

Rapport projet Deep Learning

Classification de sons pensés par analyse de signaux EEG

Menasria Racim
Cavina Théo

CONTENTS

I	Abstract	1
II	Introduction	1
III	Méthodologie	2
IV	Pré-traitement des données	2
IV-A	Zoom sur les données	2
IV-B	Normalisation et filtrage	2
IV-C	Transformée en ondelettes	2
IV-D	Wavelet scattering transform	3
V	Apprentissage	3
V-A	Premier modèle	3
V-B	Essai de modèles plus simples	4
VI	Etude d'un problème plus simple	4
VI-A	Données	5
VI-B	Preprocessing	5
VI-C	Décomposition en ondelettes de Daubechies	5
VI-D	Décomposition en ondelettes stationnaires	5
VI-E	Décomposition par wavelet scattering (avec moyennage temporel)	6
VI-F	Modèles	6
VI-G	Résultats	7
VII	Conclusion	7

I. ABSTRACT

Qui n'a jamais rêvé de pouvoir lire dans les pensées des autres ? Idée autrefois fantasque, le développement de l'analyse des électroencéphalogrammes ou EEG a permis de nombreux progrès dans la compréhension des phénomènes et représentations cognitives ces dernières années. Il ouvre la porte à l'élaboration de solutions pour le traitement des troubles du sommeil, l'amélioration de la représentation cérébrale du langage ou encore de la perception visuelle. Cependant, ces avancées s'accompagnent de défis importants, telle que la nécessité de grandes quantités de données annotées pour l'entraînement des modèles, la variabilité interindividuelle des signaux EEG et les préoccupations éthiques liées à la confidentialité des données cérébrales. Plus récemment, l'application de l'apprentissage profond dans l'analyse des signaux EEG a permis de franchir un palier de performance.

Dans cette étude exploratoire, nous proposons de traiter une tâche de classification de mots ou lettres pensées par la mise en place d'une pipeline allant du traitement et de la préparation des données EEG à leur traitement par réseaux de neurones.

Nous avons dans un premier temps travaillé à reproduire une architecture précédemment divulguée par l'état de l'art en l'appliquant à un dataset utilisé fréquemment dans la littérature. Par la suite, nous sommes revenus à une tâche de classification binaire plus simple sur un autre dataset également référencé afin d'étudier l'influence de la représentation du signal EEG sur les performances et la dépendance vis-à-vis de la variabilité interindividuelle. Le code complet et commenté est disponible à l'adresse suivante : <https://github.com/menasrac/EEGDL>

II. INTRODUCTION

a) Objectifs du projet : L'objectif du projet est de reproduire une méthodologie connue de la littérature [1] pour parvenir à développer un modèle de classification de signaux d'électroencéphalogrammes mesurant l'activité électrique du cortex cérébral lors de l'imagination de parole. Pour cela, nous nous appuyons sur 2 datasets (dataset FEIS [5] et [3]). La méthodologie se concentre sur l'application d'algorithmes de transformation ondelettes du signal ou wavelet transform et d'implémentation d'un LSTM. Cette même méthode a permis d'obtenir d'autres excellents résultats dans [2].

b) Résultats: Avant de détailler notre approche, mentionnons que nous ne sommes pas parvenus à adapter la méthode et que nous n'avons pas réussi à entraîner de modèle sur le dataset FEIS. Par conséquent, nous avons essayé de résoudre un problème plus simple sur un dataset contenant moins de labels. De cette manière, nous sommes parvenus à entraîner un LSTM sur des données EEG mais également d'autres architectures comme un MLP ou un 1D-CNN. A travers un travail de design d'architecture de petits modèles, nous comparons dans la deuxième partie de cette étude leurs performances. Nous mettons par la même occasion en évidence leur robustesse ou faiblesse face à la variabilité des signaux entre les sujets et soulignons le rôle clé du choix de la représentation des données en ondelettes dans l'apprentissage.

c) Etat de l'art: Lire les signaux électriques du cerveau pour essayer de deviner la pensée d'un individu est un objectif qui est de plus en plus en vogue depuis l'avènement du deep learning. Les cas d'usage

seraient nombreux pour aider les patients handicapés ou muets. De nouveaux modèles arrivent à extraire et utiliser toujours plus d'information à partir des signaux d'électroencéphalogramme. Avant que les réseaux de neurones et que le deep learning ne soient aussi courants, des méthodes de machine learning plus traditionnelles avec des algorithmes tels que SVM ou RandomForest ont déjà montré qu'il était possible d'inférer des commandes souhaitées par l'utilisateur à partir des signaux cérébraux [4]. Plus récemment et avec des algorithmes de deep learning, un papier annonce avoir développé un modèle basé sur un LSTM atteignant 95% de précision sur la classification de commande par la pensée pour le contrôle d'un fauteuil électrique [2]. Aussi, dans le papier de référence [1], il semble que l'architecture LSTM montre de grandes capacités pour traiter les signaux EEG (>90% d'accuracy). Ainsi, nous nous proposons de suivre un processus expérimental similaire mais impliquant d'autres actions. Pour cela, nous utiliserons le dataset FEIS [5] qui se focalise sur l'imagination de prononciation de sons de base de la langue anglaise.

III. MÉTHODOLOGIE

Notre objectif est donc de vérifier si l'approche utilisée par [1] est suffisamment générale pour être utilisée, non pas pour l'identification de position dans l'espace, mais pour la discrimination de sons imaginés par un individu. Ainsi, nous procéderons par étapes :

- 1) Prise de connaissance du dataset
- 2) Application des méthodes de pré-traitement de [1]
- 3) Entraînement du modèle
- 4) Analyse des résultats et recherche de potentielles erreurs
- 5) Simplification du problème

Afin de parvenir à des résultats fiables nous avons mis en place des datasets d'entraînement, validation et test pour chaque problème étudié. Dans la deuxième partie de notre travail, nous avons aussi mis en place des méthodes de grid search afin d'identifier les hyper-paramètres les plus prometteurs.

IV. PRÉ-TRAITEMENT DES DONNÉES

A. Zoom sur les données

FEIS est une base de données rassemblant le résultat d'enregistrements de signaux d'électroencéphalogramme de 21 personnes s'imaginant prononcer 16 différents sons de base de la langue anglaise. La base de données contient également des enregistrements audios mais nous ne nous y intéresserons pas. La méthode pour l'acquisition est également décrite :

- L'enregistrement a été réalisé avec 14 électrodes, les signaux comportent donc 14 channels
- La fréquence de mesure est de 256 Hz
- Chaque acquisition a duré 5 secondes, menant à 1280 points valeur par piste et par acquisition
- Il y a 160 acquisitions par personnes (excepté l'individu 12 avec qui il y a eu 112 acquisitions).
- Fig. 1 donne la visualisation d'un exemple d'acquisition

B. Normalisation et filtrage

Nous effectuons une normalisation et appliquons un filtre passe-bande à nos données. La normalisation est essentielle car les valeurs enregistrées par les canaux sont de l'ordre 4000 μV .

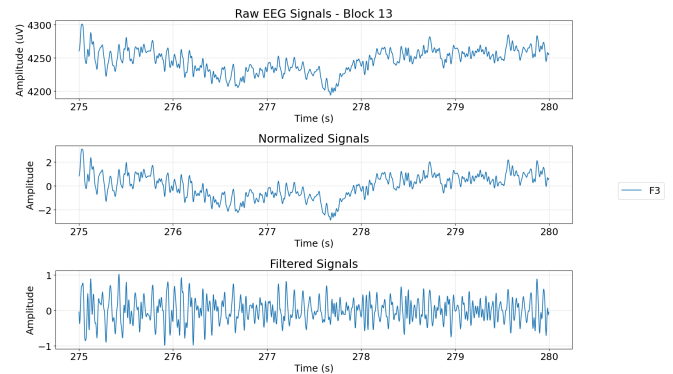


Fig. 1: Exemple du signal enregistré sur le premier canal, normalisées et filtrées pour l'individu 7 et le label "f". Il s'agit de la 13ème acquisition de l'individu 7. La légende indique le nom du canal

Remarque sur le filtrage :

Une transformée de Fourier sur nos signaux (Fig.2) révèle que leur fréquence n'est pas adéquate au preprocessing décrit dans le papier que nous avons essayé de répliquer. Tandis que [1] suggère de filtrer entre 10 et 100 Hz, la transformée de Fourier de nos signaux ne montre aucune composante au-delà de 60 Hz.

C. Transformée en ondelettes

La transformée en ondelettes est une technique d'analyse des signaux qui permet de représenter un signal simultanément dans les domaines temporel et fréquentiel. Contrairement à la transformée de Fourier, qui décompose un signal en une somme de sinusoides infinies et fournit une information globale sur les fréquences présentes sans indication de leur localisation temporelle, la transformée en ondelettes utilise des fonctions appelées "ondelettes" qui sont localisées à la fois dans le temps et en fréquence. Cette approche offre une analyse multi-résolution, permettant de capturer efficacement les caractéristiques locales des signaux, ce qui est particulièrement utile pour les signaux non-stationnaires tels que les signaux EEG ou présentant des discontinuités.

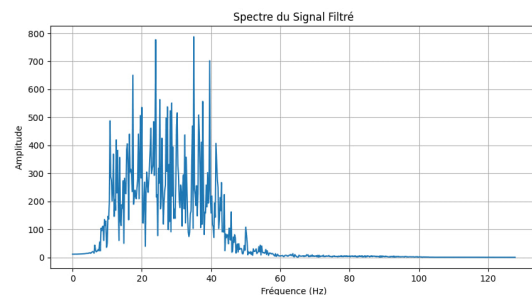


Fig. 2: Spectre d'un signal de FEIS après filtrage entre 10 et 100Hz.

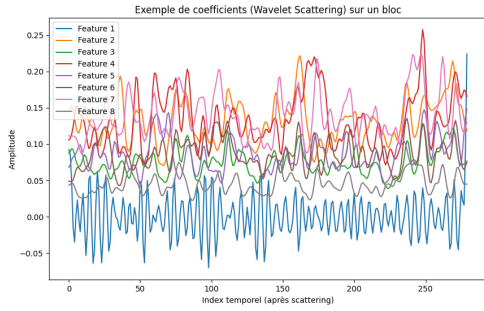


Fig. 3: Représentation des 8 premiers coefficients de scattering à notre fréquence d'échantillonnage 256Hz pour sans moyennage temporel pour préparation de l'apprentissage du LSTM

D. Wavelet scattering transform

La transformée de scattering par ondelettes est une extension de la transformée en ondelettes qui vise à extraire des caractéristiques invariantes et stables des signaux, tout en préservant leur structure hiérarchique. Elle consiste en une cascade de convolutions avec des ondelettes suivies de modules et de moyennes, capturant ainsi des informations de plus en plus fines à chaque couche. Cette méthode est particulièrement efficace pour la classification de signaux complexes, car elle fournit une représentation robuste aux variations et aux déformations, tout en réduisant la dimensionnalité des données.

Kymatio est une bibliothèque Python dédiée à l'implémentation de la transformée de scattering par ondelettes. Compatible avec des frameworks d'apprentissage profond tels que PyTorch, Kymatio permet d'intégrer facilement la transformée de scattering dans des pipelines de machine learning, facilitant ainsi l'extraction de caractéristiques pour diverses applications, notamment en traitement du signal et en reconnaissance de formes.

Lors de l'utilisation de la transformée de scattering par ondelettes, il est possible d'appliquer une opération de moyenne sur l'axe temporel pour obtenir des coefficients invariants par translation, ce qui est utile pour certaines tâches de classification. Cependant, si l'objectif est d'entraîner un réseau de neurones récurrent de type LSTM (Long Short-Term Memory), il peut être préférable de conserver la dimension temporelle des coefficients de scattering. Cela permet au LSTM de modéliser les dépendances temporelles présentes dans les signaux, en exploitant pleinement la structure séquentielle des données pour des performances accrues dans les tâches de prédiction ou de classification temporelle.

V. APPRENTISSAGE

A. Premier modèle

Une fois le traitement des données effectué, nous avons essayé d'implémenter le modèle (Fig.5) et le processus d'apprentissage décrit dans [1].

Malheureusement, certaines informations (taille de la couche cachée par exemple) quant à l'architecture ne sont pas précisées. Nos données étant différentes,

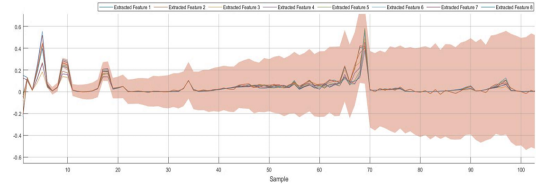


Figure 5: Eight – extracted features using wavelet scattering transformation at 250 Hz (250 samples per second).

Fig. 4: Représentation des 8 premiers coefficients de scattering à 250Hz sans moyennage temporel pour préparation de l'apprentissage du LSTM dans le papier de référence

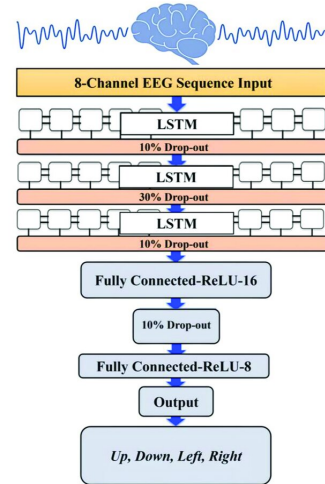


Fig. 5: Architecture implémentée dans [1]

nous n'aurions certainement pas pris les mêmes hyper-paramètres pour notre modèle.

Nous n'explicitons pas de valeurs précises pour les hyper-paramètres car nous ne sommes pas parvenus à faire fonctionner le processus d'apprentissage pour un LSTM, nous avons fait varier les hyper-paramètres dans les intervalles suivants:

- LSTM (nombre de couche $\in [3, 10]$, taille de la couche cachée $\in [20, 120]$)
- Couche dense (nombre de neurones $\in [32, 128]$)
- Couche dense (nombre de neurones = 16, nombre de labels)

Pour la phase d'entraînement, nous avons utilisé la Cross Entropy comme fonction de coût (en n'appliquant donc pas de couche SoftMax en sortie du modèle) à laquelle nous avons ajouté une régularisation L2. Pour aider à la généralisation du modèle, nous avons également ajouté du Dropout. Différentes valeurs du learning rate ont été essayées (dans $[0.05, 0.0001]$), sans succès.

Nous ne sommes pas parvenus à observer de phénomène d'apprentissage autre que de l'apprentissage par cœur du dataset d'entraînement. Nous avons principalement observé 2 tendances lors des phases d'entraînement. Dans certains cas, le réseau n'apprend pas (Fig. 6), dans d'autres cas, on observe un apprentissage par cœur des données d'entraînement (Fig. 7). La taille du modèle semble permettre ou non cet apprentissage par cœur.

À l'issue des entraînements, nous observons à l'aide des matrices de confusion (sur l'ensemble de test) que dans le cas de non-apprentissage, le modèle se retrouve

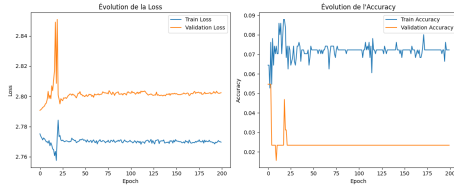


Fig. 6: Exemple de courbes d'apprentissage lorsque rien n'est appris (learning rate = 5×10^{-4} et 3 couches cachées de taille 32)

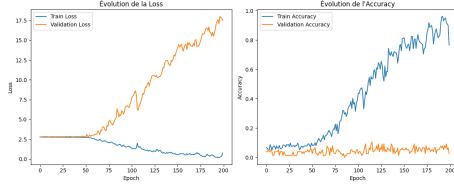


Fig. 7: Exemple de courbes d'apprentissage lorsque le modèle apprend par coeur (learning rate = 5×10^{-4} et 7 couches cachées de taille 128)

à systématiquement choisir le même label (Fig.8). Dans le cas d'apprentissage par cœur, le modèle semble choisir des labels au hasard (Fig. 9).

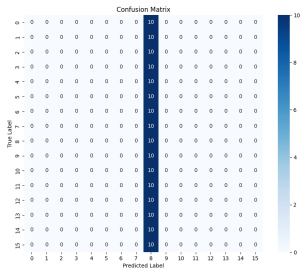


Fig. 8: Exemple de matrice de confusion dans le cas où rien n'est appris

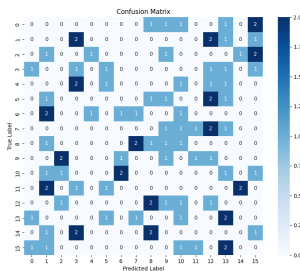


Fig. 9: Exemple de matrice de confusion dans le cas où le modèle apprend par coeur

Nous n'avons pas réussi à faire émerger un phénomène d'apprentissage d'un pattern au cours de nos entraînements. Il nous semble que le problème vient du traitement de nos données. Comme nous le décrivons dans la section suivante, des architectures plus simples, avec lesquelles il est plus difficile de faire de grosses erreurs d'implémentation, ne montrent aucune preuve qu'un pattern existe dans nos données.

B. Essai de modèles plus simples

Afin de réduire l'incertitude quant aux erreurs potentielles sur l'utilisation des modèles, nous avons étudié l'implémentation d'autres architectures, beaucoup plus simples. En effet, nous avons implémenté des MLP (Multi Layer Perceptron) et des réseaux convolutifs sur le jeu de données FEIS. Encore une fois, nous n'avons pas observé d'autre tendance que l'apprentissage par cœur ou pas d'apprentissage du tout, quelque soit le nombre de couches et les hyper-paramètres.

Finalement, nous pensons que la source de nos problèmes vient probablement du traitement de nos données. La méthode de wavelet scattering transformation nous était inconnue avant cette étude et il est probable que nous n'ayons pas bien réalisé son implémentation malgré toutes nos précautions. Cependant, il est aussi possible que la tâche réalisée dans [1] se reproduise mal sur notre dataset. En effet, une des remarques faites par les auteurs de la méthode est que les mots considérés dans leur cas activent d'autres régions du cerveau que dans le cas de prononciation de son. Leur nombre de canaux est réduit et certaines électrodes ciblent des parties du cortex traitant la motricité. De plus, les mots qu'ils considèrent sont très différents les uns des autres, là où les 16 phonèmes de notre dataset se ressemblent grandement parfois. Ainsi, nous ne sommes pas certains que la méthode soit généralisable, que ce soit l'approche avec un LSTM, ou le pré-traitement des données avec la wavelet scattering transformation.

Remarque: Nous avons également essayé de brancher des LSTM, CNN et MLP directement sur les signaux EEG normalisés, sans appliquer de wavelet transformation. Nous avons essayé cette approche dans l'espoir d'observer un phénomène d'apprentissage (même si non performant) car nous conservions l'aspect temporel des données. Nous supposons que conserver la notion de temporalité aiderait le LSTM à déceler un pattern. Ici aussi cependant, nous ne sommes pas parvenus à obtenir un phénomène d'apprentissage satisfaisant.

VI. ETUDE D'UN PROBLÈME PLUS SIMPLE

La tâche à laquelle nous nous sommes précédemment intéressés n'est pas simple et malgré tous nos efforts, nous n'avons pas réussi à généraliser l'architecture décrite dans [1] sur le dataset FEIS à 21 sujets et 16 classes. Cependant, loin de se démotiver, nous avons décidé de nous intéresser au même problème sur un autre dataset moins complexe. A vrai dire, le dataset HablaImaginada tel que décrit dans notre déclaration de projet initiale était celui que nous ambitionnions d'utiliser dans le cadre de ce projet.

Il a été publié dans [3] en 2017. Toutefois, malgré toutes nos demandes pour accéder au dataset original et en dépit de 3 tentatives de prise de contact depuis le mois d'Octobre avec différents contributeurs l'ayant utilisé, il ne nous a pas été possible d'accéder à l'intégralité du dataset d'origine. La seule ressource que nous avons trouvée à propos de ce dataset est en libre accès dans le cadre d'une compétition Kaggle disponible à l'adresse suivante: <https://www.kaggle.com/competitions/habla-imaginada-clasificacion-senales-EEG/overview>

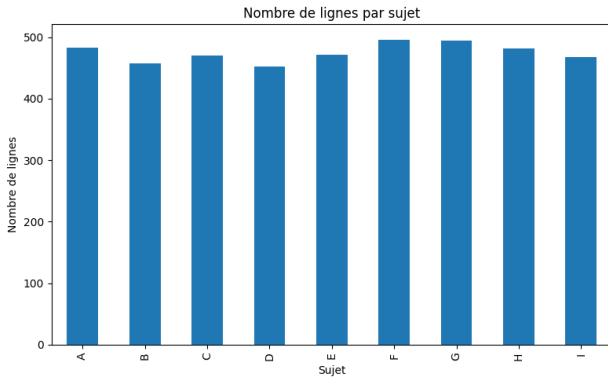


Fig. 10: Nombre d'échantillons par sujet : on constate une répartition plutôt homogène

A. Données

Si l'on considère à la fois les voyelles pensées et les commandes, le dataset d'origine HablaImaginada comprend un total de 9314 enregistrements (4201 pour les voyelles et 5113 pour les commandes). Chaque enregistrement dure 4 secondes, ce qui représente 37 256 secondes (10,3 heures) de signaux EEG sur 6 canaux. La base est structurée en 15 dossiers et 47 fichiers, avec un poids total de 2,48 Go.

La compétition Kaggle et par conséquent notre étude en utilisant une version simplifiée qui ne fait pas la distinction au détail entre les 11 classes d'origine mais seulement entre deux classes "Voyelle" ou "Commande". On peut donc résumer notre nouvelle tâche à un problème de classification binaire consistant à déterminer à partir des signaux EEG si la pensée correspond à une voyelle ou à un ordre. Enfin notre dataset contient 4272 lignes d'enregistrements réparties en 9 sujets. Chacune de ces lignes comporte pour chacun des 6 canaux 4096 valeurs.

Remarque importante: Etant donné que ce dataset provient d'une ancienne compétition Kaggle, nous n'avons pas accès aux annotations du testset. Nous avons ainsi décidé de découper les données d'entraînement en trois datasets : train/val/test afin de mettre en place un protocole d'étude rigoureux.

B. Preprocessing

Exactement comme nous l'avons réalisé pour le dataset FEIS, après visualisation des données, nous filtrons et normalisons les données.

Reamarque: Dans [3], les auteurs indiquent avoir filtré les signaux entre 2 et 40 Hz après acquisition pour supprimer les artefacts donc il est normal que notre filtre ne change pas l'allure du signal. Nous procédons donc à la normalisation puis aux transformations en ondelettes comme précédemment avec quelques nuances que nous introduisons dans le paragraphe suivant.

C. Décomposition en ondelettes de Daubechies

Contrairement à la décomposition de scattering vue précédemment, la décomposition en ondelettes repose sur le choix d'une ondelette unique, ici la Daubechies d'ordre

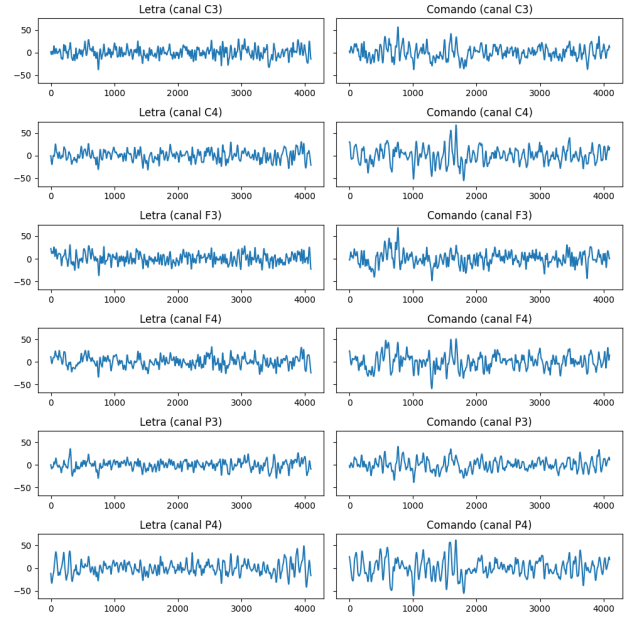


Fig. 11: Visualisation d'échantillons correspondant à des lettres et des ordres sur différents canaux

4 (db4), car chaque ondelette possède des propriétés spécifiques qui influencent l'analyse du signal.

L'ondelette db4 est souvent privilégiée pour les signaux physiologiques en raison de sa bonne capacité à capturer les transitions et les variations rapides, tout en minimisant la distorsion. Cette approche est celle privilégiée par les auteurs du dataset dans [3].

Le processus de décomposition fonctionne par un filtrage en banc de filtres : à chaque niveau, le signal est séparé en une composante d'approximation (basses fréquences) et une composante de détail (hautes fréquences) via des filtres passe-bas et passe-haut. Ensuite, seule la partie d'approximation est retenue et décomposée à nouveau au niveau suivant. Ce procédé permet d'obtenir une représentation multi-résolution du signal, facilitant l'analyse des tendances globales ainsi que des détails fins, tout en réduisant la redondance des données.

D. Décomposition en ondelettes stationnaires

On se propose ici d'opter pour une décomposition mono-ondelette pour s'en tenir à un preprocessing simple. La transformée en ondelettes stationnaires (SWT) décompose un signal en plusieurs niveaux de fréquence sans altérer sa dimension temporelle, ce qui permet de conserver l'ordre chronologique des informations. Cette propriété est particulièrement utile pour entraîner des modèles séquentiels comme les LSTM, car elle offre une représentation riche en détails à différentes échelles tout en préservant la structure temporelle du signal. Parmi ses avantages, la SWT permet de capturer simultanément des caractéristiques globales et locales, améliorant ainsi la capacité du LSTM à modéliser des dynamiques complexes. Cependant, cet enrichissement de l'information se traduit par une augmentation du nombre de features, ce qui peut accroître la complexité du modèle et le risque de surapprentissage, en plus d'impliquer un coût computationnel plus élevé.

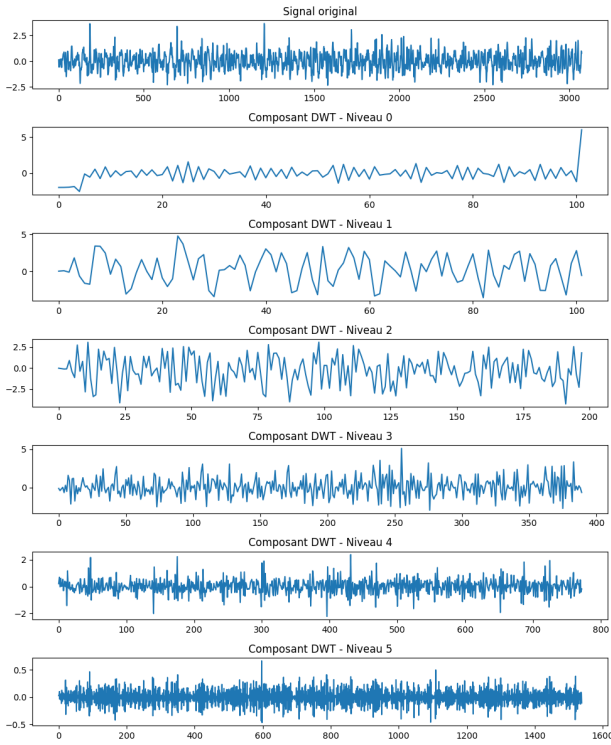


Fig. 12: Exemple de décomposition DWT sur l'un des signaux filtrés et normalisés

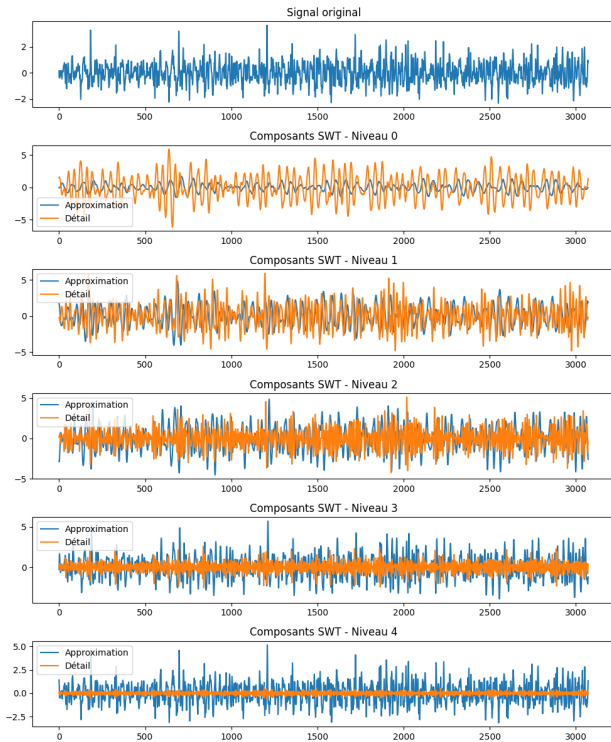


Fig. 13: Pour chaque niveau de la décomposition SWT, deux courbes : L'approximation (cA), qui représente la composante basse fréquence du signal, c'est-à-dire une version lissée qui capture la tendance générale et les caractéristiques globales du signal. Le détail (cD), qui contient les informations sur les variations rapides et les hautes fréquences.

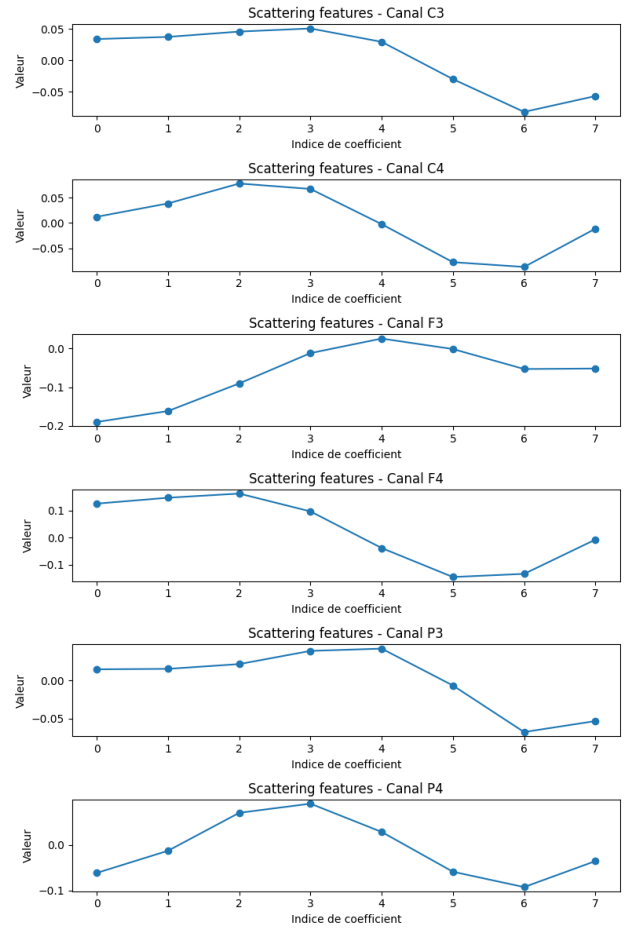


Fig. 14: 8 premiers coefficients de scattering avec moyennage temporel

E. Décomposition par wavelet scattering (avec moyennage temporel)

Enfin de la même manière que dans l'étude menée sur la dataset FEIS, on réalise la transformée en ondelettes du signal mais cette fois-ci avec moyennage temporel. En effet, ici notre objectif n'est plus d'entraîner un LSTM prédéfini dans un autre papier, mais de construire from scratch des architectures de réseaux de neurones pour apprendre sur ces données. Sans chercher à conserver la dimension temporelle du signal, on s'intéresse plutôt à la valeur des coefficients dans cette partie. A titre indicatif on propose une visualisation des 8 premiers sans forcément y chercher un sens physique.

F. Modèles

Dans cette étude, trois architectures ont été implémentées pour la classification des signaux.

Le modèle MLP, entraîné sur des caractéristiques issues du wavlet scattering ou la DWT, reçoit en entrée des tenseurs aplatis et comporte plusieurs couches entièrement connectées avec activations ReLU et dropout pour la régularisation. Un protocole de recherche d'hyperparamètres rigoureux, basé sur une recherche par grille et un mécanisme d'early stopping, permet d'optimiser la configuration (dimensions et nombre de

couches, taux de dropout, taux d'apprentissage, taille de batch, learning rate).

Le réseau 1D CNN, quant à lui, exploite des caractéristiques issues d'une transformation par SWT ou DWT. Il intègre plusieurs blocs convolutionnels composés de couches de convolution 1D, activations ReLU et opérations de max pooling, suivis de couches fully connected pour la phase de classification. La recherche d'hyperparamètres porte ici sur le nombre de filtres, les tailles de noyaux, la dimension des couches fully connected, ainsi que sur les taux de dropout, d'apprentissage et la taille de batch.

Le MLP offre l'avantage d'une implémentation simple et d'un entraînement rapide sur des données prétraitées, bien qu'il ne capture pas explicitement les dépendances temporelles. En revanche, le 1D CNN, malgré une complexité computationnelle supérieure et une sensibilité accrue aux réglages, permet d'extraire efficacement les motifs locaux et temporels des signaux.

Enfin, le LSTM tire profit de la décomposition SWT car celle-ci conserve la structure temporelle du signal tout en fournissant des informations à différentes échelles. L'approximation capture la tendance globale du signal, tandis que les détails révèlent les fluctuations fines. Cette représentation hiérarchisée enrichit l'entrée du LSTM, lui permettant de mieux apprendre et modéliser les dépendances temporelles complexes.

Pour chacune des trois implémentations, la forme des inputs a été adaptée au modèle et à la nature de la décomposition en ondelettes. Le détail est disponible et commenté dans le code.

On se propose par ailleurs de mesurer l'influence du nombre de sujets sur l'accuracy. Pour ce faire on compare les performances du meilleur modèle déterminé à chaque étape sur 1,3,5,7 et 9 sujets à l'issue des grid search.

G. Résultats

Nombre de sujets	MLP	MLP on DWT	1D CNN	1D CNN on DWT	LSTM on SWT
1	81%	75%	73%	63%	74%
3	75%	72%	67%	68%	62%
5	69%	71%	63%	71%	69%
7	66%	70%	63%	71%	70%
9	59%	71%	64%	72%	67%

TABLE I: Accuracy du meilleur modèle sur le test set en fonction du nombre de sujets pour différentes architectures.

Quelques remarques: On observe des tendances différentes. Globalement, il semble qu'avec le scattering avec moyenne temporelle, le CNN et le MLP perdent en précision plus on augmente le nombre de sujets. En revanche, il semble qu'avec l'information fournie par les ondelettes de Daubechies, la précision ne se dégrade que peu pour le MLP voire s'améliore pour le 1D CNN qui tire partie du nombre croissant de données au fur et à mesure qu'on intègre tous les participants. Enfin, bien qu'il n'apprenne rien ni avec le scattering moyenné ni avec la DWT, le LSTM a été cette fois ci capable d'apprendre sur les données ayant subi la SWT, ce qui nous conforte concernant le fait que cette transformation préserve la richesse temporelle du signal.

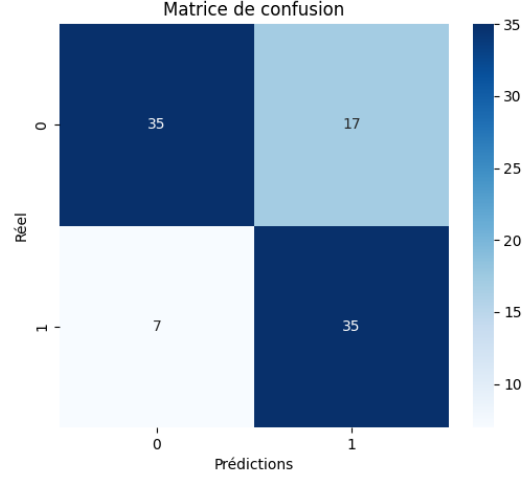


Fig. 15: Exemple de matrice de confusion en test sur 1 sujet

Par ailleurs, ces résultats ne sont certes pas au niveau de l'état de l'art de 2025 qui score à près de 90/100 sur des tâches similaires, mais restent bien supérieurs à ceux des premiers modèles de machine learning qui dépassaient à peine la prédiction au hasard [3]

VII. CONCLUSION

Ce travail de recherche sur la classification des signaux EEG nous a appris beaucoup sur la façon de les appréhender. Il nous a permis de développer les bons réflexes de preprocessing et de prendre en main des outils clés tels que la transformée en ondelettes et ses variantes. Même si nous n'avons pas réussi à identifier quelle étape de la chaîne d'apprentissage nous bloquait dans l'obtention d'un modèle efficace et que nous ne sommes pas en mesure de juger de la capacité de généralisation de l'architecture et méthode de pré-traitement présentées dans [1], nous avons su rebondir en testant plusieurs architectures sur une tâche plus abordable. Bien que nos résultats ne renversent pas l'état de l'art, nous sommes parvenus à nous approprier le problème dans jusqu'à la mise au point de nos propres architectures. Par ailleurs, nous mettons en relief à travers ce travail la complexité liée à la variabilité des signaux EEG inter sujets et l'avantage des transformées en ondelettes de Daubechies et stationnaires pour répondre à cette problématique. Enfin, pour plus d'études, nous pourrions envisager la mise en place d'autres architectures de DeepLearning sur FEIS. Pour répondre à notre difficulté initiale, il serait possible d'entreprendre l'entraînement d'un LSTM à partir de transformées en ondelettes autres que la scattering transform que nous avons employée en première approche.

REFERENCES

- [1] Mokhles Abdulghani, Wilbur Walters, and Khalid Abed. "Imagined Speech Classification Using EEG and Deep Learning". In: *Bioengineering* 10 (May 2023). DOI: [10.3390/bioengineering10060649](https://doi.org/10.3390/bioengineering10060649).

- [2] Mokhles M. Abdulghani, Wilbur L. Walters, and Khalid H. Abed. “EEG Classifier Using Wavelet Scattering Transform-Based Features and Deep Learning for Wheelchair Steering”. In: *2022 International Conference on Computational Science and Computational Intelligence (CSCI)*. 2022, pp. 401–405. DOI: [10.1109/CSCI58124.2022.00077](https://doi.org/10.1109/CSCI58124.2022.00077).
- [3] Germán Coretto, Iván Gareis, and Hugo Rufiner. “Open access database of EEG signals recorded during imagined speech”. In: Jan. 2017, p. 1016002. DOI: [10.1117/12.2255697](https://doi.org/10.1117/12.2255697).
- [4] Mesfer Al Duhayyim et al. “Intelligent Machine Learning Based EEG Signal Classification Model”. In: *Computers, Materials and Continua* 71.1 (2021), pp. 1821–1835. ISSN: 1546-2218. DOI: <https://doi.org/10.32604/cmc.2022.021119>. URL: <https://www.sciencedirect.com/science/article/pii/S1546221821000084>.
- [5] J. Clayton S. Wellington. *Fourteen-channel EEG with Imagined Speech (FEIS) dataset*. Tech. rep. University of Edinburgh, Edinburgh, UK, 2019. DOI: [doi:10.5281/zenodo.3369178](https://doi.org/10.5281/zenodo.3369178).