



Institut Supérieur de l'Aéronautique et de l'Espace

S U P A E R O

PIE NIA03 - RAPPORT FINAL

**EEG-STROKE : DÉTECTION DE L'INTENTION DE
MOUVEMENT DANS LE SIGNAL EEG POUR LE CHU DE
TOULOUSE.**

*APPENZELLER Brice, DARIO Mathieu, DELAVANDE
Julien, DENIAU Aurélien, GOMEL Jules, IGBIDA Rayanne*

10 mars 2024

Table des matières

1	Introduction	2
1.1	Contexte	2
1.2	Objectifs	2
1.3	Etat de l'art	3
2	Gestion de projet	8
2.1	Planning du projet	8
2.2	Organisation dans l'équipe	9
2.3	Organisation avec le client	10
3	Méthodes	11
3.1	Recherche et développement (R&D)	11
3.2	Ingénierie des données	22
3.3	Développement d'application	30
4	Résultats	32
4.1	La pipeline finale	32
4.2	L'application	36
5	Discussion	38
5.1	Points forts de notre travail	38
5.2	Limites de notre solution actuelle	38
6	La suite de ce projet	40
6.1	Processus expérimental	40
6.2	Algorithmes	41
6.3	Intention de mouvement	41
6.4	Gestion de projet	41
7	Conclusion de ce PIE	42
7.1	Bilan sur la solution	42
7.2	Bilan sur la gestion du projet	42
8	Annexes	44
8.1	Organisation du dossier GitHub	44
8.2	Structure et format des données	45
8.3	Utilisation de l'application	46

1 Introduction

1.1 Contexte

L'Accident Vasculaire Cérébral (AVC) représente la première cause de handicap acquis chez l'adulte. À la suite d'un AVC, plusieurs perspectives sont possibles selon la gravité et l'évolution des séquelles. Une des séquelles les plus importantes chez les patients est l'hémiplégie, caractérisée par une paralysie ou une perte de mobilité partielle ne touchant que la moitié du corps. La rééducation pour limiter l'impact de cette séquelle est encore aujourd'hui un enjeu majeur.

Une perspective porteuse d'espoir dans ce domaine repose sur la plasticité cérébrale et est appelée *neurofeedback*. Ce concept repose sur l'exploitation des signaux cérébraux d'un utilisateur pour délivrer un retour à cet utilisateur sur son état mental ou ses signaux cérébraux : cette méthode offre de nombreuses applications pour la formation de pilotes, le soin de patients avec des syndrômes d'hyperactivités ou dans notre cas, pour la rééducation de patients hémiplégiques. Dans ce dernier cas, cette thérapie exploite les signaux cérébraux produits pour générer un mouvement, qui vont être reconnus pour déclencher un mouvement du côté paralysé - au moyen d'un effecteur comme un exosquelette par exemple - et ainsi travailler la plasticité cérébrale de cette zone dans le but de retrouver une meilleure liberté de mouvement.

Notre projet "**BCI-EEG-STROKE**" s'inscrit dans la continuité de recherches antérieures menées par le Centre Hospitalier Universitaire (CHU) de Toulouse, notre commanditaire, sur la rééducation et le suivi des patients post-AVC. Ce projet vise à être appliqué dans le domaine de la recherche, explorant une utilisation concrète du neurofeedback dans la rééducation des patients post-AVC. Dans ce contexte, le patient est encouragé à exprimer l'intention de déplacer son bras du côté affecté. Cette intention de mouvement, détectée par un algorithme, déclenche l'activation d'un effecteur pour concrétiser le mouvement. Le retour d'information reçu par le patient est ainsi la manifestation physique du mouvement, résultant de la génération d'une intention de mouvement par son cerveau. En particulier, du côté lésé, cela vise à stimuler la plasticité cérébrale, pour engendrer ainsi l'amélioration des fonctions motrices du patient.

Les années précédentes, dans le cadre des PIE, un prototype de détection de l'intention du mouvement dans les signaux électriques cérébraux, ou électroencéphalogramme (EEG), a été produit. Il détectait l'intention de mouvement avec une précision que le client estimait améliorable. Ce programme fonctionne hors-ligne et le client souhaite d'abord une amélioration de la précision de la détection avant d'entamer tout développement de la partie temps réel de la détection de l'intention du mouvement.

1.2 Objectifs

L'objectif principal de notre projet est **la recherche et le développement d'un algorithme de détection de l'intention du mouvement** à partir de signaux EEG présentant une précision satisfaisant notre client, dépassant celle de l'année dernière. Cette détection doit s'effectuer en se reposant uniquement sur les signaux provenant du côté du cerveau où est apparu l'AVC, appelé le côté lésé, et sur les mouvements d'extension du coude du bras parétique afin de satisfaire le paradigme de neurofeedback dans lequel

notre projet s'inscrit.

Le second objectif consiste à mettre en place des **algorithmes pour le traitement des données, l'entraînement et l'évaluation de modèles de manière standardisée et rigoureuse**. Ainsi, tous les traitements devront être réalisés de façon reproductible, documentée et rigoureuse afin de garantir une démarche de recherche sérieuse. Il nous est possible de réutiliser les travaux de l'année précédente, mais nous n'avons pas d'obligation de ce côté.

Enfin, le dernier objectif est **l'intégration de notre algorithme de détection dans une interface homme-machine (IHM)**. Le but est de réaliser une application de bureau autonome, pour les ordinateurs du CHU, lesquels sont sous `Windows` et ne possèdent pas d'interpréteur `Python`. Une application similaire adaptée à la solution du PIE précédent ayant déjà été développée, nous pourrions au choix, intégrer nos algorithmes dans l'application du PIE précédent si sa structure et sa modularité le permettent ou la refondre entièrement. Ainsi, une attention particulière sera portée à la convivialité et à la facilité d'utilisation de cette interface. De même, la clarté, les explications et l'optimisation de notre code devront être particulièrement soignées afin de faciliter son utilisation par de futurs étudiants, des chercheurs et le personnel médical.

1.3 Etat de l'art

Dans le cadre de notre PIE qui est particulièrement axé sur la recherche, une étude approfondie de l'état de l'art semble capitale.

Notre PIE se situant au carrefour entre neurosciences, médical, ingénierie et science des données, une première approche des concepts fondamentaux nous permet d'avoir un meilleur aperçu des limitations et standards actuels dans ces différents domaines, en particulier pour ce qui est en lien avec les interfaces cerveau-machine et leur utilisation pour la rééducation post-AVC.

Les **interfaces cerveau-machine** (ICM ou Brain Computer Interface, BCI en anglais), et plus spécifiquement les interfaces basées sur l'électroencéphalographie (EEG), constituent une avancée significative en neuroingénierie, offrant des perspectives prometteuses dans un large nombre de domaines, dont la rééducation post-AVC.

Le neurofeedback représente une approche innovante et prometteuse pour la rééducation post-AVC. Cette méthode repose sur la capacité du cerveau à s'auto-réguler en utilisant des techniques de biofeedback qui permettent aux patients de visualiser et de modifier leurs propres activités cérébrales en temps réel. Pour les individus ayant subi un AVC, le neurofeedback offre la possibilité de cibler spécifiquement les régions du cerveau touchées par l'accident vasculaire cérébral, facilitant ainsi la récupération des fonctions altérées telles que la motricité, le langage ou la cognition. En permettant aux patients de prendre conscience et de contrôler leur activité cérébrale, le neurofeedback favorise la plasticité neuronale et la réorganisation des réseaux neuronaux endommagés, ce qui peut conduire à une amélioration significative des capacités fonctionnelles et à une meilleure qualité de vie après un AVC.

Concernant les algorithmes de traitement des données, il est important de souligner l'importance de la revue de M. Rashid et al [10] de 2020 sur l'état actuel, les challenges et

les possibilités offertes par les BCI utilisant l'EEG et la revue de F. Lotte de 2018 [7] sur les algorithmes de classification utilisés en BCI avec EEG qui nous ont largement servi pour la réalisation de l'état de l'art suivant.

Pré-traitement de données

Les signaux recueillis sont des séries temporelles propres à chaque électrode. Bien que les signaux présentent une similarité importante entre eux - surtout entre les électrodes adjacentes, chaque signal représente une somme pondérée de l'activité cérébrale totale. Ces signaux sont susceptibles d'être perturbés par divers bruits tels que le clignement des yeux (particulièrement visible sur les électrodes frontales), le serrage de mâchoire (qui impacte un large spectre de fréquences), les mouvements corporels (fréquences hautes), et les interférences électriques, notamment le courant alternatif à 50 ou 60 Hz.

Il existe une grande variété de solutions pour éliminer ces artefacts et les bruits dans les données, mais il n'existe pas de standard dans ce domaine. Parmi les techniques les plus couramment utilisées, on trouve le **filtrage temporel**, qui permet de supprimer les artefacts en se basant sur les variations temporelles des signaux [8]. Le filtre temporel le plus utilisé est le filtrage passe-bande, permettant à la fois de supprimer les artefacts liés aux mouvements parasites, ainsi que les bruits à basse fréquence. Ce filtre peut entraîner une perte d'informations importantes.

Le **filtrage spatial**, permettant la séparation des sources à partir des électrodes est une approche majeure pour le débruitage et la reconstruction du signal cérébral réel. Le filtrage spatial dans l'analyse EEG repose sur la notion que certaines électrodes sont plus importantes que d'autres en fonction de la zone cérébrale d'intérêt. Par exemple, des stimuli visuels nécessiteront une attention particulière aux électrodes occipitales, tandis que l'activité du cortex moteur sera plus pertinente dans la région supérieure de la tête. Cependant, il est complexe de prédire exactement où positionner les électrodes en raison des diffractions dans le cerveau liées aux différentes couches : l'approche est donc algorithmique.

L'**analyse en composante principale** (PCA) est un algorithme clé de reconstruction de sources. En assignant des poids spécifiques à chaque électrode, on extrait des combinaisons linéaires des signaux des électrodes, appelées les *sources*. La PCA vise à maximiser la variance entre ces sources pour distinguer les combinaisons linéaires les plus importantes de notre ensemble de signaux et ainsi reconstruire l'activité électrique d'une zone spécifique du cerveau à partir des électrodes. Une limitation de la PCA réside dans l'hypothèse que les signaux physiologiques sont orthogonaux, ce qui est une hypothèse non vérifiée en signal EEG.

Pour pallier cette lacune, l'analyse en composantes indépendantes (ICA) est utilisée [6], qui supprime l'hypothèse d'orthogonalité des signaux physiologiques. Au lieu de cela, l'ICA cherche à extraire des composantes qui sont statistiquement indépendantes, ce qui permet de mieux capturer les structures complexes et non linéaires présentes dans les signaux EEG. L'ICA est devenue une méthode privilégiée dans le domaine de la neuroimagerie pour extraire des informations significatives à partir des signaux enregistrés par les électrodes.

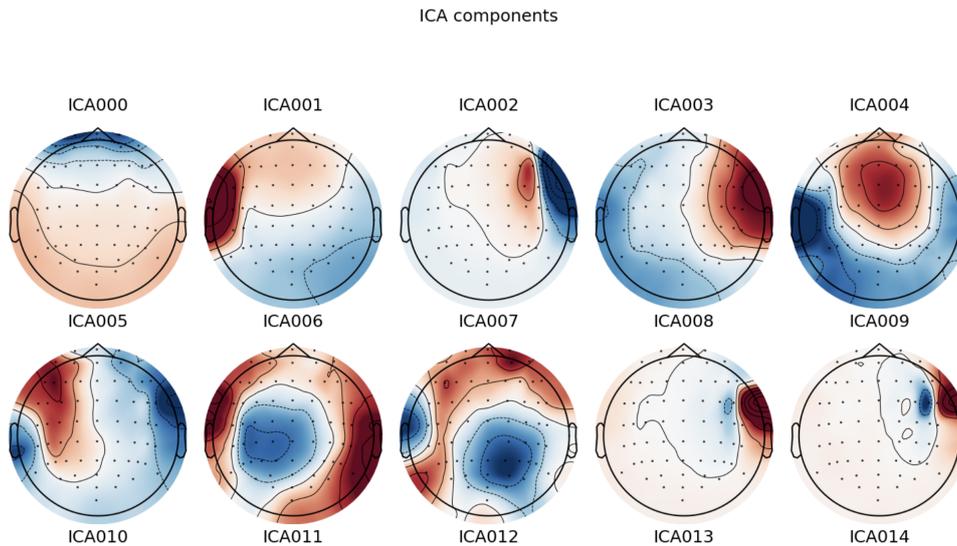


FIGURE 1 – Exemple d’ICA faite sur les données de l’un des patients

xDawn est un autre outil pour le filtrage spatial, efficace et dynamique [11]. Cette méthode vise à atténuer les interférences et à améliorer la séparabilité des signaux en adaptant dynamiquement les filtres spatiaux en fonction des caractéristiques des données. Concrètement, xDawn identifie les composantes du signal les plus pertinentes pour une tâche donnée et les pondère afin de minimiser l’impact du bruit et des artefacts, améliorant ainsi la qualité de l’analyse ultérieure.

Extraction de caractéristiques

Il existe deux types principaux de caractéristiques (ou *features*) dans l’analyse des signaux cérébraux les caractéristiques temporelles et fréquentielles.

Les caractéristiques temporelles consistent en la concaténation des signaux provenant de toutes les électrodes, ce qui est particulièrement bénéfique pour les Potentiels Évoqués (en anglais, *Event-Related Potentials* ou ERP), permettant de capturer des événements provoqués par des stimuli spécifiques.

Les caractéristiques fréquentielles se concentrent sur des mesures telles que la puissance de bande, offrant une perspective sur l’activité oscillatoire du cerveau. Cette approche est particulièrement adaptée à la détection des variations de fréquence associées à des processus mentaux spécifiques, tels que ceux observés dans l’imagerie mentale.

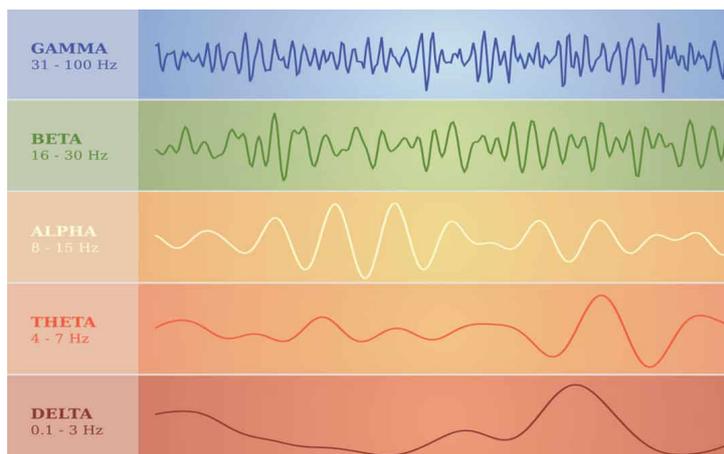


FIGURE 2 – Les différentes bandes de fréquence

Ces caractéristiques peuvent également être représentées sous forme de matrices ou de tenseurs, offrant ainsi des structures adaptées à des analyses plus complexes. Cependant, la combinaison des caractéristiques temporelles et fréquentielles peut conduire à une augmentation de la dimensionnalité des données, ce qui peut rendre l'analyse plus complexe [5]. Face à cela, l'utilisation d'une étape de sélection de caractéristiques devient pertinente. Cette démarche vise à identifier les caractéristiques les plus pertinentes, permettant ainsi de réduire la complexité des données sans compromettre la qualité de l'analyse. Une sélection judicieuse des caractéristiques peut faciliter la modélisation tout en préservant les informations essentielles contenues dans les signaux cérébraux.

En complément des caractéristiques temporelles et fréquentielles traditionnellement extraites, plusieurs autres techniques d'extraction de caractéristiques se distinguent. Parmi celles-ci, la méthode des Common Spatial Patterns (CSP), la décomposition en ondelettes (ou wavelets), et l'analyse de la covariance dans l'espace tangentiel (aussi appelée approche Riemannienne) sont notables. Ces approches offrent des perspectives variées pour la représentation des signaux cérébraux, permettant ainsi une exploration approfondie des informations contenues dans ces signaux.

La méthode **Common Spatial Patterns** (CSP) est reconnue pour sa capacité à améliorer la séparabilité des signaux EEG, en maximisant la variance des signaux associés à une activité cérébrale spécifique tout en minimisant la variance des signaux liés à d'autres activités. Cette approche est efficace pour les tâches impliquant la discrimination entre différents états mentaux ou différentes activités motrices. En exploitant ces propriétés, la méthode CSP contribue significativement à l'extraction d'informations pertinentes à partir des signaux EEG.

La **décomposition en ondelettes** se distingue par sa capacité à offrir une analyse multirésolution des signaux EEG, permettant de capturer des informations à la fois fréquentielles et temporelles. Cette technique s'avère essentielle pour examiner la dynamique des signaux cérébraux à différentes échelles, facilitant ainsi la détection de motifs subtils associés à des fonctions ou des réponses cérébrales spécifiques.

L'**analyse de covariance dans l'espace tangentiel**, inspirée de la géométrie Riemannienne, offre une perspective unique sur la structure des signaux EEG. En considérant les signaux comme des points dans un espace géométrique, cette méthode permet d'exploiter la structure intrinsèque des données pour une caractérisation plus précise et robuste

des états cérébraux. Cette approche se révèle particulièrement adaptée pour traiter la variabilité naturelle des signaux EEG et pour améliorer la classification dans les systèmes BCI, facilitant la séparation et la classification des différents états mentaux ou activités cérébrales.

Classification

L'analyse des signaux cérébraux fait appel à une gamme variée de méthodes, chacune offrant ses propres avantages et inconvénients. On peut distinguer à la manière de [7] : la classification adaptative, les approches supervisées et l'apprentissage profond.

La **classification adaptative** se décline en approches supervisées et non supervisées, offrant ainsi une flexibilité dans le traitement des données [7]. Il est important de noter que les méthodes non supervisées peuvent parfois souffrir d'un manque de robustesse. En parallèle, bien que les approches adaptatives offrent une certaine flexibilité, elles peuvent se révéler inadaptées aux interfaces cerveau-ordinateur (BCI) lorsque le feedback varie.

Les **approches supervisées**, telles que les Support Vector Machines (SVM), les forêts aléatoires, l'analyse discriminante linéaire (LDA), la régression logistique et les méthodes de boosting, offrent des cadres bien établis pour la classification basée sur des caractéristiques extraites préalablement. Ces algorithmes de classification supervisée sont largement reconnus pour leur efficacité dans la distinction des différents états ou réponses cérébrales à partir de signaux EEG. Ils sont adaptés pour traiter des ensembles de données étiquetés, où les classes des échantillons sont connues à l'avance, et sont souvent utilisés dans le domaine des interfaces cerveau-ordinateur (BCI) pour des applications telles que la détection de l'intention de mouvement ou la classification d'états mentaux. Le défi de ses approches réside dans l'étiquetage des jeux de données physiologiques.

Enfin, l'**apprentissage profond** ou *deep learning* émerge comme une approche prometteuse, offrant la capacité d'apprendre des représentations de haut niveau à partir des données brutes. Toutefois, les algorithmes de Deep Learning nécessitent généralement une grande quantité de données pour converger vers une précision intéressante. En 2018, malgré son potentiel, le Deep Learning n'avait pas encore été pleinement exploré ni testé dans le domaine de l'analyse des signaux cérébraux. Cela souligne la nécessité de mener davantage de recherches pour comprendre pleinement ses avantages et ses limitations dans ce contexte spécifique.

2 Gestion de projet

Cette partie est développée plus largement dans notre plan de développement, qui est un autre livrable du projet.

2.1 Planning du projet

L'organisation de notre projet a été soutenue par un macro-planning articulé autour du diagramme de Gantt présenté en figure (3). Ce diagramme nous a servi de référence pour planifier et suivre l'avancement de nos tâches de manière régulière. Nous avons régulièrement comparé la progression effective de nos travaux avec les échéances théoriques prévues sur le diagramme de Gantt initial.

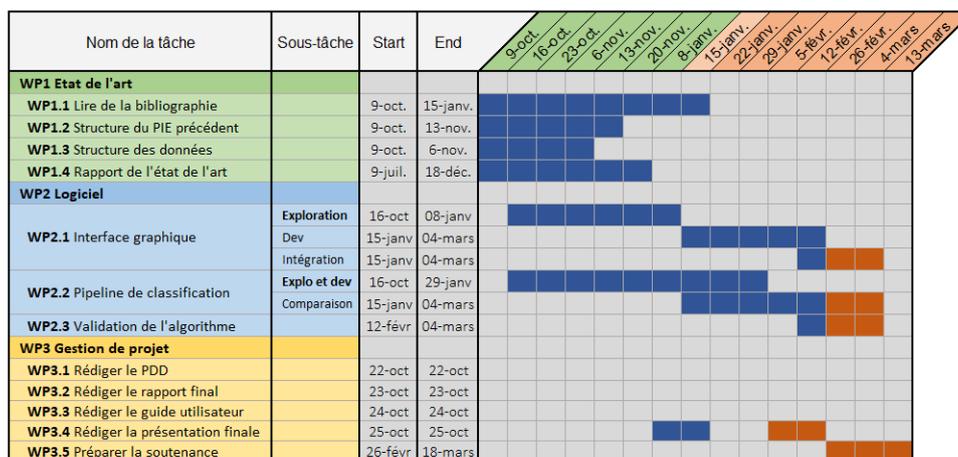


FIGURE 3 – Structure du diagramme de Gantt (image datant du 15 février)

La méthode s'est avérée efficace, car elle nous a offert une visibilité constante sur les différentes échéances du projet. En nous référant au diagramme de Gantt, nous avons pu identifier rapidement les retards éventuels, ajuster nos priorités et prendre des mesures correctives le cas échéant. Cela a contribué à maintenir le projet sur la bonne voie, en assurant une gestion proactive du calendrier et en évitant les éventuels dépassements de délais. La structure du diagramme de Gantt, dont la figure (3) montre l'état datant du 15 février, a servi de cadre solide pour la planification et l'exécution des tâches. En suivant cette approche, notre équipe a pu maintenir un rythme de travail régulier, anticiper les éventuels obstacles et garantir une progression cohérente du projet. Ce processus itératif de suivi a favorisé une meilleure gestion du temps et des ressources, renforçant ainsi notre capacité à respecter les délais et à atteindre les objectifs fixés.

Si notre macro-planning suit plutôt bien nos avancées réelles, nous avons dû augmenter nos efforts sur les derniers mois. Le manque de ressource du PIE précédent nous a pénalisé au début et la compréhension du problème a rendu les premiers mois laborieux. Ce retard est aussi dû au manque de connaissances sur le domaine car les cours de traitement d'apprentissage automatique et de traitement EEG arrivent plus tardivement. Enfin, comme expliqué plus largement par la suite, la réorganisation de notre travail au cours du projet nous a permis de devenir plus efficaces dans la seconde moitié du déroulé du projet.

2.2 Organisation dans l'équipe

Répartition des tâches

Pour garantir une répartition efficace des tâches au sein de l'équipe, nous avons utilisé un tableau Jira en complément du suivi global du projet avec le diagramme de Gantt. Ce tableau nous a offert une visibilité détaillée sur les différentes activités à réaliser, les responsabilités attribuées et l'état d'avancement de chaque tâche. Le tableau Jira a servi de tableau de bord centralisé en plus du diagramme de Gantt, permettant à chaque membre de l'équipe de visualiser clairement les tickets assignés et les actions à entreprendre. Cette approche a facilité la collaboration en assurant une compréhension commune des objectifs individuels et collectifs. Ainsi, chacun s'est vu attribuer des tâches en fonction de ses préférences et de ses compétences afin d'améliorer l'efficacité du travail et ses tâches se sont vues modifiées en cas de besoin.

Travail conjoint

L'aspect du travail conjoint au sein de notre équipe a été facilité par l'utilisation de divers outils collaboratifs.

Pour la partie **gestion de projet**, nous avons opté pour la rédaction des fichiers partagés sur Overleaf, permettant à chaque membre de contribuer simultanément au document, en distanciel et à tout moment. Couplée au tableau Jira, cette approche collaborative a favorisé une répartition efficace des tâches et une fusion efficace des contributions de chacun dans les différents livrables.

Parallèlement, pour la partie **développement**, nous avons choisi d'utiliser un dossier **GitHub**. Cette plate-forme nous a permis de programmer de manière simultanée sur des branches distinctes tout en partageant une base commune. La gestion des versions sur GitHub a facilité le suivi des modifications, la résolution des conflits et la fusion des avancées significatives dans le code. Cette méthodologie de travail parallèle a grandement optimisé notre processus de développement, en permettant une collaboration en temps réel tout en préservant l'intégrité du code.

Communication dans l'équipe

Lors de chaque séance de PIE, nous avons instauré des points réguliers au cours desquels chaque membre de l'équipe a eu l'occasion de présenter ses avancées individuelles. Ensuite, nous réalisons une évaluation collective de l'avancement global du projet afin de comparer à l'avancement théorique et ainsi prendre des contre-mesures si nécessaire. Nous procédions à une revue des tâches futures attribuées à chacun, permettant ainsi d'apporter d'éventuelles modifications pour maintenir un niveau optimal d'efficacité et d'agilité, tant pour le projet dans son ensemble que pour chaque participant tout au long de son déroulement.

Cette approche a été complétée par notre facilité à nous voir au quotidien et par des logiciels de messagerie instantanée pour nous organiser rapidement.

2.3 Organisation avec le client

Avec notre client, le CHU de Toulouse représenté par le docteur David Gasq, nous avons planifié à l'avance des réunions rapides permettant de faire le point de manière succincte sur notre avancée, nos questionnements et obtenir ses retours. Nous avons aussi effectué quelques entrevues de manière physique au CHU, ce qui nous a permis d'entretenir un meilleur contact et une meilleure compréhension du sujet.

Cette organisation méthodique et à l'avance nous a permis de maintenir une bonne communication tout au long du projet avec le client. Nous regrettons n'avoir pas eu du temps pour plus d'entrevue sur site, limitée par l'emploi du temps de chacun.

3 Méthodes

Notre projet s’articule autour de trois axes de travail interconnectés et réalisés en parallèle : la recherche et développement (R&D), l’ingénierie des données et le développement d’application. Ces trois aspects sont détaillés ci-dessous.

3.1 Recherche et développement (R&D)

Le secteur de R&D s’est focalisé autour de trois axes principaux : la compréhension approfondie et la visualisation des données EEG, le développement d’algorithmes pour la préparation, l’extraction de caractéristiques, et la classification des signaux EEG, et enfin, l’optimisation des hyperparamètres pour les algorithmes les plus prometteurs. Ces domaines ont interagi de manière itérative, constituant un cycle dynamique de recherche et d’application. Le processus de R&D a engendré une construction méthodique et itérative de procédures pour l’extraction et le traitement des données, l’entraînement et le test des pipelines, ainsi que le développement de techniques de visualisation adaptées. Ces procédures seront détaillées dans la section 3.2.

Compréhension des données et visualisation

La première étape de notre démarche R&D a été de s’immerger dans les données EEG à travers une série de techniques d’extraction, de labélisation, de prétraitement, et d’epoching, utilisant notamment la bibliothèque MNE pour la manipulation et l’analyse des signaux. Cette exploration initiale nous a permis de comprendre les nuances et la complexité des données, facilitant l’identification des défis spécifiques à leur traitement. Par ailleurs, le développement d’outils de visualisation a joué un rôle crucial dans cette phase, offrant des insights visuels directs sur les caractéristiques et comportements des signaux, ainsi que sur l’efficacité des méthodes de pré-traitement appliquées. Ces visualisations ont été essentielles pour guider les décisions de conception des étapes suivantes du processus de R&D, permettant une évaluation rapide et intuitive des stratégies d’extraction de caractéristiques et de classification. Comme expliqué précédemment, les procédures d’extraction, labélisation et pré-traitement retenues sont explicitées dans la section 3.2.

Recherche des meilleurs algorithmes de prédiction

La seconde phase de notre recherche s’est articulée autour de la définition de protocoles de tests précis (voir section 3.2) et de l’exploration d’algorithmes de prédiction adaptés aux spécificités des données EEG en imagerie motrice (MI). Cette exploration a inclus l’examen de diverses méthodologies mentionnées dans la littérature scientifique, visant à traiter, analyser et classifier efficacement les signaux EEG, particulièrement dans le contexte de l’imagerie motrice qui implique la projection mentale de mouvements.

Notre attention s’est initialement portée sur la capacité à prédire des mouvements à partir des signaux EEG de patients en phase de rééducation post-AVC, une suggestion de notre référent école. Les défis inhérents à l’analyse de données EEG de patients ayant subi un AVC distinguent notre travail des études préexistantes focalisées sur des sujets

sains. En effet, la prédiction de mouvement dans ce contexte ajoute une couche de complexité, étant donné les potentiels dommages cérébraux affectant la génération de signaux EEG. Cet axe de recherche nous a permis, dans une première approche, de démontrer la faisabilité de classer ces signaux particuliers.

Nous avons ainsi choisi de tester diverses méthodes de pré-traitement des données extraites de l'état de l'art avec différentes méthodes de classification, pour la plupart issues de la librairie `sklearn`. En effet, nous avons constaté dans la littérature que les caractéristiques (aussi appelées *features* en anglais) extraites des signaux EEG et les modèles sont étroitement liés. Il convient donc de visualiser l'algorithme de traitement de données - pré-traitement, extraction de features et modèle de classification - dans son entièreté, ce qui est appelé une **pipeline** de traitement de données.

La multiplicité des pipelines proposées dans la littérature nous a permis d'évaluer différentes précisions, constituant ainsi une sorte de benchmark de ces méthodes. Le tableau ci-dessous, récapitule l'ensemble des pipelines de traitement des données qui ont été explorées lors de ce benchmark.

Pipeline					Refs
CSP				+ <i>classifier</i>	[1, 4, 9]
xDAWN	+	Covariance	+	Normalisation + <i>classifier</i>	[11]
Covariance	+	Tangent Space	+	Normalisation + <i>classifier</i>	[3, 2, 12]

TABLE 1 – Récapitulatif des pipelines de traitement de données explorées, avec leurs références bibliographiques Les classifieurs sont choisis parmi *Random Forest*, *SVM*, *LDA* et *Logistic Regression*.

Les développements des explorations menées sur les différentes méthodes mentionnées ci-dessus sont disponibles dans les sections suivantes. Nous avons décidé d'effectuer un pré-traitement commun à toutes les *pipelines* afin de standardiser le filtrage temporel des données, la sélection de la plage de fréquence, l'extraction des *epochs* et la sélection des électrodes. Les explorations effectuées portent donc sur méthodes de filtrage spatial et celles de classification.

Les évaluations des différentes pipelines se font par *cross-validation* en 4 plis (4-fold) sur 4 configurations de tests possibles (tests A, B, C ou X) qui sont détaillés ci-après en section 3.2. Pour chaque évaluation, la configuration de pré-traitement utilisée (partie commune à toutes les pipelines) est décrite ci-après (résultats recalculés après optimisation de ces hyperparamètres, cf section 20). Cela comprend notamment les paramètres de pré-traitement temporel explicités ci-dessus, en plus des paramètres spécifiques à la méthode choisie.

```
EPOCHS_TMIN: -0.75 s
EPOCHS_TMAX: +0.95 s (EPOCHS_LENGTH = 1.7 s)
EPOCHS_EMPTY_FROM_MVT_TMINS: -6 s
FMIN: 9 Hz
FMAX: 30 Hz
```

Voici une description détaillée de chacun traitements explorés :

— Pipeline : filtrage spatial avec CSP

La résolution de problèmes liés à la classification des signaux EEG dans le contexte de l'imagerie motrice, avec une quantité limitée de données, peut bénéficier de l'application de techniques de filtrage spatial. Parmi ces méthodes, le *Common Spatial Patterns* (ou **CSP**) est particulièrement utilisée pour extraire des caractéristiques en maximisant la discriminabilité de deux classes.

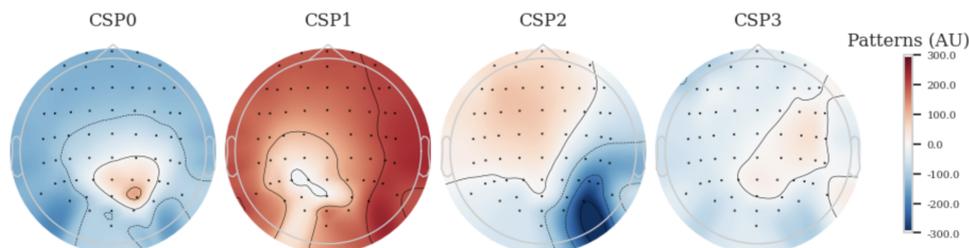


FIGURE 4 – Exemple de CSP faite sur des données publiques

Cette approche est particulièrement pertinente dans le cadre du projet car le déchiffrement des intentions motrices du cerveau à partir des signaux EEG est un point fort des CSP. Les CSP sont des filtres spatiaux qui maximisent la variance entre deux classes tout en minimisant la variance intra-classe, permettant ainsi de mettre en évidence les motifs spatiaux les plus discriminants.

Dans le cas spécifique de l'imagerie motrice avec des données EEG limitées, les CSP peuvent aider à extraire des caractéristiques pertinentes en identifiant les *patterns* cérébraux associés à différentes tâches motrices. Cette méthode est particulièrement utile lorsque les enregistrements EEG sont sujets à des variations interindividuelles. En optimisant la séparation spatiale entre les classes d'intérêt, les CSP peuvent améliorer la robustesse du modèle de classification, même avec des jeux de données restreints. De plus, cette approche peut contribuer à atténuer les effets du bruit et des artefacts, renforçant ainsi la capacité du modèle à généraliser à de nouveaux enregistrements.

Les CSP sont définis mathématiquement en tant que combinaison linéaire des signaux EEG. En conséquence, toutes les caractéristiques extraites sont initialement dérivées des signaux EEG, chaque signal étant assigné à un poids optimisé. Ces poids sont ajustés de manière à ce que les caractéristiques retenues maximisent la variance interclasse contenue dans les signaux EEG d'origine. Cette approche présente l'avantage de préserver pratiquement toute l'information contenue dans les signaux, tout en les organisant de manière à améliorer les performances des algorithmes de classification. Parallèlement, elle permet d'éliminer les éléments superflus.

Pour illustrer l'effet de la CSP sur des données EEG dans le cadre d'une classification, voici la répartition des données d'une session dans "l'espace des CSP" :

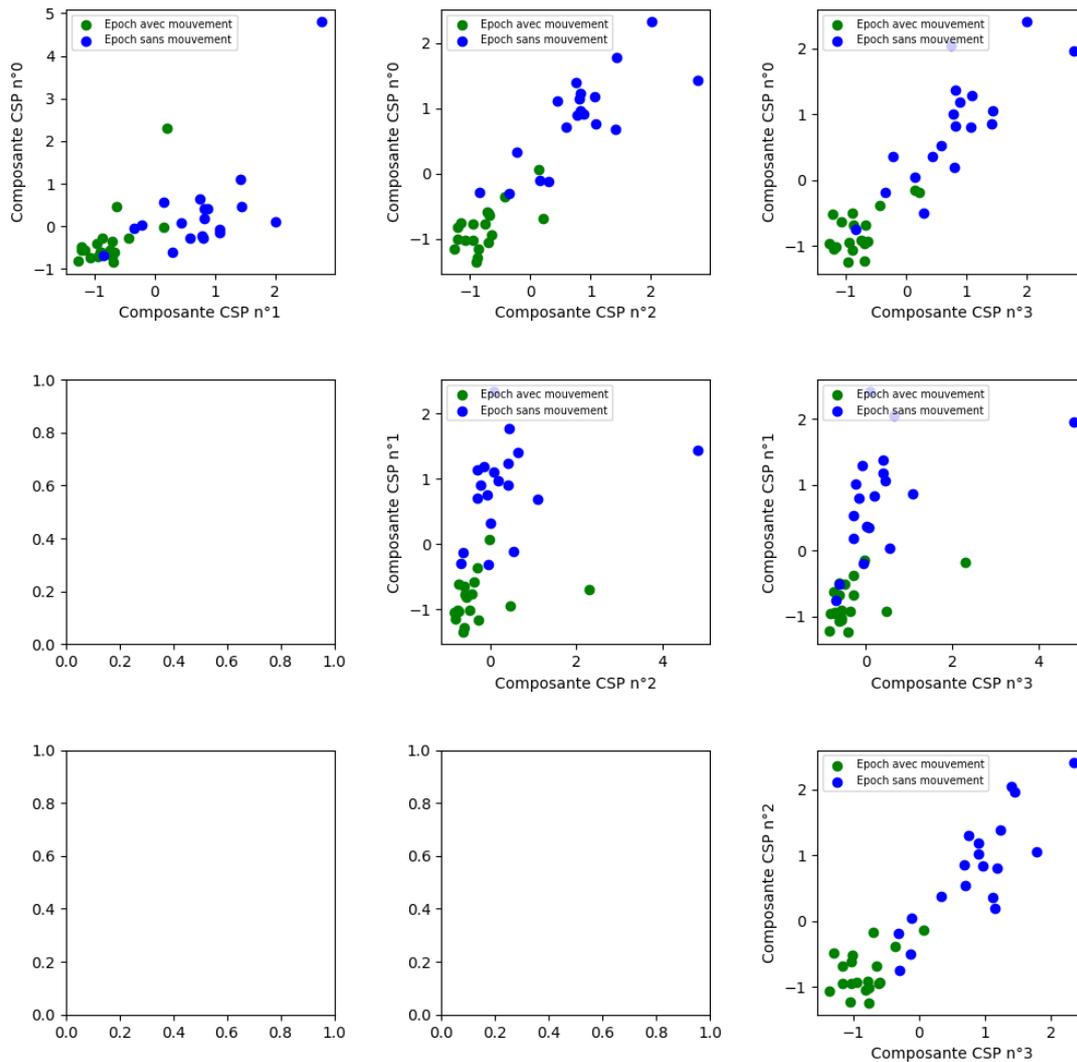


FIGURE 5 – Projection des classes dans l'espace des CSP sur une session

Cet exemple montre bien l'utilité de la CSP. Les deux classes mouvement et non-mouvement sont bien discriminées dans l'espace des CSP, ce qui permet de faciliter la classification. Les résultats de cette pipeline étudiée durant la R&D sont sur la table 2.

— Pipeline : filtrage spatial avec xDAWN

Le filtrage spatial avec xDAWN est une autre approche importante dans le traitement des signaux EEG pour l'imagerie motrice. Contrairement aux CSP, qui se concentrent sur la maximisation de la variance interclasse, xDAWN vise à minimiser la variance totale tout en conservant les informations pertinentes pour la tâche de classification. Cette méthode est particulièrement utile lorsque les signaux EEG sont contaminés par différents types de bruit et d'artefacts, car elle permet de réduire leur impact tout en préservant les caractéristiques discriminantes des données.

Le filtrage xDAWN fonctionne en identifiant les directions spatiales dans lesquelles la variance est maximale pour les différentes classes d'intérêt. En projetant les données EEG sur ces directions spatiales, xDAWN permet de réduire la dimen-

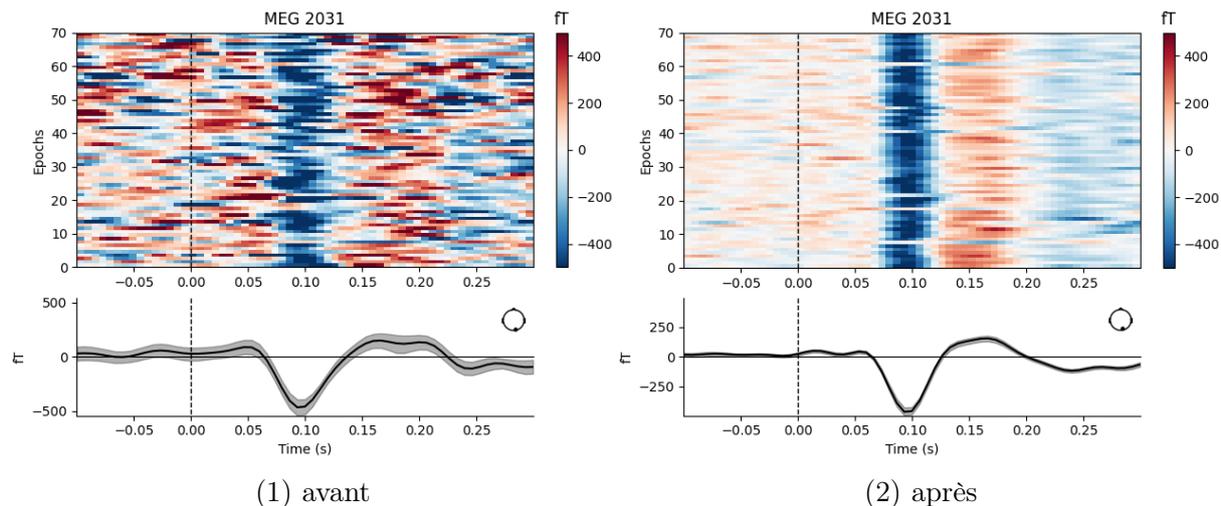


FIGURE 6 – Exemple d’utilisation de l’algorithme xDAWN sur des signaux MEG publiques dans le but de réduire le bruit contenu dans le signal

sionnalité du problème tout en conservant les informations importantes pour la classification. De plus, cette approche est robuste aux variations interindividuelles, ce qui en fait une méthode adaptée pour les applications où les données sont limitées et où la généralisation est cruciale.

Il est cependant nécessaire de noter que l’algorithme xDAWN est principalement utilisé pour les données EEG qui contiennent des potentiels évoqués (potentiellement difficilement adaptable aux problématiques d’imagerie motrice). [11] détaille en particulier le fonctionnement de l’algorithme, qui maximise le rapport signal de potentiel évoqué sur bruit. Au contraire du CSP, xDAWN est une méthode dédiée à la reconnaissance de formes temporelles dans les signaux et non à la puissance spectrale (en CSP). Ces filtres sont notamment appris à l’aide de méthodes de décomposition en valeurs singulières. Les résultats de cette pipeline étudiés durant la R&D sont sur la table 3.

— Pipeline : projection Riemannienne et espace tangent

Les méthodes par projection Riemannienne et utilisation d’espace tangent sont des approches avancées pour le traitement des données EEG en imagerie motrice. Ces méthodes exploitent la structure géométrique des matrices de covariance des signaux EEG pour extraire des caractéristiques discriminantes. En utilisant la géométrie Riemannienne, ces techniques permettent de représenter les données EEG dans un espace tangent où les opérations classiques de machine learning peuvent être appliquées de manière efficace.

La projection Riemannienne consiste à projeter les matrices de covariance des signaux EEG dans un *espace tangent* tangent à l’espace de Hilbert des matrices de covariance. Cette projection a pour but de linéariser l’espace des matrices de covariance, facilitant ainsi l’application de méthodes de classification classiques.

Cette méthode englobe plusieurs étapes, notamment les matrices de covariances, les caractéristiques de l’espace tangent et la normalisation. Chacune de ces étapes apporte une contribution spécifique :

- **Matrices de covariances** : La première étape consiste à calculer la matrice de covariance des signaux EEG. La covariance mesure la façon dont deux variables changent ensemble. Pour les signaux EEG, cela permet de comprendre comment différents capteurs (ou canaux) enregistrent des signaux qui sont corrélés entre eux. Cela est essentiel pour saisir les relations spatiales entre les différents canaux EEG. Il s'agit essentiellement de trouver des modèles dans la façon dont les différentes parties du cerveau émettent des signaux ensemble pendant différents mouvements.
- **Caractéristiques de l'espace tangent** : Après avoir été calculées, les matrices de covariance, sont souvent projetées dans un espace tangent. L'espace tangent est un espace linéaire qui "touche" une surface non linéaire (dans notre cas, l'espace des matrices de covariance) en un point. Cette linéarisation des données de covariance, qui peuvent naturellement être non linéaires et résider sur un collecteur (une surface courbe dans un espace de dimension supérieure), facilite l'application des techniques standard d'apprentissage automatique, qui donnent généralement de meilleurs résultats avec des données linéaires. Cette étape permet ainsi de simplifier et d'aplatir des données complexes et incurvées sous une forme plus compréhensible et plus facile à traiter.

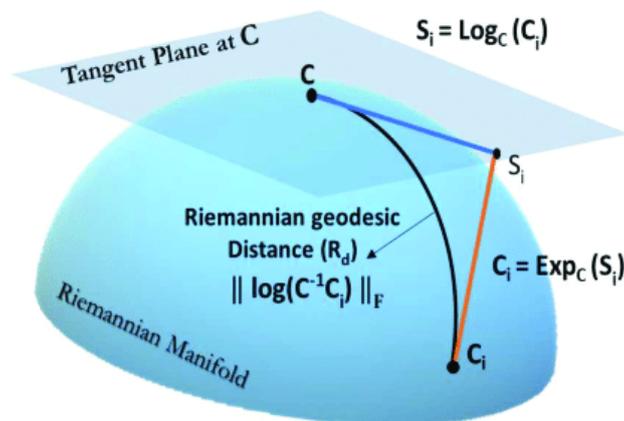


FIGURE 7 – Illustration de l'espace tangent à un point C

- **Normalisation** : La mise à l'échelle veille à ce que toutes les caractéristiques aient la même pondération.

Les résultats de cette pipeline étudiés durant la R&D sont sur la table 4 et cette pipeline est détaillée dans la partie 4.1.

- **Un mot sur les méthodes de *deep learning***

Nous avons également exploré, en début de projet, une stratégie impliquant des méthodes de *deep learning* (notamment réseaux de neurones à convolutions ou *CNN*) pour traiter les données EEG. Bien qu'encore peu validées dans la littérature, ces méthodes semblaient donner de bons résultats dans les articles qui en

faisaient mention. Nous avons cependant constaté que de telles méthodes ne pouvaient s'appliquer à notre problématique, principalement à cause de la quantité trop importante de données qu'elles nécessitent pour s'entraîner et par conséquent de notre jeu de données trop réduit.

Nous nous sommes donc concentrés sur des algorithmes de *machine learning* basés sur des modèles et sur la mise en place d'un pré-traitement efficace des données, dans le but de tirer profit du peu de données disponibles.

Si les résultats finaux de nos algorithmes ont reposé sur notre ingénierie de données et sont décrits dans la partie 4, nous avons motivé cette ingénierie de données par des résultats intermédiaires obtenus durant la R&D.

— Résultat de la pipeline avec filtrage spatial CSP

Classifier	Score A	Score B	Score C	Score X
Random Forest	0.722	0.707	0.671	0.730
SVM	0.721	0.716	0.708	0.729
LDA	0.715	0.711	0.699	0.714
Logistic Regression	0.714	0.712	0.699	0.712
Adaboost	0.700	0.711	0.694	0.710

TABLE 2 – Scores de la pipeline pour différents modèles de classification, sur les différents tests.

— Résultats de la pipeline avec filtrage spatial xDawn

Classifier	Score A	Score B	Score C	Score X
Random Forest	0.630	0.624	0.588	0.645
SVM	0.544	0.566	0.561	0.563
LDA	0.584	0.589	0.560	0.588
Logistic Regression	0.582	0.604	0.569	0.608
Adaboost	0.603	0.610	0.578	0.627

TABLE 3 – Scores de la pipeline pour différents modèles de classification, sur les différents tests.

— Les résultats de la pipeline avec le plan tangent sont présentés ci-dessous :

Classifier	Score A	Score B	Score C	Score X
Random Forest	0.803	0.796	0.718	0.812
SVM	0.855	0.840	0.762	0.857
LDA	0.797	0.760	0.723	0.794
Logistic Regression	0.778	0.767	0.701	0.776
Adaboost	0.769	0.744	0.691	0.770

TABLE 4 – Scores de la pipeline pour différents modèles de classification, sur les différents tests.

— Les résultats de la pipeline avec un simple standardiseur et un classifieur sur les données brutes :

Classifier	Score A	Score B	Score C	Score X
Random Forest	0.607	0.618	0.619	0.614
SVM	0.570	0.566	0.569	0.565
LDA	0.492	0.483	0.490	0.493
Logistic Regression	0.493	0.481	0.494	0.494
Adaboost	0.577	0.576	0.586	0.585

TABLE 5 – Scores de la pipeline pour différents modèles de classification, sur les différents tests.

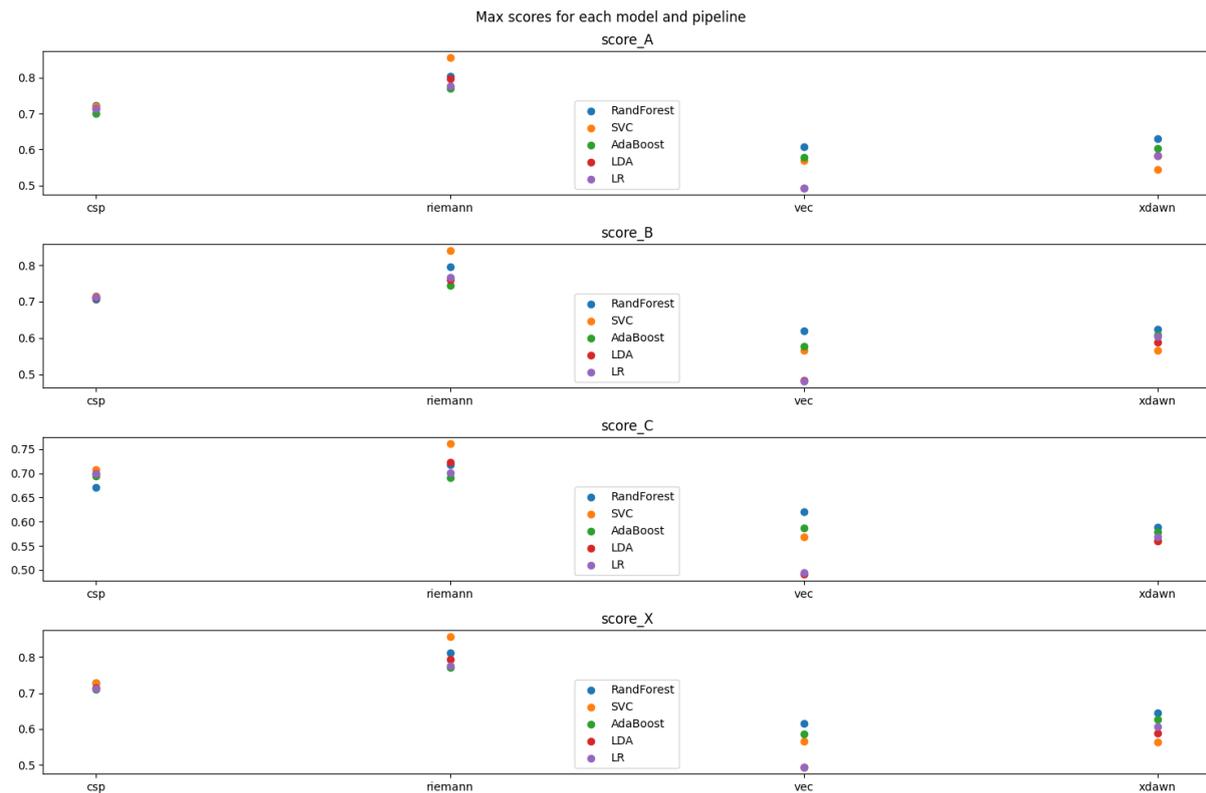


FIGURE 8 – Les différents scores en fonction de la pipeline et du classifieur utilisé

Ces résultats nous montrent la supériorité de pipeline reposant sur le plan tangent, compte tenus des différents scores. Les classifieurs étudiés offrent différentes performances, qui devront être quantifiées plus en détails lors de notre ingénierie de données.

Optimisation des hyperparamètres

Enfin, nous avons réalisé une recherche d'hyperparamètres de prétraitement sur la pipeline prometteuse (projection Riemannienne et espace tangent).

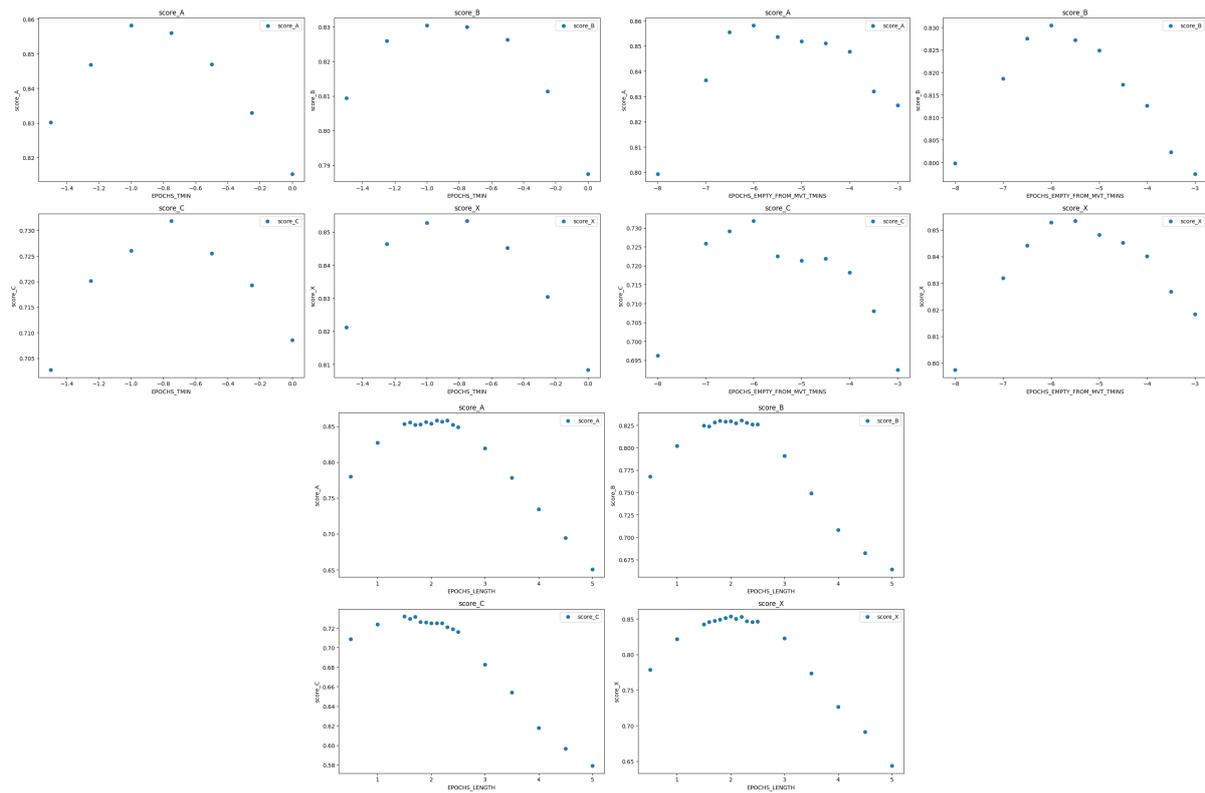


FIGURE 9 – Résultats de la recherche d’hyperparamètre - Différents scores de précision en fonction de la en fonction du début et longueur de l’époque, intervalle de temps entre une époque de non mouvement et une de mouvement

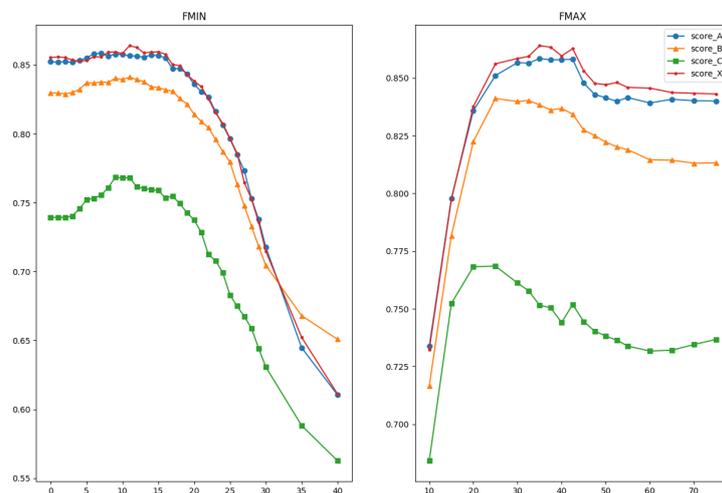


FIGURE 10 – Résultats de la recherche d’hyperparamètres - Différents scores de précision en fonction de la fréquences de filtrage temporel

En prenant comme critère de sélection la moyenne sur les les scores A, B et C décrits dans 3.2, nous retenons preprocessing suivant :

EPOCHS_TMIN: -0.75 s
 EPOCHS_TMAX: $+0.95$ s (EPOCHS_LENGTH = 1.7 s)

EPOCHS_EMPTY_FROM_MVT_TMINS: -6 s
 FMIN: 9 Hz
 FMAX: 30 Hz

Ce même pré-processing (optimisé) a été utilisé pour tester les différentes pipelines comme expliqué dans la sous partie précédente. Cependant, le pré-processing ayant été fait sur la pipeline riemannienne, on peut noter qu'il n'est peut être pas exactement adapté aux autres pipelines testées.

Recherche sur l'intention de mouvement

Notre recherche s'est étendue à la prédiction de l'intention de mouvement, nécessitant la définition d'une nouvelle catégorie d'époques, en plus des catégories existantes de mouvement et de non-mouvement. Cette nouvelle catégorie, que nous appelons intention de mouvement, représente un défi majeur en termes de labellisation et de positionnement temporel. ce laps de temps est décrit dans la littérature, chez des patients sains, allant de 500 ms à 2 s avant le mouvement. Il est important de noter que chez les patients ayant subi un AVC, ce laps de temps pourrait être nettement plus long.

Nous avons adapté notre pipeline riemannien pour intégrer cette catégorie d'intention de mouvement, chargeant ainsi le classifieur de distinguer entre intention de mouvement, mouvement effectif et absence de mouvement. Malgré une recherche approfondie d'hyperparamètres de preprocessing adaptés à cette tâche, les résultats obtenus jusqu'à présent ont été modestes, reflétant la complexité de prédire précisément l'intention de mouvement à partir des signaux EEG, surtout dans le contexte de pathologies neurologiques.

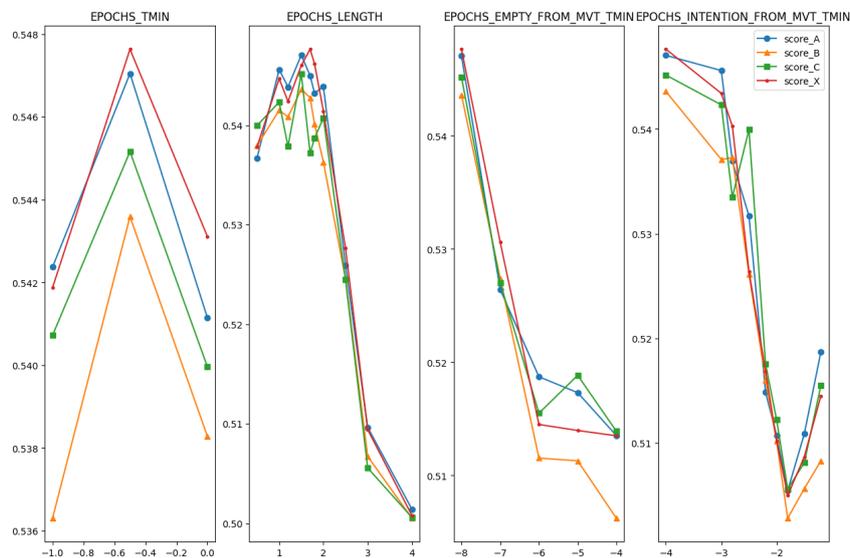


FIGURE 11 – Scores en fonction des début des différentes époques et de la durée de cette dernière

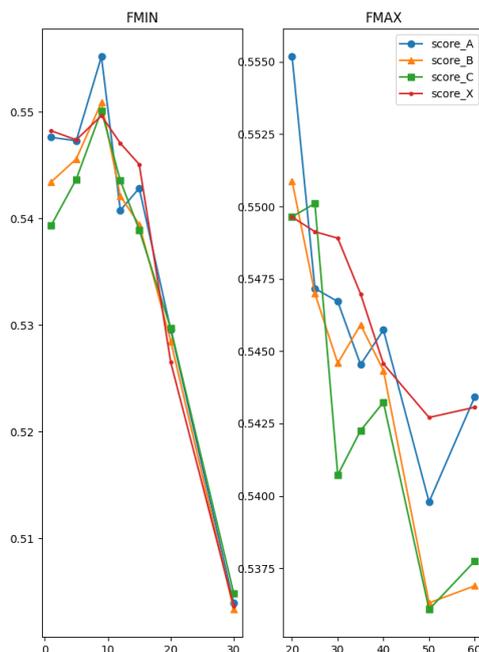


FIGURE 12 – Scores en fonction des fréquences de filtrage

Classifieur	Score A	Score B	Score C	Score X
SVM	0.555	0.550	0.549	0.549

TABLE 6 – Scores de la pipeline Riemann SVM sur l'intention de mouvement pour les différents tests.

Ainsi les meilleurs hyperparamètres trouvés sont les suivants :

EPOCHS_TMIN: -0.5 s
 EPOCHS_TMAX: +0.85 s (EPOCHS_LENGTH = 1.35 s)
 EPOCHS_EMPTY_FROM_MVT_TMINS: -8 s
 EPOCHS_INTENTION_FROM_MVT_TMIN: -4 s
 FMIN: 9 Hz
 FMAX: 20 Hz

Il est notable que les époques correspondant à l'intention de mouvement, identifiées suite à l'optimisation par recherche exhaustive (grid search), se situent bien avant le mouvement réel. Cette observation suggère deux hypothèses potentielles : soit les patients présentent une réactivité significativement réduite en raison de leurs lésions cérébrales, soit l'algorithme n'a pas réussi à converger correctement, optimisant potentiellement selon un critère non adéquat.

Par conséquent, nos efforts pour développer un algorithme capable de prédire l'intention de mouvement n'ont pas abouti aux succès escomptés. Cette situation met en lumière les défis inhérents à la prédiction de l'intention de mouvement, notamment dans le contexte des affections neurologiques, et souligne la nécessité d'une réévaluation des méthodes d'optimisation ou des critères de performance utilisés.

3.2 Ingénierie des données

Extraction et préparation des données

Pour l'entièreté du développement de notre solution, nous avons utilisé des données issues de manipulations de différentes études réalisées par le CHU. Ces expérimentations impliquaient des séries de flexions et extensions du coude par des patients post-AVC. Les données recueillies et utilisées sont les enregistrements bruts des signaux EEG (64 électrodes en configuration 10-20 cf figure (13)) et des signaux cinématiques (capteurs de vitesse et accélération du coude).

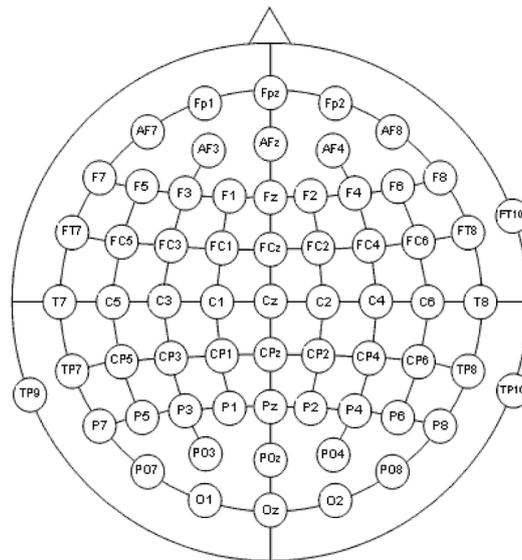


FIGURE 13 – Schéma représentant la position des 64 électrodes en configuration 10-20

La base de données mise à disposition contient les enregistrements de 42 patients, avec entre deux et trois sessions par patient. Au total, 89 fichiers au format `.npy` sont fournis (le groupe du PIE précédent s'était appliqué à adapter les données `.mat` en format `.npy`). Pour chaque session (correspondant à un fichier) sont disponibles l'enregistrement de la sous-session de mouvements du bras gauche (données EEG, cinématiques, index de mouvement), l'enregistrement de la sous-session de mouvements du bras droit (données EEG, cinématiques, index de mouvement), ainsi qu'un certain nombre de méta-données concernant le protocole et le patient (identifiant session, côté lésé, numéro d'essai, fréquence d'échantillonnage, index patient).

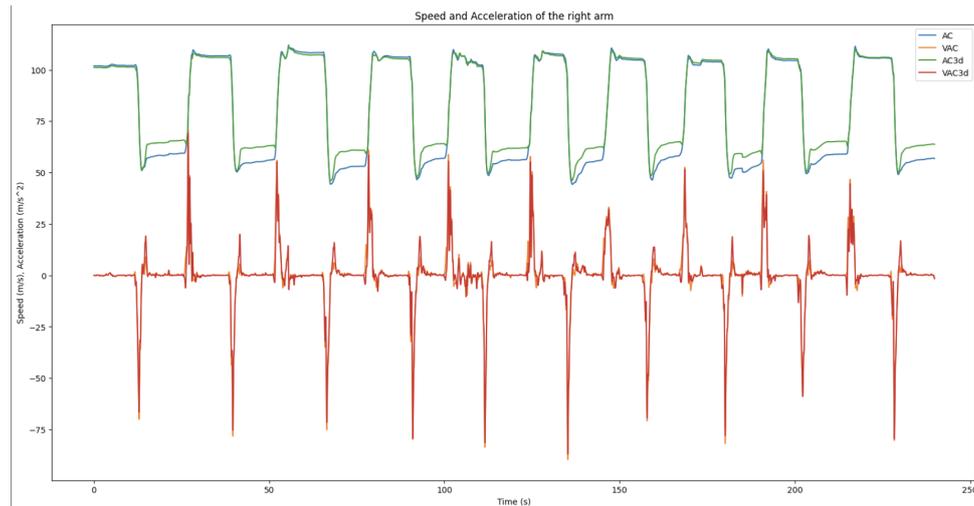


FIGURE 14 – Exemple de données cinématiques dont nous disposons

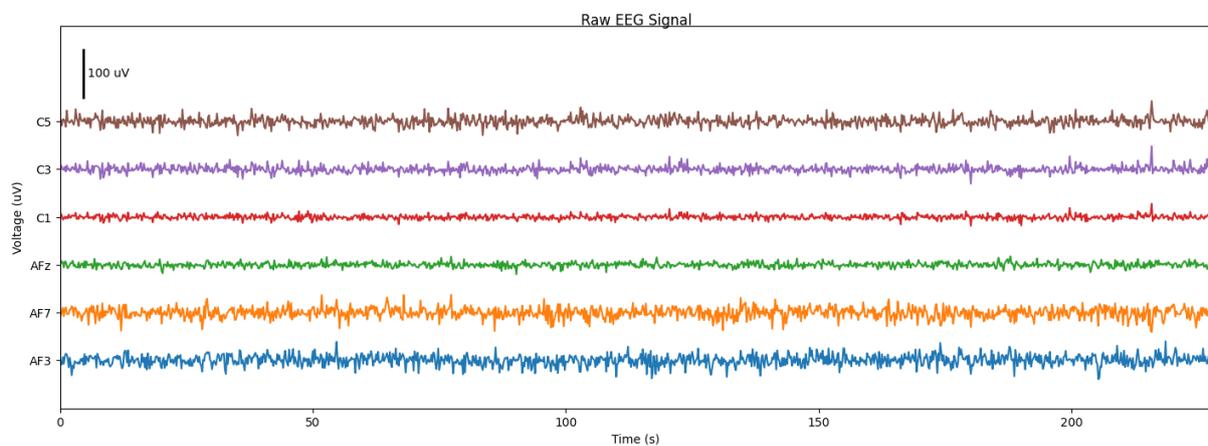


FIGURE 15 – Exemples d'échantillons de données EEG brutes dont nous disposons

Pré-traitement des données

Le pré-traitement des données est une étape clé dans notre implémentation, puisqu'elle permet de préparer les données pour une analyse ultérieure en appliquant une série de techniques et de transformations afin de garantir leur qualité et leur cohérence.

Pour notre projet, nous avons choisi de travailler avec la librairie python `mne`, qui est un package permettant l'exploration, la visualisation et l'analyse de données neurophysiologiques telles que les données provenant d'EEG. Nous avons donc implémenté une classe `DataLoader` qui permet le pré-traitement des données issues de fichiers `.npy` et les transforme en objet `mne.io.RawArray`, ce qui permet une utilisation cohérente de la librairie.

Le pré-traitement des données se décompose en plusieurs étapes. La première consiste au chargement et traitement rudimentaire des données afin d'obtenir un format de données exploitable dans la suite. La seconde consiste à l'application des premiers filtrages temporels et spatiaux sur les données.

La première étape se décompose de la sorte :

1. Les données sont tout d'abord chargées et vérifiées afin de s'assurer de leur intégrité. Nous vérifions notamment la présence de données manquantes, d'éventuelles erreurs de format, ou encore des incohérences dans les méta-données.
2. Une fois chargées et validées, les données sont préparées pour l'analyse. On en extrait et structure les informations pertinentes telles que l'identifiant du patient, le côté parétique, ainsi que la détermination des fréquences des données cinématiques et EEG.
3. Ensuite, les signaux bruts sont traités pour les rendre compatibles avec les exigences de notre projet. Parmi les étapes réalisées, on peut citer :
 - Un ré-échantillonnage des signaux pour harmoniser les fréquences d'acquisition des données cinématiques et de l'EEG. Cela permet une cohérence dans les données et facilite l'analyse conjointe des signaux cinématiques et EEG.
 - Après analyse, on constate que certaines données cinématiques manquent dans les données brutes. Une interpolation est alors faite entre les points des signaux cinématiques.

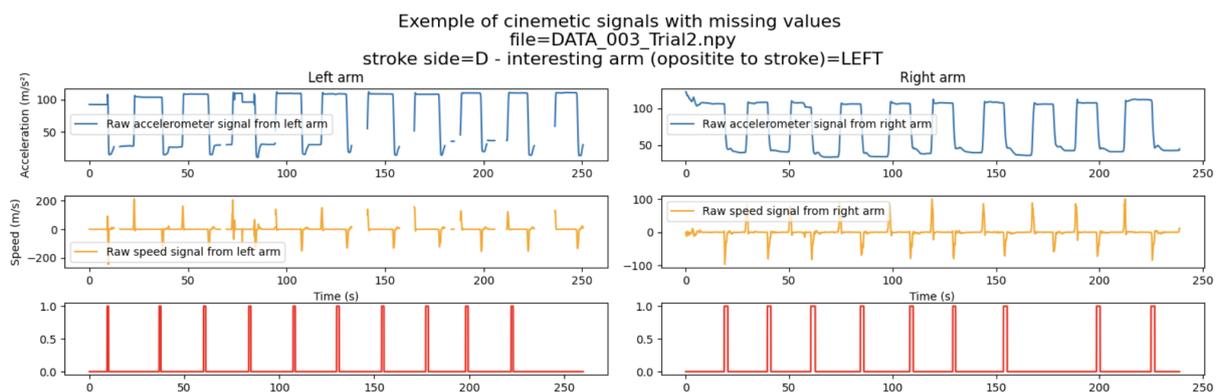


FIGURE 16 – Exemple de données d'accélération manquantes pour le bras gauche

4. Une autre étape importante du pré-traitement des données est la création d'étiquettes sur nos signaux, notamment d'étiquettes sur le mouvement (il n'est pas vraiment possible d'étiqueter de façon déterministe l'intention de mouvement avec les données à disposition). Cela nécessite l'analyse des signaux bruts, comme les données de l'accéléromètre, pour identifier et catégoriser les différents types de mouvements. Cette étape se décompose ainsi :
 - Les étiquettes sont attribuées à partir des données de l'accéléromètre. Une vérification est alors nécessaire afin de s'assurer qu'aucune donnée d'accélération ne manque. Si c'est le cas, les valeurs manquantes sont remplacées par la dernière valeur (non NaN) rencontrée.
 - Le but est ensuite d'assigner à chaque échantillon une étiquette, indiquant s'il s'agit d'un mouvement de flexion, d'extension ou s'il n'y a pas de mouvement. Pour cela, un lissage des données d'accélération est fait en considérant les valeurs supérieures à la moyenne comme des moments de mouvement. La distinction entre mouvement de flexion ou d'extension est ensuite faite en comparant

les valeurs d'accélération lissées aux valeurs originales, afin de mettre en évidence les changements de direction du mouvement.

- Enfin, une vérification est effectuée pour éliminer les étiquettes de mouvement étant trop proches les unes des autres dans une séquence de données (à moins de 5s) puisque le temps moyen entre deux mouvements est d'environ 10s. Cela garantit une séparation adéquate entre les mouvements pour une analyse plus précise et non polluée. Notons que les patients victimes d'AVC ont des difficultés à effectuer un mouvement du bras, rendant le mouvement parfois saccadé, d'où la présence occasionnelle de mouvements très proches dans le temps.

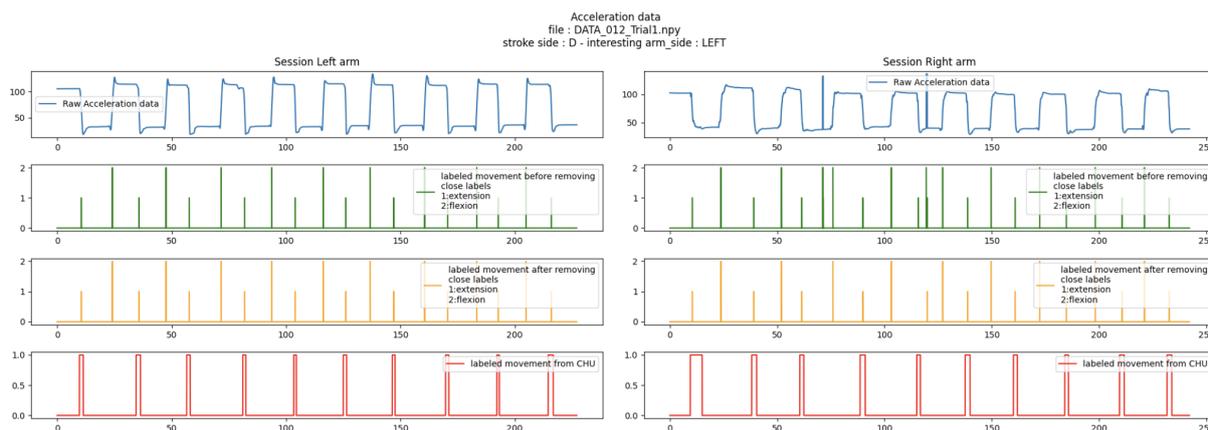


FIGURE 17 – Exemple de nouvel étiquetage des données

Comme indiqué précédemment, les résultats de ce pré-traitement sont stockés dans des objets `mne.io.RawArray` de la librairie `mne`. Ces objets sont particulièrement utiles, car ils offrent une représentation structurée et standardisée des données EEG et cinématiques, facilitant ainsi leur manipulation et analyse ultérieures. Ils permettent également de stocker les étiquettes de mouvement associées avec toutes les autres données dans un seul conteneur, ce qui simplifie leur gestion et leur visualisation. De plus, les fonctionnalités intégrées de la librairie `mne`, telles que les méthodes de filtrage, de découpage en segment, de détection d'événements et de visualisation, peuvent être directement appliquées à ces objets. Tout cela permet une gestion efficace et une analyse cohérente des données EEG et cinématiques.

Après avoir effectué toutes ces opérations de chargement, il est nécessaire de **pré-traiter les données** afin qu'elles soient adaptées à l'entrée des algorithmes de classification. Il s'agit de la seconde partie de traitement :

1. Pour cela, nous utilisons uniquement les sessions enregistrant les mouvements du bras opposé au côté où l'AVC s'est produit, et seules les électrodes situées du côté de l'AVC sont prises en compte. Cela garantit une analyse spécifique des mouvements post-AVC, et permet de se concentrer sur l'utilisation du côté lésé du cerveau.
2. Les données sont ensuite filtrées pour ne conserver que les composantes EEG dans une certaine plage de fréquences. Classiquement, on ne conserve que les fréquences

comprises entre 1 et 40 Hz. Cela permet de se concentrer sur les signaux pertinents, en atténuant les hautes fréquences indésirables, les interférences électriques à 50 Hz, ainsi que les fréquences très basses telles que les dérives DC (tension continue) qui peuvent être présentes. Dans notre cas, la plage de fréquences a été optimisée comme hyperparamètre et les résultats semblent nous diriger vers une plage de fréquences optimale de 9 à 30 Hz.

3. Les données sont finalement segmentées en époques autour du début de chaque mouvement. Elles sont étiquetées comme des périodes d'extension, de flexion ou de repos. Cette segmentation est réalisée grâce à la librairie `mne` pour les périodes de mouvement. Les périodes de non-mouvement sont quant à elles obtenues en prenant des époques commençant 4 secondes avant le début du mouvement, ce qui est possible puisque le temps entre deux mouvements varie entre 5 et 12 secondes comme montré sur la figure 18.

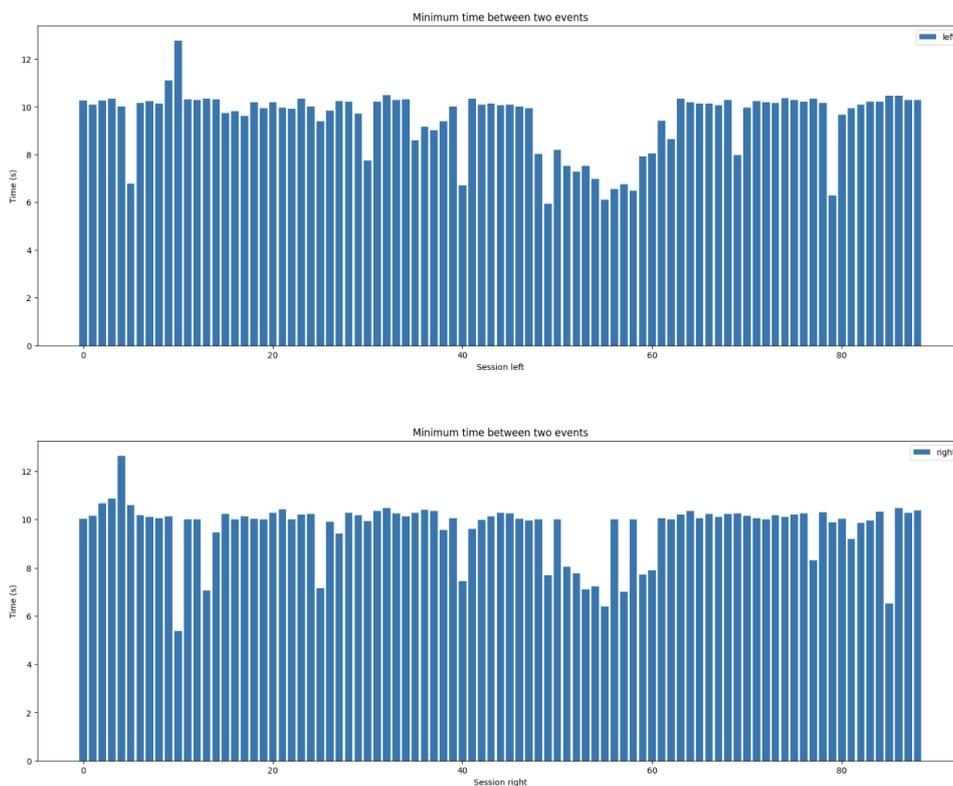


FIGURE 18 – Étude du temps entre deux mouvements

Afin de simplifier l'analyse, un mode de classification binaire est également possible. Dans ce cas, nous ne distinguons que les périodes de mouvement et de non-mouvement : les flexions et extensions sont confondues et sont toutes étiquetées de la même façon.

Développement des métriques

Les méthodes d'évaluation des performances de la solution implémentée se déclinent en deux catégories. Nous avons d'abord mis en place une **évaluation des performances**

sur les données époquées (tranches de signaux autour des mouvements et d'absence de mouvement) selon une procédure en trois étapes. Nous avons également mis en place une **évaluation des performances orientée 'temps-réel'** utilisant des fenêtres glissantes sur l'ensemble des données à tester et qui permet une appréciation visuelle de la classification.

1. Méthode d'évaluation sur données "epoched".

Afin d'obtenir les performances de notre *pipeline* implémentée, nous avons mis en place une procédure de tests en 4 temps, utilisant des données dites 'epoched', c'est-à-dire :

- **Phase de test A** : Tout d'abord, l'idée est de partir de données de sessions de patients et de les répartir en données d'entraînement et de validation, à l'échelle des *epochs* extraites par pré-traitement. L'objectif est donc d'avoir pour chaque session de chaque patient, des *epochs* présentes dans le *training set* et dans le *validation set*. De cette façon, le modèle sera testé sur des données (*epochs*) nouvelles, mais provenant de patients et sessions déjà vues lors de l'entraînement.
- **Phase de test B** : Ensuite, l'objectif est de mesurer la généralisation du modèle à des sessions nouvelles, de patients déjà connus. C'est-à-dire, on entraîne le modèle sur des sessions (toujours découpées en *epochs*) des patients et on teste sur des nouvelles sessions des mêmes patients.
- **Phase de test C** : Ensuite, nous cherchons à mesurer la généralisation à des sessions de patients inconnus, c'est-à-dire que nous entraînons le modèle sur des données de patients (toutes sessions) et testons sur des données de nouveaux patients.
- **Phase de test X** : Enfin, un dernier test a été mis en place pour évaluer les modèles sur l'ensemble des données (patients, sessions) mélangées, dont les *epochs* sont réparties entre *training set* et *testing set* aléatoirement.

L'ensemble de ces tests est réalisé en validation croisée avec un paramètre de reproductibilité, assurant ainsi une évaluation fiable et reproductible des performances. Dans le cadre de nos recherches nous avons fixé le $RANDOM_STATE = 42$ et $N_SPLITS = 4$.

Avec ces 4 tests (A, B, C et X), il nous est possible de caractériser au mieux la performance de la solution. En effet, l'idée finale est d'entraîner le modèle sur l'ensemble des données disponibles afin de l'utiliser dans des conditions réelles où il s'agira de : classifier des sessions de patients déjà connus (test B), ou bien de classifier des sessions de nouveaux patients (test C).

Si nous reprenons les résultats obtenus, il est normal d'obtenir un score A (85.5%) meilleur que le score B (84.0%) qui est lui-même meilleur que le score C (75.8%). En effet, compte-tenu du fait que lors de la phase de test A le modèle ait déjà vu le patient et la session durant l'entraînement, il aura plus de facilité à réaliser une prédiction correcte par rapport à la phase de test B où il a déjà rencontré le patient mais pas sur cette session. Encore plus complexe, lors de la phase de test C, le modèle ne fait face qu'à des patients et des sessions qu'il n'a jamais vus d'où un score moins élevé mais qui traduit la capacité au modèle à généraliser.

L’observation selon laquelle le modèle démontre une amélioration de ses performances lorsqu’il est préalablement entraîné sur les données d’un patient spécifique souligne l’importance cruciale de l’individualisation du processus d’apprentissage automatique dans le domaine médical. Cette spécificité des performances suggère que l’adaptation du modèle à la singularité des données d’un patient particulier favorise une meilleure compréhension des schémas subtils et des variations individuelles. Si la quantité de patients présents dans les données venait à augmenter, on pourrait s’attendre à une meilleure généralisation car le modèle aurait en mémoire une multitude de schémas ce qui finirait à terme par englober la majorité des patrons possibles.

En attendant, la stratégie d’entraîner le modèle après une session de test avec un patient particulier, intégrant ainsi les données de ce patient dans le dataset d’entraînement, représente une solution pragmatique. Cette approche vise à équilibrer la nécessité d’individualisation avec la capacité du modèle à généraliser efficacement (le score du modèle passant de ce fait de 75.8% à 84.0%). Un patient victime d’AVC se faisant accompagner lors de plusieurs séances, cette stratégie ne semble pas irréalisable.

2. Méthode d’évaluation avec fenêtres glissantes

Avec l’objectif de permettre une appréciation des performances de la *pipeline* en fonctionnement réel, nous avons mis au point une méthode utilisant des fenêtres glissantes pour classifier les données de test. L’idée est de faire parcourir une fenêtre le long de l’enregistrement EEG qui constitue les données de test, et à intervalles de temps réguliers d’effectuer une prédiction basée sur les données contenues dans la fenêtre.

Ainsi, chaque instant est contenu dans plusieurs fenêtres ce qui permet d’identifier le nombre de fenêtres qui prédisent un mouvement à cet instant et également le nombre de fenêtres qui prédisent qu’il n’y a pas de mouvement à cet instant.

Cette méthode permet ainsi d’obtenir une fonction de densité de probabilité de prédiction du mouvement sur chaque échantillon des données du signal EEG. En utilisant un seuil judicieusement choisi (laissé en paramètre de la solution) l’utilisateur peut alors obtenir une fonction de décision sur la présence de mouvement ou non. Un exemple de visuel pouvant être obtenu est présenté sur la figure 19.

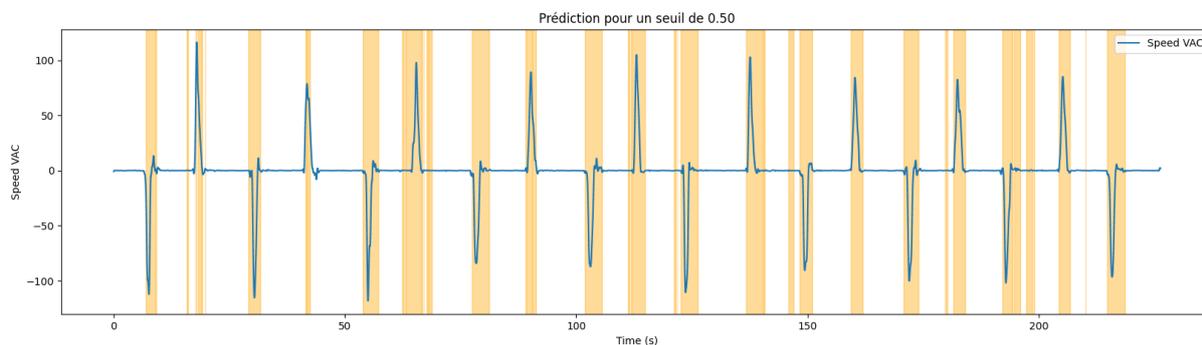


FIGURE 19 – Exemple de prédiction de mouvement avec la méthode des fenêtres glissantes, en considérant un seuil de 0.5. Le signal bleu représente le signal cinématique de la vitesse tandis que les zones colorées (orange) représentent les zones où un mouvement a été prédit.

L’exploitation des prédictions générées par le modèle est simple et efficace. Notre autre méthode d’évaluation se concentrait sur le score, mesurant la capacité du modèle à prédire

la présence ou l'absence de mouvements au niveau d'une époque. La particularité de cette méthode-ci réside dans le fait que chaque instant est évalué à plusieurs reprises. Même si le modèle commet une erreur ponctuelle dans une de ses prédictions, la performance globale du modèle demeure excellente en considérant la moyenne sur l'ensemble des fenêtres évaluées. Cette approche offre une perspective nuancée qui met en lumière la robustesse et la fiabilité du modèle.

Entraînement des modèles et recherche d'hyperparamètres

Face à la complexité des données issues de 89 patients et la nécessité d'un traitement intensif, nous avons optimisé le processus d'entraînement et de recherche d'hyperparamètres en exploitant les serveurs Rainman de notre école, conçus pour gérer des calculs lourds et permettre la parallélisation des tâches. Grâce à l'utilisation de **Slurm** et **Python 3.8.8**, nous avons automatisé et accéléré significativement ce processus, répartissant les charges de travail sur plusieurs nœuds et cœurs du cluster pour le préprocessing et l'entraînement des modèles.

Cette approche a facilité une exploration exhaustive des méthodes d'extraction de caractéristiques et de pré-traitement, en s'inspirant des avancées récentes dans la littérature scientifique. Nous avons ainsi ajusté finement les paramètres des époques (durée, positionnement pour différents types de mouvement) et optimisé les fréquences de filtrage pour améliorer la précision de notre pipeline.

L'exploration de diverses techniques d'extraction de caractéristiques (CSP, Riemann, Xdawn) et de classifieurs (Régression Logistique, SVM, Forêts aléatoires) a permis une évaluation approfondie de leur efficacité dans notre contexte. De plus, nous avons pu tester des hyperparamètres spécifiques à certaines pipelines, comme les paramètres internes au SVM, adaptant ainsi notre approche aux spécificités de chaque modèle.

Cette méthode rigoureuse et hautement parallélisée a grandement contribué à l'optimisation de nos modèles, assurant une personnalisation et une précision maximales dans l'analyse des données EEG pour la détection de mouvement.

Stratégie utilisée pour l'inférence sur les signaux brutes

Pour l'inférence sur les signaux EEG bruts, nous avons employé une technique de fenêtres glissantes, permettant de découper le signal en segments partiellement superposés. Chaque segment est ensuite analysé par notre modèle classificateur. Cette approche assure que chaque point du signal est évalué à travers plusieurs fenêtres, facilitant ainsi la distinction entre les instants de mouvement et ceux de non-mouvement.

La superposition des fenêtres génère une distribution de probabilités pour chaque point du signal, reflétant la probabilité de détecter un mouvement à ce moment précis. L'application d'un seuil, ajustable selon les besoins de l'utilisateur, permet de convertir cette distribution en une décision binaire de mouvement ou de non-mouvement. Ce processus aboutit à une représentation claire et informative de l'activité motrice détectée, exemplifiée dans la figure 14, offrant ainsi une méthode efficace pour l'analyse des signaux EEG en temps réel.

3.3 Développement d'application

Le produit commandé par le client inclut la réalisation d'une interface homme-machine (IHM). Nous avons initialement prévu de rester fidèles à la structure de développement de l'année précédente, avec des ajustements pour nos nouveaux programmes. Cependant, étant donné que la structure existante était spécifiquement conçue pour le programme de l'année dernière et qu'elle s'est avérée difficilement adaptable à nos besoins, nous avons été confrontés à la nécessité de la reconstruire intégralement pour répondre aux exigences actuelles.

Nous avons suivi une approche classique et modulaire en divisant le développement de l'interface en deux parties distinctes, le *frontend* et le *backend*. Cela nous a permis de concentrer nos efforts d'un côté sur les aspects visuels, tels que les affichages et le design graphique, et de l'autre côté sur la manipulation et traitement des données.

Développement du frontend

Pour le frontend, nous avons opté pour **Streamlit**, apprécié pour sa capacité à transformer des scripts Python en applications web de manière intuitive. Cette solution nous a permis de développer rapidement une interface utilisateur conviviale, favorisant un prototypage et une itération agiles. Pour rendre l'application accessible, nous avons utilisé **Uvicorn**, un serveur web ASGI léger, pour héberger notre application Streamlit. Ce serveur, connu pour sa rapidité, supporte les communications asynchrones, garantissant ainsi une interaction efficace et réactive avec les utilisateurs. L'application est ainsi accessible via le navigateur, offrant une expérience utilisateur transparente et sans nécessité d'installations supplémentaires.

Développement du backend

Le backend de notre application a été développé avec le framework Python **FastAPI**, un choix motivé par sa performance et sa facilité d'intégration. **FastAPI** héberge nos modèles, scripts de chargement des données, ainsi que nos algorithmes de préprocessing et d'inférence. L'ensemble est orchestré via une API REST, facilitant la communication avec le frontend. Ce regroupement au sein d'un même framework assure une gestion optimisée des ressources, renforce la sécurité des données, et améliore les performances de traitement. L'utilisation de **FastAPI** a permis de simplifier le déploiement, la maintenance, et les mises à jour, contribuant ainsi à une expérience utilisateur plus fluide.

Communication entre le frontend et le backend

La communication entre le frontend et le backend est établie via HTTP, utilisant l'architecture REST pour un échange de données structuré et efficace. **FastAPI** facilite cette interaction en offrant une interface simplifiée pour la création et la gestion des endpoints API, tandis que la bibliothèque **requests** de Python permet d'envoyer et de recevoir ces requêtes HTTP de manière intuitive. Cette combinaison offre une méthode robuste et flexible pour la communication entre les différentes parties de l'application, assurant une intégration fluide et une expérience utilisateur optimisée.

Déploiement et intégration chez le client

Pour finaliser notre application et la rendre facilement distribuable, nous avons également intégré une étape de compilation qui permet de créer une version standalone de l'application, compatible avec des environnements Windows sans nécessiter d'installation de Python ou d'autres dépendances. Pour cela, nous avons utilisé `PyInstaller`, un outil qui analyse nos scripts Python et crée un exécutable Windows. `PyInstaller` collecte toutes les bibliothèques nécessaires, les fichiers de ressources et les modules Python, puis les emballe dans un seul exécutable. Cette approche simplifie grandement le déploiement de l'application chez les utilisateurs finaux, car ils peuvent exécuter l'application directement sans passer par une phase d'installation complexe. De plus, cela garantit que l'application fonctionne de manière identique sur tous les systèmes, éliminant les problèmes de compatibilité ou de versions de dépendances.

4 Résultats

Cette partie résume les résultats découlant de l'exploration de chaque processus (pipelines), nos avancées concrètes à la suite de ceux-ci.

Après avoir exploré les différentes approches mentionnées ci-dessus et compte tenu des résultats, nous avons opté, dans le cadre la prédiction de mouvement, pour la pipeline utilisant la géométrie Riemannienne comme pipeline finale intégrée à notre produit. Ce choix s'est imposé naturellement compte tenu des résultats peu prometteurs obtenus avec les autres méthodes explorées. Les détails de notre développement sur cette pipeline sont présentés dans la section suivante.

4.1 La pipeline finale

Traitement des données

Comme indiqué précédemment, le traitement de données convertit une base de données mise en ligne sur l'application en un élément de la classe `DataLoader` standardisée sur laquelle nous pouvons appliquer notre extraction de features.

Les choix des paramètres pour ce traitement sont basés sur les résultats de la comparaison effectuée à l'aide des serveurs de Rainman, qui est présentée dans la figure (20).

Les paramètres retenus avec cette méthode sont ceux-ci :

```
EPOCHS_TMIN: -0.75 s
EPOCHS_LENGTH: 1.7 s
EPOCHS_EMPTY_FROM_MVT_TMINS: -6 s
FMIN: 9 Hz
FMAX: 30 Hz
```

Ces hyperparamètres traduisent que les époques sont choisies autour d'un mouvement, en commençant 0.75 seconde avant le mouvement et durant 1.7 seconde. Pour obtenir des époques ne contenant aucun mouvement pour équilibrer le dataset, nous prenons les signaux 6 secondes avant un mouvement : chaque mouvement étant séparé de plus ou moins 10 secondes, nous nous assurons de n'avoir aucun mouvement. D'autre part, seules les fréquences comprises entre 9 et 30 Hz sont conservées (Figure 21).

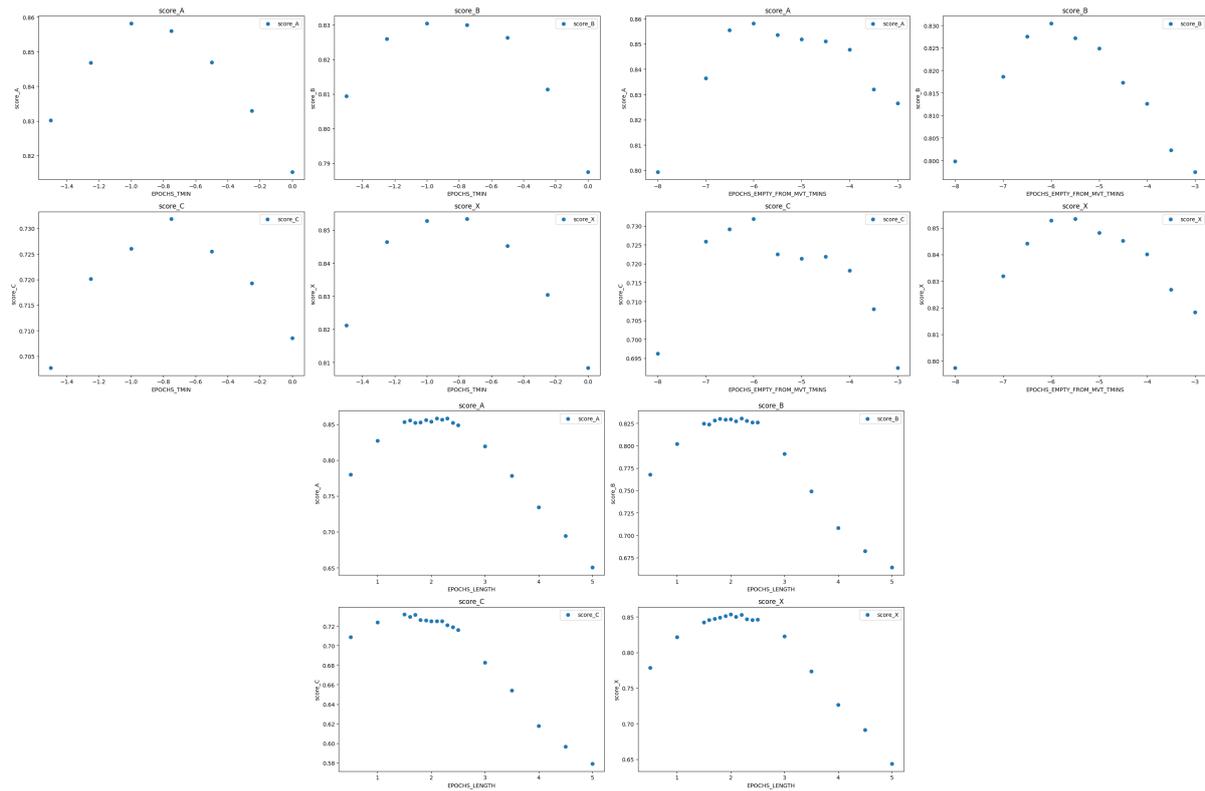


FIGURE 20 – Résultats de la recherche d’hyperparamètre - Différents scores de précision en fonction de la en fonction du début et longueur de l’époque, intervalle de temps entre une époque de non mouvement et une de mouvement

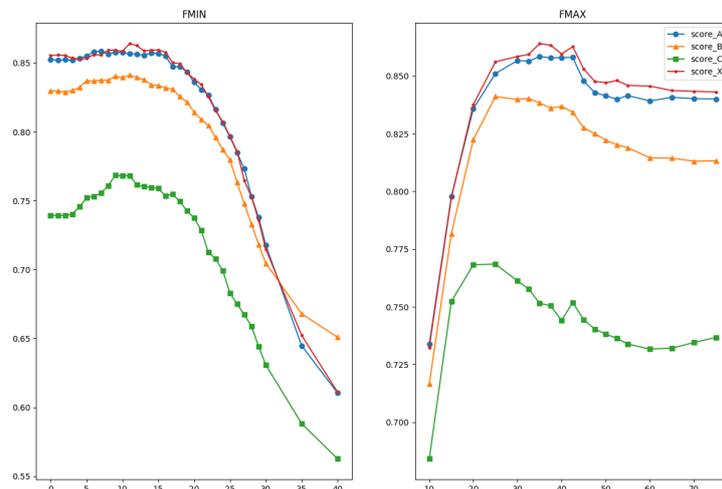


FIGURE 21 – Résultats de la recherche d’hyperparamètres - Différents scores de précision en fonction de la fréquences de filtrage temporel

Extraction de caractéristiques

Il est ressorti de notre recherche que la méthode d’extraction de caractéristiques menant aux plus grands scores est la méthode riemanienne, comme indiqué sur la figure 22.

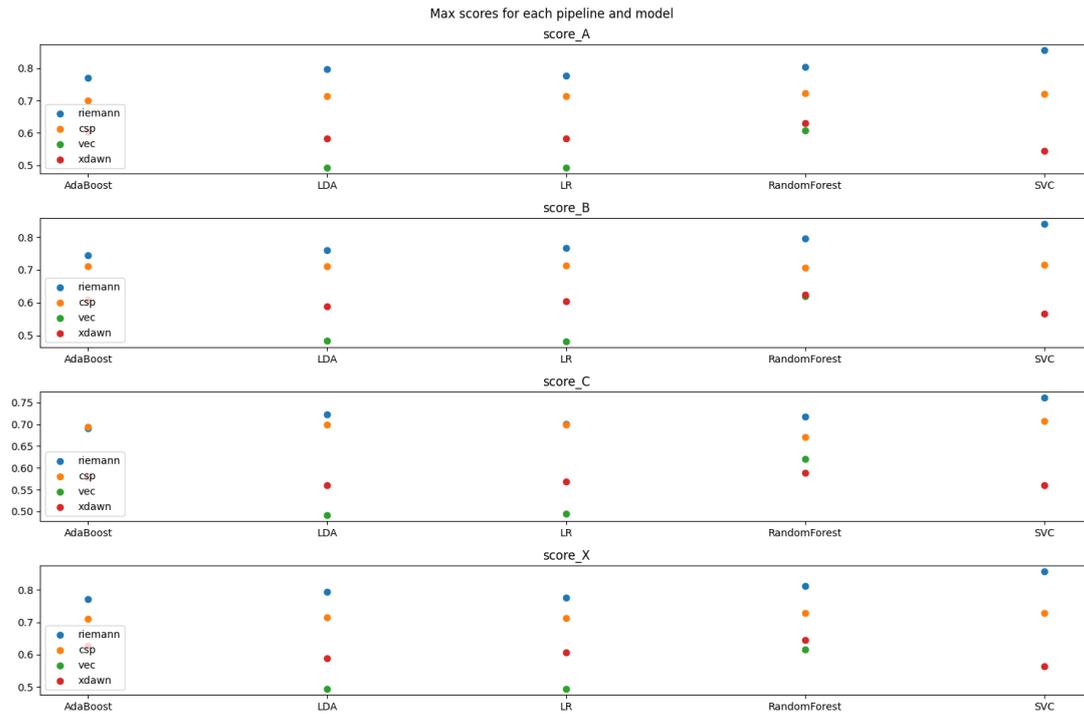


FIGURE 22 – Les différents scores selon la méthode d'extraction de caractéristiques utilisée

Classification

Comme précédemment, nous avons exploré différents classifieurs et croisé leurs résultats après plusieurs simulations sur le serveur de Rainman.

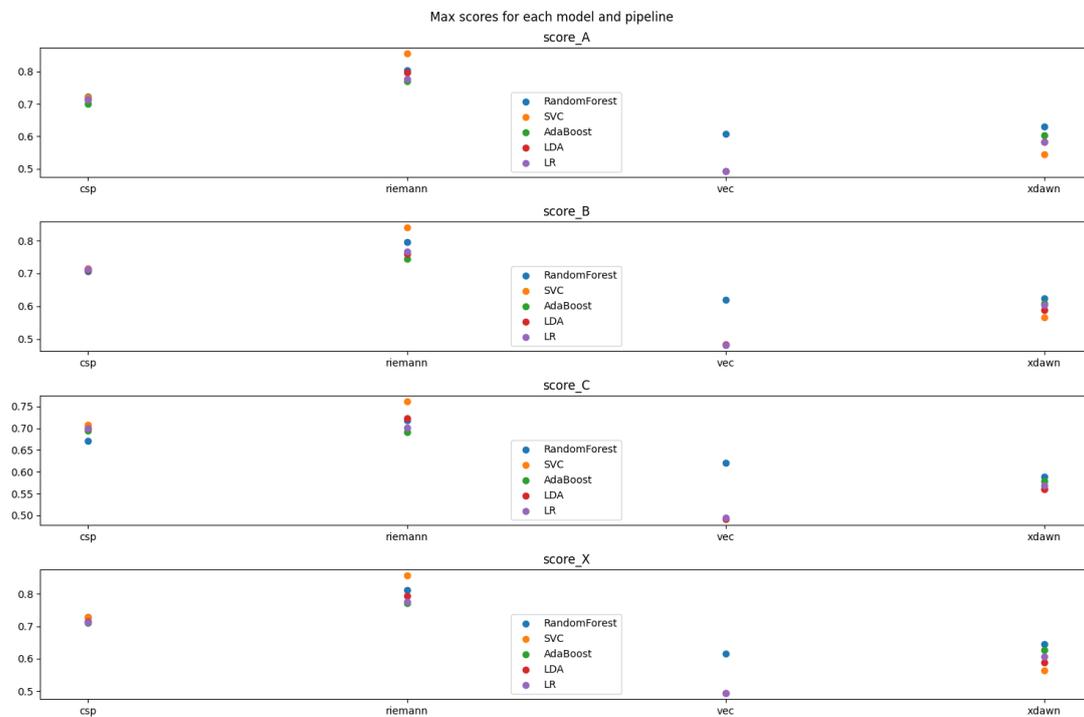


FIGURE 23 – Les différents scores selon le classifieur utilisé

Suite à ces résultats, nous avons décidé de garder le Support Vector Classifier (SVC), qui se révèle être le plus performant avec la méthode d'extraction de feature utilisée ici (appelée méthode de Riemann dans le graphe). Ce classifieur permet d'avoir un équilibre optimal entre précision et généralisation, ce qui en fait le choix le plus robuste parmi les classifieurs testés.

En effet, il est capable de séparer efficacement les classes dans des espaces de haute dimension, ce qui le rend adapté à la classification de données complexes comme celles issues de l'extraction des caractéristiques EEG. De plus, le SVC utilise un concept de marge maximale pour déterminer la frontière de décision entre les classes, ce qui lui permet de généraliser efficacement à de nouveaux exemples tout en minimisant le risque de surajustement. Un schéma explicatif est disponible ci-dessous (la dimension n'est rien d'autre que le nombre de caractéristiques) :

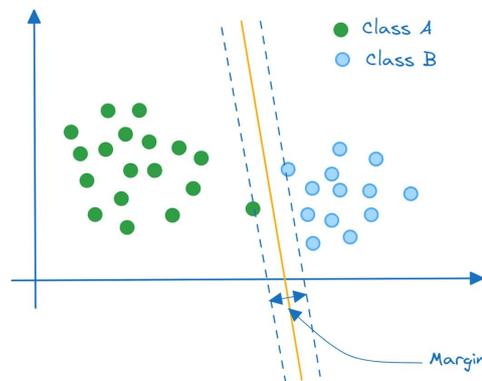


FIGURE 24 – Schéma introduisant le principe du Support Vector Classifier (SVC) en dimension 2

De plus, le SVC est capable de gérer des ensembles de données non linéaires grâce à l'utilisation de noyaux, qui permettent de projeter les données dans un espace de plus grande dimension où elles peuvent être linéairement séparées. Cela lui confère une grande flexibilité pour modéliser des relations complexes entre les caractéristiques extraites des signaux EEG et les intentions de mouvements.

Performances de détection du mouvement

Grâce à ces serveurs de Rainman, nous avons également pu entraîner et optimiser notre modèle final. En utilisant les hyperparamètres précédemment définis, nous avons procédé à l'entraînement du modèle sur ces serveurs. Cette approche nous a permis de bénéficier de toutes les données disponibles pour mener à bien nos différentes méthodes d'évaluation. Nous obtenons des scores d'évaluation autour de 0.85 :

```
score_A 85.5%
score_B 84.0%
score_C 75.8%
score_X 85.6%
```

Prédiction sur l'intention de mouvement

Pour le cas de la prédiction de l'intention de mouvement, nous avons jugé que les scores trouvés dans le cadre de la R&D ne justifiaient pas l'intégration de la meilleure pipeline trouvée dans la solution. Ainsi, un travail de recherche supplémentaire est requis pour explorer d'autres pistes d'algorithmes et optimisations.

4.2 L'application

Caractéristiques du frontend

L'application développée en `Streamlit` permet le traitement des données EEG et la détection du mouvement dans un jeu de données que l'on dépose.

L'application est interactive et son utilisation est claire. Le dépôt de fichiers se fait dans la barre latérale qui affiche aussi les informations du jeu de données. Les différents onglets offrent chacun un ensemble de fonctionnalités pour interagir avec notre algorithme.

Le premier onglet "Paramètres" permet de visualiser les signaux EEG et d'ignorer certaines électrodes du côté parétique si elles ont eu un problème technique ou pour appliquer un paradigme précis. Il permet aussi de rogner le début et la fin de l'expérience pour n'inclure que les données d'intérêt pour la détection. Enfin, cet onglet offre le choix de choisir un modèle, dans le cas où plusieurs modèles sont implémentés pour la détection.

Le second onglet récapitule les paramètres choisis précédemment, avec une gestion des paramètres par défaut si aucune sélection n'a été réalisée. Un bouton permet d'appliquer la pipeline de détection.

Le troisième onglet offre des figures récapitulant la détection d'intention de mouvement avec un seuil, comme cela est fait dans la figure 19.

Cette application possède un guide utilisateur détaillé dans le dernier onglet et des messages d'aide à chaque étape en cas de besoin pour guider au maximum l'expérience de son utilisateur. Sa mise en fonctionnement est détaillée à la fois dans les fichiers `README.md` présents dans notre dossier et dans la partie (8.3).

Caractéristiques du backend

Notre architecture backend est conçue pour faciliter une interaction fluide et efficace avec le frontend, en offrant les fonctionnalités essentielles via trois routes principales :

`/health/` : Cette route sert de vérificateur de l'état du système. En effectuant une requête GET vers cette route, le frontend peut rapidement déterminer si le backend est opérationnel. Elle est essentielle pour les diagnostics et la surveillance de la santé du système, garantissant une communication constante entre le frontend et le backend.

`/upload/` : Conçue pour la gestion des données, cette route accepte les fichiers de session au format `.mat` (MATLAB) ou `.npy` (NumPy), les formats standards pour les données EEG et cinématiques. Lorsqu'un fichier est téléchargé via cette route, le backend procède à l'extraction des métadonnées associées, génère des visualisations des données EEG (et cinématiques, le cas échéant), et identifie les modèles de prédiction disponibles pour ces données. Cette route est cruciale pour préparer et visualiser les données avant

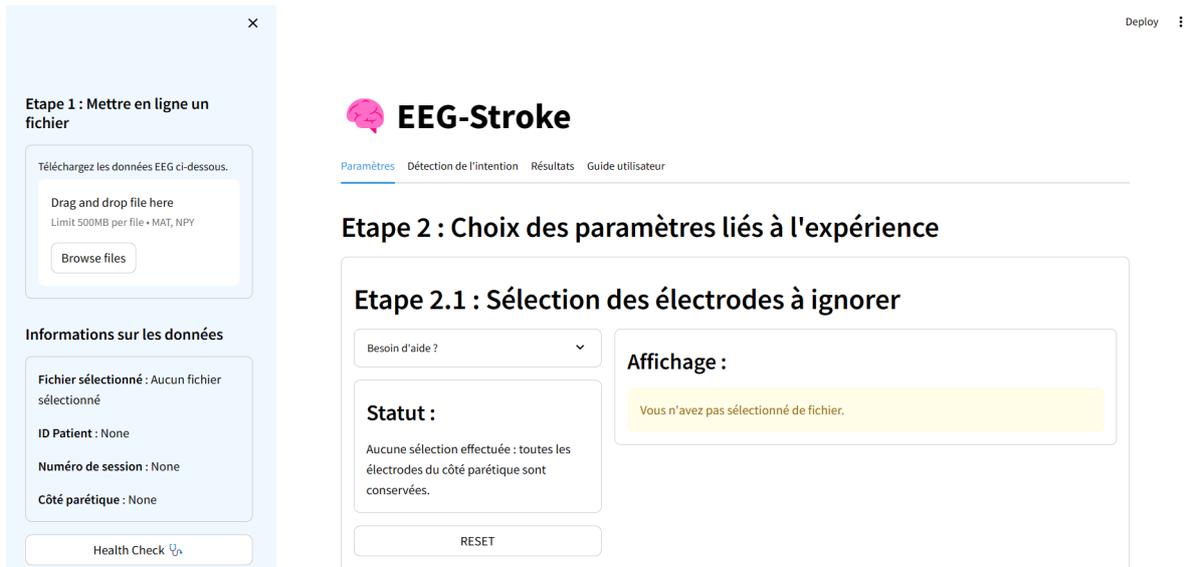


FIGURE 25 – Page d'accueil de l'application



FIGURE 26 – Documentation de l'API fastAPI

toute opération de prédiction, assurant que l'utilisateur dispose de toutes les informations nécessaires pour prendre des décisions éclairées sur le traitement des données.

`/predict/` : Cette route est au cœur de la fonctionnalité de prédiction de notre système. En tenant compte des choix de l'utilisateur effectués via le frontend (telles que le découpage des données, le choix du modèle de prédiction, et la gestion des électrodes défectueuses), elle exécute le modèle de prédiction sélectionné sur les données spécifiées. Les résultats de la prédiction sont ensuite renvoyés au frontend.

Ces trois routes constituent le socle de notre backend, assurant une interface robuste et flexible pour le traitement et l'analyse des données EEG. Chaque route est conçue pour répondre à des besoins spécifiques de notre application, depuis la vérification de la santé du système jusqu'à la visualisation des données et l'exécution de modèles de prédiction complexes, facilitant ainsi une expérience utilisateur cohérente et intuitive.

5 Discussion

5.1 Points forts de notre travail

Notre projet se distingue par plusieurs points forts. Tout d'abord, la rigueur documentaire de nos explorations constitue un atout majeur, offrant des informations claires et concises. Cette caractéristique facilite grandement l'utilisation de nos travaux par différents publics, notamment les chercheurs, les étudiants et le personnel hospitalier.

La justification de nos méthodes, basée sur des articles de recherche, renforce la crédibilité de notre approche. En nous appuyant sur des références scientifiques, nous avons établi des fondations solides pour l'ensemble de nos travaux, en particulier pour cadrer et justifier nos explorations. Aussi, l'ingénierie de données se distingue par sa rigueur et sa transparence, assurant une base robuste pour nos résultats. Notre structure de données est standardisée et fonctionnelle, tout comme notre méthode d'évaluation qui permet une comparaison rigoureuse des résultats de chaque algorithme.

Concernant la précision des algorithmes que nous avons implémentés, nous sommes satisfaits de la précision obtenue dans la détection du mouvement. Cette performance, associée à notre approche méthodologique, fait de notre solution une base solide pour les projets futurs intéressés. Si nos algorithmes se sont intéressés à la détection du mouvement en premier lieu, et non de l'intention, cela a permis d'obtenir des résultats de précision tangibles.

Enfin, notre interface, conçue pour interagir avec notre programme, est à la fois compréhensible et fonctionnelle. Cette convivialité contribue à une expérience utilisateur satisfaisante, renforçant ainsi l'accessibilité de notre solution dans divers contextes. Aussi, l'utilisation de packages modernes et accessibles facilitera sa modification par la suite.

5.2 Limites de notre solution actuelle

Notre PIE possède plusieurs limitations. Tout d'abord, la principale préoccupation réside dans les défis inhérents à la précision et à la fiabilité des algorithmes de détection de l'intention de mouvement basés sur des signaux EEG. Les variations interindividuelles dans les données EEG, ainsi que la présence d'artefacts et d'interférences externes, peuvent grandement affecter la performance de ces algorithmes, nécessitant une validation rigoureuse pour garantir leur efficacité clinique.

De plus, le projet a été confronté à des contraintes de temps et de ressources, notamment en ce qui concerne la familiarisation avec les données et la documentation du projet précédent. Le manque d'une explication claire sur l'utilisation du code et sur la méthodologie de validation des modèles a entraîné des retards dans la progression du projet, compromettant ainsi l'optimisation des ressources disponibles et la pleine exploitation du potentiel des données existantes.

De plus, les contraintes liées aux données est un défi majeur : le jeu de données disponible est très limité. Chaque session ne comporte qu'un nombre restreint de mouvements, avec seulement 2 à 3 essais par patient. Cette quantité limitée de données rend l'apprentissage des algorithmes difficile, limitant leur capacité à généraliser et à apprendre des patterns complexes, ce qui compromet la robustesse des modèles développés. En consé-

quence, il est difficile d'obtenir des performances élevées et généralisables et certaines techniques, comme le deep learning sont difficilement applicables actuellement.

Un des enjeux de cette étude a été d'analyser des données de patients atteints d'AVC en se focalisant sur le côté lésé pour favoriser le neurofeedback. Néanmoins, les données EEG sont fortement affectées par la présence d'une lésion dans le cerveau. Ainsi, les données EEG du côté lésé sont beaucoup plus bruitées que le côté non lésé ce qui rend notre étude plus complexe. La figure 29 montre bien la différence des qualités de données entre les deux côtés avec deux électrodes spatialement équivalentes.

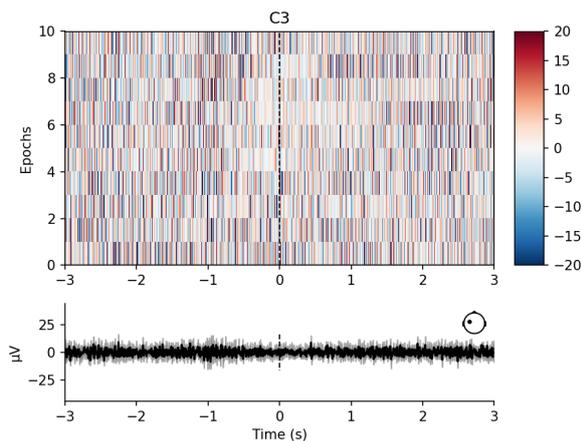


FIGURE 27 – Côté lésé

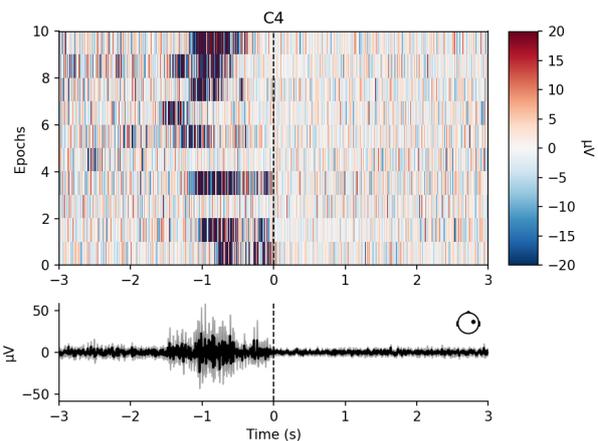


FIGURE 28 – Côté non lésé

FIGURE 29 – Données d'une session réparties en époques centrées sur le mouvement

Une autre limitation majeure réside dans la labélisation de l'intention de mouvement pour sa détection qui demeure l'objectif principal : il est difficile de déterminer avec précision le moment exact de cette intention. Grâce aux données cinématiques, nous avons une connaissance plus fiable du début de mouvement, qui n'est pas accessible avec l'intention. Cette incertitude ajoute une complexité supplémentaire à l'entraînement des algorithmes et rend leur robustesse difficile à évaluer, car leur performance dépend directement de la labélisation.

De plus, l'évaluation de la performance des algorithmes à l'aide de scores standard est une première étape mais leur performance en contexte clinique réel doit être évaluée.

En récapitulant, notre travail a abouti à la création d'un outil de détection du mouvement qui se révèle être d'une efficacité satisfaisante en se basant sur les enregistrements des signaux EEG de patients post-AVC. Cependant, une lacune majeure persiste dans notre capacité à détecter l'intention de mouvement, une tâche bien plus complexe, principalement en raison des limitations de ressources mentionnées précédemment.

6 La suite de ce projet

Ces limitations soulignent la nécessité de continuer le travail que nous avons commencé, dans le cadre d'un stage de recherche ou de prochains PIE. Nous pouvons donc relever les pistes d'améliorations, que nous pensons utiles de considérer à la reprise de notre projet dans les années à venir.

6.1 Processus expérimental

- Il pourrait être judicieux de prévoir un rapprochement d'un ou deux membres du groupe avec le processus expérimental, en allant par exemple directement assister aux expériences qui enrichissent la base de données. Des avantages non négligeables pourraient en découler :
 1. Repertorier les artefacts serait utile. En effet, nous avons remarqué que dans certains cas les données étaient bruitées par des artefacts. Par exemple, lors de la labélisation des phases de mouvements ou d'immobilités, nous avons noté des mouvements parasites (figure 30) dans les données cinématiques de certaines sessions ce qui rendaient la labélisation du mouvement beaucoup plus compliqué. Les données EEG sont elles aussi contaminées par ces artefacts. Cela peut complexifier énormément la classification par nos modèles. Certains artefacts sont connus comme le clignement des yeux. Il serait intéressant de comprendre si d'autres artefacts peuvent nuire à la qualité des signaux afin de mieux les prendre en compte dans le traitement des données.

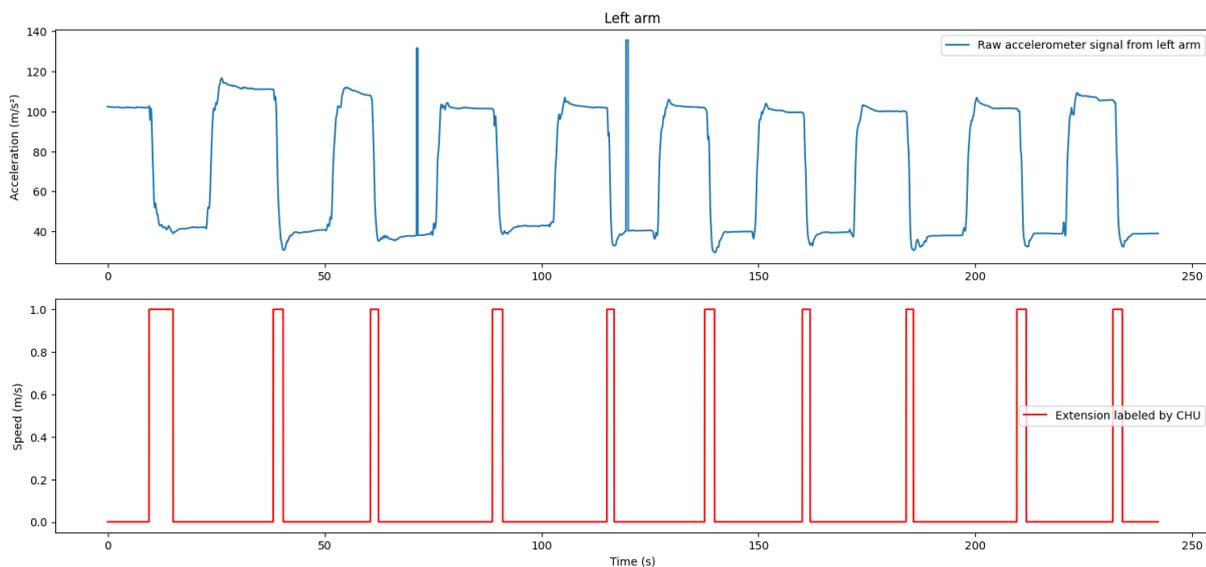


FIGURE 30 – Données cinématiques d'un patient contenant deux artefacts

2. La suggestion d'améliorations au protocole expérimental pourrait être facilitée. En effet, afin d'augmenter le niveau de précision et la fiabilité des classificateurs, des points d'améliorations peuvent être pensés directement lors du processus expérimental. Pour illustrer, notre but est de détecter l'intention de mouvement et non le mouvement. Dès lors, en utilisant uniquement les données cinématiques,

il est très compliqué de labéliser avec précision l'intention de mouvement des patients. C'est une des limites que nous avons soulevé lors de notre étude. Pour pallier à cette perte de précision dans la labélisation, on pourrait alors penser à modifier légèrement l'expérience en utilisant un signal sonore périodique qui indiquerait au patient quand faire le mouvement. Ce signal sonore serait inclus dans la base de données et agirait comme un stimulus permettant de mieux labeliser l'intention de mouvement. Il serait alors possible de labeliser comme intention du mouvement, le moment qui sépare le signal sonore et le mouvement capturé grâce à l'accéléromètre. Le patient pourrait attendre une seconde ou deux après le signal sonore afin d'éviter que l'algorithme s'appuie sur l'ERP (Event-related potential) au niveau du cortex auditif.

6.2 Algorithmes

Il est crucial d'explorer davantage les possibilités d'amélioration des algorithmes qui permettraient de passer de la détection du mouvement à la détection plus difficile de l'intention de mouvement.

Une piste d'exploration des méthodes de machine learning non-supervisé telle que le *K-means* pourraient également être intéressante. De telles approches pourraient effectivement permettre de considérer des modèles de classification adaptatif, se rapprochant des notions de transfer learning, qui apprendraient de manière non-supervisée à généraliser les caractéristiques clés d'un patient à l'autre.

Enfin, une phase supplémentaire de développement consiste à préparer l'outil pour une utilisation éventuelle en ligne. Actuellement, notre solution est utilisée hors ligne, afin de détecter les mouvements ou intentions de mouvements sur des données préalablement acquises. L'objectif final de ce projet est de pouvoir détecter en temps réel ces intentions de mouvements, dans le but d'intégrer la solution dans une interface cerveau-machine fonctionnelle en temps réel (BCI). Cette évolution permettrait une interaction directe et fluide entre l'utilisateur et notre système, ouvrant ainsi la voie à des applications cliniques plus dynamiques et efficaces dans le domaine de la rééducation post-AVC.

6.3 Intention de mouvement

Il incombera à la prochaine équipe PIE de s'intéresser à la problématique de détection de l'intention de mouvement en s'appuyant sur nos résultats pour la détection de mouvement. Une problématique principale étant la labelisation de l'intention de mouvement.

6.4 Gestion de projet

Un effort significatif a été consacré à la qualité du code livré. La plupart des fichiers fournis sont accompagnés d'explications détaillées et de précisions techniques, facilitant ainsi la compréhension du flux de pensée par les futurs intervenants. Cette approche vise à permettre une continuation fluide du travail déjà accompli et le prochain PIE devra veiller à s'inscrire dans une démarche du même type afin de veiller à l'amélioration continue du process.

7 Conclusion de ce PIE

Le projet "BCI-EEG-STROKE" représente une avancée prometteuse dans le domaine de la rééducation post-AVC en cherchant à exploiter la plasticité cérébrale à travers le neurofeedback. En permettant aux patients de générer des mouvements du côté paralysé grâce à leurs propres signaux cérébraux, cette approche offre une voie innovante pour stimuler la récupération motrice. Malgré les défis rencontrés, nous sommes très satisfaits du travail accompli. Nous avons délivré un produit fonctionnel, efficace et facile d'utilisation au CHU de Toulouse, tout en documentant le processus mis en place afin d'aider les prochains groupes à rapidement prendre en main le sujet sans perdre de temps sur des méthodes déjà testées

7.1 Bilan sur la solution

Comme énoncé dans la section précédente, nous avons pu relever des points positifs à notre travail. Tout d'abord, la rigueur documentaire de nos recherches offre des informations claires et concises, soutenue par des références scientifiques, consolidant ainsi la crédibilité de notre approche. Par ailleurs, notre ingénierie de données rigoureuse et transparente établit des fondements solides pour nos travaux. De plus, concernant les résultats obtenus, la précision satisfaisante de nos algorithmes de détection du mouvement témoigne de l'efficacité de notre approche méthodologique. Enfin, la conception d'une interface utilisateur conviviale contribue à une expérience utilisateur positive, renforçant ainsi l'accessibilité de notre solution dans divers contextes.

Cependant, malgré ces avancées, notre projet est confronté à plusieurs limitations importantes. En particulier, la détection précise de l'intention de mouvement reste un défi majeur en raison des variations interindividuelles dans les données EEG et des contraintes de ressources. De plus, la qualification des algorithmes développés est complexe et nécessite une approche nuancée pour évaluer leur efficacité réelle dans un environnement clinique.

Pour aller de l'avant, il est crucial d'explorer davantage les possibilités d'amélioration des algorithmes, en particulier en se concentrant sur des approches adaptatives. De plus, une préparation pour une utilisation en ligne de notre solution est nécessaire afin de permettre une détection en temps réel des intentions de mouvements, ouvrant ainsi la voie à des applications cliniques plus dynamiques et efficaces dans le domaine de la rééducation post-AVC.

7.2 Bilan sur la gestion du projet

Concernant l'organisation mise en place, nous pouvons affirmer qu'elle présente plusieurs points forts ainsi que quelques faces d'améliorations potentielles.

Parmi les points positifs, l'utilisation efficace d'outils de gestion de projet tels que Jira ou des diagrammes de Gantt offre une visibilité claire sur les tâches à accomplir et leur progression. Cette approche facilite la planification et le suivi des activités, contribuant ainsi à la cohérence et à la coordination de l'équipe. De plus, l'utilisation d'outils collaboratifs comme Overleaf pour la rédaction des rapports et GitHub pour le développement permet une collaboration simultanée des membres de l'équipe, favorisant une répartition efficace des tâches et une fusion harmonieuse des contributions individuelles. La communication

régulière et transparente instaurée lors des séances de PIE a permis à chaque membre de présenter ses avancées, d'évaluer collectivement l'avancement global du projet et de planifier les futures actions, ce qui favorise une compréhension commune des objectifs et une collaboration efficace. En ce qui concerne l'interaction avec le client, la planification de réunions rapides et d'entrevues physiques au CHU a permis de maintenir un contact étroit, de faire le point sur l'avancement du projet et d'obtenir des retours précieux.

Parmi les défis rencontrés, nous pouvons mentionner le manque de temps couplé au manque de ressources provenant du projet de l'année précédente. En effet, le temps pris pour se familiariser avec les données s'est révélé plus long que prévu, d'autant plus que nous n'avions pas de documentation sur l'utilisation ou la construction du code du PIE précédent. En l'absence d'une explication détaillée sur la manière de reprendre le code et sur la méthodologie de validation et d'entraînement des modèles, nous avons dû consacrer des efforts supplémentaires d'exploration et d'appropriation du problème, refaisant ainsi potentiellement des parties déjà faites mais non documentées et intelligibles du PIE précédent. Malgré ces efforts, cette non-utilisation des ressources du projet précédent a entraîné des retards dans notre progression. Pour pallier à cela et aider les prochains projets, nous avons mis en place des notebooks documentés expliquant chaque étape du développement.

Un deuxième défi rencontré a été l'organisation initiale des tâches entre les membres du groupe. En divisant les tâches en gros blocs tels que le pré-traitement, l'extraction des caractéristiques et la classification, nous avons réalisé que cela ne permettait pas une familiarisation adéquate avec les données. En effet, nous nous sommes rendus compte que chaque bloc était étroitement inter-connecté avec les autres. Nous avons donc réajusté notre approche en réalisant des explorations individuelles de pipelines entières comprenant toutes des parties pré-traitement, extraction de caractéristiques et classification, ce qui nous a permis de trouver des solutions plus efficaces sans être limités par des blocages ou retards dans le développement d'autres blocs.

Enfin, un point regrettable a été le manque de participation aux expériences pratiques avec les patients atteints d'AVC au CHU. En raison de contraintes d'emploi du temps, nous n'avons pas pu assister à des séances de bilan avec ces patients, ce qui aurait été une expérience précieuse pour mieux comprendre leurs besoins et leurs défis. Bien que nous ayons eu l'opportunité d'assister à des bilans avec d'autres patients présentant des problèmes moteurs, cela ne remplace pas la valeur d'une expérience directe avec les personnes concernées par notre projet

8 Annexes

Cette partie répertorie toutes les informations supplémentaires à ce rapport n'ayant pas leur place dans le corps principal du rapport.

8.1 Organisation du dossier GitHub

Nous avons utilisé l'outil d'aide au développement et à la gestion de version de code `git`, à travers la plateforme GitHub. Nous détaillons dans cette annexe la structure détaillée du `repository` GitHub que nous avons utilisé. Le `repository` se veut cependant explicable par lui-même avec dans chaque dossier un `README.md` qu'il ne faut pas hésiter à consulter, des notebooks commentés et mis en formes ainsi qu'un code documenté.

Le `repo` se décompose en divers dossiers principaux qui contiennent les fichiers de code correspondant aux différents axes de développements que nous avons eu. En détail :

- Le dossier `app/` contient les codes de l'interface homme-machine (rendu client final). Le dossier `app/` se décompose lui-même en sous-dossiers `frontend/` qui contient respectivement et `backend/` qui contient les codes liés à l'utilisation des modèles entraînés et du traitement des données (réalisés en fond).
- Le dossier `data/` contient les données fournis par le CHU au format `.npy`. Notons que ces données restent en local (ne sont pas stockées dans le `repo git`) car bien trop lourdes en terme de stockage.
- Le dossier `docs/` contient les articles et notes prises lors de notre recherche bibliographique et constitution de l'état de l'art.
- Le dossier `gestion_projet/` contient les documents liés à la gestion de projet. On y retrouve le Plan de Développement (PDD), les supports des soutenances technique (18/12) et mid-term (15/01), ainsi que les ressources fournies par l'école pour aider à la constitution des livrables (PDD, rapport technique, fiche de synthèse, etc).
- Le dossier `notebooks/` contient les notebooks qui ont été rédigés durant la phase exploratoire du projet. Ce dossier se décompose lui-même en plusieurs sous-dossiers, permettant de regrouper les explorations par thématiques. On retrouve dans ce dossier :
 - un ensemble d'explorations générales sur la structure des données, les différentes techniques de pré-traitement et sur des outils de visualisation des données et des résultats.
 - un ensemble d'explorations faites sur des `pipeline` de traitement des données EEG complètes (chargement, pré-traitement, extraction des caractéristiques, classification).
 - un ensemble de notebooks sur des méthodes *golden standard* mettant en valeur certaines techniques qui pouvaient s'avérer utiles pour le projet (méthode de *deep learning*, de géométrie Riemannienne, etc).
 - un ensemble de notebooks permettant l'optimisation des hyperparamètres de

la pipeline finale que l'on a choisi d'implémenter dans le produit client.

- Le dossier `src/` contient notre librairie "**ML-EEG-TOOLS**" qui rassemble les fonctions utiles pour :
 - le traitement des données EEG (extraction des données, chargement des données, préparation des données),
 - le test des modèles (les quatre types de tests que nous avons définis : test A, test B, test C, test X),
 - l'entraînement des modèles (procédure de *grid search* pour l'optimisation des paramètres du modèle choisi).

Le dossier `src/training` contient les les script de configuration et les résultats bruts des recherche et entraînements sur le cluster de calcul

8.2 Structure et format des données

Structure des données brutes

La structure des données brutes extraites des fichiers `.npy` est la suivante :

- [0] IDENTIFIANT SESSION : identifiant unique pour chaque session de test (e.g. '004')
- [1] CÔTÉ DE LA LÉSION : côté du cerveau qui est lésé (G pour gauche, D pour droite)
- [2] NUMÉRO DE L'ESSAI : identifiant de l'essai en cours
- [3] FRÉQUENCE ÉCHANTILLONAGE ACCÉLÉROMÈTRE : fréquence à laquelle les données d'accélération sont échantillonnées (e.g. 125Hz)
- [4] DONNÉES SESSION BRAS GAUCHE :
 - CÔTE : gauche
 - INDEX DE MOUVEMENT : liste des index où un mouvement a été détecté
 - DONNÉES EEG : liste de paires ['nom', 'données'] où 'nom' est le nom de l'électrode et 'données' sont les mesures EEG correspondantes.
 - DONNÉES CINÉMATIQUES : listes de la vitesse et accélération en 2D et de la vitesse et accélération en 3D.
- [5] DONNÉES SESSION BRAS DROIT :
 - CÔTE : droit
 - INDEX DE MOUVEMENT : liste des index où un mouvement a été détecté
 - DONNÉES EEG : liste de paires ['nom', 'données'] où 'nom' est le nom de l'électrode et 'données' sont les mesures EEG correspondantes.
 - DONNÉES CINÉMATIQUES : listes de la vitesse et accélération

en 2D et de la vitesse et accélération en 3D.

[6] INDEX PATIENT : identifiant unique du patient.

Pour note, la fréquence d'échantillonnage des signaux EEG est de 1024 Hz et le nombre d'électrode 64.

Format de donnée mne

Après chargement, les données sont mises dans des objets `mne.io.RawArray` :

```
class mne.io.RawArray(data, info, first_samp=0, copy='auto', verbose=None)
    """
    Raw object from numpy array.

    Parameters:
    data : array, shape (n_channels, n_times)
        The channels' time series.

    info : mne.Info
        The mne.Info object with information about the sensors and methods
        of measurement. Consider using mne.create_info() to populate this
        structure. This may be modified in place by the class.

    first_samp : int
        First sample offset used during recording (default 0).

    copy : {'data', 'info', 'both', 'auto', None}
        Determines what gets copied on instantiation. "auto" (default)
        will copy info, and copy "data" only if necessary to get to double
        floating point precision.

    verbose : bool | str | int | None
        Control verbosity of the logging output. If None, use the default
        verbosity level. See the logging documentation and mne.verbose()
        for details. Should only be passed as a keyword argument.
    """
```

8.3 Utilisation de l'application

Afin de démarrer l'application, il faut démarrer le backend et le frontend. Le démarrage de cette application nécessite Python 3. Les informations plus complètes sont disponibles dans les documentations des programmes.

Démarrage du frontend

Les étapes 1), 2) et 4) ne sont à réaliser que la première fois.

- 1) Démarrez un terminal dans le dossier `./app/frontend`.
- 2) Construisez un environnement virtuel en lançant la commande `python -m venv venv`.
- 3) Activez l'environnement virtuel en lançant la commande `./venv/Scripts/activate`
- 4) installez les packages nécessaires en lançant `pip install -r requirements.txt`
- 5) Lancez l'application en executant la commande `streamlit run frontend_main.py`.
- 6) L'application démarre en local et est accessible à l'URL indiqué. Elle s'ouvre dans votre navigateur.

La page d'accueil est représentée sur la figure 25.

Démarrage du backend

Les étapes 1), 2) et 4) ne sont à réaliser que la première fois.

- 1) Démarrez un terminal dans le dossier `./app/backend`.
- 2) Construisez un environnement virtuel en lançant la commande `python -m venv venv`.
- 3) Activez l'environnement virtuel en lançant la commande `./venv/Scripts/activate`
- 4) Installez les packages nécessaires en lançant `pip install -r requirements.txt`
- 5) Une fois les dépendances installées, lancez le backend avec la commande `uvicorn main:app --reload`.
- 6) Le serveur du backend démarre en local et est accessible à l'URL indiqué.

Démarrage de l'app avec Makefile

L'outil `make` présent nativement sur de nombreux systèmes d'exploitation permet d'encapsuler des commandes. Ainsi en se plaçant dans le dossier racine du projet,

- 1) `make setup-app` installera les dépendances et l'environnement virtuel nécessaire au backend et frontend.
- 2) `make run-app` lancera simultanément le backend et frontend.

Il faut noter que les commandes ont été faites pour une système d'exploitation Windows possédant Python3.8 ou plus.

Guide utilisateur

Si vous ne savez pas comment utiliser l'application, consultez l'onglet "Guide utilisateur" disponible sur l'application comme montré sur la figure 31. De même, chaque étape est ponctuée d'un cadre "Besoin d'aide?" offrant des indications sur l'étape actuelle.

Etape 1 : Mettre en ligne un fichier

Téléchargez les données EEG ci-dessous.

Drag and drop file here
Limit 500MB per file • MAT, NPY

Browse files

Informations sur les données

Fichier sélectionné : Aucun fichier sélectionné

ID Patient : None

Numéro de session : None

Côté parétique : None

Health Check

EEG-Stroke

Paramètres Détection de l'intention Résultats Guide utilisateur

Bienvenue dans le guide utilisateur !

Vous êtes sur l'onglet contenant le guide utilisateur de l'application EEG-Stroke : si vous avez besoin d'aide pour utiliser cette application vous êtes au bon endroit !

Cette application permet de détecter l'intention de mouvement dans le signal électroencéphalographique de patients post-AVC qui réalisent des séries de flexion-extension du bras.

Elle s'inscrit dans une volonté d'utiliser le neurofeedback pour la rééducation de ces patients.

Avec ce guide vous allez apprendre à :

1. Mettre en ligne un fichier EEG,
2. Sélectionner les électrodes ayant des problèmes et le début et la fin de l'expérience,
3. Paramétrer et réaliser la détection de l'intention de mouvement.

FIGURE 31 – Page du guide utilisateur de l'interface graphique

Bibliographie

- [1] David Steyrl et AL. “Random forests in non-invasive sensorimotor rhythm brain-computer interfaces : A practical and convenient non-linear classifier”. In : *Biomedical Engineering / Biomedizinische Technik* 61.1 (2016), p. 77-86.
- [2] Alexandre BARACHANT et al. “Classification of covariance matrices using a Riemannian-based kernel for BCI applications”. In : *Neurocomputing* 112 (2013). Advances in artificial neural networks, machine learning, and computational intelligence, p. 172-178. ISSN : 0925-2312. DOI : <https://doi.org/10.1016/j.neucom.2012.12.039>. URL : <https://www.sciencedirect.com/science/article/pii/S0925231213001574>.
- [3] Alexandre BARACHANT et al. “Multiclass Brain-Computer Interface Classification by Riemannian Geometry”. In : *IEEE Transactions on Biomedical Engineering* 59.4 (2012), p. 920-928. DOI : [10.1109/TBME.2011.2172210](https://doi.org/10.1109/TBME.2011.2172210).
- [4] Benjamin BLANKERTZ et al. “Optimizing Spatial filters for Robust EEG Single-Trial Analysis”. In : *IEEE Signal Processing Magazine* 25.1 (2008), p. 41-56. DOI : [10.1109/MSP.2008.4408441](https://doi.org/10.1109/MSP.2008.4408441).
- [5] Krauledat M Losch F Curio G Müller K R DORNHEGE G Blankertz B. “Combined optimization of spatial and temporal filters for improving brain-computer interfacing”. In : *IEEE Trans. Biomed. Eng.* (2006).
- [6] Senhadji L Comon P KACHENOURA A Albera L. “ICA : a potential tool for BCI systems”. In : *IEEE Signal Process Mag* (2008).
- [7] F LOTTE et al. “A review of classification algorithms for EEG-based brain-computer interfaces : a 10 year update”. In : *Journal of Neural Engineering* 15.3 (2018), p. 031005. DOI : [10.1088/1741-2552/aab2f2](https://doi.org/10.1088/1741-2552/aab2f2). URL : <https://dx.doi.org/10.1088/1741-2552/aab2f2>.
- [8] Norizam Sulaiman et al. MAMUNUR RASHID. “Current Status, Challenges, and Possible Solutions of EEG-Based Brain-Computer Interface : A Comprehensive Review”. In : *Frontiers in Neurobotics* (2020).
- [9] H. RAMOSER, J. MULLER-GERKING et G. PFURTSCHELLER. “Optimal spatial filtering of single trial EEG during imagined hand movement”. In : *IEEE Transactions on Rehabilitation Engineering* 8.4 (2000), p. 441-446. DOI : [10.1109/86.895946](https://doi.org/10.1109/86.895946).
- [10] Mamunur RASHID et al. “Current Status, Challenges, and Possible Solutions of EEG-Based Brain-Computer Interface : A Comprehensive Review”. In : *Frontiers in Neurobotics* 14 (2020). ISSN : 1662-5218. DOI : [10.3389/fnbot.2020.00025](https://doi.org/10.3389/fnbot.2020.00025). URL : <https://www.frontiersin.org/articles/10.3389/fnbot.2020.00025>.

-
- [11] Bertrand RIVET* et al. “xDAWN Algorithm to Enhance Evoked Potentials : Application to Brain–Computer Interface”. In : *IEEE Transactions on Biomedical Engineering* 56.8 (2009), p. 2035-2043. DOI : [10.1109/TBME.2009.2012869](https://doi.org/10.1109/TBME.2009.2012869).
- [12] Florian YGER. “A review of kernels on covariance matrices for BCI applications”. In : (2013), p. 1-6. DOI : [10.1109/MLSP.2013.6661972](https://doi.org/10.1109/MLSP.2013.6661972).