TODO

Daniël Jochems
David Huizinga
Niek Grimbergen
Daan van Dam

***Abstract*—**

I. Introduction

II. Methods

A. Data Engineering

1) Normalization: To prepare the data for the model, we implemented a normalization procedure using the Normalizer class from sklearn. The normalize function applies L2 normalization to both the training and test datasets. With L2 normalization, each vector is scaled so that its norm equals 1, while preserving the direction of the original data. This helps reduce differences in scal or magnitude across the different sensor readings and labels, this will improve the models' learning process. The normalize function works by first fitting separate normalizers on the training data for X and y. These fitted normalizers are then used to transform both the training and test sets so that the test data is placed on the same scale as the training data.

2) Downsampling:

B. Model Description

C. Hyperparameter Tuning

III. Results

IV. Discussion

An attempt was made to do some specific MEG preprocessing like namely, a fixed bandpass filter, prepare channel z-scoring, and baseline correction. The inspiration for these came from the MNE tools package and the corresponding paper by Gramfort et al [1]. This applied preprocessing before model training severaly impacted results. Together they destroyed the signal for effective classification, rendering the model useless at classifying. per channel z-scoring alone was effective.

AI Statement

References

[1] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen *et al.*, "Meg and eeg data analysis with mne-python," *Frontiers in Neuroinformatics*, vol. 7, p. 267, 2013.

Supplementary Materials

[Placeholder for supplementary figures or results]