Technium Vol. 15, pp. 45-59 (2023)
ISSN: 2668-778X
www.techniumscience.com

EEG-based Mouse Cursor Control

Naila Allahverdiyeva, Aghamirza Fataliyev

Process Automation Engineering department Baku Higher Oil School

naila.allakhverdiyeva@bhos.edu.az, agamirza.fataliyev.std@bhos.edu.az

Abstract. The last decade has seen an increase in the use of artificial intelligence (AI) and machine learning. Recent advances in the field of BC have led to renewed interest in the use of electroencephalography (EEG) for different fields. EEG is used in medical and biomedical applications such as analyzing mental workload and fatigue, diagnosing brain tumors, and rehabilitation of central nervous system disorders; EEG-based movement analysis and classification is widely used in many areas, from clinical applications to brain-machine interface and robotic applications. This article reviews applications of several BC algorithms used in EEG signal processing, introducing commonly used algorithms, typical application scenarios, key advances, and current problems. The study explored current ML applications in EEG, including brain-computer interfaces, cognitive neuroscience, diagnosis of brain disorders. First, the basic principles of ML algorithms used in EEG signal processing, including convolutional neural networks, support vector machines, K-nearest neighbor, and omnidirectional convolutional neural networks, are briefly described. Additionally, a general survey of BC applications used in EEG analysis is presented. As a result, it was determined that SVM methods were used most in the studies, and the study topics were mainly on epilepsy, BCI, and Emotion, and least on Sleep States and Perception.

Keywords. Electroencephalography, Machine learning, Signal processing, Feature extraction, EEG

I. Introduction

Electroencephalography (EEG) is the only brain imaging method with high temporal and spatial resolution of electrical fields produced by the brain. EEG records electrical signals from the brain, thus providing the ability to extract valuable information about brain activity. EEG signals allow the monitoring of electrical brain activity produced by neurons. Machine learning (ML) tools from EEG have gained importance in recent years in schizophrenia, mental disorders, and other related diseases (Mohammad,2022).

EEG, signal analysis, epilepsy detection, diagnosis of Alzheimer's disease, sleep disorder analysis, etc. It is one of the main tools for the auxiliary analysis of nervous system diseases that are widely studied in different fields. In today's world of big and complex data-driven applications, research engineers focus on studies such as bioinformatics, medical imaging, pervasive sensing, medical informatics, and public health (Al-Omari, Qahwaji., and Ipson, 2008).

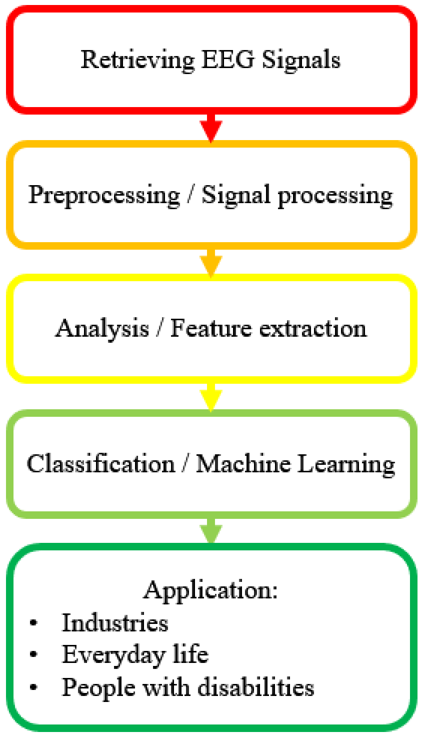
Such studies also include those related to neuroengineering research, which involves brain activity using a non-invasive measurement technique called EEG. The basic concept of EEG involves measuring electrical activity (voltage change) across the scalp. The EEG signal is one of the most complex in health data; Deep learning (DL) techniques can be used in various applications such as insomnia diagnosis, seizure detection, sleep studies, emotion recognition, and brain-computer interface (BCI) (Homri, Yacoub and Ellouze, 2012). Brain signal analysis and acquisition is also an essential field, regardless of any field, i.e. manufacturing, network, social, financial, social sciences. It can be used as evidence of good scientific research that can be used for accurate "diagnosis" and "analysis" purposes in measurable hospitals. ESA is designed to work with images. Deep Neural Networks (DSA) is a newly added subfield of BC. Inspired by artificial neural networks (Momammad, Emad, Aya and Khaled, 2010). This type of neural network is a multilayer perceptron (MCA) with a special topology and contains multiple hidden layers. Convolutional Neural Networks (ESA) are replacing time-consuming traditional feature extraction and classification algorithms. This neural network is used for object A review of topics using machine learning to process EEG signals BUFBD 5-1,2022 126 recognition and handwritten character recognition. Adopting a lazy learning method, K-Nearest Neighbor (KYK) is adopted, which determines class labels according to the most frequent value among kn neighbors. Various numbers of neighbors (kn) are taken into account to determine which value of kn gives the best performance. The random forest model adopts an ensemble classification method that predicts classes based on the results of multiple decision trees (MCA) (Borís, Laure, Clémence, Laurence, Thierry and Franck,2015).

A linear support vector machine (SVM) separates seizure data from non-seizure data based on a hyperplane that maximizes the distance between two classes. Advanced machine learning (IMM) is a single hidden layer feedforward neural network (FNA) and learning with local consistency of data has been used in recent years to improve the performance of existing machine learning models. These are nonlinear classifiers, SVM, and multilayer perceptron (MCA) based neural networks that allow drawing complex optimal classification boundaries (Al-Omari, Qahwaji., and Ipson,2010).

The main purpose of this thesis is to demonstrate the potential and power of EEG signal processing on a seemingly simple example - mouse cursor control. Several previous works and methods of implementation, as well as their pros, cons, and comparisons, will be discussed in addition to possible ways of development.

Below provided picture demonstrates the general flowchart which all methods follow. It is important to keep in mind that the background processes taking place in each stepwill vary from one approach to another.

Picture 1. Background

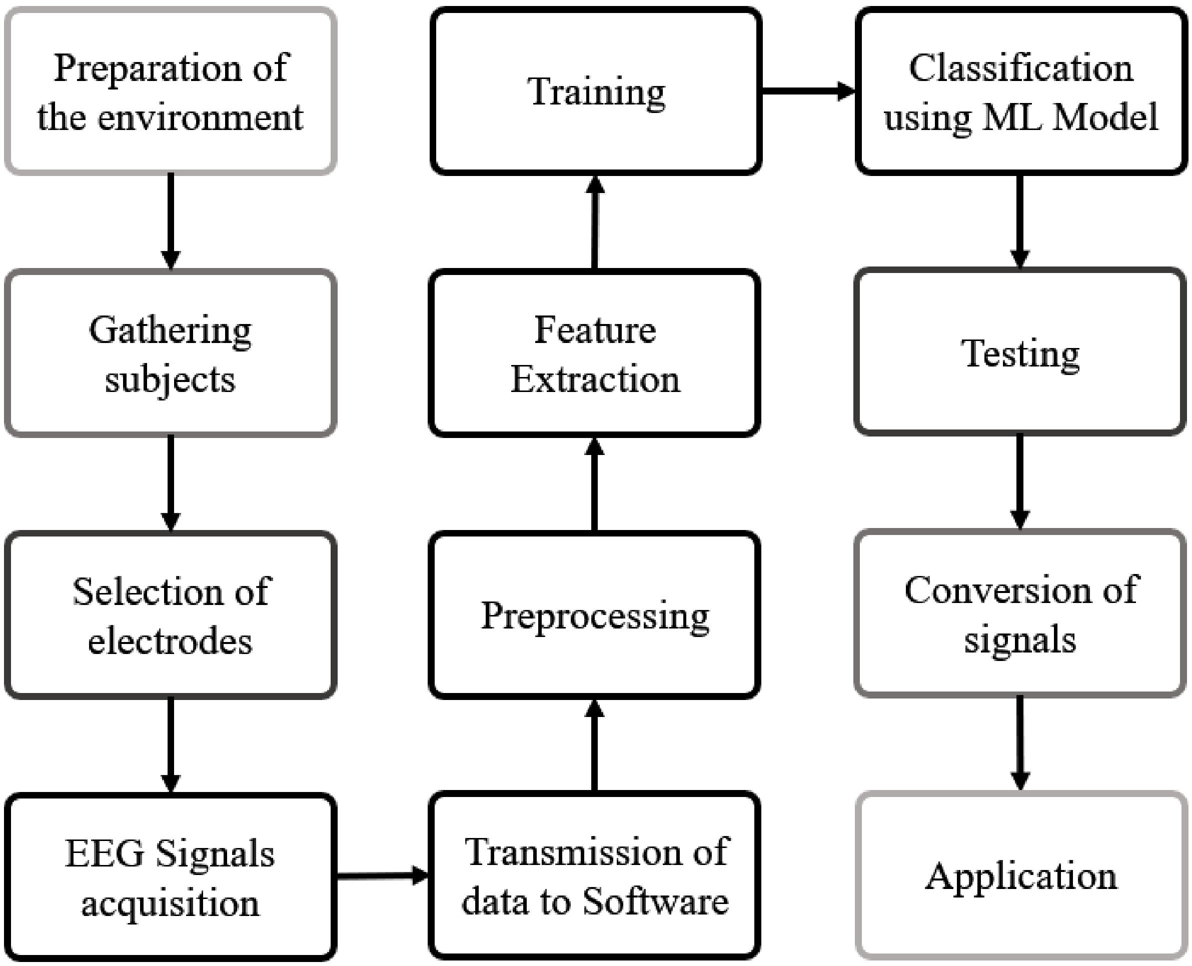


Flowchart demonstrating the general procedure.

II. Methodology

The complete system consists of several steps and stages, starting with preparing the environment for the EEG recording sessions and concluding with applying the created conversion software to control objects and devices (Benjamin, Claudia, Sebastian, 2010).
The flowchart is given below:

Figure 1. Flowchart of the system



As mentioned in the previous section, I will use machine learning algorithms in my approach and focus on the correct classification of signals, as it is the most challenging part of this project. Nevertheless, before that, I will go through all the primary processes and discuss all the proper steps to classify the signals in the best feasible way (Phinyomark, Limsakul and Phukpattaranont, 2010).

III. Results and discussion

Dataset

We will work with one of the popular motor imagery BCI dataset that is available online provided by the Berlin BCI group.

Description and details:

- Healthy volunteers were involved.
- Number and material of electrodes: 59, Ag/AgCl
- Band-pass filtered: 0.05 to 200 Hz
- Sample rate: 100
- Digitized at 1000 Hz

In the beginning, 2 sets included visual cues represented as arrows pointed left and right appearing on a computer screen. Then, the subject had to perform the motor imagery task displayed in the form of cues for 4 seconds. These tasks were interchanged with two 2 seconds long intervals: black display and a cross in front of the displayed cues.

* The interesting feature of this dataset is that some trials were artificially generated.

Loading the data

For all the main programming VS Code is used. Before loading the data, we have to import NumPy and Scipy.io libraries and functions, as we will deal with the .mat file.

Then, we extract the information (See Figure 2.) encoded to the set (we are working with the 3rd out of 4 sets), such as sampling rate, used channel names, event onsets, in other words time samples where the triggering happens, and event codes (markers). There are 2 labels in this trial: right and left, which we are going to use as 2 separate classes.

Figure 2. Information about the set

Name ▲	Type	Size	Value
channel_names	list	59	['AF3', 'AF4', 'F5', 'F3', 'F1', 'Fz', 'F2', 'F4', 'F6', 'FC5',
cl_lab	list	2	['left', 'right']
cl1	str_	4	left
cl2	str_	5	right
EEG	ndarray	(59, 190473)	[[-36 -203 -384 ... -256 -287 -289]
event_codes	ndarray	(1, 200)	[[1 1 1 -1 1 1 1 -1 1 1 1 -1 1 -1 -1 -1 1 -1 -
event_onsets	ndarray	(1, 200)	[[2095 2895 3695 4495 5295 6095 6895
file	str	62	'C:\Users\ASUS\AppData\Local\Temp\ipykernel_2
labels	ndarray	(1, 190473)	[[0 0 0 ... 0 0 0]]
m	dict	6	{'__header__': b'MATLAB 5.0 MAT-file, Platform: G
nchannels	int		59
nclasses	int		2
nevents	int		1
nsamples	int		190473
sample_rate	uint8		100

Segmentation and Power Spectral Density

We define the trials dictionary in order to store the trials of each class. The time window is 0.5 to 2.5 seconds. So, the number of samples will be equal to the number of

Technium Vol. 15, pp. 45-59 (2023)
ISSN: 2668-778X
www.techniumscience.com

time windows. We have 200 samples, so now we can loop over the classes using event codes and event onsets.

We allocate first the memory for the trial in our dictionary and extract each trial from our EEG. Our EEG set has a continuous shape that has 190473 time points, so we obtain the trials for two classes for right and left-hand movement (See Figure 3.).

Figure 3. Creating a dictionary and extracting the trials

```python
# Dictionary to store the trials in, each class gets an entry
trials = {}
win = np.arange(int(0.5*sample_rate), int(2.5*sample_rate)) # The time window (in samples) to extract for each trial, here 0.5 - 2.5 s
nsamples = len(win)  # Length of the time window

# Loop over the classes (right, foot)
for cl, code in zip(cl_lab, np.unique(event_codes)):
    cl_onsets = event_onsets[event_codes == code]   # Extract the onsets for the class
    trials[cl] = np.zeros((nchannels, nsamples, len(cl_onsets)))    # Allocate memory for the trials

    # Extract each trial
    for i, onset in enumerate(cl_onsets):
        trials[cl][:,:,i] = EEG[:, win+onset]

# Some information about the dimensionality of the data (channels x time x trials)
print('Shape of trials[cl1]:', trials[cl1].shape)
print('Shape of trials[cl2]:', trials[cl2].shape)
```

✓ 0.0s Python

```
Shape of trials[cl1]: (59, 200, 100)
Shape of trials[cl2]: (59, 200, 100)
```

Power Spectra Density or PSD

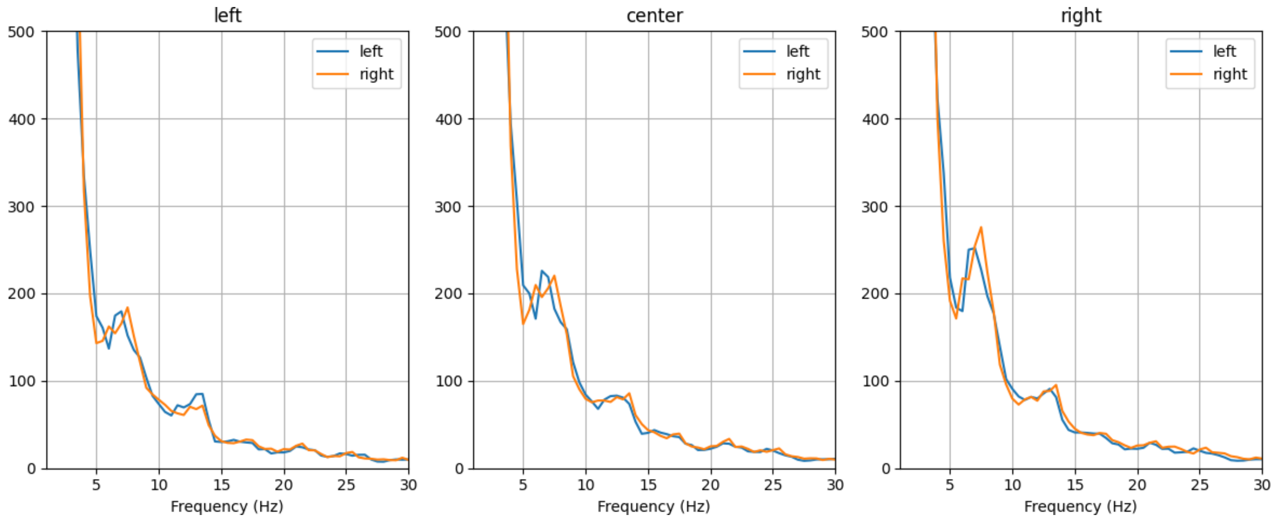
Since the feature we are looking for is a frequency feature which is related to decreasein mu activity in EEG we have to plot or analyze the power spectral density of the EEG trials.

We define **psd( )** for the evaluation of power spectral density. We simply loop through each trials using the power spectral density function **mlab.psd( )**, which uses Welch's average periodogram method. Then, we apply the PSD on all trials and then store it as the following dictionary.

By using the function **plot_psd( )** we can plot the PSD that was calculated previously.However, because plotting of all channels will clutter displays, we define the channelswe want to plot (See Figure 4.).

So, needless to say, we select three channels, C3, CZ and C4 and from the plot it is visible that there is a power spectral change for both classes.

Figure 4. Power Spectral Density Plot



So, from the mu-activity we see some suppressions and differences in the features, which will be used as input to our classifier (Magdalena, Catalina, Enrique and Manuel, 2012).

Due to the event-related desynchronization, at C4 the mu of left-hand is < than the muof right-hand, while at C3 it's the other way around. At CZ, the mu values are almost equal.

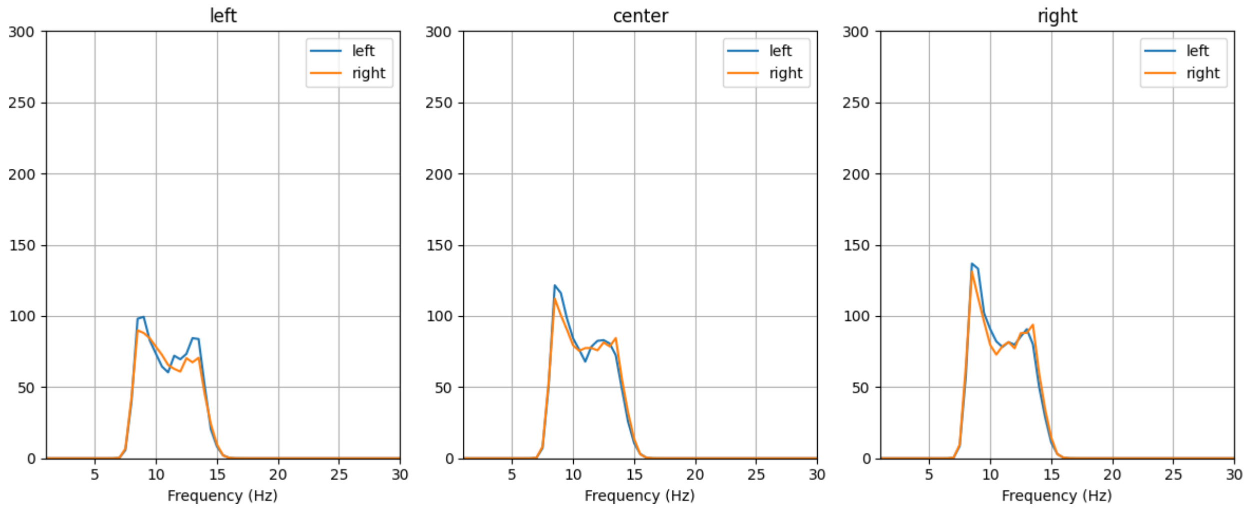
BCI control

Our eventual goal is to make a classifier to use on these data for distinguishing between sides and BCI control. To achieve that, ML algorithm will be used after determining and finding the expected trend for mu activity and testing on the sets.

So, firstly, we define our bandpass filter as a function **bandpass( )**, which allows us toapply bandpass filter to our signal between the frequency band of interest (defines low-pass and high-pass parameters). We pass the trials along with sampling frequencyrate and the IIR (infinite impulse response) filter is applied. Then we apply the bandpass filter on class 1 and class 2 data.

So, after we defined the cutoff frequency of our band pass filter as 8 hertz and 15 hertz that's why we don't see any other spectral features outside the frequency band of interest, and we can see suppression and increase of the mu band as well as more clear features (See Figure 5.).

Figure 5. Plot of bandpass filtered PSD



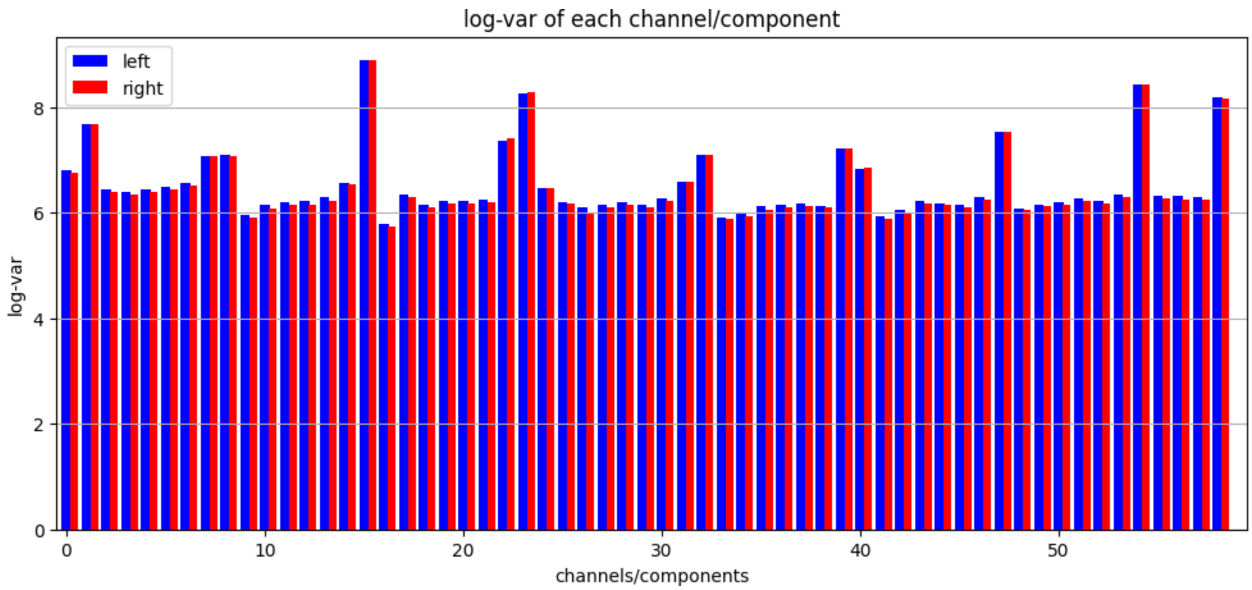
Next, we use the logvar( ) function to compute the logarithm of the variance. We are using NumPy to calculate the variance of our input data and trials along the channel axis and performing taking the logarithm.

In the following steps we extract the logarithmic features resulting in a single variablefor each trial. Each one of the dimensionality of 100 corresponds to one trial, so **logvar( )** is just one number we are using as input feature.

We can visualize **logvar( )** of each channel using the **plot_logvar( )** function of each trial across all channels. By that we obtain the bar chart that shows the logarithm of the variance for left and right, where we can see some differences in the values, most of which are very small, but at least we observed that there is some differencebetween two classes (See Figure 6.).

Technium Vol. 15, pp. 45-59 (2023)
ISSN: 2668-778X
www.techniumscience.com

Figure 6. Logarithm of Variance of each channel



Now, we need to maximize the difference between both classes. It will be easier for us, if we avoid using all the channels, but use mixtures of them that will maximize the variance difference of both classes. For this we will use a very celebrated algorithm called Common Spatial Patterns (CSP).

(Junichi U. 2021). The mixtures will be called spatial filters.

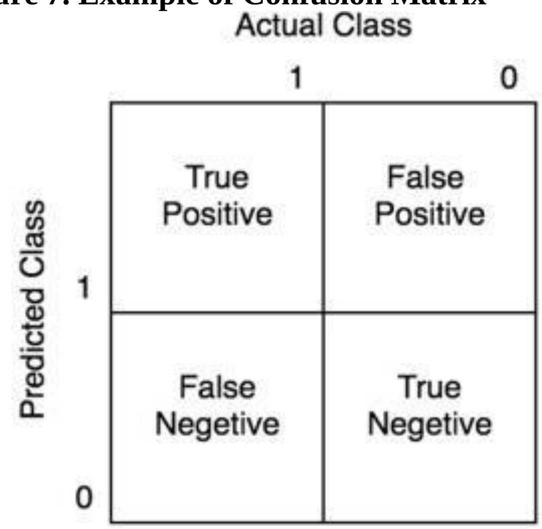
Linear classification

The differences between training and testing are significant – compared to training and testing has more mistakes, as more points are misclassified.

We can calculate and show accuracy by creating an array, in other words, a confusion matrix. The numbers at the diagonal (true) show us the numbers of correctly classifiedtrials, while the numbers of any incorrectly classified trials will be in the corners (false) (See Figure 7.).

Technium Vol. 15, pp. 45-59 (2023)
ISSN: 2668-778X
www.techniumscience.com

Figure 7. Example of Confusion Matrix



So, as a result, we see that for 5 trials with left-hand and 4 trials with the right-hand movement the classification was done inaccurately. So, total accuracy is 91 (See Figure 8.).

Figure 8. Confusion matrix - Result

```
Confusion matrix:
[[45  4]
 [ 5 46]]

Accuracy: 0.910
```

This result was achieved while taking the ratio of 50/50. We can check also for other sets and will get the following results (See Table 1).

Table 1. Results of using 50/50 Training/Testing Ratio

Sides \ Sets	*№ 1*	*№ 2*	*№ 3*	*№ 4*
Left-hand: True	43	45	45	39

Technium Vol. 15, pp. 45-59 (2023)
ISSN: 2668-778X
www.techniumscience.com

Left-hand: False	7	5	5	11
Right-hand: True	42	46	46	50
Right-hand: False	8	4	4	0
Accuracy	***85 %***	***91 %***	***91 %***	***89 %***

However, in most cases training/testing ratio is taken as > 1 (50 / 50 = 1), usually 80/20. Below we will check the accuracies for every set with 60/40, 70/30, 80/20 and 90/10 training/testing ratios (See Tables 2 to 5).

Table 2. Results of using 60/40 Training/Testing Ratio

Sides \ Sets	*№ 1*	*№ 2*	*№ 3*	*№ 4*
Left-hand: True	31	34	36	36
Left-hand: False	9	6	4	4
Right-hand: True	38	34	38	35
Right-hand: False	2	6	2	5
Accuracy	***86.3 %***	***85 %***	***92.5 %***	***88.7 %***

Table 3. Results of using 70/30 Training/Testing Ratio

Sides \ Sets	*№ 1*	*№ 2*	*№ 3*	*№ 4*
Left-hand: True	27	25	30	28
Left-hand: False	3	5	0	2
Right-hand: True	27	26	29	28
Right-hand: False	3	4	1	2

Technium Vol. 15, pp. 45-59 (2023)
ISSN: 2668-778X
www.techniumscience.com

Accuracy	***90 %***	***85 %***	***98.3 %***	***93.3 %***

Table 4. Results of using 80/20 Training/Testing Ratio

Sides \ Sets	*№ 1*	*№ 2*	*№ 3*	*№ 4*
Left-hand: True	16	19	20	18
Left-hand: False	4	1	0	2
Right-hand: True	20	15	19	19
Right-hand: False	0	5	1	1
Accuracy	***90 %***	***85 %***	***97.5 %***	***92.5 %***

Table 5. Results of using 90/10 Training/Testing Ratio

Sides \ Sets	*№ 1*	*№ 2*	*№ 3*	*№ 4*
Left-hand: True	10	10	10	10
Left-hand: False	0	0	0	0
Right-hand: True	8	7	10	10
Right-hand: False	2	3	0	0
Accuracy	***90 %***	***85 %***	***100 %***	***100 %***

It can be seen from the tables that as the ratio increased, the accuracy also increased inmost cases. We even have 100% accuracy as result for Sets 3 and 4. However, this doesn't necessarily imply that the system is totally accurate. More tests and adjustments should take place in order to ensure the proper operability of the system.

For more demonstration, scatter plots for training and testing data of Set 3 for all the ratios are given below.

Figure 9. Scatter Plots of Set 3; Ratio = 60 / 40

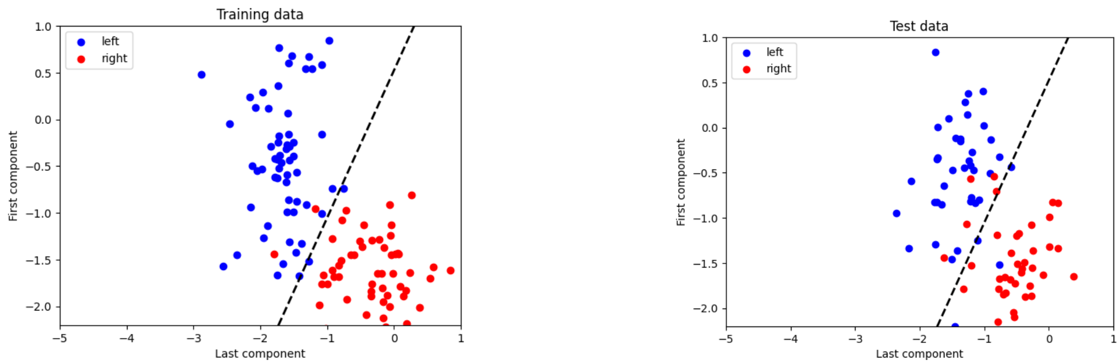


Figure 10. Scatter Plots of Set 3; Ratio = 70 / 30

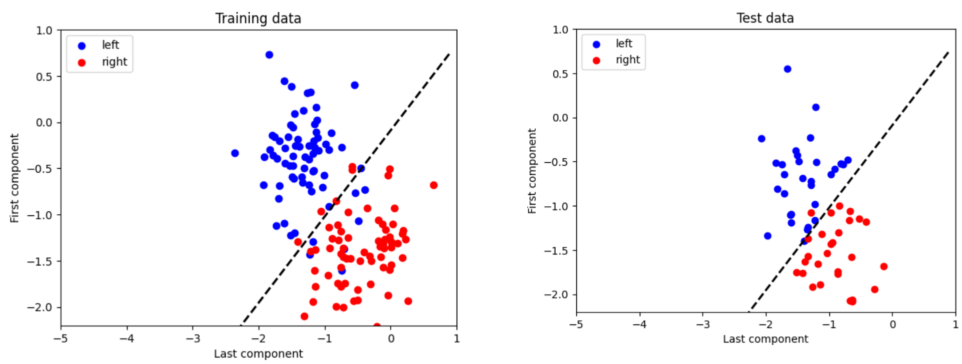


Figure 11. Scatter Plots of Set 3; Ratio = 80 / 20

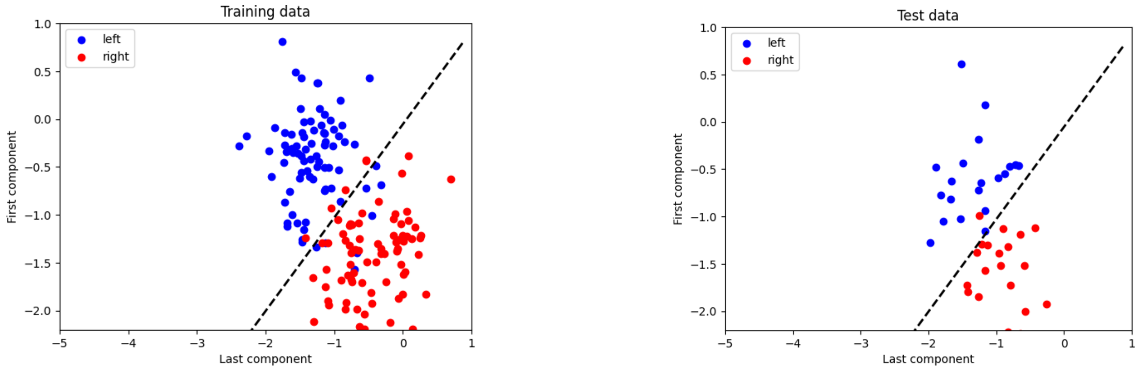
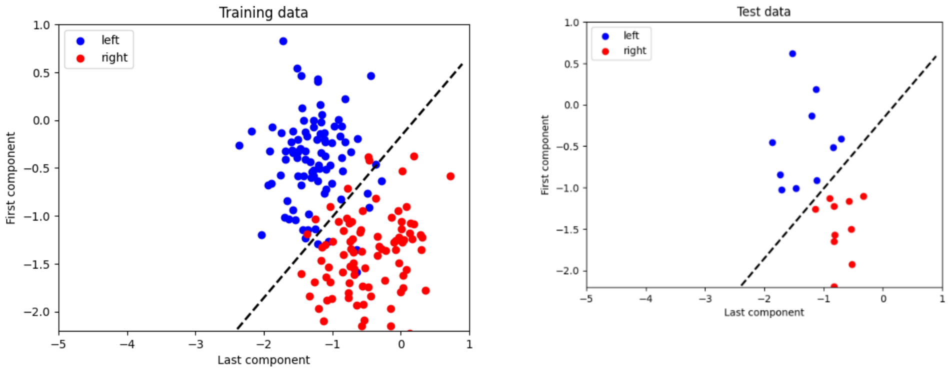


Figure 12. Scatter Plots of Set 3; Ratio = 90 / 10



IV. Recommendations

1. Expanding the dataset with properly recorded data and precisely predefined time stamps of movement/imagination tasks.
2. Increasing the amount of data used for training and testing of the algorithms.
3. Creating an application or software for conversion of brain signals, after application of machine learning, for control of mouse cursor or other devices.

V. Conclusion

To sum up, after analyzing different attempts and approaches by the corresponding researchers, I decided to use the movement imagination technique, as our first main task is to find an exact way to generate the signals in the brain, which should also be comfortable and straightforward for the people with disabilities, together with machine learning, for the extraction of necessary features and classification of them to classes after some processing.

As I experienced, creating a BCI based on Sensorimotor rhythms can be considered one of the best approaches due to its universality and independence in terms of external stimuli.

Technium Vol. 15, pp. 45-59 (2023)
ISSN: 2668-778X
www.techniumscience.com

I implemented the CSP (Common Spatial Pattern) algorithm, which requires a calculation of the covariance matrices and logarithm of variance, and LDA (Linear Discriminant Analysis), a commonly used classifier for multi-imagery tasks.

As was witnessed in the previous part, the minimal accuracy obtained from the applied strategy was 85%, which implies that the system is relatively reliable. At the same time, the maximal accuracy was 100%.

References

Al-Omari M., Qahwaji R.C.T., and Ipson S. (2010). "Machine learning-basedinvestigation of the associations between CMEs and filaments," 2010, p.90.

Al-Omari M., Qahwaji R.C.T., and Ipson S. (2008)."Automated Prediction of CMEsUsing Machine Learning of CME – Flare Associations," p.167.

Benjamin B., Claudia S., Sebastian H. (2010). "Neurophysiological predictor of SMR-based BCIperformance," *NeuroImage,* p.56.

Benjamin B., Guido D., Matthias K., Klaus-Robert M. and Gabriel C. (2021), "The non-invasive Berlin BrainComputer Interface: Fast acquisition of effectiveperformance in untrained subjects," *NeuroImage, p.102.*

Borís B., Laure S., Clémence R., Laurence C., Thierry H. and Franck V.(2015). "Spatialand temporal resolutions of EEG: Is it really black and white? A scalp current density view," *International Journal of Psychophysiology*.

Homri, S. Yacoub and Ellouze N. (2012). "Optimal segments selection for EEGclassification," in *6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications*.

Junichi U.(2021). "Use of common average reference and large-Laplacian spatial-filters enhances EEG signal-to-noise ratios in intrinsic sensorimotor activity," *Journalof Neuroscience Methods*.

Magdalena S., Catalina L., Enrique E. and Manuel R. (2012). "Mu-rhythm, visualprocessing and motor control," *Clinical Neurophysiology*.

Mohammad H. A., Ayman A., Aiman T., Ali M. B. and Adnan M. (2014). "Interface, EEG Mouse: A Machine Learning-Based Brain Computer," in *International Journal of Advanced Computer Science and Applications*.

Mohammad H. M. (2022). "A novel metaheuristic optimization method for robust spatial filter designation and classification of speech imagery tasks in EEG Brain-Computer Interface," *Artificial Intelligence-Based Brain-Computer Interface*.

Momammad H. A., Emad A. A., Aya S. and Khaled A. (2010). "Wavelet-Based Feature Extraction for the Analysis of EEG Signals Associated with Imagined Fists and Feet Movements," 2014.

Phinyomark A., Limsakul C. and Phukpattaranont P.(2010). "Optimal Wavelet Functions in Wavelet Denoising for Multifunction Myoelectric Control," in *ECTITransactions on Electrical Eng., Electronics, and Communications*.