Establishing an automated pipeline to select an AI model for brain-computer interfaces (BCIs)

Kaveh Zabihi
663036

Arnhem,
Master Engineering Systems

FRONTMATTER

Student: Kaveh Zabihi

Project title: Establishing automated pipeline to select AI model for brain-computer interfaces (BCIs)

Company: Sogeti

Supervisors: Shahryar Masoumi, Marith Dieker

Date: February 2023

Sogeti	HAN university
Shahryar Masoumi	Marith Dieker
< place >, date: March 2023	Arnhem, date:

Table of Contents

Figures

Tables

1 BACKGROUND

In recent decades, the means of interaction between individuals and computers have undergone significant development. While keyboards, mouses, touchscreens, and even voices are currently the most common methods for interacting with computers, this study is specifically concerned with brain-activated controls to read directly from the brain, which are also referred to as Brain Computer Interfaces(Vidal', 1973). BCIs are devices that can read and interpret brain activity and translate it into commands that can be used to control a machine or an application. These interfaces have the potential to revolutionize the way we interact with technology in gaming, entertainment, and particularly for individuals with disabilities or those who are unable to use traditional means of interaction.

This project is conducted within Sogeti (Vianen, The Netherlands), and in the research and development department (SogetiLabs). Their goal is to build a proof-of-concept to inspire the company's employees and demonstrate the company's capabilities in the field of brain signal processing to clients. Ultimately, they want to create a game inspired by Atari's classic 'Breakout' that can be controlled by the player's thoughts of moving left or right.

This study aims to improve the functionality of the BCI system by implementing an automated Machine Learning (ML) pipeline instead of ensemble model (previous model) that can select the most appropriate AI model for the EEG-based BCI system. In addition, a new EEG helmet has been added to the system, which requires proper connectivity to the automated ML pipeline.

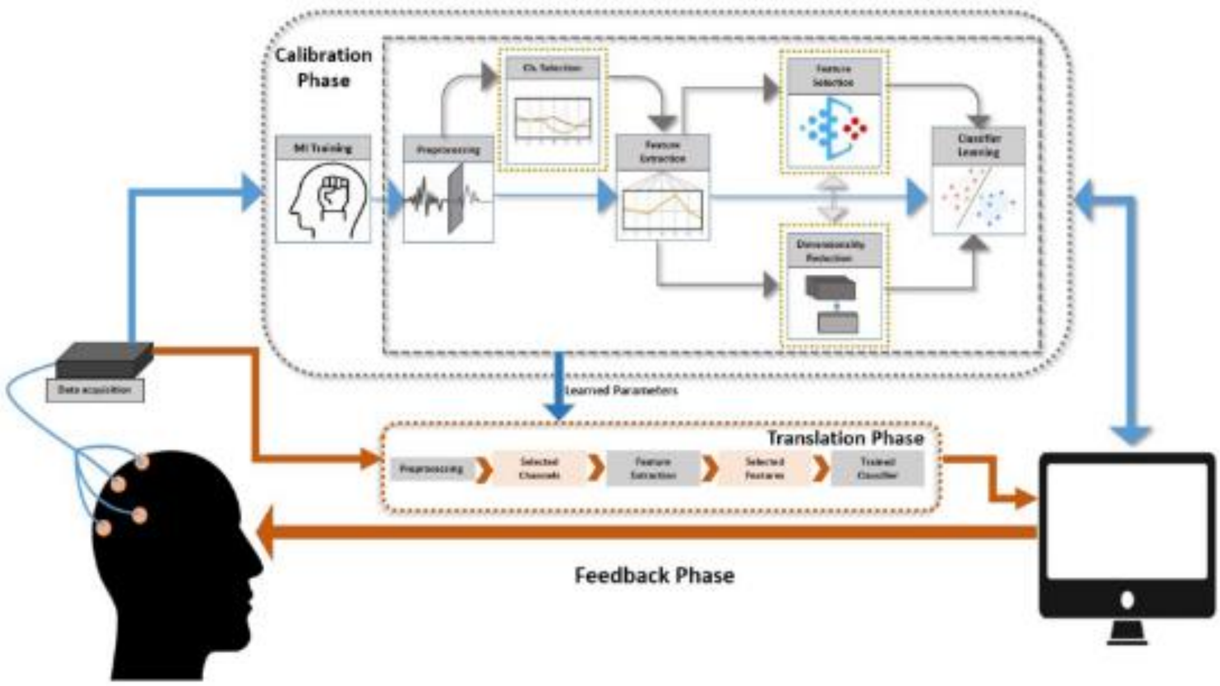

Figure 1. Representing the typical structure of MI-based brain-computer interfaces (BCI)(Singh, Hussain, Lal, & Guesgen, 2021)

A BCI application requires a hardware device to record data from the brain, and a model includes features extraction, features selection, and classification. In the following section, each of these components will be described based on previous research and works in the field.

1.1 Problem Definition

The goal of this project is to establish an automated ML pipeline to select the most appropriate AI model for the EEG-based BCI. Additionally, a new EEG helmet has been added to this project, require to be connected appropriately to the Automated ML pipeline.

1.2 Project objective(s)

The objectives of this project are to Implement an automated ML pipeline that can evaluate and compare multiple ML algorithms for BCI systems using the preprocessed EEG data, with a focus on optimizing the training time at the beginning of the game and improving the accuracy, which is currently at 55%. This involves connecting the new EEG helmet to record proper brain data.

1.3 Research Questions

What is the best-automated ML pipeline selection method to optimize training time at the beginning of the game and increase model accuracy? vering the above question requires the following sub-questions:

1. What modifications can be made to the code repository to accommodate the new EEG helmet (ThinkPulse) sensor configuration?
2. What is the best algorithm for automated ML model selection to improve functionality machine learning algorithms in the BCI repository?
3. What are the steps to implementing a preprocessing method for feature extraction, feature selection, and AI models of the BCI repository in an automated ML pipeline?
4. How to evaluate the accuracy and the robustness of the Automate ML pipeline?

2 LITERRATURE REVIEW

This section presents a review of relevant literature related this project. It begins by introducing the background of brain computer interfaces and electroencephalography, and then outlines their current uses. The section then delves into various methods of feature extraction, feature selection, and classification that have been employed in previous EEG studies. Finally, it concludes by discussing various automated machine learning pipelines.

2.1 Brain Computer Interfaces

BCIs come in two types: invasive and non-invasive. Invasive BCIs involve surgery to place sensors for measuring brain activity underneath the scalp. Although this method provides a high signal-to-noise ratio, there are risks associated with surgery, and once the sensors are in place, they cannot be moved to measure other brain areas. Non-invasive BCIs, on the other hand, rely on sensors placed on or near the scalp, eliminating the need for surgery (Martini et al., 2020). One of the most used methods is Electroencephalography (EEG) due to its cost-effectiveness compared to other techniques such as functional magnetic resonance imaging (fMRI) and magnetoencephalography (MEG). It is predicted that EEG-based BCIs, particularly those using motor imagery (MI), will become cutting-edge technologies in both the clinical and entertainment sectors (Padfield et al., 2019). EEGs measure the brain activity that comes from neurons in the neocortex, a thin layer of cells that is 2-5mm thick. The human brain has approximately 86 billion neurons, and these neurons communicate through synapses. A synapse involves two neurons, a pre-synaptic neuron, and a post-synaptic neuron. The pre-synaptic neuron is located before the synapse and sends signals to the post-synaptic neuron. The communication between these neurons happens through a chemical process in which neurotransmitters are passed between the cells(Portillo-Lara et al., 2021).

The electrodes cannot detect signals from single neurons due to their small size. Therefore, the activity of thousands of neurons needs to be recorded simultaneously to observe any signal. EEG charts represent the intensity of voltage waves of a group of neurons measured by each sensor. When the brain is performing a task, the amplitude of the waves is lower, and the voltage waves become smaller until the task is completed. Brainwaves are measured in hertz, which indicates the number of cycles per second. These studies categorized frequencies of brainwaves into four different bandwidths (Portillo-Lara et al., 2021), five different bandwidths(Garcia-Moreno et al., 2020), and six different bandwidths (Cano et al., 2022).

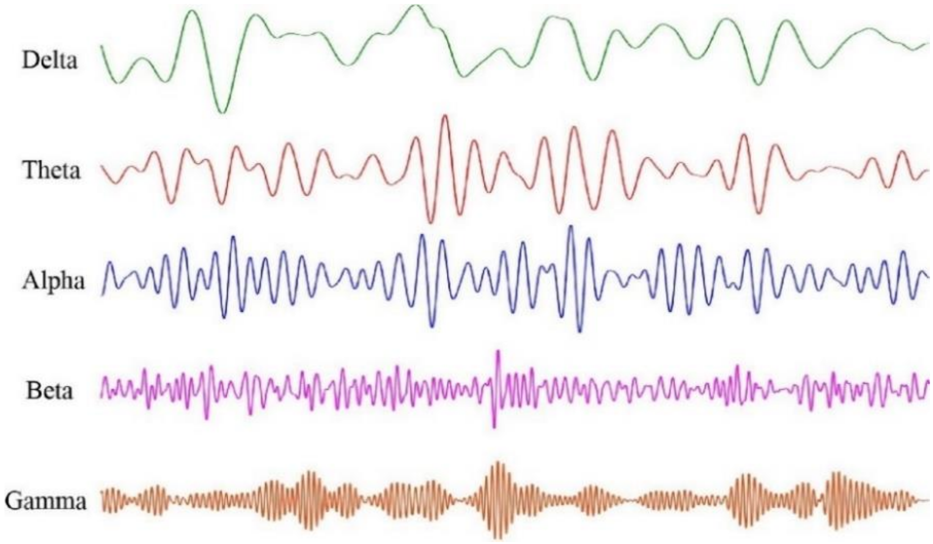

Figure 2. Frequencies of brainwaves into five different bandwidths (the delta (0.5–4 Hz), theta (4–8 Hz), alpha (8–13 Hz), beta (13–30 Hz), and gamma (>30 Hz))(Wang & Wang, 2021).

2.2 Feature Extraction

Feature extraction is a critical step in the analysis of EEG signals for BCI tasks. The goal of feature extraction is to transform the raw EEG data into a set of features that can be used for classification, such as the presence or absence of a specific movement. The most used methods for feature extraction can be classified into four groups: time-domain, frequency-domain, time-frequency analysis, and spatial-domain(Padfield et al., 2019). The time-domain method stands for the electrical activity of neurons in voltage per unit of time. The frequency-domain method quantifies the amount of activity within a specific frequency range over a certain timeframe. The time-frequency analysis method analyses the signal in both the time and frequency domain simultaneously, providing a more complex representation of the data. In contrast, the spatial-domain method includes the locations of the sensors in the analysis, which allows for a more comprehensive understanding of the spatial distribution of the neural activity (Singh, Hussain, Lal, & Guesgen, 2021).

2.3 Feature Selection

Commonly used algorithms for feature selection and dimensionality reduction include principal component analysis (PCA), independent component analysis (ICA), sequential forward and backward searches. Evolutionary algorithms such as particle swarm optimization, differential evolution, and artificial bee colony optimization have also been recently explored by researchers in BCI applications. Most common feature selection techniques have limitations: despite having a good variance, classification accuracy is often poor, perhaps because of redundant features that are not removed by basic feature selection techniques. In simple feature extraction techniques, features are typically linearly transformed without considering the classifier stage, while evolutionary algorithms have demonstrated success in tasks with a large feature search space (Baig et al., 2017).

(Kevric & Subasi, 2017) suggested that nonlinear filtering methods such as multiscale principal component analysis (MSPCA) are a preferable alternative as they could remove noise effectively while preserving sharp transitions rather than linear approaches. Additionally, they improved MSPCA's classification accuracy by using the nonlinear methods instead of the liner methods.

2.4 Classification Methods

Classification in EEG-based BCI involves categorizing EEG signal patterns into classes that correspond to different cognitive or motor tasks. Accurate classification is crucial for the effective operation of EEG-based BCI applications, allowing users to control external devices or communicate with the environment using their brain activity.

Commonly used methods for classification are Support Vector Machine (SVM), k-Nearest Neighbor (KNN), Logistic Regression (LR), and Naive Bayes (NB) in machine learning algorithms. These methods have a classification accuracy that can vary between 57.50% to 95.70%. In comparison, deep learning algorithms such as Neural Network (NN), Long and Short-Term Memory (LSTM), and Deep Belief Network (DBN) have classification accuracy ranging from 63.38% to 97.56%(Wang & Wang, 2021).

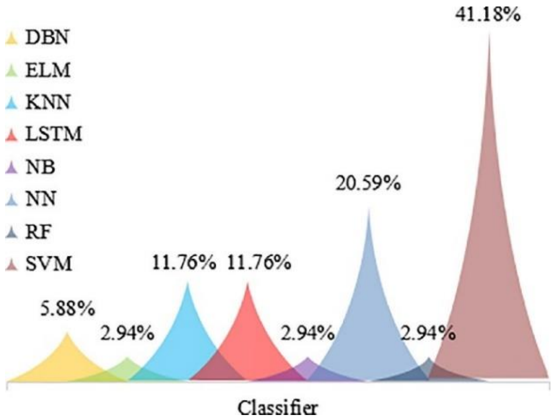

Figure 3. Methods of the emotional classification using EEG signals(Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Random Forest (RF), Neutral Network (NN), Naive Bayes (NB), Long and Short Term Memory (LSTM), Extreme Learning Machine (ELM)).(Wang & Wang, 2021).

Among the most widely used algorithms are SVM and its variants, such as MC-LSSVM, LSSVM, etc., accounting for 41.18%, while the most frequently used kernel functions are radial basis functions (RBF), linear, Gaussian, etc.(Wang & Wang, 2021).

Table 1. presented by (Garcia-Moreno et al., 2020) provides an extensive overview of studies conducted on MI data. The table highlights that LSTM, SVM, and LR have achieved accuracy levels above 90%, with LSTM being the best performing classifier with multiple classes and still achieving an accuracy of over 90%. This classification provides some key insights such as the most used classifiers include SVM, LR, k-NN, LDA, and NN variants. In their research studies, the highest accuracies were attained by LSTM and TCN while SVM and LR are the non-NN classifiers that have achieved the highest accuracies.

Table 2. Review of previous works about motor imagery using machine learning(Garcia-Moreno et al., 2020).

Method	Channels	Intrusive	Own Dataset	Subjects	Classes	Session Size	Validation Split	Accuracy
CNN+LSTM	EEG: 64	Yes	No	Cross-Subject: 108	Five	120 s	75–25%	98.3%
LSTM	EEG: 64	Yes	No	Intra-Subject: 109	Five	120 s	5 × 5-fold	97.8%
CNN+LSTM	Muse: 4	Low	Yes	Intra-Subject: 1	Four	30 s	90–10%	80.13%
CNN+LSTM	Muse: 4	Low	Yes	Cross-Subject: 4	Binary	20 s	90–10%	98.9%
SVM	Muse: 4	Low	Yes	Intra-Subject: 8	Binary	10 s	4-fold	95.1%
SVM	EEG: 2	Yes	No	Intra-Subject: 2	Binary	9 s	50–50%	82.14%
CNN	EEG: 28	Yes	Yes	Intra-Subject: 2	Binary	5 s	80–20%	86.41%
RLDA	EEG: 22	Yes	No	Intra-Subject: 9	Four	4 s	ICV	73.7%
CNN	EEG: 44	Yes	Yes	Intra-Subject: 20	Four	4 s		93.9%
LSTM	EEG: 6	Yes	No	Intra-Subject: 9	Binary	4 s	5 × 5-fold	79.6%
CNN	EEG: 3	Yes	No	Intra-Subject: 9	Binary	4 s	60%–40%	78.44%
CNN+SAE	EEG: 3	Yes	No	Intra-Subject: 9	Binary	4 s	10 × 10-fold	77.6%
LR	EEG: 128	Yes	Yes	Intra-Subject: 29	Three	3 s	50%–50%	90.5%
CNN	EEG: 22	Yes	No	Intra-Subject: 9 Cross-Subject: 9	Four	2 s	4-fold	~70% ~40%

The classification can be summarized as follows:

- The most used classifiers are: SVM, LR, k-NN, and Neural networks variants.
- The highest accuracy in research is reached with the LSTM and its combinations.
- The non-Neural networks that have the highest accuracies are the SVM and LR.

2.5 Automated machine learning pipeline

Automated machine learning, known as AutoML, has become a popular area of interest in both industry and academia due to the extensive involvement of human experts in every aspect of traditional machine learning, which requires significant knowledge and effort to achieve satisfactory performance. Machine learning techniques have become deeply rooted in our daily lives, the development of AutoML aims to make these techniques more accessible and reduce the dependence on experienced human experts(Yao et al., 2018). Due to the lack of a dedicated study exploring an automated ML pipeline in EEG-based BCI applications, this section reviews the prior literature on Automated ML pipelines in different fields.

(Chatterjee & Byun, 2022) studied classifying EEG signals' positive, negative, and neutral emotional states by utilizing Pycaret as AutoML to increase emotions classification efficacy. In order to train the model using the selected features, a random forest, a light gradient boosting machine, and a gradient-boosting-based stacking ensemble classifier (RLGB-SE) were used as base classifiers at level 0 of the model selection. Based on the results of each base classifier, the random forest (RF) was trained at level 1. To avoid overfitting, they used a k-fold cross-validation technique as a frequently used validation method. Their model achieved 99.55% classification accuracy.

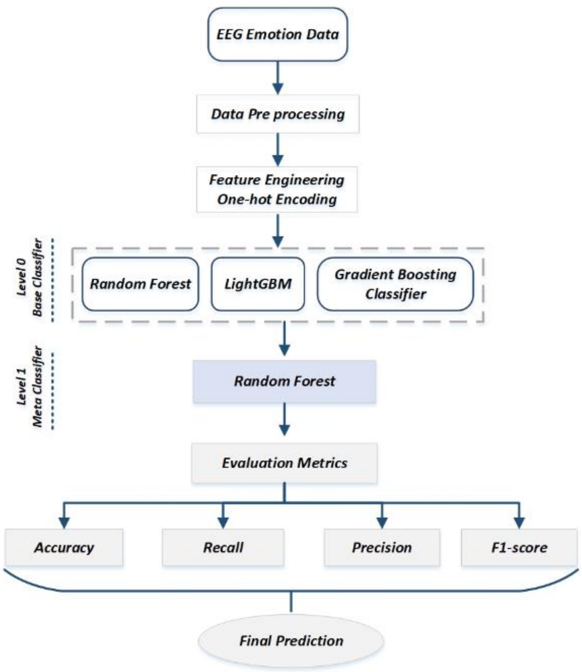

Figure 4. Base concept of proposed methodology(Chatterjee & Byun, 2022).

(Zhao et al., 2022)have presented a novel approach named AutoDESS that applies dynamic ensemble selection (DES) techniques in automated machine learning. This is particularly significant since ensemble learning is an essential component of the model ensemble aspect of the AutoML framework. Two main ensemble strategies, namely stacked generalization and static selection have been used in recent decades to exploit the high performance of classification tasks. Table 2. displays a comparison between six distinct AutoML Pipelines utilizing different ensemble approaches.

Table 3. Current AutoML framework in ensemble part(Zhao et al., 2022)

AutoML Framework	Ensemble Method
Auto-sklearn	Static Selection
TPOT	Stacked Generalization
H2O AutoML[5]	Stacked Generalization + Bagging
AutoGluon	Stacked Generalization (Multi-Layer)
FLAML	Stacked Generalization (Optional)
mljarsupervised	Static Selection + Stacked Generalization

AutoDESS constructs a diverse set of ensemble strategies and employs Bayesian optimization to improve model performance, particularly in addressing imbalanced datasets. The authors conducted extensive evaluations on various datasets, which demonstrated that AutoDESS outperforms existing approaches in terms of identifying the best or near-best strategies within the same CPU runtime.

(Occhipinti et al., 2022) present a study that aimed to investigate and optimize the performance of machine learning text classifiers commonly used for spam filtering. The authors surveyed 12 classifiers and proposed a new pipeline that optimized hyperparameters and applied natural language processing methods to improve performance on the Enron dataset. The classifiers were evaluated using metrics such as F-score (accuracy), precision, recall, and run time. The analysis identified an effective machine learning model to classify the dataset with an F-score of 94% and showed that the proposed pipeline achieved good accuracy for spam filtering on the dataset.

(Feurer et al., 2015.) studied on automated machine learning using effective Bayesian optimization method. They have introduced a robust new AutoML system named AUTO-SKLEARN based on scikit-learn that include 15 classifiers, 14 feature preprocessing methods, and 4 data preprocessing methods, and a structured hypothesis space with 110 hyperparameters. AUTO-SKLEARN surpasses existing AutoML techniques by automatically considering previous performance on similar datasets and building ensembles from the models evaluated during optimization. The authors also demonstrate the performance improvements resulting from their contributions and gain insights into the effectiveness of individual components of the AUTO-SKLEARN system.

(Renggli et al., 2021) defined their challenge in Machine learning operations (MLOps) is to determine how to estimate the error that the best possible machine learning model can achieve, without the need for expensive training. This is done by estimating the Bayes error rate (BER), which is related to the data distribution and is difficult to estimate accurately with limited data.

To predict student dropout problems in Sub-Saharan African countries, (Mnyawami et al., 2022) compared ensemble learning and KNORA-AutoML. In comparison to the conventional ensemble of optimized ML models with accuracy = 96%, precision = 70%, and AUC = 78%, KNORA-AutoML scored 97% accuracy, 71% precision, and 87% AUC. The accuracy, precision, and AUC of the KNORA-AutoML model increased by 0.6%, 0.8%, and 8.7%, respectively.

(Consuegra-Ayala et al., 2022) proposed a two-phase optimization system that employs Auto-ML tools, particularly AutoGOAL, to generate robust classifiers for classification problems. The first phase of the system utilizes a probabilistic strategy to create a diverse collection of base models by selecting the best combination of algorithms and hyperparameters. In the second phase, Auto-ML techniques are used to ensemble those models. They mentioned that, despite the time-consuming nature of Auto-ML tools compared to standard machine learning libraries, it is essential to use all available resources to achieve optimal results.

3 METHODOLOGY

In this chapter, the methodology of this study will be outlined. Initially, it describes the process of EEG recordings, including information on the participants, equipment utilized, and sensors' position. The subsequent sections will explain data preparation and feature extraction, as well as the application of an automated approach for model selection. The last section covers the validation and evaluation method.

3.1 Data Acquisition

3.1.1 Subjects

This project looks to develop a personalized classification model for an individual, as opposed to a universal model trained on multiple individuals. To ensure that the recordings were conducted accurately, the researcher used himself as a participant. Participants who are mostly colleagues and friends will be used for validation based on the playability of the game.

3.1.2 Experimental Setup

This study used an OpenBCI EEG Mark IV headset with the Cyton, WiFi boards, and ThinkPulse sensors which collects data 6 EEG channels. ThinkPulse is an EEG sensor system that is currently in use, featuring flexible polymeric dry sensors. The system is designed to conveniently supplement the standard electrode set of the Ultracortex and enhance the user's overall experience, enabling prolonged recording sessions. Data collection and processing and Auto ML pipeline are done with the OpenBCI GUI and Python with packages including brainflow, MNE, and Pycaret. Data is collected using the brainflow Python package over a direct WiFi connection to a computer at a frequency of 1000Hz. The EEG sensor locations used the 10-20 system, with locations ['Fp1', 'Fp2', 'C3', 'C4', 'P7', 'P8'] as shown in figure below:

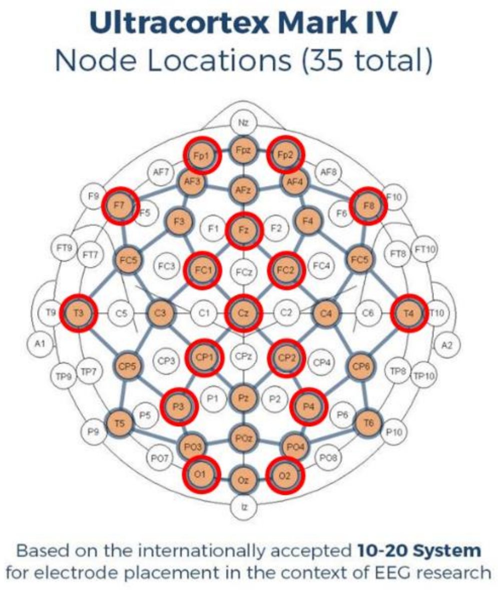

Figure 5. Electrode locations according to the internationally accepted 10-20 system for EEG recordings(Kuhn & George, 2021).

In order to obtain the data from the helmet, a video is displayed to the subject. This video contains the dropping of a ball and the movement of a bar similar to the game. This video has been designed to stimulate the subject's brain to think about the direction of the bar while the sensors collect brainwaves data and saves as CSV file.

Figure 6. Stimulus presentation for the motor imagery tasks during the recording of EEG training data.

In the video, there are two parts: a rest segment with a duration of one second without the ball on the screen, and a brick movement in which the ball is dropped from the top of the screen with a duration of four seconds, and the brick moves in the direction it is falling in.

3.2 Preprocessing

3.2.1 Row Data

EEG raw data is the unprocessed signal captured by electrodes placed on the scalp, which represents the electrical activity of the brain. The frequency analysis of the raw data can provide insight into the neural processes underlying different cognitive tasks. However, artifacts can obscure the underlying neural activity and may need to be identified and removed. Examples of artifacts include electrical interference from power grids, eye movements, muscle contractions, and electronic devices.

Effective artifact removal is crucial for accurate interpretation of the EEG signal and the success of BCI applications.

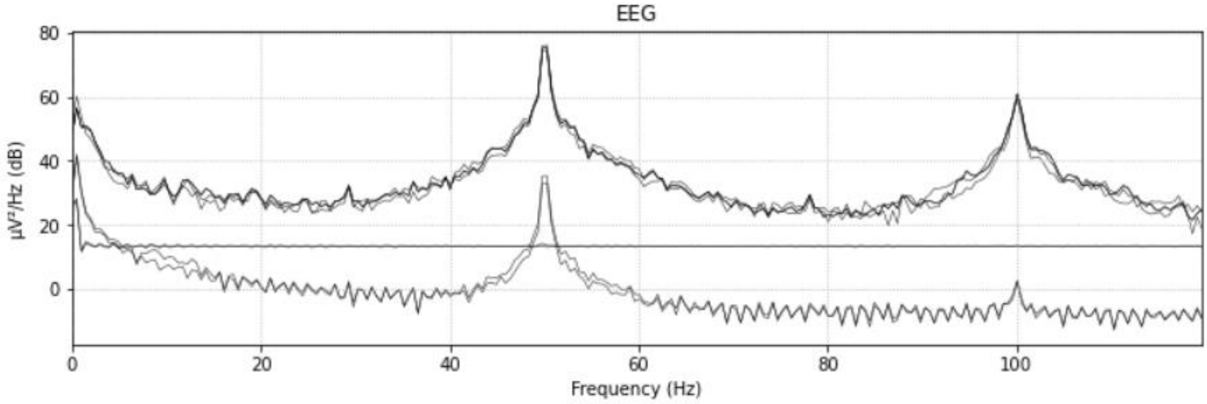

Figure 7. Frequency domain representation of the raw EEG data. Each sensor's amplitude is displayed per frequency.

3.2.2 Data preparation

After collecting the dataset, it is processed through a pipeline consisting of several steps. The first step involves using the sliding window method to generate datapoints. Next, the data is filtered using a Finite Impulse Response (FIR) filter to remove noise and improve its quality. The third step involves scaling the data using the Standard Scalar of the data point and storing the parameters for the game.

1. Remove the first 800 ms of data at the beginning of each task, as it is unlikely to contain useful information due to the participant's reaction time when a new stimulus was shown.
2. Remove the last 200 ms of data at the end of each task, as it is assumed that the participant stops performing the mental task when the ball in the stimuli has almost hit the bar and the bar no longer needs to be moved.
3. Divide the tasks into a train, validation, and test set, making sure that data from a specific task is never present in multiple sets to prevent classifiers from predicting data from a task on which the model has already been trained.
4. Ensure that the number of tasks of each class is balanced in a set to avoid bias. In total, 36 tasks were decided upon, equal to 12 tasks per action (left, middle, right).
5. Split the tasks into overlapping time windows of 512 milliseconds each with an intermediate step of 120 data points, ensuring that the time windows of one specific task always come in the same set. This ensures that the classifying models are always tested on data from tasks they have not seen before.

After completing these steps, the data is now ready to have its noise and unwanted brainwave frequencies removed, which will be described in the following section.

3.2.3 Standard preprocessing transformations

The recorded signal is typically contaminated with noise and other physiological artifacts. Therefore, it is necessary to filter the EEG signal before any analysis. A bandpass filter is one of the commonly used filters in EEG processing. A bandpass filter is a filter that allows a range of frequencies to pass through while attenuating frequencies outside of that range. By using a bandpass filter to extract signals in the 8 to 40 Hz frequency range, researchers can isolate the motor imagery signals from other sources of noise in the EEG signal (Lu et al., 2019).
By calculating overlapping filtered data windows h(n) with the nth data point from a certain window of length M, the FIR filter works as follows:

$$h(n) = \frac{\sin\left(w_c\left(n - \frac{M-1}{2}\right)\right)}{\pi\left(n - \frac{M-1}{2}\right)} \quad (1)$$

For each h(n), the Hamming window is utilized:

$$w_c(n) = 0.54 - 0.46cos\left(\frac{2n\pi}{M-1}\right) \quad (2)$$

The FIR filter now gives the desired outcome for one window. By using this with overlapping windows of size M, new output data is produced.
As for the second transformation, it is a standard scalar. Data scaling is commonly recognized to improve almost all machine learning models, as it ensures that all sensors have the same scale of data.

$$z = \frac{x - \mu}{s} \quad (3)$$

Here x is the training data, μ is the mean, and s is the standard deviation. Standard deviation can be calculated by using Equation (3). Here, N is the size of population:

$$s = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} \quad (3)$$

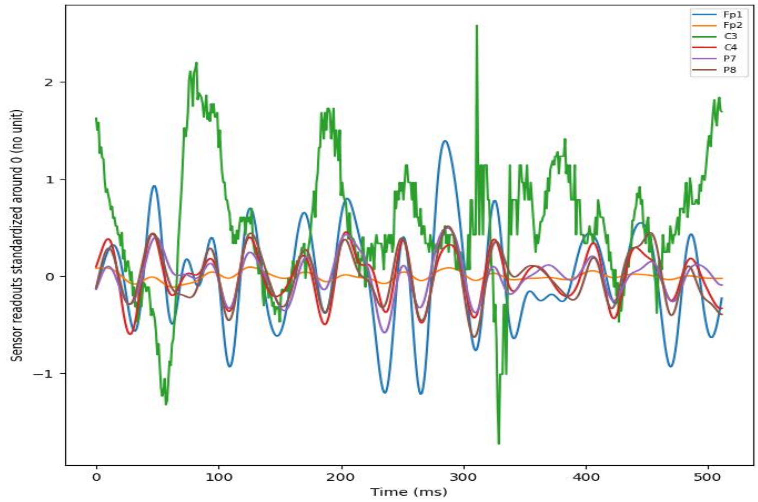

Figure 8. Sensors output after FIR filter and standard scalar process.

As a result, the mean and standard deviation are saved for use in real-time data, validation, and test data.

3.3 Feature extraction

EEG signals are complex and typically consist of various frequency components. Feature extraction is a crucial step in EEG signal processing that involves identifying and extracting informative features from raw EEG data. The extracted features can then be used for further analysis, classification, or visualization.
In the fourth step, the Common Spatial Pattern (CSP) method is applied to extract relevant features, and the CSP parameters are saved for the game.

3.3.1 Common spatial patterns (CSP)

One popular method for feature extraction in EEG signal processing is the Common Spatial Patterns (CSP) method, which has been utilized in previous work in this project(Padfield et al., 2019) and(Singh, Hussain, Lal, Guesgen, et al., 2021). CSPs aim to maximize variance within one class while minimizing it within the other, such as left-hand versus right-hand motor imagery or EEG signals associated with different mental states.
The filter matrix $W \in \mathbb{R}^{C\times C}$contains a set of CSP filters estimated for each channel, where c represents the number of channels. In the CSP algorithm, the covariance matrixes $\Sigma^{(c)}$ for each class is computed by concatenating time-windows, with the explanation simplified for a two-class scenario.
By solving the eigenvalue problem, the W scaler can be calculated(Blankertz et al., 2008):

$$\Sigma^{(+)}w = \lambda\Sigma^{(-)}w \tag{5}$$
$$\lambda = \frac{\lambda_j^{(+)}}{\lambda_j^{(-)}} \qquad \mathrm{j} = 1, \dots, \mathrm{J}$$
$$\lambda_j^{(+)} + \lambda_j^{(-)} = 1 \qquad \mathrm{j} = 1, \dots, \mathrm{J}$$

The $\lambda_i^{(c)}$ is defined as in equation (6) with column vector j:

$$\lambda_i^{(c)} = w_i^T\Sigma^{(c)}w_i \tag{6}$$

Then the transformation of the data is a simple multiplication of the scaler $w^T$and all data points X as shown in the following:

$$x_{CSP}(t) = w^T * X \tag{7}$$

The class variances are sorted in the output. The first column (the "+" Class) has the highest variance and the last column (the "-" Class) the lowest variance. The result of CSP preprocessing will be used for the discrete wavelet transformation as the second feature extraction method.

3.3.2 Discrete wavelet transformation (WDT)

Discrete Wavelet Transforms (DWTs) have emerged as a powerful tool for analyzing EEG signals in BCI applications. DWT is a mathematical technique that decomposes a signal into a series of wavelet coefficients, capturing both frequency and temporal information simultaneously.

For any j^{th}level, the DWT algorithm decomposes a signal into low and high frequency coefficients, called approximate coefficient A_j and detail coefficient D_i respectively as given by equation (8)(Kant et al., 2019):

$$A_j = \sum_i X[i]LP[2n-1] \tag{7}$$

$$A_j = \sum_i X[i]HP[2n-1] \tag{8}$$

Where $LP[n]$ represents the low pass filter and $LP[n]$ represents the high pass filter.

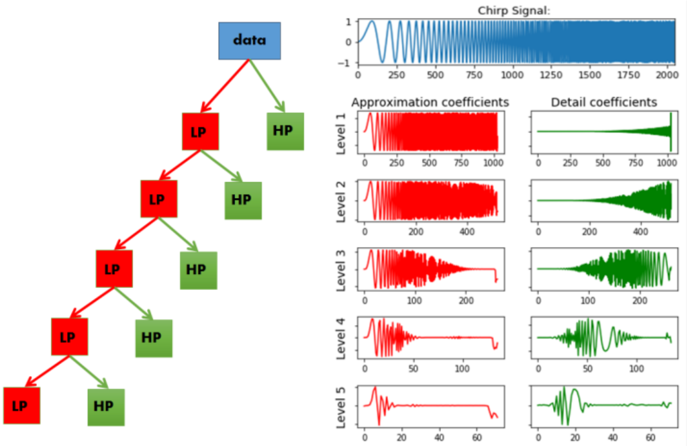

Figure 9. The concept of the different levels (from level 1 to 5) in wavelet transforms is displayed on a chirp signal. The high pass and low pass filters applied on the signal at each level is showed on the left side (Taspinar, 2018).

3.4 Auto ML

Automated Machine Learning is a technology that automates the process of building, testing, and deploying machine learning models. It is designed to minimize the need for human intervention, which can significantly reduce the time, cost, and expertise required for machine learning projects. AutoML can be used to perform various tasks, including data preprocessing, feature engineering, model selection, hyperparameter tuning, and deployment. In this project data preprocessing and feature extraction is based on previous methods on this project and AutoML is responsible for model selection and hyperparameter tuning and model development.

One of the primary benefits of AutoML is to identify the best machine learning model for problem. The traditional machine learning typically need to select a model architecture and tune its hyperparameters manually. This process can be time-consuming and require a significant amount of expertise. With AutoML, you can automate this process and let the algorithm search for the best model automatically.

AutoML algorithms typically use a combination of techniques, such as genetic algorithms, neural architecture search, and Bayesian optimization, to explore the search space of potential models and select the best one based on performance metrics such as accuracy, precision, recall, and F1 score.

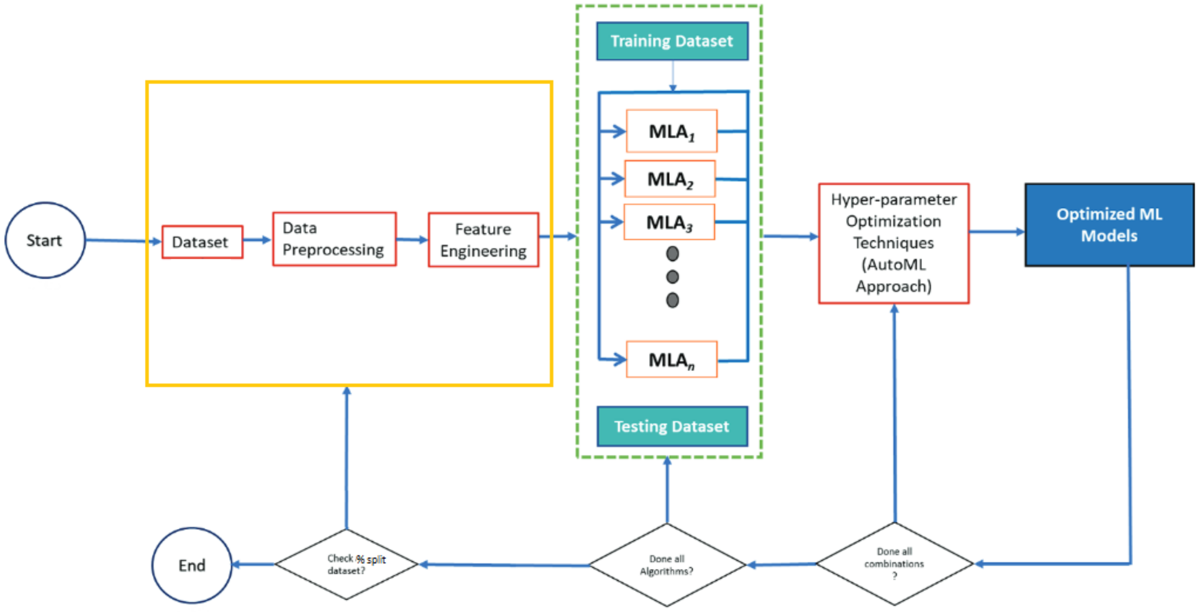

Figure 10. Automated machine learning prediction model (Mnyawami et al., 2022)

For comparing models, Pycaret trains and evaluates models on the training data and provides a summary of their performance metrics, such as accuracy, precision, recall, and F1 score. Pycaret capable of using specific models to apply for evaluation manually or compare all available model in the Pycaret model repository.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.7689	0.8047	0.5602	0.7208	0.6279	0.4641	0.4736	1.5090
ridge	Ridge Classifier	0.7670	0.0000	0.5497	0.7235	0.6221	0.4581	0.4690	0.0360
lda	Linear Discriminant Analysis	0.7670	0.8055	0.5550	0.7202	0.6243	0.4594	0.4695	0.0680
rf	Random Forest Classifier	0.7485	0.7911	0.5284	0.6811	0.5924	0.4150	0.4238	0.1640
nb	Naive Bayes	0.7427	0.7955	0.5702	0.6543	0.6043	0.4156	0.4215	0.0490
catboost	CatBoost Classifier	0.7410	0.7993	0.5278	0.6630	0.5851	0.4005	0.4078	0.2030
gbc	Gradient Boosting Classifier	0.7373	0.7918	0.5550	0.6445	0.5931	0.4013	0.4059	0.0960
ada	Ada Boost Classifier	0.7372	0.7799	0.5275	0.6585	0.5796	0.3926	0.4017	0.1100
et	Extra Trees Classifier	0.7299	0.7788	0.4965	0.6516	0.5596	0.3706	0.3802	0.1560
qda	Quadratic Discriminant Analysis	0.7282	0.7894	0.5281	0.6558	0.5736	0.3785	0.3910	0.0460
lightgbm	Light Gradient Boosting Machine	0.7133	0.7645	0.5398	0.6036	0.5650	0.3534	0.3580	0.2690
knn	K Neighbors Classifier	0.7001	0.7164	0.5020	0.5982	0.5413	0.3209	0.3271	0.0590
dt	Decision Tree Classifier	0.6928	0.6512	0.5137	0.5636	0.5328	0.3070	0.3098	0.0500
xgboost	Extreme Gradient Boosting	0.6853	0.7516	0.4912	0.5620	0.5216	0.2887	0.2922	0.0840
dummy	Dummy Classifier	0.6518	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0410
svm	SVM - Linear Kernel	0.5954	0.0000	0.3395	0.4090	0.2671	0.0720	0.0912	0.0400

Figure 11. Training and evaluation of the performance of all estimators available in the model library for a classification problem ('diabetes' opensource library) using cross-validation, of which the output is a scoring grid with average cross-validated scores (Quickstart - Docs, n.d.).

Tuning hyperparameters using a grid search or a random search over a range of hyperparameters and selects the best ones based on performance metrics.
Beside creating a best model, Pycaret is capable of creating Ensemble models which combines the predictions of multiple models to improve the overall performance of the model.

3.5 Breakout game

There are three main phases in the game: obtaining the subject's data, training the model, and playing which are introduced as follows.

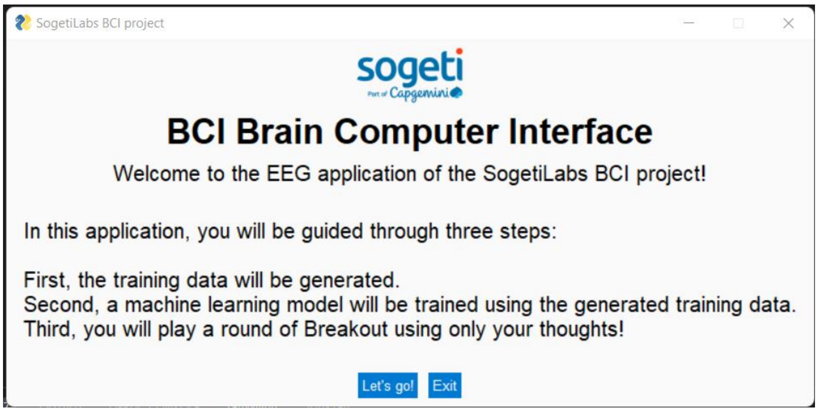

Figure 12. The game starts with an introduction about the steps

During this phase, the subject will be instructed to think of specific physical actions in a timed manner. These actions include steer left with left arm for a duration of 3 seconds, Stay straight for another 3 seconds, and then steer right with right arm for the final 3 seconds.

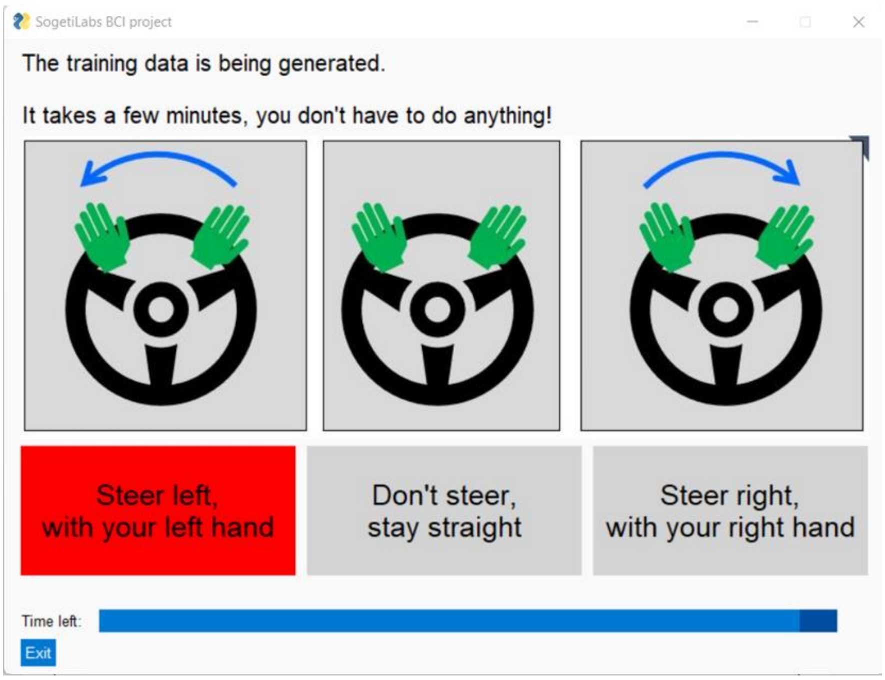

Figure 13. Data acquisition and production for playing the game at the beginning of the game.

In this step recoded data undergoes a series of steps as following. Firstly, the sliding window method is employed to generate datapoints. Subsequently, the data is filtered using a Finite Impulse Response (FIR) filter. Following this, the data points are scaled using the Standard Scalar technique, and the resulting parameters are stored for future reference in the game. Next, the Common Spatial Pattern (CSP) and Discrete Wavelet Transforms (DWTs) methods are applied to extract relevant features and extracted features parameters are saved for the game. Subsequently, the best model is selected and trained using the processed data. Finally, the trained model is saved in the repository for playing the game. This pipeline is crucial in ensuring that the data is properly preprocessed and trained on the most relevant features, resulting in an accurate and reliable model.

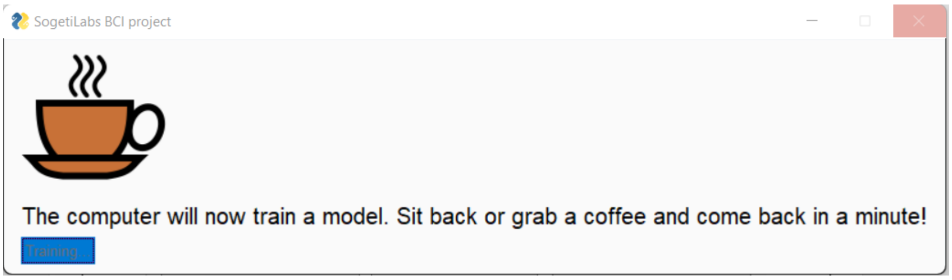

Figure 14. Graphical User Interface (GUI) of the training phase

3.5.1 Playing the game and validation

The game can begin after the model has been trained.

1. During this phase, the subject plays the game while focusing on moving the bar to intercept the ball.
2. As the ultimate goal is to create a playable version of the breakout game, much of the validation process involves playing the game and observing the results.
3. This validation is primarily qualitative, meaning that the performance of the game is assessed through observation.
4. Any necessary modifications to the game are based on these observations.
5. To conduct this validation, ten subjects will play the game multiple times.

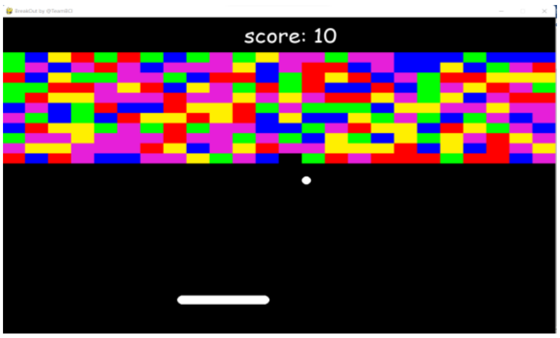

Figure 15. This figure illustrates the game Breakout. The player can only determine the movement of the bar and aims to break the colored bricks.

REFERENCES

Baig, M. Z., Aslam, N., Shum, H. P. H., & Zhang, L. (2017). Differential evolution algorithm as a tool for optimal feature subset selection in motor imagery EEG. *Expert Systems with Applications*, *90*, 184–195. https://doi.org/10.1016/J.ESWA.2017.07.033

Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., & Müller, K. R. (2008). Optimizing spatial filters for robust EEG single-trial analysis. *IEEE Signal Processing Magazine*, *25*(1), 41–56. https://doi.org/10.1109/MSP.2008.4408441

Cano, L. A., Pizá, A. G., & Farfán, F. D. (2022). Electrophysiological Biomarkers for the Assessment of Motor Efficiency in Sport. *Journal of Sports Research*, *9*(1), 10–25. https://doi.org/10.18488/90.v9i1.2897

Chatterjee, S., & Byun, Y. C. (2022). EEG-Based Emotion Classification Using Stacking Ensemble Approach. *Sensors 2022, Vol. 22, Page 8550*, *22*(21), 8550. https://doi.org/10.3390/S22218550

Consuegra-Ayala, J. P., Gutiérrez, Y., Almeida-Cruz, Y., & Palomar, M. (2022). Intelligent ensembling of auto-ML system outputs for solving classification problems. *Information Sciences*, *609*, 766–780. https://doi.org/10.1016/J.INS.2022.07.061

Feurer, M., Klein, A., Jost, K. E., Springenberg, T., Blum, M., & Hutter, F. (n.d.). *Efficient and Robust Automated Machine Learning*. http://automl.org

Garcia-Moreno, F. M., Bermudez-Edo, M., Garrido, J. L., & Rodríguez-Fórtiz, M. J. (2020). Reducing response time in motor imagery using a headband and deep learning. *Sensors (Switzerland)*, *20*(23), 1–18. https://doi.org/10.3390/s20236730

Kant, P., Hazarika, J., & Laskar, S. H. (2019). Wavelet transform based approach for EEG feature selection of motor imagery data for braincomputer interfaces. *Proceedings of the 3rd International Conference on Inventive Systems and Control, ICISC 2019*, 101–105. https://doi.org/10.1109/ICISC44355.2019.9036445

Kevric, J., & Subasi, A. (2017). Comparison of signal decomposition methods in classification of EEG signals for motor-imagery BCI system. *Biomed. Signal Process. Control.*, *31*, 398–406. https://doi.org/10.1016/J.BSPC.2016.09.007

Kuhn, S., & George, N. (2021). *PERFORMANCE OF OPENBCI EEG BINARY INTENT CLASSIFICATION WITH LARYNGEAL IMAGERY A PREPRINT*.

Lu, N., Yin, T., & Jing, X. (2019). A temporal convolution network solution for eeg motor imagery classification. *Proceedings - 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering, BIBE 2019*, 796–799. https://doi.org/10.1109/BIBE.2019.00148

Martini, M. L., Oermann, E. K., Opie, N. L., Panov, F., Oxley, T., & Yaeger, K. (2020). Sensor Modalities for Brain-Computer Interface Technology: A Comprehensive Literature Review. *Clinical Neurosurgery*, *86*(2), E108–E117. https://doi.org/10.1093/NEUROS/NYZ286

Mnyawami, Y. N., Maziku, H. H., & Mushi, J. C. (2022). Comparative Study of AutoML Approach, Conventional Ensemble Learning Method, and KNearest Oracle-AutoML Model for Predicting Student Dropouts in Sub-Saharan African Countries. *Applied Artificial Intelligence*, *36*(1). https://doi.org/10.1080/08839514.2022.2145632

Occhipinti, A., Rogers, L., & Angione, C. (2022). A pipeline and comparative study of 12 machine learning models for text classification. *Expert Systems with Applications*, *201*, 117193. https://doi.org/10.1016/J.ESWA.2022.117193

Padfield, N., Zabalza, J., Zhao, H., Masero, V., & Ren, J. (2019). EEG-based brain-computer interfaces using motor-imagery: Techniques and challenges. *Sensors (Switzerland)*, *19*(6). https://doi.org/10.3390/s19061423

Portillo-Lara, R., Tahirbegi, B., Chapman, C. A. R., Goding, J. A., & Green, R. A. (2021). Mind the gap: State-of-the-art technologies and applications for EEG-based brain-computer interfaces. In *APL*

Bioengineering (Vol. 5, Issue 3). American Institute of Physics Inc. https://doi.org/10.1063/5.0047237

Quickstart - Docs. (n.d.). Retrieved May 9, 2023, from https://pycaret.gitbook.io/docs/get-started/quickstart

Renggli, C., Rimanic, L., Gürel, N. M., Karlaš, B., Wu, W., & Zhang, C. (2021). *A Data Quality-Driven View of MLOps*. http://arxiv.org/abs/2102.07750

Singh, A., Hussain, A. A., Lal, S., & Guesgen, H. W. (2021). A comprehensive review on critical issues and possible solutions of motor imagery based electroencephalography brain-computer interface. In *Sensors* (Vol. 21, Issue 6, pp. 1–35). MDPI AG. https://doi.org/10.3390/s21062173

Singh, A., Hussain, A. A., Lal, S., Guesgen, H. W., Singh, A. ;, Hussain, A. A. ;, Lal, S. ;, & Guesgen, H. W. A. (2021). *A Comprehensive Review on Critical Issues and Possible Solutions of Motor Imagery Based Electroencephalography Brain-Computer Interface Comprehensive Review on Critical Issues and Possible Solutions of Motor Imagery Based Electroencephalography Brain-Computer Interface. Sensors*. https://doi.org/10.3390/s21062173

Vidal', J. J. (1973). *TOWARD DIRECT BRAIN-COMPUTER COMMUNICATION 9027*. www.annualreviews.org

Wang, J., & Wang, M. (2021). Review of the emotional feature extraction and classification using EEG signals. *Cognitive Robotics*, *1*, 29–40. https://doi.org/10.1016/J.COGR.2021.04.001

Yao, Q., Wang, M., Chen, Y., Dai, W., Li, Y.-F., Tu, W.-W., Yang, Q., & Yu, Y. (2018). *Taking Human out of Learning Applications: A Survey on Automated Machine Learning*. https://doi.org/10.48550/arxiv.1810.13306

Zhao, Y., Zhang, R., & Li, X. (2022). *AutoDES: AutoML Pipeline Generation of Classification with Dynamic Ensemble Strategy Selection*. http://arxiv.org/abs/2201.00207

Appendix

When evaluating the performance of machine learning models for EEG BCI (brain-computer interface) applications, commonly used performance metrics include accuracy, sensitivity, specificity, precision, recall, and F1 score. Here there are a brief of application of these methods:
Accuracy is the proportion of correctly classified instances out of the total instances. In EEG BCI applications, accuracy can be a useful metric to measure overall performance.

Sensitivity measures the proportion of true positive instances (correctly predicted positive instances) out of all positive instances. In EEG BCI applications, sensitivity is an important metric as it reflects the ability of the model to correctly identify positive instances, such as the detection of a specific brainwave pattern associated with a particular task or activity.

Specificity measures the proportion of true negative instances (correctly predicted negative instances) out of all negative instances. In EEG BCI applications, specificity can be important when there is a need to identify and filter out irrelevant or unwanted signals.

Precision measures the proportion of true positive instances out of all instances predicted as positive. In EEG BCI applications, precision can be useful when a high level of confidence is required in the model's predictions.

Recall measures the proportion of true positive instances out of all actual positive instances. In EEG BCI applications, recall can be useful when the goal is to detect as many positive instances as possible, even if there are some false positives.

F1 score is the harmonic mean of precision and recall and provides a balance between the two metrics. In EEG BCI applications, F1 score can be useful when both precision and recall are important.
The choice of performance metric depends on the specific requirements and objectives of the EEG BCI application. For example, if the goal is to maximize detection of a specific brainwave pattern, sensitivity may be the most important metric, whereas if the goal is to minimize false positives, specificity and precision may be more important.