

Rapport final (Juin 2023)

**Brain-Computer Interface :
Application d'authentification à base
de l'électroencéphalogramme**

Projet encadré par :

Dr. BOUBCHIR Larbi

Professeur en Computer Science & IA

Projet présenté par :

GROLLEAU Antoine

MARTINEZ Clara

OURY Marie

PIALOUX Louise

Étudiant à l'ESME Paris

Étudiante à Sup'Biotech Paris

Étudiante à l'ESME Paris

Étudiante à Sup'Biotech Paris

Abréviations

ACC : Précision (*Accuracy*)

CNN : *Convolutional Neural Network*

BCI : Brain Computer-Interface

EEG : Électroencéphalogramme ou Électroencéphalographie

ERP : *Event Related Potentials*

IA : Intelligence artificielle

ICM : Interface Cerveau Machine

GAN : *Generative Adversarial network*

KNN : *K-Nearest Neighbors*

LDA : *Linear Discriminant Analysis*

LSTM : *Long Short-Term Memory*

ML : Machine Learning

MCC : *Mathew Correlation Coefficient*

MLP : Perceptron multicouche

P300 : Potentiel évoqué

SSVEP : Potentiels Évoqués Visuels Stables (*Steady-State Visual Evoked Potentials*)

SVM : *Support Vector Machine*

RNN : *Recurrent Neural Network*

Glossaire

Biométrie : Techniques informatiques mesurant et analysant statistiquement les caractéristiques biologiques, physiques et comportementales des individus dans le but de les reconnaître automatiquement.

Canvas : Canvas de Tkinter est un outil utilisé avec Python pour la création d'interfaces graphiques.

Encéphalogramme : Image médicale radiologique représentant l'activité du cerveau. Elle est obtenue par suite d'un électroencéphalogramme.

Intelligence artificielle : Algorithmes informatiques utilisés par une machine dans le but d'imiter l'intelligence humaine et son comportement comme son raisonnement ou sa créativité.

Machine Learning : Algorithme informatique développé pour découvrir des « *patterns* », soit des motifs récurrents dans des ensembles de données.

Neurone : Cellule nerveuse qui reçoit, transmet, traite et produit des informations.

Wavelet Transform : Cette technique qui encode les données EEG originales en utilisant les ondelettes, appelées blocs de construction simples.

Table des Illustrations

Figure 1 : Architecture générale d'un système Brain Computer Interface.

Figure 2 : Architecture générale des systèmes de contrôle des signaux utilisés en BCI.

Figure 3 : Algorithmes de classification couramment utilisés en BCI.

Figure 4 : Interface graphique des 3 formes clignotantes à des fréquences 3,7 et 11 hertz pour la simulation visuelle.

Figure 5 : Carte des capteurs pour contrôler la qualité de la connectivité sur le logiciel EmotivLauncher.

Figure 6 : Paramètre utilisé pour configurer OpenVibe Acquisition Server.

Figure 7 : Serveur d'acquisition des données sur le logiciel Open Vibe.

Figure 8 : Répartitions des électrodes Système EEG 14 canaux EPOC +.

Figure 9 : Approche de l'optimisation des hyper-paramètres pour post-ajustement du modèle Random Forest. (Koehrsen, 2018)

Figure 10 : Représentation schématique de la validation croisée. (Sklearn, 2022)

Figure 11 : Représentation schématique d'une procédure de validation croisée. (Sklearn, 2022)

Figure 12 : Matrice tridimensionnelle composées des données EEG.

Figure 13 : Tableau récapitulant les résultats obtenus sur différents modèles de machine learning avec et sans hyper paramètres ainsi que pour l'évaluation sur une nouvelle base de données

Figure 14 2 : Matrice de corrélation obtenue avec l'algorithme de Random Forest présentant une authentification BCI de 100%

Table des matières

Introduction.....	6
Partie I : Analyse Théorique et Bibliographique	8
A. Un nouveau domaine de recherche : Brain Computer Interface	8
B. Structure du BCI.....	8
1) Électroencéphalogramme (EEG).....	10
2) Signaux de contrôle du cerveau	11
C. Les caractéristiques des signaux EEG	13
D. Classification en BCI	13
Partie II : Matériels et Méthodes.....	15
A. Acquisition des données	15
1) L'interface graphique	15
2) Génération des signaux	17
B. Extraction des données	18
1) Extraction des signaux	18
C . Classification des signaux.....	20
1) Approche générale de Machine Learning	20
2) Préparation des données	20
3) Construction du modèle.....	21
4) Procédure de validation	25
5) Évaluation du modèle.....	26
Partie III : Présentation et Analyse des Résultats.....	27
A. Acquisition des signaux	27
B. Classification en Machine Learning.....	28
1) Prédiction d'une application d'authentification en BCI.....	28
2) Généralisation du modèle.....	31
C. Discussion	31
Conclusion	33
Bibliographie	34

Introduction

Ces dernières années, les informations biométriques ont gagné en acceptabilité en raison de leur fiabilité et de leur adaptabilité, ainsi, elles ont été grandement utilisées pour des applications d'identification. Les systèmes d'identification biométrique reposent principalement sur les caractéristiques physiologiques intrinsèques et uniques à chaque individu (par exemple, le visage (Geof, 2013), l'iris (Neal, 2013), la rétine (Fahreddin, 2016), la voix (Steven Goldstein, 2016) ou l'empreinte digitale (JA Unar, 2014). Cependant, les systèmes d'identification actuels présentent des limites ; tels que les masques anti-surveillance contrecarrant la reconnaissance faciale, les lentilles de contact qui peuvent tromper la reconnaissance de l'iris, les vocodeurs qui peuvent compromettre l'identification de la voix et les films d'empreintes digitales qui peuvent tromper les capteurs d'empreintes digitales.

Le système d'identification basé sur les signaux EEG (électroencéphalographie) est une approche émergente en biométrie physiologique. De tels systèmes mesurent la réponse cérébrale d'un individu à un certain nombre de stimuli sous forme de signaux EEG, qui enregistrent les oscillations neurales électromagnétiques, invisibles et intouchables. Ces caractéristiques rendent l'identification basée sur les EEG hautement résiliente aux attaques et protègent contre la menace d'être trompé, ce qui est souvent le cas avec d'autres techniques d'identification. Les signaux EEG présentent des avantages inhérents significatifs par rapport à d'autres biométries, tels que la résilience aux attaques, l'universalité, l'unicité et l'accessibilité (Xiang, 2017).

Cependant, malgré les efforts récents, la recherche sur l'identification basée sur les EEG en est encore à ses débuts et plusieurs défis clés existent. L'un des défis les plus importants est la stabilité médiocre du système d'identification, qui peut fonctionner correctement à un moment, mais échouer à un autre en raison d'interférences dans les signaux EEG (Leonard J Trejo, 2015). Pour résoudre ce problème, les chercheurs tentent d'apprendre une représentation robuste et fiable via la décomposition des modèles EEG. Un autre défi est lié aux performances, notamment en termes d'exactitude, de robustesse et d'adaptabilité. Les algorithmes d'identification existants dépendent fortement de l'environnement de collecte EEG, ce qui peut entraîner une baisse de l'exactitude. Ainsi, les chercheurs cherchent à développer un algorithme d'identification EEG universel qui puisse fonctionner efficacement dans une variété d'environnements réels.

Dans le cadre de notre projet, les activités cérébrales collectées via l'électroencéphalogramme (EEG) sont exploitées afin d'authentifier une personne via une application de BCI (Brain-Computer Interface). À terme, une application du projet serait l'authentification d'un utilisateur via une interface visuelle lui permettant de déverrouiller son espace personnel tel que sa boîte mail.

Les principales contributions de ce projet sont les suivantes :

- Nous avons conçu et réalisé une expérience EEG en recueillant des données locales du monde réel qui sont collectés séparément dans le cadre d'un essai unique et d'essais multiples. Cette expérience EEG a nécessité la création d'une interface graphique visuelle par les outils d'UX Design pour la simulation visuelle et la génération des données EEG.
- Nous avons analysé la décomposition du modèle EEG afin de proposer le modèle le plus stable et le plus facile à distinguer pour l'identification de l'utilisateur.
- Nous avons développé une méthode de classification des données EEG par les outils informatique de Machine Learning Pour cela, nous avons mis en place et déterminé trois étapes à suivre. Tout d'abord, le développement d'une interface graphique pour la stimulation visuelle sur Python, puis l'acquisition des données avec le casque EEG ainsi que leur traitement, et enfin leur classification sur Python avec du Machine Learning permettant à notre approche de rechercher automatiquement les caractéristiques les plus discriminantes pour l'identification et, par conséquent, de fonctionner de manière robuste et adaptative sur différents ensembles de données et environnements de collecte.

Partie I : Analyse Théorique et Bibliographique

A. Un nouveau domaine de recherche : Brain Computer Interface

Un système BCI ou ICM en français (Interface Cerveau Machine) est un système informatisé qui permet la communication directe entre un cerveau humain et un ordinateur, sans communication musculaire. Il acquiert des signaux cérébraux, les analyse et les traduit en commandes qui sont transmises à un dispositif de sortie afin d'effectuer une action souhaitée (Mridha, 2021).

Les systèmes de BCI n'utilisent pas de nerfs périphériques (nerfs reliant cerveau et moelle épinière aux muscles et aux organes du corps) et les muscles comme voie de communication, mais uniquement les signaux produits par le système nerveux central (cerveau et la moelle épinière). Concrètement, un BCI permet de contrôler par la pensée un ordinateur ou une prothèse sans utiliser ses bras, mains ou jambes.

Les applications de BCI sont principalement concentrées sur les applications médicales visant à remplacer ou à restaurer le fonctionnement du système nerveux central (SNC) perdu à cause d'une maladie ou d'un accident (Hara, 2015). Les BCIs peuvent être utilisés pour diverses applications telles que le remplacement du SNC (Bousseta, 2018), l'évaluation et le diagnostic (Shim, 2016), la thérapie, la rééducation (Mane, 2020), et/ou le calcul affectif. Les BCIs en milieu clinique peuvent également aider à l'évaluation et au diagnostic.

En effet, les patients tétraplégiques ou atteints du locked-in syndrome (le patient pense, mais ne peut bouger que les paupières) se retrouvent paralysés, car leur système nerveux est touché (Ardali, 2019). Porteurs d'une lésion accidentelle de la moelle épinière qui interrompt le transfert de l'information du cortex moteur (centre de décision) vers les centres générateurs de mouvements situés dans la moelle épinière en dessous de la lésion. Cependant, comme ils possèdent encore, pour la plupart, leurs capacités cognitives, le BCI pourrait les aider à retrouver une certaine autonomie.

B. Structure du BCI

Le système BCI fonctionne en boucle fermée : chaque action effectuée par l'utilisateur est suivie d'un retour d'information. Dans notre cas, l'observation sur l'interface graphique d'une forme entraîne une commande de l'ordinateur qui permet l'authentification de l'utilisateur sur son espace personnel.

Le cerveau est constitué de milliards de nerfs qui relient des milliards de synapses pour communiquer. Les processus allant de la prise de signaux du cerveau humain à la transformation en commande exploitable sont illustrés dans la Figure 1 et décrits ci-dessous :

- Acquisition de signal : Dans le cas de la BCI, il s'agit d'un processus de prise d'échantillons de signaux qui mesurent l'activité du cerveau et de les transformer en commandes qui peuvent contrôler une application virtuelle ou réelle.
- Prétraitement : Les signaux collectés du cerveau sont bruyants et altérés par des artefacts. Cette étape aide à nettoyer ce bruit et ces artefacts avec différentes méthodes et filtrages afin d'améliorer les signaux.
- Extraction de caractéristiques : Cette étape implique l'analyse du signal et l'extraction de données. Comme le signal d'activité cérébrale est compliqué, il est difficile d'extraire des informations utiles juste en l'analysant. Il est donc nécessaire d'employer des algorithmes de traitement qui permettent l'extraction des caractéristiques d'un cerveau d'un individu donné.
- Classification : Des techniques de classification des signaux, sans artefacts, permettent d'identifier et d'authentifier la personne devant l'interface graphique visuelle.
- Contrôle des dispositifs : L'étape de classification envoie une commande au dispositif ou à l'application de retour d'information.

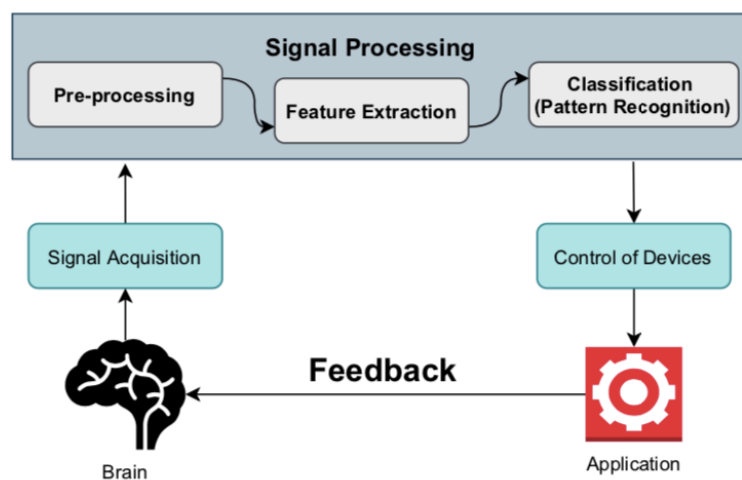


Figure 1: Architecture générale d'un système Brain Computer Interface. (Mridha, 2021)

Un système BCI comprend un système d'acquisition et de traitement des signaux cérébraux, un système de classification et un système de traduction de ces signaux en commande. En effet, l'activité cérébrale générée par le patient est mesurée par des capteurs. Ces signaux sont ensuite transmis à un ordinateur afin de les analyser pour en extraire les données utiles, puis sont transformés en commande pour la machine.

Il existe 3 modes d'enregistrement de données pour réaliser un système BCI (Mridha et al., 2021) :

- (1) Le mode invasif : les électrodes sont implantées directement à l'intérieur du cerveau grâce à une neurochirurgie. Si cette méthode permet d'obtenir des signaux très précis d'une petite population, la procédure reste dangereuse en plus d'être coûteuse.
- (2) Le mode semi-invasif nécessite également l'intervention d'un neurochirurgien, mais dans ce cas les électrodes sont placées sous la dure-mère, la membrane qui entoure le cerveau. Elle permet d'éviter certains bruits dû à l'os du crâne et aux cheveux qui peuvent rendre l'interprétation des signaux plus facile.
- (3) La troisième méthode est la moins coûteuse et la moins dangereuse, elle est dite non invasive et permet l'enregistrement des signaux cérébraux à partir d'électrodes formant un casque que l'on dépose directement sur le crâne. L'invention de la méthode EEG non invasive a révolutionné la recherche et a permis un accès aux signaux cérébraux beaucoup plus accessible pour n'apporte qui, même si la présence de certains bruits peut altérer la qualité du signal (Mridha et al., 2021).

Les signaux cérébraux sont ensuite envoyés à un logiciel externe, qui les classe, les analyse et les interprète. Leur durée, fréquence ainsi que leur répartition dans l'espace peuvent être différents et rendent donc les signaux plus ou moins difficiles à traiter. Un prétraitement et un filtrage sont nécessaires afin de supprimer le bruit de fond. Enfin, une classification est réalisée pour identifier les caractéristiques des signaux pour distinguer les différentes intentions ou actions de l'utilisateur.

1) Électroencéphalogramme (EEG)

Le cerveau humain mature est composé de plus de 100 milliards de cellules de traitement de l'information appelées neurones (Maldonado & Alsayouri, 2023). Les neurones communiquent entre eux par des signaux électriques. L'électroencéphalographie (EEG) est une méthode qui peut être non invasive et relativement peu coûteuse pour mesurer l'activité électrique de ces neurones.

En effet, elle recueille plus précisément les fluctuations de champs électriques produites par l'activité des neurones. C'est une technique d'exploration de la dynamique cérébrale qui permet d'enregistrer l'activité du cerveau avec une remarquable résolution temporelle, très proche de celle du fonctionnement des neurones. Les électrodes EEG peuvent être placées de manière intracrânienne, c'est-à-dire, sous la boîte crânienne, soit à la surface, soit en profondeur du tissu cérébrale comme dit précédemment. Mais elles sont généralement disposées sur le cuir chevelu. Dans ce dernier cas, les électrodes EEG de surface captent un champ électrique produit par les neurones corticaux du cortex, structure située au-dessous du

crâne. Chaque électrode recueille alors l'activité de plusieurs dizaines à centaines de milliers de neurones (Light et al., 2010).

2) Signaux de contrôle du cerveau

Le BCI repose sur l'amplification des signaux du cerveau. Il existe trois groupes de signaux de contrôle : évoqués, spontanés et hybrides. Certains signaux sont plus difficiles à extraire et nécessitent un prétraitement supplémentaire. La classification des signaux de contrôle est présentée dans la Figure 2.

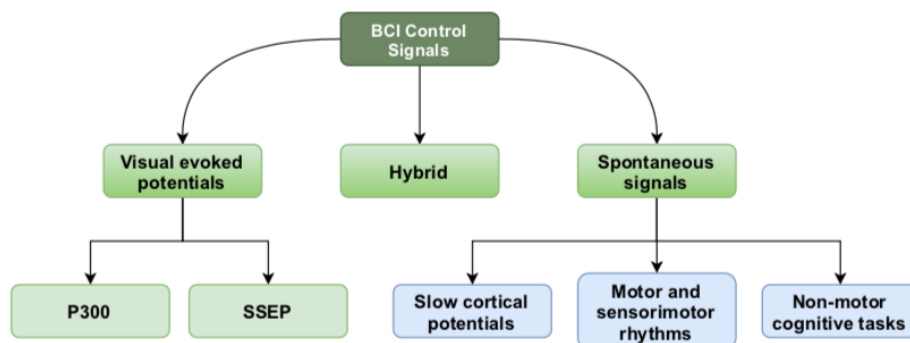


Figure 2 : Architecture générale des systèmes de contrôle des signaux utilisés en BCI. (Mridha, 2021)

Cette image présente une vue d'ensemble détaillée des trois groupes de signaux de contrôle : (1) signaux évoqués (*evoked signals*), (2) signaux spontanés (*spontaneous signals*) et (3) signaux hybrides (*hybrid signals*).

a. Visual Evoked Potentials

Le BCI se base sur l'amplification d'un signal venant directement du cerveau. En fonction de la complexité des signaux, il peut être nécessaire d'appliquer des traitements préliminaires avant l'utilisation des signaux.

Il existe 3 grands types de signaux qui peuvent être extraits du cerveau. La première classe de signaux correspond aux signaux évoqués, ils surviennent en réponse à un stimulus visuel comme un écran qui affiche des formes clignotantes à différentes fréquences. Ce concept est basé sur le fait que lorsque l'utilisateur se concentre visuellement sur un objet clignotant, alors le système informatique conclut que l'utilisateur a sélectionné celui-ci. Pour cette classe-là, une stimulation externe est essentielle à l'obtention du signal évoqué. Cette notion de sélection rend les ondes évoquées très pertinentes pour un modèle d'authentification. C'est pourquoi nous nous intéresserons davantage à cette classe qui compte d'ailleurs deux classes sous-jacentes. L'onde P300 et l'onde SVEP qui seront détaillées dans les parties suivantes.

Il existe aussi la classe des signaux spontanés qui ne nécessitent pas de stimuli extérieurs. En effet, le cerveau est un organe complexe dont l'activité électrique ne s'arrête jamais. En

fonction de si le corps est endormi, au repos ou en activité, différentes fréquences d'ondes peuvent être observées (Mridha et al., 2021).

b. Steady-State Evoked Potentials (SSEP)

Le SSEP signal est produit lorsque le patient est confronté à des stimuli périodiques (fréquence constante) comme une image clignotante, un son modulé ou des vibrations. La force du signal EEG dans le cerveau est supposée augmenter pour répondre à la fréquence du stimulus. C'est-à-dire que l'on observe une augmentation de l'amplitude de l'EEG aux fréquences de scintillement et à leurs harmoniques.

On appelle les ondes provenant d'un stimulus visuel, les ondes SSVEP (*Steady-State Visual Potential*) (İşcan & Nikulin, 2018). Ce sont d'ailleurs ces ondes qui nous seront utiles pour notre projet d'authentification.

c. P300 Evoked Potentials (P300)

Un potentiel évoqué, également connu sous le nom d'ERP (Event Related Potential), est une modification du potentiel électrique du système nerveux en réponse à une stimulation. Une stimulation peut être externe comme un son ou une image, ou interne comme une activité cognitive. On retrouve deux types de potentiels, les potentiels évoqués stables (SSVEP) et les transitoires (P300).

L'onde P300 est une réponse de notre cerveau à des stimuli inattendus. P pour positive et 300 parce qu'elle apparaît 300 ms après le début d'une stimulation. L'amplitude de la réponse P300 est proportionnelle à l'attention consacrée et au degré de traitement de l'information requis. On retrouve deux sous types de P3 : la P3a et la P3b (Polich, 2007).

L'onde P3a survient après 220 à 280 ms et se répartit principalement sur la partie fronto-centrale du cerveau. Elle est provoquée par un effet de surprise. L'onde P3b quant à elle a une latence comprise entre 310 et 380 ms et se répartit principalement centro-pariétalement. Elle se manifeste lorsqu'un stimulus imprévisible survient, mais demande une prise de décision, en lien avec la mémoire.

L'amplitude d'une onde électrique dans le cerveau reflète l'intensité de l'activité cérébrale. Pour l'onde P3a, l'amplitude dépend de la nouveauté du stimulus. Elle diminue donc à mesure que le sujet s'habitue. Cependant, pour la P3b, l'amplitude varie en fonction de la complexité de la tâche perceptive et cognitive. Cela signifie que plus le stimulus est plus difficile à percevoir et plus la réponse nécessite un effort cognitif élevé, plus l'amplitude de l'onde sera importante. L'amplitude de la P3b est aussi influencée par l'état de vigilance, la motivation du sujet et la probabilité de l'apparition du stimulus. Celle-ci augmente également lorsque le cible est émotionnellement significative pour le sujet. Enfin, des facteurs tels que le volume sonore du stimulus ou son intensité lumineuse ainsi que l'épaisseur du crâne du sujet peuvent faire augmenter l'amplitude de la P300 (Mridha et al., 2021).

C. Les caractéristiques des signaux EEG

Afin de réaliser une bonne analyse des signaux EEG, une extraction des caractéristiques est nécessaire. Cela consiste à décrire les signaux par les valeurs pertinentes pour notre classification. Il existe trois types d'extraction : dans le domaine temporel, le domaine fréquentiel ainsi que dans les deux domaines combinés (Mridha et al., 2021).

La caractéristique temporelle est utile lorsqu'on a besoin de déchiffrer les informations rythmiques de l'EEG. Elle décrit comment les signaux varient au cours du temps. Les propriétés du domaine temporel de l'EEG sont simples à corriger. Cependant, il contient des signaux non stationnaires qui s'altèrent dans le temps.

La caractéristique fréquentielle décrit comment la puissance du signal varie. Cette caractéristique est extraite grâce à la densité spectrale de puissance (PSD).

Enfin, la caractéristique temps fréquence est une combinaison des deux précédentes. Son analyse se fait en utilisant *Wavelet transform*.

D. Classification en BCI

Le machine learning ou apprentissage automatique est un domaine scientifique appartenant à une sous-catégorie de l'intelligence artificielle. Il permet de laisser des algorithmes découvrir des caractéristiques communes ou motifs récurrents (patterns) dans des ensembles de données. Ces données peuvent correspondre à des chiffres, des images, des signaux.

En identifiant les patterns dans ces données, les algorithmes apprennent de manière autonome et améliorent leurs performances dans l'exécution d'une tâche spécifique. Après entraînement, le modèle sera donc capable de retrouver les patterns dans de nouvelles données.

La première phase du processus d'utilisation du machine learning consiste à nettoyer et préparer les données avant de les transmettre aux algorithmes d'apprentissage.

Dans un second temps survient la construction du modèle. Dans cette phase nous devons choisir l'algorithme le plus adapté à la problématique et à la nature des données d'apprentissage. Nous testons donc différents algorithmes, les évaluons et essayons d'améliorer au maximum le modèle avec les résultats obtenus avec les données de test. On termine par une phase d'évaluation et de validation du modèle en testant les performances du modèle choisis sur de nouvelles données afin de voir s'il fait les prédictions ou les classements selon les résultats souhaités (Bishop, 2006).

Il existe différents types d'apprentissages automatiques. L'apprentissage supervisé qui apprend à partir d'échantillons de données dont les sorties sont connues et attendues. Il y existe également l'apprentissage non supervisé dont les données initiales ne sont pas annotées, c'est-à-dire qu'on ne connaît pas les résultats attendus. Et enfin l'apprentissage par renforcement ou

le modèle apprendra à partir de l'évaluation des solutions potentielles produites au cours du temps (Bishop, 2006).

Les systèmes BCI sont basés sur le développement de modèles informatiques de classification (voir Figure 3) permettant de reconnaître les motifs récurrents dans les signaux EEG et ensuite de convertir cette authentification en commandes.

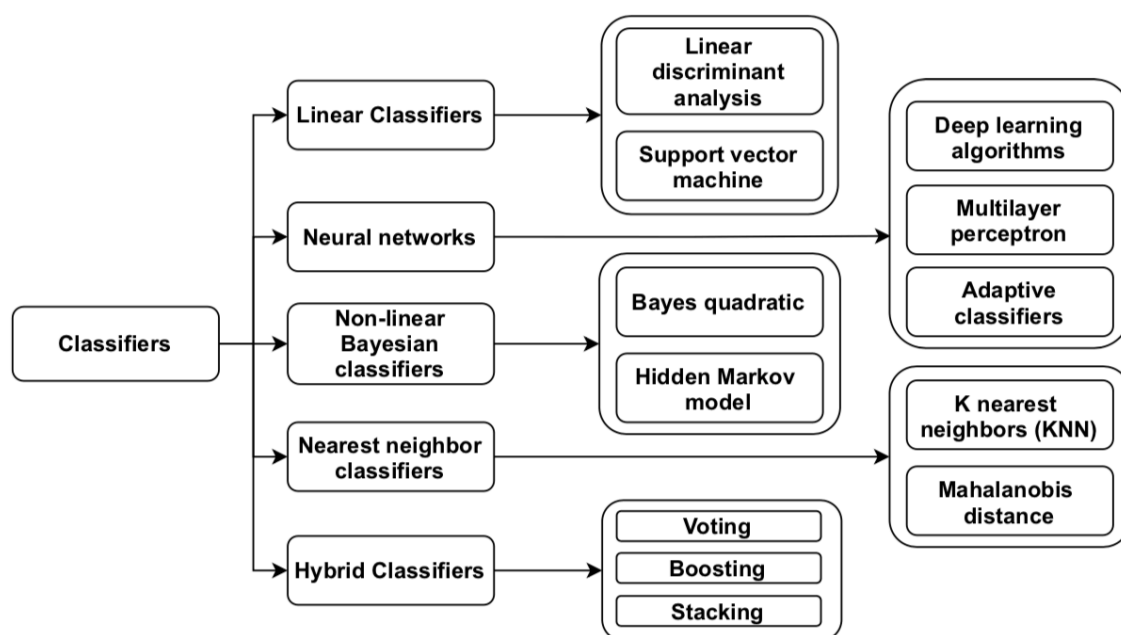


Figure 3 : Algorithmes de classification couramment utilisés en BCI. (Mridha, 2021)

Les classificateurs linéaires et les réseaux de neurones sont deux types de classificateurs les plus souvent utilisés dans les systèmes de BCI. Les classificateurs linéaires, tels que l'analyse discriminante linéaire (Linear Discriminant Analysis) (LDA) et la machine à vecteurs de support (Support Vector Machine) (SVM), utilisent des fonctions linéaires pour séparer les données en classes (Xanthopoulos, 2013), (Schuldt, 2004). LDA est simple à appliquer et produit généralement d'excellents résultats, mais est limité par sa linéarité. SVM permet une généralisation améliorée lors de la maximisation des marges.

Les réseaux de neurones, tels que le perceptron multicouche (MLP), permettent de créer des limites de décision non linéaires (Rosenblatt, 1958). Le MLP est le plus utilisé dans les systèmes BCI, mais d'autres architectures, comme les CNN (*Convolutional Neural Network*), GAN (*Generative Adversarial network*), RNN (*Recurrent Neural Network*) et LSTM (*Long Short-Term Memory*), sont également utilisées pour les BCI basées sur l'EEG. Les CNN sont utilisés pour l'analyse des signaux EEG basés sur des images et ont des applications dans la

détection de la fatigue, la classification des stades de sommeil, la détection du stress, le traitement des données d'imagerie motrice et la reconnaissance des émotions (Mridha, 2021).

Les GAN sont utilisés pour augmenter les données d'entraînement disponibles et peuvent réduire le surajustement et augmenter la précision et la robustesse du classificateur (Kavasidis, 2017). Les RNN ont une forte capacité d'extraction de fonctionnalités temporelles et spatiales et peuvent être combinés avec les CNN pour l'apprentissage de fonctionnalités temporelles et spatiales. Le LSTM est un type de RNN qui peut gérer les dépendances à long terme et est efficace pour les tâches de séries temporelles telles que la reconnaissance de l'écriture et de la voix (Mridha, 2021).

De plus, il existe aussi les classificateurs avec des paramètres adaptatifs avec les données EEG entrantes et peuvent être supervisés ou non supervisés. Les classificateurs bayésiens non linéaires, tels que le Bayes quadratique (*Bayes quadratic*) et le modèle de Markov caché (*Hidden Markov Mode*), utilisent la règle de Bayes pour calculer la probabilité a posteriori d'un vecteur de caractéristiques assigné à une seule classe (Mridha, 2021). Les classificateurs de plus proches voisins, tels que les K plus proches voisins (*K nearest neighbors*) et la distance de Mahalanobis, utilisent des vecteurs de distance pour identifier la classe dominante parmi un point non vu dans l'ensemble de données habitué à l'entraînement (Liu, 2013). Enfin, les classificateurs hybrides combinent plusieurs classificateurs de diverses manières, notamment le renforcement, le vote et la superposition. Ces derniers ont été utilisés avec succès pour classifier l'imagerie motrice et les tâches mentales, mais leur efficacité peut être entravée par leur sensibilité et leur puissance (Dou, 2020).

Partie II : Matériels et Méthodes

Voir le lien Google Drive pour accéder aux différents scripts Python :

<https://drive.google.com/drive/folders/1FiAb82i3u50inzwvYRpKyYcuEkGwRxso?usp=sharing>

A. Acquisition des données

1) L'interface graphique

L'interface graphique a été réalisée avec le langage de programmation Python (version 3.11). Par l'affichage de trois formes de trois couleurs différentes qui clignotent à des fréquences différentes et prédéfinies, on peut stimuler la partie du cortex visuel du cerveau en passant par les yeux de chaque individu. Cela permet d'identifier chaque individu par l'obtention des

signaux EEG distinct entre chaque personne grâce aux ondes SSVEP. On attribue alors une forme et sa fréquence à une personne ; chaque individu doit se concentrer sur la forme qui lui correspond durant l'acquisition. Les caractéristiques des signaux EEG sont supposées être unique pour chaque individu.

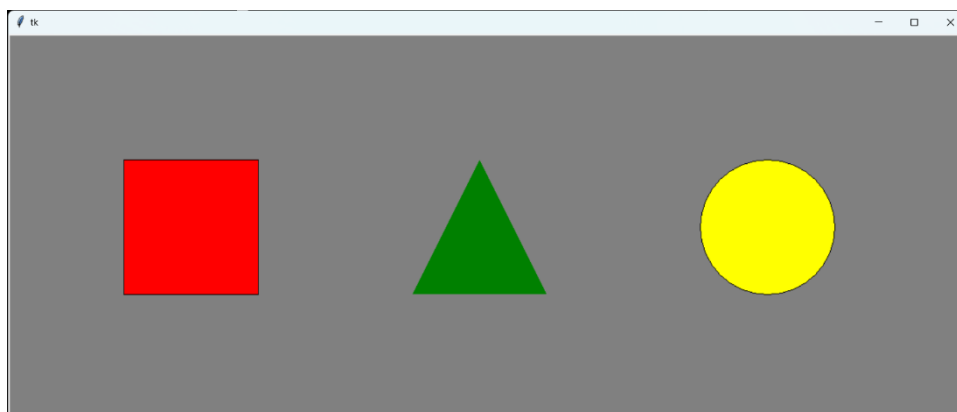


Figure 4 : Interface graphique des 3 formes clignotantes à des fréquences 3,7 et 11 hertz pour la stimulation visuelle.

Dans notre étude, nous présentons trois formes qui clignotent respectivement aux fréquences 7Hz, 11Hz et 13hz sur l'interface graphique. Ces fréquences ne possèdent aucun dénominateur commun, et permettent donc d'être différentiables et d'avoir un impact significatif sur nos signaux EEG.

Le programme a été réalisé en utilisant les fonctions suivantes :

Librairie utilisée : Tkinter

Méthode utilisée : after, canvas

a) Construction de l'interface graphique

Voir annexe 4 pour le code

Les trois fonctions utilisées : *clignotement_rect()*, *clignotement_tri()* et *clignotement_oval()* sont associées à nos trois formes, qui sont respectivement le rectangle, le triangle et le rond, permettent de faire clignoter chaque forme à une fréquence prédéfinie.

Chaque fonction contient deux boucles :

- (1) *if* : va tout d'abord retirer la forme de l'interface si son statut est égal à '*place*', c'est-à-dire, si celle-ci est affichée. Pour cela, nous utilisons la fonction *canvas.delete()*. Le statut de la forme est ensuite changé à '*not_place*' car la forme aura été précédemment supprimée.

(2) *else* : va afficher la forme si son statut est '*not_place*', soit la cachée. Pour afficher les formes, nous utilisons *canvas.create_rectangle()*, *canvas.create_polygon()* ou *canvas.create_oval()* en fonction de la forme souhaitée. Les fonctions prennent en argument les coordonnées des formes, c'est-à-dire, où elles seront placées sur l'interface ainsi que leur couleur de remplissage. Ensuite *Canvas.pack()* va ajouter la forme à l'interface pour pouvoir l'afficher. Enfin, le statut repasse à '*place*'.

Enfin, nous utilisons la fonction *Fenetre.after()* pour que la forme clignote. Cette fonction prend en argument un délai en millisecondes et une fonction à appeler. C'est-à-dire, qu'à chaque fois le délai passé, les fonctions *clignotement_rect()*, *clignotement_tri()* et *clignotement_oval()* vont être appelées et les boucles seront de nouveau exécutées.

b) Affichage de l'interface graphique

La fonction *Tk()* permet de créer en premier lieu la fenêtre d'affichage. Afin d'afficher les formes dans la fenêtre, nous créons un "canva", c'est à dire un objet similaire à la fenêtre capable de recevoir des objets de type "canvas" car c'est le cas pour toutes nos formes. Dans la fenêtre, nous devons créer un canva dans lequel nous pourrions afficher les formes. La fonction *Canvas* prend en argument la fenêtre dans lequel le canva sera affiché, sa taille (largeur et hauteur) ainsi que sa couleur de remplissage. Les coordonnées sont déterminées afin d'avoir un rendu agréable et facile d'utilisation pour l'utilisateur, c'est-à-dire, suffisamment grandes et espacées pour éviter toute confusion. Enfin, les fonctions sont appelées pour lancer le programme.

2) Génération des signaux

Lors de l'acquisition des données, nous avons utilisé le casque pour capter les ondes cérébrales EEG référence : EMOTIV EPOC+ 14-Channel Wireless EEG Headset – EMOTIV (voir Figure 5). Le casque fonctionne en Bluetooth et se connecte à l'ordinateur via une clef USB. Les 14 électrodes captent et transmettent les ondes cérébrales au logiciel sur l'ordinateur.

Pour avoir une acquisition EEG précise, le casque doit être placé sur la tête de l'utilisateur, de façon que les électrodes soient en contact avec le cuir chevelu. L'application EmotivLauncher permet de vérifier la qualité du contact des électrodes et des signaux EEG. Une fois le casque connecté, la qualité du contact est mesurée pour chaque canal et affichée dans la carte des capteurs sous la forme suivante :

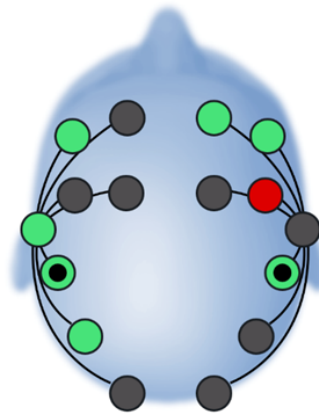


Figure 5 : Carte des capteurs pour contrôler la qualité de la connectivité sur le logiciel EmotivLauncher.

Si l'électrode est colorée en vert, cela signifie qu'elle est correctement placée, en rouge cela veut dire que le contact est mauvais et en noir, qu'aucun contact n'est détecté. Il faut dans un premier temps vérifier que le capteur est bien en contact avec la peau et non avec les cheveux. L'humidité augmente la connectivité des électrodes. Puis dans un second temps, on vérifie la qualité du signal EEG également sur l'application EmotivLauncher.

Le cortex visuel est situé dans le lobe occipital du cerveau, qui se trouve à l'arrière de la tête (Yeo, 2011). Ainsi, la priorité des électrodes fonctionnelles est attribuée à la partie arrière du casque pour capter au mieux les signaux SSVEP qui sont les réponses normales attendues après les stimuli visuels présentés par l'interface graphique.

B. Extraction des données

1) Extraction des signaux

La récupération des données du casque se fait à partir du logiciel OpenVibe (version 3.4.0) à travers le scénario présenté sur la Figure 6. En effet, en configurant les paramètres dans l'application “*OpenVibe Acquisition Server*” nous pouvons enregistrer les données EEG de chaque électrode dans un fichier CSV. Voir le tutoriel complet d'une acquisition d'un casque EPOC+ ci-joint : <https://github.com/CymatiCorp/CyKit/wiki>.

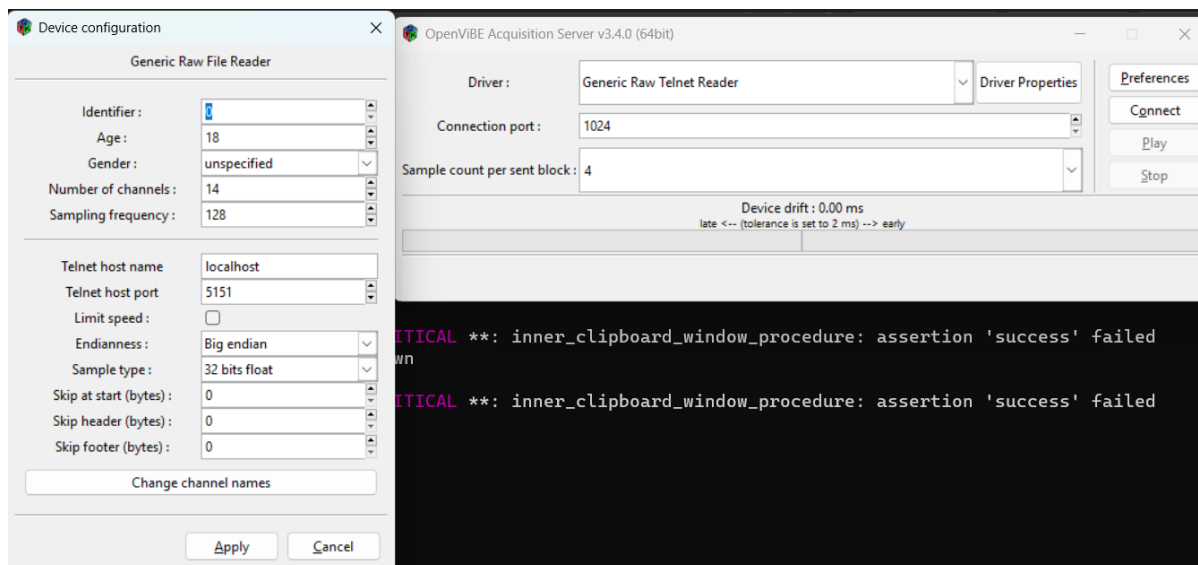


Figure 6 : Paramètre utilisé pour configurer OpenVibe Acquisition Server

Ce fichier présente alors les amplitudes de nos 14 électrodes sur une fréquence de 128Hz, soit un enregistrement toutes les 7,8125 ms. Nous effectuons cette acquisition pendant 20 secondes par essai, ce qui représente en finalité des matrices d'environ 2000*14 de données EEG.



Figure 7 : Scenario de l'acquisition des données sur le logiciel OpenVibe.

Sur le scenario ci-dessus, 3 fonctions sont utilisées : “Acquisition client”, “CSV File Writer”, “Signal display”. La première est obligatoire, elle permet de recevoir le signal EEG et sélectionner le type de périphérique d'acquisition que l'on veut utiliser, dans notre cas le “Signal Stream”.

Par la suite, on a “CSV File Writer” d'un côté et “Signal display” de l'autre. “CSV File Writer” est la fonction permettant d'enregistrer un fichier CSV de nos EEG tandis que “Signal display” nous affiche en temps réel la courbe de nos électrodes durant l'acquisition pour que l'on puisse vérifier à l'œil nu que notre acquisition fonctionne correctement.

C . Classification des signaux

1) Approche générale de Machine Learning

Ayant comme objectif une authentification à partir de la forme clignotante regardée, nous avons donc eu comme but de développer un modèle de classification capable de prédire à partir du signal EEG, quel individu est en train de regarder quelle forme. La classification des signaux a donc été élaborée avec une approche de machine learning pour différencier et donc classer nos signaux EEG en fonction de la personne.

C'est après avoir fait l'acquisition des signaux que nous avons dans un premier temps préparé les données, puis construit différents modèles de classification. Pour finir nous avons évalué les modèles puis testé avec de nouvelles données.

Nous avons donc défini 4 grandes classes de signaux EEG :

- Classe 1 : Signaux EEG quand Antoine regarde la forme ronde (clignotant a une fréquence de 13 Hertz)
- Classe 2 : Signaux EEG quand Marie regarde la forme triangulaire (clignotant a une fréquence de 11 Hertz)
- Classe 3 : Signaux EEG quand Louise regarde la forme rectangulaire (clignotant a une fréquence de 7 Hertz)
- Classe 4 = **Classe anormale** contenant les signaux EEG de Antoine, Marie, Louise regardant autre chose que la forme ou aillant les yeux fermés ainsi que des signaux d'une autre personne (Clara) regardant les 3 formes clignotantes.

Le modèle de machine learning crée est donc capable de prédire à quelle de ces 4 classes appartient un signal EEG nouveau.

2) Préparation des données

Afin d'affiner et de préciser nos résultats nous avons décidé de concentrer l'entraînement de l'algorithme sur les électrodes posées sur l'aire visuelle du cerveau soit la T7 (Temporal gauche), la P7 (Pariétal gauche), la O1 (Occipital gauche), la O2 (Occipital droit), la P8 (Pariétal droit) et la T8 (Temporal droit).

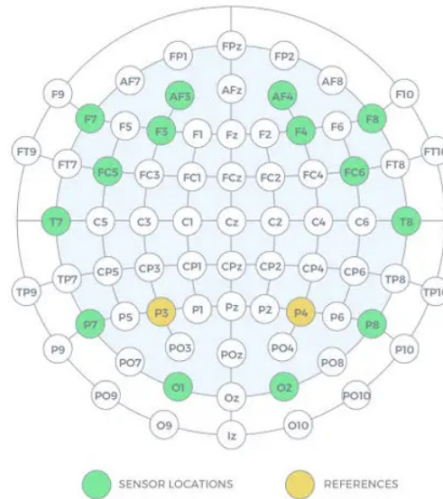


Figure 8: Répartitions des électrodes Système EEG 14 canaux EPOC +

Les colonnes vides ont été supprimées de nos DataFrames.

Enfin, nos quatre DataFrames ont été concaténées en un DataFrame.

3) Construction du modèle

a) Les features

La moyenne est une caractéristique statistique permettant d'obtenir une représentation globale du comportement ou de la tendance des données. Elle est calculée pour chaque électrode de chaque classe.

L'écart-type permet d'évaluer la dispersion des valeurs autour de la moyenne. Plus l'écart-type est grand, plus les valeurs sont éloignées de la moyenne. A l'inverse, si l'écart-type est nul, cela signifie que toutes les valeurs sont les mêmes, soit égales à la moyenne.

Le minimum permet de capturer la valeur la plus basse dans l'ensemble de données. Cela permet de trouver les valeurs extrêmes ou aberrantes.

(<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6684676/>)

b) Les algorithmes de machine learning utilisés

Decision Tree

Un arbre de décision de classification est une technique courante d'exploration de données, considérée comme une forme d'apprentissage automatique supervisé. Il est basé sur une probabilité statistique d'obtenir un résultat spécifique. L'arbre de décision a été construit pour

représenter visuellement et explicitement l'importance des descripteurs qui ont été utilisés pour classer la molécule comme bloqueur ou non au cours du processus de prise de décision de l'algorithme.

Le modèle d'apprentissage automatique est initialisé et ajusté aux données d'apprentissage. Le modèle est entraîné sur l'ensemble des données d'entraînement, puis testé pour prédire une réponse à l'aide de l'ensemble des données de test.

Random Forest

Random Forest est un ensemble d'arbres de décision simples mis en œuvre par Polishchuk et al. (Kuz'min, 2011). L'ensemble génère un nombre spécifique de résultats. Les résultats de tous les arbres sont agrégés pour obtenir "une prédiction finale comme la moyenne des prédictions des arbres individuels" (Zakharov, 2016). En effet, ce modèle utilise une approche de calcul de moyenne pour améliorer la précision de la prédiction et contrôler l'ajustement excessif. Le surajustement implique que le modèle a un domaine d'applicabilité limité en réalisant des performances médiocres avec un ensemble de données avec lequel il n'a pas été formé.

Chaque arbre a été développé comme suit :

- (i) Un échantillon bootstrap obtenu à partir de l'ensemble des données d'apprentissage est entraîné pour former l'arbre actuel. Les molécules non présentes dans l'ensemble d'entraînement de l'arbre actuel sont placées dans un sac de sortie.
- (ii) La division est effectuée en fonction des meilleurs descripteurs sélectionnés au hasard dans l'ensemble initial.
- (iii) L'arbre actuel est développé au maximum sans aucun élagage. La performance du modèle sur l'ensemble hors sac définit la sélection du modèle.

Logistic Regression

La régression logistique est une technique de classification binaire et multi-classes. Elle permet de modéliser la probabilité d'une certaine classe ou d'un événement comme la réussite ou l'échec. Elle est utilisée pour mesurer l'association entre la survenue d'un événement (variable expliquée qualitative) et les facteurs susceptibles de l'influencer (variables explicatives).

La sortie de la régression est une probabilité comprise entre 0 et 1 :

- Si la probabilité est supérieure ou égale à la valeur de seuil (par défaut 0.5) alors la classe positive (étiquetée 1) est prédite
- Si la probabilité est inférieure à la valeur de seuil, la classe négative (étiquetée 0) est prédite (El Sanharawi & Naudet, 2013)

KNN

K plus proches voisins (*K-Nearest Neighbors*) est une méthode non-paramétrique. Elle est basée sur le fait que des entrées semblables devraient avoir des variables cibles semblables. Pour prédire la classe d'une nouvelle donnée, l'algorithme cherche les k voisins les plus proches en utilisant une mesure de distance. Une fois les voisins identifiés, leurs classes sont examinées et la plus fréquente est attribuée comme prédiction. Le résultat pour une classification sera une classe d'appartenance (Larochelle, 2007)..

Classification par vote

Le classificateur de vote permet de combiner différents classificateurs, de les entraîner et de prédire sur la base de l'agrégation leur résultats.

Les critères d'agrégation peuvent être une décision combinée de vote pour chaque résultat de l'estimateur. Les critères de vote peuvent être de deux types :

Hard Voting: Le vote est calculé sur la classe de sortie prédite.

Soft Voting : Le vote est calculé sur la probabilité prédite de la classe de sortie.

Le classificateur de vote s'assure de résoudre l'erreur par n'importe quel modèle. Il permet de d'améliorer les performances (Scikit, nd)..

Hyper-paramètres

Une approche pour optimiser les arbres Random Forest est de passer au réglage des hyper-paramètres. Scikit-Learn implémente un ensemble d'hyperparamètres par défaut pour tous les modèles, mais ces variables et seuils pendant l'apprentissage ne sont pas toujours garantis comme étant optimaux.

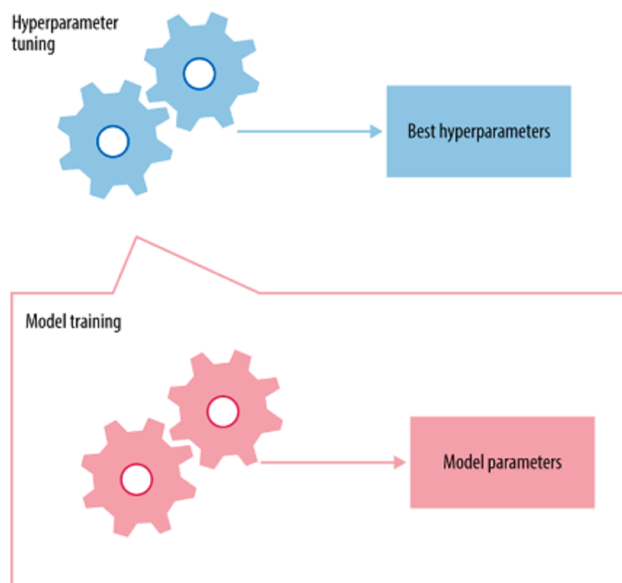


Figure 9 : Approche de l'optimisation des hyper-paramètres pour post-ajustement du modèle Random Forest. (Koehrsen, 2018)

En effet, le réglage des hyperparamètres repose davantage sur les résultats expérimentaux que sur la théorie, et la meilleure méthode pour déterminer les paramètres optimaux consiste donc à essayer de nombreuses combinaisons différentes afin d'évaluer les performances de chaque modèle.

Cependant, l'évaluation de chaque modèle uniquement sur l'ensemble d'apprentissage peut conduire à l'un des problèmes les plus fondamentaux de l'apprentissage automatique : le surajustement.

De plus, en utilisant la méthode `RandomizedSearchCV` de `Scikit-Learn`, nous pouvons définir une grille d'intervalles d'hyperparamètres et échantillonner aléatoirement à partir de cette grille.

Différents hyper-paramètres ont été ajustés :

- (i) `n_estimators` = nombre d'arbres dans la forêt.
- (ii) `min_samples_split` = nombre minimum de points de données placés dans un nœud avant que le nœud ne soit divisé.
- (iii) `min_samples_leaf` = nombre minimum de points de données autorisés dans un nœud feuille.
- (iv) `max_features` = nombre maximal de caractéristiques prises en compte pour la division d'un nœud.
- (v) `max_depth` = nombre maximal de niveaux dans chaque arbre de décision.
- (vi) `bootstrap` = méthode d'échantillonnage des points de données (avec ou sans remplacement).

Par conséquent, la procédure standard d'optimisation des hyperparamètres tient compte du surajustement grâce à la procédure de validation suivante.

Pour l'algorithme KNN, les différents hyperparamètres qui ont été ajustés sont :

- (i) `n_neighbors` = nombre de voisins
- (ii) `Weights` = fonction de pondération

Pour la régression logistique, les différents hyperparamètres qui ont été ajustés sont :

- (i) `C` = l'inverse de la force de régularisation ; doit être un flottant positif
- (ii) `Penalty` = spécifie la norme du penalty

Pour le Decision Tree, les différents hyperparamètres qui ont été ajustés sont :

- (i) `Criterion` = La fonction de mesure de la qualité d'un fractionnement

- (ii) Max_depth = la profondeur maximale de l'arbre
- (iii) Min_samples_split = Le nombre minimum d'échantillons requis pour diviser un nœud interne

4) Procédure de validation

Lorsque nous abordons un problème de machine learning, nous veillons à diviser nos données en un ensemble de formation et un ensemble de test. La validation croisée est utilisée pour éviter le surajustement et pour estimer les performances du modèle sur des données inédites. K-Fold CV, nous divisons notre ensemble d'apprentissage en K sous-ensembles, appelés folds. Le modèle est ensuite ajusté de manière itérative K fois, en entraînant à chaque fois les données sur K-1 des plis et en évaluant sur le Kème Fold, correspondant aux données de validation. À la toute fin de l'entraînement, nous faisons la moyenne des performances sur chacun des plis afin d'obtenir les mesures de validation finales du modèle. En outre, un autre ensemble, appelé ensemble de validation, sélectionne aléatoirement des composés (non utilisés dans les données de formation et de test) pour évaluer la capacité prédictive du modèle en tant qu'évaluation finale (voir la figure X).

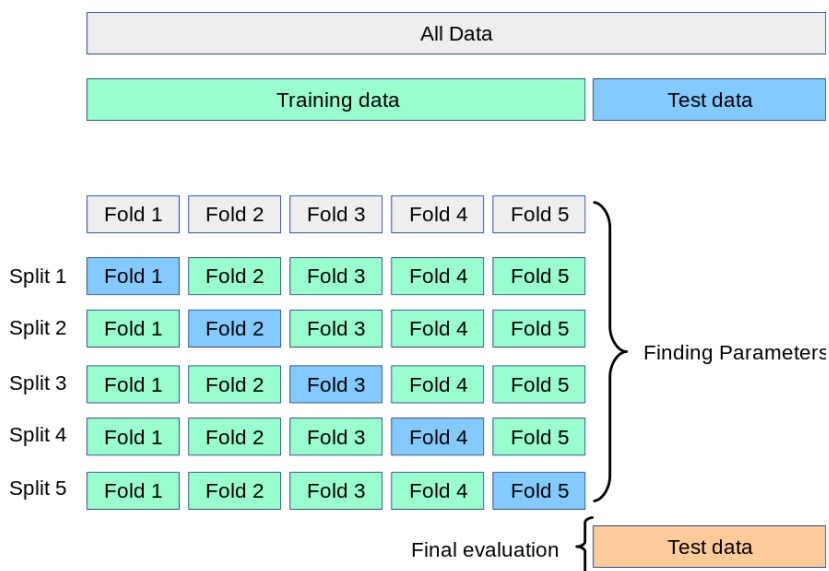


Figure 10 : Représentation schématique de la validation croisée. (Sklearn, 2022)

La procédure de validation croisée nous permet d'évaluer les performances de l'estimateur. En outre, elle décrit également la pratique courante consistant à diviser les données entre un ensemble de données d'apprentissage et un ensemble de données de test. L'objectif est d'évaluer les performances du modèle avec des données non entraînées qui peuvent imiter un scénario réel.

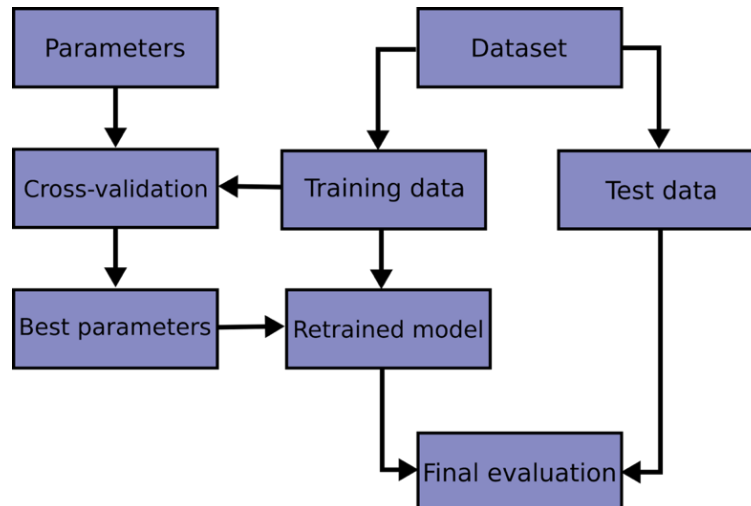


Figure 11 : Représentation schématique d'une procédure de validation croisée. (Sklearn, 2022)

Cette procédure présente la méthodologie de machine learning, y compris l'optimisation des paramètres par validation croisée.

5) Évaluation du modèle

Différentes mesures ont été utilisées pour évaluer les performances des modèles :

(i) Balance Accuracy (BA) évite de gonfler les performances estimées sur des ensembles de données déséquilibrés. Elle est définie comme la moyenne des scores de rappel par classe où chaque échantillon est pondéré en fonction de l'inverse de la prévalence de sa vraie classe.

(ii) Le coefficient de corrélation de Matthew (MCC) est utilisé pour mesurer la qualité des classifications. Il prend en compte les vrais et les faux positifs et négatifs et est généralement considéré comme une mesure équilibrée qui peut être utilisée même si les classes sont de tailles différentes. Un coefficient de +1 représente une prédiction parfaite, 0 une prédiction aléatoire moyenne et -1 une prédiction inverse.

(iii) La matrice de confusion est une matrice mesurant la qualité d'un système de classification dans l'analyse prédictive. Chaque ligne correspond à une classe réelle et chaque ligne à une classe prédite.

Partie III : Présentation et Analyse des Résultats

A. Acquisition des signaux



Name ▲	Value	Class
 tridim_ant	1920x14x2 double	double
 tridim_lou	768x14x2 double	double

Figure 12 : Matrice tridimensionnelle composées des données EEG..

La Figure 8 présente deux matrices EEG qui ont été extraites et transformées sur Matlab lors des premiers essais, une matrice correspondant à l'individu Louise et l'autre à l'individu Antoine.

Lors de la création de ces matrices, il faut que toutes les acquisitions d'une même personne soit de la même taille. Ainsi, dans notre projet, nous avons codé un programme qui analyse la taille de chaque fichier CSV d'un dossier et établit une taille maximale correspondant à la taille du fichier CSV le plus petit. Par la suite, on tronque les fichiers/matrices qui dépassent cette limite afin de pouvoir créer les matrices tridimensionnelles. Vous pouvez observer sur la figure ci-dessus que la limite maximale pour Antoine est de 1920 alors que pour Louise, elle est de 768. C'est une erreur due à un fichier CSV trop petit qui a été enregistré à partir d'une acquisition trop courte.

Pour améliorer cette étape, nous envisageons de changer de méthode et d'agrandir les petites matrices à la taille de la matrice la plus grande en les complétant avec des "NaN". De cette manière nous ne perdrons aucune donnée et il sera très facile de supprimer les "NaN" au besoin.

Voir le lien Google Drive :

<https://drive.google.com/drive/folders/1FiAb82i3u50inzwvYRpKyYcuEkGwRxso?usp=sharing>

Lors de l'acquisition des données avec le casque, nous avons rencontré une première limitation. Comme présenté sur la Figure 5, certaines électrodes (en rouge sur EmotivLauncher) ne captaient aucune onde cérébrale, ce qui diminue qualitativement la somme des signaux que l'on souhaite pouvoir exploiter à des fins d'authentification.

De plus, nous portons pour ce projet, notre intérêt vers les ondes SSVEP issues de l'aire visuelle du cerveau qui est située à l'arrière du crâne, ce qui nécessite une très bonne connectivité pour les électrodes de l'arrière du casque. Or, seulement 4 électrodes sont situées sur la zone du cortex visuel du cerveau et nous n'avons jamais obtenu une captation des 4 électrodes malgré avoir essayés avec plusieurs casques.

Cependant, pour le moment, n'ayant pas encore développé l'algorithme de classification, nous ne sommes pas en mesure de dire si cela sera vraiment conséquent sur notre projet, c'est pourquoi nous considérons cela comme une limitation actuelle.

B. Classification en Machine Learning

1) Prédiction d'une application d'authentification en BCI

Nous avons donc mis en place différents modèles de machine Learning à partir de notre base de données. Le tableau suivant présente les score d'accuracy ou d'exactitudes en pourcentage ainsi que les coefficients de corrélations associés aux modèles.

Nous avons présenté pour chaque modèle les valeurs de testing avant l'ajout des hypers paramètres (HP) ainsi que après l'ajout des hypers paramètres. Nous avons aussi ajouté les valeurs d'accuracy et les coefficients de corrélation trouvés pour la partie évaluation (testing avec un nouveau dataset).

	Decision Tree (DT)	Logistic Regression (LR)	Voting (DT,RF,KNN)	KNN	Random Forest (RF)
Accuracy (ACC)	Sans HP : 50% Avec HP : 97% Evaluation : 96%	Sans HP : 63% Avec HP : 99% Evaluation : 98%	Sans HP : 70% Avec HP : 99% Evaluation : 97%	Sans HP : 55% Avec HP : 99% Evaluation : 98%	Sans HP : 100% Avec HP : 100% Evaluation : 100%
Coefficient de corrélation	Sans HP : 55% Avec HP : 99% Evaluation : 98%	Sans HP : 65% Avec HP : 100% Evaluation : 98%	Sans HP : 73% Avec HP : 100% Evaluation : 97%	Sans HP : 56% Avec HP : 100% Evaluation : 98%	Sans HP : 100% Avec HP : 100% Evaluation : 100%

Figure 13: Tableau récapitulant les résultats obtenus sur différents modèles de machine learning avec et sans hyper paramètres ainsi que pour l'évaluation sur une nouvelle base de données

Le modèle Decision Tree, sans hyperparamètres optimisés, présente une précision relativement faible de 50%. Cependant, après avoir ajusté les hyperparamètres, la précision s'améliore considérablement pour atteindre 97% lors de l'évaluation. Cela suggère que la recherche et l'optimisation des hyperparamètres sont essentielles pour obtenir de meilleures performances de prédiction avec ce modèle.

La régression logistique, sans hyperparamètres optimisés, présente une précision de 63%. Cependant, après avoir ajusté les hyperparamètres, la précision augmente considérablement pour atteindre 99% lors de l'évaluation. Ces résultats indiquent que la régression logistique, avec une optimisation adéquate, peut être très performante dans la prédiction.

Nous constatons de manière générale une amélioration des modèles avec l'ajout des hyperparamètres. En effet, l'exactitude ou la précision de chaque modèle a atteint entre 97 et 100 % grâce aux hyperparamètres.

Le modèle de vote (Voting), en combinant les décisions de l'arbre de décision (Decision Tree), de la forêt aléatoire (Random Forest) et des k-plus proches voisins (KNN), présente une précision de 70% sans hyperparamètres optimisés. Cependant, après avoir ajusté les hyperparamètres, la précision atteint 99% lors de l'évaluation. Cela démontre l'efficacité du modèle de vote dans l'amélioration des performances prédictives grâce à l'agrégation des décisions de plusieurs modèles.

Le modèle des k-plus proches voisins (KNN), sans hyperparamètres optimisés, présente une précision relativement faible de 55%. Cependant, après avoir ajusté les hyperparamètres, l'accuracy s'améliore significativement pour atteindre 99% lors de l'évaluation. Ces résultats soulignent l'importance de l'optimisation des hyperparamètres pour obtenir de bonnes performances avec le modèle KNN.

Le modèle de forêt aléatoire (Random Forest), sans hyperparamètres optimisés, présente une précision parfaite de 100%. Même après l'optimisation des hyperparamètres, la précision reste à 100% lors de l'évaluation. Cela suggère que le modèle Random Forest est très performant et qu'il est capable de capturer des relations complexes dans les données.

En termes de coefficient de corrélation, les résultats montrent des tendances similaires à ceux de l'accuracy. Les modèles avec hyperparamètres optimisés présentent généralement une corrélation plus élevée que les modèles sans hyperparamètres optimisés. Cela indique que l'ajustement des hyperparamètres contribue à améliorer la corrélation entre les prédictions et les vraies valeurs.

En résumé, les résultats des différents modèles de machine learning montrent des améliorations significatives de précision après l'optimisation des hyperparamètres. Les modèles tels que la régression logistique, la forêt aléatoire et le modèle de vote présentent les performances les plus élevées, avec des scores de précision atteignant 99% voire 100% lors de l'évaluation. Ces résultats mettent en évidence l'importance de l'optimisation des hyperparamètres pour obtenir de bonnes performances de prédiction.

Différentes matrices de corrélations ont été obtenue, voici la matrice de corrélation obtenue avec l'algorithme de Random Forest présentant une authentification de 100% :

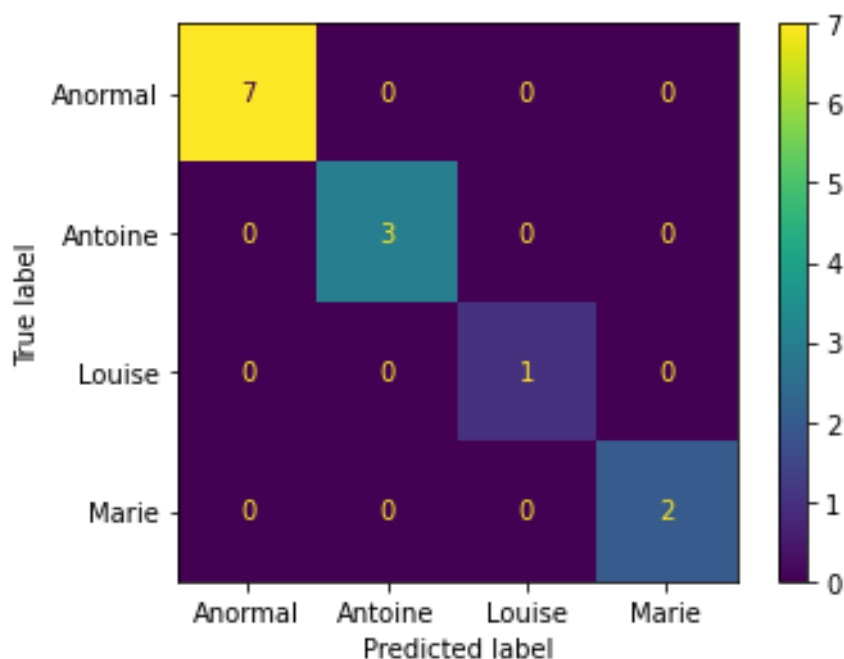


Figure 123: Meilleure matrice de confusion obtenu avec 100 % d'accuracy sur l'algorithme Random Forest

2) Généralisation du modèle

Notre modèle montre une forte capacité de généralisation qui est essentielle pour une application d'authentification réussie. Nous pouvons observer avec nos datasets d'évaluation des scores d'accuracy de 100% ce qui indique que le modèle est capable de faire des prédictions précises sur des nouvelles données. Nous avons utilisé en particulier l'algorithme de RandomForest qui a montré une excellente performance en termes de généralisation.

Ces résultats suggèrent que notre modèle est capable de reconnaître les caractéristiques clés d'authentification, même lorsque des données inconnues lui sont présentées. Cela nous apporte une confiance supplémentaire dans l'application d'authentification, renforçant sa fiabilité et son utilité dans divers contextes.

C. Discussion

Lors de notre projet d'authentification en BCI (Brain-Computer Interface), nous avons identifié plusieurs axes d'amélioration importants. Ces axes se concentrent sur l'amélioration de l'interface graphique, l'élargissement du dataset avec plus d'utilisateurs, l'utilisation de modèles de deep learning pour améliorer les prédictions et la fiabilité des modèles.

Tout d'abord, l'interface graphique joue un rôle crucial dans l'expérience utilisateur lors de l'authentification en BCI. Actuellement, notre interface graphique présente un ensemble limité de formes. Pour améliorer cette interface, nous pourrions envisager d'intégrer plus de formes et de motifs, offrant ainsi aux utilisateurs une plus grande variété de choix pour interagir avec le système. Par exemple, en ajoutant des formes géométriques complexes, des icônes personnalisées ou des éléments visuels attrayants, nous pouvons améliorer l'expérience globale de l'interface graphique et faciliter la sélection des options par les utilisateurs.

En ce qui concerne le dataset de nos données d'acquisition EEG, il est essentiel d'élargir notre échantillon d'utilisateurs. Actuellement, nous avons recueilli des données provenant d'un nombre limité de sujets (4 utilisateurs). L'ajout de plus d'utilisateurs à notre dataset nous permettra d'obtenir une représentation plus diversifiée des signaux cérébraux et de mieux généraliser notre modèle d'authentification. Plus le dataset est large et diversifié, plus notre modèle sera capable de s'adapter à différentes caractéristiques individuelles et de fournir des résultats fiables et précis.

Ensuite, l'utilisation de modèles de deep learning peut apporter des améliorations significatives à notre projet d'authentification en BCI (Xiang, 2018). Les modèles de deep learning ont démontré leur capacité à extraire des caractéristiques complexes à partir de données brutes et à améliorer les performances prédictives. En utilisant des architectures de réseaux neuronaux profonds spécifiquement conçues pour le traitement des signaux cérébraux, nous pourrions

optimiser nos modèles d'authentification et obtenir des prédictions plus précises. De plus, les modèles de deep learning peuvent également être utilisés pour la sélection automatique des features, ce qui permettra de réduire la dimensionnalité des données et d'éliminer les caractéristiques redondantes ou moins informatives, améliorant ainsi l'efficacité et la précision des modèles.

Enfin, nous avons identifié deux datasets en ligne et public qui pourraient être intégrés à notre propre dataset (İşcan, 2018), (Liu, 2020). Ces datasets en ligne offrent une opportunité unique d'augmenter la taille et la variété de notre ensemble de données, car ils sont collectés auprès d'un large éventail d'utilisateurs provenant de différentes populations et environnements. En fusionnant ces datasets avec le nôtre, nous pourrions accroître la représentativité de notre modèle et renforcer sa capacité à s'adapter à des scénarios réels et diversifiés.

Conclusion

Les activités cérébrales via l'électroencéphalogramme (EEG) peuvent être exploitées pour authentifier une personne via une application de Brain Computer Interface. Dans le cadre de ce projet, l'objectif principal est la mise en œuvre d'une application de BCI à base de la technique d'authentification visuelle SSVEP pour une authentification EEG.

Notre projet d'authentification EEG basé sur la technique d'authentification visuelle SSVEP, en utilisant un modèle de Random Forest qui a atteint un taux de précision de 100%. Ce projet a démontré que les activités cérébrales capturées par l'électroencéphalogramme (EEG) peuvent être exploitées de manière fiable pour authentifier une personne via une application de Brain Computer Interface (BCI).

Ce projet ouvre de nouvelles perspectives pour l'authentification basée sur les activités cérébrales, offrant des avantages en termes de sécurité et de commodité pour les utilisateurs. En exploitant les potentiels de l'EEG, il devient possible d'authentifier les individus sans recourir à des méthodes traditionnelles telles que les mots de passe ou les empreintes digitales.

Bibliographie

- Ardali, M. K., Rana, A., Purmohammad, M., Birbaumer, N., & Birbaumer, N. (2019). Semantic and BCI-performance in completely paralyzed patients : Possibility of language attrition in completely locked in syndrome. *Brain and Language*, 194, 93-97.
<https://doi.org/10.1016/j.bandl.2019.05.004>
- Bousseta, R.; El Ouakouak, I.; Gharbi, M.; Regragui, F. EEG based brain computer interface for controlling a robot arm movement through thought. *Irbm* 2018, 39, 129–135.
- Dou,J.;Yunus,A.P.;Bui,D.T.;Merghadi,A.;Sahana,M.;Zhu,Z.;Chen,C.W.;Han,Z.;Pham,B.T.I mprovedlandslideassessment using support vector machine with bagging, boosting, and stacking ensemble machine learning framework in a mountainous watershed, Japan. *Landslides* 2020, 17, 641–658.
- Fahredden Sadikoglu and Selin Uzelaltinbulat. 2016. Biometric Retina Identification Based on Neural Network. *Procedia Computer Science* 102 (2016), 26–33.
- Geof H Givens, J Ross Beveridge, Yui Man Lui, David S Bolme, Bruce A Draper, and P Jonathon Phillips. 2013. Biometric face recognition: from classical statistics to future challenges. *Wiley Interdisciplinary Reviews: Computational Statistics* 5, 4 (2013), 288–308.
- Hara, Y. Brain plasticity and rehabilitation in stroke patients. *J. Nippon. Med Sch.* 2015, 82, 4–13.
- İşcan, Z., & Nikulin, V. V. (2018). Steady state visual evoked potential (SSVEP) based brain-computer interface (BCI) performance under different perturbations. *PLoS ONE*, 13(1), e0191673. <https://doi.org/10.1371/journal.pone.0191673>

- JA Unar, Woo Chaw Seng, and Almas Abbasi. 2014. A review of biometric technology along with trends and prospects. *Pattern recognition* 47, 8 (2014), 2673–2688.
- Joadder MAM, Myszewski JJ, Rahman MH, Wang I. A performance based feature selection technique for subject independent MI based BCI. *Health Inf Sci Syst.* 2019 Aug 7;7(1):15. doi: 10.1007/s13755-019-0076-2. PMID: 31428313; PMCID: PMC6684676.
- Kavasidis,I.;Palazzo,S.;Spampinato,C.;Giordano,D.;Shah,M.Brain2image:Convertingbrainsignalsintoimages.InProceedings of the 25th ACM international conference on Multimedia, Mountain View, CA, USA, 23–27 October 2017; pp. 1809–1817.
- Kuz'min, V. E., Polishchuk, P. G., Artemenko, A. G., & Andronati, S. A. (2011). Interpretation of QSAR models based on Random Forest methods. *Molecular Informatics*, 30(6–7), 593–603. <https://doi.org/10.1002/minf.201000173>
- Larochelle, H. (2007). *Algorithme des k plus proches voisins*. http://www.iro.umontreal.ca/~dift3395/demo_1/theorie/kppv.pdf
- Leonard J Trejo, Karla Kubitz, Roman Rosipal, Rebekah L Kochavi, and Leslie D Montgomery. 2015. EEG-based estimation and classification of mental fatigue. *Psychology* 6, 05 (2015), 572.
- Light, G. A., Williams, L. E., Minow, F., Sprock, J., Rissling, A., Sharp, R., Swerdlow, N. R., & Braff, D. L. (2010). Electroencephalography (EEG) and event-related potentials

- (ERPs) with human participants. *Current Protocols in Neuroscience*, Chapter 6, Unit 6.25.1-24. <https://doi.org/10.1002/0471142301.ns0625s52>
- Liu B, Huang X, Wang Y, Chen X, Gao X. BETA: A Large Benchmark Database Toward SSVEP-BCI Application. *Front Neurosci.* 2020 Jun 23;14:627. doi: 10.3389/fnins.2020.00627. PMID: 32655358; PMCID: PMC7324867.
- Liu,C.;Wang,H.;Lu,Z.EEGclassificationformulticlassmotorimageryBCI.InProceedingsofthe201325thChineseControl and Decision Conference (CCDC), Guiyang, China, 25–27 May 2013; pp. 4450–4453.
- Maldonado, K. A., & Alsayouri, K. (2023). *Physiology, Brain*. In StatPearls [Internet]. StatPearls Publishing. <https://www.ncbi.nlm.nih.gov/books/NBK551718/>
- Mane, R.; Chouhan, T.; Guan, C. BCI for stroke rehabilitation: Motor and beyond. *J. Neural Eng.* 2020, 17, 041001.
- Mridha, M. F., Das, S. C., Kabir, M. M., Lima, A. A., Islam, Md. R., & Watanobe, Y. (2021). Brain-Computer Interface: Advancement and Challenges. *Sensors*, 21(17), 5746. <https://doi.org/10.3390/s21175746>
- Polich, J. Updating P300: An integrative theory of P3a and P3b. *Clin. Neurophysiol.* 2007, 118, 2128–2148.
- Rosenblatt,F.Theperceptron:Aprobabilisticmodelforinformationstorageandorganizationinthebrain.*Psychol.Rev.*1958, 65, 386.
- Sanharawi, M. E., & Naudet, F. (2013). Comprendre la régression logistique. *Journal Francais D Ophtalmologie*, 36(8), 710–715. <https://doi.org/10.1016/j.jfo.2013.05.008>

Schuldt, C.; Laptev, I.; Caputo, B. Recognizing human actions: A local SVM approach. In Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, 26 August 2004; Volume 3, pp. 32–36.

Shim, M.; Hwang, H.J.; Kim, D.W.; Lee, S.H.; Im, C.H. Machine-learning-based diagnosis of schizophrenia using combined sensor-level and source-level EEG features. *Schizophr. Res.* 2016, 176, 314–319.

Steven Goldstein. 2016. Methods and systems for voice authentication service leveraging networking. (March 8 2016). US Patent 9,282,096

Sklearn. (n.d.). 3.1. Cross-validation: evaluating estimator performance. Scikit-Learn. Retrieved December 26, 2022, from https://scikit-learn.org/stable/modules/cross_validation.html

Sklearn. (n.d.). 3.1; sklearn.ensemble.VotingClassifier. (n.d.). Scikit-learn. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html>

Xanthopoulos, P.; Pardalos, P.M.; Trafalis, T.B. Linear discriminant analysis. In *Robust Data Mining*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 27–33.

Yeo, B. T., Krienen, F. M., Sepulcre, J., Sabuncu, M. R., Lashkari, D., Hollinshead, M. O., Roffman, J. L., Smoller, J. W., Zöllei, L., Polimeni, J. R., Fischl, B., Liu, H., & Buckner, R. L. (2011). The organization of the human cerebral cortex estimated by intrinsic functional connectivity. *Journal of Neurophysiology*, 106(3), 1125–1165. <https://doi.org/10.1152/jn.00338.2011>

- Xiang Zhang, Lina Yao, Xianzhi Wang, Jessica Monaghan, and David McAlpine. 2018. A Survey on Deep Learning based Brain Computer Interface: Recent Advances and NewFrontiers. 1,1, Article1 (January2018), 66 pages.DOI: 10.1145/1122445.1122456
- Zhang, X., Yao, L., Kanhere, S. S., Liu, Y., Gu, T., & Chen, K. (2018). MindID. Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies, 2(3), 1-23. <https://doi.org/10.1145/3264959>
- Zakharov, A. V., Varlamova, E. V., Lagunin, A. A., Dmitriev, A. V., Muratov, E. N., Fourches, D., Kuz'min, V. E., Poroikov, V. V., Tropsha, A., & Nicklaus, M. C. (2016). QSAR modeling and prediction of drug-drug interactions. Molecular Pharmaceutics, 13(2), 545–556. <https://doi.org/10.1021/acs.molpharmaceut.5b00762>

Annexes



Annexe 1 : Casque EEG : Système EEG 14 canaux EPOC+

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Projet BCI-EEG (Année 2023-2024)	Mars				Avril				Mai				Juin	
TACHES	Semaine 1	Semaine 2	Semaine 3	Semaine 4	Semaine 1	Semaine 2	Semaine 3	Semaine 4	Semaine 1	Semaine 2	Semaine 3	Semaine 4	Semaine 1	Semaine 2
Recherches bibliographiques sur le sujet														
Développement de l'interface graphique														
Reflexion sur le dataset (quantité de données ? Contrôles ?)														
Acquisition des données avec le casque														
Transformation des données acquises en données exploitables pour le machine learning (matlab)														
Développement du code python pour le machine learning et recherches associées														
Optimisation du machine learning et tests														
Rédaction du rapport (partie Etat de l'art)														
Rédaction du rapport (partie matériel et méthode)														
Rédaction du rapport (conclusion et perspectives)														
Réunions	Rencontre et planification du projet		Réunion mise en commun des recherches		2 réunions via teams	1 réunion	2 réunions pour acquisitions des données	1 réunion	2 réunions de mise au point					

Annexe 2 : Diagramme de Gantt représentant notre organisation

[illegible]

Annexe 3 : Code Matlab permettant la création de matrice Tridimensionnelle à partir de fichier CSV EEG