# ML Project 2 - Report

Sarah Mikami, Samuel Waridel and Florian Kolly
*CS433, EPFL 2024 ML4Science project — TNE Laboratory*

***Abstract*—Machine Learning (ML) techniques are increasingly utilized in neuroscientific research. In particular, many studies investigated the use of ML tools on electroencephalography (EEG) recordings to classify or predict various physical and mental features, often with great success. In this work, we apply similar tools in different classification tasks: classification of action (executing a task vs. observing a task) and classification of movements (precision movement vs. force movement). Our results demonstrate that ML models, including Deep Learning models, can effectively classify actions based on EEG features. However, we also demonstrate that, for tasks where the subject is passively observing different movements, the models fail to capture the small differences in brain activity and therefore achieve low performances.**

## I. Introduction

Machine Learning (ML) is has become a central tool in neuroscience, with applications in neural engineering (e.g. Brain-Computer Interfaces, BCIs), diagnostics and fundamental research. For instance, Jalilifard et al. [1] demonstrated the efficacy of Support Vector Machine (SVM) classifiers in basic emotions recognition, while Bose et al. [2] used Random Forest classifiers to automate seizure detection in epileptic patients with high accuracy. Indeed, all major ML methods have been applied to neural signals classification[3].

In this work, we examine the applications of such ML techniques in three classification problems using electroencephalography (EEG) signals: i) distinguishing between the execution and observation of a movement; ii) differentiating between grasping a small versus a large object; iii) differentiating between observing the grasp of a small versus a large object.

Section II outlines the data acquisition process and its relevance to our classification tasks. In Section III, we describe the preprocessing of the sEEG signals and the extraction of relevant features. Sections IV and V present the models applied to our classification problems and their performance. In Section VI, we explore the idea of using only Deep Learning (DL) to directly extract features from the EEG signal. Finally, we discuss our results in Section VII and conclude in Section VIII.

## II. Data acquisition and problem specification

The dataset comprises signals recorded from four participants implanted with sEEG electrodes in distinct brain regions as part of their epilepsy treatment. During multiple experimental sessions, participants were instructed to either observe or execute specific motor tasks. These tasks involved two types of movements: a palmar grasp (force-based movement) and a pinch grasp (precision-based movement). Each session included multiple trials structured as follows: in an execution trial, the trial begins with a start cue. One second later, a light cue signals the type of object (small or large) to be targeted. This light remains illuminated for 0.5 seconds before turning off. After an additional 1.5 seconds, a ”go” signal prompts the participant to lift their hand from the resting position, grasp the indicated object, return it to its original location, and place their hand back in the resting position. Observation trials followed an identical structure, except that participants passively observed the experimenter performing the actions. A light cue indicated the participant’s role (execution or observation) in each trial.

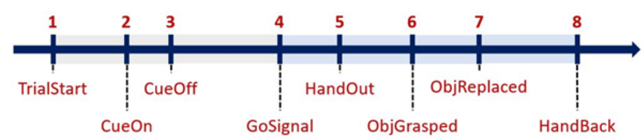


Figure 1. Timing of the trials

## III. Data analysis

In this section, we describe the preprocessing and analysis of the data. The dataset is a dictionary containing four subjects. Each subject participated to a number of sessions, ranging from only one session (subjects s6 and s7) up to three (subject s11). During each sessions, the subjects completed 256 tasks, equally distributed as observation or execution tasks. The channels are not identical across subjects.

### A. Channel responsiveness

We first filtered the channels of interest: for a given channel and task, we define i) the *baseline signal* as the average signal across all task-specific trials in the one-second window between the start of the trial (1) and the cue light turning on (2), and ii) the *effect signal* as the average signal across all task-specific trials in the one-second window surrounding the moment the object is grasped (6), whether by the subject or the experimenter. Using Welch’s method[4], we computed the power spectral density (PSD) for both the *baseline signal* and the *effect signal*. We then performed a t-test to compare the PSDs and define a *responsive channel* as a channel with a statistically significative t-test ($\alpha < 0.05$). This analysis was conducted separately for channels responsive during observation trials, execution trials, and both. As channels vary across subjects, the preprocessing was conducted individually for each subject.

### *B. Data preprocessing*

The signals (one per channel) of each trial is preprocessed in three steps.

1) **Subsampling**: the signal is subsampled from 2048 Hz to 500 Hz to reduce the computational needs. Since the EEG data contains information up to approximatively 150 Hz, this subsampling respects the Nyquist criterion.
2) **Z-score correction**: Each EEG signal is normalized with Z-score normalization, calculated using the mean and standard deviation of the *baseline signal*.
3) **Resampling**: finally, all trials are resampled to ensure they contain the same number of timepoints.

### *C. Feature extraction*

TODO

- Amplitude, shape of response $\rightarrow$ difference?
- Response pattern at the same timing?
- Similar frequency characteristics?
- Correlation? High correlation suggests congruence
- Are the results obtained similar for both movements?

## IV. Action recognition

In this first classification task, we train and optimize different family of ML models to predict whether the participant was executing the movement or observing it. The analysis is restricted to channels responsive to both movements. Given the distinct nature of the tasks being compared, we anticipate strong performance across all models.

### *A. Relevant channels*

Which regions have the most responsive channels?

- Across participant: same region? same nb of responsive channels?
- Do we find the expected regions? Link to mvt execution / observation?

For channels where we have response for both ex & obs, are the responses congruent (similar pattern) or incongruent (different)?

### *B. Baseline data*

In order to verify the claim that channel responsiveness is a critical measure for selecting channels on which to compute the features, we also generated a list of channels that are not relevant to this measure, and computed features from them. We will run each model on both sets of data, and a comparison will be done in the results.

### *C. Models*

To create and train the logistic regression, SVM and random forest models, we adopted the well-known and performant Scikit-Learn library [5]. The Deep Learning models are developed using the PyTorch library [6].

*Logistic regression:* Logistic regression (LR) is a computationally simple ML technique, and has previously used in EEG classification[7], [8].

*SVM:* We train two Singular Vector Machines (SVM) models, with and without PCA. SVM can handle high dimensional data, even for small datasets. Although more computationally than other methods, it is widely used in EEG classification with good results [9].

*Random forest:* Random Forest (RF) classifiers combine the output of multiple decision trees to reach a single result. Decision trees are algorithms classify data by creating branches depending on the value of a feature: at the end of all the branches, the leaves provide a classification result. As decision trees are prone to errors and overfitting, it is recommended to combine multiple decision trees together to improve the accuracy. A RF classifier do exactly that, by adding additional constraints to make sure the trees are different.

*MLP:* In addition to traditional ML models, we also train Multilayer Perceptron (MLP). We chose to include MLP as it is widely used in EEG classification[1]. The main issue with MLP is that they are universal approximators and are thus prone to overfitting, especially for datasets as small as ours.

### *D. Results*

We tested all models on the same test set. As expected, all models obtained very high accuracies. When using non-relevant channels, we see a large drop in performance (although it is higher than chance). This confirmed our hypothesis that we can reduce significantly the number of features by first selecting relevant channels.

[1] 3’610 results for ’EEG MLP classification’ on Google Scholar at the time of writing

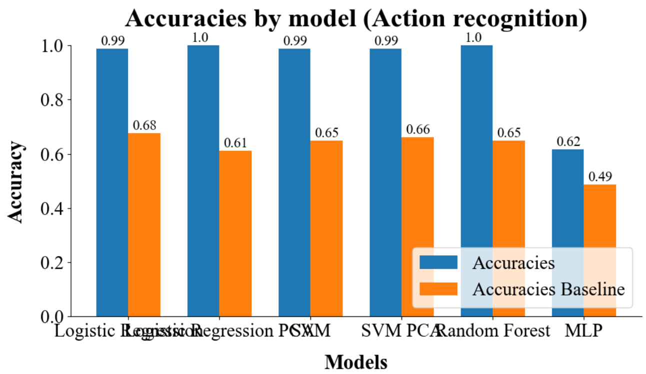


Figure 2. Accuracies by model for the action recognition task: all models perform very well when trained on features extracted from relevant channels

### E. Discussion

Which freq. bands are important? Is it the same across subjects?

## V. Movement recognition

For the second and third type of classification tasks, we analyze whether ML tools are able to classify the type of movement done (pinch grasp or force grasp) during execution and observation. We expect both tasks to be hard to learn. Observation, in particular, is a complex task, as the participant is simply seeing two different movements that have the same goal of lifting an object. In this part, we reduce the restrictions to select the *relevant channels*: it suffices for a channel to be relevant in execution or in observation respectfully to be included.

### A. Responsive channels

Which regions have the most responsive channels, are they the same for execution & observation?

- Across participant: same region? same nb of responsive channels?
- Are the channels responsive for obs also responsive for ex? Vice-versa?
- Do we find the expected regions? Link to mvt execution / observation?

For channels where we have response for both ex & obs, are the responses congruent (similar pattern) or incongruent (different)?

### B. Models

We use the same families of model as in the previous part, and this for both classifying in execution tasks and in observation tasks.

### C. Results

As expected, our results are significantly lower in those tasks than in the first part. We also observe a drop in performance between execution and observation.

*Execution trials:* The best model when classifying whether the subject has done a precision or force movement is ... TODO

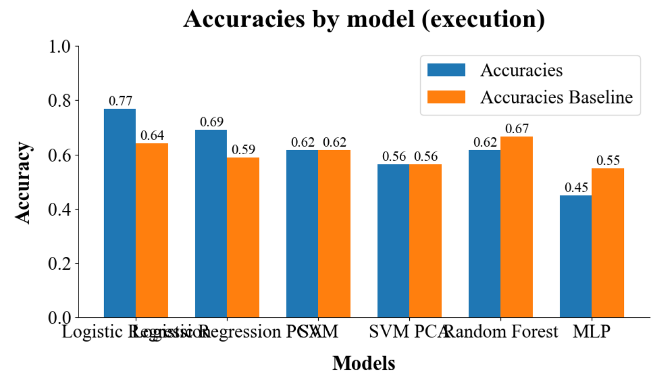


Figure 3. Accuracies by model for the movement recognition task, executed by the subjects (execution)

*Observation trials:* The best model when classifying whether the subject has seen a precision or force movement is ... TODO

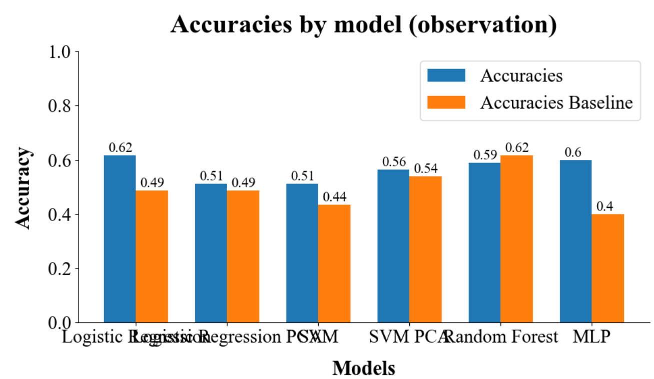


Figure 4. Accuracies by model for the movement recognition task, executed by the subjects (observation)

### D. Discussion

Which freq. bands are important? Is it the same across subjects?

## VI. Pure deep learning approach

Instead of extracting features on the relevant channels before training deep networks, one can ask whether the networks should learn to do that themselves. It's this question that we explore in this last section. We train two types of Convolution Neural Networks (CNN) directly on the trial signals. These networks contain several convolutional layers followed by linear layers. The hope is that the convolutional layers manage to extract relevant features from the signal, giving it as input to the subsequent MLP.

### A. 1D CNN

One-dimensional CNNs are composed of kernels that are convolved with the signal directly. As they operate on data in one dimension, the input will be a tensor of size $(N, C, L)$, where $N$ is the batch size, $C$ the number of (relevant) channels and $L$ the fixed length of each trial. Figure 5 shows how a kernel works in such network. This is close to how we computed how features: supposing the same kernel size as the length of the moving averages we use to create features, we would get the same features if the network learned moving averages as well. Given this and the known power of CNNs, we expect this model to have the best performance. The caveat is overfitting, as we have small datasets.

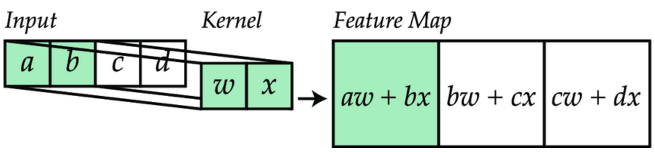


Figure 5. Example of a 1D kernel (taken from [10])

### B. 2D CNN

Two-dimensional CNNs are composed of 2D kernels that will operate on multiple channels at the same time. This means that we give as input a tensor of dimensions $(N, C, H, W)$, with $H$ the number of channels and $W$ the length of each signal. The use of $H$ and $W$ as dimensions is not an arbitrary choice. Indeed, we can think of this superposition of channels as a picture, where each signal is a position on the $y$-axis and every timepoint a position on the $x$-axis. Here, $C = 1$ as this virtual picture is simply represented by nuances of grey.

### C. Results

Both type of CNN works well.

## VII. Discussion

## VIII. Conclusion

In this project, we used various classification tools to

## IX. Ethical risks

## X. Appendix

### A. Experimental setup

The participants were asked to keep their hands on the resting positions. The led lights are used both to indicate who will realise the action and to specify which objects has to be grasped. A.) execution trial: the participant must pick the small green ball using a precise pinch grasp. B.) observation trial: the experimenter must pick the large red ball using a force grasp. C.) control action, not used in this project.

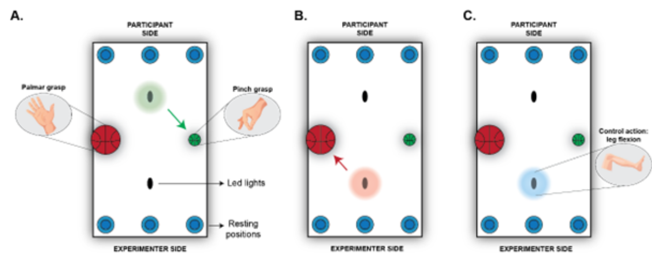


Figure 6. Experimental setup


## Acknowledgements

We thank Leonardo Pollina for taking us with him on this project, and for his quick and helpful answers to all our questions.


## References


[1] A. Jalilifard, E. B. Pizzolato, and M. K. Islam, "Emotion classification using single-channel scalp-eeg recording," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016, pp. 845–849.

[2] S. Bose, V. Rama, N. Warangal, and C. Rama Rao, "Eeg signal analysis for seizure detection using discrete wavelet transform and random forest," in *2017 International Conference on Computer and Applications (ICCA)*, 2017, pp. 369–378.

[3] M.-P. Hosseini, A. Hosseini, and K. Ahi, "A review on machine learning for eeg signal processing in bioengineering," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 204–218, 2021.

[4] O. M. Solomon, Jr, "Psd computations using welch's method. [power spectral density (psd)]," 12 1991. [Online]. Available: https://www.osti.gov/biblio/5688766

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[6] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[7] A. Subasi and E. Erçelebi, "Classification of eeg signals using neural network and logistic regression," *Computer Methods and Programs in Biomedicine*, vol. 78, no. 2, pp. 87–99, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169260705000246

[8] R. Tomioka, K. Aihara, and K.-R. Müller, "Logistic regression for single trial eeg classification," in *Advances in Neural Information Processing Systems*, B. Schölkopf, J. Platt, and T. Hoffman, Eds., vol. 19. MIT Press, 2006. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2006/file/35937e34256cf4e5b2f7da08871d2a0b-Paper.pdf

[9] M. N. A. H. Sha'abani, N. Fuad, N. Jamal, and M. F. Ismail, "knn and svm classification for eeg: A review," in *InECCE2019*, A. N. Kasruddin Nasir, M. A. Ahmad, M. S. Najib, Y. Abdul Wahab, N. A. Othman, N. M. Abd Ghani, A. Irawan, S. Khatun, R. M. T. Raja Ismail, M. M. Saari, M. R. Daud, and A. A. Mohd Faudzi, Eds. Singapore: Springer Singapore, 2020, pp. 555–565.

[10] A. Baldominos, Y. Sáez, and P. Isasi, "Evolutionary design of convolutional neural networks for human activity recognition in sensor-rich environments," *Sensors*, vol. 18, 04 2018.