification obtenue se rapproche beaucoup de celles rapportées par les méthodes les plus connues de l'état de l'art.

La recherche présentée dans [52] s'est concentrée sur l'optimisation des hyperparamètres des réseaux neuronaux à convolution en utilisant l'algorithme Tree Growth (TGA). Les hyperparamètres mis en jeu sont ceux relatifs à la structure et à l'architecture du réseau, comme le nombre de couches convolutives et denses, ainsi que le nombre de noyaux par couche et la taille des noyaux. D'autres HP tels que le taux d'apprentissage, le dropout, ou la fonction d'activation, n'étaient pas pris en considération, car cette recherche nécessite une puissance de calcul élevée. Les résultats expérimentaux ont prouvé que les algorithmes d'optimisation tels que TGA, peuvent être appliqués avec succès à l'optimisation des hyperparamètres des réseaux DL.

6.4 Méthodes bayésiennes

Par rapport à une recherche par grille ou à un réglage manuel, l'algorithme BO permet de régler conjointement plus de paramètres avec moins d'expériences et de trouver de meilleures valeurs [53]. Dans [54] les auteurs ont proposé l'utilisation de l'algorithme TPE par l'intermédiaire de l'outil Hyperas pour optimiser les hyperparamètres d'un réseau CNN pour la classification des nodules pulmonaires précoces. Les hyperparamètres mis en jeu sont : le nombre de filtres, la taille de kernel, la taille du lot, l'optimiseur et le nombre d'unités dans les couches totalement connectées. Les meilleurs résultats sont les suivants : un taux d'erreur de 12%, une sensibilité de 94%, une spécificité de 83%, une précision de 88% et une F-mesure de 89%. De plus, les auteurs ont démontré que la taille du CNN est essentielle pour atteindre l'excellent résultat de 94% de sensibilité sur une petite base de données. Les auteurs dans [55] ont proposé un cadre robuste et flexible pour l'optimisation bayésienne qui peut améliorer les performances après seulement quelques évaluations. Il offre plusieurs interfaces et préconfigurations pour des cas d'utilisation typiques, tels que l'optimisation des hyperparamètres, la résolution de problèmes d'optimisation globale à faible dimension et la configuration d'algorithmes pour bien fonctionner sur plusieurs instances de problèmes.

Dans [56], l'utilisation de l'algorithme bayésien pour l'optimisation des hyperparamètres des réseaux de neurones artificiels appliqué à la classification d'images Covid-19 est présentée. L'approche proposée a atteint un taux de précision de 94 %, surpassant ainsi tous les autres

optimiseurs concurrents qui n'atteignent que des taux de précision de 86 % et de 84 %. L'optimisation bayésienne hyperband (BOHB), présentée dans [57], est appliquée dans le but d'optimiser les hyperparamètres des réseaux profonds tels que le CNN et le LSTM utilisés pour la détection automatique d'arythmie. Les hyperparamètres mis en jeu sont le nombre de couches CNN, le nombre de filtres, la taille du kernel, la fonction d'activation et le taux d'apprentissage. Les résultats expérimentaux démontrent l'efficacité de l'approche utilisée. Les mesures de performance obtenues sont les suivantes : 0,7280 pour la précision globale et 0,5931 pour le score F1 macro-moyenné pour le modèle optimisé à l'aide de BOHB, tandis que pour le DNN conçu de manière empirique, la précision globale et le score F1 macro-moyenné étaient de 0,7110 et 0,5679, respectivement.

Dans [58] un algorithme de diagnostic de défaillance de roulement de moteur basé sur un réseau de mémoire à court terme convolutionnels par optimisation bayésienne (BO-CLSTM) est proposé. Ce dernier est entraîné sur la base du signal de vibration original du roulement. La précision du modèle final a atteint 100%. De plus, par rapport à d'autres méthodes de diagnostic de défaillance avancées basées sur l'apprentissage profond, les performances de la méthode proposée ont été nettement améliorées. Les travaux dans [5] se sont intéressés à l'optimisation automatique des hyperparamètres de l'architecture BiLSTM-CNN dans le contexte de la reconnaissance automatique de la parole par utilisation de Random Forest et l'algorithme BO. L'architecture BiLSTM-CNN optimisée a atteint une précision de 84 % et un temps d'apprentissage inférieur à 17 secondes. Dans [59], l'algorithme TPE a été appliqué aux modèles 1D-CNN et LSTM afin de sélectionner automatiquement les meilleurs hyperparamètres influents dans le cadre de la reconnaissance de la chute de personnes âgées. La méthode suggérée a été appliquée à l'ensemble de données public MobiAct et à un ensemble de données réelles pour distinguer l'activité de chute des activités quotidiennes. Comme résultats, les précisions de 99,4% avec 1D CNN, 97,5% avec LSTM et 96,3% ont été obtenues avec des données réelles. Les auteurs dans [34] présentent une approche d'optimisation des HP pour la prédiction de la qualité des pièces, appelée ATPE. Cette dernière est testée sur la base de données MNIST et les résultats expérimentaux ont montré qu'elle assure des performances optimales dans un espace à haute dimension et surpasse les algorithmes RS et TPE.

6.5 Méthodes hybrides

Les méthodes d'optimisation hybride ont émergé en combinant de manière astucieuse différentes approches d'optimisation. Cette approche novatrice a suscité un intérêt croissant parmi les chercheurs, qui ont entrepris de nombreux travaux bibliographiques visant à explorer et à évaluer

l'efficacité de l'hybridation des algorithmes d'optimisation des hyperparamètres. Ces études se sont penchées sur les performances, la robustesse et la généralisation de ces approches hybrides pour ouvrir de nouvelles perspectives prometteuses dans le domaine de l'optimisation des hyperparamètres.

Dans [60], les auteurs se sont intéressés à un modèle hybride associant PSO et GS pour optimiser les paramètres d'un SVM. Ils ont utilisé GS pour réduire l'espace de recherche et PSO pour une recherche plus détaillée dans cet espace réduit. Dans [61] une nouvelle approche d'optimisation des HP est présentée. Elle combine les avantages des algorithmes GA et de la recherche Tabou (TS), afin d'optimiser les HP des réseaux utilisés. Les résultats des expérimentations révèlent que par rapport aux méthodes de recherche aléatoire et aux méthodes d'optimisation bayésienne, l'algorithme Tabu_Genetic permet de trouver un meilleur modèle en un temps plus court. Le GA [62] a été utilisé en combinaison avec PSO et Random Forest (RF) pour créer l'approche d'optimisation hybride GAPSO-RF, afin d'améliorer la précision de la prédiction des maladies cardiaques.

Dans [63] l'auteur a décrit une méthode visant à résoudre le problème de la sélection et du réglage automatiques des algorithmes d'IA, en combinant le PSO et le recuit simulé (SA). Des résultats expérimentaux ont démontré la qualité et l'efficacité de cette méthode : dans de nombreuses expériences, les résultats obtenus ont surpassé ceux générés par des outils de pointe similaires (tels qu'Auto-SKlearn ou HyperOpt SKlearn), et le temps de recherche était beaucoup plus court. Plus encore, cette approche s'est révélée très efficace dans divers domaines, en particulier, dans le secteur médical. Dans [64], les auteurs ont développé une approche d'optimisation hybride HPSOGWO-CNN en combinant le PSO et l'algorithme Grey Wolf (GWO). HPSOGWO-CNN a été utilisé pour affiner les hyperparamètres du CNN. Le modèle résultant est fortement adapté à la détection d'anomalies du système de gouvernail. Les résultats expérimentaux ont montré que cette approche hybride basée sur CNN a surpassé les modèles KNN, SVM, BP, CNN, PSO-CNN, etc., en termes de précision, avec un score de 99,846 %.

L'article [65] expose un algorithme d'optimisation des HP qui combine les algorithmes de recherche Sparrow (SSA) et le PSO, appelé Algorithme de Recherche Sparrow Hybride (HSSA). La méthode décrite exploite l'efficacité de recherche de PSO pour atteindre une optimisation globale. L'architecture utilisée est celle de LeNet-5 et les hyperparamètres mis en jeu sont le batch size, le Learning rate, le dropout rate et les nombre d'unités dans les couches fully connected utilisées. Les résultats montrent que l'algorithme HSSA est doté d'une forte capacité de recherche globale pour éviter les solutions optimales locales, et d'une efficacité de recherche satisfaisante dans les espaces de faible et de grande dimension. Les auteurs dans [66] ont présenté un algorithme

hybride bayésien-évolutif (HBEtune) qui combine des techniques fondées sur la méta-EA et l'optimisation bayésienne et la performance de HBEtune est à la hauteur des progrès réalisés dans ce domaine.

Un nouvel algorithme d'optimisation hybride est présenté dans [67]. Il combine le PSO adaptatif (APSO) avec le whale optimizer Algorithm (WOA) afin d'optimiser les HP suivants du réseau CNN : la fonction d'activation, le batch size et le Learning rate. L'approche proposée est comparée avec d'autres algorithmes d'optimisation, tels que l'algorithme APSO et a obtenu de meilleures performances. Dans [68], les auteurs ont exploré une nouvelle méthode basée sur l'évolution pour calibrer les hyperparamètres dans le contexte de l'apprentissage automatique. La méthode hybride proposée, HyperBRKGA, combine un algorithme génétique et une méthode fondée sur l'exploitation. Les hyperparamètres mis en jeu sont le nombre de neurones dans chaque couche et le Learning rate. Dans les expériences menées pour cette étude, HyperBRKGA a été comparé à d'autres méthodologies (Grid Search, Random Search, Bayesian Optimization), et les résultats ont montré que HyperBRKGA a pu trouver des configurations d'hyperparamètres qui ont surpassé les autres configurations en termes de qualité prédictive.

7 Discussion

Dans le Tableau 1. 2, nous présentons une variété de méthodes utilisées pour optimiser les hyperparamètres des réseaux de neurones tels que RS, GS, PSO, TPE et EA. L'utilisation de la RS pour évaluer les valeurs d'hyperparamètres sélectionnées de manière aléatoire aide les analystes à explorer un vaste espace de recherche. Cependant, il ne prend pas en compte les résultats précédemment testés, il peut impliquer ainsi de nombreuses évaluations inutiles, ce qui diminue son efficacité. Concernant le GS, malgré ses points forts, il présente un certain nombre de limitations. Par exemple il est inadapté à l'optimisation conjointe d'un grand nombre d'hyperparamètres. De plus, il n'est pas bien adapté aux paramètres continus, puisqu'il nécessite une quantification *a priori* de ces derniers. En outre, la principale limite de GS et de RS est que chaque évaluation réalisée dans une itération est indépendante des évaluations précédentes. Ainsi, elles perdent un temps considérable à évaluer les zones peu performantes de l'espace de recherche. Ce problème peut être résolu par d'autres algorithmes d'optimisation, tels que TPE, BO ou GA. Ces derniers sont plus compliqués que GS et RS, mais sont en général plus performants pour les problèmes d'optimisation complexes. Ils peuvent utiliser tous les types d'hyperparamètres et sont particulièrement efficaces pour les grands espaces de configuration, puisqu'ils peuvent obtenir des solutions quasi-optimales

même en peu d'itérations. Dans cette thèse, nous avons donc choisi de nous focaliser sur les méthodes bayésiennes ainsi que la recherche par grille et la recherche aléatoire.

Tableau 1. 2: Tableau comparatif des algorithmes d'optimisation des HP utilisés pour réseaux de neurones en termes de type, algorithmes d'optimisation, architectures concernées, hyperparamètres et métriques d'évaluation

Références	Type	Algorithmes d'optimisation	Architecture	Hyperparamètres	Métriques d'évaluation
[27]	Méthodes Systématiques	RS GS	DNN	Nombre de couches cachées, Dropout, Nombre d'époques, Batch size	Accuracy :91.67% Sensitivité :95.83% Spécificité : 88.89% Précision :85.19% F1-score :90.20%
[29]	Algorithmes Génétiques	GA	CNN	Learning rate, Dropout rate, Nombre de couches CONV	Erreur : 0.37%
[31]		Evolution Différentielle (DE)	LSTM	Taille du lot, Nombre de neurones dans les couches cachées	Accuracy :77.68%
[33]		GA	CNN	Taille du kernel, Fonction d'activation Taille du filtre, Taux d'apprentissage, Optimiseur	Accuracy : 94%
[41]	Méthodes Méta- heuristiques	PSO	CNN	Nombre d'époques, Fonction d'activation	Accuracy : 97.71%
[42]		Efficient Global Optimization (EGO)	CNN	Fonction d'activation, Dropout rate	Accuracy : 86.46%
[43]		PSO	AlexNet	Taille des kernels, Padding	Accuracy : 80.15%
[45]		Orthogonal Learning-PSO (OLPSO)	VGGNet 16 VGGNet 19	Batch size Dropout rate	VGG16 Accuracy :97.9 % VGG19 Accuracy : 97.7%
[19]		PSO	ConvNet	Nombre des couches CONV, Learning rate, Batch size	Accuracy :97.67%
[54]	Méthodes bayésiennes	TPE	CNN	Nombre de filtres, Taille de kernel, Taille du lot, Optimiseur, Nombre d'unités dans les couches FC	Sensitivité : 94% Spécificité : 83% Précision : 88% F1-Score : 89%
[5]		TPE	BiLSTM-CNN	Learning rate, Nombre de couches cachées, Batch size	Accuracy :84%
[59]		TPE	1D-CNN LSTM	Taille de filtre, fonction d'activation, nombre de couches denses, nombre de couches CONV	1D-CNN Accuracy : 99.4% LSTM Accuracy :97.5%
[58]		BO	CLSTM	Learning rate, Batch size, Nombre de couches denses	Accuracy : 97.20%

[36]	Algorithmes évolutionnaires	EA	ANN	Fonction d'activation, Nombre d'époques	Accuracy : 80.96%
[37]		HOUSES	ML-CNN	Nombre de couches CONV	Accuracy : 85%
[56]	Méthodes bayésiennes	BO	ANN	Nombre de couches CONV, Nombre de couches denses	Accuracy :94%
[60]	Méthodes hybrides	PSO-GS	SVM	Nombre d'époques	Accuracy : 93.75%
[61]		Tabu_GA	CNN	Learning rate, Batch size, Dropout rate Weight decay	Accuracy :78.81%
[62]		GA-PSO	RF	Nombre d'arbres, Max_depth, Min_samples_split, Min_samples_leaf	Accuracy : 87.80%
[64]		HPSO-GWO	CNN	Taille de filtre, Nombre de filtres, Pool size	Accuracy :99.84% Precision: 99.74% Recall: 99.49% F-score: 99.61%

8 Conclusion

Dans ce premier chapitre, nous avons présenté les concepts inhérents à l'optimisation des hyperparamètres, décrit les algorithmes d'optimisation existants, les bibliothèques et frameworks disponibles pour les utilisateurs, et enfin réalisé une revue de la littérature portant sur les algorithmes d'optimisation des hyperparamètres de DNN. Nous avons observé qu'il existe de nombreuses techniques différentes, qui couvrent un large éventail d'approches, notamment les algorithmes évolutionnaires, les métaheuristiques, les méthodes exhaustives et les algorithmes bayésiens tels que la méthode TPE. Dans notre travail, nous exploiterons plus spécifiquement cette dernière combinée avec les méthodes systématiques GS et RS, pour optimiser les hyperparamètres des modèles de réseaux neuronaux profonds dédiés à l'analyse de signaux physiologiques temporels, dans le but d'améliorer leurs performances.

Chapitre 2

Traitement de Signaux EEG via les Réseaux de Deep Learning : Etat de l'art

1 Introduction

La notion de vigilance recouvre différentes définitions selon le contexte et les disciplines scientifiques telles que la neuropsychologie et la physiologie. En effet, elle peut être associée au degré d'activation du cerveau, jouant ainsi un rôle fondamental dans toutes les opérations mentales, allant de la simple détection d'informations à la complexe élaboration et expression de comportements. L'Electroencéphalographie (EEG) émerge comme le moyen le plus direct pour mesurer cette activité cérébrale. Il permet de distinguer plusieurs états de vigilance, et ainsi d'améliorer la compréhension de ce phénomène. Dans ce chapitre, nous commençons par définir le premier contexte de validation de nos concepts à savoir la détection et la quantification de la vigilance à base d'analyse des signaux EEG. Ensuite, nous présentons les techniques employées dans le cadre du prétraitement et de l'extraction des caractéristiques de ce signal. Avant de conclure, nous réalisons une étude bibliographique sur les travaux portant sur le traitement des signaux EEG par des réseaux de neurones profonds avec comme finalité la caractérisation et la quantification des états de vigilance.

2 Contexte et performance dans la conduite automobile

La conduite automobile est une tâche complexe qui exige des actions continues et une adaptation constante en fonction des changements dans un environnement en perpétuel changement. Au fil des années, les réglementations ont cherché par exemple à améliorer la sécurité et l'efficacité au volant en vain, du fait qu'elles n'ont pas considéré, le plus souvent un élément majeur : les fluctuations de vigilance du conducteur pendant la conduite. Pour évaluer les différents états de vigilance, une approche intéressante et objective repose sur l'utilisation de mesures des signaux physiologiques. A cet effet, divers types de signaux peuvent servir d'indicateurs physiologiques, tels que l'Electroencéphalographie (EEG), l'Electrocardiographie (ECG), l'Électromyographie

(EMG) et l'Electrooculographie (EOG). Ces signaux permettent respectivement de capter les activités électriques du cerveau, du cœur, des muscles et des mouvements oculaires. Parmi ces signaux, l'EEG se distingue particulièrement par sa capacité à caractériser les états de vigilance et plus généralement les aspects liés à la fonction cérébrale. Il permet en effet de mesurer les variations électriques par le biais des électrodes placées sur le cuir chevelu, retraçant ainsi une vision de l'activité cérébrale. Le choix de privilégier l'EEG pour caractériser les états de vigilance repose sur le consensus parmi les experts du domaine stipulant que ce signal reflète de manière précise et objective l'éveil mental de la personne et a l'avantage d'être plus facilement intégré dans des systèmes portables [69].

3 L'électroencéphalogramme (EEG)

3.1 Historique

L'électroencéphalographie (EEG) a été conçue par Hans Berger (Figure 2. 1), un physiologiste allemand, qui a lancé cette technique dans les années 1920 et qui a enregistré un premier signe d'activité cérébrale en 1929. Ses travaux ont été étendus par le scientifique britannique Edgar Douglas Adrian, qui a reçu le prix Nobel de physiologie (1932). Cependant, ce n'est que dans les années 1950 que l'EEG s'est réellement imposé et s'est répandu dans la pratique médicale courante, notamment dans l'épileptologie et la détection de la vigilance. Auparavant, les tracés EEG étaient réalisés sur un support analogique (papier), puis examinés visuellement par les neurologues à la recherche de signes éventuels d'anormalité. Actuellement, le signal est converti au format numérique et traité par ordinateur. Les signaux sont largement employés pour le diagnostic des foyers épileptiques, des tumeurs cérébrales et des lésions. Par ailleurs, les variations de cette activité électrique cérébrale renseigneraient sur les états de vigilance d'une façon objective.

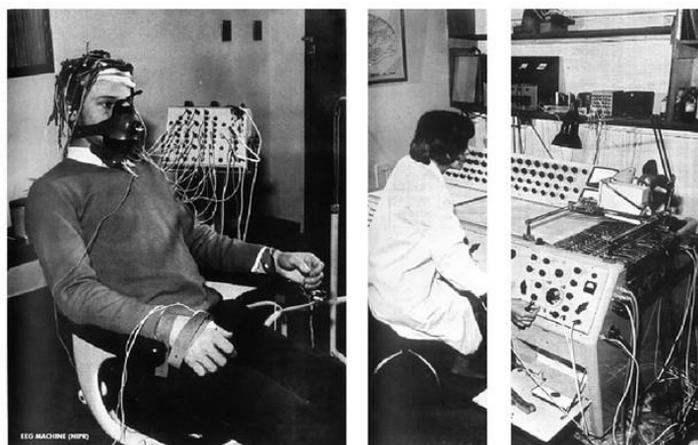


Figure 2. 1: L'inventeur de l'électroencéphalographie (EEG) : Hans Berger

3.2 Principe et définition

L'EEG estime l'activité électrique du cerveau, généralement par le biais d'un tracé temporel, mesuré par un examen connu sous le nom d'encéphalographie. Ce tracé traduit les signaux transmis sous forme d'ondes par le cerveau. On recueille ce signal à l'aide d'électrodes posées sur le cuir chevelu. Il renseigne donc sur l'activité neurophysiologique du cerveau permettant ainsi d'établir des diagnostics neurologiques. La Figure 2. 2 illustre un exemple d'un enregistrement EEG. Le tracé de ces potentiels varie dans le temps, créant ainsi des rythmes. Chaque rythme se traduit par les variations enregistrées au niveau de la fréquence et de l'amplitude de ces ondes.

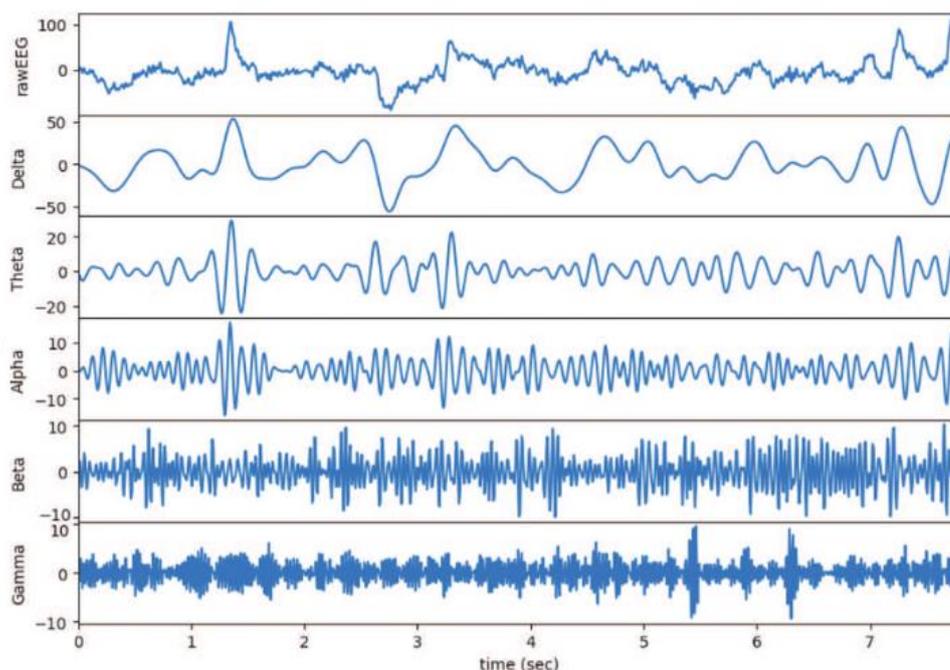


Figure 2. 2: Exemple de tracé EEG (première ligne), et sa décomposition en bandes de fréquences pertinentes (lignes suivantes). L'axe vertical représente la différence de potentiel, en μV [70].

3.3 Les ondes d'EEG et leurs significations en pratique

On appelle rythme cérébral une oscillation électromagnétique qui survient dans un intervalle spécifique de fréquences. Cette oscillation est le résultat de l'activité électrique coordonnée d'un grand nombre de neurones du cerveau, observée par l'EEG. Les caractères de ces rythmes varient selon les individus et leur état psychique, qu'il soit normal ou pathologique. Enregistrer ces rythmes permet d'identifier et de caractériser différents troubles neurologiques. Les rythmes cérébraux sont regroupés en cinq catégories selon leur fréquence (Figure 2. 3).

- **Ondes Delta** : Elles se situent dans la gamme de 0,5 à 4 Hz, avec des amplitudes variables. Ces ondes sont associées généralement au sommeil profond.

- **Ondes Thêta** : Elles ont une fréquence de 4 à 7 Hz et une amplitude qui varie de 10 à 50 μ V d'amplitude. Leur formation résulte des interactions entre les deux populations corticale et hippocampique. On les trouve surtout au niveau des régions temporales, lors de certaines tâches cognitives et pendant la phase de sommeil.
- **Ondes Alpha** : Elles ont une fréquence de 7 et 13 Hz et une amplitude allant de 30 à 50 μ V. Elles se produisent généralement au niveau des régions occipito-pariétales. Ces ondes sont associées à l'état d'éveil ou de repos mais avec yeux fermés. Ces ondes qualifient les états de conscience apaisés, particulièrement observés lors de séances de méditation ou de relaxation légères. Elles disparaissent totalement durant la période de sommeil.
- **Ondes Bêta** : Elles ont une fréquence comprise entre 13 et 30 Hz, et une légère amplitude de moins de 20 μ V. Elles sont générées dans les zones médianes des deux hémisphères, de façon parfois asynchrone. En général, ces ondes se manifestent lors de l'activité du système nerveux et sont liées au stimulus sensoriel ainsi qu'à l'activité mentale. Elles sont associées à des états d'éveil attentif et des périodes de forte activité et de concentration.
- **Ondes Gamma** : Ce sont les ondes les plus rares, dont l'amplitude est très basse et la fréquence oscille entre 30 Hz et 100 Hz. Elles sont associées à des états d'éveil attentif, ou des périodes de forte activité et de concentration.

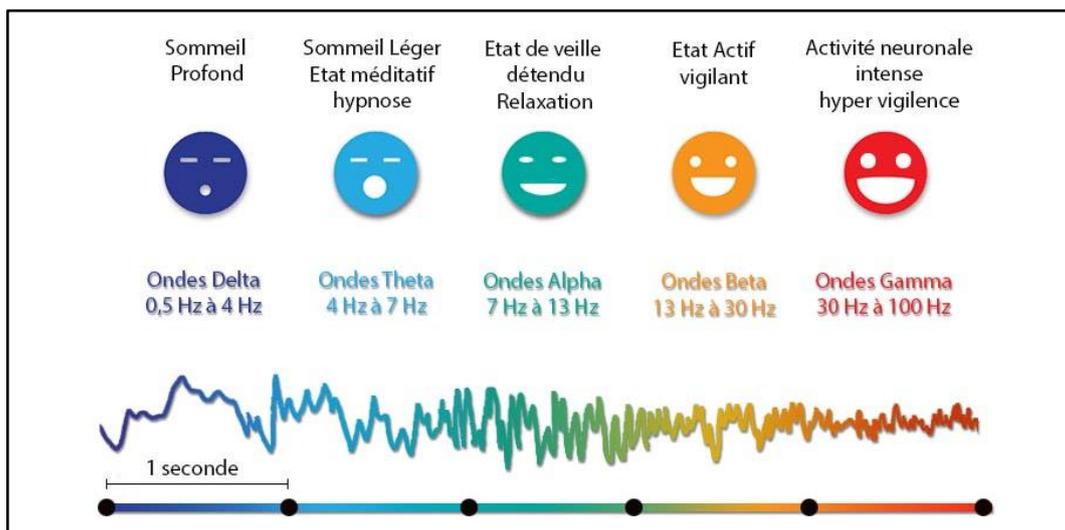


Figure 2. 3: Les rythmes cérébraux de l'EEG

La capacité de l'EEG à caractériser les états du sommeil est attestée dans la littérature [71] alors que la quantification et la modélisation des états d'éveil reste sujet à controverse dans la communauté scientifique. D'où l'intérêt d'approfondir ce sujet.

4 Traitement du signal EEG

L'analyse du signal EEG implique généralement l'application en amont d'un processus en quatre étapes distinctes : acquisition des données, prétraitement, extraction des caractéristiques et classification (comme illustré dans la Figure 2. 4). Toutes ces étapes ont un rôle primordial puisqu'elles transforment le signal électrique brut émis par le cerveau en un signal exploitable en vue d'en extraire des informations pertinentes.

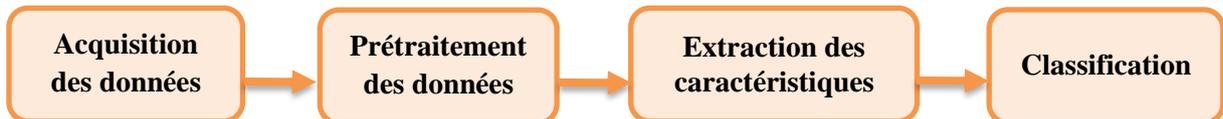


Figure 2. 4: Traitement du signal EEG

4.1 Acquisition des signaux EEG

La collecte de signaux EEG repose sur la capture, l'amplification, la modification et l'enregistrement des données d'activité enregistrées par plusieurs électrodes. Cela permet de discerner les variations de potentiel qui peuvent se manifester sur différentes zones du cuir chevelu. Pour cela, les électrodes EEG sont placées sur le cuir chevelu du patient selon un montage standardisé (par exemple système 10-20). Les emplacements des électrodes sont spécifiés selon le pourcentage des distances entre des points anatomiques de référence sur le cortex (Figure 2. 5).

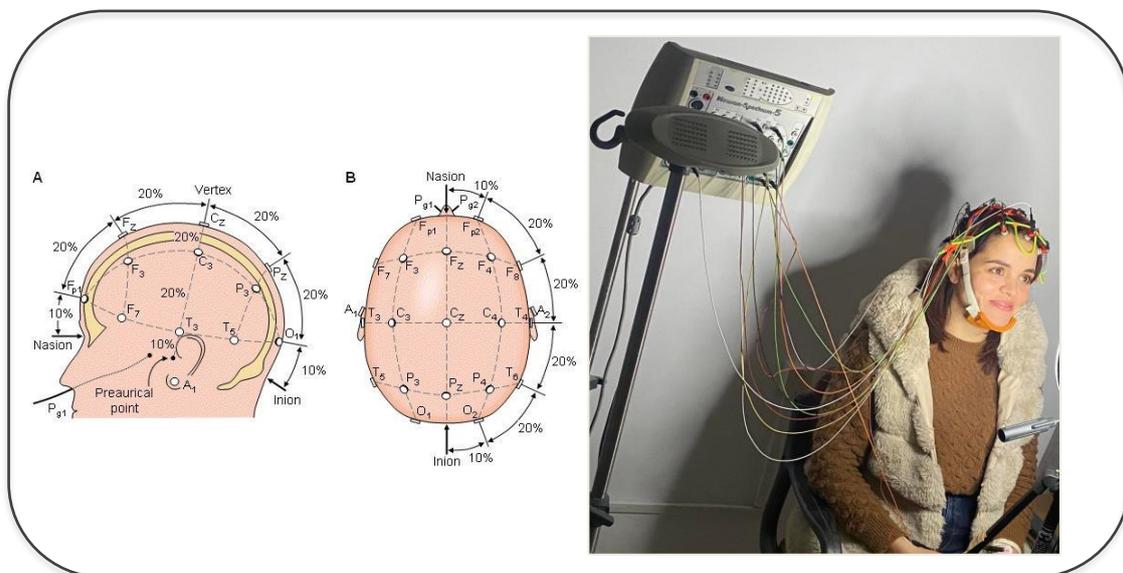


Figure 2. 5: Acquisition du signal EEG

Le signal EEG enregistré comporte généralement un certain nombre d'artefacts, ce qui nécessite des prétraitements tel que le débruitage ou l'élimination de fréquences indésirables.

4.2 Prétraitement des données

Bien que l'objectif premier du signal EEG soit la capture de l'activité cérébrale, il enregistre également des manifestations électriques provenant de sources autres que le cerveau. Ces phénomènes enregistrés qui ne découlent pas de l'activité cérébrale proprement dite sont communément désignés sous le terme d'artefacts d'où la nécessité de prétraiter le signal EEG avec comme double objectif d'éliminer les artefacts émanant des différentes sources électriques environnantes, tout en améliorant simultanément le Rapport Signal sur Bruit (RSB). Plus précisément, le prétraitement du signal EEG s'articule autour de trois étapes fondamentales : l'échantillonnage, qui assure une capture précise du signal EEG ; le filtrage, qui vise à isoler les fréquences d'intérêt tout en réduisant les interférences indésirables et enfin, le fenêtrage qui permet de découper le signal en segments temporels pertinents pour une analyse plus approfondie.

4.2.1 Échantillonnage

L'échantillonnage désigne le processus de collecte régulière de mesures du signal électrique généré par l'activité cérébrale. Il consiste à enregistrer des échantillons du signal EEG à des intervalles de temps réguliers, définis par la fréquence d'échantillonnage. Cette fréquence détermine le nombre d'échantillons par seconde et influence directement la qualité et la résolution temporelle des enregistrements EEG. Le choix d'un taux d'échantillonnage approprié est important pour garantir une représentation précise et complète de l'activité cérébrale.

4.2.2 Filtrage

Le filtrage est une étape essentielle du prétraitement du signal EEG, qui vise à éliminer certaines fréquences indésirables.

4.2.2.1 Filtrage dans le domaine fréquentiel

Un filtre passe-bande est appliqué aux signaux EEG bruts afin d'éliminer les fréquences et les hautes fréquences qui dépassent la plage de variation du rythme cérébral. Cette plage est généralement limitée aux fréquences comprises entre 0.1 à 32 Hz. A cet effet, les filtres de Butterworth et de Tchebychev constituent les filtres à réponse impulsionnelle infinie les plus fréquemment employés en tant que filtres passe-bande lors du prétraitement de l'EEG [72]. Parfois, lorsqu'il est préférable de conserver les hautes fréquences, on peut appliquer aux signaux un filtre

coupe-bande [73], dont la fréquence de coupure est de 50 ou 60 Hz, afin d'éliminer certains artefacts liés aux lignes électriques.

4.2.2.2 Filtrage dans le domaine spatial

Les filtres spatiaux peuvent être employés pour la réduction du bruit, la détection et l'élimination des artefacts corporels et l'amélioration de la localisation d'activités cérébrales spécifiques, lorsqu'un grand nombre d'électrodes sont utilisées pour l'enregistrement EEG.

Parmi les techniques couramment utilisées pour le filtrage spatial des signaux EEG, nous pouvons citer :

- ***L'Analyse en Composantes Indépendantes (ACI)*** : Sépare les sources de signaux EEG en supposant leur indépendance, permettant ainsi la séparation des signaux cérébraux des artefacts et du bruit [74].
- ***Analyse en Composantes Principales (ACP)***: Réduit la dimension des données EEG en identifiant les composantes principales pour l'extraction des caractéristiques du signal [75].
- ***CSP (Modèle Spatial Commun - Common Spatial Patterns)*** : Identifie les combinaisons linéaires d'électrodes maximisant la variance entre deux classes de signaux EEG (par exemple, signal cérébral *vs* artefacts) pour mettre en évidence les caractéristiques distinctives des signaux d'intérêt [76].

4.2.3 Fenêtrage

Cette technique est principalement utilisée pour des signaux considérés comme suffisamment stationnaires pour être restreints en termes de durée. On fixe ainsi une période (fenêtre temporelle) dans laquelle il est possible de restreindre la décomposition fréquentielle. Il existe plusieurs sortes de fenêtres :

- ***Fenêtre Rectangulaire*** : Cette fenêtre représente la méthode la plus facile et équivaut à une absence totale de pondération. Elle ne devrait être utilisée que dans les cas où la globalité du signal est incluse dans la mémoire de calcul.
- ***Fenêtre de Hann*** : Cette fenêtre représente la plus grande partie des fenêtres utilisées. Elle permet d'analyser certains signaux pseudo-stationnaires (par exemple, le bruit d'une machine). C'est le meilleur compromis entre la résolution de fréquence et la résolution d'amplitude.
- ***Fenêtre de Hamming*** : Elle est dérivée de la fenêtre de Hann, mais en mettant davantage en évidence la résolution fréquentielle. Elle a été recommandée pour la recherche d'une fréquence spécifique.

- **Fenêtre Flat Top** : Il s'agit là de la seule fenêtre permettant d'obtenir une amplitude exacte. En revanche, elle présente une très faible résolution en fréquence, ce qui la rend inadaptée lorsque le spectre est formé par plusieurs fréquences ayant des caractéristiques similaires. Elle est principalement utilisée pour l'étalonnage.

4.3 Extraction des caractéristiques

Après l'étape de prétraitement du signal, cette étape vise à extraire les caractéristiques de chaque période d'enregistrement. Cette étape peut être réalisée à partir de différents niveaux d'analyse.

4.3.1 Types d'analyses des signaux EEG

L'extraction des caractéristiques des signaux EEG se base sur plusieurs types d'analyse dont nous citons, dans le domaine temporel l'analyse aperiodique, et dans le domaine fréquentiel l'analyse bi-spectrale [77] et l'analyse spectrale.

4.3.1.1 Analyse aperiodique (Temporelle)

Une analyse aperiodique est une méthode d'analyse non paramétrique simple. Son principe repose sur le comptage du nombre de fois où le tracé EEG traverse une droite isoélectrique donnée par unité de temps (fréquence moyenne) et la puissance globale du signal EEG. Cette technique pose certains problèmes théoriques. Elle est en effet extrêmement sensible aux bruits de haute fréquence dans l'estimation des croisements de la droite isoélectrique, et extrêmement sensible aux variations minimales du signal EEG. De plus, elle ne permet pas non plus de caractériser la distribution des fréquences autour de la fréquence moyenne. Parmi les travaux qui ont utilisé l'analyse aperiodique durant la phase d'extraction des caractéristiques, nous pouvons citer [78] [79].

4.3.1.2 Analyse spectrale

Cette méthode décompose chaque signal EEG sous forme d'une somme de sinusoides de différentes fréquences et tensions, ce qui permet de reconstituer le signal EEG d'origine en superposant toutes ces sinusoides.

L'analyse spectrale se présente comme l'analyse la plus adaptée de la littérature pour quantifier les tracés EEG étant donné le lien direct entre les propriétés fréquentielles des signaux EEG et les processus neurophysiologiques [72][80].

4.3.2 Techniques d'extraction de caractéristiques

Parmi les techniques d'extraction de caractéristiques utilisées dans le domaine fréquentiel, nous mentionnons l'analyse par ondelettes (wavelet analysis) ainsi que l'analyse par transformée de Fourier (Fourier transform analysis).

4.3.2.1 Transformée en ondelette

La transformée en ondelettes est une technique permettant de représenter un signal sous la forme d'une somme pondérée d'ondes translatées ou dilatées. Elle décompose les fréquences d'un signal tout en préservant sa localisation dans l'espace. Une projection du signal de départ est effectuée sur un ensemble de fonctions de base qui varient en fréquence et en temps (phase). Celles-ci se calquent sur les fréquences que contient le signal à analyser. Cette représentation temporelle et fréquentielle du signal permet une localisation temporelle et fréquentielle du signal analysé. Plusieurs travaux ont utilisé l'analyse par ondelette pour la décomposition des signaux EEG dont [81] [82].

4.3.2.2 Transformée de Fourier

La transformée de Fourier est l'un des outils les plus utilisés dans le traitement du signal. Elle associe la "forme d'onde" habituelle, à savoir la représentation d'un signal en fonction de sa variable d'évolution, à une représentation supplémentaire d'un signal dans le domaine des fréquences. Le recours à cette description fréquentielle permet également de caractériser facilement les filtres linéaires et d'en faciliter l'étude. L'analyse de Fourier consiste donc à effectuer une analyse spectrale basée sur la notion de fréquence, de temps et de périodicité. En effet toute fonction temporelle périodique du temps peut se décomposer en une somme infinie de sinus et de cosinus ayant des fréquences commençant à zéro et augmentant par multiples entiers d'une fréquence de base. La transformée de Fourier joue systématiquement un rôle essentiel dans l'analyse des signaux stationnaires. Dans le cas des signaux non stationnaires, par exemple, les signaux EEG, une séquence de transformées de Fourier à court terme (STFT) est appliquée au signal [83] [84][85].

4.4 Classification pour la détection de la baisse de vigilance

L'objectif de la classification est de mettre en œuvre un système capable de classer les états de vigilance d'un sujet [86] [87]. Il permet ainsi de réaliser une expertise qui peut être fastidieuse à réaliser par un être humain, au regard du temps nécessaire au traitement d'un grand volume de données. Dans ce contexte, le Machine Learning peut être une alternative intéressante vu le grand nombre de techniques supervisées (arbre de décision, Random Forest, Algorithmes génétiques,

réseaux de neurones Machines à Vecteurs de Support (SVM), etc.) et non supervisées (K-Means, DBSCAN, etc.) disponibles qui peuvent être appliquées pour classifier les états de vigilance. Dans notre travail, nous nous intéressons plus spécifiquement à la classification par réseaux de neurones.

5 Étude bibliographique sur l'exploitation des Réseaux Deep Learning dans le Traitement de l'EEG

Le traitement et l'interprétation du signal EEG sont des tâches complexes en raison de sa nature bruitée, non linéaire et dynamique. Dans ce contexte, les Réseaux de Neurones Profonds ont émergé comme un outil puissant pour extraire des informations significatives à partir de ces signaux, permettant ainsi des avancées majeures dans la compréhension des processus cognitifs, de la neurologie et des interfaces cerveau-ordinateur. Dans la littérature, de nombreuses études ont été effectuées pour exploiter cette approche prometteuse. Citons à titre d'exemple la classification des états épileptiques, le traitement de la maladie de Parkinson, l'analyse des émotions et la classification des différents stades de vigilance.

Dans [88], les auteurs ont proposé une approche hybride pour la détection des crises d'épilepsie [89], dans laquelle le réseau CNN est utilisé pour extraire les caractéristiques de signal EEG, et les SVM sont utilisées pour classifier les crises d'épilepsie. Cette approche a été évaluée à l'aide de l'ensemble de données du Children's Hospital Boston-Massachusetts Institute of Technology (CHB-MIT). Selon les résultats expérimentaux, la précision du modèle combiné CNN-SVM est supérieure à celle du modèle CNN de base. Dans [90], un modèle multicanal CNN-Bi-LSTM est proposé pour la prédiction des crises d'épilepsie. Le CNN est chargé d'extraire les caractéristiques spatiales, tandis que le Bi-LSTM est utilisé pour extraire les caractéristiques temporelles. Les performances du modèle sont testées sur l'ensemble de données CHB-MIT, et la sortie est divisée en trois catégories, à savoir normale, préictale et ictale. Les résultats obtenus ont montré des scores de classification remarquables, avec une précision de 94,83%, un recall de 94,84 %, et un score F1 de 94,83%. Dans [91], les auteurs ont présenté un modèle basé sur un réseau CNN pour détecter les crises d'épilepsie à partir de données EEG brutes. La méthodologie proposée a permis d'atteindre une précision maximale de 100 %. Dans [92] des réseaux CNN ont été appliqués sur divers ensembles de données EEG intracrâniens afin prédire les crises d'épilepsie. L'approche proposée atteint une sensibilité de 89,8% et un taux de fausse prédiction (FPR) de 0,17/h sur l'ensemble de données EEG intracrâniens (iEEG) de l'hôpital de Fribourg, et une sensibilité de 89,1 % et un FPR de 0,09/h sur l'ensemble de données EEG du scalp (sEEG) de l'hôpital pour enfants de Boston-MIT. Dans [93], un CNN a été utilisé pour la prédiction des crises focales épileptiques à l'aide des signaux EEG. Des filtres de

convolution fondés sur la transformation en ondelettes du signal EEG ont été utilisés pour apprendre des signatures quantitatives pour chaque période : interictale, préictale et ictale. Les résultats expérimentaux démontrent que les performances sont prometteuses avec une sensibilité de 87,8%. Des études réalisées sur les EEG ont montré aussi que les réseaux de neurones profonds permettent de diagnostiquer les troubles neurologiques de la maladie de Parkinson (PD). Dans [94], un modèle mixte fondé sur deux réseaux CNN et RNN est proposé afin de distinguer les patients atteints de PD des témoins sains à l'aide des signaux EEG. Grâce à cette hybridation, le modèle a pu extraire les caractéristiques spatiales et temporelles de l'EEG pertinentes du point de vue des fonctions cérébrales. Les résultats de la validation ont démontré qu'il existe un lien entre les caractéristiques apprises dans le modèle et les données cliniques telles que la gravité de la maladie. Dans [95], quatre architectures basées sur l'apprentissage profond, dont deux modèles d'apprentissage profond classiques (CNN et RNN) et deux réseaux récurrents convolutifs hybrides (2D-CNN-RNN et 3D-CNN-RNN), ont été élaborés pour la détection de la maladie de Parkinson fondée sur les signaux EEG. Il en est ressorti que les modèles mixtes ont dépassé les modèles conventionnels (Accuracy : 3DCNN-RNN 82,89%, 2D-CNN-RNN 81,13%, CNN 80,89% et RNN 76,00%), grâce à la combinaison de la forte capacité de modélisation des CNN dans l'extraction de caractéristiques temporelles et l'avantage déterminant des RNN dans le traitement d'informations séquentielles.

L'évaluation des émotions à l'aide de l'EEG est un autre sujet intéressant d'application. Une analyse approfondie de l'approche basée sur la classification de la dépression à l'aide des signaux EEG est effectuée dans [96] en utilisant un réseau CNN. Ce dernier a atteint une précision de 99,31% dans la classification de la dépression à partir de signaux EEG de témoins normaux enregistrés dans l'hémisphère droit du cerveau et de 96,3 % dans l'hémisphère gauche du cerveau. Dans [97], les auteurs ont suggéré un réseau CNN pour distinguer les signaux physiologiques des artefacts non physiologiques. L'appareil Nihon Kohden a été utilisé pour acquérir les données EEG, collectées à partir de 32 canaux. La méthodologie proposée fournit des résultats de classification remarquables allant jusqu'à 99,55%. Dans [98], l'objectif était de classifier trois types d'émotion, à savoir la peur, le plaisir et la tristesse à partir de signaux EEG et d'un CNN. Une accuracy de 84,69% a été obtenue, reflétant la capacité du CNN à extraire les caractéristiques ayant le plus d'impact sur la classification. La détection de la fatigue mentale a été également abordée dans [99]. Le modèle proposé combine un CNN et un LSTM pour distinguer trois états de fatigue mentale différents : éveil, fatigue légère et fatigue sévère. Le modèle offre une sensibilité de 97,80% et une spécificité de 99,28%.

Dans [100] les auteurs ont exploré l'application de l'apprentissage pour évaluer l'état cognitif des conducteurs à partir de signaux EEG. Ils ont opté pour l'utilisation d'un CNN en raison de sa capacité intrinsèque à extraire, apprendre et généraliser des caractéristiques sans nécessiter de prétraitement

des données. Les enregistrements expérimentaux ont été réalisés lors de sessions du jeu vidéo GTAV, en utilisant le mode de conduite libre afin de simuler des situations de conduite réalistes. Les résultats obtenus ont démontré une précision de classification remarquable, de 95,76% pour la classification de la charge cognitive élevée, et de 92,57% pour les sessions à faible charge cognitive. Les travaux dans [101] ont présenté une nouvelle approche basée sur l'apprentissage profond pour classifier les stades du sommeil. Cette méthode, contrairement à d'autres, exploite l'ensemble des signaux polygraphiques (EEG, EMG, EOG) et tient compte du contexte temporel sur des données de 30 secondes. Les résultats, obtenus à partir de 61 enregistrements polysomnographies (PSG) sur 20 canaux EEG, montrent que le modèle surpasse les autres approches basées sur des CNN ou des arbres de décision.

La charge de travail mentale (MWL), le stress et la vigilance sont des processus cognitifs qui ont fait l'objet d'une analyse dans les travaux [102]. Les MWL ont été analysées et classifiées à l'aide de deux modèles, le SVM et le CNN ; les précisions de classifications avec le CNN ont surpassé celles obtenues avec SVM, passant de 80 à 87%. Par ailleurs, les réseaux de neurones profonds ont aussi été exploités avec succès pour la détection de la vigilance ainsi que de la fatigue mentale à l'aide des signaux EEG [69] [103] [104]. Dans [85], Les auteurs ont présenté une approche de prédiction de la vigilance en utilisant des LSTM tout en combinant les signaux EEG et EOG. Ces deux signaux ont été enregistrés de façon synchrone en utilisant le système Neuroscan. Les données de mouvements oculaires ont été enregistrées à l'aide de lunettes de suivi oculaire. Pour l'extraction des caractéristiques, un échantillonnage est appliqué aux signaux EEG afin de réduire la complexité de calcul et un filtre passe-bande est utilisé pour réduire le bruit et les artefacts. Les résultats ont montré que les modèles LSTM peuvent significativement améliorer les résultats d'estimation par rapport au réseau SVM. Dans [105] les auteurs ont proposé des méthodes profondes de classification des états mentaux (sommolence / alerte) des conducteurs : CNN et DNN, où les données sont des matrices de covariance d'échantillon obtenues à partir des signaux EEG. Les méthodes d'apprentissage par covariance profonde ont montré de meilleures performances que celles des méthodes d'apprentissage classiques. Dans [106], les auteurs ont exploré le potentiel de la combinaison de différents types d'architectures DL pour améliorer la performance de la classification de l'état de vigilance basée sur l'analyse des activités cérébrales à l'aide des signaux EEG. Le CNN-LSTM proposée permet d'atteindre une précision de 85%.

6 Comparaison des différentes méthodes existantes

L'exploration des réseaux de neurones profonds pour l'analyse des signaux EEG constitue une alternative prometteuse et intéressante dans le domaine de la recherche médicale. Le Tableau 2. 1 synthétise les principales méthodes évoquées plus haut.

Tableau 2. 1: Méthodes existantes pour l'exploration des réseaux de neurones profonds via l'EEG

Auteurs	Objectif de l'étude	Réseaux de neurones utilisés	Performance
[88]	Classification des crises épileptiques en utilisant une architecture hybride basée sur les CNN.	SVM-CNN	Accuracy : 95%
[90]	Prédiction des crises épileptiques fondée sur les signaux EEG	CNN-Bi-LSTM	Accuracy : 94,83% Sensitivité :94,84 % F1-Score : 94,83 %
[92]	Prédiction des troubles neurologiques de la maladie de Parkinson	RNN-CNN	Accuracy : 96.9% Sensitivité : 93.4%
[95]	Détection de la maladie de PD fondée sur les signaux EEG	3DCNN-RNN 2D-CNN-RNN CNN RNN	Accuracy : 82,89% 81,13% 80,89% 76,00%
[96]	Classification de la dépression à l'aide des signaux EEG	CNN	Accuracy : 99.31%
[97]	Détection des artefacts physiologiques et non physiologiques.	CNN	Accuracy/ 99.55%
[98]	Classification des émotions avec des signaux EEG.	CNN	Accuracy : 84,69 %
[85]	Prédiction de la vigilance en utilisant des réseaux LSTM avec des signaux EEG et EOG.	LSTM	RMSE :0.0807
[99]	Détection de la fatigue mentale avec des signaux EEG.	CNN-LSTM	Sensitivité : 97,80% Spécificité : 99,28 %
[100]	Évaluation des tâches cognitives des conducteurs à partir des signaux EEG	CNN	Accuracy : 95.76%
[101]	Classification des stades du sommeil fondée sur l'ensemble des signaux polygraphiques (EEG, EMG, EOG).	DNN	Accuracy : 87% Sensitivité : 85%
[102]	Classification de la charge de travail mentale (MWL) avec des signaux EEG.	CNN	Accuracy : 87%
[106]	Classification des états de vigilance à partir de signaux EEG	CNN-LSTM	Accuracy : 85% F1-score : 81% Précision : 80%
[92]	Prédiction des crises d'épilepsie à partir d'EEG intracrânien (iEEG).	CNN	Sensitivité : 89.8%
[93]	Prédiction des crises épileptique focales fondée sur les signaux EEG.	CNN	Sensitivité : 87.8%

Nous avons étudié différentes méthodes de classification des signaux EEG appliquées dans divers contextes médicaux tels que la vigilance, l'épilepsie, la dépression et les troubles neurologiques de la maladie de Parkinson. A l'issue de cette revue de l'état de l'art, nous avons noté une différence importante en termes de performance de classification entre les approches basées sur l'apprentissage automatique traditionnel et celles basées sur l'apprentissage profond, en particulier celles basées sur des réseaux de neurones convolutifs (CNN). En effet, les méthodes traditionnelles d'apprentissage automatique, telles que les méthodes SVM et les k-plus-proches-voisins (KNN), ont montré des performances solides, mais leur dépendance vis-à-vis du traitement d'extraction manuel des caractéristiques est un obstacle souvent rédhibitoire. En revanche, les réseaux de neurones profonds, spécifiquement les CNN, ont démontré leur efficacité en termes de performance. Au vu de ces résultats, notre orientation vers l'utilisation des réseaux de neurones profonds, et plus particulièrement des CNN, se justifie dans notre étude.

7 Description des données EEG et corpus de travail

7.1 Base de données

Sujets : Dans le cadre de cette thèse, nous avons utilisé la base de données EEG collectée sur des volontaires en bonne santé. Ces signaux EEG ont été capturés au Centre de la Vigilance et du Sommeil de la Faculté de Médecine de Monastir en adoptant le même protocole expérimental. Les enregistrements ont été réalisés chez six sujets jeunes sains, âgés de 18 à 23 ans, de sexe masculin, recrutés par contact direct et sur la base d'une adhésion explicite.

Chaque sujet a fait l'objet de 3 enregistrements de 24 heures à 15 jours d'intervalles par un holter EEG. Lors de chaque enregistrement, le volontaire remplit un questionnaire précisant les horaires approximatifs d'endormissement et de réveil nocturne et éventuellement diurne [107], pour que l'expert puisse cibler les zones de transition éveil/sommeil. En ce qui concerne le processus d'anonymisation de la base de données, nous avons adopté une approche courante consistant à supprimer simplement les champs de données contenant des informations personnelles (nom, prénom, date de naissance, numéro de sécurité sociale), qui sont remplacés par un identifiant numérique. Cette base de données est disponible sur demande et après signature d'une convention.

Enregistrements : L'équipement utilisé est un système d'enregistrement ambulatoire de longue durée à 8 voies, modèle OXFORD MEDILOG 9000. L'enregistrement analogique a été réalisé sur cassette magnétique (audio). Les signaux analogiques ont été numérisés et visualisés par un second système de lecture dédié au dispositif ambulatoire. Chaque enregistrement contient deux voies

EOG, une voie EMG au niveau du menton et cinq voies EEG. Pour l'EOG, les électrodes actives sont placées au niveau du canthus externe (à droite et à gauche) avec la référence au niveau de la mastoïde contra-latérale. L'EMG est enregistré par une dérivation bipolaire reliée à deux électrodes disposées au niveau de la houppe du menton, distantes de 2 cm. L'EEG est enregistré par des dérivations bipolaires (F3-F4 ; C3-P3 ; C3-O1 ; C4-P4 et P4-O2). Tous les signaux enregistrés ont été échantillonnés à 128 Hz. Un enregistrement de 24 heures, pour chaque sujet, est retenu par l'expert en tenant compte de la qualité.

Qualifications de l'état de vigilance par l'expert : Les signaux enregistrés ont été analysés et étiquetés par un expert spécialisé dans l'interprétation de l'EEG et de la polysomnographie. La démarche consiste à différencier 8 états annotés :

- Veille attentive : VA
- Veille calme yeux ouverts : VCYO
- Veille calme yeux fermés : VCYF
- Veille calme yeux ouvert artefactée : VCYOAR
- Veille calme yeux fermé artefactée : VCYFAR
- Somnolence : SOM
- Stade 1 de sommeil : STD1
- Mouvement : MV

Dans une deuxième phase, nous avons éliminé les parties artefactées : (VA, VCYO, VCYF, SOM, STD1).

Notre expert, qu'il soit ainsi remercié, a découpé les signaux en périodes de 4 secondes et il a choisi de regrouper les segments d'EEG selon 2 états : vigilance haute (VA, VCYO, VCYF) et vigilance basse (SOM, STD1) [107] [108] [109].

Le tableau ci-dessous résume les résultats de l'analyse et l'étiquetage réalisé par notre expert. Cette base est originale du fait qu'elle est spécifique à l'analyse de la vigilance (zones de transition) par rapport à celles rapportées dans la littérature et elle se démarque surtout par la finesse du découpage (période de 4s).

Tableau 2. 2 : Détails de la Base de Données EEG pour la Classification des États de Vigilance

	<i>Nombre de segments de 4s</i>	<i>Délai en minutes</i>	<i>Nombre des caractéristiques</i>	<i>Classe de sortie</i>	<i>Nombre des classes "veille"</i>	<i>Nombre des classes "sommeil"</i>
Patient 1	232	15 min et 46 s	21	2	109	123
Patient 2	290	19 min et 33 s	21	2	118	172
Patient 3	76	5 min et 6 s	21	2	28	48
Patient 4	362	24 min et 13 s	21	2	190	172
Patient 5	384	25 min et 6 s	21	2	187	197
Patient 6	150	10 min	21	2	60	90

D'un autre côté, elle souffre du manque d'un double étiquetage au regard du choix de cette granularité de 4s et aussi du manque de l'expertise en analyse des états de vigilance. Les durées retenues sont relativement faibles du fait que nous nous sommes intéressés aux zones de transition définies pour chaque patient.

7.2 Prétraitement

Billiard [110] a montré qu'à partir de cette quantification, il est possible d'étudier les activités EEG en fonction de leurs niveaux fréquentiels propres. Le traitement et l'extraction des paramètres des signaux EEG sont réalisés au sein de notre laboratoire [107]. L'analyse consiste à appliquer la Transformée de Fourier Rapide à Court Terme (TFRCT) au signal EEG pour obtenir un spectre dont l'amplitude est en fonction de la fréquence. Ainsi, pour chaque période de 4s du signal EEG, nous avons calculé la puissance spectrale par la TFRCT à travers une fenêtre de pondération de Hamming et une résolution de 512 points (128 échantillons par seconde). À partir de ce spectre, nous ne conservons plus que les fréquences comprises entre 0,1 et 21 Hz. Cet intervalle est caractéristique des ondes physiologiques et supprime la composante continue (DC, "fréquence" de 0 Hz). Puis le spectre de puissance est subdivisé en un ensemble de bandes élémentaires, en sommant les amplitudes spectrales appartenant à chaque bande de fréquence. Enfin, le Pourcentage de Puissance Spectrale Relative (PPSR) est calculé en faisant le rapport de la Puissance de Bande Spectrale (PBS) sur la Puissance Spectrale Totale (PST) comme le montre l'équation (2.1) :

$$PPSR_i = \frac{PBS_{[U_i, U_{i+1}]}}{PST} * 100; \begin{cases} U_i = 0.1 + (i - 1) * \Delta u ; i \in [1, \dots, k] \\ \Delta u = \frac{(21-0.1)}{k} \end{cases} \quad (2.1)$$

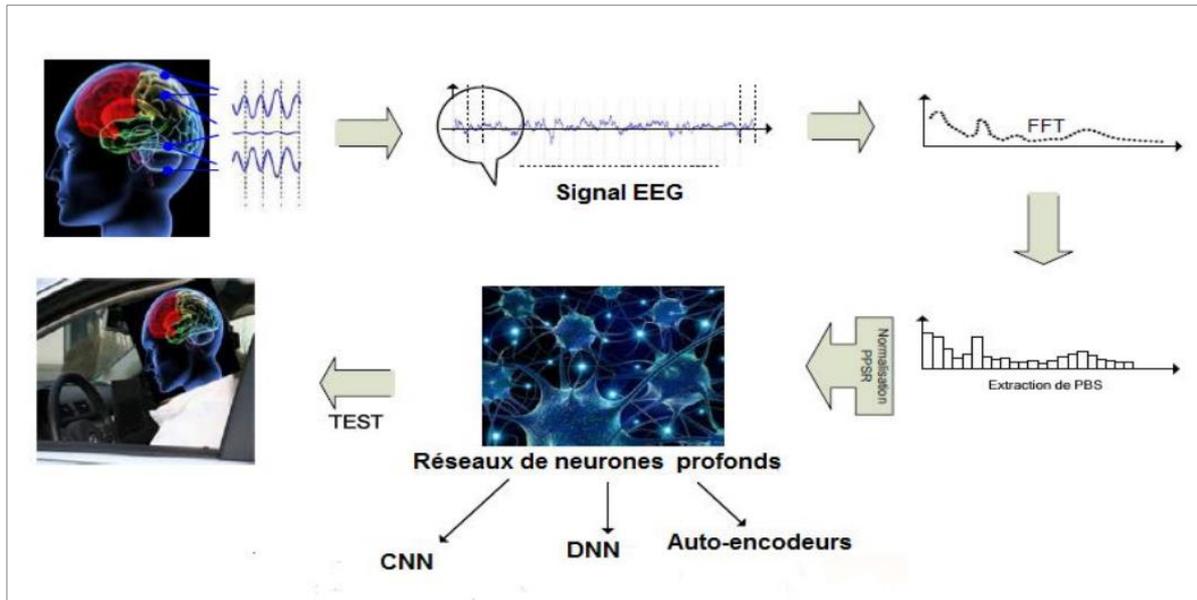


Figure 2. 6: Extraction et analyse des signaux EEG

8 Conclusion

Dans ce chapitre, nous avons décrit le premier cadre d'application de nos travaux : l'analyse des signaux EEG pour la classification des états de vigilance. Nous avons à cet effet décrit les techniques de traitement du signal EEG après acquisition, à savoir les prétraitements et les représentations dans les domaines fréquentiel et spatial.

Par la suite, nous avons réalisé une revue de la littérature portant principalement sur l'exploitation des Réseaux Deep Learning dans le traitement de l'EEG. Cette étude a démontré que les approches basées sur l'apprentissage profond dépassent en termes de performance celles obtenues par les méthodes basées Machine Learning traditionnelles. Cette conclusion suggère le développement de méthodes d'optimisation des hyperparamètres des DNN pour améliorer les résultats de la classification des états de vigilance à l'aide des signaux EEG.

Dans le chapitre suivant, nous introduisons notre méthode d'optimisation des hyperparamètres, puis, nous présentons les architectures DL utilisées et discutons les résultats de classification sur différentes architectures DL.

Chapitre 3

Optimisation des hyperparamètres des Réseaux de Neurons Profonds pour la classification des états de vigilance

1 Introduction

L'apprentissage automatique, en particulier via les développements récents des réseaux de neurones profonds (DNN) constitue une solution efficace pour un large éventail de problèmes complexes. La classification des états de vigilance est une des applications importantes des DNN, car elle constitue un outil précieux de la compréhension et de la surveillance du processus cognitif et physiologique de l'être humain. Cette classification précise revêt une importance fondamentale, que ce soit dans le domaine médical pour diagnostiquer les troubles du sommeil, dans celui de la sécurité routière ou dans celui de la détection de la fatigue cognitive. Toutefois, les performances d'un DNN dépendent largement de la configuration de ses hyperparamètres (HP). L'optimisation minutieuse de ces HP est une étape essentielle dans l'obtention de modèles DNN performants. Dans ce chapitre, nous explorerons l'utilisation des algorithmes d'optimisation dans le contexte de la classification de l'état de vigilance. En outre, nous présenterons une nouvelle technique d'optimisation des HP des réseaux de neurones pour la classification des états de vigilance.

2 Motivation

Ce travail se fonde principalement sur les problématiques liées à la classification des états de vigilance. Sur le plan médical, l'accès à un diagnostic précoce des troubles du sommeil et à une intervention rapide permettrait à des millions de personnes de voir leurs conditions de vie s'améliorer. Toutefois, l'optimisation des HP des modèles de classification est une tâche complexe et souvent pénible à réaliser. En effet, les méthodes classiques sont confrontées à des défis en termes d'efficacité et de capacité à explorer judicieusement l'espace des hyperparamètres. La nouvelle approche hybride que nous allons proposer a pour but de surmonter ces difficultés en combinant de manière adéquate des techniques d'optimisation avancées, en les intégrant de manière cohérente dans le processus d'optimisation. Grâce à cette approche innovante, nous espérons contribuer de

manière significative à améliorer la compréhension des états de vigilance humaine et à créer des systèmes de classification plus efficaces et plus précis dans divers contextes. Ceci offre de nombreuses possibilités dans la recherche et les applications pratiques des signaux physiologiques tels que les EEG.

3 Méthodologie : Nouvelle approche d'optimisation des HP des réseaux DL « GS-TPE »

Nous décrivons ci-après une nouvelle méthode pour optimiser les HP des architectures DL en combinant deux techniques distinctes : la recherche par grille (GS) et l'estimateur d'arbres Parzen (TPE). Notre approche repose sur une combinaison hiérarchique de ces deux méthodes, exploitant les avantages de chacune. Initialement, la méthode GS est utilisée pour une exploration grossière de l'espace des hyperparamètres, où un ensemble de valeurs est défini pour chaque HP à optimiser, et une grille est générée couvrant toutes les combinaisons possibles.

Chaque combinaison est évaluée sur un ensemble de validation à l'aide d'une métrique prédéfinie, soit l'accuracy. Après avoir évalué toutes les combinaisons, nous sélectionnons les 5 meilleures configurations. Ce nombre a été sélectionné en tenant compte des contraintes de temps de calcul, afin de maintenir le processus d'optimisation dans des limites raisonnables en termes de ressource. Chaque configuration parmi ces 5 correspond à une combinaison des différents HPs optimisés par GS, représentant ainsi les meilleures solutions potentielles trouvées par cette méthode. Pour utiliser ces configurations avec l'algorithme TPE, nous définissons des bornes pour chaque HP, permettant ainsi de construire un espace de recherche local autour de chaque configuration. Ensuite, TPE intervient pour affiner la recherche en explorant les voisinages des HPs dans chaque configuration. L'espace des HPs, désigné par \mathbf{x} dans le schéma, est divisé en cinq sous-espaces relatifs aux cinq configurations qui seront évaluées indépendamment par le TPE (**Figure 3. 1**).

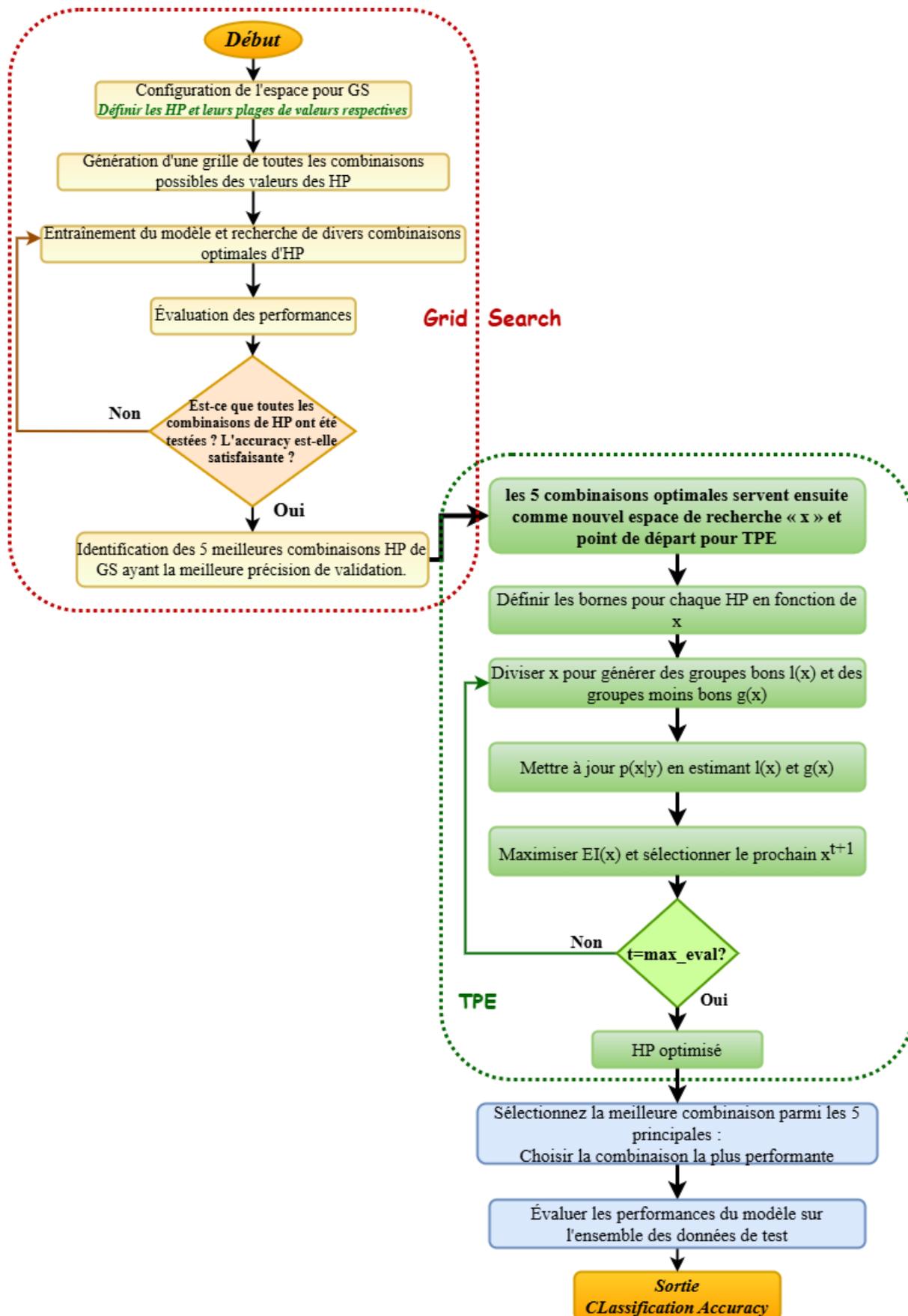


Figure 3. 1: Organigramme de l'approche d'optimisation proposée « GS-TPE »

La stratégie de recherche du TPE comporte deux phases (voir Figure 1. 15 : Pseudo-code de TPE) :

- 1) Lors d'une première phase dite " warm-up ", il explore de manière aléatoire un espace des HP. Les combinaisons de valeurs d'HP choisies sont utilisées pour former un modèle (avec l'ensemble de données d'apprentissage), qui est évalué avec l'ensemble de données de validation pour voir l'impact de chaque combinaison de valeurs d'HP choisie en termes d'accuracy.

La phase "warm-up" a une durée de $n_init = 20$ iterations dans nos expériences (valeur par défaut dans TPE), et permet de modéliser la vraisemblance bayésienne $p(x|y)$ de la valeur de HP x lorsqu'on atteint l'accuracy y . Ce modèle est ensuite réparti en deux probabilités : celle des HP prometteurs $l(x) = p(x|y > y^*)$, et celle des autres $g(x) = p(x|y < y^*)$.

Le seuil d'accuracy y^* est calculé en fonction d'un paramètre γ qui détermine la proportion des meilleurs HP appartenant à la distribution $l(x)$. Dans nos expériences on fixe $\gamma = 0,25$.

- 2) L'objectif de la deuxième phase TPE est de maximiser le ratio d'amélioration attendue $EI(x) = l(x)/g(x)$.

Ceci se fait en échantillonnant n_{EI} combinaisons de valeurs d'HP ($n_{EI} = 24$ dans nos expériences) et en choisissant celle qui maximise le EI. Puis, celle-ci est mémorisée et utilisée lors de l'itération suivante.

Lors de l'itération suivante, la TPE calcule l'accuracy et répartit les combinaisons de valeurs des HP entre les bonnes et les mauvaises, mais cette fois-ci, elle utilise toutes les combinaisons précédentes ainsi que la plus récente. Le processus est répété jusqu'à ce que le nombre déterminé d'essais soit atteint.

Dans nos expériences, nous avons utilisé les paramètres de TPE par défaut (n_init iterations = 20, $\gamma = 0,25$, $n_{EI} = 24$).

Ensuite, TPE commence à explorer le prochain espace d'HPs et répète le processus décrit ci-dessus sur les 5 espaces d'HPs définis dans notre étude. La configuration optimale, celle offrant les meilleures performances, est sélectionnée pour être évaluée sur les données de test.

Grâce à la combinaison hiérarchique de ces deux méthodes, l'exploration exhaustive de GS sert à identifier les combinaisons de HP prometteuses, puis l'exploration plus ciblée et plus efficace de l'espace de recherche est exploitée via le TPE.

4 Architectures DL utilisées pour la classification des états de la vigilance

Dans cette section, nous allons explorer les architectures DNN conçues pour la classification de l'état de vigilance. En particulier, nous nous concentrerons sur l'utilisation des architectures MLP

et LSTM. Cette dernière se prête en effet naturellement au traitement de données séquentielles. En outre, nous examinerons également l'architecture 1D-CNN et les deux architectures 1D-CNN-LSTM et l'AE-LSTM proposées précédemment dans [111], qui se sont révélées prometteuses pour la classification des états de vigilance.

4.1 Architecture MLP

La première architecture choisie est le réseau MLP. Il se caractérise par trois couches entièrement connectées, chacune jouant un rôle important pour capturer et traiter les caractéristiques pertinentes des données d'entrée. Entre ces couches, deux couches Dropout ont été ajoutées pour régulariser le modèle et réduire le surajustement en désactivant de manière aléatoire certains neurones au cours de leur apprentissage. Enfin, une couche à deux neurones de sortie est utilisée en sortie pour classer les états de veille et de sommeil.

Les HPs à optimiser pour cette architecture sont le nombre de neurones dans chacune des trois couches MLP, les deux taux Dropout-1 et Dropout-2, le taux d'apprentissage et le batch size.

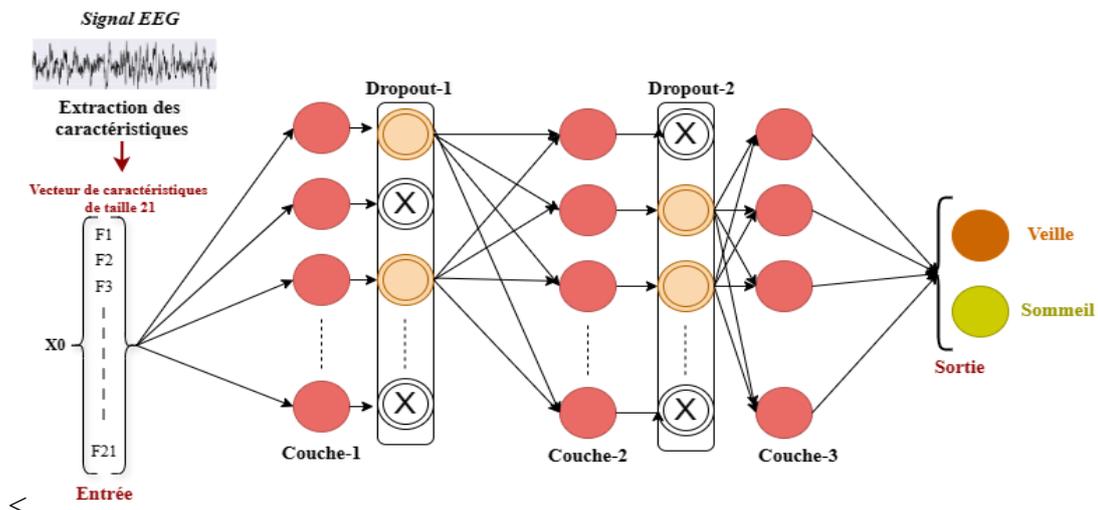


Figure 3. 2 : Architecture MLP

4.2

4.3 Architecture LSTM

Les réseaux LSTM sont des réseaux de neurones récurrents particulièrement adaptés au traitement des données séquentielles. Ils sont conçus pour capturer les dépendances temporelles à long terme en utilisant des cellules mémoire qui permettent de conserver l'information pertinente sur de longues séquences. Grâce à leur structure unique, les LSTM sont capables de gérer les problèmes de gradient qui peuvent survenir dans les réseaux récurrents classiques.

Dans notre travail, nous avons utilisé un réseau LSTM empilé composé de quatre couches pour analyser des données temporelles. Plus spécifiquement, nous injectons des vecteurs de caractéristiques de taille 21 dans ce réseau. Ces vecteurs de caractéristiques sont obtenus après un prétraitement appliqué à chaque segment de 4 secondes du signal brut. Ce prétraitement permet d'extraire des caractéristiques spécifiques, chaque caractéristique représentant 21 bandes de fréquence de 1 Hz aboutissant ainsi à un vecteur de taille 21 pour chaque intervalle de 4 secondes. La Figure 3. 3 illustre le fonctionnement détaillé de notre modèle. Chaque vecteur de caractéristiques est introduit dans la première couche LSTM (LSTM-1). Cette couche traite les données en calculant des activations basées sur les entrées actuelles et l'état caché de cette couche, qui est initialisé à zéro lors du traitement du premier vecteur. Les activations générées par LSTM-1 sont ensuite transmises à la deuxième couche LSTM (LSTM-2). La couche LSTM-2 poursuit ce traitement en utilisant les activations reçues de LSTM-1 ainsi que l'état caché de l'étape temporelle précédente pour ces calculs. Ce processus est répété pour la troisième (LSTM-3) et la quatrième couche (LSTM-4), chaque couche affinant successivement l'analyse des relations temporelles entre les caractéristiques extraites. Ainsi, il est important de souligner que les paramètres des différentes couches LSTM-1, LSTM-2, LSTM-3 et LSTM-4 se distinguent les uns des autres. Chaque couche LSTM possède ses propres paramètres et effectue des transformations spécifiques sur les données reçues, permettant ainsi de capturer de manière plus complexe et plus riche la dynamique temporelle. La récurrence s'opère au sein de chaque couche pour traiter efficacement les séquences temporelles. Par ailleurs, entre les couches LSTM-2 et LSTM-3, nous avons ajouté une couche dropout. Cette dernière aide à prévenir le surapprentissage.

À chaque étape de temps, une décision est prise. Pour classifier les données, nous employons une seule couche entièrement connectée (FC). Lors du traitement des vecteurs suivants, le même processus est répété, mais cette fois, les états cachés des couches LSTM sont ceux calculés à partir du vecteur précédent. Cela signifie que les informations importantes extraites du vecteur de caractéristiques à l'étape n-1 sont utilisées lors du traitement du vecteur de caractéristiques à l'étape n. Ce transfert d'états cachés d'une étape à l'autre permet de capturer les dépendances temporelles et d'affiner l'analyse des relations entre les caractéristiques extraites à travers les différentes couches du modèle.

Les HPs sélectionnés à optimiser dans cette architecture sont le nombre d'unités dans chaque couche LSTM, le taux de dropout, ainsi que les HPs liés au processus d'apprentissage tels que le taux d'apprentissage et la taille du lot.

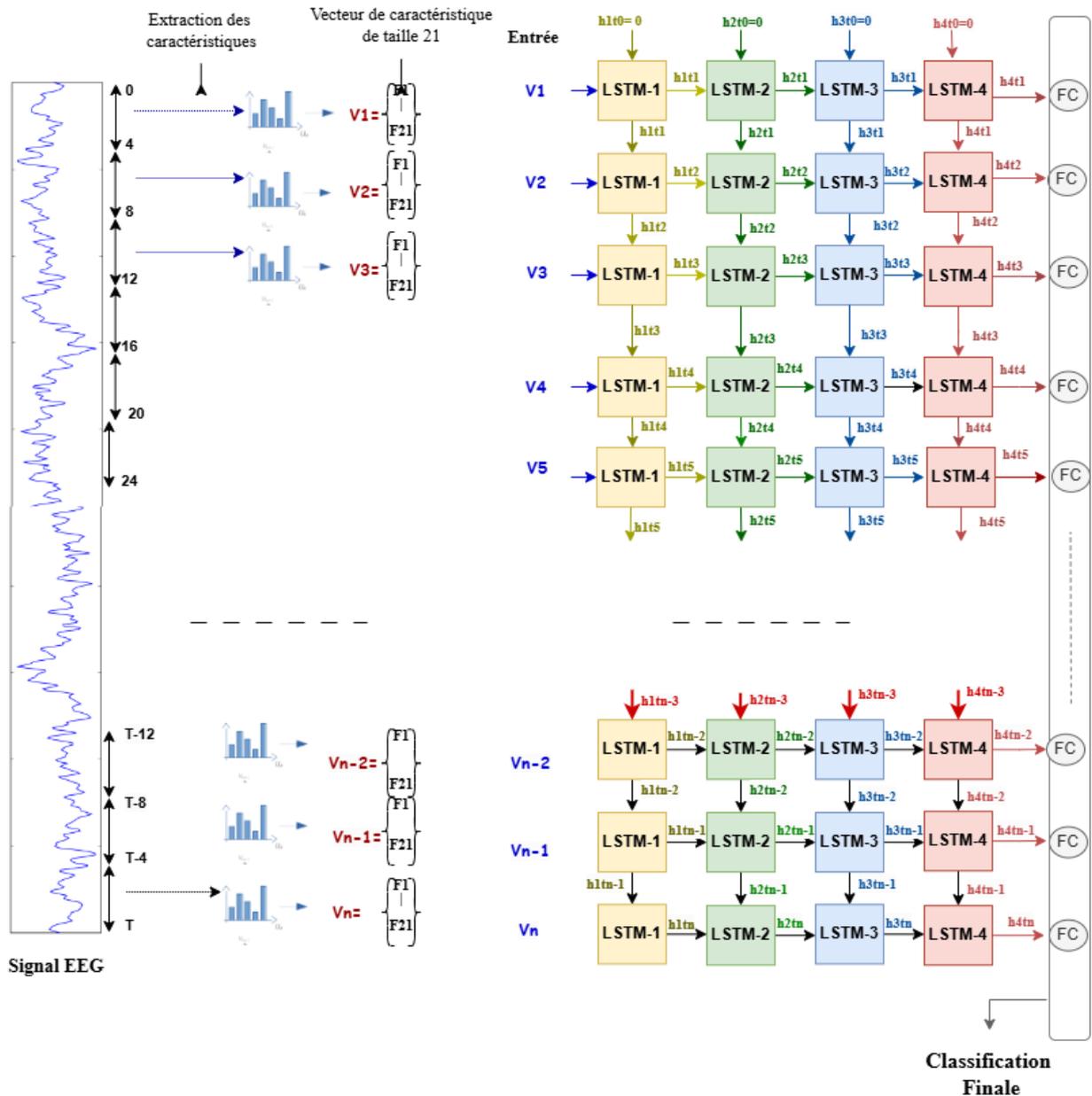


Figure 3.3 : Architecture LSTM détaillé

4.4 Architecture 1D-CNN

Le réseau 1D-CNN proposé pour la classification des états de vigilance à partir des signaux EEG se compose de trois couches de convolution, chacune suivie d'une couche de maxpooling, permettant l'extraction efficace des caractéristiques pertinentes et une couche de dropout. Après ces opérations, une couche Flatten prépare les données pour être traitées par une couche entièrement connectée, facilitant la fusion des caractéristiques extraites avant la classification finale. Il est conçu pour deux classes distinctes, ce qui correspond à l'état de veille et à l'état de sommeil (Figure 3.4). Les HPs sélectionnés pour l'optimisation sont le nombre de filtres dans chacune des couches de convolution, ainsi que le taux de Dropout, le taux d'apprentissage et la taille du lot « batch size ».

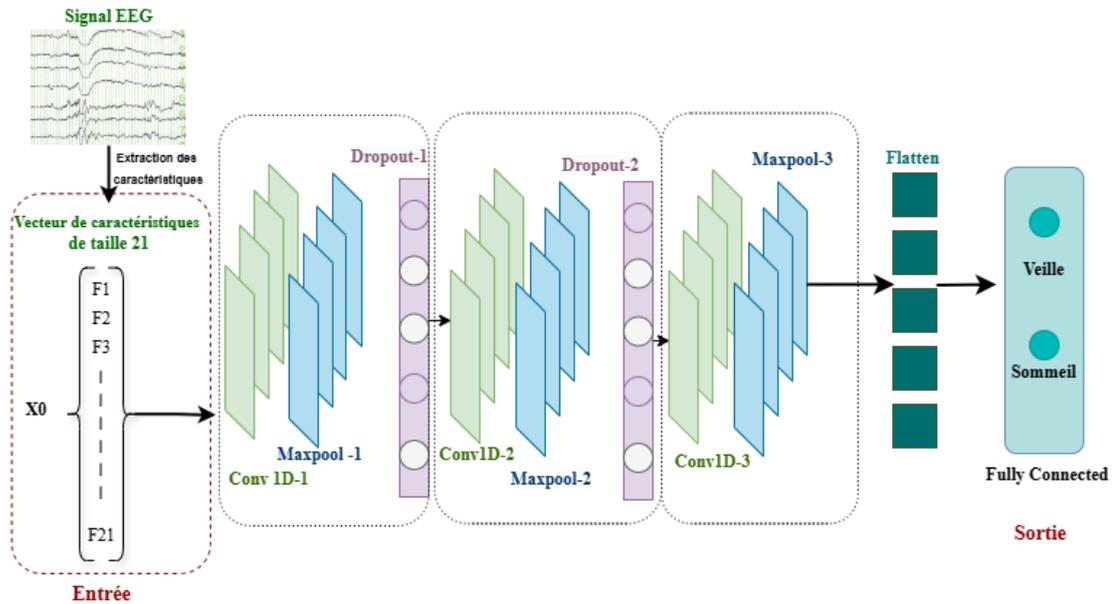


Figure 3. 4 : Architecture 1D-CNN

4.5 Architecture 1D-CNN-LSTM

Les signaux EEG représentent un ensemble de données séquentielles recueillies au fil du temps, ce qui exige des techniques avancées pour une analyse efficace. L'architecture hybride CNN-LSTM (voir Figure 3. 5) apparaît être une solution intéressante pour la classification de la vigilance, en combinant les avantages des réseaux CNN et des réseaux LSTM. D'une part, les CNN sont intégrés pour leur capacité à identifier et à extraire des caractéristiques spécifiques à partir de données d'entrée. Les vecteurs de caractéristiques sont traités par un CNN constitué de trois blocs, chacun comprenant une alternance de couches de convolution-1D et de Pooling. Les deux premiers blocs intègrent également des couches de Dropout pour prévenir le surapprentissage et améliorer la généralisation du modèle. Les caractéristiques extraites par le CNN sont ensuite introduites dans un réseau LSTM empilé, composé de trois couches. Les LSTM sont particulièrement efficaces pour capturer les dépendances temporelles dans les séquences de données EEG, grâce à leur capacité à mémoriser des informations sur de longues périodes et à traiter les séquences avec des dépendances à long terme. Ces réseaux utilisent des cellules de mémoire et des mécanismes de portes pour prévenir le problème de gradient vanishing, permettant ainsi une meilleure conservation des informations pertinentes au fil du temps.

Les HPs sélectionnés pour l'optimisation sont le nombre de filtres dans chacune des 3 premières couches de convolution pour le CNN, le nombre d'unités des couches LSTM pour le réseau LSTM.

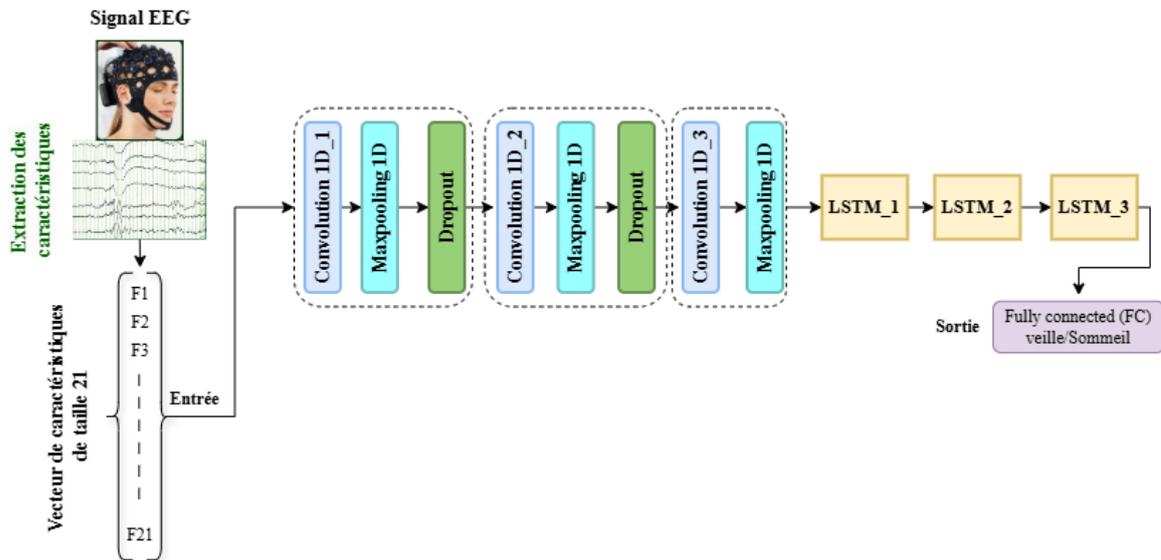


Figure 3. 5 : Architecture 1D-CNN-LSTM

4.6 Architecture AE-LSTM

L'architecture hybride AE-LSTM combine la partie encodeur d'un auto-encodeur avec un réseau LSTM. Dans un premier temps, l'auto-encodeur est entraîné à apprendre une représentation latente à dimension réduite des signaux EEG. Où chaque vecteur de cette représentation latente possède 6 caractéristiques, la taille de cet espace latent ayant été déterminée de manière empirique. Cette représentation, contenant les informations essentielles pour reconstruire les signaux, La sortie de l'encodeur (l'espace latent) sert alors d'entrée pour le réseau LSTM, composé de quatre couches. Ensuite, une seule couche entièrement connectée (FC) avec deux neurones est utilisée pour classifier les deux états : veille et sommeil. (Figure 3. 6).

Parmi les aspects intéressants de cette combinaison, les auto-encodeurs sont conçus pour apprendre les représentations latentes des données d'entrée. Grâce à l'utilisation de l'encodeur pour extraire une représentation latente des signaux EEG, il est possible de réduire considérablement la taille des données d'entrée tout en préservant des informations pertinentes. L'utilisation de LSTM permet pour sa part de capter les dépendances temporelles qui existent dans le signal EEG. Grâce à leurs cellules de mémoire et à leurs mécanismes, ils sont capables de stocker des informations sur de longues périodes et de traiter des séquences avec des dépendances à long terme.

Pour l'optimisation de cette architecture, les HPs à régler concernent principalement le réseau LSTM. Il s'agit du nombre d'unités dans chaque couche LSTM et des HPs associés à l'apprentissage du réseau, tels que le taux d'apprentissage, et la taille du lot.

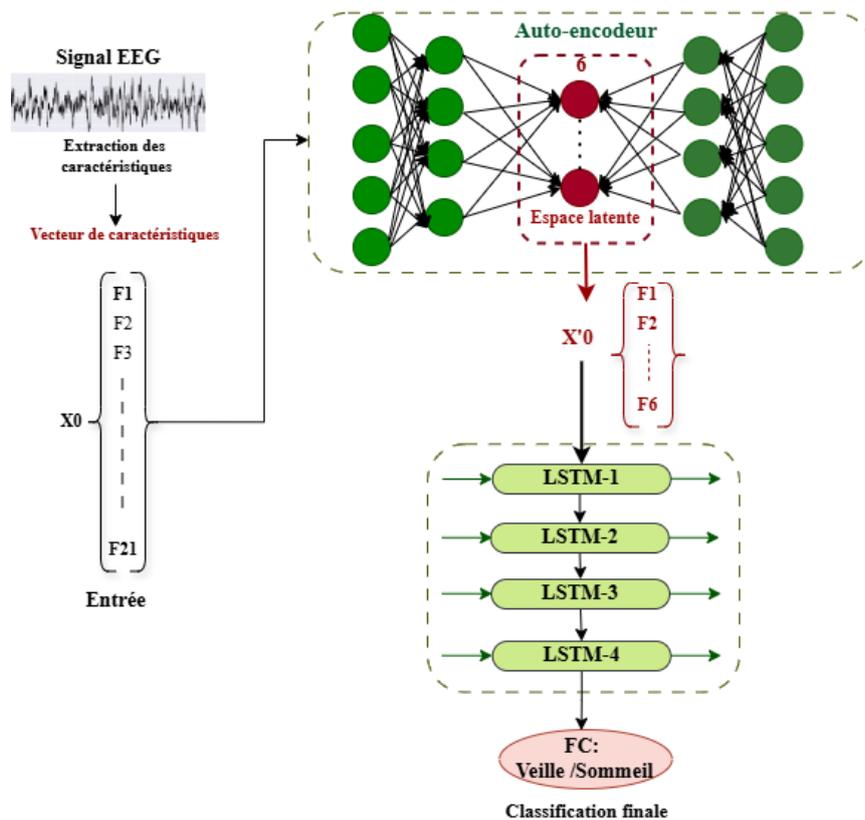


Figure 3. 6: Architecture AE-LSTM

5 HPs à optimiser

La structure des réseaux de neurones comporte un grand nombre d'HP (Tableau 3. 1). La première catégorie est liée à la configuration du modèle. Étant donné que tous les modèles de réseaux de neurones comportent une couche d'entrée et une couche de sortie, la complexité d'un modèle DNN dépend principalement du nombre de couches cachées et du nombre de neurones dans chaque couche, qui sont les deux principaux facteurs de performance du point de vue de l'architecture des modèles. Ceux-ci sont définis et adaptés en fonction de la complexité des données.

Tableau 3. 1 : Les HP d'une architecture DL pour la classification des états de vigilance

<i>Hyperparamètres liés à la configuration du modèle</i>	<i>Hyperparamètres liés à l'apprentissage</i>
Nombre des neurones par couche	Optimiseur
Nombre d'unités pour les couches LSTM	Batch size
Nombre des couches de convolution	Learning rate
Nombre de filtres	
Pool size	
Kernel size	
Dropout rate	
Fonction d'activation	

La capacité des modèles doit être suffisante pour modéliser les fonctions objectifs (ou les tâches de prédiction) en évitant le surapprentissage. Ensuite, certains types de fonctions doivent être déterminés ou ajustés. La première à configurer est le type de la fonction de perte, qui est principalement choisie selon le type de problème (par exemple, l'entropie croisée binaire pour les problèmes de classification binaire, l'entropie croisée multiclasse pour les problèmes de classification multiclasse et la RMSE pour les problèmes de régression). De plus certains HPs tels que le nombre de filtres et la taille du noyau sont des HPs clés pour les réseaux de type CNNs. Le nombre de filtres influe sur les caractéristiques extraites, le kernel size agit sur la dimension spatiale des modèles appris et la taille du Pooling permet de réduire la dimension spatiale tout en préservant les caractéristiques essentielles. Ces hyperparamètres jouent un rôle essentiel dans la capacité du modèle à extraire des informations pertinentes et à contrôler la dimensionnalité des données. Une autre caractéristique importante des HP est le type de fonction d'activation non linéaire employé, qui peut être défini comme Rectified Linear Unit (ReLU), Sigmoid ou Tangente Hyperbolique (tanh). Enfin, le type d'optimiseur peut aussi varier : SGD, Adam, Rmsprop, etc. Par ailleurs, certains autres HP sont associés au processus d'apprentissage des modèles DNN. Le taux d'apprentissage est l'un des HP essentiels des modèles DNN [112]. Ce dernier détermine la longueur des pas à chaque itération de descente de gradient, qui joue un rôle essentiel dans la convergence des modèles. Un taux d'apprentissage élevé accélère le processus d'apprentissage, cependant le gradient peut osciller autour d'un minimum local, ou ne pas converger du tout.

En revanche, un faible taux d'apprentissage favorise une convergence régulière, mais augmente considérablement la durée d'apprentissage du modèle en exigeant plus d'époques pour l'apprentissage. Le taux d'apprentissage adéquat doit permettre à la fonction objective de converger vers un minimum global en un temps raisonnable. Un autre hyper paramètre courant est le taux de "dropout". Le dropout est une méthode de régularisation classique des modèles DL, destinée à réduire l'overfitting. Lors de cette opération, la mise à jour des poids d'une partie des neurones est inhibée de manière aléatoire. Dans notre travail, nous nous concentrons sur l'optimisation des HPs des différentes architectures à l'aide de plusieurs algorithmes d'optimisation, notamment GS, RS et TPE. Les tableaux ci-après, détaillent les HPs à optimiser pour chaque architecture à savoir MLP, LSTM, 1D-CNN, 1D-CNN-LSTM et AE-LSTM, ainsi que l'espace de recherche définis pour GS, le pas utilisé et le nombre total de combinaisons à évaluer pour GS. Le choix des intervalles a été déterminé en garantissant que les valeurs empiriques utilisées dans le processus sans optimisation sont incluses dans chaque intervalle. Pour l'algorithme RS, nous utiliserons le même domaine que GS, mais nous testons de manière aléatoire une sélection de valeurs au lieu de toutes les valeurs de la grille. Nous avons sélectionné les HPs les plus influents pour chaque architecture afin de les

optimiser. Parmi ces HPs figurent le taux d'apprentissage, choisi pour toutes les architectures, le nombre de filtres pour les architectures basées sur les réseaux à convolution à savoir le 1D-CNN et le 1D-CNN-LSTM, ainsi que le nombre d'unités dans les couches LSTM pour les architectures LSTM et AE-LSTM. Le choix du batch size a également été inclus dans cette sélection.

Tableau 3. 2 : HPs clés à optimiser dans l'architecture MLP et leurs espaces de recherche pour les deux algorithmes GS et TPE

<i>Architecture</i>	<i>Hyperparamètres</i>	<i>Domaine pour GS</i>	<i>Domaine pour TPE</i>
MLP	Nbre_neurone_couche-1	{5,25, Step=10}	{5, 25}
	Nbre_neurone_couche-2	{5,25, Step=10}	{5, 25}
	Nbre_neurone_couche-3	{5,25, Step=10}	{5, 25}
	Dropout_1	{0.1,0.5, Step=0.2}	{0.1, 0.5}
	Dropout_2	{0.1, 0.5, Step=0.2}	{0.1, 0.5}
	Batch size	{16,128, Step= 2^n , $n \in \{4,5,6,7\}$ }	{16,128}
	Learning rate	$\{\mathbb{R}[10^{-2}, 10^{-4}], \text{Step}=10^{-1}\}$	$\mathbb{R}[10^{-2}, 10^{-4}]$
		2916 Combinaisons	

Tableau 3. 3 : HPs clés à optimiser dans l'architecture LSTM et leurs espaces de recherche pour les deux algorithmes GS et TPE

<i>Architecture</i>	<i>Hyperparamètres</i>	<i>Domaine pour GS</i>	<i>Domaine pour TPE</i>
LSTM	Unités_LSTM-1	{32,128, Step= 2^n , $n \in \{5,6,7\}$ }	{32,128}
	Unités_LSTM-2	{32,128, Step= 2^n , $n \in \{5,6,7\}$ }	{32,128}
	Unités_LSTM-3	{16,64, Step= 2^n , $n \in \{4,5,6\}$ }	{16,64}
	Unités_LSTM-4	{8,32, Step= 2^n , $n \in \{3,4,5\}$ }	{8, 32}
	Dropout	{0.1, 0.5, Step=0.2}	{0.1, 0.5}
	Batch size	{32,128, Step= 2^n , $n \in \{5,6,7\}$ }	{32,128}
	Learning rate	$\{\mathbb{R}[10^{-2}, 10^{-4}], \text{Step}=10^{-1}\}$	$\mathbb{R}[10^{-2}, 10^{-4}]$
		2187 Combinaisons	

Tableau 3. 4 : HPs clés à optimiser dans l'architecture 1D-CNN et leurs espaces de recherche pour les deux algorithmes GS et TPE

<i>Architecture</i>	<i>Hyperparamètres</i>	<i>Domaine pour GS</i>	<i>Domaine pour TPE</i>
1D-CNN	Filtre_Conv-1D-1	{64,512, Step= 2^n , $n \in \{7,8,9\}$ }	{64,512}
	Filtre_Conv-1D-2	{64,512, Step= 2^n , $n \in \{7,8,9\}$ }	{64,512}
	Filtre_Conv-1D-3	{64,512, Step= 2^n , $n \in \{7,8,9\}$ }	{64,512}
	Dropout_1	{0.3,0.5, Step=0.2}	{0.3,0.5}
	Batch size	{16,64, Step= 2^n , $n \in \{4,5,6\}$ }	{16,64}
	Learning rate	$\{\mathbb{R}[10^{-2}, 10^{-4}], \text{Step}=10^{-1}\}$	$\mathbb{R}[10^{-2}, 10^{-4}]$
		1152 Combinaisons	

Tableau 3. 5: HPs clés à optimiser dans l'architecture 1D-CNN-LSTM et leurs espaces de recherche pour les deux algorithmes GS et TPE

<i>Architecture</i>	<i>Hyperparamètres</i>	<i>Domaine pour GS</i>	<i>Domaine pour TPE</i>
<i>1D-CNN-LSTM</i>	Filtre_Convolution_1D-1	{32,128, Step= 2^n , $n \in \{5,6,7\}$ }	{32,128}
	Filtre_Convolution_1D-2	{32,128, Step= 2^n , $n \in \{5,6,7\}$ }	{32,128}
	Filtre_Convolution_1D-3	{64,256, Step= 2^n , $n \in \{6,7,8\}$ }	{64,256}
	Unités_LSTM_1	{10,50, Step=20}	{10, 50}
	Unités_LSTM_2	{10,50, Step=20}	{10, 50}
	Unités_LSTM_3	{5,25, Step=10}	{5, 25}
	Learning rate	{ $\mathbb{R}[10^{-2}, 10^{-4}]$, Step= 10^{-1} }	$\mathbb{R}[10^{-2}, 10^{-4}]$
		2187 Combinaisons	

Tableau 3. 6 : HPs clés à optimiser dans l'architecture AE-LSTM et leurs espaces de recherche pour les deux algorithmes GS et TPE

<i>Architecture</i>	<i>Hyperparamètres</i>	<i>Domaine pour GS</i>	<i>Domaine pour TPE</i>
<i>AE-LSTM</i>	Unités_LSTM_1	{30,50, Step=10}	{30,50}
	Unités_LSTM_2	{30,50, Step=10}	{30,50}
	Unités_LSTM_3	{30,50, Step=10}	{30,50}
	Unités_LSTM_4	{5,25, Step=10}	{5,25}
	Batch size	{16,128, Step= 2^n , $n \in \{4,5,6,7\}$ }	{16,128}
	Learning rate	{ $\mathbb{R}[10^{-2}, 10^{-4}]$, Step= 10^{-1} }	$\mathbb{R}[10^{-2}, 10^{-4}]$
		972 Combinaisons	

Pour la nouvelle approche GS-TPE, le tableau ci-dessous présente les HPs clés à optimiser pour les architectures de modèles utilisées, à savoir MLP, LSTM, 1D-CNN, 1D-CNN-LSTM et AE-LSTM, ainsi que les plages d'exploration associées pour chaque modèle. Lors de l'application de GS-TPE, nous avons maintenu les mêmes HPs et plages de valeurs que celles utilisées pour GS seul et TPE seul, afin de faciliter une comparaison cohérente des résultats. Avec GS-TPE, nous prévoyons de sélectionner les cinq meilleures configurations d'HPs identifiées par la GS. Pour ce faire, nous avons défini des bornes autour de chaque HP dans chaque configuration, permettant ainsi à TPE d'explorer les environs de ces valeurs pour chaque configuration pré-sélectionnée. Le choix des bornes a été défini de manière empirique en garantissant que les valeurs ne tombent pas en dessous de 0 pour les bornes minimales, et en fixant des bornes supérieures de manière à couvrir une plage raisonnable autour des valeurs optimales identifiées par la GS.

Tableau 3.7 : HPs et plages d'exploration pour l'approche d'optimisation GS-TPE

<i>Architecture</i>	<i>Hyperparamètres</i>	<i>Domaine</i>	<i>Step pour GS</i>	<i>Bornes des HP utilisées par TPE</i>
<i>MLP</i>	Nbre_neurone_couche-1	{5, 25}	10	± 4
	Nbre_neurone_couche-2	{5, 25}	10	± 4
	Nbre_neurone_couche-3	{5, 25}	10	± 4
	Dropout_1	{0.1, 0.5}	0.2	± 0.1
	Dropout_2	{0.1, 0.5}	0.2	± 0.1
	Batch size	{16,128}	2^n , $n \in \{4,5,6,7\}$	± 10
	Learning rate	$\mathbb{R}[10^{-2}, 10^{-4}]$	10^{-1}	± 10^{-5}
<i>LSTM</i>	Unités_LSTM-1	{32,128}	2^n , $n \in \{5,6,7\}$	± 20
	Unités_LSTM-2	{32,128}	2^n , $n \in \{5,6,7\}$	± 20
	Unités_LSTM-3	{16,64}	2^n , $n \in \{4,5,6\}$	± 20
	Unités_LSTM-4	{8, 32}	2^n , $n \in \{3,4,5\}$	± 10
	Dropout	{0.1, 0.5}	0.2	± 0.1
	Batch size	{32,128}	2^n , $n \in \{4,5,6,7\}$	± 16
	Learning rate	$\mathbb{R}[10^{-2}, 10^{-4}]$	10^{-1}	± 10^{-5}
<i>1D-CNN</i>	Filtre_Conv-1D-1	{64,512}	2^n , $n \in \{7,8,9\}$	± 20
	Filtre_Conv-1D-2	{64,512}	2^n , $n \in \{7,8,9\}$	± 20
	Filtre_Conv-1D-3	{64, 512}	2^n , $n \in \{7,8,9\}$	± 20
	Dropout_1	{0.3,0.5}	0.2	± 0.2
	Batch size	{16,64}	2^n , $n \in \{4,5,6\}$	± 10
	Learning rate	$\mathbb{R}[10^{-2}, 10^{-4}]$	10^{-1}	± 10^{-5}
	<i>1D-CNN-LSTM</i>	Filtre_Convolution_1D-1	{32,128}	2^n , $n \in \{5,6,7\}$
Filtre_Convolution_1D-2		{32,128}	2^n , $n \in \{5,6,7\}$	± 12
Filtre_Convolution_1D-3		{64,256}	2^n , $n \in \{6,7,8\}$	± 12
Unités_LSTM_1		{10,50}	20	± 5
Unités_LSTM_2		{10,50}	20	± 5
Unités_LSTM_3		{5,25}	10	± 3
Learning rate		$\mathbb{R}[10^{-2}, 10^{-4}]$	10^{-1}	± 10^{-5}
<i>AE_LSTM</i>	Unités_LSTM_1	{30, 50}	10	± 6
	Unités_LSTM_2	{30, 50}	10	± 6
	Unités_LSTM_3	{30,50}	10	± 6
	Unités_LSTM_4	{5, 25}	10	± 3
	Batch size	{16,128}	2^n , $n \in \{4,5,6,7\}$	± 10
	Learning rate	$\mathbb{R}[10^{-2}, 10^{-4}]$	10^{-1}	± 10^{-5}

6 Implémentation et résultats

6.1 Environnement de travail

Nous avons évalué les algorithmes d'optimisation des HPs avec les différentes architectures. Ces dernières ont été implémentées en utilisant Keras dont les bibliothèques sont écrites en Python. La configuration matérielle comprenait un GPU NVIDIA GeForce RTX 3090 avec 32 Go et un

processeur 11e Gen Intel® core™ i9-11900F. Pour aborder les problèmes d'optimisation des HP, nous avons utilisé le Framework Optuna, qui fournit de nombreux algorithmes, dont la TPE.

6.2 Processus de Préparation des Données EEG pour l'Entraînement des Modèles

Les modèles LSTM nécessitent des séquences de données à l'entrée, étant donné qu'ils sont conçus pour traiter et apprendre les dépendances temporaires au sein des séquences temporelles. Dans notre cas, nous travaillons avec des vecteurs de caractéristiques extraites de signaux EEG bruts. Pour entraîner efficacement nos modèles fondés sur LSTM, à savoir LSTM, CNN-LSTM et AE-LSTM, il est donc essentiel de créer des séquences à partir de ces caractéristiques EEG extraites. Ainsi, nous avons adopté la technique de la fenêtre glissante. Cette méthode consiste à générer des séquences de caractéristiques à l'aide d'une fenêtre de longueur fixe qui se déplace sur l'ensemble des données. Chaque position de la fenêtre produit une nouvelle séquence, ce qui garantit que des données contiguës dans le temps sont pleinement utilisées pour l'entraînement.

Le Tableau 3. 8 détaille le processus de création des séquences pour chaque patient, avec une séquence de taille fixe de 5. Il présente le nombre initial d'échantillons (vecteurs de caractéristiques) ainsi que le nombre de séquences générées.

Pour le mode monosujet, nous avons divisé et préparé les séquences en trois ensembles distincts, apprentissage, validation et test.

Dans le Tableau 3. 9 qui présente les informations du mode multi-sujet, les séquences de chaque patient ont été regroupées, permettant de travailler sur l'intégralité de la base de données. Nous disposons de 1494 vecteurs de caractéristiques de taille 21, à partir desquels 1470 séquences de 5 périodes et 1440 et 10 périodes chevauchées ont été créées. Ces séquences ont ensuite été divisées en 60 % pour l'apprentissage, 20 % pour la validation et 20 % pour le test.

Tableau 3. 8 : Création et répartition des séquences par patient (mode monosujet)

	<i>Nombre d'échantillons de 4s</i>	<i>Nombre de séquences chevauchées de taille 5</i>	<i>Nombre de séquence pour l'entraînement (80%)</i>	<i>Nombre de séquence pour la validation (10%)</i>	<i>Nombre de séquence pour la partie Test (10%)</i>
<i>Patient 1</i>	232	228	182	23	23
<i>Patient 2</i>	290	286	228	29	29
<i>Patient 3</i>	76	72	57	8	7
<i>Patient 4</i>	362	358	286	36	36
<i>Patient 5</i>	384	380	304	38	38
<i>Patient 6</i>	150	146	116	15	15

Nous avons également divisé les données en séquences de taille 10 pour entraîner l'architecture AE-LSTM avec des séquences de tailles différentes, et évalué les résultats. Étant donné que nous utilisons l'espace réduit de l'auto encodeur comme entrée pour le LSTM, augmenter la taille des séquences peut être avantageux, car le nombre de caractéristiques a été réduit. Cela nous permettra de déterminer si cette stratégie améliore les performances globales du modèle.

Tableau 3. 9 : : Création et répartition des séquences pour le mode multi sujet

	<i>Nombre d'échantillons de 4s</i>	<i>Nombre de séquences chevauchées</i>	<i>Nombre de séquences totales</i>	<i>Nombre de séquences pour l'entraînement (60%)</i>	<i>Nombre de séquences pour la validation (20%)</i>	<i>Nombre de séquences pour la partie Test (20%)</i>
Base complet	1494	5	1470	882	294	294
		10	1440	864	288	288

6.1 Paramétrage des expériences pour l'optimisation des HP

Nous présentons dans cette section le paramétrage des expériences d'optimisation pour les différentes architectures de réseaux étudiées. Chaque architecture, que ce soit MLP, LSTM, CNN, CNN-LSTM ou AE-LSTM, a été associée à un certain nombre de combinaisons GS et à un certain nombre d'itérations RS et TPE. Pour assurer une comparaison équitable et cohérente entre les différentes méthodes d'optimisation, nous avons établi des paramètres standardisés pour chacune d'entre elles. Nous avons choisi de fixer le nombre d'itérations pour RS à un tiers du nombre de combinaisons GS ($N_{RS} = N_{GS} / 3$). De même, nous avons appliqué 350 itérations pour l'algorithme TPE (voir Tableau 3. 10).

Nous utiliserons les mêmes paramètres pour l'algorithme GS-TPE que ceux utilisés pour GS et TPE lorsqu'ils sont employés indépendamment. De plus, nous expérimenterons le RS-TPE, qui fonctionne de manière similaire au GS-TPE, mais avec une recherche aléatoire plutôt qu'exhaustive, en utilisant les mêmes paramètres que pour RS et TPE lorsqu'ils sont employés séparément.

Tableau 3. 10 : Paramétrage des expérimentations effectuées

<i>Architectures</i>	<i>Nombre de combinaisons GS « N_{GS} »</i>	<i>Nombre d'itérations RS « $N_{GS}/3$ »</i>	<i>Nombre d'itérations TPE</i>
MLP	2916	972	350
LSTM	2187	729	350
1D-CNN	1152	384	350
CNN-LSTM	2187	729	350
AE-LSTM	972	324	350

6.2 Métriques de performance

L'analyse quantitative permet d'évaluer statistiquement la performance des résultats de classification par rapport à celle fournie par l'expert. Nous avons choisi d'utiliser l'exactitude, la précision et la sensibilité dans notre cas. Celles-ci sont basées sur les quatre paramètres :

- Vrai-Positif (VP)
- Vrai-Négatif (VN)
- Faux-Positif (FP)
- Faux-Négatif (FN)

Ils sont définis comme suit :

- VP : le nombre des échantillons prédits correctement qui appartiennent à la classe vigilante.
- VN : le nombre des échantillons bien prédits faisant partie de la classe non-vigilant.
- FP : le nombre des échantillons prédits incorrectement appartenant à la classe vigilante.
- FN : le nombre des échantillons mal prédits qui appartiennent à la classe non-vigilant.

Une classification idéale est celle qui permet d'atteindre 100% d'exactitude, de sensibilité, de spécificité et de précision. L'accuracy est calculée par l'équation (3. 1), la sensibilité par l'équation (3. 2), le F1-Score par l'équation (3. 3), la précision par l'équation (3. 4) comme suit :

$$Accuracy (Acc) = \frac{Vp+Vn}{Vp+Vn+Fp+Fn} * 100\% \quad (3. 1)$$

$$Sensibilité (Sen) = \frac{Vp}{Vp+Fn} * 100\% \quad (3. 2)$$

$$F1 - Score (F1) = 2 \frac{Pr*Sen}{Pr+Sen} * 100\% \quad (3. 3)$$

$$Précision (Pr) = \frac{Vp}{Vp+Fp} * 100\% \quad (3. 4)$$

6.3 Résultats expérimentaux et discussion

6.3.1 Résultats sans processus d'optimisation

Cette section expose les résultats obtenus avec les différentes architectures. Nos expérimentations sont déroulées en trois modes distincts :

- Le mode monosujet
- Le mode multi-sujets "Leave one patient out"
- Le mode multi-sujets "Train/Val/Test"

A. Mode mono-sujets

Ce mode d'évaluation consiste à analyser individuellement les performances de nos modèles entraînés sur chaque patient. Les données ont été divisées en ensembles d'entraînement (80%), ensembles de validation (10%) et ensembles de test (10%) pour chaque sujet, permettant ainsi une évaluation précise de la performance de chaque modèle pour chaque patient.

Le Tableau 3. 11 présente les résultats de classification pour chaque patient. Ces résultats mettent en évidence une variabilité significative des performances des différents modèles en fonction des patients. Prenons l'exemple du MLP où les précisions varient notablement, illustrant sa sensibilité aux particularités des données de chaque patient. Par exemple, pour le patient 3, le MLP atteint une précision de seulement 50.00%, tandis que pour le patient 1, cette accuracy monte à 86.95%, soulignant ainsi l'impact significatif de la variabilité individuelle des données sur les performances du modèle.

Cette variabilité se reflète aussi dans les performances de LSTM. Par exemple, l'architecture LSTM atteint une accuracy de 71.42 % pour le patient 3, tandis que le patient 6 atteint 98.66 %, ce qui est beaucoup plus élevé. Cette différence souligne la nécessité d'adapter les modèles aux caractéristiques spécifiques des patients afin d'obtenir des résultats optimaux. En revanche, le modèle 1D-CNN présente une cohérence remarquable, avec une précision relativement stable d'un patient à l'autre. Par exemple, le 1D-CNN a atteint des précisions de 82,76 % pour le patient 2 et de 89,63 % pour le patient 1, démontrant des performances constantes indépendamment des variations des données individuelles.

Tableau 3. 11: Résultats en terme d'Accuracy de classification monosujet pour les états de vigilance

	<i>Architectures</i>				
	<i>MLP</i>	<i>LSTM</i>	<i>1D-CNN</i>	<i>1D-CNN-LSTM</i>	<i>AE-LSTM</i>
<i>Patient 1</i>	86.95	98.26	89.63	99.13	99.13
<i>Patient 2</i>	75.86	88.96	82.76	91.03	80.00
<i>Patient 3</i>	50.00	71.42	75.00	71.42	85.71
<i>Patient 4</i>	72.22	86.11	65.75	91.66	94.99
<i>Patient 5</i>	71.05	85.78	79.49	86.84	91.05
<i>Patient 6</i>	76.67	98.66	80.00	97.33	100

Cependant, ce sont les modèles hybrides qui se distinguent par leur remarquable efficacité (voir

Figure 3. 7). Par exemple, le 1D-CNN-LSTM atteint une précision de 99.13 % pour le patient 1, tandis que l'AE-LSTM atteint une précision absolue de 100 % pour le patient 6. Ces performances supérieures soulignent l'impact positif de la combinaison des caractéristiques spatiales des CNN avec la capacité de modélisation temporelle des LSTM. De plus, l'AE-LSTM bénéficie également de la représentation compressée fournie par l'encodeur de l'auto-encodeur, qui capture des informations importantes pour la prédiction, lesquelles sont ensuite traitées par le LSTM. En conclusion, ces résultats soulignent davantage l'importance des modèles hybrides innovants, mettant en évidence leur pertinence dans l'adaptation aux spécificités des données individuelles.

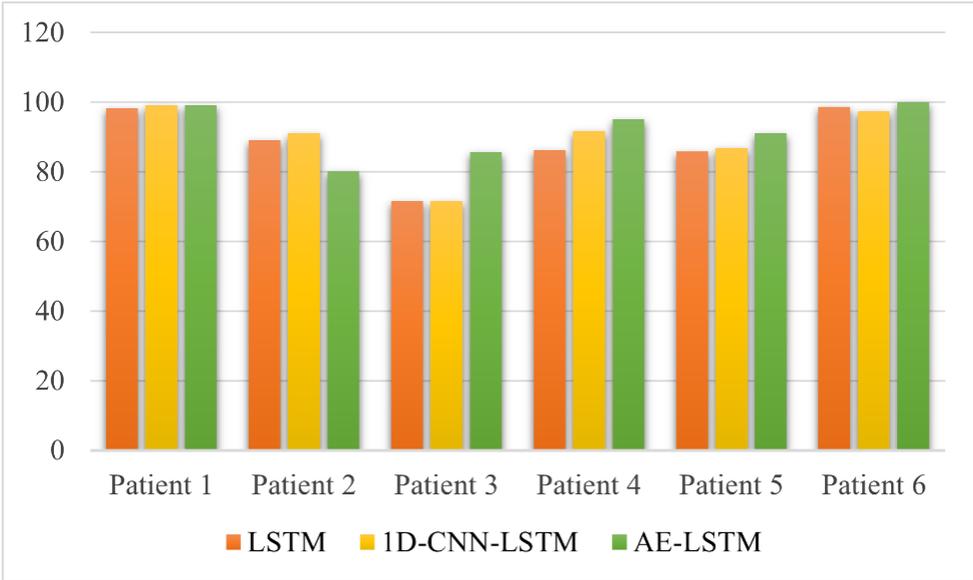


Figure 3. 7 : Analyse monosujet des performances en terme d'accuracy pour la classification des états de vigilance

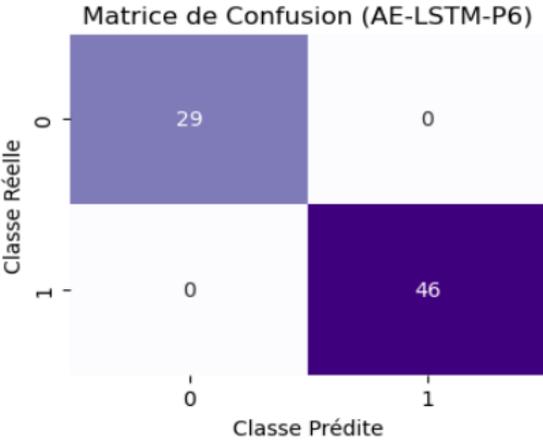


Figure 3. 8 : Matrice de confusion du modèle AE-LSTM-patient-6

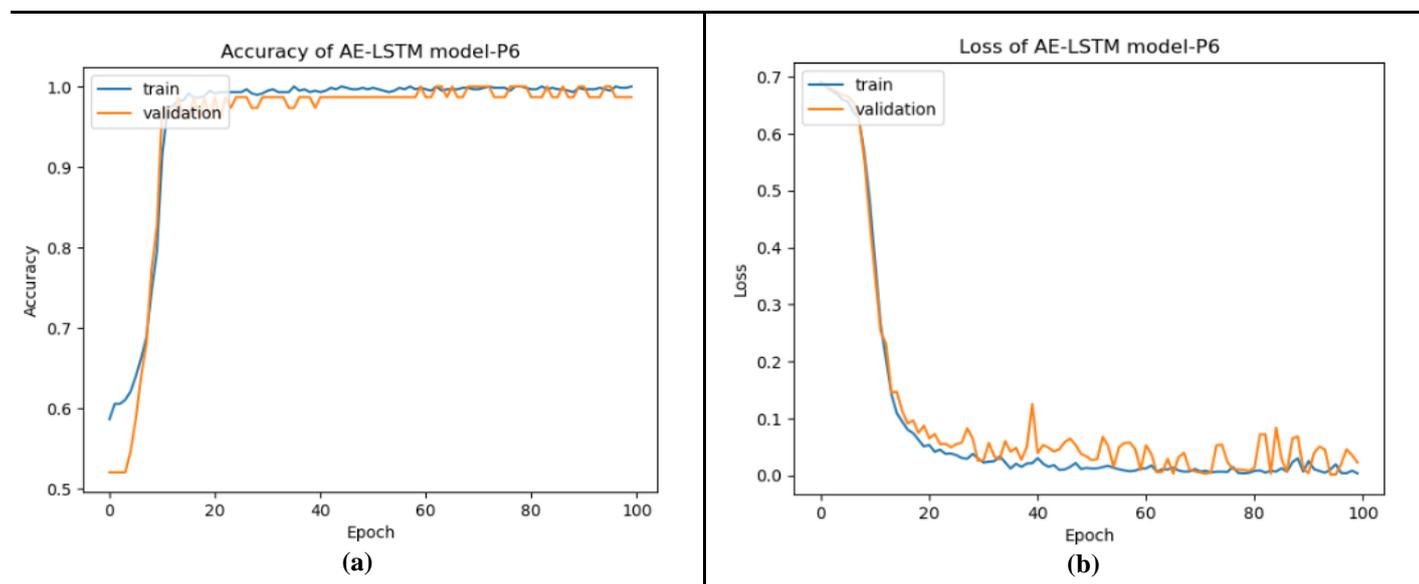


Figure 3.9 : Évolution de l'accuracy et de la Loss pour le modèle AE-LSTM sur le patient P6

B. Mode Leave One Patient Out « LOO »

Ce mode consiste à entraîner le modèle sur les données de tous les patients sauf un, puis à tester le modèle sur les données du patient exclu. Ce processus est répété pour chaque patient, ce qui permet d'évaluer la capacité du modèle à généraliser à de nouveaux patients. Les résultats de classification obtenus en utilisant différentes architectures sont présentés dans le tableau ci-dessous ainsi que les deux figures. Chaque ligne du tableau représente les performances de chaque architecture lorsqu'un seul patient est utilisé comme ensemble de test. On observe une variabilité significative des performances des différents modèles selon les patients.

L'architecture 1D-CNN-LSTM affiche la meilleure performance moyenne avec une accuracy de 68.29%, suivie par LSTM avec 64.68%, le 1D-CNN avec 61.76%, et enfin le MLP avec 58.70%.

Tableau 3.12 : Résultats en terme d'Accuracy de classification pour le mode Leave One Patient Out

Répartition des données		Architectures			
Données d'entraînement	Données de test	MLP	LSTM	1D-CNN	1D-CNN-LSTM
P2, P3, P4, P5, P6	P1	58.62	60.88	53.45	67.67
P1, P3, P4, P5, P6	P2	59.31	91.61	71.72	92.07
P1, P2, P4, P5, P6	P3	52.63	40.56	50.00	47.37
P1, P2, P3, P5, P6	P4	50.00	59.44	59.94	60.77
P1, P2, P3, P4, P6	P5	71.61	75.74	79.43	77.86
P1, P2, P3, P4, P5	P6	60.00	59.86	56.00	64.00
Accuracy Moyenne (%)		58.70	64.68	61.76	68.29

La combinaison CNN-LSTM semble tirer parti des forces de chaque méthode, capturant à la fois les caractéristiques importantes grâce au CNN et les dépendances temporelles grâce au LSTM. Cependant, une variabilité significative est observée entre les patients pour tous les modèles, avec des performances allant de 40.56% (LSTM pour P3) à 92.07% (1D-CNN-LSTM pour P2). Par exemple, le MLP montre une performance modeste avec une accuracy allant de 50.00% pour P4 à 71.61% pour P5, suggérant qu'il est sensible aux variations individuelles des données des patients. Le modèle LSTM, bien qu'il surpasse MLP avec une accuracy moyenne de 64.68%, affiche une grande variabilité entre les patients, atteignant seulement 40.56% pour P3 mais montant à 91.61% pour P2, indiquant qu'il capture mieux les dépendances temporelles présentes dans les données de certains patients mais pas dans d'autres.

Le 1D-CNN offre une performance intermédiaire avec une accuracy moyenne de 61.76%, variant de 50.00% pour P3 à 79.43% pour P5, ce qui montre qu'il peut extraire efficacement des caractéristiques spécifiques des données. Enfin, la combinaison CNN-LSTM, avec une accuracy moyenne de 68.29%, offre les meilleures performances globales, capturant plus efficacement les dépendances temporelles. Cependant, même ce modèle montre une variabilité, avec des résultats allant de 47.37% pour P3 à 92.07% pour P2, soulignant que les différences interindividuelles dans les données des patients sont importantes.

Pour résumer, la variabilité des performances des modèles d'un patient à l'autre peut être due à la nature intrinsèque des données du patient et à la sensibilité individuelle des modèles aux caractéristiques des données. Il est essentiel de prendre en compte cette variabilité lors de la sélection et de l'évaluation des modèles pour garantir des performances optimales dans des scénarios réels de prédiction médicale.

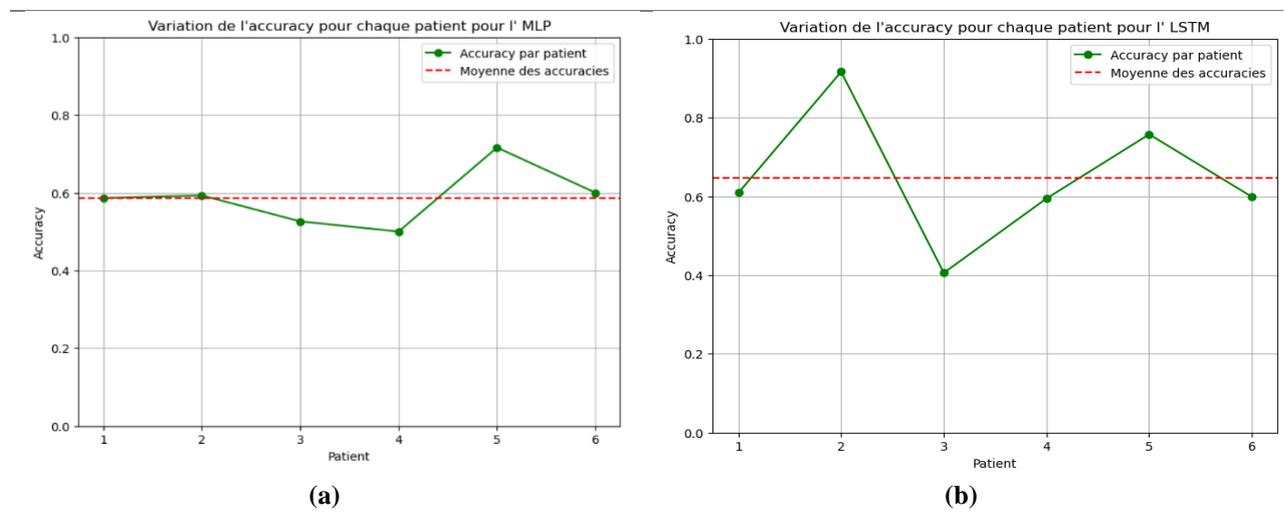


Figure 3.10 : Variation de l'accuracy pour chaque patient avec le mode LOO pour les différents architectures (a) Architecture MLP, (b) Architecture LSTM

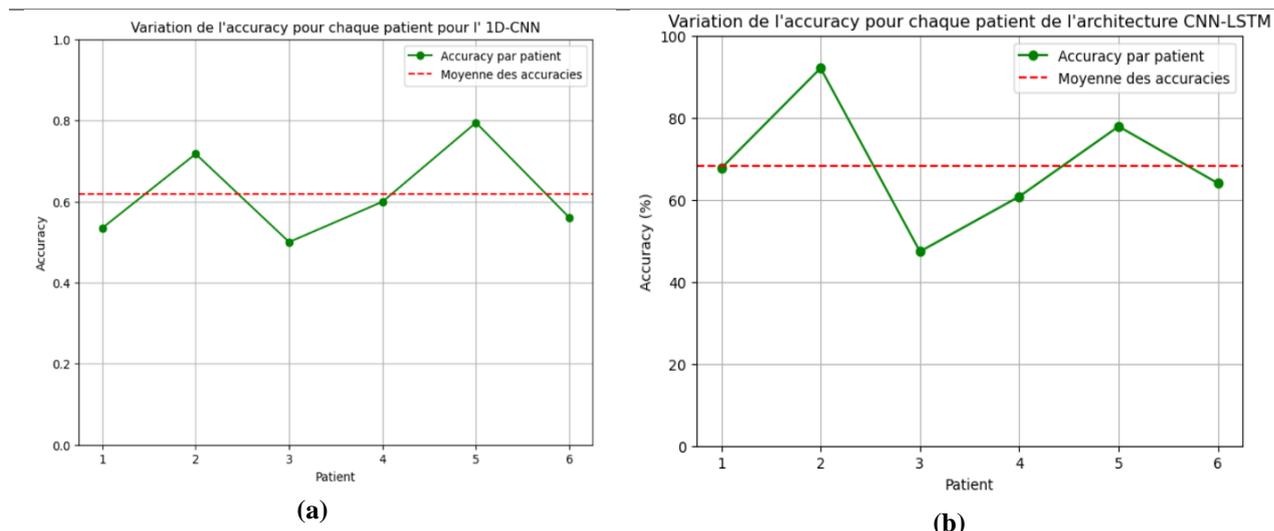


Figure 3.11 : Variation de l'accuracy pour chaque patient avec le mode LOO pour les différents architectures (a) Architecture 1D-CNN, (b) Architecture CNN- LSTM

C. Mode multi-sujets "Train/Val/Test"

Nous avons aussi testé un autre mode d'évaluation pour le mode multi-sujets, consistant à entraîner et tester les modèles sur l'ensemble des sujets. Pour la répartition des données, nous avons suivi une approche standard en utilisant 60% des données pour l'entraînement, 20% pour la validation et 20% pour la partie test. Les résultats présentés dans le Tableau ci-dessous reflètent les performances des différents modèles sur l'ensemble de la base de données, offrant ainsi une vue globale de leur efficacité.

Tableau 3.13 : Résultats obtenus pour la classification des états de vigilance avec différentes architectures selon le mode multi-sujets "Train/Val/Test"

	MLP	LSTM	1D-CNN	1D-CNN-LSTM	AE-LSTM_v1	AE_LSTM_v2
Accuracy	69.23	86.39	72.90	89.79	88.43	92.60
Précision	69.28	87.26	72.99	89.80	88.69	92.60
Recall	69.23	86.39	72.90	89.79	88.43	92.60
F1-score	69.14	86.35	72.91	89.79	88.43	92.60

Il est important de souligner que les modèles AE-LSTM_v1 et AE-LSTM_v2 se distinguent par la taille des séquences utilisées pour l'entraînement. AE-LSTM_v1 utilise des séquences de taille 5, tandis que l'AE-LSTM_v2 utilise des séquences de taille 10. Cette expérimentation avec des séquences plus longues est justifiée par la réduction du nombre de caractéristiques après l'encodage, qui permet de multiplier la longueur des séquences temporelles à moindre coût.

Nous remarquons que les modèles LSTM et ses variantes, ainsi que les modèles basés sur le 1D-CNN, affichent des niveaux d'accuracy compris entre 72.90% et 89.79%. En revanche, le MLP présente des accuracies légèrement inférieures.

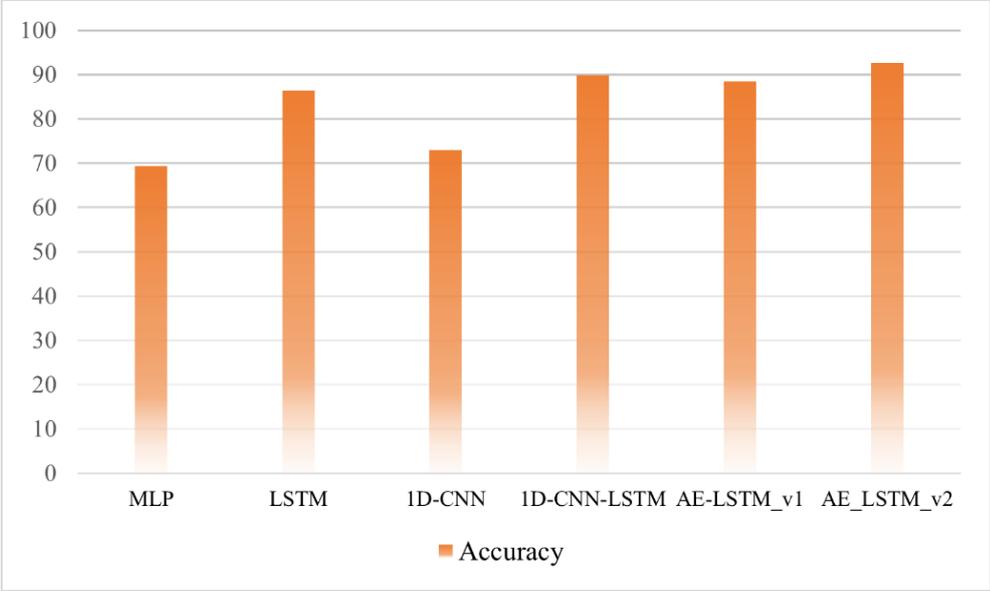


Figure 3.12: Analyse comparative des performances des différentes architectures étudié

La différence de longueur des séquences d'apprentissage entre AE-LSTM_v1 et AE-LSTM_v2 peut influencer significativement la capacité des modèles à saisir les motifs temporels dans les données et peut expliquer les différences d'accuracy observées (voir Figure 3.13).

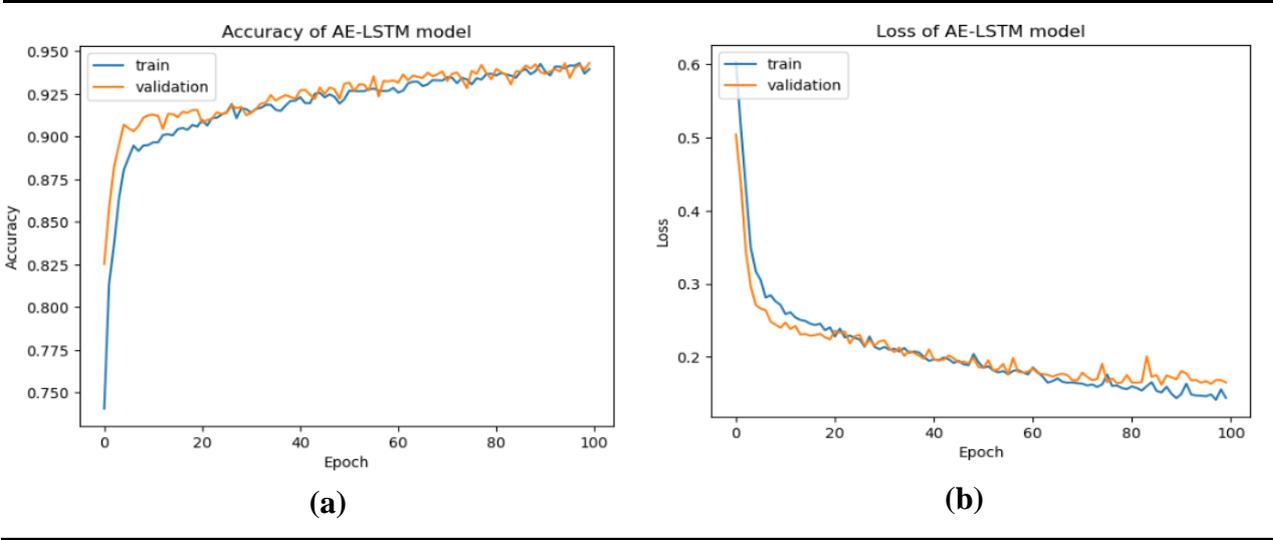


Figure 3.13: Évolution de l'Accuracy et de la Loss pour le Modèle AE-LSTM_v2

6.3.2 Résultats avec optimisation des hyperparamètres

Dans cette section, nous présenterons les différents résultats expérimentaux obtenus en utilisant différents algorithmes d'optimisation. Nous avons utilisé l'approche proposée GS-TPE ainsi que les techniques d'optimisation GS, RS et TPE et aussi le RS-TPE. Nos expériences comprennent l'utilisation de deux modes : tout d'abord, nous présenterons les résultats en utilisant le mode mono-sujet, puis nous examinerons le mode multi-sujets.

A. Mode Mono-sujets

Dans le cadre de l'optimisation des HPs pour le mode mono-sujet, nous avons choisi de tester différents algorithmes d'optimisation sur un modèle. Nous avons opté pour une architecture 1D-CNN en utilisant les données du patient 5, car ce patient dispose d'un grand nombre de données comparé aux autres.

Cependant, les résultats initiaux (sans processus d'optimisation) n'étaient pas très bons. Pour améliorer les performances de classification de ce modèle, nous avons décidé d'utiliser les algorithmes d'optimisation TPE et GS-TPE spécifiquement pour ce patient. Les données ont été réparties en trois ensembles : 80 % pour l'entraînement, 10 % pour la validation et 10 % pour le test. Le Tableau 3. 14 présente les performances en terme d'accuracy obtenus après l'optimisation des HPs de l'architecture 1D-CNN en utilisant les algorithmes TPE et GS-TPE. Les résultats montrent une amélioration significative des performances, indiquant que le réglage des HPs a permis d'optimiser le modèle.

Tableau 3. 14 : Résultats obtenus sur les données de test pour l'architecture 1D-CNN en utilisant l'approche GS-TPE

	<i>Accuracy (%)</i>		
	<i>Non optimisé</i>	<i>TPE</i>	<i>GS-TPE</i>
1D-CNN-P5	79.49	81.57	84.21

Pour le premier algorithme (TPE), la courbe de variation de l'accuracy en fonction du nombre d'itérations (Figure 3. 14) nous permet également de mieux comprendre l'évolution des performances du modèle 1D-CNN au cours de l'optimisation des HPs. On observe sur cette courbe une augmentation progressive de l'accuracy à chaque itération de TPE. Celle-ci atteint finalement son maximum, 87,18 %, à l'itération 95 sur les données de validation.

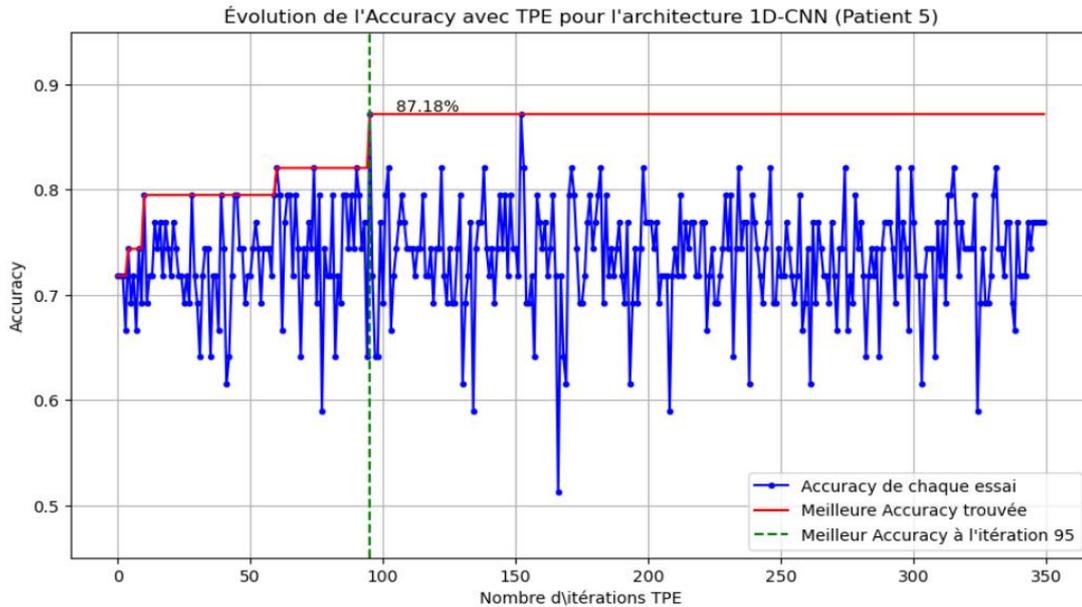


Figure 3. 14 : Évolution de l'accuracy du modèle 1D-CNN du patient-5 en fonction du nombre d'itérations TPE

Par la suite, l'objectif est d'améliorer le résultat en termes d'accuracy par rapport à TPE pour évaluer l'efficacité de l'approche GS-TPE dans l'obtention de performances supérieures. D'après le Tableau 3. 14 , nous remarquons une amélioration des résultats lors de l'utilisation de notre nouvelle approche hybride, GS-TPE. L'accuracy a dépassé celle obtenue avec le TPE seul et avec la version non optimisée, avec respectivement 81.57 % et 79.49 %. Dans un premier temps, nous avons utilisé GS pour identifier les cinq meilleures configurations, comme le montre le Tableau 3. 15. Ensuite, en appliquant des bornes de recherche au voisinage de chacune de ces configurations, TPE a exploré de nouveaux espaces pour améliorer les résultats. D'après le tableau, TPE est capable d'améliorer les résultats de la configuration 4 qui est la meilleure, avec une accuracy de 89.74 % sur les données de validation. Cependant, les autres configurations sont restées les mêmes.

En examinant la Figure 3. 15, qui représente l'évolution de l'accuracy des différentes configurations en fonction du nombre d'itérations du TPE, nous constatons que la meilleure configuration a atteint sa valeur maximale sur les données de validation à l'itération 125. Cette observation suggère que l'algorithme TPE a exploré efficacement l'espace des HPs dès les premières itérations.

Tableau 3. 15 : Performance des configurations GS-TPE en termes d'accuracy sur les données de validation du patient-5 (1D-CNN)

	<i>Accuracy/Données de validation (%)</i>	
	<i>GS</i>	<i>TPE</i>
<i>Configuration-1</i>	87.18	87.18
<i>Configuration-2</i>	87.18	87.18
<i>Configuration-3</i>	87.18	87.18
<i>Configuration-4</i>	87.18	89.74
<i>Configuration-5</i>	86.10	87.18

Évolution de l'accuracy du modèle 1D-CNN-P5 en fonction du nombre d'itérations TPE pour chaque configuration (GSTPE)

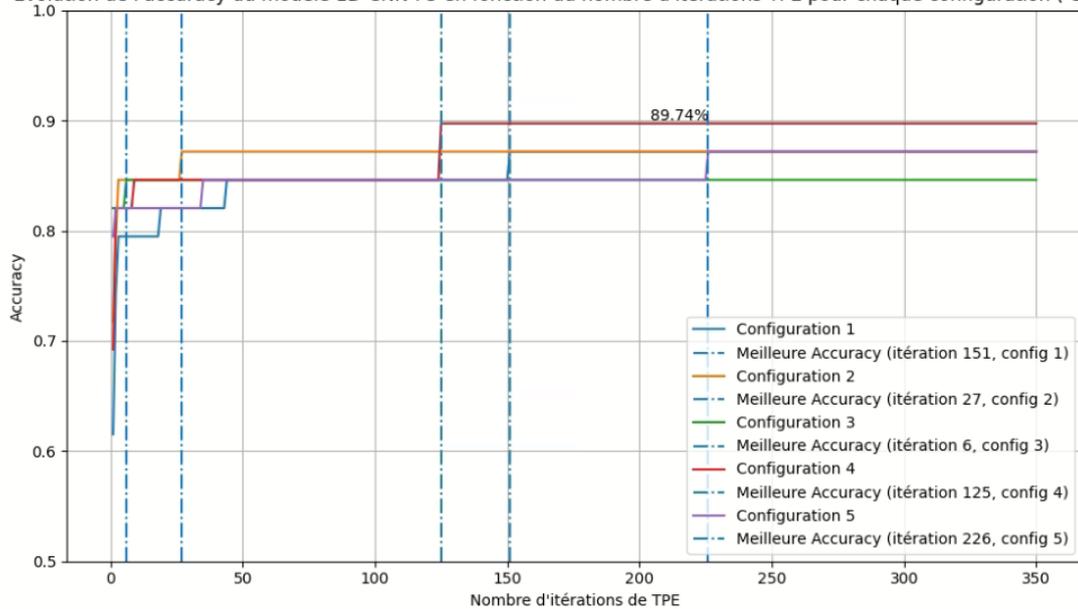


Figure 3. 15 : Performance en terme d'Accuracy de l'approche GSTPE pour l'architecture 1D-CNN selon le nombre d'itérations : Partie TPE

Synthèse relative aux meilleures configurations d'HPs avec GS-TPE pour l'architecture 1D-CNN

Le Tableau 3. 16 présente les nouvelles configurations optimales d'HPs identifiées par l'approche TPE. Ces configurations ont été retenues en raison de leur performance supérieure sur les données de validation, comparativement aux configurations initiales identifiées par GS. En observant les résultats, nous pouvons constater que TPE a effectué des ajustements significatifs par rapport à la configuration initiale identifiée par GS (*Config-4*). Par exemple, dans la configuration optimale (*Config-4'*) identifiée par TPE, le nombre de filtres dans les couches de convolution a été légèrement réduit, passant de 128 à 125 pour la première couche, de 256 à 244 pour la deuxième couche et de 512 à 532 pour la troisième couche. Ces ajustements ont permis de réduire la complexité du modèle tout en préservant sa capacité à extraire des caractéristiques pertinentes des données. De plus, le taux de dropout a été légèrement réduit à 0.47. Concernant le batch size, une

réduction de 64 à 54 a été observée, accélérant ainsi le processus d'entraînement du modèle tout en maintenant sa stabilité. Enfin, le taux d'apprentissage est resté inchangé à 0.0001, assurant une convergence rapide lors de l'entraînement du modèle. Ces ajustements apportés par TPE ont permis d'améliorer les performances du modèle, soulignant ainsi l'importance de l'optimisation des HPs dans la construction de modèles d'apprentissage automatique performants.

Tableau 3. 16 : Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'architecture 1D-CNN

<i>HPs/Configurations GS-TPE</i>	<i>Meilleure configuration identifiée par GS</i>				
	<i>Config-1</i>	<i>Config-2</i>	<i>Config-3</i>	<i>Config-4</i>	<i>Config-5</i>
Filtre_Conv-1D-1	64	64	256	128	64
Filtre_Conv-1D-2	128	128	256	256	64
Filtre_Conv-1D-3	64	512	128	512	256
Dropout_1	0.5	0.3	0.5	0.5	0.3
Learning rate	0.0001	0.0001	0.0001	0.0001	0.0001
Batch size	64	32	64	64	64
	<i>Nouvelles configurations optimales issues de TPE</i>				
	<i>Config-1'</i>	<i>Config-2'</i>	<i>Config-3'</i>	<i>Config-4'</i>	<i>Config-5'</i>
Filtre_Conv-1D-1	74	71	258	125	74
Filtre_Conv-1D-2	116	124	269	244	66
Filtre_Conv-1D-3	72	501	116	532	270
Dropout_1	0.6	0.33	0.64	0.47	0.27
Learning rate	0.0001	9.21 ^e -05	9.76 ^e -05	0.0001	9.35 ^e -05
Batch size	54	32	74	54	54

B. Mode Multi-sujets

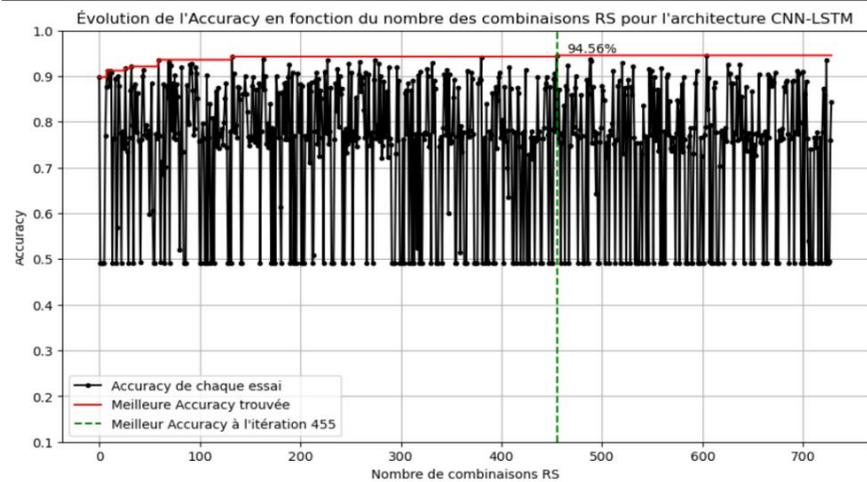
Dans le cadre de l'optimisation des HPs pour le mode multi-sujet, nous avons choisi de tester différents algorithmes d'optimisation, à savoir RS, GS, TPE, RS-TPE, ainsi que notre nouvelle approche GS-TPE. Nous avons utilisé la même répartition des données que lors de la classification sans processus d'optimisation, avec 60 % des données pour l'entraînement, 20 % pour la validation et 20 % pour le test.

Le tableau ci-dessous présente les résultats obtenus en termes d'accuracy qui a été évaluée sur les données de validation ainsi que sur les données de test. Pour évaluer l'efficacité de RS et GS, nous avons comparé les résultats obtenus avec ces deux algorithmes sur l'ensemble des modèles. Les résultats montrent que, pour la plupart des modèles, GS offre des performances légèrement meilleures que RS. Par exemple, pour le modèle MLP, GS a atteint une accuracy de validation de 75.92% et une accuracy de test de 72.24%, contre 74.92% et 71.23% respectivement pour RS.

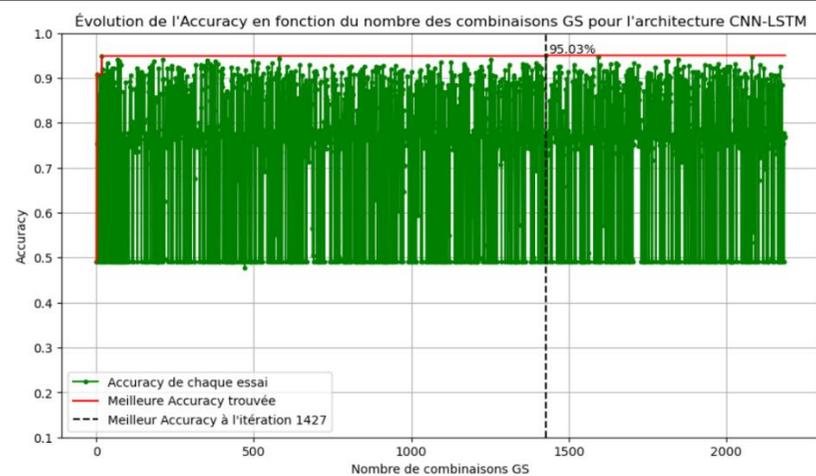
De même, pour le modèle LSTM (Figure 3. 17), GS a obtenu une accuracy de validation de 92.99% et une accuracy de test de 89.38%, comparativement à 91.16% et 87.55% pour RS. Ces résultats suggèrent que GS est généralement plus performant que RS, bien que les améliorations soient souvent modestes. Pour le 1D-CNN-LSTM, on constate des différences significatives entre les différentes méthodes d'optimisation. En utilisant RS, nous avons obtenu une accuracy de validation de 94.56 % et de test de 91.02 %. En comparaison, GS a amélioré ces résultats, obtenant une accuracy de validation de 95.03 % et de test de 92.51 %. Cela indique que GS est plus efficace que RS pour optimiser le modèle 1D-CNN-LSTM, étant donné qu'il explore tout l'espace.

Tableau 3. 17 : Résultats multi-sujets obtenus avec optimisation pour la classification des états de vigilance

	<i>Non Optimisé</i>	<i>RS</i>		<i>GS</i>		<i>TPE</i>		<i>RS-TPE</i>		<i>GS-TPE</i>	
	<i>Test_Acc</i>	<i>Validation_ACC</i>	<i>Test_Acc</i>	<i>Validation_ACC</i>	<i>Test_Acc</i>	<i>Validation_ACC</i>	<i>Test_Acc</i>	<i>Validation_ACC</i>	<i>Test_Acc</i>	<i>Validation_ACC</i>	<i>Test_Acc</i>
<i>MLP</i>	69.23	74.92	71.23	75.92	72.24	75.92	73.24	76.59	71.90	77.26	74.58
<i>ID-CNN</i>	72.24	75.25	73.24	73.58	72.90	74.92	73.57	74.92	72.90	75.25	74.25
<i>LSTM</i>	86.39	91.16	87.55	92.99	89.38	92.45	90.20	93.81	90.40	94.01	91.15
<i>ID-CNN-LSTM</i>	89.79	94.56	91.02	95.03	92.51	95.37	93.12	95.71	91.08	97.48	93.74
<i>AE-LSTM</i>	88.43	94.76	91.97	95.24	92.58	90.07	88.37	95.17	93.33	95.31	93.53

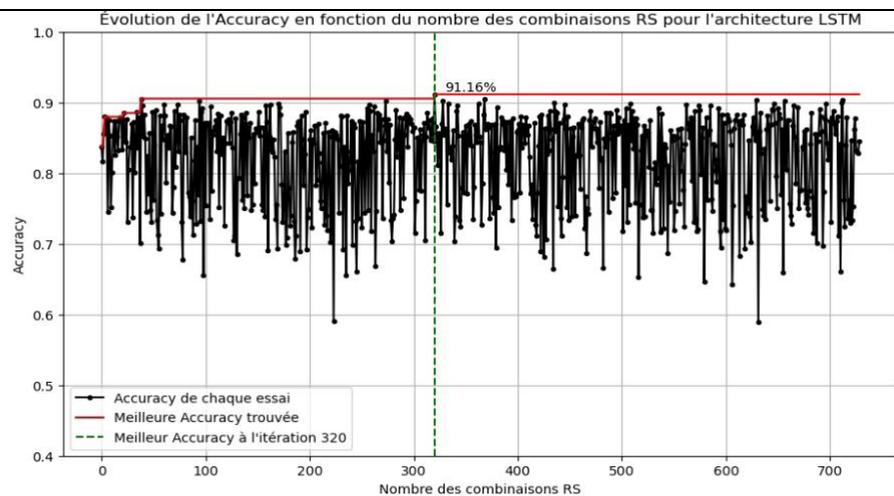


(a)

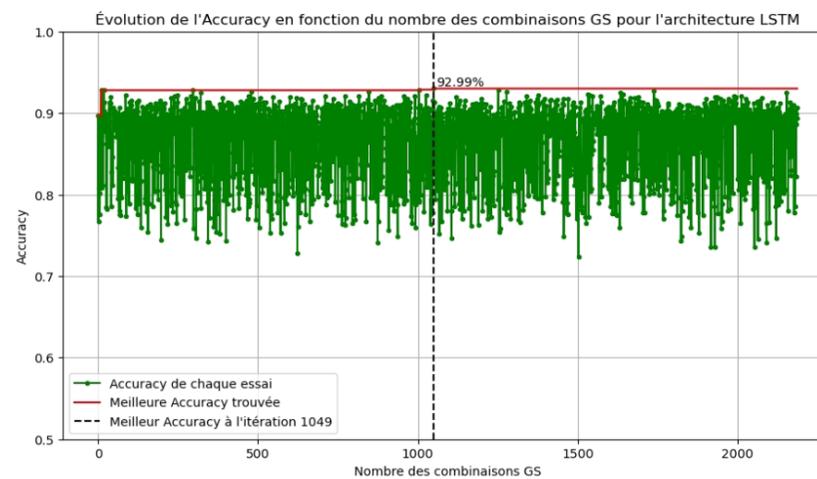


(b)

Figure 3. 16 : (a) Evolution de l'accuracy en fonction du nombre des combinaisons RS pour l'architecture CNN-LSTM (b) Evolution de l'accuracy en fonction du nombre des combinaisons GS pour l'architecture 1D-CNN-LSTM



(a)



(b)

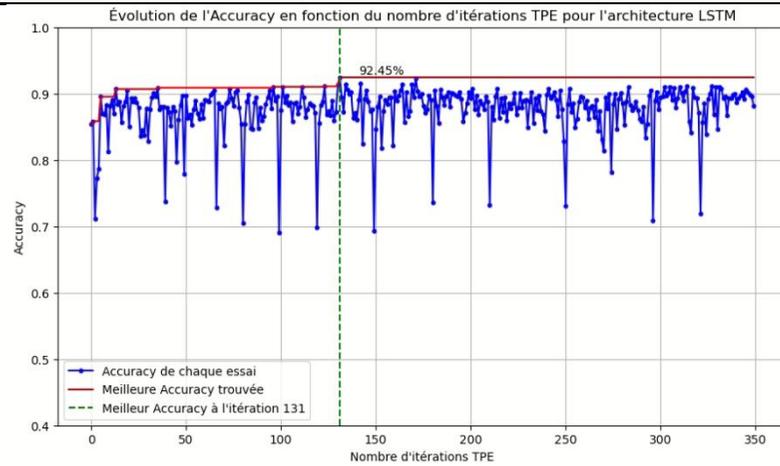
Figure 3. 17 : (a) Evolution de l'Accuracy en fonction du nombre des combinaisons RS pour l'architecture LSTM (b) Evolution de l'Accuracy en fonction du nombre des combinaisons GS pour l'architecture LSTM

D'après la Figure 3. 18 (b) l'utilisation du TPE a donné des résultats encore meilleurs. La meilleure performance (95.37% sur les données de validation) a été atteinte à l'itération 329 sur un total de 350 itérations. Ceci souligne le fait que l'optimisation des HPs du modèle 1D-CNN-LSTM est un processus itératif qui nécessite plusieurs essais pour converger vers des résultats optimaux. Ces résultats soulignent l'efficacité du TPE pour affiner les HPs et améliorer les performances du modèle.

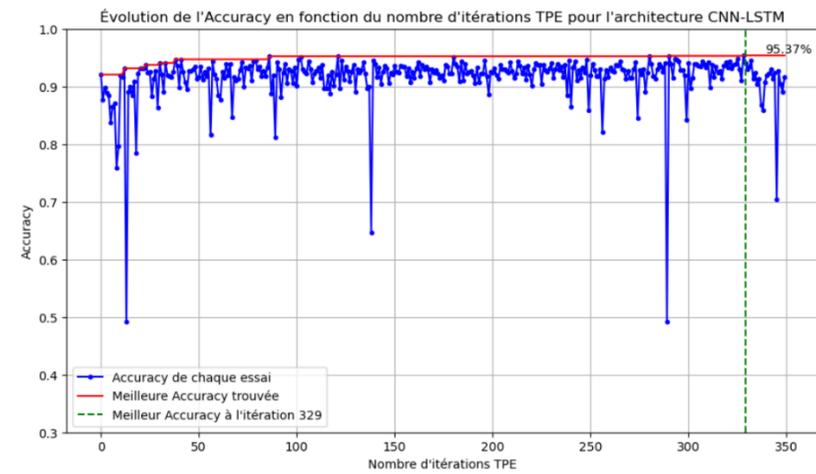
Pour les approches RS-TPE et GS-TPE, le tableau ci-dessus montre clairement l'impact positif de l'intégration de TPE sur les performances des modèles. En comparant RS avec RS-TPE, et GS avec GS-TPE, il apparaît que l'hybridation avec TPE améliore les résultats en affinant la recherche des HPs. Pour le modèle MLP, RS-TPE améliore la précision de test de 71.23% à 71.90%. Bien que l'amélioration soit légère, elle démontre que l'ajout de TPE apporte une valeur ajoutée. Pour le modèle LSTM, l'amélioration est plus marquée, la précision de test passant de 87.55% à 90.40%. En comparant GS à GS-TPE, on constate également des améliorations substantielles. Pour MLP, GS-TPE augmente la précision de test de 72.24% à 74.58%, et pour LSTM, de 89.38% à 91.15%.

- Analyse des résultats pour les deux approches GS-TPE et RS-TPE

Pour évaluer les performances des approches GS-TPE et RS-TPE, nous nous focalisons sur les architectures basées sur les LSTM. Le Tableau 3. 18 présente les performances en terme d'accuracy sur les données de validation pour les architectures LSTM, CNN-LSTM et AE-LSTM avec GS-TPE. Les résultats montrent que l'intégration de TPE améliore systématiquement les performances obtenues par GS. Pour chaque architecture, le tableau présente les performances en terme d'accuracy des cinq meilleures configurations après GS, suivies des nouvelles valeurs d'accuracy après l'affinement par TPE. Pour le modèle LSTM, la meilleure accuracy après l'utilisation de TPE est obtenue avec la configuration 4, passant de 92.78% à 94.01% (voir Figure 3. 20 (b)). De même, pour le modèle 1D-CNN-LSTM, la meilleure performance après raffinement est obtenue avec la configuration 3, soit 97,48 %. Et pour le modèle AE-LSTM, c'est la configuration 4 qui a enregistré la meilleure performance, passant de 94.76 % à 95.31 %.



(a)



(b)

Figure 3. 18 : Evolution de l'Accuracy en fonction du nombre d'itérations TPE pour l'architecture LSTM (b) Evolution de l'Accuracy en fonction du nombre d'itérations TPE pour l'architecture CNN-LSTM

Tableau 3. 18 : Performance des configurations GS-TPE en termes d'accuracy sur les données de validation des architectures LSTM, CNN-LSTM et AE-LSTM

	<i>LSTM</i>		<i>CNN-LSTM</i>		<i>AE-LSTM</i>	
	<i>Accuracy/Données de validation (%)</i>		<i>Accuracy/Données de validation (%)</i>		<i>Accuracy/Données de validation (%)</i>	
	<i>GS</i>	<i>TPE</i>	<i>GS</i>	<i>TPE</i>	<i>GS</i>	<i>TPE</i>
<i>Configuration-1</i>	92.99	93.74	95.03	96.73	95.24	95.24
<i>Configuration-2</i>	92.85	93.19	94.89	97.00	94.89	95.17
<i>Configuration-3</i>	92.78	93.67	94.76	97.48	94.89	95.10
<i>Configuration-4</i>	92.78	94.01	94.62	96.32	94.76	95.31
<i>Configuration-5</i>	92.78	93.67	94.28	96.80	94.76	95.17

En abordant l'approche RS-TPE, qui combine le RS et l'algorithme TPE, il convient de noter sa distinction majeure par rapport au GS-TPE. RS sélectionne aléatoirement les hyperparamètres pour chaque combinaison et les teste selon le nombre d'itérations que nous avons définies, contrairement à GS qui évalue systématiquement toutes les combinaisons de la grille.

D'après le Tableau 3. 19 et la Figure 3. 22, nous voyons que pour l'architecture LSTM, la configuration optimale choisie est la configuration 4, atteignant le meilleur résultat à l'itération 144. En revanche, la configuration optimale pour le modèle 1D-CNN-LSTM est la cinquième, et sa performance se stabilise à partir de l'itération 213, avec une accuracy de 95.71%. Quant au modèle MLP, la cinquième configuration est également conservée, mais sa performance plafonne à l'itération 7.

Dans l'ensemble, ces observations soulignent que chaque modèle dépend de sa propre architecture et de sa complexité d'apprentissage. Par exemple, pour l'architecture LSTM, une convergence relativement rapide à l'itération 144 suggère une adaptabilité efficace aux données d'apprentissage. En revanche, le modèle 1D-CNN-LSTM, avec ses performances stabilisées à partir de l'itération 213, témoigne de la nécessité de disposer de plus de temps pour explorer l'espace des HPs en raison de sa complexité.

Tableau 3. 19: Performance des configurations RS-TPE en termes d'accuracy sur les données de validation des architectures LSTM, CNN-LSTM et AE-LSTM

	<i>LSTM</i>		<i>CNN-LSTM</i>		<i>AE-LSTM</i>	
	<i>Accuracy/Données de validation (%)</i>		<i>Accuracy/Données de validation (%)</i>		<i>Accuracy/Données de validation (%)</i>	
	<i>RS</i>	<i>TPE</i>	<i>RS</i>	<i>TPE</i>	<i>RS</i>	<i>TPE</i>
<i>Configuration-1</i>	91.15	93.74	94.56	95.44	94.76	94.96
<i>Configuration-2</i>	90.54	93.60	94.42	95.23	94.69	95.03
<i>Configuration-3</i>	90.47	93.74	94.28	95.55	94.62	94.76
<i>Configuration-4</i>	90.40	93.81	94.01	94.48	94.62	95.17
<i>Configuration-5</i>	90.40	93.60	93.80	95.71	94.55	95.10

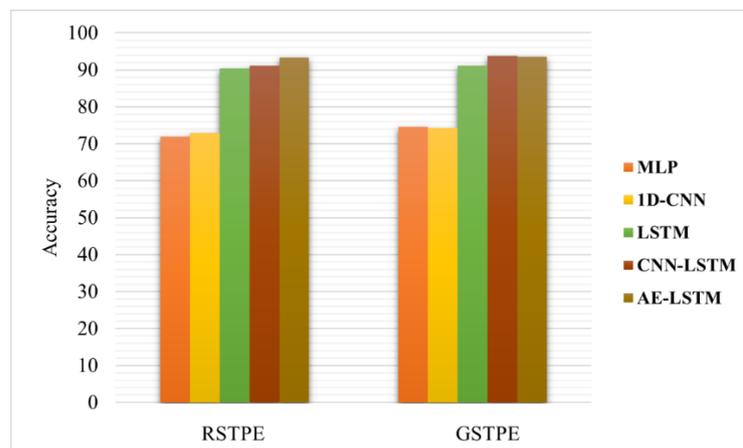
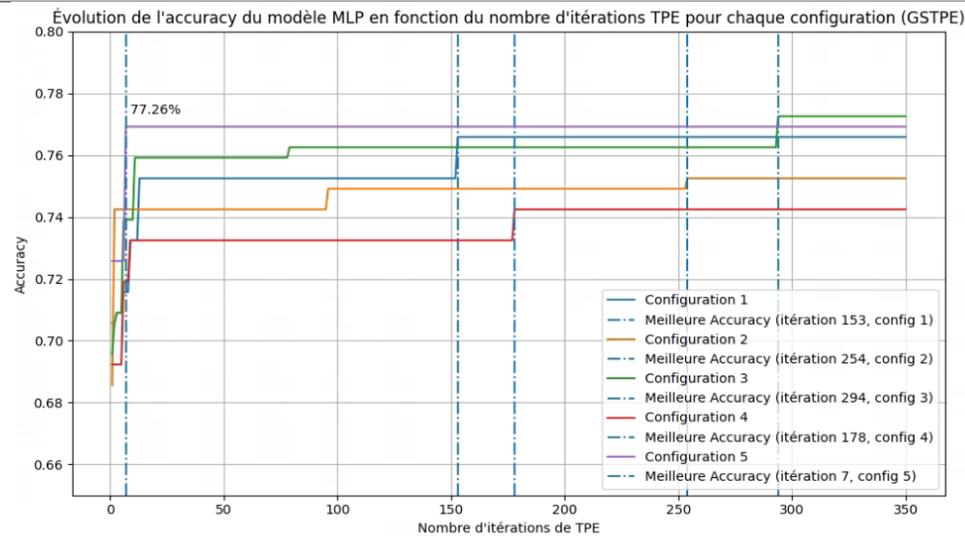
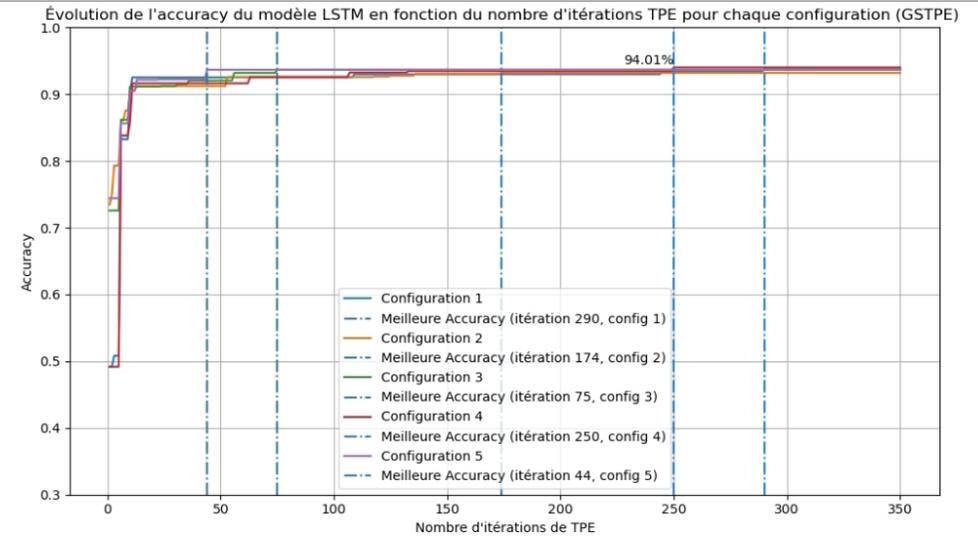


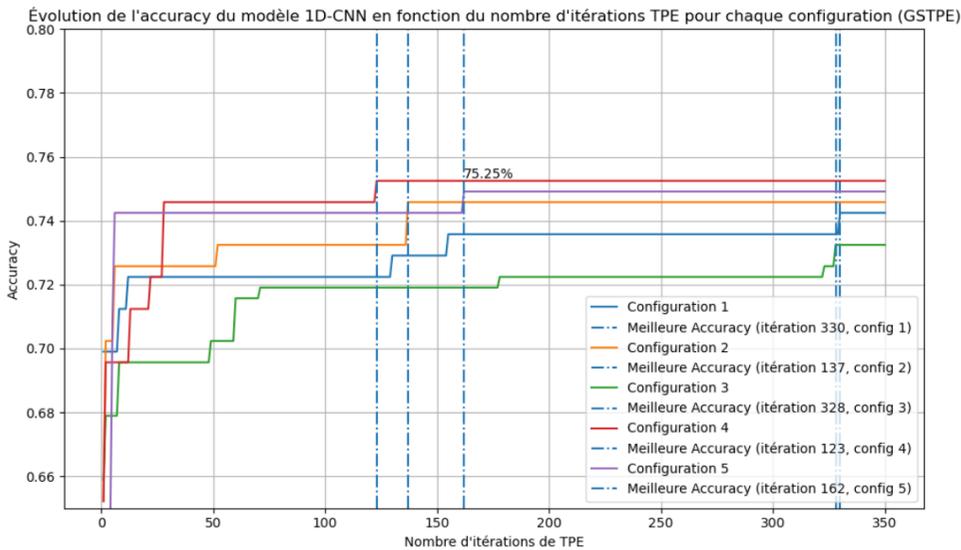
Figure 3. 19 : Comparaison des Performances de Modèles pour les deux différents approches GS-TPE et RS-TPE



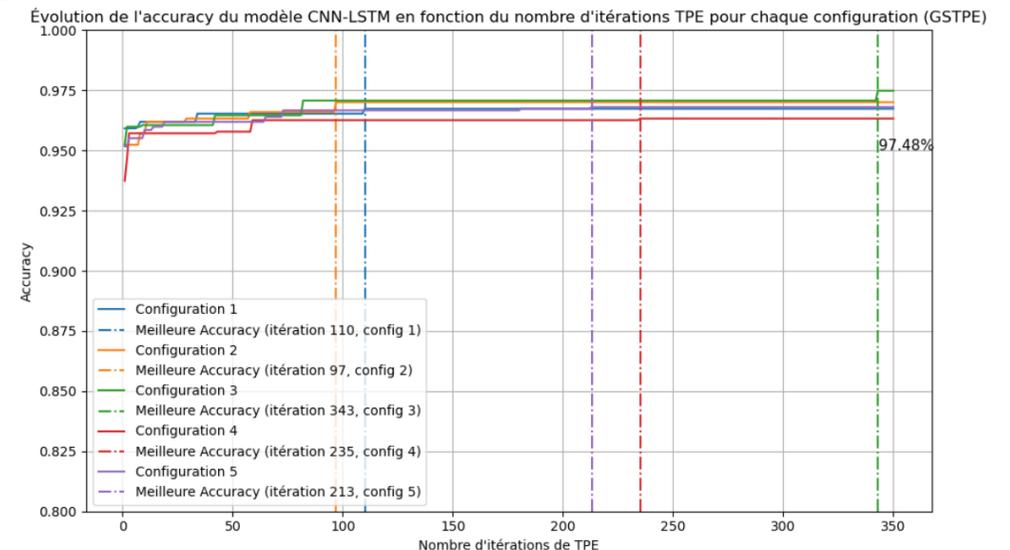
(a)



(b)



(c)



(d)

Figure 3. 20: : Analyse de l'évolution de la meilleure Accuracy des 5 configurations pour l'approche GSTPE en fonction du nombre d'itérations TPE pour (a) l'architecture MLP (b) l'architecture LSTM (c) l'architecture 1D-CNN (d) l'architecture CNN-LSTM

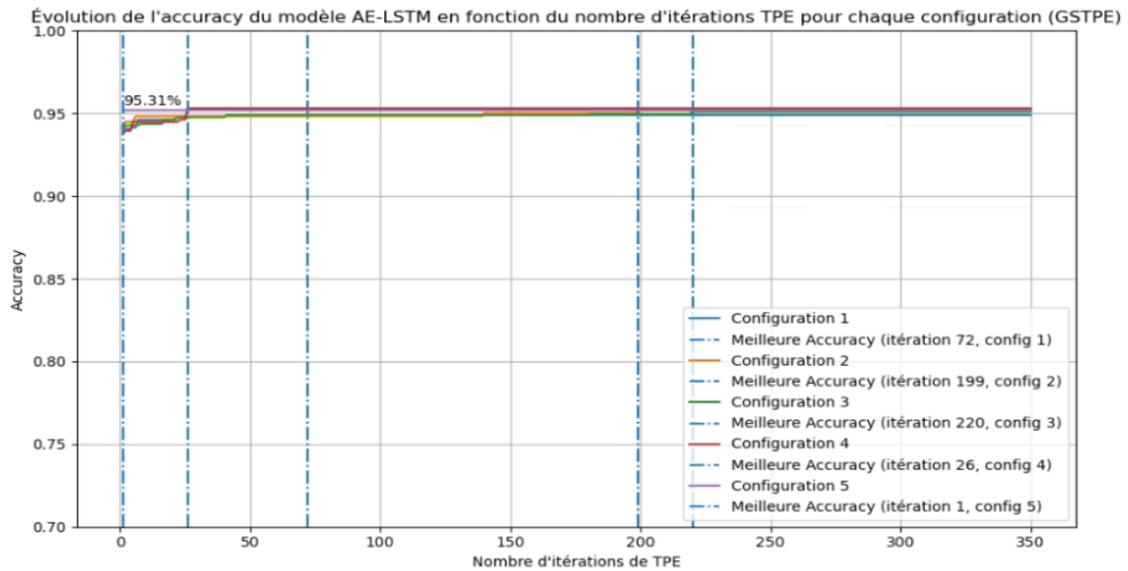
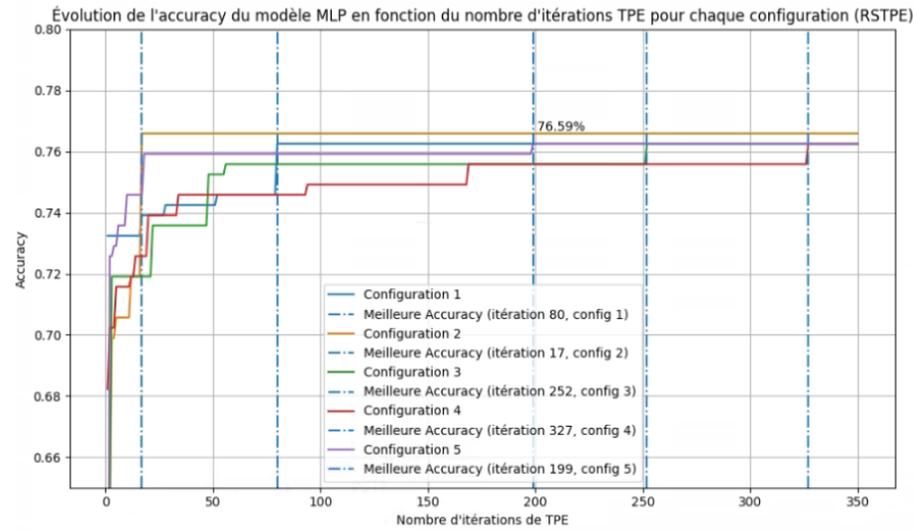
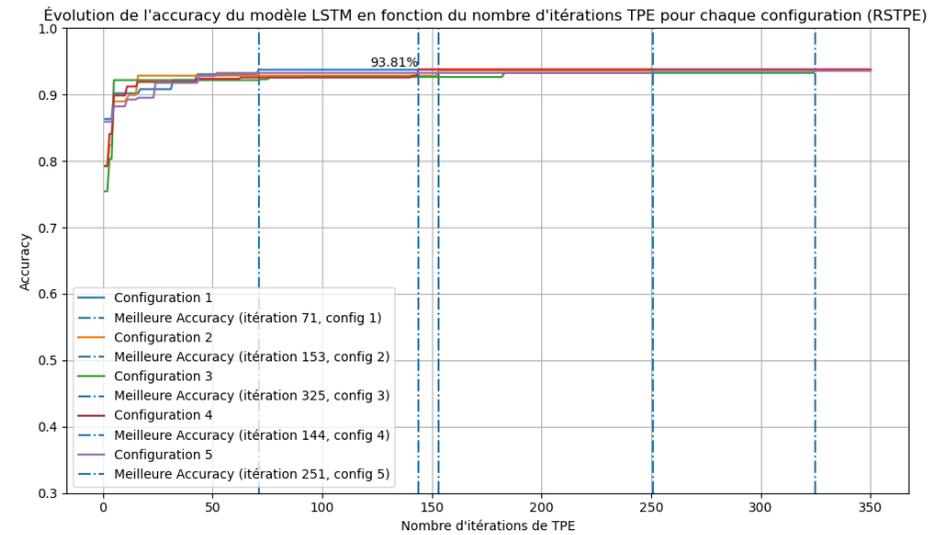


Figure 3. 21 : Analyse de l'évolution de la meilleure Accuracy des 5 configurations pour l'approche GSTPE de l'architecture AE-LSTM en fonction du nombre d'itérations TPE

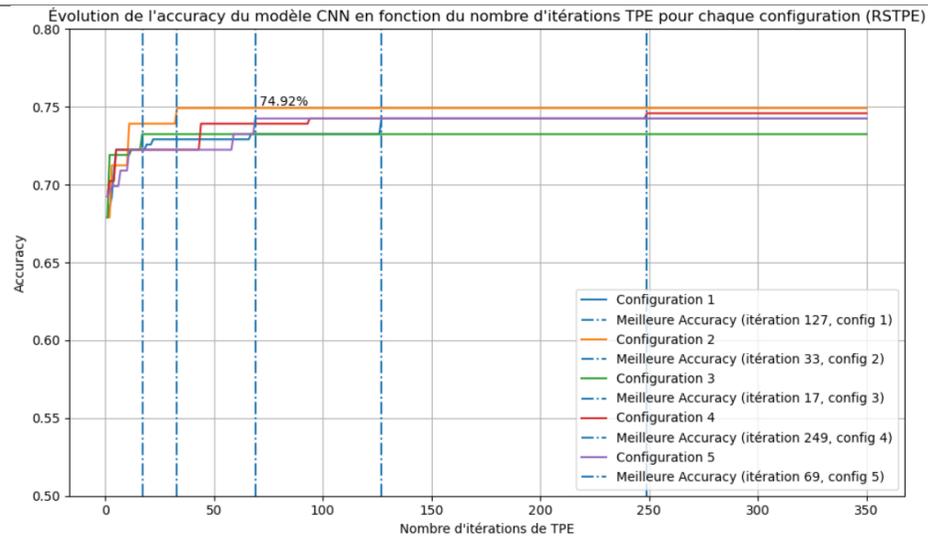
De plus, pour l'architecture AE-LSTM avec GS-TPE, les performances se stabilisent dès l'itération 26, avec une accuracy de 95.31 % (Figure 3. 21), ce qui indique une très bonne capacité d'apprentissage et d'adaptation rapide aux données. En revanche, pour RS-TPE, la performance maximale est atteinte à l'itération 178 dans la configuration 4 (Figure 3. 23), ce qui suggère que RS-TPE a besoin d'un plus grand nombre d'itérations pour atteindre son plein potentiel. Donc l'analyse révèle que l'approche GS-TPE surpasse de manière significative l'approche RS-TPE en termes de performances sur les données de validation et de test pour tous les modèles évalués, y compris MLP, LSTM, 1D-CNN, 1D-CNN-LSTM et AE-LSTM (Figure 3. 19). Les matrices de confusion montrent que les configurations optimisées par GS-TPE offrent de meilleures performances que celles optimisées par RS-TPE. Pour l'architecture LSTM (Figure 3. 24), GS-TPE réduit les erreurs de classification par rapport à RS-TPE (Figure 3. 25). Pour l'architecture 1D-CNN-LSTM (Figure 3. 26), l'avantage de GS-TPE est encore plus prononcé, avec la combinaison 1D-CNN-LSTM et GS-TPE affichant les meilleures performances globales et les erreurs de classification les plus faibles. Cela suggère que l'approche systématique de GS-TPE pour tester toutes les combinaisons de paramètres permet une meilleure exploration et une optimisation plus fine, tandis que l'approche aléatoire de RS-TPE peut laisser des combinaisons de paramètres non explorées, conduisant à des performances légèrement inférieures.



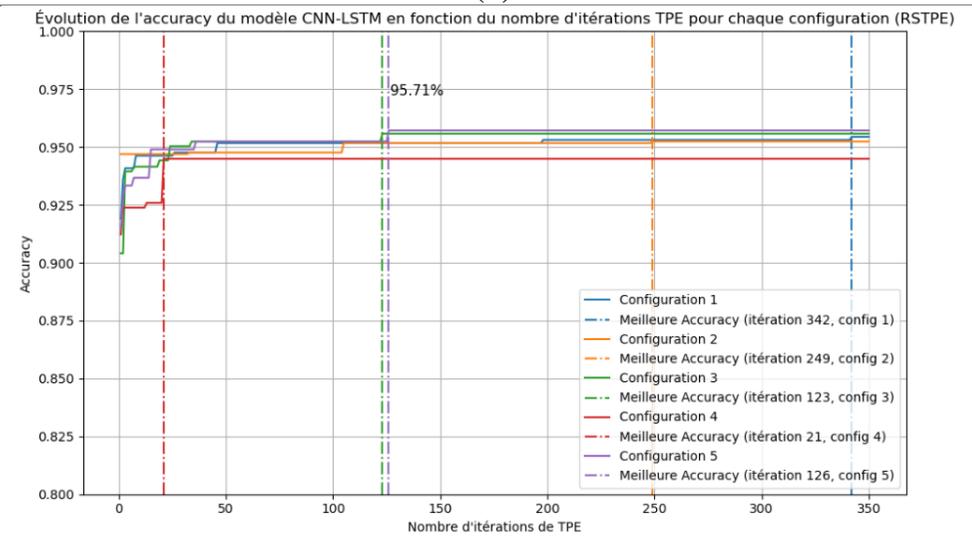
(a)



(b)



(c)



(d)

Figure 3. 22 : Analyse de l'évolution de la meilleure Accuracy des 5 configurations pour l'approche RSTPE en fonction du nombre d'itérations TPE pour (a) l'architecture MLP (b) l'architecture LSTM (c) l'architecture 1D-CNN (d) l'architecture CNN-LSTM

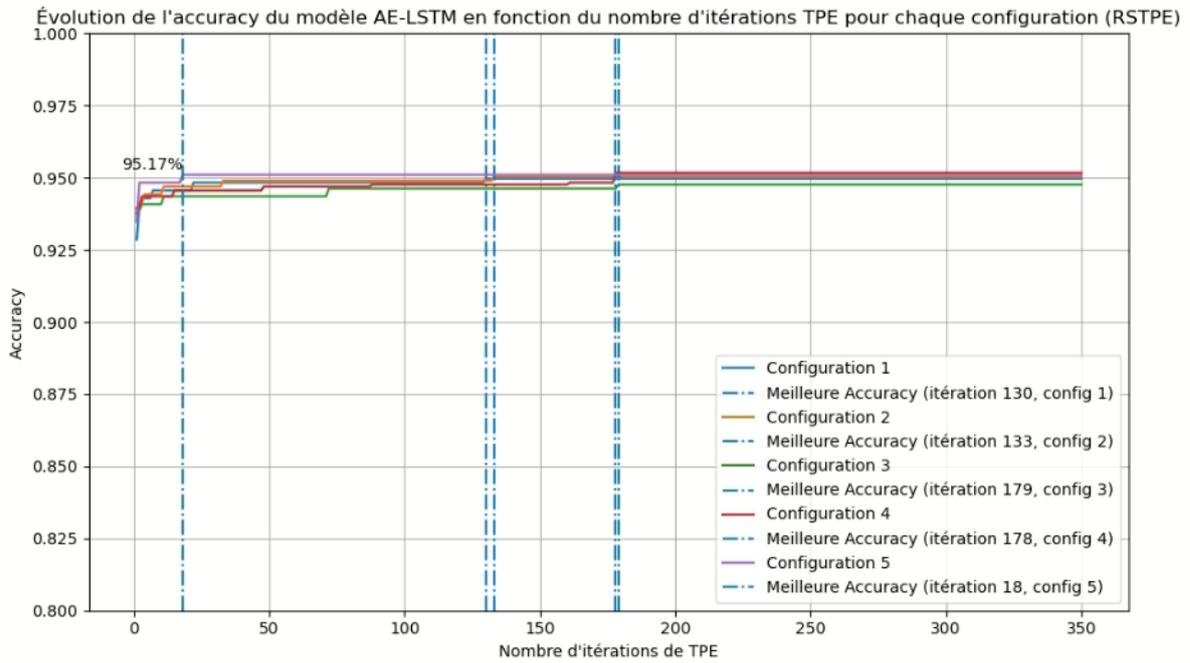


Figure 3. 23: Analyse de l'évolution de la meilleure Accuracy des 5 configurations pour l'approche RSTPE de l'architecture AE-LSTM en fonction du nombre d'itérations TPE

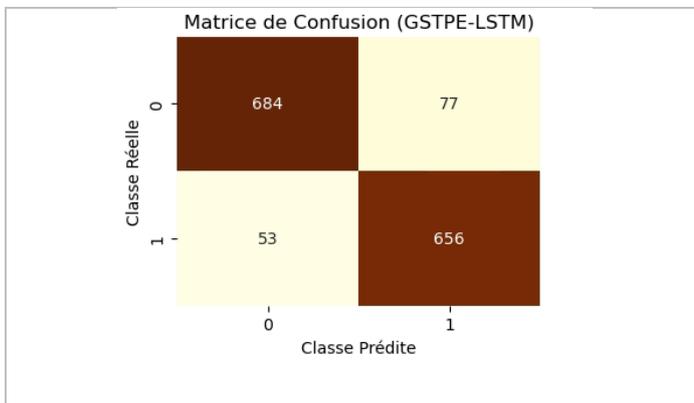


Figure 3. 24 : Matrice de confusion de l'architecture LSTM avec GS-TPE

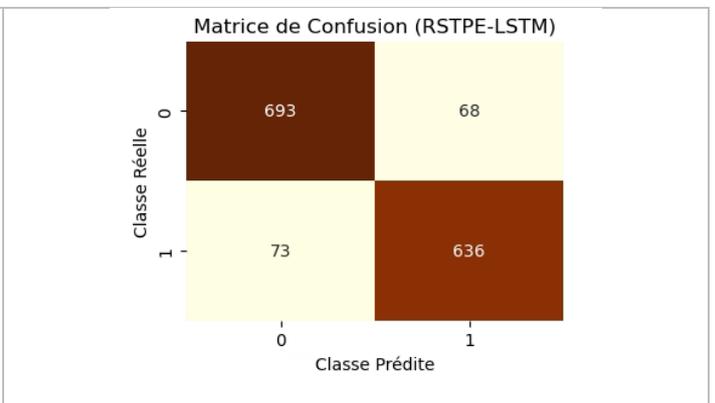


Figure 3. 25 : Matrice de confusion de l'architecture LSTM avec RS-TPE

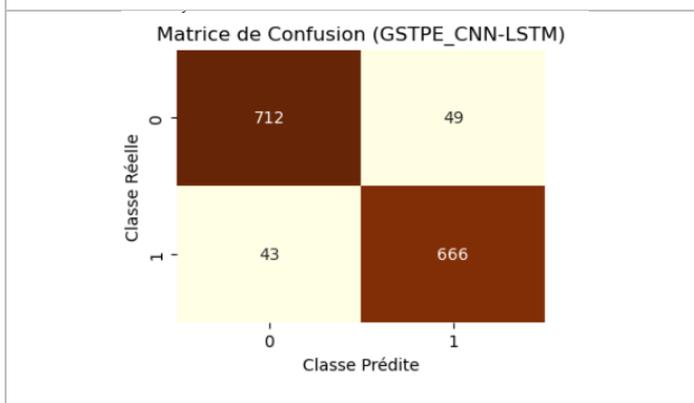


Figure 3. 26 : Matrice de confusion de l'architecture CNN-LSTM avec GS-TPE

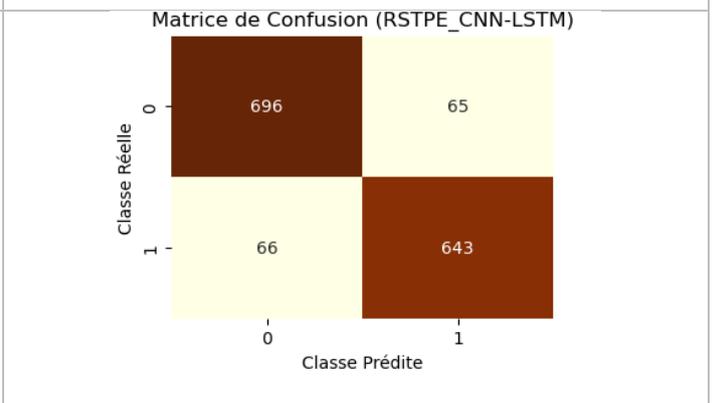


Figure 3. 27 : Matrice de confusion de l'architecture CNN-LSTM avec RS-TPE

Synthèse et détails relatifs aux meilleures configurations d'HPs pour les différents modèles expérimentés

Les tableaux ci-dessous présentent les meilleures configurations d'HPs identifiées par l'approche GS-TPE pour l'architectures LSTM, 1D-CNN-LSTM et AE-LSTM. Ils sont divisés en deux sections : la première montre les configurations initiales trouvées par GS, et la seconde montre les configurations optimales après affinement avec TPE.

Dans le Tableau 3. 20 décrivant l'architecture LSTM, les configurations varient en unités LSTM (8 à 64), en dropout (autour de 0.5) et en batch size (32 à 128). Après l'affinement, les unités LSTM sont ajustées pour équilibrer la capacité d'apprentissage, dans la config 1', le nombre d'unités LSTM-1 a été augmenté à 67 et les autres couches ont été ajustées pour mieux capturer les caractéristiques. Le dropout reste stable, assurant une régularisation efficace, tandis que les tailles de batch sont réajustées, avec des valeurs allant jusqu'à 24 pour des mises à jour de poids plus fréquentes. Particulièrement, la Config-4' a montré les meilleures performances en termes d'accuracy, ce qui justifie son choix final. Elle comporte une légère réduction des unités LSTM-1 à 62, des unités LSTM-2 à 58, et des unités LSTM-3 à 18, ainsi qu'un ajustement du taux d'apprentissage de 0,001 à 0,003. Ces modifications ont permis d'améliorer la précision du modèle tout en assurant une convergence plus rapide.

Tableau 3. 20 : Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'architecture LSTM

<i>HPs/Configurations GS-TPE</i>	<i>Meilleure configuration identifiée par GS</i>				
	<i>Config-1</i>	<i>Config-2</i>	<i>Config-3</i>	<i>Config-4</i>	<i>Config-5</i>
Unités_LSTM-1	64	32	64	64	64
Unités_LSTM-2	64	64	64	64	32
Unités_LSTM-3	64	32	64	32	16
Unités_LSTM-4	8	16	32	16	16
Dropout	0.5	0.5	0.3	0.5	0.1
Batch size	128	128	128	32	32
Learning_rate	0.01	0.01	0.01	0.001	0.001
	<i>Nouvelles configurations optimales issues de TPE</i>				
	<i>Config-1'</i>	<i>Config-2'</i>	<i>Config-3'</i>	<i>Config-4'</i>	<i>Config-5'</i>
Unités_LSTM-1	67	37	82	62	71
Unités_LSTM-2	52	58	70	58	24
Unités_LSTM-3	82	43	62	18	11
Unités_LSTM-4	14	7	32	18	17
Dropout	0.49	0.50	0.29	0.49	0.10
Batch size	24	128	112	32	48
Learning_rate	0.002	0.0038	0.004	0.003	0.004

Pour l'architecture 1D-CNN-LSTM, après avoir utilisé notre méthode GS-TPE pour explorer les HPs, nous avons constaté que parmi les configurations testées, la cinquième configuration s'est révélée la plus performante (Figure 3. 20). Cette configuration utilisait 128 filtres de convolution pour chaque couche, avec respectivement 10, 50 et 15 unités LSTM, et un taux d'apprentissage de 0,001. Après une optimisation par TPE, la configuration a été affinée pour ajuster ces valeurs à 131, 137 et 124 filtres de convolution, et à 9, 48 et 14 unités LSTM, avec un taux d'apprentissage réduit à 0,0009. Pour l'architecture AE-LSTM (Figure 3. 21), les cinq meilleures configurations GS initiales ont été optimisées et améliorées. Parmi eux, la configuration initiale 4 (*config 4*), identifiée par l'approche GS, a été optimisée en une nouvelle configuration (*config 4'*) à l'aide de l'algorithme TPE, donnant ainsi les meilleurs résultats. Les améliorations résultant du TPE consistent à réévaluer et à ajuster les unités dans les différentes couches LSTM, et à réduire la taille du lot. Ces changements permettent d'améliorer les performances du modèle.

Tableau 3. 21: Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'architecture CNN-LSTM

<i>HPs/Configurations GS-TPE</i>	<i>Meilleure configuration identifiée par GS</i>				
	<i>Config-1</i>	<i>Config-2</i>	<i>Config-3</i>	<i>Config-4</i>	<i>Config-5</i>
Filtre_Convolution_1D-1	128	128	128	128	128
Filtre_Convolution_1D-2	64	64	32	32	128
Filtre_Convolution_1D-3	128	128	128	64	128
Unités_LSTM_1	30	50	30	50	10
Unités_LSTM_2	50	50	10	30	50
Unités_LSTM_3	25	25	15	25	15
Learning rate	0.001	0.001	0.001	0.001	0.001
	<i>Nouvelles configurations optimales issues de TPE</i>				
	<i>Config-1'</i>	<i>Config-2'</i>	<i>Config-3'</i>	<i>Config-4'</i>	<i>Config-5'</i>
Filtre_Convolution_1D-1	131	127	131	119	131
Filtre_Convolution_1D-2	52	63	42	27	137
Filtre_Convolution_1D-3	116	129	130	70	124
Unités_LSTM_1	34	54	28	55	9
Unités_LSTM_2	52	45	10	26	48
Unités_LSTM_3	24	25	14	26	14
Learning rate	0.001	0.001	0.0009	0.0009	0.0009

Tableau 3. 22: Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'architecture AE-LSTM

<i>HPs/Configurations GS-TPE</i>	<i>Meilleure configuration identifiée par GS</i>				
	<i>Config-1</i>	<i>Config-2</i>	<i>Config-3</i>	<i>Config-4</i>	<i>Config-5</i>
Unités_LSTM_1	30	40	40	30	30
Unités_LSTM_2	50	30	40	30	30
Unités_LSTM_3	50	50	50	50	50

Unités_LSTM_4	15	15	5	15	25
Batch size	32	16	32	16	32
Learning rate	0.01	0.01	0.01	0.01	0.01
	<i>Nouvelles configurations optimales issues de TPE</i>				
	<i>Config-1'</i>	<i>Config-2'</i>	<i>Config-3'</i>	<i>Config-4'</i>	<i>Config-5'</i>
Unités_LSTM_1	36	38	35	35	29
Unités_LSTM_2	51	31	44	30	31
Unités_LSTM_3	61	48	50	46	54
Unités_LSTM_4	16	13	8	16	27
Batch size	42	16	42	16	22
Learning_rate	0.01	0.009	0.009	0.009	0.01

7 Conclusion

Dans ce chapitre, une approche d'optimisation hiérarchique, consistant à sélectionner un nombre restreint de “bonnes” configurations fournies par une approche systématique (Grid Search ou Random Search), puis à appliquer une recherche plus fine dans les voisinages de ces configurations candidates par la méthode du Tree Parzen Estimator, a été appliquée avec succès dans le but d'améliorer la performance des réseaux DNN pour la classification des états de vigilance. Cette méthode a démontré un potentiel considérable pour améliorer les performances des modèles. L'optimisation hiérarchique permet de focaliser l'exploration des hyperparamètres sur les zones les plus prometteuses de l'espace de recherche, rendant le processus d'optimisation plus efficace. En outre, cette approche ouvre des perspectives prometteuses pour la poursuite des travaux de recherche, notamment dans l'analyse des gestes et des postures pour la classification de différentes pathologies, offrant ainsi de nouvelles avenues pour le développement de solutions innovantes et efficaces dans divers domaines cliniques et technologiques.

Dans la suite, nous nous proposons de valider nos concepts dans le domaine de l'analyse du geste et des postures pour la classification de différentes pathologies.

Chapitre 4

Application des DNN pour la classification des pathologies affectant le mouvement humain

1 Introduction

Dans la continuité du chapitre précédent, nous nous proposons de valider notre approche d'optimisation GS-TPE destinée à améliorer la performance des DNN appliqués au domaine de l'analyse du geste et des postures pour la classification de différentes pathologies. Les entrées dans ce cas seront des vidéos d'analyse de la marche. Comme pathologie cible, nous avons opté pour l'arthrose de genou. Nous nous sommes aussi intéressés à la maladie de Parkinson pour vérifier l'aptitude de nos architectures optimisées à différencier les deux pathologies.

Les préoccupations liées à la santé et au bien-être sont de plus en plus présentes dans notre société, sur les plans professionnels et personnels. L'arthrose et la maladie de Parkinson font partie des problèmes de santé qui touchent de plus en plus d'individus et suscitent donc une attention croissante. Bien que ces pathologies soient différentes par nature et mécanisme, elles ont un point commun : leurs effets significatifs par rapport aux conditions réelles des patients. En effet, l'arthrose, maladie inflammatoire et dégénérative des articulations, comme la maladie de Parkinson, une affection neurodégénérative, perturbent les mouvements et les fonctions corporelles, entraînant ainsi d'importantes difficultés pour la marche. Actuellement, la prise en charge de ces maladies passe par un diagnostic basé sur l'examen clinique et l'imagerie, et porte principalement sur la gestion des symptômes et des douleurs. Il existe cependant un intérêt croissant pour la prévention, afin d'éviter les blessures et d'améliorer la qualité de vie des personnes concernées, avec une focalisation sur l'analyse des mouvements. Dans cette perspective, ce chapitre examinera les approches récentes visant à améliorer la prise en charge et à prévenir l'arthrose ainsi que la maladie de Parkinson, spécifiquement celles qui se focalisent sur l'analyse de la marche. Nous étudierons les progrès récents dans la recherche et la technologie ainsi que les interventions potentielles qui pourraient contribuer à atténuer l'impact de ces maladies sur ceux qui en souffrent. Dans ce cadre, nous

proposons une démarche originale basée sur les DNN pour l'analyse de la marche sans utilisation de marqueurs.

2 Pathologies du mouvement et des gestes humains

Un domaine essentiel de la recherche médicale est l'étude des pathologies qui affectent les mouvements et les gestes humains. Ces affections peuvent être très variées, de la maladie de Parkinson à l'arthrose. Les médecins et les chercheurs se concentrent de plus en plus sur ces maladies pour mieux analyser le comportement et l'évolution de la pathologie, mettre en œuvre des méthodes de diagnostic plus précises afin de créer des traitements efficaces. La présente section est consacrée à une description succincte de chacune des maladies traitées dans ce chapitre.

2.1 Arthrose

L'arthrose se caractérise par une dégénérescence progressive du tissu cartilagineux articulaire, qui atteint tous les autres composants de l'articulation, y compris ceux situés au niveau de la membrane colloïdale, de l'os sous-chondral, de la capsule et des ligaments. Selon l'Organisation Mondiale de la Santé (OMS), l'arthrose se définit de la manière suivante : "La maladie ostéoarthritique est le résultat à la fois de mécanismes mécaniques ainsi que de phénomènes biologiques qui déstabilisent l'équilibre entre la synthèse et la dégradation du cartilage et de l'os sous-chondral" (Figure 4. 1).

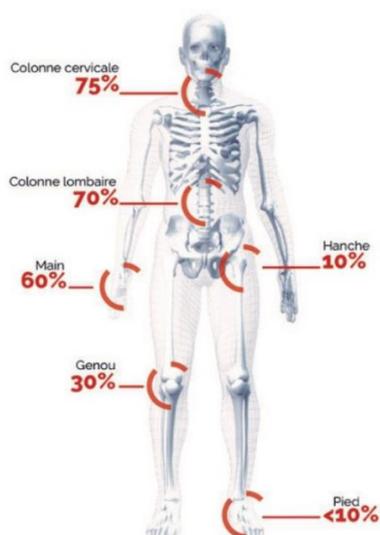


Figure 4. 1 : Les principales articulations touchées par l'arthrose diagnostiquée en radiologie [113]

Les causes de ce déséquilibre sont multiples : génétiques, congénitales, métaboliques ou traumatiques. L'arthrose affecte la totalité des organes de l'articulation diarthritique et entraîne des

modifications morphologiques, biochimiques, moléculaires et biomécaniques de la matrice cartilagineuse, entraînant un assouplissement, des fissures, des ulcérations et une disparition des cartilages des articulations, ainsi qu'une sclérose des os et la formation d'ostéophytes et de géodes. La pathologie entraîne des douleurs et des raideurs articulaires, ainsi qu'un éventuel gonflement de l'articulation, avec des degrés variables d'inflammation locale". Elle se caractérise donc par une modification phénotypique et métabolique des cellules du cartilage : les chondrocytes, entraînant un déséquilibre entre anabolisme et catabolisme en faveur du catabolisme. Dans notre travail, nous nous intéressons particulièrement à l'arthrose du genou.

2.2 Arthrose de Genou : Gonarthrose

2.2.1 Définition

L'arthrose du genou, appelée gonarthrose, est une dégradation du cartilage des articulations qui permettent les mouvements du genou par le glissement des surfaces cartilagineuses (Figure 4. 2). Dans la plupart des cas, elle touche l'articulation fémoro-tibiale. Cependant, elle peut également toucher l'articulation fémoro-patellaire. Dans quelques cas, les deux articulations sont touchées simultanément. [114]. L'arthrose du genou est une véritable maladie qui impacte la vie quotidienne des personnes atteintes. Les principales contraintes associées à la maladie sont la douleur et les limitations fonctionnelles dues à l'enflure et/ou à la raideur du genou. À cela s'ajoute d'autres symptômes comme une déformation progressive (en genou *varum* ou genou *valgum*) et une diminution de l'espace compris entre le fémur et le tibia, allant jusqu'au contact entre les deux (pincement de l'interligne articulaire). Ces symptômes peuvent entraîner un isolement social et un syndrome dépressif chez les personnes les plus touchées [115]. Par conséquent, les personnes souffrant d'arthrose connaissent une diminution importante de leur qualité de vie, affectant l'une ou plusieurs de ses dimensions [116].

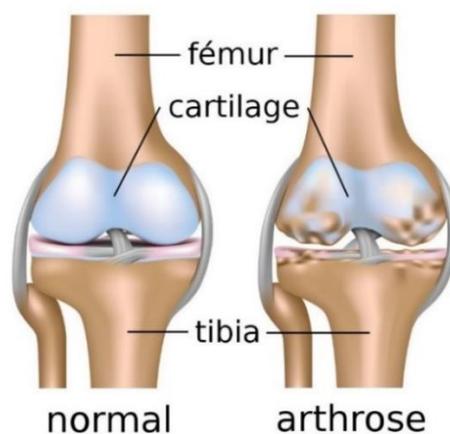


Figure 4. 2 : Arthrose de Genou

La prise en charge de l'arthrose du genou englobe diverses approches visant à soulager la douleur, à améliorer la mobilité et à optimiser la qualité de vie des patients. Elle comprend souvent des modalités telles que l'utilisation de médicaments anti-inflammatoires et analgésiques, la physiothérapie pour renforcer les muscles et améliorer la fonction articulaire, et des dispositifs de soutien tels que des orthèses. Les interventions plus avancées, comme les injections intra-articulaires et, dans les cas sévères, la chirurgie, peuvent également être envisagées. La perte de poids, l'adoption d'un mode de vie actif et l'éducation du patient sur la gestion autonome de la maladie jouent également un rôle clé. Une approche personnalisée, élaborée en concertation avec les professionnels de la santé, est essentielle pour répondre aux besoins spécifiques de chaque patient, en tenant compte du stade de l'arthrose et des objectifs de traitement [117].

2.2.2 Facteurs de risques

Il est difficile de déterminer précisément les origines de l'ostéoarthrite. Celle-ci est un processus qui résulte d'une combinaison de facteurs locaux (traumatismes et exercice physique), systémiques (obésité, alimentation), mécaniques, cellulaires et biochimiques, ainsi que de facteurs liés aux tissus.

Parmi les principaux éléments responsables des symptômes de l'arthrose du genou citons : l'obésité (odds ratio = 2,66) et les traumatismes antérieurs (odds ratio = 2,83) [118]. Certains de ces facteurs peuvent être corrigés, tels que la prise de poids, tandis que d'autres sont incontournables, à savoir l'âge et le sexe (Tableau 4. 1).

Tableau 4. 1 : Facteurs de risque liés à l'arthrose de Genou

Facteurs globaux	Facteurs bio-mécaniques
<ul style="list-style-type: none"> ▪ Sexe ▪ Conditions métaboliques ▪ Age ▪ Ethnicité ▪ Origine génétique 	<ul style="list-style-type: none"> ▪ Travail intensif ▪ Problèmes d'obésité ▪ Blessures articulaires

2.3 Maladie de Parkinson

Les personnes atteintes de la maladie de Parkinson sont généralement âgées de 45 à 70 ans. C'est la deuxième maladie neurodégénérative la plus fréquente après la maladie d'Alzheimer. La maladie de Parkinson peut présenter des formes particulièrement sévères et résistantes au traitement. Elle se caractérise par une akinésie (difficulté à déclencher un mouvement suivi d'une perte de vitesse), une rigidité et un tremblement latents. Il existe aujourd'hui des traitements dont l'efficacité est limitée et qui ne permettent que d'atténuer les symptômes sans arrêter ni même ralentir la progression de la pathologie. Si le fonctionnement du Parkinson est connu, son origine reste encore

inconnue, mais on pense qu'elle résulte d'une combinaison de facteurs environnementaux et de facteurs génétiques déterminants (Figure 4. 3). La prise en charge de la maladie de Parkinson constitue un défi complexe, nécessitant une approche multidisciplinaire pour améliorer la qualité de vie des patients. Les traitements actuels visent à atténuer les symptômes, tels que les tremblements, la rigidité musculaire et la bradykinésie, tout en cherchant à ralentir la progression de la maladie. Les médicaments dopaminergiques demeurent au cœur du traitement pharmacologique, offrant un soulagement symptomatique significatif. Cependant, la gestion à long terme de la maladie exige une adaptation constante des schémas thérapeutiques en raison de la variabilité des symptômes et des éventuels effets secondaires des médicaments. En parallèle, les interventions non pharmacologiques, telles que la physiothérapie, l'ergothérapie et la stimulation cérébrale profonde, jouent un rôle crucial dans l'amélioration de la fonction motrice et de la qualité de vie. Une approche holistique, intégrant les aspects médicaux, psychologiques et sociaux, est essentielle pour optimiser la prise en charge globale de la maladie de Parkinson et répondre aux besoins individuels des patients [119] [120].

Nous avons choisi d'intégrer dans notre démarche l'analyse de la marche des patients pour vérifier l'aptitude de nos approches à différencier l'expression au niveau du mouvement des deux pathologies, arthrose du genou et Parkinson.

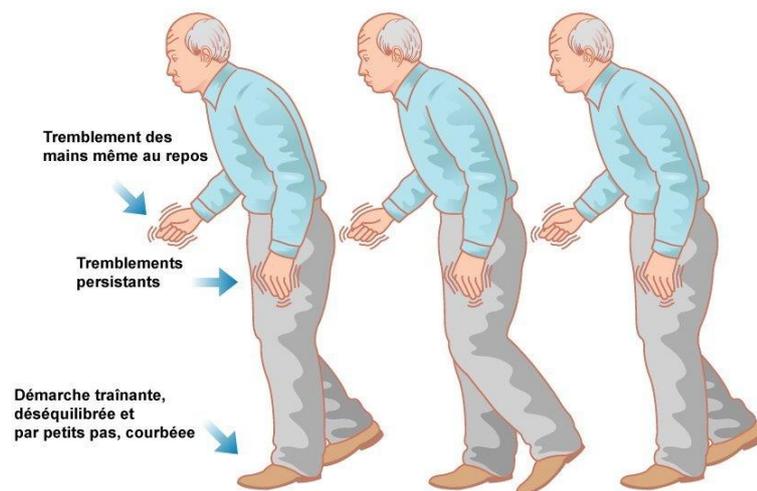


Figure 4. 3 : La maladie de Parkinson

3 Classification et Prédiction des Pathologies liées aux Gestes Humains par algorithmes : Revue de littérature

La catégorisation des pathologies liées aux gestes, comme la maladie de Parkinson ou l'arthrose du genou, est un sujet qui intéresse de nombreuses recherches. De nombreux experts qui ont étudié diverses méthodes basées sur les DNN ont permis des avancées dans le diagnostic et le

suivi de ces pathologies complexes. Les auteurs dans [121], ont mis au point une méthode basée sur un CNN visant à améliorer la classification de la gravité de l'arthrose à partir d'images radiographiques. Les résultats obtenus indiquent que cette méthode est plus précise que les méthodes classiques, à savoir 77,24 %. Dans l'étude [122], les auteurs ont étendu leurs travaux antérieurs sur l'analyse de la relation entre l'épaisseur du cartilage sur radiographies et le degré de gravité de l'arthrose, en intégrant l'analyse de la marche. Un réseau de neurone artificiel était utilisé comme méthode d'apprentissage automatique et selon les résultats de l'expérience, l'ajout de points de rotule a amélioré la performance de manière remarquable, passant d'une AUC de 82.2% à une AUC de 90.3%. Dans l'article [123], les auteurs ont proposé une méthode de détection des troubles de la marche fondée sur les DNN. Ils ont validé leur méthode dans le cadre de l'identification des individus atteints par la maladie de Parkinson (PD) Le modèle DNN élaboré a atteint une précision de classification moyenne de 85,7 % et un F1-Score de 85%. Une technique basée sur la vision par ordinateur a été proposée dans [124] pour l'analyse de la démarche visant à classer les gestes normaux ou ceux de la maladie de Parkinson, en utilisant une caméra RVB. Une méthode de réseau de neurones profonds R-CNN destinée à la segmentation des objets, est appliquée pour extraire les silhouettes humaines à partir de séquences vidéo. Celle-ci permet de classer avec succès les marches normales/Parkinson avec un taux de précision supérieur à 97 %. L'étude [125] propose un système intelligent basé sur un CNN, pour diagnostiquer le Parkinson et la gravité de l'atteinte en analysant des données de la marche. Les expériences ont été réalisées sur la base de données Physionet, et le modèle proposé s'est avéré efficace et dépasse les méthodes similaires existantes sur le plan des performances de classification. Un réseau CNN a été mis en œuvre pour la classification de la maladie de Parkinson afin d'aider les médecins à diagnostiquer la pathologie à un stade précoce [126]. La précision de classification a atteint 88,1 %. Pour les travaux d'analyse de la marche sans marqueurs, sujet de notre étude, nous citons l'étude [127]. Les auteurs ont montré le potentiel des techniques sans marqueur pour la prédiction de la maladie de Parkinson. Dans un premier temps, les sujets font l'objet d'une extraction des caractéristiques de la marche à l'aide d'Alphapose à partir de vidéo brute. Ensuite, huit caractéristiques sont retenues et analysées, et cinq d'entre elles sont systématiquement sélectionnées, qui offre une précision de 93 %, une Accuracy de 96 % et une sensibilité de 92 %. Dans [128], un cadre a été proposé pour classer précisément la démarche anormale (KOA et PD) et la démarche normale (NM) en utilisant une approche fondée sur la vision. L'objectif de cette étude est quadruple : Premièrement, un nouvel ensemble de données de marche a été créé et présenté ; deuxièmement, la segmentation de la région d'intérêt (ROI) a été réalisée à l'aide d'une technique améliorée ; troisièmement, des paramètres de marche, c'est-à-dire des caractéristiques spatiotemporelles (SPT) et d'autres caractéristiques de mouvement de corps ont été

évaluées en utilisant les méthodes statistiques ; enfin, la classification des sujets KOA, PD et NM a été effectuée avec le KNN. Les résultats révèlent la pertinence des caractéristiques de la marche prises en compte dans cette étude avec une précision allant jusqu'à 90,06 %, un degré de sensibilité égal à 85,54 %, ainsi qu'une spécificité de 90,44 % et une précision de 89,66 %. Dans la même famille des méthodes DNN d'analyse de la marche sans marqueurs, nous citons les GCN (Graph Convolutional Networks) [129] qui se sont révélés être des techniques prometteuses dans le domaine de la classification des pathologies liées à la posture. Ces réseaux conviennent parfaitement au traitement des graphes et leur utilisation devient de plus en plus populaire dans le but d'améliorer la précision des diagnostics dans des domaines tels que la classification de la maladie de Parkinson et de l'arthrose du genou [130] [131]. Dans [132], les auteurs ont présenté la première implémentation du réseau convolutif à base de graphes spatio-temporels (STGCN) basée sur le modèle BlazePose pour la reconnaissance d'action humaine. La précision obtenue est de 40,1 % sur un ensemble de données acquises par la caméra Kinetics. Pour l'ensemble de données de la base « NTU-RGB+D », la précision obtenue est de 87,59 % et de 92,1 % pour les modèles Cross-Stick et NTU-RGB+D, respectivement. Dans [133], les auteurs de cette étude suggèrent une nouvelle méthode de modélisation spatio-temporelle (WM-STGCN) pour l'analyse de la marche pour la classification de la maladie de Parkinson. En outre, diverses stratégies sont utilisées pour augmenter les données du squelette. Selon les résultats expérimentaux, la méthode proposée a permis d'obtenir une précision de 87,5 % surpassant les modèles LSTM, KNN et arbre de décision.

Tableau 4. 2: Récapitulatif sur les modèles DNN relatifs à la classification des pathologies liées aux gestes humains

<i>Auteurs</i>	<i>Objectif de l'étude</i>	<i>Architectures Utilisées</i>	<i>Performance</i>
[123]	La détection des troubles de la marche pour identifier les personnes atteintes de la maladie de Parkinson.	ANN	Accuracy : 85,7 % F1-Score de 85%
[121]	La classification de la gravité de l'arthrose de genou.	CNN	Accuracy : 77,24 %
[124]	L'analyse de la marche visant à distinguer les gestes normaux de ceux de la maladie de Parkinson.	R-CNN	Accuracy : 97 %
[126]	La classification de la maladie de Parkinson afin d'aider les médecins à diagnostiquer la maladie à un stade précoce.	CNN	Accuracy : 88.1%
[127]	La prédiction de la maladie de Parkinson	CNN	Précision : 93 % Accuracy : 96 % Sensibilité : 92 %
[128]	L'analyse de la marche pour la classification des personnes atteintes de la maladie de	KNN	Accuracy : 90.06% Sensibilité/ 85.54%

	Parkinson, de KOA et des individus sains.		Spécificité : 90.44% Précision de 89.66%
[133]	L'analyse de la marche pour la classification de la maladie de Parkinson	WM-STGCN	Accuracy : 87.5%
[132]	La reconnaissance d'action humaine	STGCN	Accuracy « NTU-RGBD » :87,59% Accuracy « Cross-Stick NTU-RGB+D »: 92.1%
[134]	Le diagnostic de la maladie de Parkinson	B-GCN	Accuracy : 95,59 %

Comme le montre le Tableau 4. 2, une revue des modèles de classification pour les maladies liées au mouvement telles que la maladie de Parkinson et l'arthrose met en évidence diverses approches, allant des méthodes classiques d'apprentissage automatique telles que SVM et MLP à des modèles d'apprentissage profond tels que CNN. Néanmoins, le réseau convolutionnels sur graphes (GCN) se révèle être une solution prometteuse. Le réseau GCN se distingue par sa capacité à traiter les données graphiquement structurées, ce qui est essentiel pour notre objectif de classification de vidéos liées au mouvement humain. En effet, il intègre les liens étroits entre les différentes régions corporelles et les séquences temporelles, ce qui améliore considérablement les résultats de la classification. Pour cette raison, nous allons utiliser GCN.

4 Modélisation du corps humain

La modélisation des corps humains représente une partie importante de l'estimation de pose qui permet de visualiser les points clés et les fonctionnalités extraites de données d'entrée. Il existe généralement trois types de modèles pour la modélisation du corps humain (Figure 4. 4). À savoir, le modèle cinématique qui estime la pose en 2D/3D, le modèle planaire servant à estimer la pose en 2D, ou encore le modèle volumétrique destiné à l'estimation de la pose en 3D.

4.1 Modèle cinématique

Le modèle cinématique utilise un ensemble de positions des articulations ainsi que les orientations des membres afin de représenter la structure du corps humain.

4.2 Modèle planaire

Le modèle planaire est utilisée pour représenter la forme et l'apparence d'un corps humain, comme le montre la Figure 4. 4(b). Selon le modèle planaire, la représentation des parties du corps se fait généralement par des rectangles qui épousent approximativement les contours du corps humain.

4.3 Modèle volumétrique

Avec l'intérêt croissant que suscite la reconstruction 3D du corps humain, plusieurs modèles volumétriques du corps humain (dont un modèle donné dans la Figure 4. 4(c)) ont été proposés pour une large sélection de corps humains.

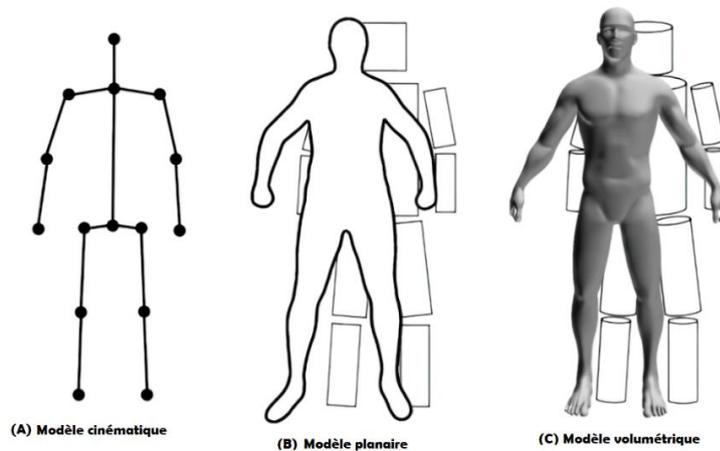


Figure 4. 4: Modélisation du corps humain [135]

5 Estimation de pose humaine

Avec les techniques d'estimation de la pose humaine, il est possible d'estimer, à partir d'une vidéo, les positions 2D/3D ou les emplacements spatiaux de points clefs du corps humain (Figure 4. 5). Une bonne méthode d'estimation de la pose doit être robuste aux variations de l'apparence humaine, liées aux très nombreux degrés de liberté que représentent les articulations, mais aussi aux vêtements ou à la coiffure. Pour relever ces défis, les approches fondées sur les DNN ont fait une véritable avancée dans l'estimation de pose en améliorant considérablement la performance.

On distingue deux types d'estimation de la pose humaine (Figure 4. 6) :

- Méthodes relatives à l'estimation de la pose humaine pour un seul individu.
- Méthodes relatives à l'estimation de la pose pour plusieurs individus.

Les approches fondées sur une seule personne sont réparties en deux catégories en fonction de la manière dont elles prédisent les points clés : les techniques basées sur la régression et les approches fondées sur les cartes thermiques. Les premières utilisent les cartes de caractéristiques (feature maps) de sortie pour régresser directement les points clés, tandis que les secondes génèrent d'abord des Heatmaps (la valeur du pixel dans la heatmaps indique la probabilité d'existence du point clé à cette position) et prédisent les points clés sur la base de ces cartes. D'autre part les approches multi-personnes peuvent être classées en deux catégories selon leur méthodologie : les approches descendantes (Top-down) et les approches ascendantes (Bottom-up). Les approches descendantes s'exécutent en deux étapes : la détection des personnes et l'estimation des points clés pour chaque

personne. Les méthodes ascendantes suivent des étapes similaires mais dans un ordre inversé : la première étape consiste à localiser tous les éléments clés d'une image et la seconde à regrouper ces points clés en fonction de la personne à laquelle ils appartiennent.

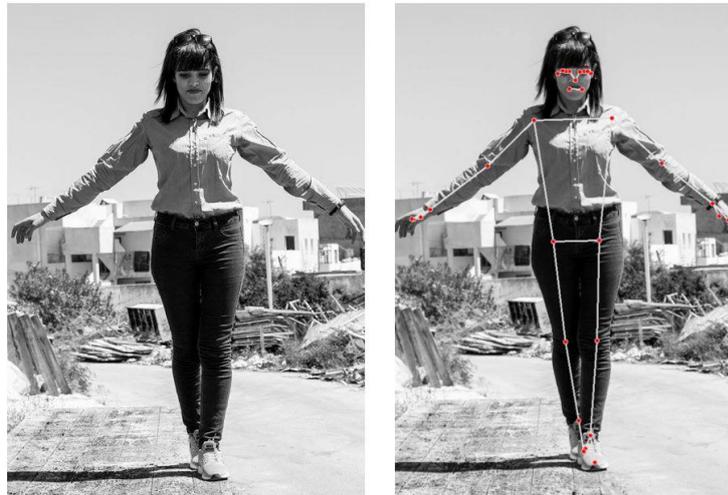


Figure 4. 5 : Estimation de la pose humaine

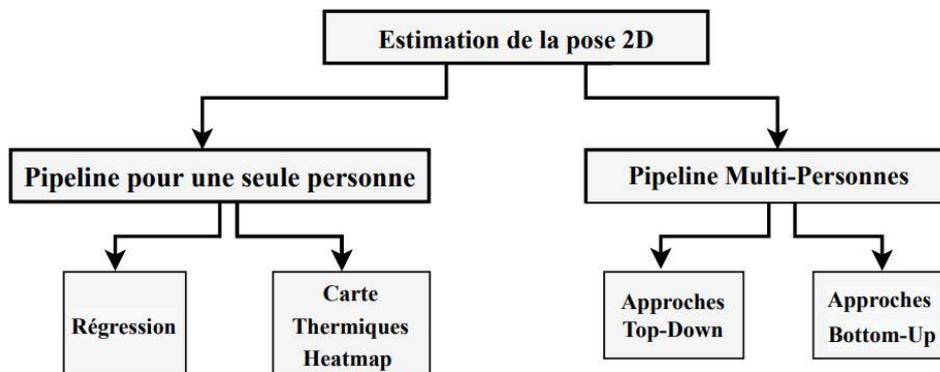


Figure 4. 6 : Taxonomie de l'estimation de la pose 2D

Dans la section suivante, nous discutons en détail certaines approches récentes, en nous focalisant sur les techniques estimant la pose pour une seule personne.

5.1 Estimation de la pose d'une seule personne

Les approches à une seule personne détectent la pose de la personne concernée dans une image ou une vidéo. Étant donné les informations de positions estimées, l'objectif de ces approches est de trouver la position des points clés (les articulations) dans chaque zone. En d'autres termes, cela revient à régler un problème de régression dans lequel le nombre de points clés est implicitement donné. Le modèle prend une image en entrée et comporte deux parties (Figure 4. 7). La première joue le rôle d'encodeur et va permettre, en utilisant des réseaux CNN, d'extraire les caractéristiques sous la forme d'un vecteur. La deuxième partie est une des méthodes fondées généralement sur la

régression ou sur les heatmaps. Cette deuxième partie joue donc le rôle de décodeur et permet d'estimer la pose humaine à partir du vecteur caractéristique.

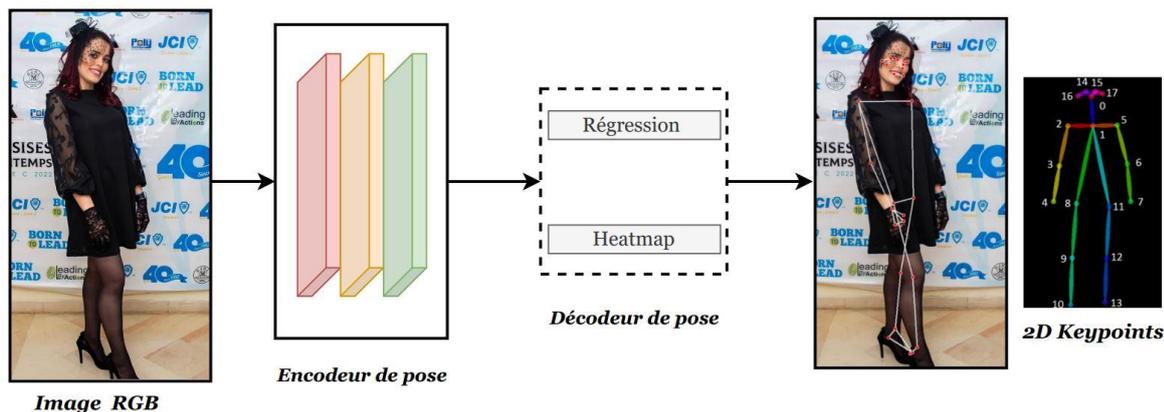


Figure 4. 7 : Système Codeur-Décodeur pour l'estimation de la pose humaine

5.2 Métriques d'évaluation

Pour évaluer un modèle d'estimation de pose à partir de points clés, la métrique choisie doit prendre en compte plusieurs facteurs et caractéristiques (par exemple, articulations du haut et du bas du corps humain, estimation de la pose simple ou multiple). Le Tableau 4. 3 décrit les différentes métriques d'évaluation des modèles d'estimation de la pose humaine.

Tableau 4. 3: Métriques d'évaluation des modèles d'estimation de la pose du corps humain

Métriques	Définition
<i>Pourcentage de parties correctes (PCP)</i>	Mesure le pourcentage de bonne détection des membres.
<i>Pourcentage des points clés corrects (PCK)</i>	Évalue la précision de localisation des différents points clés.
<i>Pourcentage de points clés corrects relativement à la taille de la tête avec un seuil de 0.5 (PCKh@0.5)</i>	Évalue la précision en tenant compte de la localisation des points clés par rapport à la taille de la tête, avec un seuil spécifique.
<i>Pourcentage d'articulations détectées (PDJ)</i>	Évalue la précision de détection des différentes articulations.
<i>Précision moyenne (AP)</i>	Mesure la proportion de vrais positifs parmi le nombre de réponses positives.
<i>Rappel moyen (AR)</i>	Mesure le ratio qui existe entre les vrais positifs et le nombre total de sujets (points/membres) à repérer.

6 Étude Bibliographique sur les techniques d'estimation de la pose humaine basées sur les DNN

Dans cette section, nous présentons la bibliographie spécifique à l'estimation de la pose basée DNN. L'estimation de la pose humaine a toujours été un problème difficile qui retient l'attention,

dû à un champ d'applications très large allant de la classification d'images à la reconnaissance d'activités. Un des défis majeurs est la localisation des points clés, eu égard à l'extrême variabilité des poses du corps humain. DeepPose [136] a été le premier Framework important portant sur le DL pour l'estimation de pose 2D. Il a atteint des performances élevées. DeepPose formule l'estimation de la pose comme une tâche de régression basée sur un CNN vis-à-vis des articulations du corps. En outre, les estimations de pose sont affinées en utilisant une cascade de régressions pour obtenir un meilleur résultat. L'approche proposée effectue l'estimation de la pose d'une manière globale, ou holistique. En d'autres termes, même si certaines articulations sont cachées, elles peuvent être estimées. Les auteurs soutiennent que les CNNs fournissent naturellement ce type de raisonnement holistique et démontrent de bons résultats. Les auteurs dans [137] ont adopté une technique de régression tenant compte de la structure, appelée "régression de pose compositionnelle", basée sur les os (arêtes du graphe) et la structure de la pose au lieu de la représentation traditionnelle basée sur les articulations (sommets du graphe). En outre, un réseau CNN combinant l'approche holistique et celle fondée sur l'apparence locale a été proposé dans [138] afin de réaliser deux tâches : la détection des articulations, qui détermine si une zone contient une articulation du corps, et la localisation des articulations, qui prédit l'emplacement exact de l'articulation dans la zone. Chaque tâche est associée à une fonction de perte, et la combinaison des deux tâches permet d'améliorer les résultats. Une autre méthode dite DeepCut [139] décrit la répartition des points détectés relative à chaque personne dans une image d'entrée donnée. Cette méthode permet d'améliorer les performances en introduisant des critères de sélection par paires liés à l'image. D'autres travaux ont employé la structure fondée sur les heatmaps, qui offrent des informations plus riches en termes de localisation, afin de faciliter l'entraînement des CNN. Les auteurs dans [140] ont combiné un système de détection des parties corporelles à base de CNN avec un module de localisation spatiale destiné à l'estimation de la pose 2D. Un modèle séquentiel basé sur les CNN dans [141] est introduit, appelé Convolutional Pose Machines (CPM), destiné à détecter l'emplacement des principales articulations grâce à un traitement multi-étapes (les CNN de chaque étape exploitent les cartographies 2D générées lors des étapes précédentes et produisent des prédictions de plus en plus précises de la localisation de certaines parties du corps). En outre dans [142], les auteurs ont combiné un système de détection des parties corporelles à base de CNN, à un module de localisation spatiale destiné à l'estimation de la pose 2D. Dans l'article [143], l'architecture ConvNet, une architecture CNN multi-résolution, est proposée afin de générer des heatmaps discrètes plutôt qu'une régression continue qui prédit la probabilité de l'emplacement de chaque articulation dans les images RVB. La performance du modèle est évaluée à l'aide de PCK et PCKh@0.5 sur les bases de données FLIC [144] et MPII [145] respectivement, dans lesquels les

résultats ont dépassé ceux des modèles de l'état de l'art. Dans [146], un modèle ConvNet pour prédire les poses de corps humains 2D a été mis en œuvre. Il est composé de deux parties : le module Feed-Forward et le module récurrent. Le module Feed-Forward agit principalement comme un détecteur de "parties" indépendantes, qui régresse les heatmaps de points clés, mais qui ignore largement le contexte de la configuration des autres parties, en raison des champs récepteurs résultants réduits. En revanche, le module récurrent apporte plus de contexte à chaque fois qu'il est exécuté, notamment parce que le champ récepteur résultant est augmenté à chaque itération. Les résultats expérimentaux révèlent que le modèle est capable de capturer le contexte de manière itérative, ce qui améliore les résultats de localisation. Les auteurs dans [147] ont proposé une architecture à deux niveaux avec un détecteur de personnes R-CNN qui crée des boîtes de délimitation des corps humains et un estimateur pour prédire l'emplacement des points clés en utilisant une agrégation de heatmaps. Cette approche fonctionne bien dans les scènes occultées et encombrées. Le succès des Generative Adversarial Networks (GAN) [148], a entraîné leur utilisation dans l'estimation de la pose humaine pour générer des configurations de pose physiologiquement plausibles, et pour ainsi différencier les prédictions fiables de celles douteuses, et aussi déduire la position probable des parties du corps occultées. Les auteurs dans [149] adoptent les réseaux GAN afin de faciliter la correction des prédictions de pose. Afin de bien capturer les corrélations structurelles entre les articulations du corps, le GAN est utilisé de manière multi-tâches pour prédire les poses et les cartes thermiques d'occultation. Cette méthode est donc particulièrement robuste aux occultations.

Il existe d'autres méthodes qui combinent à la fois les heatmaps et la régression. Dans [150], les auteurs ont proposé BlazePose, une architecture basée sur les CNN pour l'estimation de la pose humaine qui a été adaptée à l'inférence en temps réel sur les appareils mobiles. De plus, une nouvelle typologie qui utilise 33 points sur le corps humain a été utilisée dans le but de permettre des utilisations exigeantes en termes de performances, telles que le langage gestuel, le suivi du yoga et de l'activité physique. En outre, DeeperCut [151] est une version plus proche et améliorée de celle présentée dans DeepCut [139]. DeeperCut utilise une stratégie d'optimisation progressive qui permet d'explorer l'espace de recherche d'une façon efficace, ce qui conduit également à une meilleure performance. Grâce à l'adaptation de ResNet, ce travail a pu résoudre le problème de la dissipation des gradients. Dans [152], les auteurs exposent un Framework qui détecte et associe conjointement des points clés spatio-temporels en une seule étape, ce qui constitue un algorithme de détection et de suivi en temps réel. Ils présentent une architecture de réseau CNN générique qui utilise des champs complexes pour détecter et construire une pose spatio-temporelle sous la forme d'un graphe unique, dont les nœuds sont les points clés sémantiques (par exemple, les articulations

du corps d'une personne) dans plusieurs images. Les expériences montrent que la méthode offre une précision compétitive et qu'elle est plus rapide que les autres sur de nombreux ensembles de données publics, tels que COCO, ou CrowdPose. L'article [153] introduit une nouvelle architecture de réseaux de convolutions destinée à l'estimation de la pose humaine. Les caractéristiques sont regroupées afin de capturer de manière optimale les diverses relations spatiales associées avec le corps. Cette architecture est dénommée "stacked hourglass" en raison des étapes successives de Pooling et de upsampling effectuées afin de produire une série finale de prédictions. Les performances obtenues sur les benchmarks FLIC et MPII dépassent celles de toutes les techniques concurrentes.

L'estimation de la pose humaine fournit des informations quantitatives sur le mouvement humain qui permettent aux médecins de diagnostiquer certaines maladies complexes, de suivre un programme de rééducation et de gérer une thérapie physique. Les auteurs dans [154] ont élaboré un système d'estimation de la pose destiné à évaluer le degré de gravité de la maladie de Parkinson. Les auteurs dans [155] ont mis en œuvre un système de thérapie physique basé sur la pose pour évaluer et conseiller les patients à domicile. De plus, un tel système peut être utilisé pour détecter une action anormale et prédire les actions à venir. Des alertes sont envoyées immédiatement si le système estime une situation comme potentiellement dangereuse. Dans [156], les auteurs ont utilisé les algorithmes d'estimation de poses pour la surveillance et la détection des chutes afin de fournir une assistance instantanée. De plus, les méthodes d'estimation de pose sont capables de produire des informations fiables sur la posture de patients en milieu hospitalier, ce qui permet de renforcer la recherche sur la corrélation entre l'état neurologique et la posture.

Tableau 4. 4: Récapitulatif sur les modèles relatifs à l'estimation de la pose 2D ou 3D

<i>Modèles</i>	<i>Architecture backbone</i>	<i>Single/Multi Person</i>	<i>Nombre des points clés (kp, 2d ou 3d)</i>	<i>Base de données</i>	<i>Métriques d'évaluation</i>
DeepPose	AlexNet	Single Person	17 kp 2d	LSP, FLIC	PDJ, PCP
DeepCut	R-CNN	Single Person	17 kp 3d	LSP, MPII	PCK, AUC
CPM	VGGNet	Single Person	17 kp 2d	LSP, FLIC, MPII	PCK@0.1 PCK@0.2 PCK@0.5
ConvNet Pose	ConvNet	Single Person	17 kp 3d	FLIC, MPII	PCK PCKh@0.5 PCK@0.05
BlazePose	CNN	Single Person	33 kp 3d	Yoga Dataset	PCK@0.2

Stacked Hourglass	ResNet	Single Person	17 kp 3d	FLIC, MII	PCK PCKh@0.5 PCK@0.2
Openpipaf	ResNet50	Single/multi-Person	17 kp 3d	CrowdPose, COCO	AP
Open Pose	CNN	Single/Multi-Person	17 kp 3d	MII, COCO	AP, AR, MAP
Adversarial Pose Net	GAN	Single/multi-Person	17 kp 2d	LSP, MII	PCK@0.2 PCKh@0.5
MoveNet	MobileNetV2	Single Person	17 kp 3d	COCO, Active Dataset	AP
Alpha Pose	CNN	Single/multi-Person	17 kp 3d	MSCOCO Keypoints Challenge, MII	AP

Nous constatons qu'il y a une grande variété d'approches d'estimation de pose (Tableau 4. 4). Ces méthodes varient en termes de précision, du nombre de points clés détectés et de la capacité à suivre les mouvements du corps humain. Après avoir soigneusement analysé ces alternatives, nous avons opté pour BlazePose essentiellement pour sa capacité à détecter un nombre plus élevé de points que les autres méthodes. Cette caractéristique est particulièrement avantageuse pour nos objectifs, car elle nous permet d'obtenir une représentation plus riche et plus détaillée de la pose du corps humain, en particulier dans la perspective d'une exploration fine de la structure locale. La finalité est de proposer une approche pour la classification des atteintes d'arthrose du genou basée DNN à partir de séquences vidéo acquises sans marqueurs. L'originalité sera sur trois plans : (1) une méthode basée DNN et vidéo sans marqueurs pour la classification des arthroses du genou, (2) une estimation des poses globale, puis régionale, (3) l'optimisation GS-TPE des performances des réseaux retenus.

7 Implémentation des architectures DNN à base de graphes pour la classification de l'arthrose du Genou

7.1 Base de données relative à la marche

Dans la présente thèse, nous avons choisi de travailler avec des données publiques appelées "Gait Dataset for Knee Osteoarthritis and Parkinson's Disease Analysis With Severity Levels" (ensemble de données sur la démarche pour l'analyse de l'arthrose du genou et de la maladie de

Parkinson avec des niveaux de gravité) [157]. Cette base de données comprend 96 sujets, dont 50 sujets atteints d'arthrose du genou (KOA), 16 sujets atteints de la maladie de Parkinson (PD) et 30 sujets normaux ou sains (NM). Les données relatives aux personnes atteintes de KOA et de PD ont été recueillies dans différents hôpitaux, et des volontaires en bonne santé ont aussi été engagés pour analyser les déviations de la démarche. Pour évaluer la gravité des maladies, des cliniciens experts ont utilisé les échelles KL et H&Y pour le KOA et la maladie de Parkinson (PD) respectivement, comme indiqué dans les travaux [158] et [159]. La base est composée de vidéos capturées par une seule caméra NIKON DSLR 5300 placée à 8 mètres du tapis de marche dans le secteur de l'hôpital. Afin de faciliter le suivi, un ensemble de 6 marqueurs réfléchissants passifs de couleur rouge ont été fixés sur les articulations de chaque sujet. Le principal objectif de cette base est d'examiner les déviations observées dans la marche des patients par rapport à celle d'individus ayant une démarche normale. Le Tableau 4. 5 présente brièvement l'ensemble de données.

Tableau 4. 5: Description détaillée de la base de données KOA-PD-NM en indiquant les sujets concernés et leurs données biologiques.

	<i>Niveau de gravité</i>	<i>Âge moyen</i>	<i>Taille moyenne (En m)</i>	<i>Sujets par genre</i>
<i>Sujets Normaux</i>	Sain/Normal (NM)	45	1.6	30 : 17F, 13M
<i>Knee Osteoarthritis (KOA)</i>	Précoce (EL)	47.1	1.54	15 : 14F, 1M
	Modéré (MD)	59.8	1.58	20 : 12F, 8M
	Sévère (SV)	62.4	1.54	15 : 12F, 3M
<i>Parkinson (PD)</i>	Précoce (EL)	66.5	1.67	6 M
	Modéré (MD)	69.8	1.61	7 : 2F, 5M
	Sévère (SV)	70	1.66	3 : 1F, 2M

7.2 Méthodologie

Dans le cadre de notre première expérimentation, nous avons élaboré dans un premier temps un pipeline d'analyse de vidéos médicales issues de la base présentée ci-dessus NM-PD-KOA. Nous adoptons dans notre approche BlazePose pour extraire les points clés du corps, suivi d'une étape de normalisation, et enfin nous utilisons ST-GCN pour la classification.

7.2.1 Pipeline de BlazePose-STGCN : 3D Keypoints Tracking pour la classification de l'arthrose du genou

Cette section présente en détail le processus proposé pour la classification de KOA et de la maladie de Parkinson basée sur des vidéos, ainsi que les multiples étapes qui y sont liées (Figure 4. 8). Tout d'abord, les vidéos sont traitées à l'aide de la technique BlazePose, qui permet d'extraire et de suivre les articulations du corps pour obtenir une séquence de points clés du squelette en 3D avec des coordonnées (x, y, z). Deux expérimentations distinctes seront menées : la première consistera à extraire l'ensemble des articulations identifiées par BlazePose (33 points), tandis que la seconde se focalise sur des articulations spécifiques, déterminées en fonction des zones d'intérêt définies par les rhumatologues (8 points autour du genou). Ces données sont désormais une séquence de graphes, dont les nœuds sont les articulations du corps, étiquetées avec leurs coordonnées 3D et d'autres caractéristiques si nécessaire. Celles-ci sont ensuite introduites dans le STGCN qui produit des résultats de classification pour faire la distinction entre les sujets normaux et pathologiques, en particulier pour la détection de KOA et de PD à l'aide d'un classificateur binaire softmax. Nous détaillons ci-après les différentes parties du processus présenté.

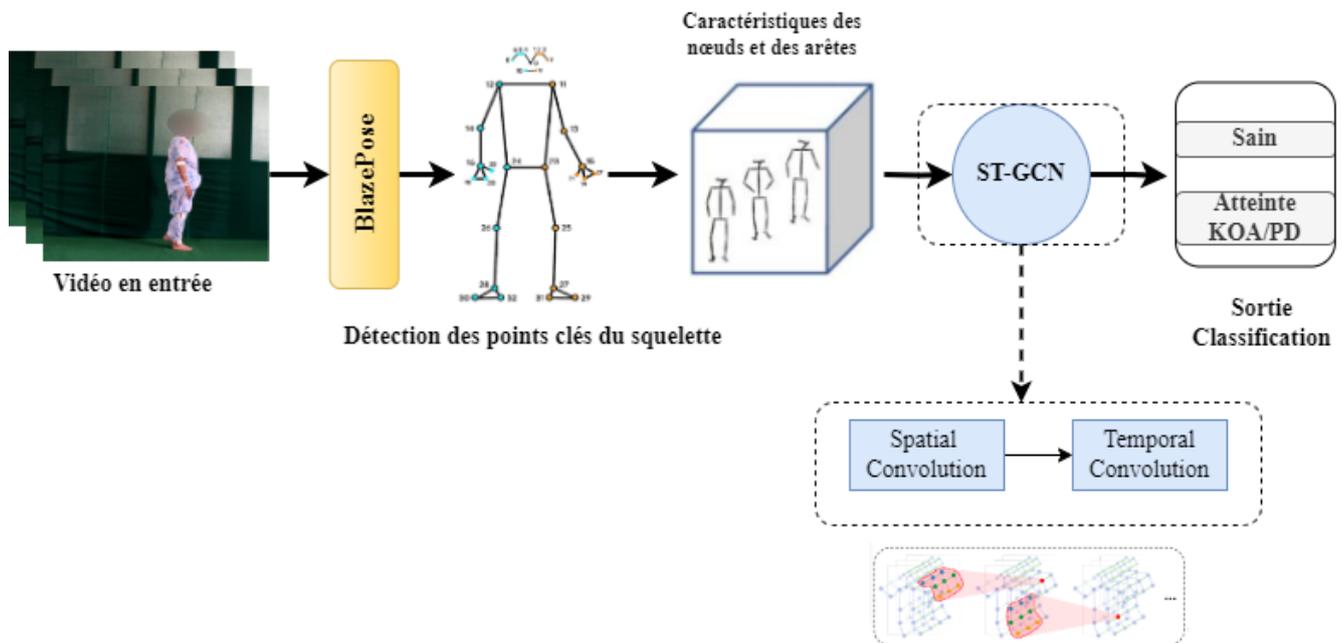


Figure 4. 8 : Processus de BlazePose-STGCN

7.2.1.1 Extraction de squelette via BlazePose : Détection des articulations sans marqueurs

Nous avons utilisé la technique BlazePose pour l'analyse vidéo et l'extraction des points clés. Cette méthode sans marqueur permet de détecter les coordonnées (x,y,z) pour chaque articulation dans des vidéos peu contrôlées, contrairement aux méthodes traditionnelles qui exigent de placer

les marqueurs sur le corps des sujets [128] [157]. Une des caractéristiques remarquables de BlazePose est sa capacité à détecter et à suivre 33 articulations du corps humain. Ces derniers forment des repères anatomiques essentiels tels que les coudes, les poignets, les hanches, etc (Figure 4. 9). En capturant ces points clés, BlazePose offre une représentation compacte et complète de la pose et du mouvement du corps. Le concept de base de BlazePose consiste à utiliser un DNN pour déterminer l'emplacement des articulations en 2D et en 3D à partir d'images RVB ou de séquences vidéo. Ce réseau a été entraîné sur un grand nombre de données étiquetées, lui permettant d'apprendre les relations spatiales et les modèles visuels des différentes articulations. Cela lui permet de détecter les articulations de manière robuste et précise, même dans des scénarios complexes et dynamiques [150]. Par ailleurs, une des caractéristiques importantes de BlazePose réside au niveau de ses performances en temps réel : il est exécuté à des taux de fréquence élevés, ce qui permet le suivi en temps réel des 33 articulations. Grâce à ce grand nombre d'articulations, il est possible d'améliorer considérablement la performance des travaux précédents, qui se limitaient à six articulations indiquées par l'ajout de marqueurs sur le corps [128].

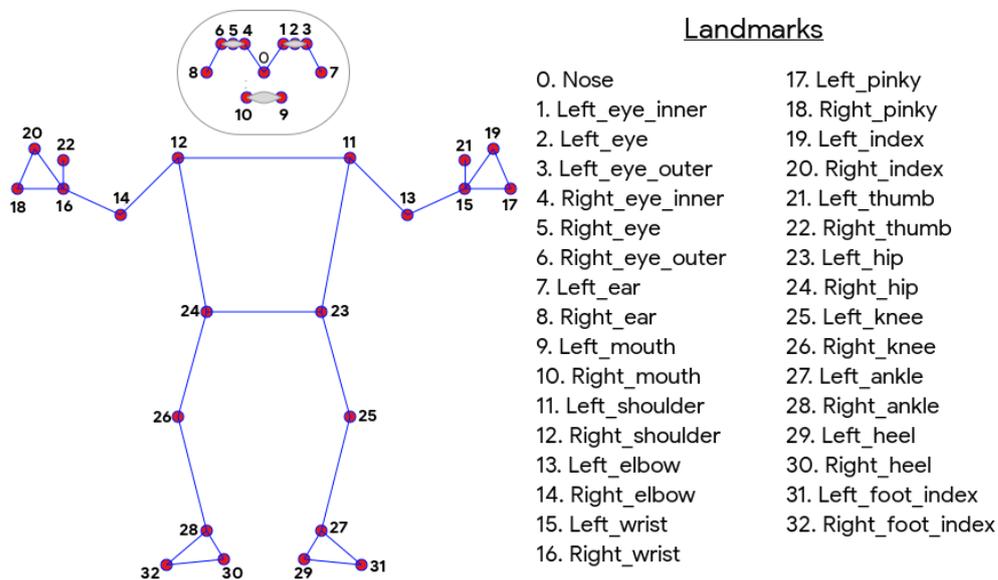


Figure 4. 9 : BlazePose

Nous nous concentrons sur le suivi de k articulations du corps pendant une séquence de marche capturée dans une vidéo avec T images, pour laquelle BlazePose fournit une séquence de coordonnées 3D (équation(4. 1)) :

$$\{ (x_i^t, y_i^t, z_i^t); i = 1 \dots k, t = 1 \dots T \} \quad (4. 1)$$

La séquence de squelettes qui en résulte, comme le montre la Figure 4. 10, résume la marche du sujet.



Figure 4. 10 : Séquence illustrative à base de squelette à l'aide de BlazePose

Ainsi, nous envisageons d'utiliser BlazePose de deux manières distinctes, globale et locale. Dans un premier temps, nous exploiterons cette technologie pour extraire la totalité des 33 articulations (globale). Pour la seconde approche, notre modèle se focalisera sur une région précise, notamment les articulations liées au genou (locale). Cette démarche s'inscrit dans le cadre de notre étude sur l'arthrose du genou, où les indications de notre experte partenaire rhumatologue nous ont orientés vers des articulations spécifiques qu'elle juge cruciales pour l'évaluation de la maladie.

Les articulations considérées comme pertinentes par la rhumatologue pour l'évaluation de l'arthrose du genou comprennent les points suivants (Figure 4. 11) :

- 25 : *Genou gauche*
- 26 : *Genou droit*
- 27 : *Cheville gauche*
- 28 : *Cheville droite*
- 29 : *Talon gauche*
- 30 : *Talon droit*
- 31 : *Index du pied gauche*
- 32 : *Index du pied droit*

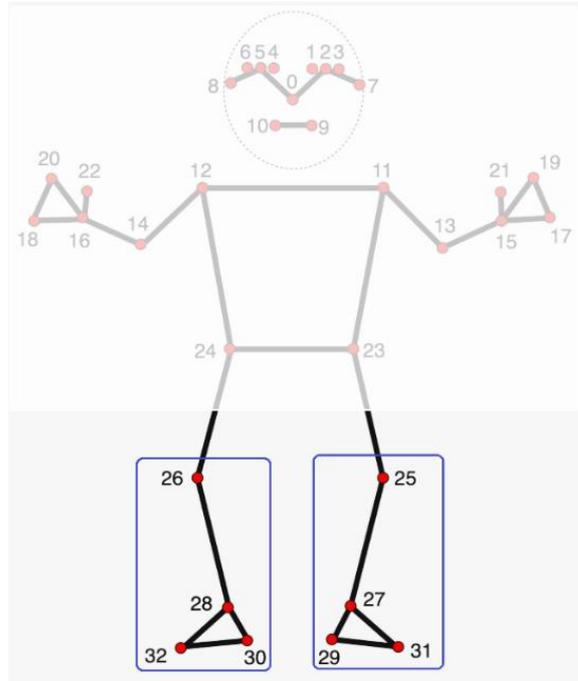


Figure 4. 11 : Articulations spécifiques de BlazePose

7.2.1.2 STGCN: Spatial-Temporal Graph Convolutional Network

STGCN est un modèle qui a été utilisé dans divers domaines, y compris la reconnaissance d'actions. Il utilise des réseaux de convolution de graphes (GCN) pour traiter les données du squelette [129]. Dans le cadre de la classification des pathologies liées au gestes humains, nous avons adapté ce modèle pour la classification de l'arthrose de Genou ainsi que le Parkinson. Le réseau STGCN est une architecture DNN spécialement conçue pour traiter les graphes spatio-temporels, ce qui le rend particulièrement adapté aux tâches qui impliquent des informations séquentielles ou temporelles.

- *Construction du graphe*

Une démarche humaine forme une séquence constituée de vecteurs qui représentent les coordonnées 3D des articulations humaines. En utilisant ST-GCN [129], des graphes spatio-temporels ont été créés pour représenter les informations structurées dans les séquences de squelettes. Nous définissons un graphe non orienté $G = (V, E)$ avec une séquence de squelette composée de N articulations (dans notre cas $N=33$ ou $N=8$) et de T images. Dans le graphe, les sommets $V = \{v_{ti} | t = 1, \dots, T, i = 1, \dots, N\}$ représentent toutes les articulations dans une séquence de squelettes. La figure 4.12(a) fournit une représentation graphique de cette structure, où chaque nœud (sommets) représente une articulation (skeleton joint).

Quant à l'ensemble des arêtes E , on peut le partitionner en 2 sous-ensembles, E_s et E_t . Le premier forme les arêtes spatiales (lignes vertes dans la Figure 4. 12(a)), qui comprennent toutes les

connexions naturelles du corps humain dans une image donnée, $E_s = \{(V_{ii}, V_{ij})\}$ Le deuxième forme les arêtes temporelles (lignes rouges dans la Figure 4. 12(a)), qui résultent des connexions d'articulations analogues entre deux images adjacentes : $E_t = \{(V_{ii}, V_{(t+1)i})\}$.

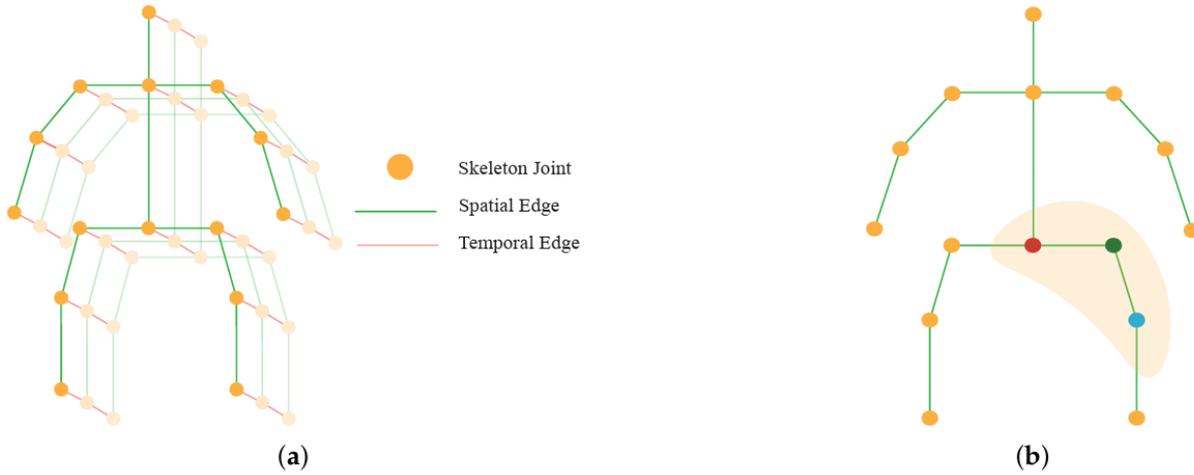


Figure 4. 12 : (a) Séquence de graphes de squelettes (b) Représentation des différentes articulations dans le graphe selon leur position

▪ **Matrice d'adjacence**

Dans le cadre du ST-GCN, l'idée centrale réside dans la représentation spatiotemporelle des séquences temporelles articulaires. Chaque image de squelette est initialement conceptualisée comme un graphe indépendant, noté $G = (V, E)$ comme mentionné précédemment. La matrice d'adjacence A exprime ces connexions, où $A_{ij} = 1$ indique la présence d'une connexion entre les articulations i et j , et $A_{ij} = 0$ signifie l'absence de connexion (Figure 4. 13). L'extension vers le ST-GCN introduit une dimension temporelle dans le graphe, créant ainsi un graphe spatiotemporel. Ce graphe étendu combine à la fois les informations spatiales et temporelles. La matrice d'adjacence spécifique au ST-GCN encapsule ces connexions spatiotemporelles.

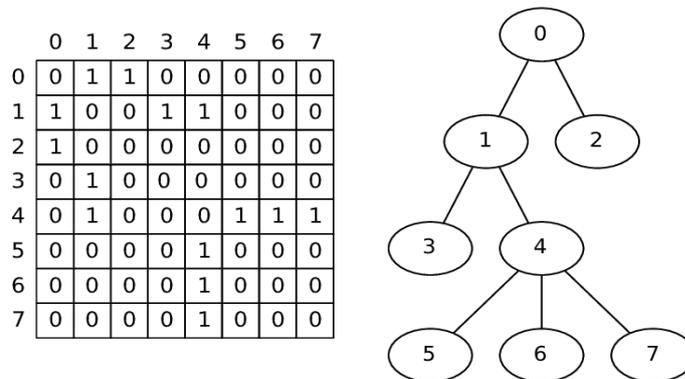


Figure 4. 13: Graphe non orienté et sa matrice d'adjacence

- **Stratégies d'échantillonnage et de partitionnement**

Une fois que la base du graphe est établie avec la matrice d'adjacence, il est indispensable de présenter les stratégies d'échantillonnage et de partitionnement. Le traitement des convolutions sur des images en 2D est simple et consiste à utiliser une grille régulière avec un rectangle représentant le support spatial. Cependant, dans le contexte des graphes, la topologie n'est plus régulière et le support spatial (voisinage) de chaque point traité est défini comme ses nœuds adjacents. La Figure 4. 14(b) illustre ce concept pour une seule image : les nœuds encerclés représentent la zone d'échantillonnage du filtre de convolution appliqué à chaque point rouge. Pour résumer, la stratégie d'échantillonnage employée par le STGCN [129] est basée sur les dépendances mécaniques entre les articulations plutôt que sur le voisinage spatial. La stratégie de partitionnement a pour but de regrouper les articulations en fonction des caractéristiques des mouvements humains, que l'on peut classer [9] selon qu'il s'agit de mouvements concentriques ou excentriques, en divisant les points de la région d'échantillonnage en trois sous-ensembles :

- Le nœud racine (ou articulation centrale), surligné en vert dans la Figure 4. 14(c).
- Le groupe central (en bleu sur la Figure 4. 14(c)), composé des nœuds les plus proches du centre de gravité du squelette (représenté par la croix noire).
- Le groupe centrifuge (en jaune sur la Figure 4. 14(c)), qui comprend les nœuds les plus éloignés du centre de gravité.

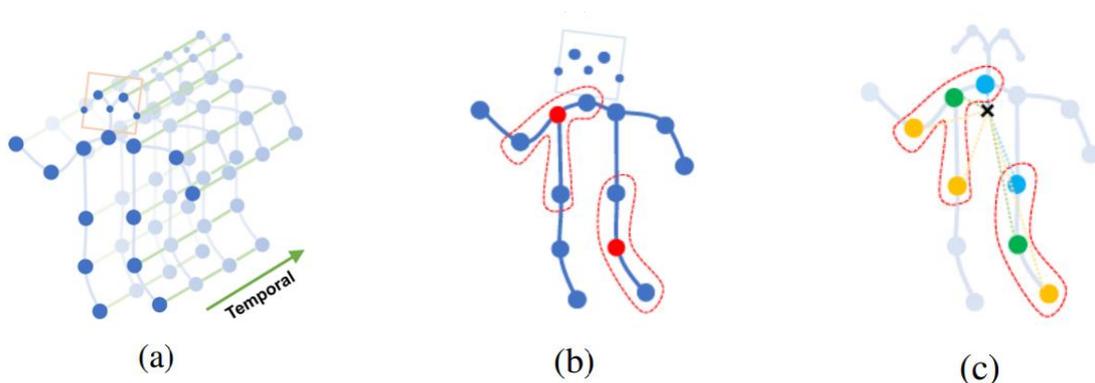


Figure 4. 14: (a) Séquence de graphes de squelettes, représentant le mouvement humain dans le temps et l'espace. (b) Stratégie d'échantillonnage d'une couche de convolution pour une seule image. (c) Stratégie de partitionnement de la configuration spatiale

Dans le modèle STGCN, le centre de gravité est défini comme étant la moyenne des coordonnées de toutes les articulations du squelette dans une image. Grâce au partitionnement de la configuration spatiale [129], chaque articulation reçoit une étiquette basée sur des partitions. Ces partitions

définissent des poids distincts pour le modèle, ce qui facilite l'analyse spatiale et l'apprentissage. Pour capturer la dimension temporelle, le STGCN étend le concept de convolution des graphes évoqué précédemment en considérant cette dimension comme une séquence de squelettes de graphes empilés consécutivement, comme le montre la Figure 4. 14(a). Il en résulte un ensemble de graphes voisins. Sachant que chaque articulation d'un graphe doit être reliée par une arête à l'articulation correspondante dans les images voisines précédente et suivante, STGCN applique efficacement les convolutions spatiotemporelles aux données de la séquence de poses humaines.

- *Architecture du modèle*

La structure du STGCN utilisé pour classifier l'arthrose du genou est composée de 10 blocs STGC (Figure 4. 15), dont chacun est conçu pour extraire des informations importantes à partir des coordonnées du squelette des vidéos. Ils fonctionnent séquentiellement, chacun utilisant la sortie du bloc précédent comme entrée, ce qui permet de capturer progressivement des caractéristiques spatio-temporelles plus complexes. À la suite du dernier bloc STGC, une couche de Global Pooling agrège les caractéristiques, ce qui permet d'obtenir une représentation intégrale du mouvement. Finalement, une couche entièrement connectée assure la classification, en utilisant les caractéristiques agrégées pour identifier la présence ou l'absence d'arthrose du genou.

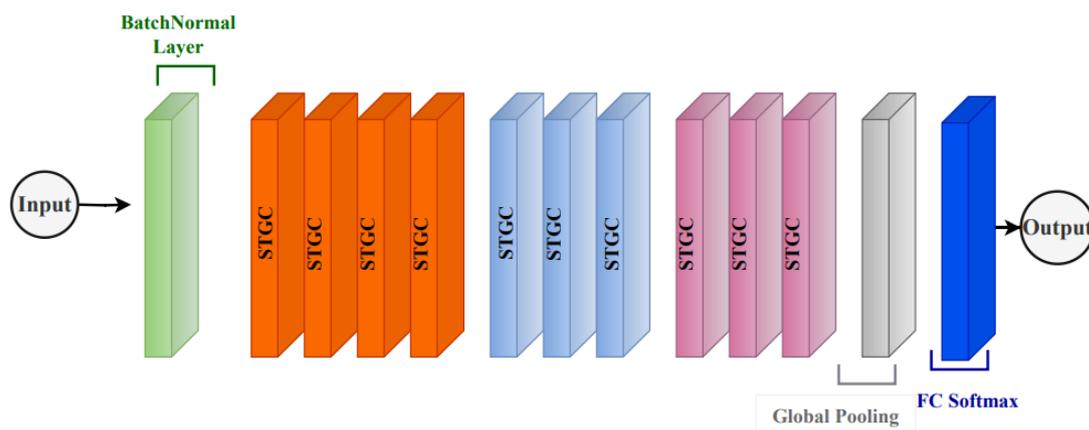


Figure 4. 15: Architecture STGCN

Chaque unité STGC (Figure 4. 16) intègre deux composants importants pour la compréhension des mouvements humains à partir des données du squelette : la Convolution de Graphe (GCN) et le Temporal Convolutional Network (TCN). Tout d'abord, le GCN applique une couche de convolution au graphe construit à partir des coordonnées des articulations. Cette opération exploite les relations spatiales entre les articulations (intra-squelette), en identifiant des motifs complexes dans la structure du squelette. La sortie de cette étape contient des informations spatio-temporelles essentielles, fournissant une représentation riche et dynamique des interactions entre les différentes

parties du corps. La sortie du GCN est ensuite transmise à une couche TCN. Ce dernier joue un rôle essentiel dans la modélisation des relations temporelles inter-images. Elle utilise pour ce faire une fenêtre temporelle constante, ce qui permet de capturer les variations temporelles tout au long de la séquence. Enfin, les vecteurs de caractéristiques combinés, spatiaux et temporels, sont introduits dans un classificateur constitué d'une couche entièrement connectée normalisée par un Softmax.

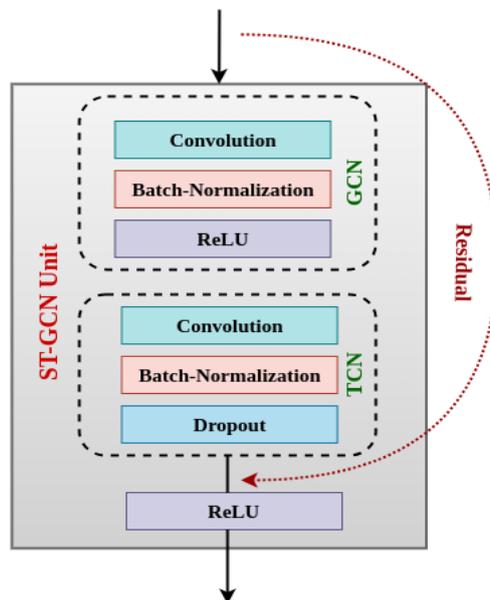


Figure 4. 16 : Bloc STGC

7.3 Expérimentation

En reprenant le Tableau 4. 5 de la présentation de la base de données utilisée et nous décrivons ci-après l’harmonisation de la durée des vidéos. Notre choix consiste à faire un découpage en vidéos de durée constante de 4s (soit $4 \times 50 = 200$ frames). Ce choix nous est imposé par la durée minimale des vidéos qui est de 4s pour les sujets sains. Nous avons éliminé les vidéos vides ou illisibles, ainsi que les vidéos des trois patients PD à atteinte sévère du fait qu’ils sont accompagnés par un infirmier lors de l’enregistrement vidéo. Nous avons résumé dans le Tableau 4. 6 notre base de données modifiée. Le nombre final de vidéos exploitables à longueur unifiée de 4s (200 frames) est de 317 alors qu’il était de 186 vidéos à durées variables.

Tableau 4. 6 : Description de la base de données à durée normalisée

<i>Patients</i>	<i>Nombre de vidéo (Durée min : 4s, Durée max : 24s)</i>	<i>Nombre de vidéos exploitables de durées 200 frames</i>	<i>Total par classe</i>
Sains	60	59	59
KAO légère	30	45	206
KAO Modérée	40	72	
KAO Sévère	30	89	
PD Légère	12	24	52

PD Modérée	14	28	
Total	186	317	317

Cette démarche nous a permis d'harmoniser les durées des vidéos et d'augmenter le nombre de nos échantillons. La deuxième variable dans nos expérimentations est le nombre de point clés ou d'articulations suivies. Nous avons opté pour deux configurations, une première dite globale à 33 points et une deuxième locale à 8 points. De ce fait l'entrée de notre modèle sera de dimension $T \times V \times C$, avec T le nombre de frames (ici 200), V le nombre de points clés ou articulations (33 ou 8) et C le nombre de coordonnées de chaque point (fixées à 3). Le Tableau 4. 7 résume les différentes configurations testées.

Tableau 4. 7 : Description des configurations de validation de notre pipeline

	Type de classification	Nombre de frames par vidéo	Nombre de vidéos	Nombre d'articulations	Optimisation des HP du réseau
Analyse globale					
Expérience 1	NM vs KOA	200	265	33	Non
Expérience 2	KOA vs (NM+PD_EL)	200	289	33	Non
Expérience 3	KOA vs (NM+PD_EL_MD)	200	317	33	Non
Expérience 4	NM vs KOA vs PD	200	317	33	Non
Expérience 5	NM vs KOA_EL vs KOA_MD vs KOA_SV	200	265	33	Non
Analyse locale					
Expérience 6	NM vs KOA	200	265	8	Non
Expérience 7	NM vs KOA	200	265	8	Oui
Expérience 8	KOA vs (NM+PD_EL)	200	289	8	Non
Expérience 9	KOA vs (NM+PD_EL)	200	289	8	Oui
Expérience 10	KOA vs (NM+PD_EL_MD)	200	317	8	Non
Expérience 11	KOA vs (NM+PD_EL_MD)	200	317	8	Oui
Expérience 12	NM vs KOA vs PD	200	317	8	Non
Expérience 13	NM vs KOA vs PD	200	317	8	Oui
Expérience 14	NM vs KOA_EL vs KOA_MD vs KOA_SV	200	265	8	Non
Expérience 15	NM vs KOA_EL vs KOA_MD vs KOA_SV	200	265	8	Oui

Il en résulte 15 expériences réalisées. La première consiste à différencier les sains des KOA. La deuxième consiste à fusionner les données des sujets NM avec celles des patients à un stade précoce de Parkinson (PD_EL) dans le but d'analyser l'influence de cette combinaison sur les résultats de la différenciation. La troisième consiste à fusionner les données des sujets NM avec celles des patients à des stades précoce et modéré de Parkinson (PD_EL+MD) afin d'évaluer la robustesse du modèle STGCN à cette nouvelle perturbation. La quatrième consiste à classifier les trois états : sains,

arthrose et Parkinson. La cinquième consiste à classifier les quatre états : sain, arthrose précoce, arthrose modérée et arthrose sévère.

Dans les dix expériences suivantes de 6 à 15, nous avons repris les 5 premières configurations mais avec un nombre de points clés de 8 (analyse locale). Les numéros pairs ici pour l'analyse locale étant sans optimisation des HP, les numéros pairs, avec. Dans les expériences sans processus d'optimisation, les données ont été divisées en bases d'entraînement (80%) et de test (20%). Chaque vidéo présente une seule personne. Pour les expériences avec optimisation des HP nous avons divisé la partie d'entraînement (80%) en deux parties : 60% pour l'entraînement et réservé le 20% restant pour la validation. Et nous avons conservé la même partie test qui est 20%

Dans le Tableau 4. 8 et le Tableau 4. 9, nous présentons les différentes composantes et la dimension résultante des données d'entrée dans le cas de l'étude globale et locale respectivement.

Tableau 4. 8 : Présentation du paramétrage d'une étude globale à 33 points (expériences de 1 à 10)

Paramètres	Représentation	Valeur
Nombre de canaux	C	3
Nombre de frames par vidéo	T	200
Nombre d'articulations	V	33
Dimension des données	$3 \times 200 \times 33 = 19\ 800$	

Tableau 4. 9 : Présentation du paramétrage d'une étude globale à 8 points (expériences de 11 à 20)

Paramètres	Représentation	Valeur
Nombre de canaux	C	3
Nombre de frames par vidéo	T	200
Nombre d'articulations	V	8
Dimension des données	$3 \times 200 \times 8 = 4\ 800$	

7.3.1 Environnement de travail

Nous avons implémenté le modèle STGCN à l'aide de PyTorch dans un environnement Ubuntu. La configuration matérielle comprenait un GPU NVIDIA GeForce RTX 3090 avec 32 Go et un processeur 11e Gen Intel® core™ i9-11900F. Pour faciliter l'entraînement et le test du modèle de classification de la démarche KOA, nous avons utilisé CUDA, OpenCV et d'autres bibliothèques essentielles.

7.3.2 Métriques de performance

Dans notre contexte, les vrais positifs (TP) correspondent au nombre de patients correctement classés comme atteints de la maladie, tandis que les vrais négatifs (TN) correspondent aux individus

correctement classés comme sains. Les faux positifs (FP) représentent les individus sains classés comme malades par erreur, et les faux négatifs (FN) représentent les patients malades classés comme sains par erreur.

- L'accuracy fournit une indication globale de la manière dont le modèle de classification permet de distinguer les cas des patients malades des cas sains.
- Une sensibilité (ou Recall) élevée indique qu'il y a peu de maladies négligées.
- Une forte spécificité signifie que le modèle peut éliminer la majorité des sujets non malades.
- La précision évalue la proportion de personnes réellement malades parmi celles classées comme telles.

7.3.3 HPs et modalités d'apprentissage dans les expériences sans processus d'optimisation

Nous avons entraîné le réseau STGCN sur 120 époques avec un batch size de 8. Un taux d'apprentissage de 0,01 a été sélectionné et, pour les époques 40 et 80, le taux d'apprentissage a été réduit par un facteur 10 et on a utilisé l'optimiseur SGD. Enfin, pour éviter l'overfitting, un dropout rate de 0.5 a été utilisé (Tableau 4. 10).

Tableau 4. 10: HPs utilisés dans les expériences sans processus d'optimisation

HP	Valeur
Optimiseur	SGD
Nombre d'époques	120
Batch size	8
Learning rate	0.01
Dropout rate	0.5

7.3.3.1 Résultats et discussion avec l'ensemble des articulations BlazePose (33 points, analyse globale)

Nous rapportons dans le Tableau 4. 11 les performances des expériences qui consistent à différencier les sujets sains par rapport à ceux atteints par l'arthrose ou la maladie de Parkinson avec un nombre de points clés de 33 (analyse globale).

Tableau 4. 11 : Résultats expérimentaux de l'analyse globale à 33 points pour la classification de l'arthrose de genou et de la maladie de Parkinson

	Type de classification	Nombre de classes	Nombre d'articulations	Métriques de performances			
				Accuracy	Sensibilité	Spécificité	Précision
Expérience 1	NM vs KOA	2	33	100%	100%	100%	100%
Expérience 2	(NM+PD_EL) vs KOA	2	33	91.37%	81.25%	95.23%	86.66%
Expérience 3	(NM+ PD_EL_MD) vs KOA	2	33	79.68%	57.89%	88.88%	68.75%
Expérience 4	NM vs KOA vs PD	3	33	78.12%	75.37%	75.37%	75.37%
Expérience 5	NM vs KOA_EL vs KOA_MD vs KOA_SV	4	33	81.13%	87.59%	81.13%	87.59%
Moyenne				86.06%	80.42%	88.122%	83.674%

Pour la première expérience à deux classes NM et KOA, les performances sont maximales. La matrice de confusion ci-dessous précise la répartition (Figure 4. 17).

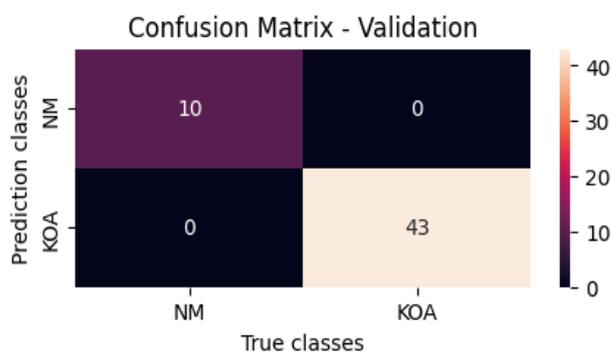


Figure 4. 17 : Matrice de confusion pour l'expérience 1

Pour les expériences 2 et 3 nous avons injecté des éléments perturbateurs. Pour la configuration 2, nous avons intégré les patients atteints de Parkinson précoce (PD_EL) dans le groupe des sujets sains et pour la configuration 3 nous avons associé à ce groupe les malades atteints de Parkinson précoce et modéré (PD_EL_MD). Dans la matrice de confusion correspondant à l'expérience 2 (Figure 4. 18), nous remarquons que le classifieur commence à faire une confusion entre les groupes. Cette tendance se confirme dans l'expérience 3 et l'erreur est plus importante surtout dans la classe des "patients sains" intégrant Parkinson précoce et modéré (Figure 4. 19).

Dans l'expérience 4 où nous avons comme objectif de différencier les trois groupes NM, KOA et PD, notre méthode classe bien les NM mais il fait la confusion entre Parkinson et arthrose. C'est la matrice de confusion de la Figure 4. 20.

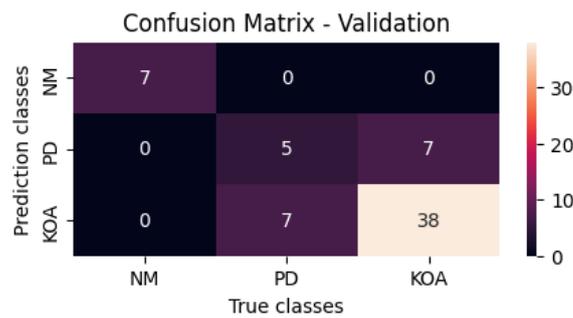
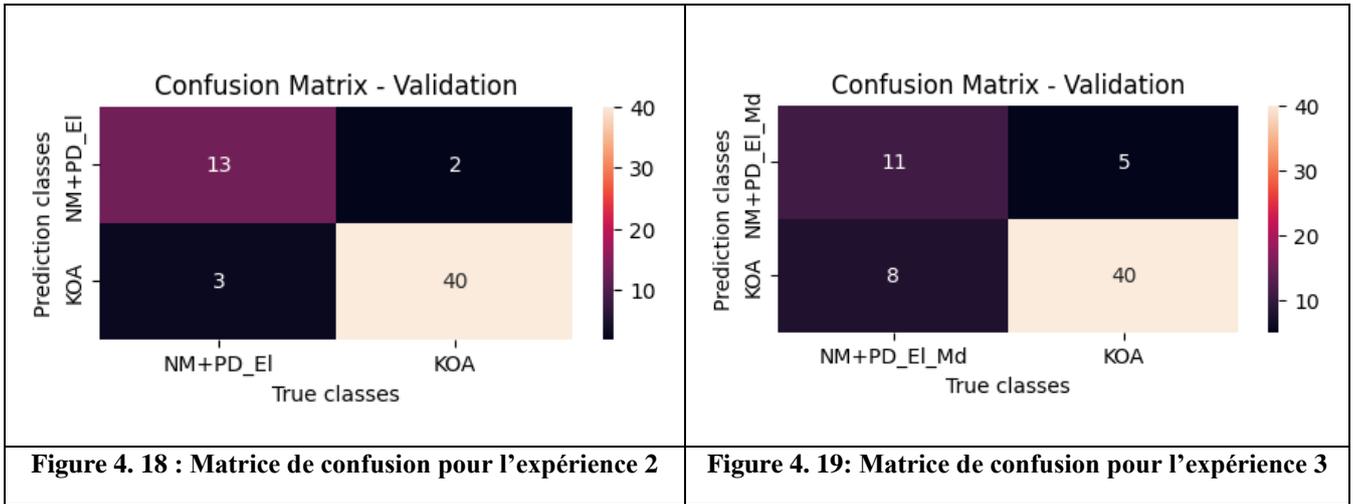


Figure 4. 20 : Matrice de confusion pour l'expérience 4

L'expérience 5 consiste à classifier les sujets en fonction de leur niveau d'arthrose. Nous obtenons des résultats excellents pour la classification des sains par rapport aux sujets à arthrose précoce. Ce résultat est intéressant surtout pour prédire la survenue de l'arthrose précocement. Nous notons une confusion entre arthrose modérée et sévère qui peut être acceptée à ce stade du fait que nous n'avons la possibilité d'interpeler le clinicien ou de contacter de nouveau le patient (base publique) (Figure 4. 21). L'intérêt pour nous est d'avoir nos propres patients pour vérifier ces confusions.

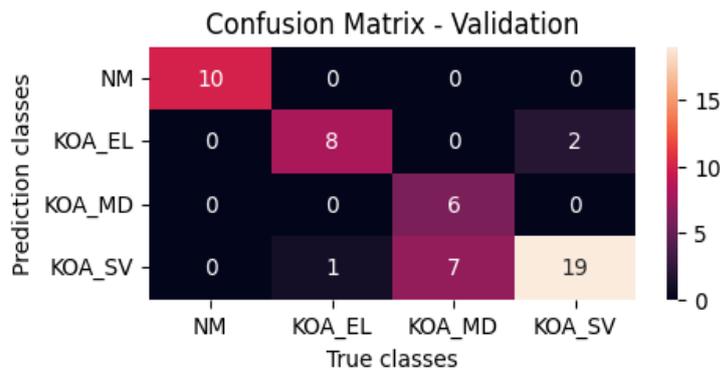


Figure 4. 21: : Matrice de confusion pour l'expérience 5

7.3.3.2 Résultats et discussion avec un nombre limité d'articulations BlazePose (analyse locale à 8 points)

Nous avons repris les mêmes expériences en réalisant une analyse locale avec 8 points clés autour du genou. Les résultats restent de qualité. Une comparaison entre l'analyse globale et l'analyse locale montre que :

- L'analyse globale est meilleure pour la différenciation entre NM et KOA (Figure 4. 17 *versus* Figure 4. 22).
- L'analyse globale est meilleure aussi pour la différenciation entre les différents stades de la KOA (Figure 4. 21 *versus* Figure 4. 26).
- L'analyse locale est plus performante dans la classification NM *vs* KOA *vs* PD surtout pour bien classer les KOA. Cette voie devrait être mieux explorée avec une base plus riche (Figure 4. 20 *versus* Figure 4. 25).

Tableau 4. 12 : Résultats expérimentaux de l'analyse locale à 8 points pour la classification de l'arthrose du genou et de la maladie de Parkinson

	Type de classification	Nombre de classes	Nombres d'articulations	Métriques de performances			
				Accuracy	Sensibilité	Spécificité	Précision
Expérience 6	NM <i>vs</i> KOA	2	8	96.22%	90%	97.76%	90%
Expérience 8	KOA <i>vs</i> (NM+PD_EL)	2	8	89.65%	68.75%	97.61%	91.66%
Expérience 10	KOA <i>vs</i> (NM+PD_EL_MD)	2	8	75%	52.63%	84.44%	58.82%
Expérience 12	NM <i>vs</i> KOA <i>vs</i> PD	3	8	81.25%	92.98%	92.98%	64.68%
Expérience 14	NM <i>vs</i> KOAP <i>vs</i> KOAM <i>vs</i> KOAS	4	8	71.69%	80%	80%	69.11%
Moyenne				82.762%	76.822%	90.558%	74.854%

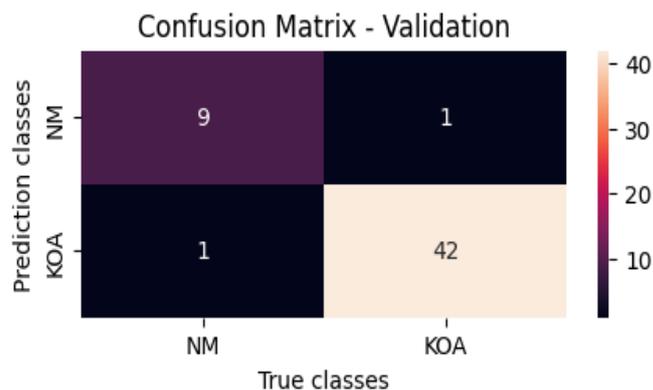


Figure 4. 22: Matrice de confusion pour l'expérience 6

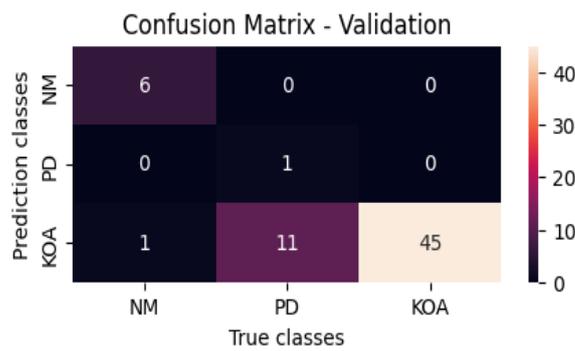
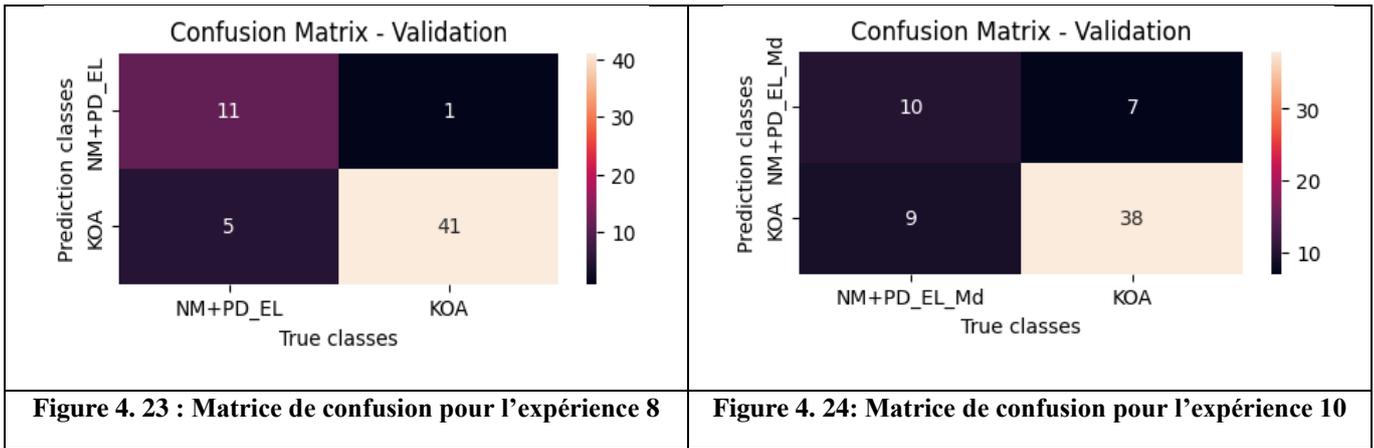


Figure 4. 25: Matrice de confusion pour l'expérience 12

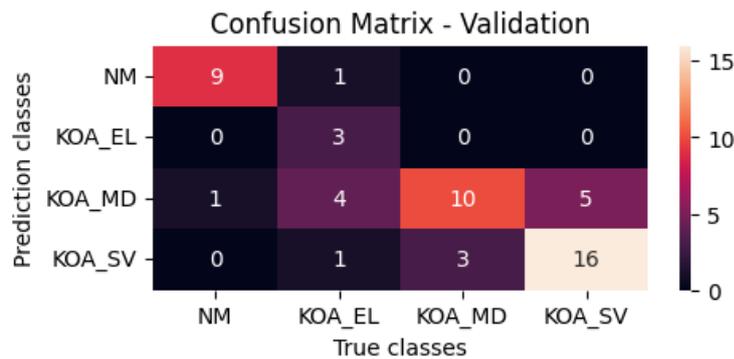


Figure 4. 26 : Matrice de confusion pour l'expérience 14

7.3.4 Optimisation des hyperparamètres de l'architecture ST-GCN

Comme pour le chapitre précédent, l'optimisation des HPs du réseau STGCN est utile pour améliorer les performances de classification de l'arthrose du genou. Nous envisageons deux types d'HPs :

- Les HPs liés à l'architecture.
- Les HPs liés à l'apprentissage.

En ce qui concerne les HPs liés à l'architecture, tels que le nombre de canaux dans chaque couche STGCN, la stratégie de Pooling spatial-temporel, et le taux de dropout, ils exercent une influence directe sur la complexité et la capacité du modèle à extraire des caractéristiques pertinentes. La taille des filtres temporels est particulièrement importante, impactant la capacité du modèle à extraire des caractéristiques spatio-temporelles pertinentes. Il est essentiel de sélectionner cette taille de manière minutieuse pour capturer efficacement les relations présentes dans le graphe spatio-temporel associé aux données médicales. D'un autre côté, les HPs liés à l'apprentissage, tels que le Learning rate, l'optimiseur et le batch size, jouent un rôle important dans la convergence du modèle lors de l'entraînement.

Tableau 4. 13 : HPs clés à optimiser pour l'architecture STGCN

HPs liés à l'architecture	<i>Domaine</i>	<i>Step pour GS</i>	<i>Bornes des HP utilisées par TPE</i>
Ratio entre les nombres de canaux dans chaque couche STGCN	{1,3}	1	---
Temporal Kernel Size	{7,11}	2	---
HPs liés à l'apprentissage			
Learning_rate	$\mathbb{R}[10^{-2}, 10^{-4}]$	10^{-1}	$\pm 10^{-5}$
Batch size	{4,8}	4	± 2

Nous avons utilisé l'approche GS-TPE afin d'optimiser les HPs de l'architecture STGCN. Initialement, nous avons effectué une recherche exhaustive à l'aide de GS sur un ensemble de 54 combinaisons d'HPs (voir Tableau 4. 13). Par la suite, nous avons appliqué TPE (en fixant le nombre d'itérations à 60) autour des meilleures configurations fournies par GS. À chaque itération de TPE, nous avons entraîné le modèle pendant 100 epochs.

En raison de contraintes matérielles liées au temps de calcul et à la disponibilité des ressources de calcul, nous avons choisi de ne sélectionner que les trois meilleures configurations identifiées par GS pour être optimisées par TPE.

7.3.4.1 Résultats expérimentaux et discussion

Le Tableau ci-dessous présente les résultats en termes d'exactitude obtenus après l'optimisation des HPs avec l'approche GS-TPE dans les différentes expériences de l'analyse locale. Nous notons une amélioration significative par rapport à la version non optimisée, comme indiqué dans le Tableau 4. 12. Initialement, l'approche GS a été utilisée pour identifier les meilleures configurations, comme illustré dans le Tableau 4.15. En raison des limitations de ressources de

calcul, seules trois configurations ont été sélectionnées. Ensuite, en appliquant des bornes sur chaque HP de ces configurations (voir Tableau 4. 13), TPE a exploré de nouveaux espaces pour améliorer les résultats. Le Tableau 4.15 montre que TPE a réussi à améliorer les résultats en termes d'exactitude des trois configurations pour différentes expériences. Dans l'expérience 11, la configuration 2 s'est révélée la meilleure, avec une exactitude de 90.20% sur les données de validation. Pour les expériences 13 et 15, c'est la configuration 1 qui, après raffinement par TPE, a fourni les meilleurs résultats. Les HPs de chaque configuration sélectionnée ont été appliqués pour évaluer le modèle STGCN sur les données de test. Selon le Tableau ci-dessous, une amélioration significative des performances sur les données de test a été observée pour la classification NM vs KOA, atteignant 98.11% par rapport aux résultats initiaux non optimisés qui étaient de 96.22 %. Notamment, dans l'expérience 11, où les patients PD ont été introduits comme éléments perturbateurs, une amélioration des résultats de 75% à 84.37% a été observée.

Tableau 4. 14 : Résultats expérimentaux obtenus avec optimisation des HP de STGCN pour l'analyse locale à 8 points pour la classification de l'arthrose de genou et de la maladie de Parkinson

		<i>Nbr classe</i>	<i>Nombre d'articulations</i>	<i>Accuracy sans optimisation (%)</i>	<i>Accuracy avec GS-TPE (%)</i>
				<i>Acc-Test</i>	<i>Acc-Test</i>
<i>Expérience 7</i>	NM vs KOA	2	8	96.22	98.11
<i>Expérience 9</i>	KOA vs (NM+PD_EL)	2	8	89.65	89.65
<i>Expérience 11</i>	KOA vs (NM+PD_EL_MD)	2	8	75	84.37
<i>Expérience 13</i>	NM vs KOA vs PD	3	8	81.25	82.81
<i>Expérience 15</i>	NM vs KOA_EL vs KOA_MD vs KOA_SV	4	8	71.69	73.57

Tableau 4. 15 : Performance des configurations GS-TPE en termes d'accuracy sur les données de validation en utilisant STGCN

		<i>Accuracy_validation (%)</i>	
		<i>GS</i>	<i>TPE</i>
<i>Expérience 7</i>	<i>Configuration-1</i>	100	100
	<i>Configuration-2</i>	100	100
	<i>Configuration-3</i>	97.67	100
<hr/>			
<i>Expérience 9</i>	<i>Configuration-1</i>	89.36	91.49
	<i>Configuration-2</i>	89.36	91.48
	<i>Configuration-3</i>	87.23	91.48
<hr/>			
<i>Expérience 11</i>	<i>Configuration-1</i>	76.47	88.24
	<i>Configuration-2</i>	74.50	90.20
	<i>Configuration-3</i>	74.50	88.24
<hr/>			
<i>Expérience 13</i>	<i>Configuration-1</i>	82.35	86.27
	<i>Configuration-2</i>	82.35	84.31
	<i>Configuration-3</i>	80.39	84.31
<hr/>			
<i>Expérience 15</i>	<i>Configuration-1</i>	72.09	81.40
	<i>Configuration-2</i>	72.09	74.41
	<i>Configuration-3</i>	69.76	69.76

Les figures ci-dessous présentent graphiquement les résultats contenus dans le Tableau ci-dessus, illustrant la variation de la précision des 3 configurations de chaque expérience en fonction du nombre d'itérations du TPE. Pour l'expérience 9, l'analyse révèle que la meilleure performance a été obtenue à l'itération 19, tandis que pour l'expérience 11, caractérisée par la présence de patients atteints de PD en tant qu'éléments perturbateurs, la meilleure performance est observée à l'itération 21.

L'expérience 13, qui implique une classification à 3 classes (NM vs KOA vs PD), a nécessité un nombre plus élevé d'itérations pour atteindre sa meilleure performance, qui a été observée à l'itération 54. Cela suggère que la nature même de la tâche de classification, avec trois classes distinctes, peut nécessiter une exploration plus approfondie de l'espace des HPs et une optimisation plus minutieuse du modèle pour atteindre des performances optimales. Ainsi, même si l'expérience 11 présente également des données complexes en introduisant des patients PD comme éléments perturbateurs, la nature de la tâche de classification à plusieurs classes dans l'expérience 13 peut demander un processus d'optimisation plus complexe.

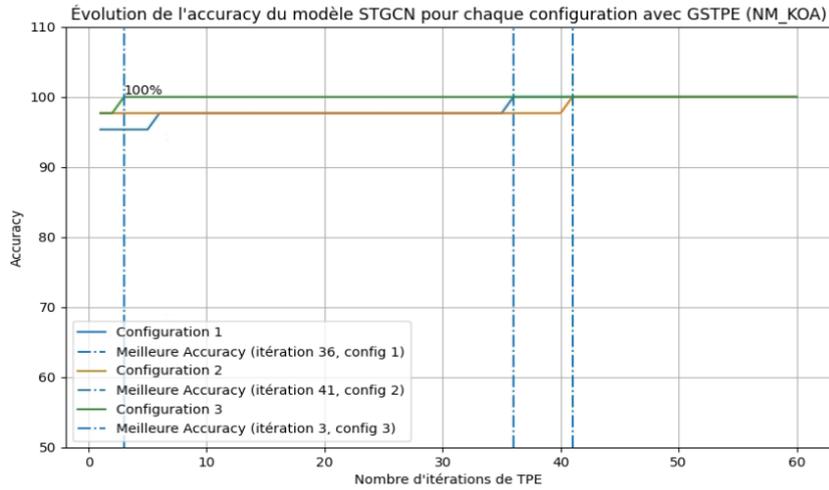


Figure 4. 27 : Évolution de la performance du modèle STGCN en fonction du nombre d'itérations TPE pour chaque configuration explorée (Expérience-7)

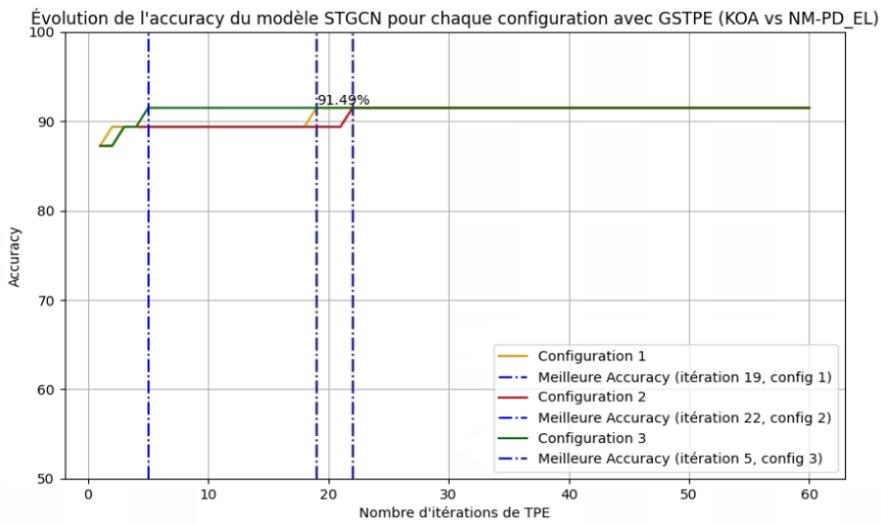


Figure 4. 28: Évolution de la performance du modèle STGCN en fonction du nombre d'itérations TPE pour chaque configuration explorée (Expérience-9)

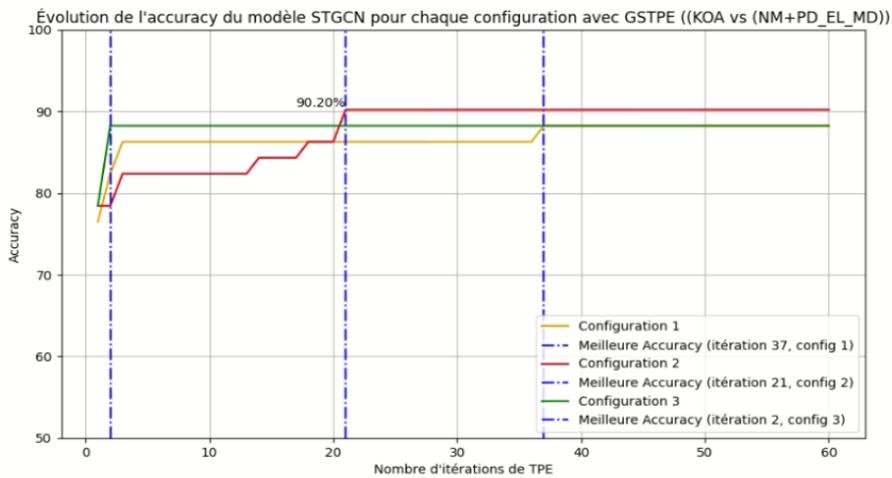


Figure 4. 29: Évolution de la performance du modèle STGCN en fonction du nombre d'itérations TPE pour chaque configuration explorée (Expérience-11)

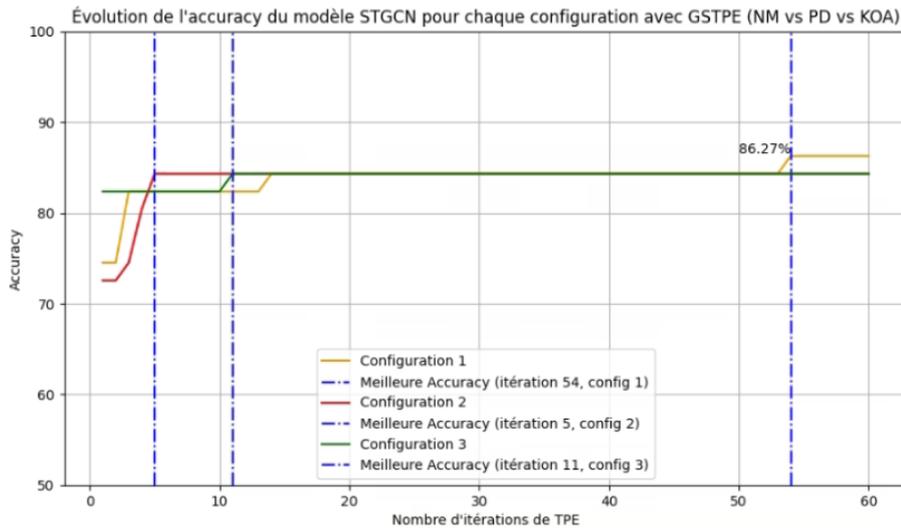


Figure 4. 30: Évolution de la performance du modèle STGCN en fonction du nombre d'itérations TPE pour chaque configuration explorée (Expérience-13)

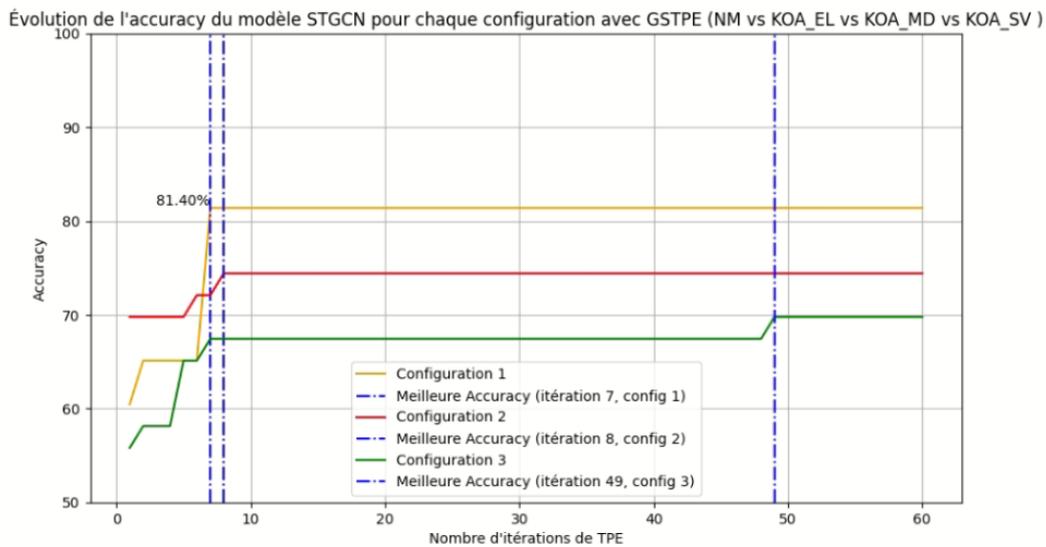


Figure 4. 31: Évolution de la performance du modèle STGCN en fonction du nombre d'itérations TPE pour chaque configuration explorée (Expérience-15)

L'examen des matrices de confusion avant et après l'optimisation montre que l'optimisation des HPs a conduit à une amélioration significative des performances. Plus spécifiquement, nous pouvons observer des améliorations notables dans la capacité du modèle à classer correctement certaines classes. Par exemple, dans la matrice de l'expérience 7 (Figure 4. 32 (a)), nous remarquons que le modèle a réussi à classer correctement les sujets sains (NM) par rapport aux résultats obtenus sans optimisation (Figure 4. 22).

De manière similaire, dans l'expérience 11 (Figure 4. 32 (c)), nous constatons que le modèle STGCN, après optimisation des HPs avec GS-TPE, a réussi à classer les patients atteints de KOA

ainsi que les sujets sains (NM) même en présence de patients PD perturbateurs, par rapport aux résultats obtenus sans optimisation (Figure 4. 24).

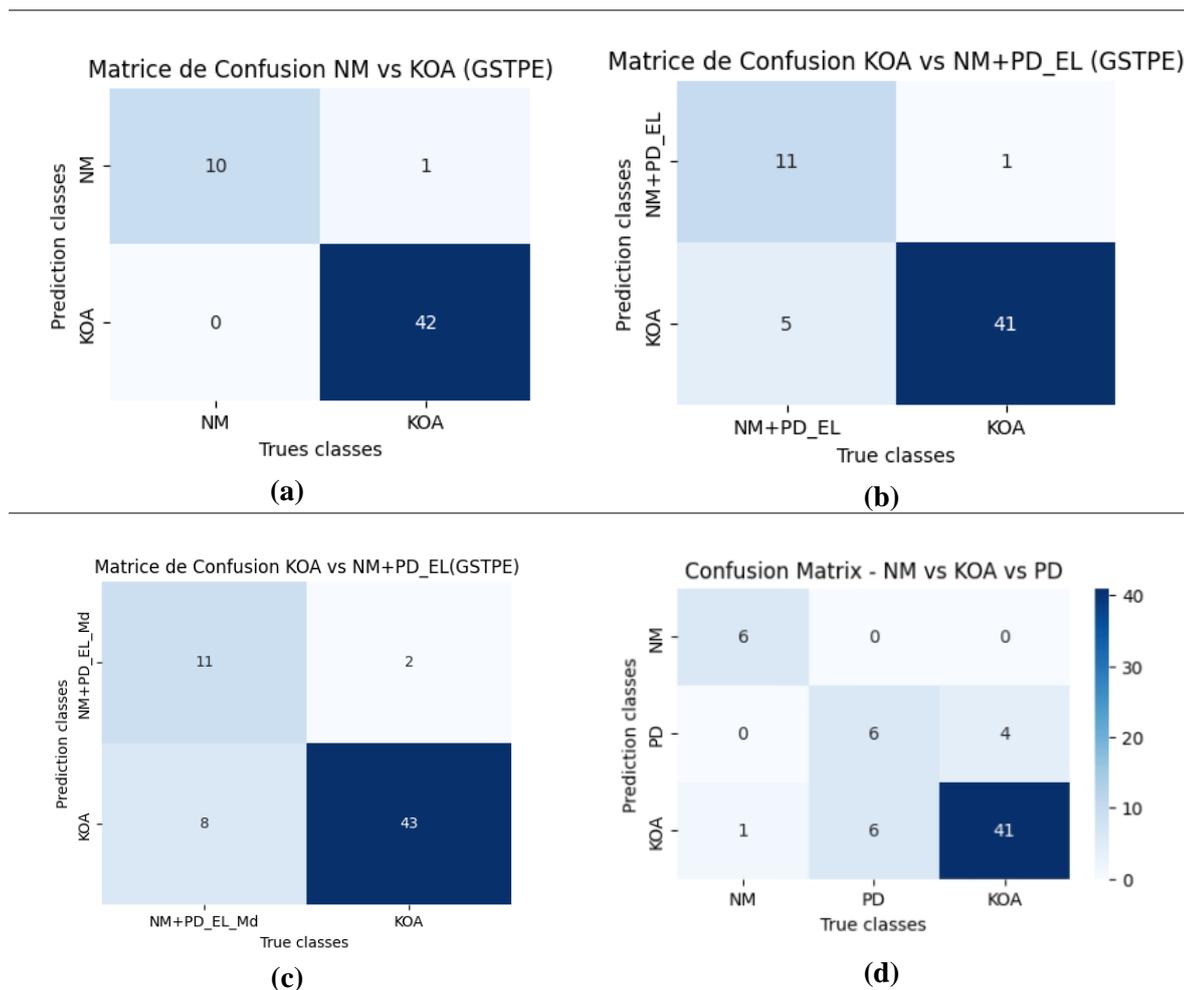


Figure 4. 32 : Matrices de confusion des différentes expériences après l'optimisation des HPs avec GS-TPE : (a) Expérience 7, (b) Expérience 9, (c) Expérience 11, et (d) Expérience 13

Matrice de Confusion NM vs KOA_EL vs KOA_MD vs KOA_SV (GSTPE)

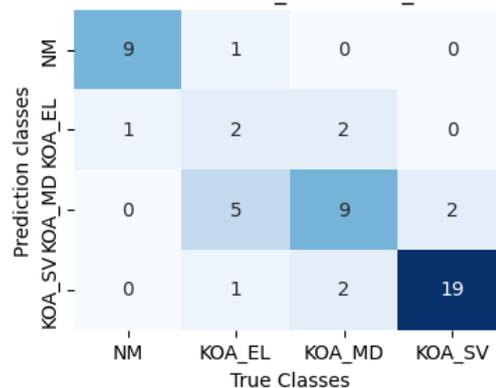


Figure 4. 33 : Matrice de confusion après l'optimisation des hyperparamètres avec GS-TPE pour l'expérience 15

7.3.4.2 Synthèse et détails relatifs aux meilleures configurations d'HPs pour l'architecture STGCN

Les résultats présentés dans les tableaux ci-dessous reflètent les configurations optimales des HP obtenus par GS-TPE pour le réseau STGCN dans le cadre de la classification des patients atteints d'arthrose du genou ainsi que la maladie de Parkinson relatives aux expériences 7, 9, 11,13 et 15.

Nous remarquons que deux HPs en particulier ont une influence significative sur les performances du modèle : le batch size et le Learning rate. Ces deux HPs ont été sélectionnés pour une exploration plus approfondie avec l'algorithme TPE après une première phase d'optimisation avec GS qui a pris en compte quatre HPs au total. Cette décision reflète la reconnaissance de l'importance de ces deux HPs dans la configuration du modèle, ainsi que la volonté d'obtenir des performances optimales en ajustant finement ces HP. En examinant l'expérience 7 (Tableau 4. 16), nous observons que lors de l'utilisation de la première partie de GS-TPE, qui est l'algorithme GS, les trois meilleures configurations sélectionnées partagent une valeur de batch size commune, mais diffèrent en ce qui concerne le Learning rate et la taille du noyau temporel. Après une exploration supplémentaire avec TPE, le batch size est ajustée à 5.

Nous remarquons des variations significatives dans le ratio entre les nombres de canaux dans chaque couche STGCN pour les expériences 9, 11 et 13 (voir Tableaux ci-dessous). Les configurations optimales trouvées par TPE varient entre 4 et 7, soulignant l'importance de la diversité de canaux pour capturer les caractéristiques spatio-temporelles complexes liées à l'arthrose du genou et la maladie de Parkinson.

En conclusion, l'optimisation des HPs par l'approche GS-TPE a permis d'améliorer les performances globales du modèle STGCN. Les ajustements apportés ont joué un rôle important dans la précision et la généralisation du modèle pour la classification des différentes pathologies étudiées.

Tableau 4. 16: Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'expérience 7

<i>HPs/ Configurations GS-TPE</i>	<i>Meilleures configurations identifiées par GS</i>		
	<i>Config-1</i>	<i>Config-2</i>	<i>Config-3</i>
Ratio	1	3	2
Temporal Kernel Size	9	11	11
Batch_size	4	4	4
Learning_rate	0.001	0.01	0.01
	<i>Nouvelles configurations optimales issues de TPE</i>		
	<i>Config-1'</i>	<i>Config-2'</i>	<i>Config-3'</i>
Batch_size	5	5	3
Learning_rate	0.010	0.012	0.03

Tableau 4. 17: Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'expérience 9

<i>HPs/ Configurations GS-TPE</i>	<i>Meilleures configurations identifiées par GS</i>		
	<i>Config-1</i>	<i>Config-2</i>	<i>Config-3</i>
Ratio	1	2	2
Temporal Kernel Size	11	11	9
Batch_size	4	8	8
Learning_rate	0.001	0.001	0.001
	<i>Nouvelles configurations optimales issues de TPE</i>		
	<i>Config-1'</i>	<i>Config-2'</i>	<i>Config-3'</i>
Batch_size	5	7	8
Learning_rate	0.0009	0.0009	0.0009

Tableau 4. 18: Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'expérience 11

<i>HPs/ Configurations GS-TPE</i>	<i>Meilleures configurations identifiées par GS</i>		
	<i>Config-1</i>	<i>Config-2</i>	<i>Config-3</i>
Ratio	3	1	2
Temporal Kernel Size	9	9	7
Batch_size	8	4	8
Learning_rate	0.0001	0.001	0.0001
	<i>Nouvelles configurations optimales issues de TPE</i>		
	<i>Config-1'</i>	<i>Config-2'</i>	<i>Config-3'</i>
Batch_size	8	4	10
Learning_rate	0.0001	0.0009	0.0001

Tableau 4. 19: Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'expérience 13

<i>HPs/ Configurations GS-TPE</i>	<i>Meilleures configurations identifiées par GS</i>		
	<i>Config-1</i>	<i>Config-2</i>	<i>Config-3</i>
Ratio	3	3	1
Temporal Kernel Size	11	7	11
Batch_size	8	8	8
Learning_rate	0.0001	0.001	0.01
	<i>Nouvelles configurations optimales issues de TPE</i>		
	<i>Config-1'</i>	<i>Config-2'</i>	<i>Config-3'</i>
Batch_size	7	8	6
Learning_rate	9.27e-05	0.0009	0.001

Tableau 4. 20: Les meilleures configurations d'HPs obtenues avec l'approche GS-TPE pour l'expérience 15

<i>HPs/ Configurations GS-TPE</i>	<i>Meilleures configurations identifiées par GS</i>		
	<i>Config-1</i>	<i>Config-2</i>	<i>Config-3</i>
Ratio	1	3	3
Temporal Kernel Size	11	11	9
Batch_size	4	8	8

Learning_rate	0.001	0.001	0.0001
	<i>Nouvelles configurations optimales issues de TPE</i>		
	<i>Config-1'</i>	<i>Config-2'</i>	<i>Config-3'</i>
Batch_size	5	9	7
Learning_rate	0.01	0.0009	9.32e-05

7.3.5 Comparaison avec une méthode existante de détection de la KOA

Le tableau ci-dessous présente une comparaison entre notre approche et une méthode de référence existante pour la classification de l'arthrose du genou (NM-KOA) [128]. C'est le seul article à utiliser cette base, bien qu'elle n'ait été employée que par cet article.

Dans un premier temps, nous avons utilisé l'architecture STGCN sans optimisation des HPs, en considérant 33 points d'articulation, ce qui nous a permis d'obtenir une précision de 100%. Ensuite, nous avons réduit le nombre de points d'articulation à 8 (au niveau du membre inférieur), ce qui nous a permis d'obtenir une précision de 96.22%. Nous avons ensuite intégré l'approche d'optimisation GS-TPE à STGCN, ce qui a permis d'améliorer la précision à 98.11%.

À titre de comparaison, la méthode existante utilisant l'algorithme KNN avec la même base de données NM-PD-KOA, mais sans optimisation des hyperparamètres et en utilisant 6 points d'articulation, a abouti à une précision de 95.16%. Notons que la méthode utilisée dans la littérature s'appuie sur des marqueurs et une segmentation de la région d'intérêt, alors que nous avons utilisé une technique sans marqueur basée sur BlazePose. La seule différence réside dans cette comparaison et dans notre travail où nous avons modifié les données de la base en effectuant un découpage des vidéos en segments de 4 secondes.

Ces résultats montrent que notre modèle STGCN, sans et avec optimisation, surpasse significativement la méthode existante en termes d'accuracy.

Tableau 4. 21: Bilan comparatif entre l'approche proposé et les méthodes existantes

Auteurs	Approche	Base de données	Type de classification	Nombre de points clés	Optimisation des HP	Accuracy (%)
Ours	GSTPE-STGCN	NM-PD-KOA	NM-KOA	8	Oui	98.11
	STGCN	NM-PD-KOA	NM-KOA	8	Non	96.22

	STGCN	NM- PD- KOA	NM-KOA	33	Non	100
Kour et al [128]	KNN	NM- PD- KOA	NM-KOA	6	Non	95.16

8 Conclusion

Dans ce chapitre, nous nous sommes fixés comme objectif de valider notre approche d'optimisation GS-TPE pour l'amélioration de la performance des DNN dans le domaine de l'analyse du geste et des postures pour la classification de différentes pathologies. Les entrées dans ce cas sont des vidéos d'analyse de la marche. Comme pathologie cible, nous avons opté pour l'arthrose du genou (KOA). Nous avons en outre utilisé des vidéos de patients atteints par la maladie de Parkinson comme élément perturbateur pour vérifier la robustesse de nos architectures optimisées à différencier les deux pathologies observées par des acquisitions vidéo de même type. Nous avons choisi une base de données publique appelée NM-PD-KOA "Gait Dataset for Knee Osteoarthritis and Parkinson's Disease Analysis With Severity Levels" (ensemble de données vidéos de la marche pour l'exploration de l'arthrose du genou et de la maladie de Parkinson). Le principal objectif de cette base est d'examiner les déviations observées dans la marche des patients par rapport à celle d'individus ayant une marche normale. L'intérêt de cette base est qu'elle contient deux pathologies, la principale étant l'arthrose de genou et la secondaire étant utilisée comme perturbateur : la maladie de Parkinson.

Nous avons procédé à une analyse des recherches existantes sur la classification de ces pathologies, ainsi que sur les techniques d'estimation de la pose humaine existantes sans marqueurs. Nous avons implémenté un modèle basé sur les graphes, en utilisant BlazePose, la technique de détection sans marqueur la plus pertinente vis-à-vis de nos objectifs. Nous avons proposé une harmonisation des données basée sur un découpage temporelle plus fin et uniforme à 200 frames. Cette uniformisation temporelle a permis une augmentation des données du fait que nous passons de 186 vidéos à durées variables à 317 vidéos de 200 frames. Nous avons montré que l'analyse globale à 33 points satisfait nos objectifs et qu'une analyse locale à 8 points clés donne plus de performances pour la classification à 3 sorties. Cette comparaison entre analyse globale et locale mérite d'être mieux approfondie en enrichissant notre base et en liant le choix des points clés à l'anatomie fonctionnelle. Nous avons aussi montré l'apport de notre démarche d'optimisation GS-TPE dans l'amélioration des performances. Nous avons conclu enfin, que même en présence d'un élément perturbateur, à savoir une autre pathologie (Parkinson), les résultats restent meilleurs par rapport à l'approche rapportée dans la littérature.

L'ensemble de ces résultats confirment la pertinence de nos choix et ouvrent des perspectives intéressantes sur les plans fondamental et clinique. Une mise en œuvre clinique est entamée sur des sujets suivis par notre service partenaire de rhumatologie.

Conclusion générale et perspectives

Cette exploration de plusieurs domaines de l'apprentissage automatique pour l'analyse de signaux physiologiques temporels et la quantification de pathologies humaines nous a permis de comprendre les défis et les avantages liés à l'optimisation des réseaux de neurones profonds dans divers contextes d'application. Nos contributions dans ce cadre consistent en des propositions d'architectures DL qui répondent aux objectifs d'applications spécifiques caractérisées par une variation temporelle de données physiologiques et en une optimisation des HPs de modèles retenus.

Le premier chapitre pose le cadre général en étudiant les concepts clés de l'apprentissage profond, en soulignant l'importance fondamentale des HPs dans la configuration et la performance des modèles. Puis il présente les différents algorithmes d'optimisation des HPs, les bibliothèques existantes, et brosse un état de l'art des techniques utilisées dans des domaines connexes au nôtre. Nous avons conclu ce chapitre en concentrant notre attention sur les méthodes systématiques GS/RS et sur la méthode bayésienne TPE afin d'optimiser les HPs, un choix stratégique retenu tout au long de ce travail.

Le deuxième chapitre est consacré au premier domaine de validation de nos approches, soit la quantification et la détection de la vigilance fondée sur l'électroencéphalographie (EEG). Ce signal temporel renseigne sur l'activité électrique cérébrale. Les techniques de prétraitement et d'extraction des caractéristiques de ce signal ont été examinées, établissant le contexte du traitement des signaux EEG et leur classification par des réseaux de neurones profonds.

Le troisième chapitre détaille nos différents modèles DL implémentés pour la classification des états de vigilance. Nous avons présenté notre approche hybride combinant hiérarchiquement méthode systématique et TPE pour l'optimisation des modèles DNN retenus et nous l'avons positionnée par rapport aux techniques de référence. Nous avons décrit la base de données de validation et rapporté nos résultats d'optimisation par la méthode hybride. Nous avons mis en évidence les améliorations des performances liées à l'utilisation de notre démarche.

Le quatrième chapitre étend le domaine d'application de notre approche d'optimisation à l'analyse de la marche pour la classification de diverses pathologies par analyse des enregistrements vidéo. Le point commun des deux applications est l'évolution temporelle de données physiologiques, l'EEG dans la première et les séquences de posture (graphes de squelette) extraites des vidéos dans la deuxième.

Nous avons procédé à une analyse profonde des recherches existantes sur la classification de ces pathologies, ainsi que sur les techniques d'estimation de la pose humaine existantes par vidéo sans

marqueurs. Nous avons retenu une approche basée sur le suivi des graphes formés par des points clés extraits par BlazePose, la technique de détection sans marqueur la plus pertinente vis-à-vis de nos objectifs. Nous avons adopté un DNN appelé STGCN pour l'analyse des gestes avec deux configurations, l'une globale à 33 points et l'autre locale à 8 points clés autour de la région explorée. Nous avons appliqué notre approche GS-TPE pour l'optimisation du modèle implémenté dans différentes configurations. Nous avons utilisé une base de données publique appelée NM-PD-KOA "Gait Dataset for Knee Osteoarthritis and Parkinson's Disease Analysis With Severity Levels" formée par un ensemble de données vidéos de la marche pour l'exploration de l'arthrose du genou (KOA) et de la maladie de Parkinson (PD), et comportant des sujets sains (NM). 20 expériences ont été réalisées. 5 configurations à 33 points et 5 configurations à 8 points pour une analyse locale ont été expérimentées. Parmi ces expériences, des classifications binaires avec et sans perturbateur et des classifications à trois et quatre sorties ont été testées. Les résultats des 5 configurations à 8 points pour l'analyse locale ont été rapportés et étudiés, avant et après optimisation des HP.

Nous avons montré que dans toutes les expériences l'optimisation a permis une amélioration significative des performances et nous avons pu identifier les valeurs optimales des HPs relatives à chaque configuration.

Nous avons montré l'aptitude de nos modèles à différencier entre les classes spécifiées, avec des performances acceptables et qui varient selon le nombre de sorties et le type de l'analyse globale ou locale. L'analyse globale a permis la meilleure différenciation entre NM et KOA (2 classes) et la différenciation entre sains et les différents stades de la KOA (4 classes). Ce résultat est d'autant plus intéressant qu'il permet la détection précoce de l'arthrose du genou. L'analyse locale était performante dans la classification des NM vs KOA vs PD, en particulier pour bien différencier les KOA. Les résultats de cette exploration locale étaient néanmoins loin de nos attentes et la comparaison entre analyse globale et locale mérite d'être mieux approfondie en enrichissant notre base et en liant le choix des points clés à l'anatomie fonctionnelle. Nous avons conclu enfin, que même en présence d'un élément perturbateur, à savoir une autre pathologie (Parkinson), les résultats restent meilleurs par rapport à l'approche existante dans la littérature après optimisation.

L'ensemble de ces résultats confirment la pertinence de nos choix et ouvrent des perspectives intéressantes sur les plans fondamental et clinique.

Cette thèse a apporté une contribution significative à la compréhension et à l'amélioration des techniques d'optimisation des réseaux de neurones profonds dans les domaines d'analyse et classification de signaux bio-médicaux temporels. Compte tenu de la complexité de l'exploration d'espaces d'HPs de grandes dimensions, nous nous sommes limités dans le cadre de cette thèse à une optimisation mono-critère, i.e. avec une seule fonction objectif liée à une métrique de qualité

de la classification, sans prendre en compte explicitement la complexité (empreinte mémoire, temps d'inférence) des modèles générés. Il faut néanmoins préciser que, même mono-critère, cette approche se révèle en pratique vertueuse *aussi* sur le plan de la complexité, dans la mesure où l'optimisation des HP permet en général d'améliorer la performance d'un modèle léger, tandis qu'une approche classique cherchera à augmenter la complexité du modèle (plus de couches, plus de canaux par couches, etc.).

Pour autant, l'un des enjeux majeurs des modèles DL pour la médecine concerne leur déploiement dans des systèmes mobiles, embarqués, et faible coût, afin de faciliter la diffusion des applications concernées dans des zones distantes et/ou faiblement médicalisées. Dans cet objectif, il sera nécessaire d'intégrer de façon plus explicite les critères d'efficacité computationnelle à l'optimisation des HPs. Cet aspect constitue donc une des premières perspectives de notre travail de thèse.

Sur le plan applicatif et médical, les résultats obtenus dans le cadre de la classification des états de vigilance et de l'analyse de la marche ouvrent la voie à des applications prometteuses dans le domaine médical et au-delà. Les approches proposées dans l'analyse du signal EEG peuvent être étendues vers des modèles end-to-end afin de traiter les signaux temporels bruts, ainsi qu'à d'autres signaux physiologiques comme l'électrocardiogramme (ECG), pour caractériser d'autres pathologies. Un autre signal intéressant pour nous mais avec une autre granularité temporelle est l'électromyogramme (EMG). Ce signal peut être couplé à l'analyse vidéo de la marche pour optimiser nos classifications et prédictions. Dans ce volet, une mise en œuvre clinique est entamée sur des sujets suivis par notre service partenaire de rhumatologie. A ce jour, nous avons exploré six sujets, quatre sains et deux à arthrose de genou modérée. Grâce aux accès dont nous bénéficions dans notre laboratoire aux dossiers et aux patients, nous pourrions vérifier les décisions litigieuses et ainsi améliorer le fonctionnement de nos modèles. Enfin notre concept d'analyse locale pour les pathologies rhumatologiques reste d'actualité et nous envisageons de la revoir afin de mieux choisir les points clés ou d'injecter d'autres points pertinents.

Références bibliographiques

- [1] B. Ait Skourt, A. El Hassani, et A. Majda, « Lung CT Image Segmentation Using Deep Neural Networks », *Procedia Comput. Sci.*, vol. 127, p. 109-113, janv. 2018, doi: 10.1016/j.procs.2018.01.104.
- [2] A. Niedzwiecki, M. W. Roomi, T. Kalinovsky, et M. Rath, « Anticancer Efficacy of Polyphenols and Their Combinations », *Nutrients*, vol. 8, n° 9, p. 552, sept. 2016, doi: 10.3390/nu8090552.
- [3] D. Cardon, J.-P. Cointet, et A. Mazières, « La revanche des neurones. L'invention des machines inductives et la controverse de l'intelligence artificielle », *Réseaux*, vol. 211, n° 5, p. 173-220, 2018, doi: 10.3917/res.211.0173.
- [4] M. Ettaouil, M. LAZAAR, K. Elmoutaouakil, et K. Haddouch, « A new algorithm for optimization of the kohonen network architectures using the continuous hopfield networks », *WSEAS Trans. Comput.*, vol. 12, p. 155-163, avr. 2013.
- [5] A. B. Gumelar, E. M. Yuniarno, D. P. Adi, A. G. Sooai, I. Sugiarto, et M. H. Purnomo, « BiLSTM-CNN Hyperparameter Optimization for Speech Emotion and Stress Recognition », in *2021 International Electronics Symposium (IES)*, sept. 2021, p. 156-161. doi: 10.1109/IES53407.2021.9594024.
- [6] V. Bibaeva, « Using Metaheuristics for Hyper-Parameter Optimization of Convolutional Neural Networks », in *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, sept. 2018, p. 1-6. doi: 10.1109/MLSP.2018.8516989.
- [7] J. Bergstra et Y. Bengio, « Random Search for Hyper-Parameter Optimization ».
- [8] S. M. LaValle et M. S. Branicky, « On the Relationship between Classical Grid Search and Probabilistic Roadmaps », in *Algorithmic Foundations of Robotics V*, J.-D. Boissonnat, J. Burdick, K. Goldberg, et S. Hutchinson, Éd., in Springer Tracts in Advanced Robotics. , Berlin, Heidelberg: Springer, 2004, p. 59-75. doi: 10.1007/978-3-540-45058-0_5.
- [9] « Behera et Nain - 2022 - GSO-CRS grid search optimization for collaborativ.pdf ».
- [10] N. Quadrianto, K. Kersting, et Z. Xu, « Gaussian Process », in *Encyclopedia of Machine Learning*, C. Sammut et G. I. Webb, Éd., Boston, MA: Springer US, 2010, p. 428-439. doi: 10.1007/978-0-387-30164-8_324.
- [11] K. Wood, A. M. Dunton, A. Muyskens, et B. W. Priest, « Scalable Gaussian Process Hyperparameter Optimization via Coverage Regularization », 22 septembre 2022, *arXiv*: arXiv:2209.11280. doi: 10.48550/arXiv.2209.11280.
- [12] S. Manzhos et M. Ihara, « On the optimization of hyperparameters in Gaussian process regression with the help of low-order high-dimensional model representation », *J. Math. Chem.*, vol. 61, n° 1, p. 7-20, janv. 2023, doi: 10.1007/s10910-022-01407-x.
- [13] « Ozaki et al. - 2022 - Multiobjective Tree-Structured Parzen Estimator.pdf ».
- [14] J. Bergstra, D. Yamins, et D. Cox, « Hyperopt: A Python Library for Optimizing the Hyperparameters of Machine Learning Algorithms », janv. 2013, p. 13-19. doi: 10.25080/Majora-8b375195-003.
- [15] « Orion: A framework for distributed hyperparameter optimisation », Xavier Bouthillier. Consulté le: 22 août 2023. [En ligne]. Disponible sur: <https://bouthilx.github.io/projects/1-orion/>
- [16] J. Kennedy et R. Eberhart, « Particle swarm optimization », in *Proceedings of ICNN'95 - International Conference on Neural Networks*, nov. 1995, p. 1942-1948 vol.4. doi: 10.1109/ICNN.1995.488968.
- [17] G. Pereira, « Particle Swarm Optimization », mai 2011.
- [18] Z. Gao, Y. Li, Y. Yang, X. Wang, N. Dong, et H.-D. Chiang, « A GPSO-optimized convolutional neural networks for EEG-based emotion recognition », *Neurocomputing*, vol. 380, p. 225-235, mars 2020, doi: 10.1016/j.neucom.2019.10.096.
- [19] M. Khalifa, M. Ammar, W. Ouarda, et A. Alimi, « Particle swarm optimization for deep learning of convolution neural network », nov. 2017, p. 1-5. doi: 10.1109/SCCSIT.2017.8293059.
- [20] T. Alam, S. Qamar, A. Dixit, et M. Benaida, *Genetic Algorithm: Reviews, Implementations, and Applications*. 2020. doi: 10.36227/tehrxiv.12657173.
- [21] F. Pedregosa *et al.*, « Scikit-learn: Machine Learning in Python », 5 juin 2018, *arXiv*: arXiv:1201.0490. doi: 10.48550/arXiv.1201.0490.
- [22] B. Komer, J. Bergstra, et C. Eliasmith, « Hyperopt-Sklearn: Automatic Hyperparameter Configuration for Scikit-Learn », présenté à Python in Science Conference, Austin, Texas, 2014, p. 32-37. doi: 10.25080/Majora-14bd3278-006.

- [23] T. Akiba, S. Sano, T. Yanase, T. Ohta, et M. Koyama, « Optuna: A Next-generation Hyperparameter Optimization Framework », 25 juillet 2019, *arXiv*: arXiv:1907.10902. doi: 10.48550/arXiv.1907.10902.
- [24] L. Hertel, J. Collado, P. Sadowski, J. Ott, et P. Baldi, « Sherpa: Robust Hyperparameter Optimization for Machine Learning », 8 mai 2020, *arXiv*: arXiv:2005.04048. doi: 10.48550/arXiv.2005.04048.
- [25] P. Bennet, C. Doerr, A. Moreau, J. Rapin, F. Teytaud, et O. Teytaud, « Nevergrad: black-box optimization platform », *ACM SIGEVOlution*, vol. 14, n° 1, p. 8-15, avr. 2021, doi: 10.1145/3460310.3460312.
- [26] V. Gandarillas et J. T. Hwang, « TALOS: A toolbox for spacecraft conceptual design », 27 mars 2023, *arXiv*: arXiv:2303.14936. doi: 10.48550/arXiv.2303.14936.
- [27] F. F. Firdaus, H. A. Nugroho, et I. Soesanti, « Deep Neural Network with Hyperparameter Tuning for Detection of Heart Disease », *2021 IEEE Asia Pac. Conf. Wirel. Mob. APWiMob*, p. 59-65, avr. 2021, doi: 10.1109/APWiMob51111.2021.9435250.
- [28] A. Cheraghali, M. Hajiaghayi-Keshteli, et M. M. Paydar, « Tree Growth Algorithm (TGA): A novel approach for solving optimization problems », *Eng. Appl. Artif. Intell.*, vol. 72, mai 2018, doi: 10.1016/j.engappai.2018.04.021.
- [29] A. Baldominos, Y. Saez, et P. Isasi, « Evolutionary convolutional neural networks: An application to handwriting recognition », *Neurocomputing*, vol. 283, p. 38-52, mars 2018, doi: 10.1016/j.neucom.2017.12.049.
- [30] L. Thames, R. Abler, et D. Schaefer, « Parametric optimization of artificial neural networks for signal approximation applications », mars 2011, p. 209-215. doi: 10.1145/2016039.2016095.
- [31] B. Nakisa, N. Rastgoo, A. Rakotonirainy, F. Maire, et V. Chandran, « Long Short Term Memory Hyperparameter Optimization for a Neural Network Based Emotion Recognition Framework », *IEEE Access*, vol. PP, p. 1-1, sept. 2018, doi: 10.1109/ACCESS.2018.2868361.
- [32] A. A. Heidari, H. Faris, S. Mirjalili, et I. Aljarah, « An Efficient Hybrid Multilayer Perceptron Neural Network with Grasshopper Optimization », *Soft Comput.*, vol. 23 aliasgharheidari.com, p. 7941-7958, sept. 2019, doi: 10.1007/s00500-018-3424-2.
- [33] M. Awadalla et A. Galal, « Deep Neural Network Hyper-Parameters Optimization for Face Classification », in *2021 International Symposium on Electrical, Electronics and Information Engineering*, in ISEEIE 2021. New York, NY, USA: Association for Computing Machinery, juill. 2021, p. 224-229. doi: 10.1145/3459104.3459143.
- [34] L. Wen, X. Ye, et L. Gao, « A new automatic machine learning based hyperparameter optimization for workpiece quality prediction », *Meas. Control*, vol. 53, p. 002029402093234, juill. 2020, doi: 10.1177/0020294020932347.
- [35] S. Young, D. Rose, T. Karnowski, S.-H. Lim, et R. Patton, « Optimizing deep learning hyper-parameters through an evolutionary algorithm », nov. 2015, p. 1-5. doi: 10.1145/2834892.2834896.
- [36] S. Mohammad. R. Hashemi, E. Kozegar, M. M. Deramgozin, et B. Minaei-Bidgoli, « Training Feed-forward Neural Networks using Asexual Reproduction Optimization (ARO) Algorithm », in *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, févr. 2019, p. 809-812. doi: 10.1109/KBEI.2019.8735035.
- [37] M. Zhang, H. Li, S. Pan, J. Lyu, S. Ling, et S. Su, « Convolutional Neural Networks-Based Lung Nodule Classification: A Surrogate-Assisted Evolutionary Algorithm for Hyperparameter Optimization », *IEEE Trans. Evol. Comput.*, vol. 25, n° 5, p. 869-882, oct. 2021, doi: 10.1109/TEVC.2021.3060833.
- [38] S.-H. Kim, Z. W. Geem, et G.-T. Han, « Hyperparameter Optimization Method Based on Harmony Search Algorithm to Improve Performance of 1D CNN Human Respiration Pattern Recognition System », *Sensors*, vol. 20, n° 13, p. 3697, juill. 2020, doi: 10.3390/s20133697.
- [39] Y. Yoo, « Hyperparameter optimization of deep neural network using univariate dynamic encoding algorithm for searches », *Knowl.-Based Syst.*, vol. 178, p. 74-83, août 2019, doi: 10.1016/j.knosys.2019.04.019.
- [40] I. Tahir, U. Qamar, H. Abbas, B. Zeb, et S. Abid, « Classification of EEG Signal by Training Neural Network with Swarm Optimization for Identification of Epilepsy », in *Proceedings of the 2019 11th International Conference on Machine Learning and Computing*, in ICMLC '19. New York, NY, USA: Association for Computing Machinery, févr. 2019, p. 197-203. doi: 10.1145/3318299.3318374.
- [41] Y. Chhabra, S. Varshney, et Ankita, « Hybrid particle swarm training for convolution neural network (CNN) », août 2017, p. 1-3. doi: 10.1109/IC3.2017.8284356.

- [42] N. van Stein, H. Wang, et T. Bäck, « Automatic Configuration of Deep Neural Networks with Parallel Efficient Global Optimization », juill. 2019, p. 1-7. doi: 10.1109/IJCNN.2019.8851720.
- [43] T. Yamasaki, T. Honma, et K. Aizawa, « Efficient Optimization of Convolutional Neural Networks Using Particle Swarm Optimization », avr. 2017, p. 70-73. doi: 10.1109/BigMM.2017.69.
- [44] H. Rajaguru et S. Prabhakar, « Bayesian linear discriminant analysis with hybrid ABC-PSO classifier for classifying epilepsy from EEG signals », juill. 2017, p. 977-980. doi: 10.1109/ICCMC.2017.8282613.
- [45] A. Darwish, D. Ezzat, et A. E. Hassanien, « An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis », *Swarm Evol. Comput.*, vol. 52, p. 100616, nov. 2019, doi: 10.1016/j.swevo.2019.100616.
- [46] P. Ribalta Lorenzo, J. Nalepa, L. Sanchez, et J. Ranilla, « Hyper-parameter selection in deep neural networks using parallel particle swarm optimization », juill. 2017, p. 1864-1871. doi: 10.1145/3067695.3084211.
- [47] N. Bacanin, T. Bezdán, E. Tuba, I. Strumberger, et M. Tuba, « Optimizing Convolutional Neural Network Hyperparameters by Enhanced Swarm Intelligence Metaheuristics », *Algorithms*, vol. 13, p. 67, mars 2020, doi: 10.3390/a13030067.
- [48] V. Bibaeva, « Using Metaheuristics for Hyper-Parameter Optimization of Convolutional Neural Networks », in *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, Aalborg: IEEE, sept. 2018, p. 1-6. doi: 10.1109/MLSP.2018.8516989.
- [49] I. Strumberger, E. Tuba, N. Bacanin, M. Zivkovic, M. Beko, et M. Tuba, « Designing Convolutional Neural Network Architecture by the Firefly Algorithm », mai 2019, p. 59-65. doi: 10.1109/YEF-ECE.2019.8740818.
- [50] Y. Wang, H. Zhang, et G. Zhang, « cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyperparameters of convolutional neural networks », *Swarm Evol. Comput.*, vol. 49, p. 114-123, sept. 2019, doi: 10.1016/j.swevo.2019.06.002.
- [51] R. Brito, S. Fong, Y. Zhuang, et Y. Wu, « Generating Neural Networks with Optimal Features through Particle Swarm Optimization », in *Proceedings of the International Conference on Big Data and Internet of Thing*, in BDIOT2017. New York, NY, USA: Association for Computing Machinery, déc. 2017, p. 96-101. doi: 10.1145/3175684.3175701.
- [52] I. Strumberger, E. Tuba, N. Bacanin, R. Jovanovic, et M. Tuba, « Convolutional Neural Network Architecture Design by the Tree Growth Algorithm Framework », in *2019 International Joint Conference on Neural Networks (IJCNN)*, juill. 2019, p. 1-8. doi: 10.1109/IJCNN.2019.8851755.
- [53] M. Injadat, F. Salo, A. B. Nassif, A. Essex, et A. Shami, « Bayesian Optimization with Machine Learning Algorithms Towards Anomaly Detection », in *2018 IEEE Global Communications Conference (GLOBECOM)*, déc. 2018, p. 1-6. doi: 10.1109/GLOCOM.2018.8647714.
- [54] L. L. Lima, J. R. Ferreira, et M. C. Oliveira, « Efficient Hyperparameter Optimization of Convolutional Neural Networks on Classification of Early Pulmonary Nodules », in *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*, juin 2019, p. 144-149. doi: 10.1109/CBMS.2019.00039.
- [55] M. Lindauer *et al.*, « SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization », 8 février 2022, *arXiv*: arXiv:2109.09831. doi: 10.48550/arXiv.2109.09831.
- [56] M. Fakhfakh, B. Bouaziz, F. Gargouri, et L. Chaari, « Bayesian optimization for artificial neural networks: application to Covid-19 image classification », in *2022 International Conference on Technology Innovations for Healthcare (ICTIH)*, sept. 2022, p. 34-37. doi: 10.1109/ICTIH57289.2022.10112015.
- [57] S. Han, H. Eom, J. Kim, et C. Park, « Optimal DNN architecture search using Bayesian Optimization Hyperband for arrhythmia detection », in *2020 IEEE Wireless Power Transfer Conference (WPTC)*, nov. 2020, p. 357-360. doi: 10.1109/WPTC48563.2020.9295590.
- [58] Z. Li, Y. Wang, et J. Ma, « Fault Diagnosis of Motor Bearings Based on a Convolutional Long Short-Term Memory Network of Bayesian Optimization », *IEEE Access*, vol. 9, p. 97546-97556, 2021, doi: 10.1109/ACCESS.2021.3093363.
- [59] R. A. Lateef et A. R. Abbas, « Hyper parameter Optimization of Deep Learning Models for Fall Recognition using Tree-structured Parzen Estimator », in *2022 Fifth College of Science International Conference of Recent Trends in Information Technology (CSCTIT)*, nov. 2022, p. 66-71. doi: 10.1109/CSCTIT56299.2022.10145716.

- [60] T. Xiao, D. Ren, S. Lei, J. Zhang, et X. Liu, « Based on grid-search and PSO parameter optimization for Support Vector Machine », *Proc. World Congr. Intell. Control Autom. WCICA*, vol. 2015, p. 1529-1533, mars 2015, doi: 10.1109/WCICA.2014.7052946.
- [61] B. Guo, Hu, M. C. Wu, et Z. Peng, « The Tabu_Genetic Algorithm: A Novel Method for Hyper-Parameter Optimization of Learning Algorithms », *Electronics*, vol. 8, p. 579, mai 2019, doi: 10.3390/electronics8050579.
- [62] M. Ismail, A. Hagag, et E.-S. El-Dahshan, « A hybrid GA and PSO optimized approach for heart-disease prediction based on random forest », *Multimed. Tools Appl.*, vol. 81, mars 2022, doi: 10.1007/s11042-022-12425-x.
- [63] Z. Czako, G. Sebestyen, et A. Hangan, « AutomaticAI – A hybrid approach for automatic artificial intelligence algorithm selection and hyperparameter tuning », *Expert Syst. Appl.*, vol. 182, p. 115225, nov. 2021, doi: 10.1016/j.eswa.2021.115225.
- [64] W. Wang, R. Yang, C. Guo, et H. Qin, « CNN-Based Hybrid Optimization for Anomaly Detection of Rudder System », *IEEE Access*, vol. PP, p. 1-1, sept. 2021, doi: 10.1109/ACCESS.2021.3109630.
- [65] R. Boks, H. Wang, et T. Bäck, « A Modular Hybridization of Particle Swarm Optimization and Differential Evolution », in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, juill. 2020, p. 1418-1425. doi: 10.1145/3377929.3398123.
- [66] L. Atkinson, R. Müller-Bady, et M. Kappes, « Hybrid bayesian evolutionary optimization for hyperparameter tuning », in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, in GECCO '20. New York, NY, USA: Association for Computing Machinery, juill. 2020, p. 225-226. doi: 10.1145/3377929.3389952.
- [67] A. Bahaa, A. Sayed, L. Elfangary, et H. Fahmy, « A novel hybrid optimization enabled robust CNN algorithm for an IoT network intrusion detection approach », *PLOS ONE*, vol. 17, n° 12, p. e0278493, déc. 2022, doi: 10.1371/journal.pone.0278493.
- [68] L. Japa, M. Serqueira, I. Mendonça, M. Aritsugi, E. Bezerra, et P. H. González, « A Population-Based Hybrid Approach for Hyperparameter Optimization of Neural Networks », *IEEE Access*, vol. 11, p. 50752-50768, 2023, doi: 10.1109/ACCESS.2023.3277310.
- [69] B. K. Khalifa, M. H. Bedoui, M. Dogui, et F. Alexandre, « Analysis of vigilance states by neural networks », in *Proceedings. 2004 International Conference on Information and Communication Technologies: From Theory to Applications, 2004.*, avr. 2004, p. 429-430. doi: 10.1109/ICTTA.2004.1307815.
- [70] N. Bajaj, « Wavelets for EEG Analysis », 2020. doi: 10.5772/intechopen.94398.
- [71] I. Lambert et L. Peter-Derex, « Spotlight on Sleep Stage Classification Based on EEG », *Nat. Sci. Sleep*, vol. Volume 15, p. 479-490, juin 2023, doi: 10.2147/NSS.S401270.
- [72] H. Shuang, W. Chuanfeng, et W. Qi, « Recognition of fatigue status of pilots based on deep sparse auto-encoding network », in *2017 36th Chinese Control Conference (CCC)*, juill. 2017, p. 10945-10950. doi: 10.23919/ChiCC.2017.8029103.
- [73] M. N. Tibdewal, M. Mahadevappa, A. K. Ray, M. Malokar, et H. R. Dey, « Power line and ocular artifact denoising from EEG using notch filter and wavelet transform », in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, mars 2016, p. 1654-1659.
- [74] A. Delorme et S. Makeig, « EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis », *J. Neurosci. Methods*, vol. 134, n° 1, p. 9-21, mars 2004, doi: 10.1016/j.jneumeth.2003.10.009.
- [75] A. Turnip, « Noise reduction and brain mapping based robust principal component analysis », in *2015 IEEE 12th International Conference on Networking, Sensing and Control*, avr. 2015, p. 550-553. doi: 10.1109/ICNSC.2015.7116096.
- [76] O. Falzon, K. Camilleri, et J. Muscat, « The analytic common spatial patterns method for EEG-based BCI data », *J. Neural Eng.*, vol. 9, p. 045009, juill. 2012, doi: 10.1088/1741-2560/9/4/045009.
- [77] Y. Liu et Q. Liu, « Convolutional neural networks with large-margin softmax loss function for cognitive load recognition », juill. 2017, p. 4045-4049. doi: 10.23919/ChiCC.2017.8027991.
- [78] M. Völker, R. T. Schirrmester, L. D. J. Fiederer, W. Burgard, et T. Ball, « Deep Transfer Learning for Error Decoding from Non-Invasive EEG », 10 janvier 2018, *arXiv: arXiv:1710.09139*. doi: 10.48550/arXiv.1710.09139.

- [79] S. Tian, Y. Wang, G. Dong, W. Pei, et H. Chen, « Mental Fatigue Estimation Using EEG in a Vigilance Task and Resting States », in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, juill. 2018, p. 1980-1983. doi: 10.1109/EMBC.2018.8512666.
- [80] R. T. Schirrmester, L. Gemein, K. Eggensperger, F. Hutter, et T. Ball, « Deep learning with convolutional neural networks for decoding and visualization of EEG pathology », 11 janvier 2018, *arXiv*: arXiv:1708.08012. doi: 10.48550/arXiv.1708.08012.
- [81] H. Nazeran, G. Sierra, F. Ebrahimi, et K. Setarehdan, « Wavelet-based EEG denoising for automatic sleep stage classification », présenté à *CONIELECOMP 2011 - 21st International Conference on Electronics Communications and Computers, Proceedings*, févr. 2011, p. 295-298. doi: 10.1109/CONIELECOMP.2011.5749325.
- [82] A. R. Hassan et M. I. H. Bhuiyan, « Dual tree complex wavelet transform for sleep state identification from single channel electroencephalogram », in *2015 IEEE International Conference on Telecommunications and Photonics (ICTP)*, déc. 2015, p. 1-5. doi: 10.1109/ICTP.2015.7427924.
- [83] H. Dong, A. Supratak, W. Pan, C. Wu, P. M. Matthews, et Y. Guo, « Mixed Neural Network Approach for Temporal Sleep Stage Classification », *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, n° 2, p. 324-333, févr. 2018, doi: 10.1109/TNSRE.2017.2733220.
- [84] L.-H. Du, W. Liu, W.-L. Zheng, et B.-L. Lu, « Detecting driving fatigue with multimodal deep learning », in *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*, mai 2017, p. 74-77. doi: 10.1109/NER.2017.8008295.
- [85] N. Zhang, W.-L. Zheng, W. Liu, et B.-L. Lu, « Continuous Vigilance Estimation Using LSTM Neural Networks », in *Neural Information Processing*, A. Hirose, S. Ozawa, K. Doya, K. Ikeda, M. Lee, et D. Liu, Éd., in *Lecture Notes in Computer Science*. Cham: Springer International Publishing, 2016, p. 530-537. doi: 10.1007/978-3-319-46672-9_59.
- [86] C. Han, Y. Yang, X. Sun, M. Yang, et Y. Qin, « SVM-based Multi-classification for Detection of Vigilance Levels with Single-Channel EEG Signals », in *2019 Chinese Control And Decision Conference (CCDC)*, juin 2019, p. 607-612. doi: 10.1109/CCDC.2019.8833229.
- [87] F. M. Al-Shargie, O. Hassanin, U. Tariq, et H. Al-Nashash, « EEG-Based Semantic Vigilance Level Classification Using Directed Connectivity Patterns and Graph Theory Analysis », *IEEE Access*, vol. 8, p. 115941-115956, 2020, doi: 10.1109/ACCESS.2020.3004504.
- [88] A. Saidi, S. Ben Othman, et S. Ben Saoud, « A novel epileptic seizure detection system using scalp EEG signals based on hybrid CNN-SVM classifier », in *2021 IEEE Symposium on Industrial Electronics & Applications (ISIEA)*, juill. 2021, p. 1-6. doi: 10.1109/ISIEA51897.2021.9510002.
- [89] I. Assali, A. Ghazi Blaiech, A. Ben Abdallah, K. Ben Khalifa, M. Carrère, et M. Hédi Bedoui, « CNN-based classification of epileptic states for seizure prediction using combined temporal and spectral features », *Biomed. Signal Process. Control*, vol. 82, p. 104519, avr. 2023, doi: 10.1016/j.bspc.2022.104519.
- [90] Y. Ma *et al.*, « A Multi-Channel Feature Fusion CNN-Bi-LSTM Epilepsy EEG Classification and Prediction Model Based on Attention Mechanism », *IEEE Access*, vol. 11, p. 62855-62864, 2023, doi: 10.1109/ACCESS.2023.3287927.
- [91] K. Shekogar et S. Dour, « Identification of Epileptic Seizures using CNN on Noisy EEG Signals », in *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, déc. 2022, p. 185-188. doi: 10.1109/ICECA55336.2022.10009127.
- [92] N. Truong *et al.*, « Convolutional neural networks for seizure prediction using intracranial and scalp electroencephalogram », *Neural Netw.*, vol. 105, mai 2018, doi: 10.1016/j.neunet.2018.04.018.
- [93] H. Khan, L. Marcuse, M. Fields, K. Swann, et B. Yener, « Focal Onset Seizure Prediction Using Convolutional Networks », *IEEE Trans. Biomed. Eng.*, vol. 65, n° 9, p. 2109-2118, sept. 2018, doi: 10.1109/TBME.2017.2785401.
- [94] S. Lee, R. Hussein, et M. J. McKeown, « A Deep Convolutional-Recurrent Neural Network Architecture for Parkinson's Disease EEG Classification », in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, nov. 2019, p. 1-4. doi: 10.1109/GlobalSIP45357.2019.8969309.
- [95] X. Shi, T. Wang, L. Wang, H. Liu, et N. Yan, « Hybrid Convolutional Recurrent Neural Networks Outperform CNN and RNN in Task-state EEG Detection for Parkinson's Disease », in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, nov. 2019, p. 939-944. doi: 10.1109/APSIPAASC47483.2019.9023190.

- [96] P. Sandheep, S. Vineeth, M. Poulouse, et D. P. Subha, « Performance analysis of deep learning CNN in classification of depression EEG signals », in *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, oct. 2019, p. 1339-1344. doi: 10.1109/TENCON.2019.8929254.
- [97] A. Boudaya, S. Chaabene, B. Bouaziz, H. Zouari, S. ben Jemea, et L. Chaari, « Physiological/Non-physiological artifacts classification using EEG signals based on CNN », in *2022 International Conference on Technology Innovations for Healthcare (ICTIH)*, sept. 2022, p. 26-29. doi: 10.1109/ICTIH57289.2022.10112088.
- [98] H. Donmez et N. Ozkurt, « Emotion Classification from EEG Signals in Convolutional Neural Networks », in *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*, oct. 2019, p. 1-6. doi: 10.1109/ASYU48272.2019.8946364.
- [99] M. Su, W. Li, F. Peng, W. Zhou, R. Zhang, et Y. Wen, « EEG-Based Mental Fatigue Detection Using CNN-LSTM », in *2022 16th ICME International Conference on Complex Medical Engineering (CME)*, nov. 2022, p. 302-305. doi: 10.1109/CME55444.2022.10063316.
- [100] M. A. Almogbel, A. H. Dang, et W. Kameyama, « EEG-signals based cognitive workload detection of vehicle driver using deep learning », in *2018 20th International Conference on Advanced Communication Technology (ICACT)*, févr. 2018, p. 256-259. doi: 10.23919/ICACT.2018.8323716.
- [101] S. Chambon, M. Galtier, P. Arnal, G. Wainrib, et A. Gramfort, « A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series », 27 novembre 2017, *arXiv*: arXiv:1707.03321. Consulté le: 21 août 2023. [En ligne]. Disponible sur: <http://arxiv.org/abs/1707.03321>
- [102] U. Asgher, K. Khalil, Y. Ayaz, R. Ahmad, et M. J. Khan, « Classification of Mental Workload (MWL) using Support Vector Machines (SVM) and Convolutional Neural Networks (CNN) », in *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, janv. 2020, p. 1-6. doi: 10.1109/iCoMET48670.2020.9073799.
- [103] A. Abidi, K. Ben Khalifa, R. Ben Cheikh, C. A. Valderrama Sakuyama, et M. H. Bedoui, « Automatic Detection of Drowsiness in EEG Records Based on Machine Learning Approaches », *Neural Process. Lett.*, vol. 54, n° 6, p. 5225-5249, déc. 2022, doi: 10.1007/s11063-022-10858-x.
- [104] R. Lachtar, S. Jovanovic, K. B. Khalifa, R. B. Cheikh, S. Weber, et M. H. Bedoui, « Drowsiness Detection with a Limited Number of EEG Physiological Signals », in *2022 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS)*, juin 2022, p. 01-05. doi: 10.1109/DTS55284.2022.9809842.
- [105] M. Hajnoroozi, J. M. Zhang, et Y. Huang, « Driver's fatigue prediction by deep covariance learning from EEG », in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, oct. 2017, p. 240-245. doi: 10.1109/SMC.2017.8122609.
- [106] S. Khessiba, A. Blaiech, K. Ben Khalifa, A. Ben Abdallah, et M. H. Bedoui, « Correction to: Innovative deep learning models for EEG-based vigilance detection », *Neural Comput. Appl.*, vol. 34, janv. 2022, doi: 10.1007/s00521-021-06187-0.
- [107] K. B. Khalifa, M. H. Bedoui, M. Dogui, et F. Alexandre, « Analysis of vigilance states by neural networks », in *Proceedings. 2004 International Conference on Information and Communication Technologies: From Theory to Applications, 2004.*, Damascus, Syria: IEEE, 2004, p. 429-430. doi: 10.1109/ICTTA.2004.1307815.
- [108] K. B. Khalifa, M. H. Bédoui, R. Raychev, M. Dogui, et F. Alexandre, « Séparation des niveaux de vigilance, à partir d'un signal EEG par les cartes auto-organisatrices de Kohonen ».
- [109] M. Boubaker, M. Akil, K. Ben Khalifa, T. Grandpierre, et M. H. Bedoui, « Implementation of an LVQ neural network with a variable size: algorithmic specification, architectural exploration and optimized implementation on FPGA devices », *Neural Comput. Appl.*, vol. 19, n° 2, p. 283-297, mars 2010, doi: 10.1007/s00521-009-0296-5.
- [110] M. Billiard, « Histoire de la médecine du sommeil en France », *Médecine Sommeil*, vol. 15, n° 3, p. 130-150, août 2018, doi: 10.1016/j.msom.2018.06.001.
- [111] S. Khessiba, A. G. Blaiech, K. Ben Khalifa, A. Ben Abdallah, et M. H. Bedoui, « Innovative deep learning models for EEG-based vigilance detection », *Neural Comput. Appl.*, vol. 33, n° 12, p. 6921-6937, juin 2021, doi: 10.1007/s00521-020-05467-5.
- [112] Y. Ozaki, M. Yano, et M. Onishi, « Effective hyperparameter optimization using Nelder-Mead method in deep learning », *IPSJ Trans. Comput. Vis. Appl.*, vol. 9, n° 1, p. 20, nov. 2017, doi: 10.1186/s41074-017-0030-7.

- [113] « Arthrose », Viggo Petersen. Consulté le: 29 janvier 2024. [En ligne]. Disponible sur: <http://www.viggopetersen.fr/espace-patient/tout-savoir-sur-la-maladie/arthrose/>
- [114] K. D. Brandt, E. L. Radin, P. A. Dieppe, et L. van de Putte, « Yet more evidence that osteoarthritis is not a cartilage disease », *Ann. Rheum. Dis.*, vol. 65, n° 10, p. 1261-1264, oct. 2006, doi: 10.1136/ard.2006.058347.
- [115] W. H. Harris et C. B. Sledge, « Total hip and total knee replacement (1) », *N. Engl. J. Med.*, vol. 323, n° 11, p. 725-731, sept. 1990, doi: 10.1056/NEJM199009133231106.
- [116] R. Erivan *et al.*, « Knee surgery trends and projections in France from 2008 to 2070 », *Orthop. Traumatol. Surg. Res. OTSR*, vol. 106, n° 5, p. 893-902, sept. 2020, doi: 10.1016/j.otsr.2020.02.018.
- [117] K. Bennell, D. Hunter, et R. Hinman, « Management of osteoarthritis of the knee », *BMJ*, vol. 345, p. e4934, juill. 2012, doi: 10.1136/bmj.e4934.
- [118] V. Silverwood, M. Blagojevic-Bucknall, C. Jinks, J. L. Jordan, J. Protheroe, et K. P. Jordan, « Current evidence on risk factors for knee osteoarthritis in older adults: a systematic review and meta-analysis », *Osteoarthritis Cartilage*, vol. 23, n° 4, p. 507-515, avr. 2015, doi: 10.1016/j.joca.2014.11.019.
- [119] A. Bayés et J. Hernández-Vara, « Management of Parkinson's Disease: Challenges and Solutions », 2023, p. 1-19. doi: 10.1201/9781032632865-1.
- [120] A. Ramirez-Zamora et T. Tsuboi, « Hospital Management of Parkinson Disease Patients », *Clin. Geriatr. Med.*, vol. 36, n° 1, p. 173-181, févr. 2020, doi: 10.1016/j.cger.2019.09.009.
- [121] R. Tri Wahyuningrum, A. Yasid, et G. Jacob Verkerke, « Deep Neural Networks for Automatic Classification of Knee Osteoarthritis Severity Based on X-ray Images », in *Proceedings of the 2020 8th International Conference on Information Technology: IoT and Smart City*, in ICIT '20. New York, NY, USA: Association for Computing Machinery, avr. 2021, p. 110-114. doi: 10.1145/3446999.3447020.
- [122] Y. Du, J. Shan, R. Almajalid, et M. Zhang, « Knee osteoarthritis severity level classification using whole knee cartilage damage Index and ANN », in *Proceedings of the 2018 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies*, Washington DC: ACM, sept. 2018, p. 19-21. doi: 10.1145/3278576.3278585.
- [123] Y. Guo, X. Wu, L. Shen, Z. Zhang, et Y. Zhang, « Method of gait disorders in Parkinson's disease classification based on machine learning algorithms », in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Chongqing, China: IEEE, mai 2019, p. 768-772. doi: 10.1109/ITAIC.2019.8785586.
- [124] L. Gong, J. Li, M. Yu, M. Zhu, et R. Clifford, « A novel computer vision based gait analysis technique for normal and Parkinson's gaits classification », in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCOM/CyberSciTech)*, août 2020, p. 209-215. doi: 10.1109/DASC-PiCom-CBDCOM-CyberSciTech49142.2020.00045.
- [125] N. P. Sharma, I. Junaid, et S. Ari, « Early Diagnosis of Parkinson's Disease and Severity Assessment based on Gait using 1D-CNN », in *2023 2nd International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)*, avr. 2023, p. 1-6. doi: 10.1109/ICSTSN57873.2023.10151641.
- [126] Shivangi, A. Johri, et A. Tripathi, « Parkinson Disease Detection Using Deep Neural Networks », in *2019 Twelfth International Conference on Contemporary Computing (IC3)*, août 2019, p. 1-4. doi: 10.1109/IC3.2019.8844941.
- [127] R. Kaur, R. W. Motl, R. Sowers, et M. E. Hernandez, « A Vision-Based Framework for Predicting Multiple Sclerosis and Parkinson's Disease Gait Dysfunctions—A Deep Learning Approach », *IEEE J. Biomed. Health Inform.*, vol. 27, n° 1, p. 190-201, janv. 2023, doi: 10.1109/JBHI.2022.3208077.
- [128] N. Kour, S. Gupta, et S. Arora, « A vision-based clinical analysis for classification of knee osteoarthritis, Parkinson's disease and normal gait with severity based on k-nearest neighbour », *Expert Syst.*, vol. 39, févr. 2022, doi: 10.1111/exsy.12955.
- [129] S. Yan, Y. Xiong, et D. Lin, « Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition », 25 janvier 2018, *arXiv: arXiv:1801.07455*. doi: 10.48550/arXiv.1801.07455.
- [130] C. Von Tycowicz, « Towards Shape-Based Knee Osteoarthritis Classification Using Graph Convolutional Networks », in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, Iowa City, IA, USA: IEEE, avr. 2020, p. 750-753. doi: 10.1109/ISBI45749.2020.9098687.

- [131] T. Connie, T. B. Aderinola, T. S. Ong, M. K. O. Goh, B. Erfianto, et B. Purnama, « Pose-Based Gait Analysis for Diagnosis of Parkinson’s Disease », *Algorithms*, vol. 15, n° 12, p. 474, déc. 2022, doi: 10.3390/a15120474.
- [132] M. S. Alsawadi, E.-S. M. El-kenawy, et M. Rio, « Using BlazePose on Spatial Temporal Graph Convolutional Networks for Action Recognition », *Comput. Mater. Contin.*, vol. 74, n° 1, p. 19-36, 2023, doi: 10.32604/cmc.2023.032499.
- [133] J. Zhang, J. Lim, M.-H. Kim, S. Hur, et T.-M. Chung, « WM-STGCN: A Novel Spatiotemporal Modeling Method for Parkinsonian Gait Recognition », *Sensors*, vol. 23, n° 10, p. 4980, mai 2023, doi: 10.3390/s23104980.
- [134] T. Lyu et H. Guo, « BGCN: An EEG-based Graphical Classification Method for Parkinson’s Disease Diagnosis with Heuristic Functional Connectivity Speculation », in *2023 11th International IEEE/EMBS Conference on Neural Engineering (NER)*, avr. 2023, p. 1-4. doi: 10.1109/NER52421.2023.10123796.
- [135] « Human Pose Estimation: Deep Learning Approach [2023 Guide] ». Consulté le: 29 janvier 2024. [En ligne]. Disponible sur: <https://www.v7labs.com/blog/human-pose-estimation-guide>
- [136] A. Toshev et C. Szegedy, « DeepPose: Human Pose Estimation via Deep Neural Networks », in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, juin 2014, p. 1653-1660. doi: 10.1109/CVPR.2014.214.
- [137] X. Sun, J. Shang, S. Liang, et Y. Wei, « Compositional Human Pose Regression », 1 août 2017, *arXiv*: arXiv:1704.00159. doi: 10.48550/arXiv.1704.00159.
- [138] X. Fan, K. Zheng, Y. Lin, et S. Wang, « Combining Local Appearance and Holistic View: Dual-Source Deep Neural Networks for Human Pose Estimation », 27 avril 2015, *arXiv*: arXiv:1504.07159. doi: 10.48550/arXiv.1504.07159.
- [139] L. Pishchulin *et al.*, « DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation », 26 avril 2016, *arXiv*: arXiv:1511.06645. doi: 10.48550/arXiv.1511.06645.
- [140] J. Tompson, A. Jain, Y. LeCun, et C. Bregler, « Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation », 17 septembre 2014, *arXiv*: arXiv:1406.2984. doi: 10.48550/arXiv.1406.2984.
- [141] S.-E. Wei, V. Ramakrishna, T. Kanade, et Y. Sheikh, « Convolutional Pose Machines », 11 avril 2016, *arXiv*: arXiv:1602.00134. doi: 10.48550/arXiv.1602.00134.
- [142] B. Wattanacheep et O. Chitsobhuk, « Camera Pose Estimation using CNN », in *Proceedings of the 3rd International Conference on Control and Computer Vision*, in ICCCV ’20. New York, NY, USA: Association for Computing Machinery, janv. 2021, p. 84-88. doi: 10.1145/3425577.3425593.
- [143] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, et C. Bregler, « Efficient Object Localization Using Convolutional Networks », 9 juin 2015, *arXiv*: arXiv:1411.4280. doi: 10.48550/arXiv.1411.4280.
- [144] B. Sapp et B. Taskar, « MODEC: Multimodal Decomposable Models for Human Pose Estimation », in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, in CVPR ’13. USA: IEEE Computer Society, juin 2013, p. 3674-3681. doi: 10.1109/CVPR.2013.471.
- [145] M. Andriluka, L. Pishchulin, P. Gehler, et B. Schiele, « 2D Human Pose Estimation: New Benchmark and State of the Art Analysis », présenté à Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, p. 3686-3693. Consulté le: 30 août 2023. [En ligne]. Disponible sur: https://openaccess.thecvf.com/content_cvpr_2014/html/Andriluka_2D_Human_Pose_2014_CVPR_paper.html
- [146] V. Belagiannis et A. Zisserman, « Recurrent Human Pose Estimation », in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, mai 2017, p. 468-475. doi: 10.1109/FG.2017.64.
- [147] G. Papandreou *et al.*, « Towards Accurate Multi-person Pose Estimation in the Wild », 14 avril 2017, *arXiv*: arXiv:1701.01779. doi: 10.48550/arXiv.1701.01779.
- [148] I. J. Goodfellow *et al.*, « Generative Adversarial Networks », 10 juin 2014, *arXiv*: arXiv:1406.2661. Consulté le: 30 août 2023. [En ligne]. Disponible sur: <http://arxiv.org/abs/1406.2661>
- [149] Y. Chen, C. Shen, X.-S. Wei, L. Liu, et J. Yang, « Adversarial PoseNet: A Structure-aware Convolutional Network for Human Pose Estimation », 1 mai 2017, *arXiv*: arXiv:1705.00389. doi: 10.48550/arXiv.1705.00389.

- [150] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, et M. Grundmann, « BlazePose: On-device Real-time Body Pose tracking », 17 juin 2020, *arXiv*: arXiv:2006.10204. doi: 10.48550/arXiv.2006.10204.
- [151] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, et B. Schiele, « DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model », 30 novembre 2016, *arXiv*: arXiv:1605.03170. doi: 10.48550/arXiv.1605.03170.
- [152] S. Kreiss, L. Bertoni, et A. Alahi, « OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association », *IEEE Trans. Intell. Transp. Syst.*, vol. 23, n° 8, p. 13498-13511, août 2022, doi: 10.1109/TITS.2021.3124981.
- [153] A. Newell, K. Yang, et J. Deng, « Stacked Hourglass Networks for Human Pose Estimation », 26 juillet 2016, *arXiv*: arXiv:1603.06937. doi: 10.48550/arXiv.1603.06937.
- [154] M. Lu *et al.*, *Vision-based Estimation of MDS-UPDRS Gait Scores for Assessing Parkinson's Disease Motor Severity*. 2020.
- [155] Y. Gu, S. Pandit, E. Saraee, T. Nordahl, T. Ellis, et M. Betke, « Home-Based Physical Therapy with an Interactive Computer Vision System », in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Seoul, Korea (South): IEEE, oct. 2019, p. 2619-2628. doi: 10.1109/ICCVW.2019.00320.
- [156] W. Chen, Z. Jiang, H. Guo, et X. Ni, « Fall Detection Based on Key Points of Human-Skeleton Using OpenPose », *Symmetry*, vol. 12, p. 744, mai 2020, doi: 10.3390/sym12050744.
- [157] N. Kour, S. Gupta, et S. Arora, « Gait Dataset for Knee Osteoarthritis and Parkinson's Disease Analysis With Severity Levels », vol. 1, déc. 2020, doi: 10.17632/44pfnysy89.1.
- [158] S. Bhat, U. R. Acharya, Y. Hagiwara, N. Dadmehr, et H. Adeli, « Parkinson's disease: Cause factors, measurable indicators, and early diagnosis », *Comput. Biol. Med.*, vol. 102, p. 234-241, nov. 2018, doi: 10.1016/j.combiomed.2018.09.008.
- [159] S. Gornale, « A Survey on Exploration and Classification of Osteoarthritis Using Image Processing Techniques », *Int. J. Sci. Eng. Res.*, vol. 7, p. 334-355, juin 2016.

Titre : Stratégies d'optimisation des hyperparamètres de réseaux de neurones appliqués aux signaux temporels biomédicaux

Mots clés : Classification de séries temporelles biomédicales, Optimisation des hyperparamètres, Apprentissage Profond, Analyse du geste.

Résumé : Les progrès récents en apprentissage automatique ouvrent de nouvelles perspectives pour les systèmes d'aide à la décision médicale. Les modèles d'apprentissage profond, en particulier les réseaux de neurones à convolution (CNN), ont démontré des performances exceptionnelles dans les contextes médicaux, offrant des capacités de détection, de classification et de prédiction de diverses pathologies. Cependant, l'optimisation des réseaux CNN pose un défi significatif en raison de nombreux hyperparamètres gouvernant leur performance. Cette thèse aborde ce défi en proposant des méthodologies innovantes pour l'optimisation efficace des hyperparamètres des réseaux CNN dans le contexte des systèmes d'aide à la décision médicale. Deux principales contributions sont présentées :

Le développement d'une méthode d'optimisation hybride, GS-TPE pour régler efficacement les hyperparamètres des modèles de réseaux profonds utilisés dans la classification des états de vigilance à partir de signaux physiologiques tels que l'EEG. GS-TPE a montré des améliorations significatives des performances par rapport aux méthodes d'optimisation classiques.

L'introduction d'une nouvelle architecture de réseau profond, STGCN, pour classer les gestes liés à l'arthrose du genou et à la maladie de Parkinson à partir de l'analyse vidéo de la marche. STGCN capture les caractéristiques spatio-temporelles des gestes, démontrant son efficacité dans la discrimination entre différentes pathologies, impactant potentiellement les pratiques cliniques.

Title : Optimization strategies for neural networks hyperparameters applied to biomedical temporal signals.

Keywords : Classification of biomedical time series, Hyperparameter Optimisation, Deep Learning, Gesture analysis.

Abstract : Recent advancements in machine learning have opened up new possibilities for medical decision support systems. Deep learning models, particularly convolutional neural networks (CNNs), have demonstrated outstanding performance in medical contexts, providing capabilities for detection, classification, and prediction of various pathologies. However, optimizing CNNs poses a significant challenge due to numerous hyperparameters governing their performance. This thesis addresses this challenge by proposing innovative methodologies for efficiently optimizing the hyperparameters of CNNs in the context of medical decision support systems. Two main contributions are presented :

The development of a hybrid optimization method GS-TPE to effectively optimize the hyperparameters of deep neural network models used in classifying vigilance states from physiological signals such as EEG. GS-TPE has shown significant performance improvements compared to classical optimization methods.

The introduction of a new deep network architecture, STGCN, for classifying gestures related to Knee Osteoarthritis and Parkinson's disease from video gait analysis. STGCN captures the Spatio-Temporal characteristics of gestures, demonstrating its effectiveness in discriminating between different pathologies, potentially impacting clinical practices.