Pseudo-online framework for BCI evaluation : a MOABB perspective

Igor Carrara, Théodore Papadopoulo

PSEUDO-ONLINE FRAMEWORK FOR BCI EVALUATION: A MOABB PERSPECTIVE

Igor Carrara[1,2] Théodore Papadopoulo[1,2]
[1] Université Côte d'Azur (UCA)
[2] Centre Inria d'Université Côte d'Azur, Cronos Team
igor.carrara@inria.fr and theodore.papadopoulo@inria.fr

Abstract

Objective: BCI (Brain-Computer Interface) technology operates in three modes: *online*, *offline*, and *pseudo-online*. In the *online* mode, real-time EEG data is constantly analyzed. In *offline* mode, the signal is acquired and processed afterwards. The *pseudo-online* mode processes collected data as if they were received in real-time. The main difference is that the *offline* mode often analyzes the whole data, while the *online* and *pseudo-online* modes only analyze data in short time windows. *Offline* analysis is usually done with asynchronous BCIs, which restricts analysis to predefined time windows. Asynchronous BCI, compatible with *online* and *pseudo-online* modes, allows flexible mental activity duration. *Offline* processing tends to be more accurate, while *online* analysis is better for therapeutic applications. *Pseudo-online* implementation approximates *online* processing without real-time constraints. Many BCI studies being *offline* introduce biases compared to real-life scenarios, impacting classification algorithm performance. *Approach*: The objective of this research paper is therefore to extend the current MOABB framework, operating in *offline* mode, so as to allow a comparison of different algorithms in a *pseudo-online* setting with the use of a technology based on overlapping sliding windows. To do this will require the introduction of a idle state event in the dataset that takes into account all different possibilities that are not task thinking. To validate the performance of the algorithms we will use the normalized Matthews Correlation Coefficient (nMCC) and the Information Transfer Rate (ITR). *Main results*: We analyzed the state-of-the-art algorithms of the last 15 years over several Motor Imagery (MI) datasets composed by several subjects, showing the differences between the two approaches from a statistical point of view. *Significance*: The ability to analyze the performance of different algorithms in *offline* and *pseudo-online* modes will allow the BCI community to obtain more accurate and comprehensive reports regarding the performance of classification algorithms.

Keywords BCI-EEG, Asynchronous BCI, Riemann Geometry, MOABB, Pseudo Online BCI, Deep Learning, Machine Learning.

1 Introduction

Brain Computer Interface (BCI) is a technology that allows a digital device to be controlled through brain activity signals. In recent years, many diverse modalities for acquiring the signal produced by the brain during a specific cognitive task have been developed. In general, we can categorize such procedures into non invasive, with techniques like Electroencephalogram (EEG) [1] or invasive as the recent Endovascular Electrodes [2]. EEG is a non-invasive acquisition technique, with high time resolution and is relatively inexpensive. For these reasons, we will focus on BCI-EEG. During this research, we will focus on Motor Imagery (MI) tasks, i.e., when the user changes his mental activity by thinking of performing a particular body movement, but the overall framework is generic and can be applied in many different BCI contexts.

A BCI technology can operate in 3 different modalities: the *online* mode, which requires to constantly analyze the new input data based on real-time EEG data, the *offline* mode where the signal is first acquired and saved, and then processed later with no real time constraints. Lastly, the *pseudo-online* mode does not process the data in real-time during the experiment but the collected data are processed a posteriori as if they were received online. The main differences are that in the *offline* mode the whole data is available for the analysis, while in the *online* or *pseudo-online* modes, the data is typically analyzed in a short time window running across the signal. The *online* and *pseudo-online* differ by the amount of time that can be used to process the data i.e. the *pseudo-online* method analyzes the same data as the *online* method but with no real time constraint on the processing time.

Offline analysis is usually done with synchronous BCIs, i.e. BCIs that process the signal in predefined time windows where a mental task is performed (e.g. the imagination of a movement) and discards the remaining signal, thus creating a mode of interaction that is unnatural for real-life applications. In contrast, an asynchronous BCI, compatible both with *online* and *pseudo-online* modalities allows a given mental activity to be performed for the duration decided by the subject and not restricted to specific time windows. In particular, such BCIs must be able to distinguish the brain signal between intervals of rest or idle periods vs mental activity. It might as well be able to decide between different types of mental activities.

Usually *offline* processing of the EEG signals turns out to be more accurate, while a *online* signal analysis approach generally produces results that are less accurate but more better suited for use in a therapeutic application [3]. The *pseudo-online* implementation can be used as a methodology that best approximates the *online* process while relaxing the real-time constraint for the processing, thus showing the best attainable performance for a specific BCI task. Many BCI studies are tested only *offline*, thus generating unrealistic performance compared to real-life scenarios [4]. Such approaches introduce an important research bias, as many new classification algorithms created to perform well *offline* lose their competitive nature in real-life applications. It is therefore of particular importance for the advancement of BCI technology that algorithms are validated in *online* or *pseudo-online* mode. Some studies test their algorithms *online*, but their datasets and codes are not always made public, making the data analysis unreproducible. All of this has an extremely negative influence on the speed of progress in the BCI field, making it particularly difficult and complex for people to reproduce published results. As a matter of fact, even just trying to reproduce the performance of state-of-the-art algorithms on a specific dataset is complex and time demanding. In addition, the subjects collected in each dataset are usually few, which is statistically non significant, thus comparing different algorithms on different datasets can produce even antithetical results.

To solve some of these issues for the *offline* mode, the MOABB [5] framework was introduced to test the performance of different classification algorithms on identical datasets and identical preprocessing pipelines. This framework has been a turning point for the BCI community. However, it does not currently include *pseudo-online* testing, thus having a lower impact on *online* BCI quality.

Our first contribution is to propose a *pseudo-online* extension of MOABB, with the use of a technology based on overlapping sliding windows, which also enables the integration of analyses in asynchronous mode.

In addition, we created a performance dashboard comparing the best-known algorithms in BCI classification. This dashboard can be used as a starting point for comparing one's own new algorithm. We took care to test the best state-of-the-art pipelines produced in the last 15 years. This list cannot be understood as definitive but rather as a starting point. To perform this comparison, it was also necessary to extend MOABB, currently based on the scikit-learn library, with the deep learning frameworks of TensorFlow and Keras.

Ultimately, the goal of this paper is to introduce a framework for *pseudo online* analysis of BCI datasets, so as to enable rapid advancement of performance in the BCI community and also

to make the community more inclusive to people with different backgrounds. The framework is showcased with EEG Motor Imagery with 4 datasets but the framework can be easily extended to other MI datasets, data acquisition procedures or even other types of BCIs.

The following article is structured as follows: in 2, we describe the framework and the pipelines considered in the state of the art; 3 lists the results obtained using within-session and cross-session evaluation. Finally, 4 analyzes the implications of the framework and its current limitations; 5 summarizes the results of our study.

2 Methods

In this section, we list the different Motor Imagery (MI) datasets considered in this research, along with the methodology used to transform a regular *offline* dataset into a format suitable for a *pseudo online* analysis. This procedure is applicable to every dataset recorded using normal procedures to be processed *offline*, providing a great versatility to the framework. This concept of extending a synchronously recorded dataset to an asynchronous approach has already been proposed in [6].

We will also explain the concept of paradigm and the possible different evaluation procedures. We will then present the statistical analysis that we used and devote special attention to the metrics considered in that analysis. The differences of the proposed framework with respect to the standard MOABB is described in the Figure 1. The whole project is implemented using Python3 and is based on the use of the MNE [7], PyRiemann [8], scikit-learn [9], TensorFlow [10], MOABB [5] and SciKeras libraries.

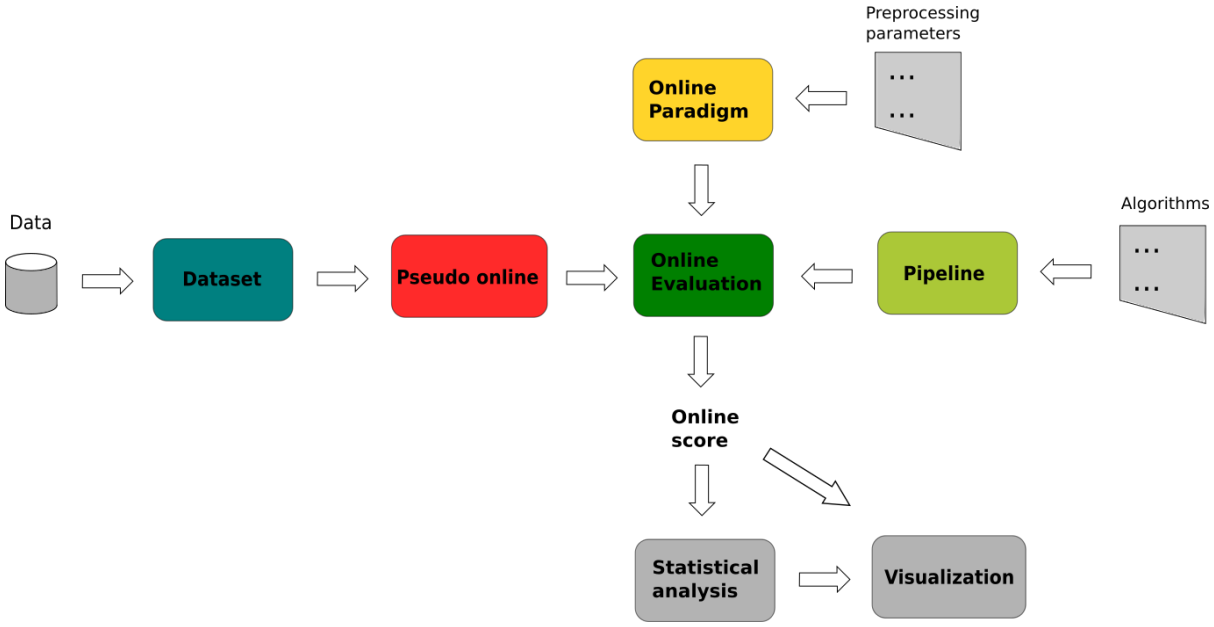


Figure 1: Representation of the framework for the *pseudo-online* architecture, partially inspired by [5].

2.1 Datasets

We consider 4 open-access motor imagery *offline* BCI datasets consisting of several subjects for each dataset and several sessions for each subject. 1 contains all the details about these datasets.

Each of these datasets include a stimulus channel (*stim* channel) that marks an events only when the subject is actively engaged in a task. To align the datasets with online situations, the first step to transform it is to introduce a *nothing* event for each part that is not associated to

Table 1: Datasets considered during this study.

Dataset	Subjects	Channels	Sampling Rate	Sessions	Task
BNCI2014001 [11]	9	22	250 Hz	2	4
BNCI2015001 [12]	12	13	512 Hz	1	2
BNCI2014002 [13]	14	15	512 Hz	5	2
BNCI2014004 [14]	9	3	250 Hz	1	2

a task. This inclusion allows for a performance evaluation that better reflects real-life scenarios. In practical applications, individuals may have periods where they actively attempt to perform a task, while they may be engaged in various unrelated thoughts such as daydreaming at other times. The *nothing* event is designed to encapsulate these diverse possibilities that are not task-related.

The introduction of the *nothing* event, however, introduces an important issue; the dataset now turns out to be strongly unbalanced toward that new class. We will analyze this problem and propose possible solutions in 2.3.1.

Using this procedure, we were able to test algorithms for the classification of 5 MI tasks (BNCI2014001) or 3 MI tasks (BNCI2014002, BNCI2014004, BNCI2015001).

2.2 Paradigm

Following the line drawn by MOABB, we consider the paradigm as a way to transform continuous data to trials, i.e., the basic elements for any machine learning algorithm. In addition, the paradigm is used to set the preprocessing of the continuous data, keeping it unique for all datasets and all subjects considered in order to allow a fair comparison.

To enable the framework to operate in pseudo-online mode, an extension of the methodology is required, taking into account that tasks can vary in duration. This extension is achieved by employing a sliding window approach. In the MOABB framework, a single trial is extracted for each task performed by the subject, resulting in a pure signal consisting of only one epoch extracted from each task. However, in order to achieve a performance evaluation that closely resembles a real BCI application, it is necessary to transform the dataset into an asynchronous format using an overlapping sliding windows approach. The idea is to select a sliding windows with size T that is smaller than the total time of the task and run it on the continuous data with a step size, which is controlled using the overlapping parameter (see 2). In general, the optimal values for the length of the sliding window and the overlapping is a trade-off between accuracy and response speed.

The sliding window approach can be seen as a data augmentation procedure. This approach actually generates a significantly greater number of trials per class compared to the original method [15].

Implementing the sliding windows approach introduces the challenge of generating windows that contain a mixture of events, as the sliding operation spans across the entire continuous data. To deal with this issue, we had to assign a unique label to these events, in order to create a dataset that is compatible with most machine learning algorithm. The assignment of this unique label is based on the percentage of data contained within a specific mixed window. We consider that windows are small enough so that only two events can appear in a window. Let us call a and b respectively the percentages of the window with the first and second events:

- If $a > b$, we assign the label of the initial event.
- If $a \leq b$, we assign the label of the last event. This label is used in case of equality because the subject intent is to perform the new task.

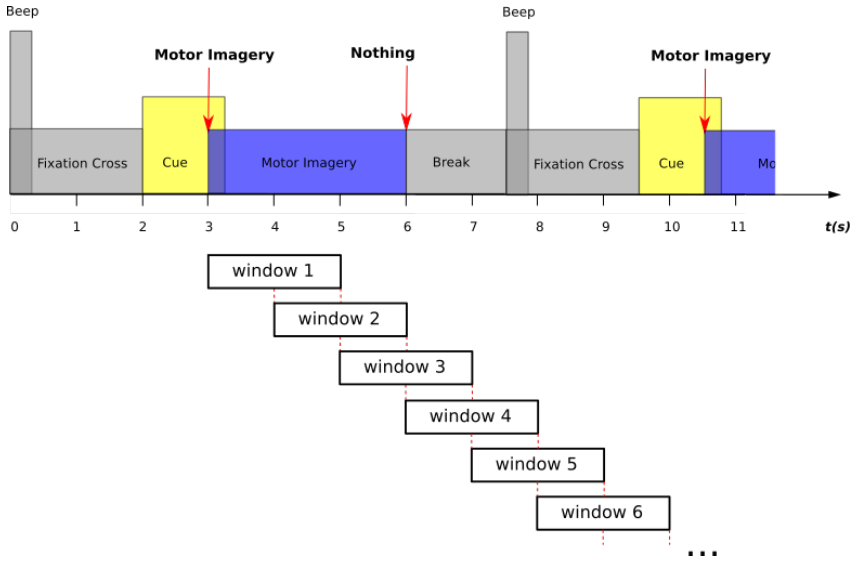


Figure 2: Figure explaining the introduction of the *nothing* event and the sliding windows in the BCI Dataset 2a (BNCI2014001) [11]. In this example, we use a window of 2 seconds with an overlapping of 50%

A detailed representation of this procedure can be found in 3.

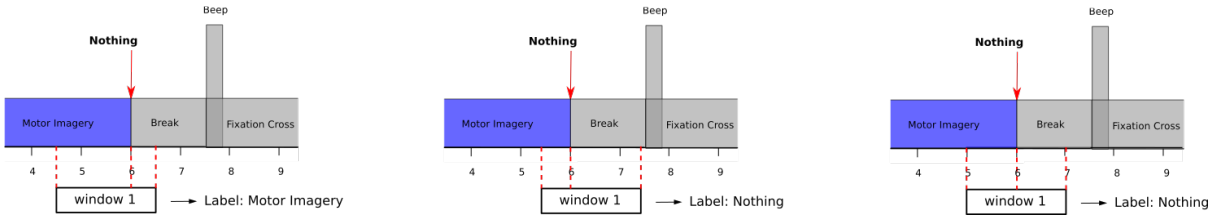


Figure 3: The different ways to treat the windows that contain two events.

We enforce the evaluation process to include all the different tasks plus the new *nothing* task, generating a non binary classification. This problem will be taken in account in 2.3.1.

2.3 Evaluation

Having split the continuous data using the sliding window approach, we are now ready to evaluate the performance of several algorithms on these modified datasets. In this section, we discuss the metrics used as well as the possible different evaluations types: Within-Session and Cross-Session.

2.3.1 Metrics

The transformation of the dataset introduces an important issue: the transformed dataset is strongly unbalanced with respect to the *nothing* event. Furthermore, the transformation always introduces a new class, so that we have to deal with non-binary classification throughout our processing.

Different solutions are possible in such situations. Collecting a large amount of data for BCI purposes can be expensive and time-consuming, making data cancellation (randomly delete samples from the majority class to balance the class distribution) an impractical option. To solve this problem, we adopted a metric that is reliable with unbalanced datasets [16]. The standard measure used in BCI is accuracy, which gives reliable results for balanced datasets. When this

condition fails, accuracy produces an overly optimistic performance estimation. While such a metric is perfectly adequate to evaluate the performance in a synchronized BCI where usually the datasets are balanced, it no longer is in an asynchronous setting. In such a situation, the BCI literature recommends the use of Cohen's Kappa coefficient [16, 17]. There are however other measures to deal with unbalanced datasets [18] such as Matthews Correlation Coefficient (MCC) [19], which was introduced by Matthews in the case of binary classification [19] and generalized to multi-class problems [20]. Recent research has shown that Cohen's Kappa and MCC performance measures are very similar in most situations, but may differ in others. This leads to anomalous performance for Cohen's Kappa in certain situations, which is why we preferred to use MCC [21, 22]. In addition, MCC has been shown to be much more informative than several metrics including ROC-AUC in binary classification [23]. In the case of binary classification MCC is defined as:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}\ , \tag{1}$$

where TP, TN, FP and FN are respectively the number of true positives, true negatives, false positives and false negatives defined using the confusion matrix. MCC lies in the range [-1, +1], with -1 and +1 being reached respectively in case of perfect misclassification and perfect classification. MCC = 0 is the expected value for the coin tossing classifier [24]. We decided to use the normalized version of MCC in the framework, the nMCC is defined as nMCC = $\frac{\text{MCC}+1}{2}$.

Such normalization projects the original range [-1,+1] into the interval [0, +1], where +1 correspond to a perfect classification while $nMCC = 0.5$ is for prediction similar to random guessing. Identical considerations apply to its extension to multi label classification.

The performance of a BCI system can also be evaluated by how much information can be transferred without committing errors in a specific time frame, i.e., the bit-rate of the system. Usually, the information transfer rate (ITR) is used, of which two definitions have been formulated: the first, which was proposed by Wolpaw *et al* [25] includes the use of accuracy in its definition and thus is based on the same assumptions as accuracy. In our situation, we therefore decided to adopt the definition of ITR given by Nykopp [26, 27], that is based on the concept of Mutual Information (MI) [28]. The MI between two discrete random variables X and Y is defined as

$$MI(X;Y) = \sum_{y \varepsilon Y} \sum_{x \varepsilon X} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{2}$$

where $p(x, y)$ is the joint probability of realizing events x and y simultaneously and $p(x)$ and $p(y)$ is the probability associated with the individual variables. The logarithm used in this context is base two, as information is measured and conveyed in bits. ITR is then defined as the amount of information transmitted per minute (bits/minute)

$$ITR = MI(X,Y)\frac{60}{T} \tag{3}$$

where T (seconds/symbol) is the time in seconds needed to transmit a symbol, in our case to select a task. Similar considerations apply to its extension to multi label classification.

2.3.2 Within-Session Evaluation Procedure

The Within-Session evaluation procedure involves evaluating performance directly within the same session of a certain subject. The current evaluation method employed in MOABB utilizes a 5-fold Cross Validation. However, in the Pseudo-Online extension, we decided to not use Cross Validation as our objective is to explicitly preserve the causal relationship within the data. Therefore, we enforced that the test dataset temporally follows the training portion. For

the same reason, we also decided to not shuffle the data. Since, in the datasets considered, there is not a predefined split in the training and test part, we followed the state-of-the-art procedure, which is to split the dataset into a training dataset containing the first 80% windows and a test dataset containing the 20% remaining ones. For the hyper-parameter, we used a 5-fold cross validation on the training dataset.

2.3.3 Cross-Session Evaluation Procedure

The Cross-Session evaluation procedure focuses on a single subject and incorporates all sessions except one for the training phase, while utilizing the remaining session for the testing phase. This approach is carried out with a Leave One Out Cross Validation.

In order to allow a complete fairness of the approach, we performed a Nested Cross Validation when the hyper parameter tuning was necessary [29]. However, it is noticeable that the performance obtained using the nested approach is statistically similar to that obtained with the less computationally intensive flat cross-validation approach, a finding that is aligned with some recent research [30]. We chose to produce the Nested Cross Validation results, but those obtained using the flat cross-validation are given for comparison in the appendix in Table 8.

2.3.4 Pipelines Considered

We considered different state of the art pipelines for Motor Imagery classification in BCI described in 2. This list covers the algorithms that have shown good classification performance in the last 15 years. This list is not intended as definitive but rather as a starting point: each research group will be able to add its algorithm to the dashboard, after testing it/them in the same setting.

To perform this comparison, it was also necessary to extend MOABB, currently based on the sole scikit-learn library to enable it to use the Deep Learning (DL) frameworks of TensorFlow and Keras.

In recent years, DL algorithms became increasingly popular for solving extremely complex problems that could not be solved by traditional Machine Learning (ML) approaches. The popularity of such algorithms is due to recent successes in a wide variety of fields, from Natural Language processing [31] to image recognition [32]. Only recently have such algorithms begun to be applied for BCI classification.

Conventional ML – non DL – algorithms are not suitable to process directly the raw data. There is usually a feature extraction step that is designed with some domain expertise. On the contrary, DL models have shown remarkable capabilities in automatically learning and extracting relevant features from raw data, such as EEG signals. In addition, these models proved to be particularly adaptable and generelizable to new subjects and sessions. Despite their potential, DL models are not extensively used in the field of BCI due to several challenges: they typically demand a substantial volume of training data, which can be difficult and costly to acquire in BCI due to the specialized equipment and expertise required. Additionally, they are often perceived as black boxes, lacking interpretability and making it challenging to understand the decision-making process. In BCI applications, interpretability is essential for fostering user trust, gaining clinical acceptance, and enabling effective feedback mechanisms.

To replicate the state of the art, we used the Keras [33] framework and the KerasClassifier function from the SciKeras package. With this library, it is possible to create a deep learning architecture and convert it into a scikit-learn pipeline that can be integrated directly into the standard MOABB framework. In order to make the results more stable, every deep learning pipeline is preceded by a standardization step which puts each channel to zero mean and unit standard deviation. We also apply a re-sampling procedure to ensures that each architecture incorporates a temporal filter that is aligned with the implementation provided in the state-of-the-art techniques. Moreover, the same sliding window parameters were used for all algorithms

in order to allow for a fair comparison. Details on the DL hyper parameters are given in Table 3.

Table 2: Pipelines considered in this study.

Name Pipeline	Feature Extraction	Classifier	References
MDM	Spatial Covariance estimated with Sample Estimator	Mean Distance to Mean (MDM)	[34]
Cov + EN	Spatial Covariance estimated with Sample Estimator mapped to Tangent Space	Optimized Elastic Network (EL)	[35]
FgMDM	Spatial Covariance estimated with Sample Estimator	Minimum Distance to Mean with geodesic filtering (FgMDM)	[34]
TANG + SVM	Spatial Covariance estimated with Sample Estimator mapped to Tangent Space	Optimized Support Vector Machine (SVM)	[34]
AUG + TANG + SVM	Augmented Spatial Covariance estimated with Sample Estimator mapped to Tangent Space	Optimized SVM	[36]
CSP + LDA	Common Spatial Patterns (CSP)	Optimized shrinkage Linear Discriminant Analysis (LDA)	[37]
CSP + RF	CSP	Optimized Random Forest (RF)	[37]
CSP + SVM	CSP	Optimized SVM	[37]
AR + SVM	Autoregressive Coefficient	Optimized SVM	[38]
AR + LR	Autoregressive Coefficient	Optimized Linear Regression (LR)	[38]
FBCSP+LDA	Filter Bank Common Spatial Patterns (FBCSP)	Optimized Shrinkage LDA	[39, 37]
FBCSP+SVM	FBCSP	Optimized SVM	[39]
FBCSP+MLP	FBCSP	MLP	[39]
FBCSP+RF	FBCSP	Optimized RF	[39]
ShallowConvNet	Standardized and resample EEG signal at 250Hz	CNN	[40]
DeepConvNet	Standardized and resample EEG signal at 250Hz	CNN	[40]
EEGNet 8 2	Standardized and resample EEG signal at 128Hz	CNN with architecture EEGNet	[41]
EEGTCNet	Standardized and resample EEG signal at 250Hz	CNN with architecture EEGTCNet	[42]
EEGITNet	Standardized and resample EEG signal at 128Hz	CNN with architecture EEGITNet	[43]
EEGNeX 8 32	Standardized and resample EEG signal at 128Hz	CNN with architecture EEGNeX	[44]

When possible, the hyper-parameters of the classification models were optimized using a Grid Search procedure. We did not create an ablation study for the DL models since we faithfully reproduced the architectures proposed in the respective references.

3 Results

In this section, we report the performance results obtained with the pipelines considered. Ultimately, to validate the robustness and validity of our *pseudo-online* approach, we tested the algorithms on different datasets, subjects and tasks.

3.1 Paradigm

The sliding window is defined to have a 2 s duration windows with a 50% overlapping.

Table 3: Common parameters for DL pipelines.

Parameter	**Value**
Epoch	300
Batch Size	64
Validation Split	0.2
Loss	Sparse Categorical Crossentropy
Optimizer	Adam Learning Rate = 0.0009
Callbacks ES	Early Stopping Patience = 75 Monitor = Validation Loss
Callbacks LR	ReduceLROnPlateau Patience = 75 Monitor = Validation Loss Factor = 0.5

Except for the filter-bank base algorithms, we applied on all datasets a standard preprocessing – for the motor imagery task – band-pass filter in the region [8; 30] Hz. For pipelines based on the Filter Bank paradigm, we used 6 different non overlapping windows in order to filter the EEG signal into 8–12 Hz, 12–16 Hz, 16–20 Hz, 20–24 Hz, 24–28 Hz, 28–35 Hz.

3.2 Pseudo Online Evaluation

3.2.1 Within-Session Evaluation

We give the results of the *pseudo-online* evaluation using the within-session methodology in Table 4. These results are also displayed in the appendix (8, 9, 10 and 11), showing also a detailed study of the statistical significance.

Table 4: Performance Pseudo Online Within-Session Evaluation. Results for the DL architecture are listed after the two line.

Pipeline	BNCI2014002	BNCI2014004	BNCI2015001	BNCI2014001
MDM	0.66 ± 0.09	0.63 ± 0.06	0.72 ± 0.07	0.67 ± 0.05
Cov + EN	0.65 ± 0.09	0.62 ± 0.08	0.73 ± 0.08	0.69 ± 0.07
FgMDM	0.67 ± 0.09	0.64 ± 0.06	0.74 ± 0.07	0.70 ± 0.07
TANG + SVM	0.69 ± 0.10	0.61 ± 0.08	0.74 ± 0.08	0.70 ± 0.08
AUG + TANG + SVM	$\mathbf{0.70 \pm 0.10}$	$\mathbf{0.66 \pm 0.09}$	$\mathbf{0.76 \pm 0.08}$	0.70 ± 0.07
CSP + LDA	0.58 ± 0.10	0.60 ± 0.08	0.65 ± 0.09	0.57 ± 0.05
CSP + RF	0.59 ± 0.10	0.61 ± 0.06	0.65 ± 0.08	0.56 ± 0.05
CSP + SVM	0.59 ± 0.10	0.60 ± 0.08	0.65 ± 0.08	0.57 ± 0.05
FBCSP+LDA	0.69 ± 0.10	0.65 ± 0.09	0.74 ± 0.08	$\mathbf{0.71 \pm 0.06}$
FBCSP+SVM	0.69 ± 0.11	0.64 ± 0.09	0.74 ± 0.09	0.70 ± 0.07
FBCSP+MLP	0.67 ± 0.11	0.65 ± 0.08	0.72 ± 0.09	0.69 ± 0.06
FBCSP+RF	0.69 ± 0.10	0.63 ± 0.08	0.73 ± 0.09	0.68 ± 0.07
ShallowConvNet	0.64 ± 0.10	0.58 ± 0.06	0.73 ± 0.08	0.69 ± 0.08
DeepConvNet	0.61 ± 0.09	0.58 ± 0.05	0.63 ± 0.08	0.61 ± 0.07
EEGNet 8 2	0.65 ± 0.08	0.59 ± 0.07	0.73 ± 0.07	0.67 ± 0.07
EEG ITNet	0.61 ± 0.08	0.57 ± 0.06	0.68 ± 0.08	0.64 ± 0.08
EEG TCNet	0.64 ± 0.10	0.58 ± 0.08	0.70 ± 0.08	0.66 ± 0.07
EEGNeX 8 32	0.58 ± 0.08	0.56 ± 0.06	0.66 ± 0.07	0.61 ± 0.07

3.2.2 Cross-Session Evaluation

We give the results of the *pseudo-online* evaluation using the Cross-Session methodology in Table 5. These results are also displayed in the appendix (12, 13 and 14), showing also a detailed study of the statistical significance.

Table 5: Performance Pseudo Online Cross-Session Evaluation Using Nested Cross Validation. Results for the DL architecture are listed after the two line.

Pipeline	BNCI2014004	BNCI2015001	BNCI2014001
MDM	0.61 ± 0.06	0.67 ± 0.07	0.65 ± 0.05
Cov + EN	0.59 ± 0.07	0.69 ± 0.07	0.67 ± 0.06
FgMDM	0.62 ± 0.06	0.69 ± 0.06	0.67 ± 0.06
TANG + SVM	0.57 ± 0.07	0.68 ± 0.07	0.65 ± 0.08
AUG + TANG + SVM	**0.63 ± 0.08**	**0.72 ± 0.06**	**0.69 ± 0.06**
CSP + LDA	0.58 ± 0.07	0.61 ± 0.07	0.57 ± 0.05
CSP + RF	0.58 ± 0.05	0.61 ± 0.07	0.56 ± 0.04
CSP + SVM	0.57 ± 0.07	0.61 ± 0.07	0.56 ± 0.05
FBCSP+LDA	0.62 ± 0.07	0.70 ± 0.07	0.68 ± 0.06
FBCSP+SVM	0.61 ± 0.07	0.68 ± 0.08	0.67 ± 0.06
FBCSP+MLP	0.62 ± 0.06	0.68 ± 0.08	0.65 ± 0.06
FBCSP+RF	0.61 ± 0.07	0.67 ± 0.08	0.66 ± 0.06
ShallowConvNet	0.55 ± 0.05	0.70 ± 0.07	0.69 ± 0.06
DeepConvNet	0.56 ± 0.04	0.64 ± 0.05	0.62 ± 0.07
EEGNet 8 2	0.56 ± 0.05	0.70 ± 0.06	0.67 ± 0.07
EEG ITNet	0.55 ± 0.05	0.68 ± 0.07	0.65 ± 0.07
EEG TCNet	0.56 ± 0.06	0.70 ± 0.06	0.66 ± 0.06
EEGNeX 8 32	0.55 ± 0.05	0.65 ± 0.06	0.61 ± 0.06

4 Discussion

The numerous results performed on different datasets with different classification tasks, both binary and multi-class, showed a good alignment with state-of-the-art results. Overall, the method that shows the best performance is the augmented covariance method with classification using SVM on the tangent space [36]. In some situations, the FBCSP-based algorithms also obtain comparable results (BNCI2014001, BNCI2014002), while in the remaining considered datasets (BNCI2014004, BNCI2015001), the performance produced by the augmented covariance method appears to be superior.

However, the augmented covariance method depends on the selection of two hyper parameters with a grid search which is computationally intensive due to the increasing size of the augmented covariance matrix with the order parameter [45].

In this analysis, it is shown that in general DL algorithms have lower performance than standard machine learning algorithms. One possible explanation is that more data are needed to train the models efficiently, eventually using data augmentation procedures, such as introducing random Gaussian noise on the data or time inversion procedures [46].

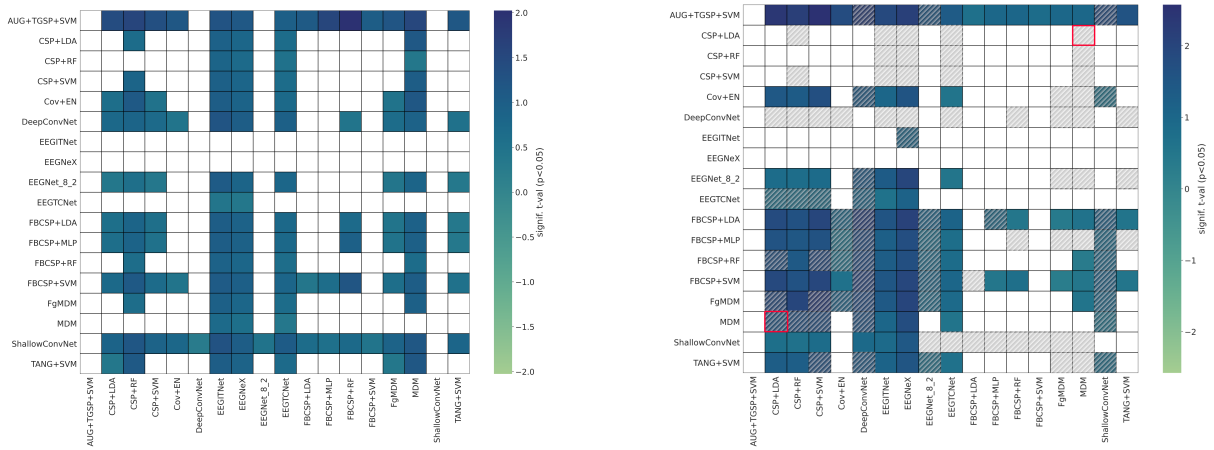
4.1 Comparison Pseudo Online vs Offline Evaluation

In general, it is observed that the performance achieved through pseudo-online evaluation is lower compared to offline evaluation. This can be attributed to several factors. Firstly, in order to enable real-time applications, the duration of epoch sliding window is typically reduced, which can impact the accuracy of classification. Secondly, the introduction of the *nothing* event introduces an additional class to the already complex classification task. The *nothing* event encompasses a wide range of mental phenomena, resulting in high variability within this class, which in turn affects the overall performance of the classification task.

4.1.1 Within-Session Evaluation

We want to emphasize the change in performance and ranking for the best algorithms using the standard offline against the pseudo-online using the Within-Session evaluation methodology, both using nMCC as metric. The performance of the *offline* methodology turns out to be much better compared to the *pseudo-online* evaluation as shown in Tables 6 and 4.

A detailed analysis of Figure 4 reveals several noteworthy differences indicated by the gray regions in Figure 4(b). Firstly, the ACM methodology outperforms the state of the art approaches in both *offline* and *pseudo online* evaluations. Secondly, some DL algorithms seem to be more stable in the *pseudo online* approach for the BNCI2014001 dataset. This can be attributed to the utilization of sliding windows as a data augmentation technique, highlighting the significant dependence of – at least some – DL algorithms on large amounts of data. The ranking of certain algorithms is completely opposite: the red box in Figure 4(b) representing the pipelines "CSP+LDA" and "MDM" are an example of such a ranking change. Figure 5 presents a meta-analysis of these two pipelines. The results demonstrate the superiority of the "CSP+LDA" pipeline in the *offline* evaluation that is completely reversed in the *pseudo-online* approach.



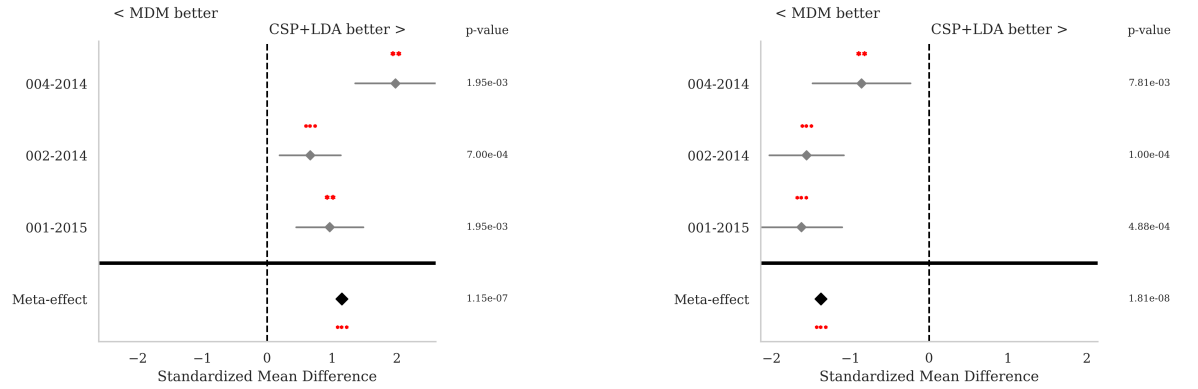
(a) (b)

Figure 4: Result for 3 task classification using results of dataset BNCI2015001, BNCI2014002 and BNCI2014004 using Within-Session evaluation. Plot (a) shows the meta analysis of the different methods considered using offline evaluation. Plot (b) shows the meta analysis of the different methods considered using pseudo online evaluation. The grey zone is where we find a statistical significant difference between the two evaluations. Red boxes indicate the behaviour analyzed in 5, which plots the significance that the algorithm on the y-axis is better than the one on the x-axis. The color represents the significance level of the difference of accuracy, in terms of t-values, and we show only the significant interactions ($p < 0.05$).

4.1.2 Cross-Session Evaluation

We made the same analysis of change in performance and ranking for the best algorithms for the Cross-Session evaluation. To compare the performance between *offline* and *pseudo-online* refer to Tables 7 and 5.

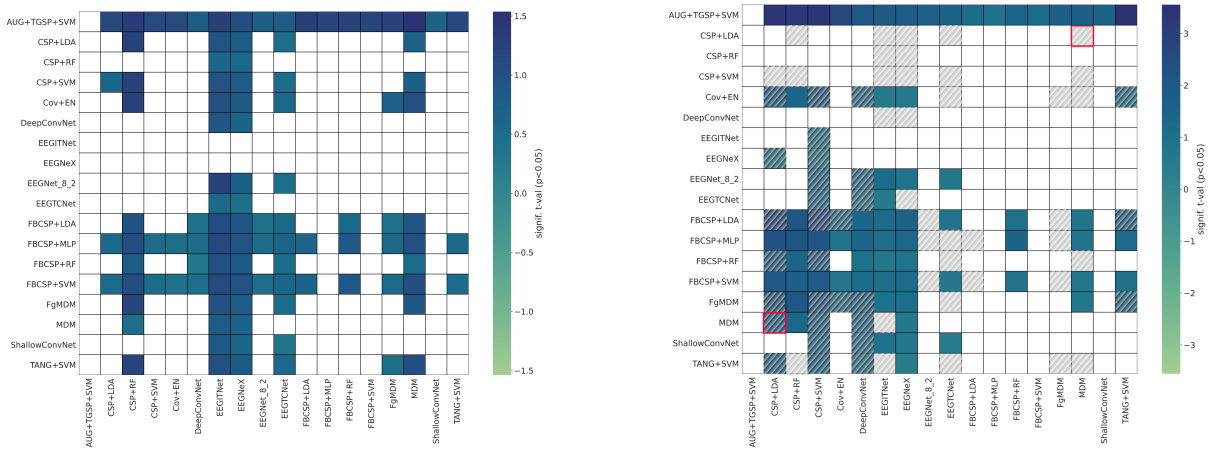
A detailed analysis of Figure 6 reveals several noteworthy differences indicated by the gray regions in Figure 6(b). The findings in the Cross-Session case align with our previous observations. In this case also, we noticed a complete reversal in the ranking of certain algorithms compared to the results highlighted in Figure 6(b). Specifically, the red boxes representing the pipelines "CSP+LDA" and "MDM" exemplify this ranking discrepancy. To further analyze



(a) (b)

Figure 5: Result for Meta Analysis of CSP+LDA vs MDM on 3 task classification using results of dataset BNCI2015001, BNCI2014002 and BNCI2014004 using Within-Session evaluation. Plot (A) shows the meta analysis using offline evaluation. Plot (B) shows the meta analysis using pseudo online evaluation. We show the standardized mean differences, while p-values are computed as one-tailed Wilcoxon signed-rank test for the hypothesis given as title of the plot and the gray bar denote 95% interval. Here, * stands for $p < 0.05$, ** for $p < 0.01$, and *** for $p < 0.001$.

these pipelines, we conducted a meta-analysis, in Figure 7. The outcome of this analysis clearly demonstrates the superiority of the "CSP+LDA" pipeline in the *offline* evaluation, whereas the *pseudo-online* approach completely reverses this superiority.



(a) (b)

Figure 6: Result for 3 task classification using results of dataset BNCI2015001 and BNCI2014004 using Cross-Session evaluation. Plot (a) shows the meta analysis of the different methods considered using offline evaluation. Plot (b) shows the meta analysis of the different methods considered using pseudo online evaluation. The grey zone is where we find a statistical significant difference between the two evaluations. Red boxes indicate the behaviour analyzed in 7, which plots the significance that the algorithm on the y-axis is better than the one on the x-axis. The color represents the significance level of the difference of accuracy, in terms of t-values, and we show only the significant interactions ($p < 0.05$).

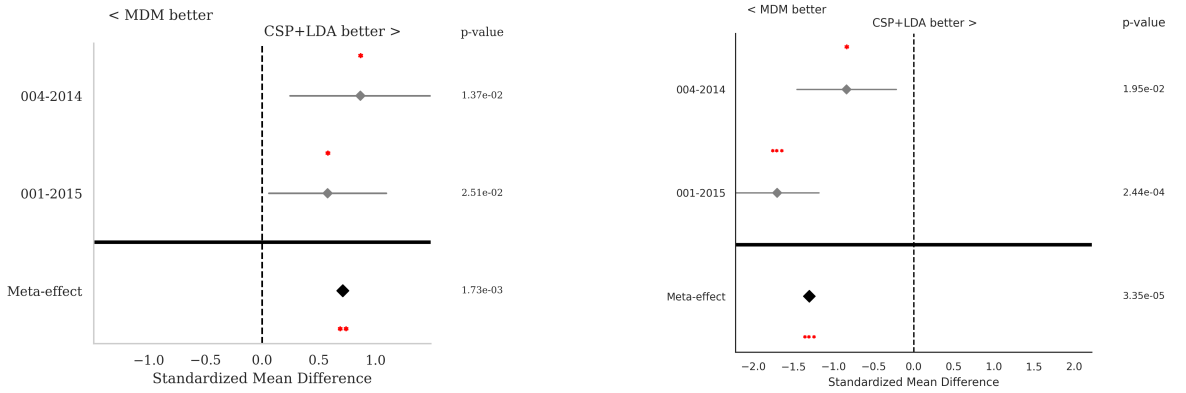


Figure 7: Result for Meta Analysis of CSP+LDA vs MDM on 3 task classification using results of dataset BNCI2015001 and BNCI2014004 using Cross-Session evaluation. Plot (a) shows the meta analysis using offline evaluation. Plot (b) shows the meta analysis using pseudo online evaluation. We show the standardized mean differences, while p-values are computed as one-tailed Wilcoxon signed-rank test for the hypothesis given as title of the plot and the gray bar denote 95% interval. Here, * stands for $p < 0.05$, ** for $p < 0.01$, and *** for $p < 0.001$.

5 Conclusion

In this research, we introduced an extension of the current MOABB framework in order to provide a framework to test different algorithms in a *pseudo-online* evaluation. In particular, this modification is based on the use of an overlapping sliding windows approach and on the introduction of an *idle* state in the normal Motor Imagery datasets. In order to verify the functioning of such a framework, we tested some of the most efficient algorithms produced in the state-of-the-art of the last 15 years using both ML and DL algorithms. With such a statistical analysis, we show how the augmented covariance approach produces superior performance compared to the state of the art, considering different classification task and different evaluation procedures. We also showed that the efficiency and ranking of the algorithms is highly dependent on the type of analysis – *offline* or *pseudo-online* – performed. The *pseudo-online* mode also exhibited some more stable performance for some combinations of DL algorithms and datasets. In conclusion, the ability to analyze the performance of various algorithms in both *offline* and *pseudo-online* modes can significantly accelerate the progress of classification algorithms in the BCI community. By conducting evaluations in *offline* mode initially and then validating the results in *pseudo-online* mode, researchers can effectively enhance the performance of these algorithms. This iterative approach enables the identification of strengths, weaknesses, and areas for improvement, leading to advancements in BCI classification algorithms at a faster pace.

Acknowledgment

This work has been partially funded by a EUR DS4H/Neuromod fellowship. The authors are grateful to the OPAL infrastructure from Université Côte d'Azur for providing resources and support. We would like to thanks S. Chevallier for supporting the idea of integrating this approach in MOABB.

Data and Code Availability

The code will be soon integrated[1] in the MOABB library.

References

[1] Hans Berger. "Über das elektroenkephalogramm des menschen". In: *Archiv für psychiatrie und nervenkrankheiten* 87.1 (1929), pp. 527–570.

[2] Thomas J Oxley et al. "Motor neuroprosthesis implanted with neurointerventional surgery improves capacity for activities of daily living tasks in severe paralysis: first in-human experience". In: *Journal of neurointerventional surgery* 13.2 (2021), pp. 102–108.

[3] Marisol Rodrıéguez-Ugarte et al. "Personalized offline and pseudo-online BCI models to detect pedaling intent". In: *Frontiers in neuroinformatics* 11 (2017), p. 45.

[4] Janne Lehtonen et al. "Online classification of single EEG trials during finger movements". In: *IEEE Transactions on Biomedical Engineering* 55.2 (2008), pp. 713–720.

[5] Vinay Jayaram and Alexandre Barachant. "MOABB: trustworthy algorithm benchmarking for BCIs". In: *Journal of neural engineering* 15.6 (2018), p. 066011.

[6] E.B. Sadeghian and M.H. Moradi. "Continuous Detection of Motor Imagery in a Four-Class Asynchronous BCI". In: *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*. 2007, pp. 3241–3244. DOI: 10.1109/IEMBS.2007.4353020.

[7] Alexandre Gramfort et al. "MEG and EEG Data Analysis with MNE-Python". In: *Frontiers in Neuroscience* 7.267 (2013), pp. 1–13. DOI: 10.3389/fnins.2013.00267.

[8] A. Barachant and J-R. King. *pyRiemann v0.2.2*. June 2015. DOI: 10.5281/zenodo.18982. URL: http://dx.doi.org/10.5281/zenodo.18982.

[9] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[10] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[11] Michael Tangermann et al. "Review of the BCI competition IV". In: *Frontiers in neuroscience* (2012), p. 55.

[12] Josef Faller et al. "Autocalibration and recurrent adaptation: Towards a plug and play online ERD-BCI". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 20.3 (2012), pp. 313–319.

[13] David Steyrl et al. "Random forests in non-invasive sensorimotor rhythm brain-computer interfaces: a practical and convenient non-linear classifier". In: *Biomedical Engineering/Biomedizinische Technik* 61.1 (2016), pp. 77–86.

[14] R. Leeb et al. "Brain–computer communication: motivation, aim, and impact of exploring a virtual apartment". In: *Neural Systems and Rehabilitation Engineering, IEEE Transactions on* 15.4 (2007), pp. 473–482.

[15] Elnaz Lashgari, Dehua Liang, and Uri Maoz. "Data augmentation for deep-learning-based electroencephalography". In: *Journal of Neuroscience Methods* 346 (2020), p. 108885.

[16] Eoin Thomas, Matthew Dyson, and Maureen Clerc. "An analysis of performance evaluation for motor-imagery based BCI". In: *Journal of Neural Engineering* 10.3 (June 2013). DOI: 10.1088/1741-2560/10/3/031001. URL: http://hal.inria.fr/hal-00821971.

[1] Will be changed to "is integrated" for final publication.

[17] Jacob Cohen. "A coefficient of agreement for nominal scales". In: *Educational and psychological measurement* 20.1 (1960), pp. 37–46.

[18] Qiuming Zhu. "On the performance of Matthews correlation coefficient (MCC) for imbalanced dataset". In: *Pattern Recognition Letters* 136 (2020), pp. 71–80.

[19] Brian W Matthews. "Comparison of the predicted and observed secondary structure of T4 phage lysozyme". In: *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405.2 (1975), pp. 442–451.

[20] Jan Gorodkin. "Comparing two K-category assignments by a K-category correlation coefficient". In: *Computational biology and chemistry* 28.5-6 (2004), pp. 367–374.

[21] Rosario Delgado and Xavier-Andoni Tibau. "Why Cohen's Kappa should be avoided as performance measure in classification". In: *PloS one* 14.9 (2019), e0222916.

[22] Davide Chicco, Matthijs J Warrens, and Giuseppe Jurman. "The Matthews correlation coefficient (MCC) is more informative than Cohen's Kappa and Brier score in binary classification assessment". In: *IEEE Access* 9 (2021), pp. 78368–78381.

[23] Davide Chicco and Giuseppe Jurman. "The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification". In: *BioData Mining* 16.1 (2023), pp. 1–23.

[24] Davide Chicco and Giuseppe Jurman. "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation". In: *BMC genomics* 21.1 (2020), pp. 1–13.

[25] J.R. Wolpaw et al. "EEG-based communication: improved accuracy by response verification". In: *Rehabilitation Engineering, IEEE Transactions on* 6.3 (1998), pp. 326–333.

[26] Tommi Nykopp et al. "Statistical modelling issues for the adaptive brain interface". In: *Helsinki: Helsinki University of Technology* (2001).

[27] Sahar Sadeghi and Ali Maleki. "Accurate estimation of information transfer rate based on symbol occurrence probability in brain-computer interfaces". In: *Biomedical Signal Processing and Control* 54 (2019), p. 101607.

[28] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. "Estimating mutual information". In: *Physical review E* 69.6 (2004), p. 066138.

[29] Gavin C Cawley and Nicola LC Talbot. "On over-fitting in model selection and subsequent selection bias in performance evaluation". In: *The Journal of Machine Learning Research* 11 (2010), pp. 2079–2107.

[30] Jacques Wainer and Gavin Cawley. "Nested cross-validation when selecting classifiers is overzealous for most practical applications". In: *Expert Systems with Applications* 182 (2021), p. 115222.

[31] Ronan Collobert et al. "Natural language processing (almost) from scratch". In: *Journal of machine learning research* 12.ARTICLE (2011), pp. 2493–2537.

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[33] François Chollet et al. *Keras*. `https://keras.io`. 2015.

[34] Alexandre Barachant et al. "Riemannian geometry applied to BCI classification". In: *International conference on latent variable analysis and signal separation*. Springer. 2010, pp. 629–636.

[35] Marie-Constance Corsi et al. "Functional connectivity ensemble method to enhance BCI performance (FUCONE)". In: *IEEE Transactions on Biomedical Engineering* (2022).

[36] Igor Carrara and Théodore Papadopoulo. "Classification of BCI-EEG based on augmented covariance matrix". In: *arXiv preprint arXiv:2302.04508* (2023).

[37] Fabien Lotte et al. "A Review of Classification Algorithms for EEG-based Brain-Computer Interfaces: A 10-year Update". In: *Journal of Neural Engineering* (Apr. 2018), p. 55. URL: https://hal.inria.fr/hal-01846433.

[38] Mohammed Zeki Al Faiz and Ammar A Al-Hamadani. "Online brain computer interface based five classes EEG to control humanoid robotic hand". In: *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*. IEEE. 2019, pp. 406–410.

[39] Kai Keng Ang et al. "Filter bank common spatial pattern (FBCSP) in brain-computer interface". In: *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE. 2008, pp. 2390–2397.

[40] Robin Tibor Schirrmeister et al. "Deep learning with convolutional neural networks for EEG decoding and visualization". In: *Human brain mapping* 38.11 (2017), pp. 5391–5420.

[41] Vernon J Lawhern et al. "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces". In: *Journal of neural engineering* 15.5 (2018), p. 056013.

[42] Thorir Mar Ingolfsson et al. "EEG-TCNet: An accurate temporal convolutional network for embedded motor-imagery brain–machine interfaces". In: *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2020, pp. 2958–2965.

[43] Abbas Salami, Javier Andreu-Perez, and Helge Gillmeister. "EEG-ITNet: An Explainable Inception Temporal Convolutional Network for Motor Imagery Classification". In: *IEEE Access* 10 (2022), pp. 36672–36685.

[44] Xia Chen et al. "Toward reliable signals decoding for electroencephalogram: A benchmark study to EEGNeX". In: *arXiv preprint arXiv:2207.12369* (2022).

[45] Kisung You and Hae-Jeong Park. "Geometric learning of functional brain network on the correlation manifold". In: *Scientific reports* 12.1 (2022), pp. 1–13.

[46] Cédric Rommel et al. "Data augmentation for learning predictive models on EEG: a systematic comparison". In: *Journal of Neural Engineering* 19.6 (2022), p. 066020.

Appendix

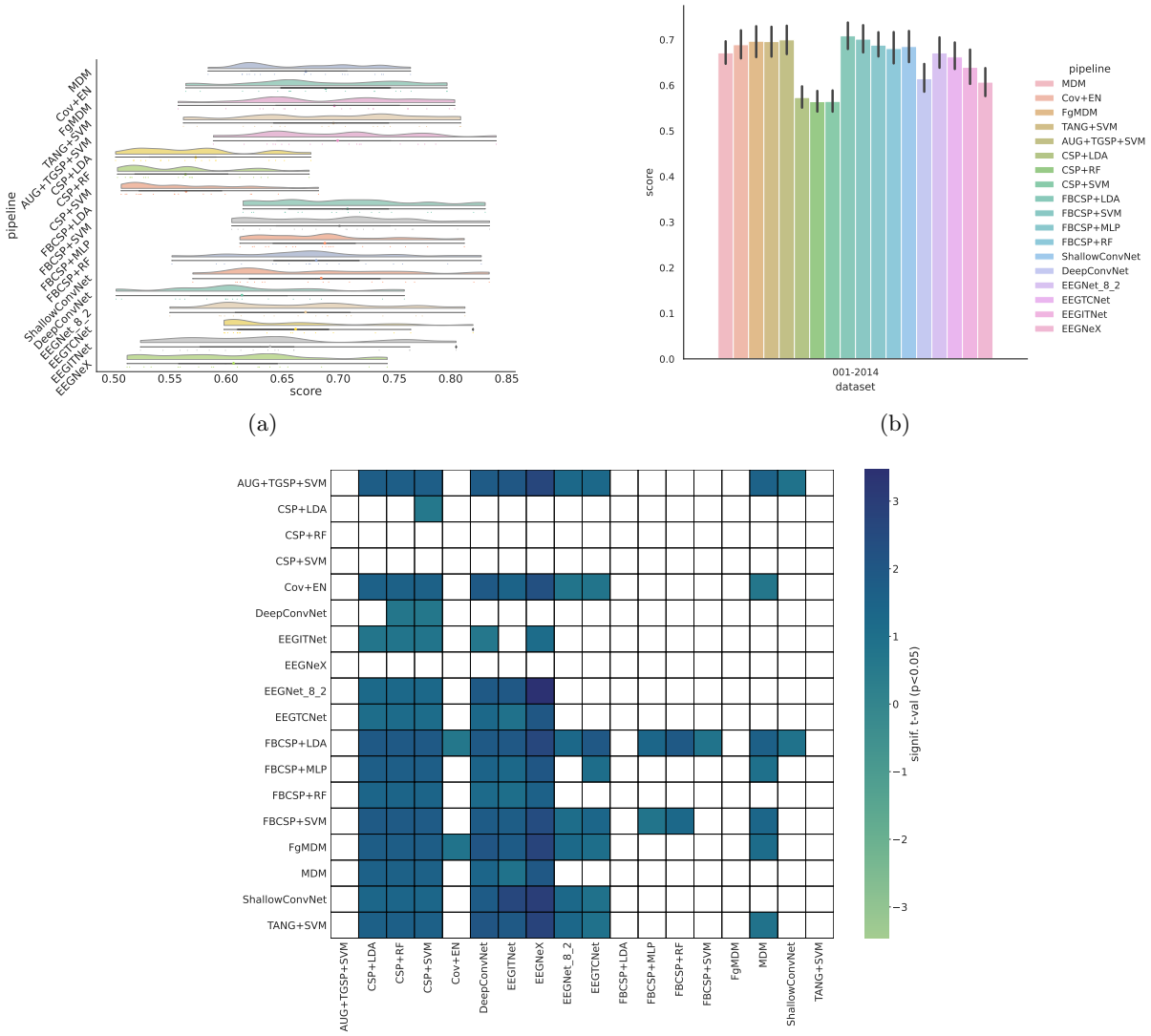


Figure 8: Result for BNCI2014001 classification, using Within-Session evaluation. Plot (a) shows the rain clouds plots for each pipeline, showing the distribution of the score of every subject. Plot (b) shows a bar plot of the score with the error of the different pipelines and for every datasets considered. Plot (c) shows the meta analysis of the different methods considered. This plots the significance that the algorithm on the y-axis is better than the one on the x-axis. The color represents the significance level of the difference of accuracy, in terms of t-values, and we show only the significant interactions ($p < 0.05$).

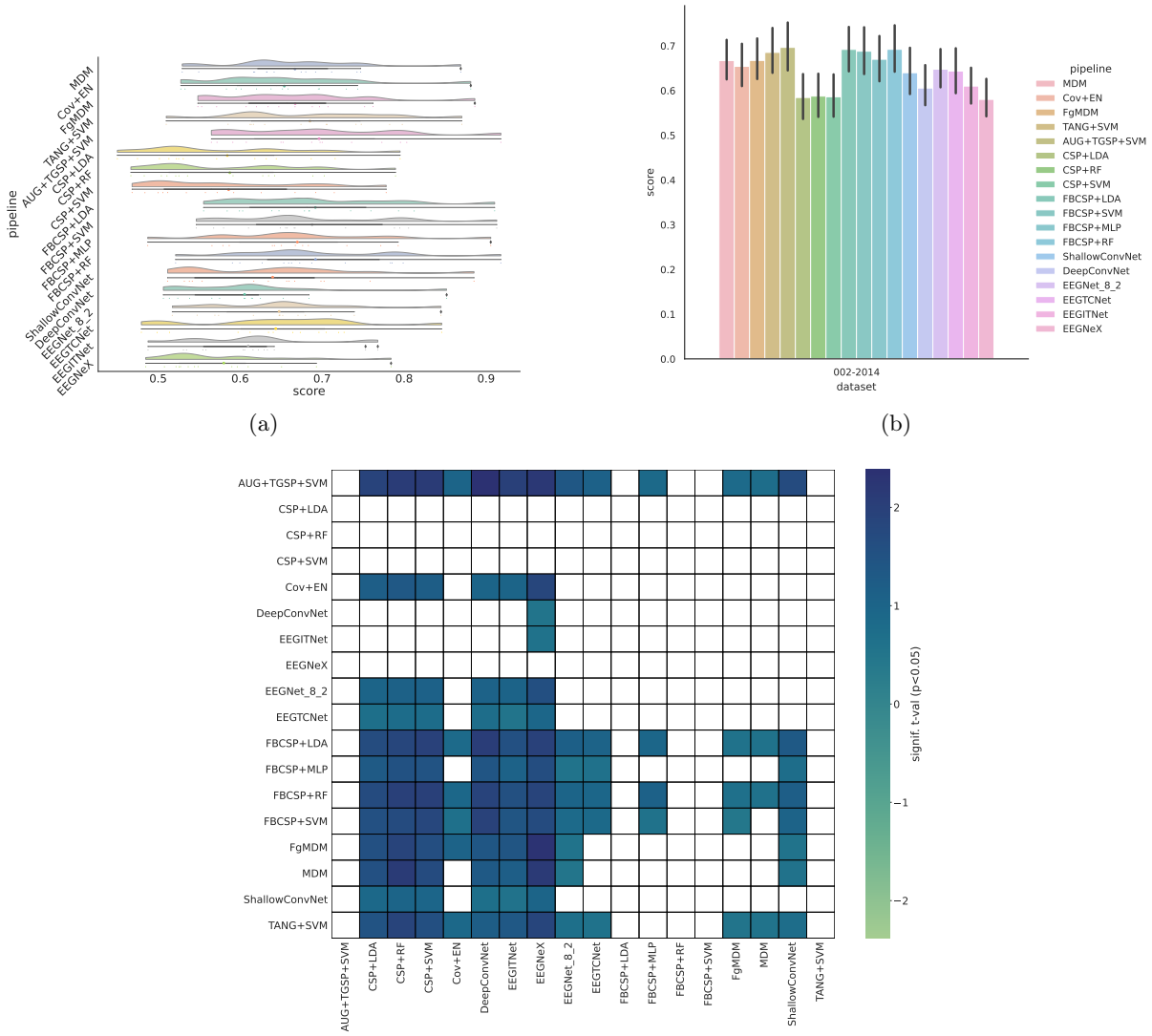


Figure 9: Result for BNCI2014002 classification, using Within-Session evaluation. Plot (a) shows the rain clouds plots for each pipeline, showing the distribution of the score of every subject. Plot (b) shows a bar plot of the score with the error of the different pipelines and for every datasets considered. Plot (c) shows the meta analysis of the different methods considered. This plots the significance that the algorithm on the y-axis is better than the one on the x-axis. The color represents the significance level of the difference of accuracy, in terms of t-values, and we show only the significant interactions ($p < 0.05$).

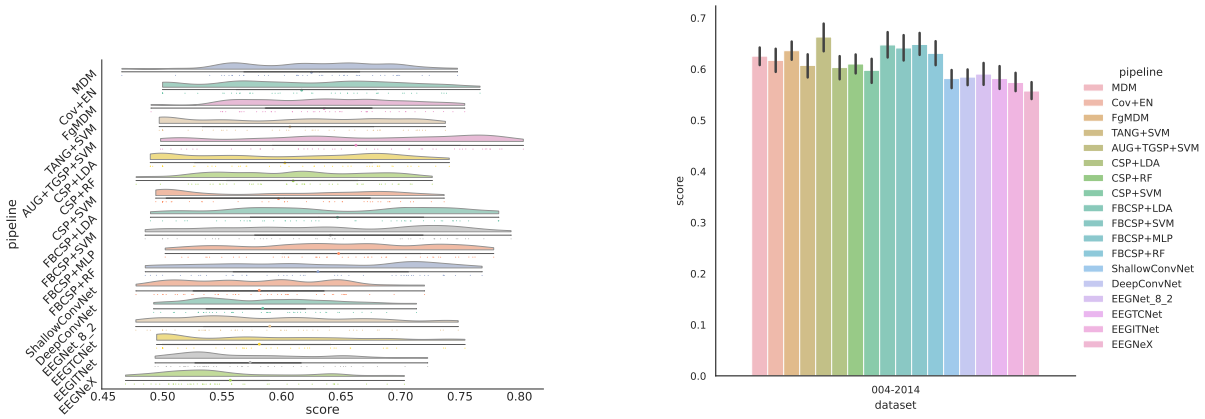


Figure 10: Result for BNCI2014004 classification, using Within-Session evaluation. Plot (a) shows the rain clouds plots for each pipeline, showing the distribution of the score of every subject. Plot (b) shows a bar plot of the score with the error of the different pipelines and for every datasets considered. Plot (c) shows the meta analysis of the different methods considered. This plots the significance that the algorithm on the y-axis is better than the one on the x-axis. The color represents the significance level of the difference of accuracy, in terms of t-values, and we show only the significant interactions ($p < 0.05$).

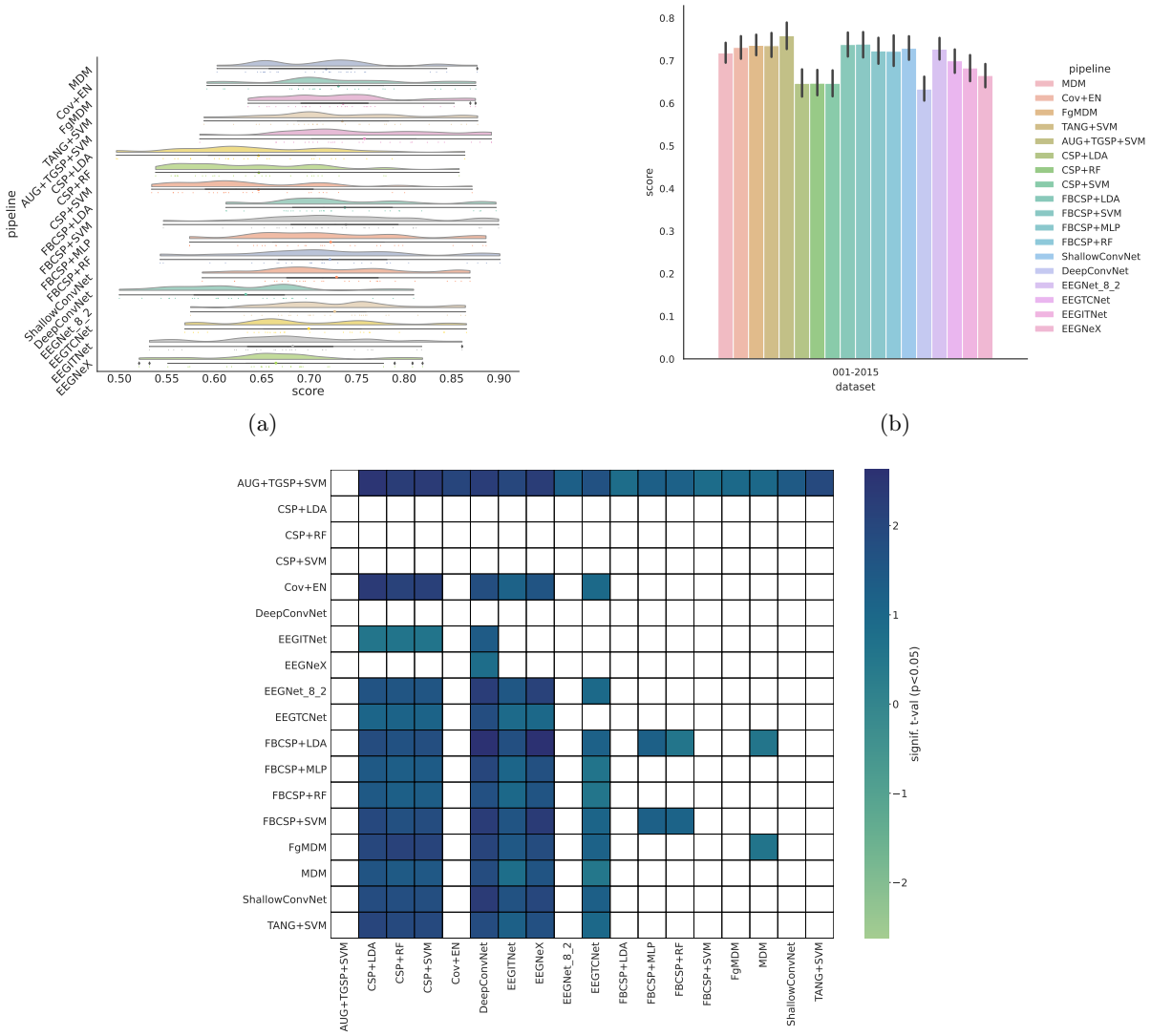


Figure 11: Result for BNCI2015001 classification, using Within-Session evaluation. Plot (a) shows the rain clouds plots for each pipeline, showing the distribution of the score of every subject. Plot (b) shows a bar plot of the score with the error of the different pipelines and for every datasets considered. Plot (c) shows the meta analysis of the different methods considered. This plots the significance that the algorithm on the y-axis is better than the one on the x-axis. The color represents the significance level of the difference of accuracy, in terms of t-values, and we show only the significant interactions ($p < 0.05$).

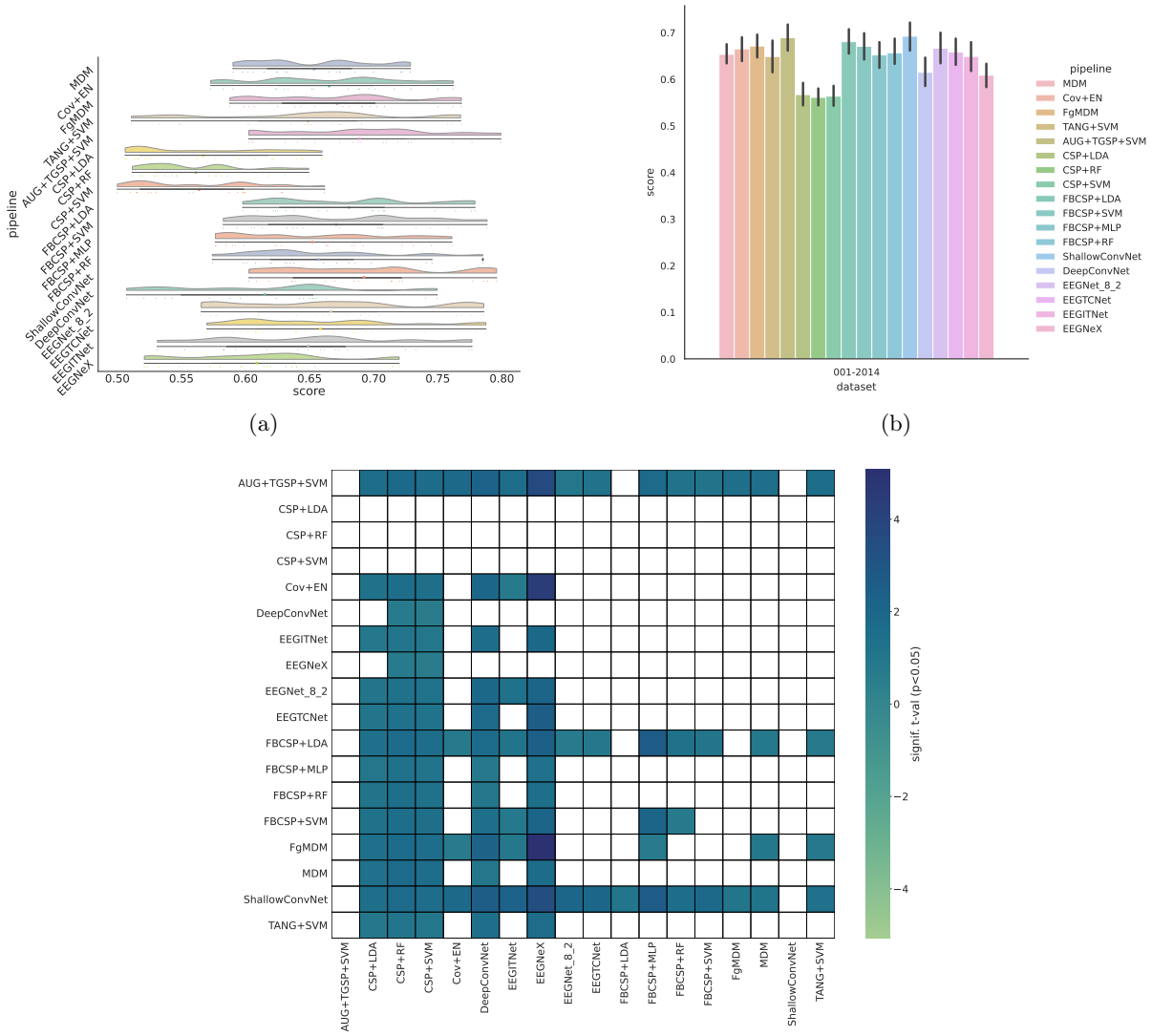


Figure 12: Result for BNCI2014001 classification, using Cross-Session evaluation. Plot (a) shows the rain clouds plots for each pipeline, showing the distribution of the score of every subject. Plot (b) shows a bar plot of the score with the error of the different pipelines and for every datasets considered. Plot (c) shows the meta analysis of the different methods considered. This plots the significance that the algorithm on the y-axis is better than the one on the x-axis. The color represents the significance level of the difference of accuracy, in terms of t-values, and we show only the significant interactions ($p < 0.05$).

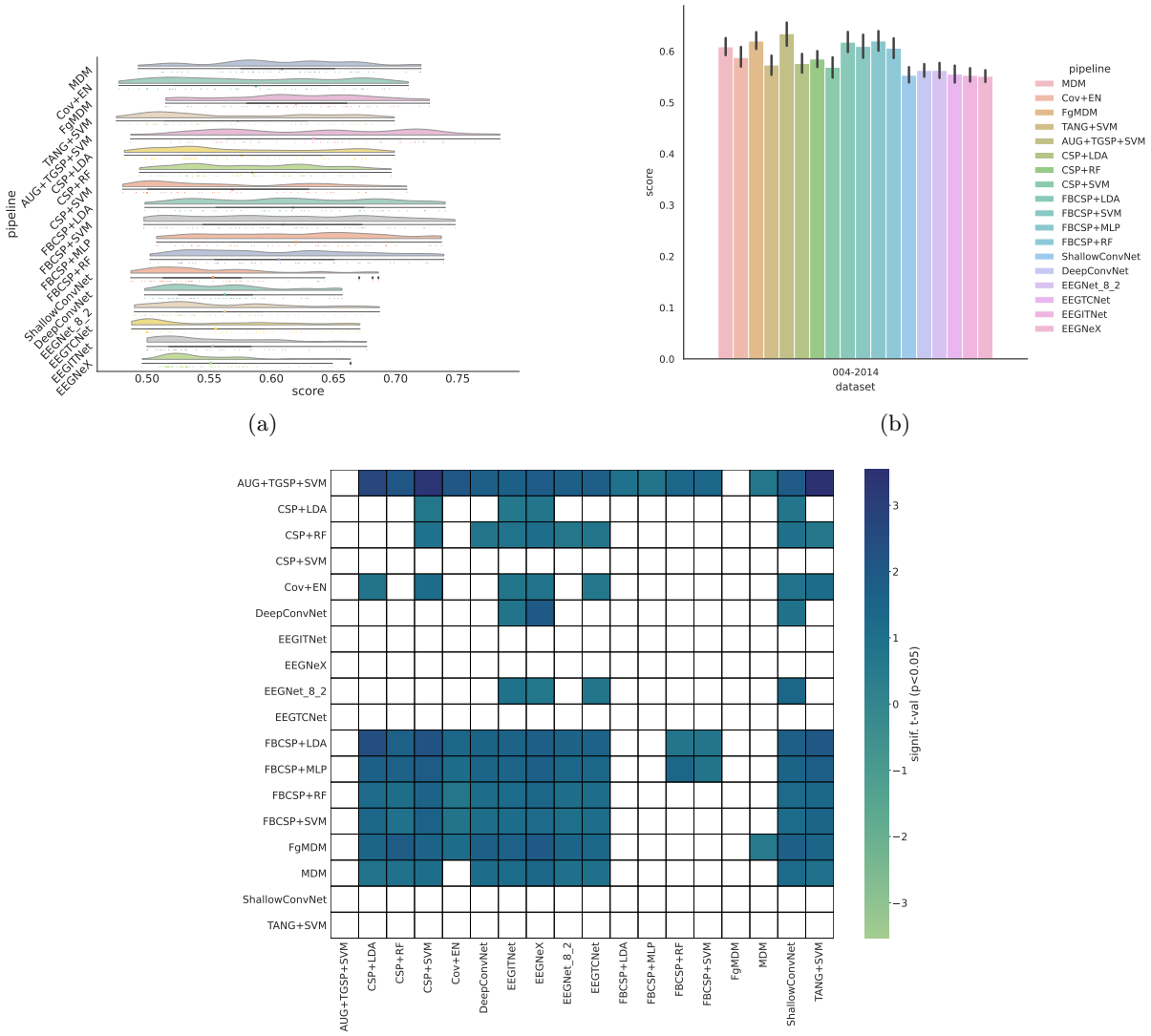


Figure 13: Result for BNCI2014004 classification, using Cross-Session evaluation. Plot (a) shows the rain clouds plots for each pipeline, showing the distribution of the score of every subject. Plot (b) shows a bar plot of the score with the error of the different pipelines and for every datasets considered. Plot (c) shows the meta analysis of the different methods considered. This plots the significance that the algorithm on the y-axis is better than the one on the x-axis. The color represents the significance level of the difference of accuracy, in terms of t-values, and we show only the significant interactions ($p < 0.05$).

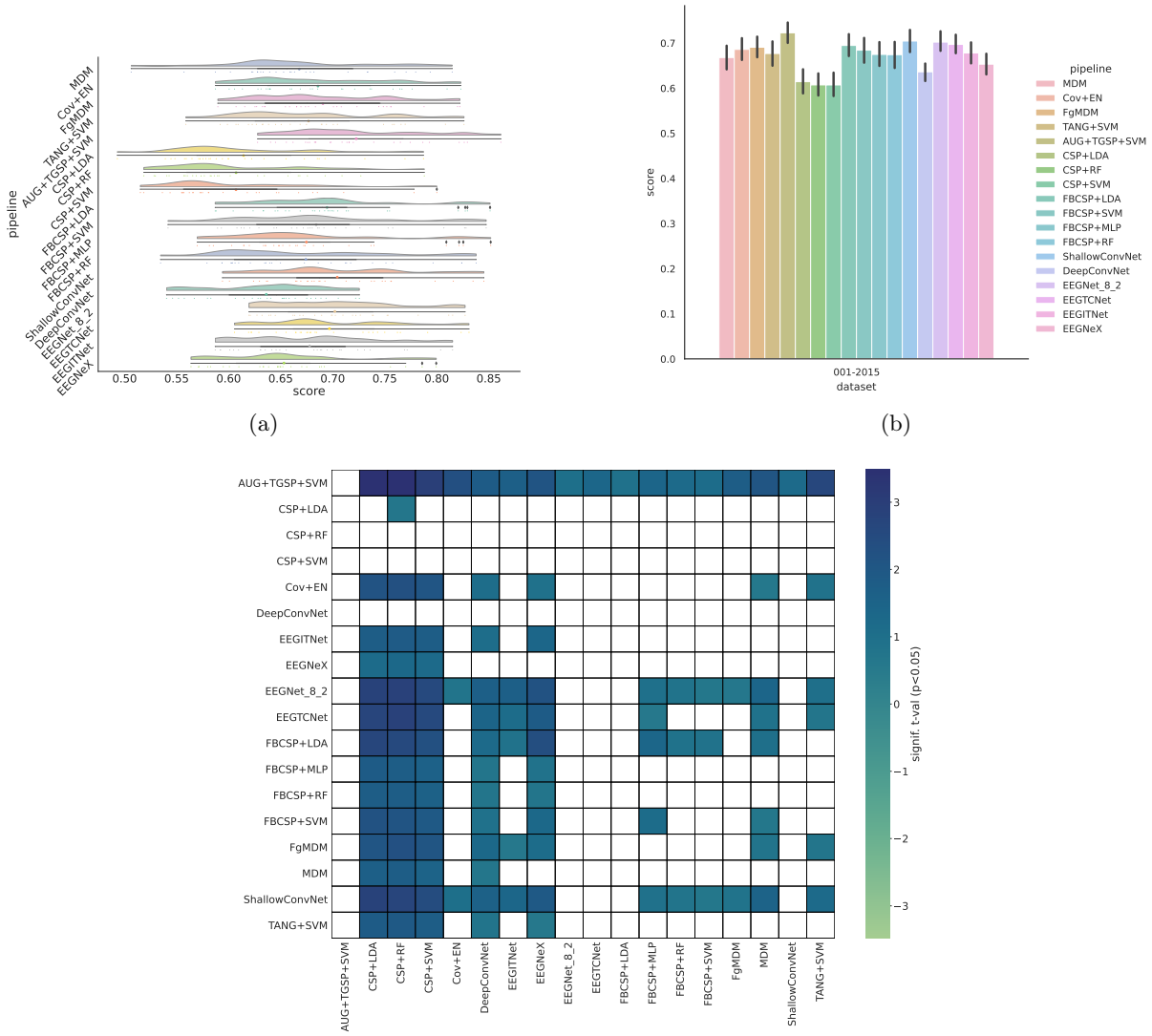


Figure 14: Result for BNCI2015001 classification, using Cross-Session evaluation. Plot (a) shows the rain clouds plots for each pipeline, showing the distribution of the score of every subject. Plot (b) shows a bar plot of the score with the error of the different pipelines and for every datasets considered. Plot (c) shows the meta analysis of the different methods considered. This plots the significance that the algorithm on the y-axis is better than the one on the x-axis. The color represents the significance level of the difference of accuracy, in terms of t-values, and we show only the significant interactions ($p < 0.05$).

Table 6: Performance Offline Within-Session Evaluation using the package MOABB changing the scoring to use nMCC. Results for the DL architecture are listed after the two line.

Pipeline	BNCI2014002	BNCI2014004	BNCI2015001	BNCI2014001
MDM	0.73 ± 0.15	0.73 ± 0.14	0.81 ± 0.14	0.81 ± 0.11
Cov + EN	0.82 ± 0.12	0.75 ± 0.14	0.86 ± 0.10	0.83 ± 0.10
FgMDM	0.81 ± 0.11	0.74 ± 0.14	0.85 ± 0.11	0.81 ± 0.11
TANG + SVM	0.81 ± 0.12	0.75 ± 0.14	0.85 ± 0.11	0.82 ± 0.10
AUG + TANG + SVM	0.84 ± 0.11	**0.81 ± 0.12**	0.90 ± 0.08	**0.86 ± 0.09**
CSP + LDA	0.80 ± 0.13	0.74 ± 0.14	0.84 ± 0.11	0.79 ± 0.10
CSP + RF	0.79 ± 0.12	0.72 ± 0.14	0.83 ± 0.11	0.78 ± 0.10
CSP + SVM	0.80 ± 0.13	0.75 ± 0.15	0.84 ± 0.10	0.80 ± 0.11
FBCSP+LDA	0.81 ± 0.13	0.77 ± 0.14	0.88 ± 0.09	0.84 ± 0.09
FBCSP+SVM	0.82 ± 0.12	0.78 ± 0.13	0.88 ± 0.08	0.84 ± 0.09
FBCSP+MLP	0.81 ± 0.12	0.78 ± 0.14	0.87 ± 0.09	0.83 ± 0.09
FBCSP+RF	0.81 ± 0.13	0.75 ± 0.14	0.86 ± 0.09	0.82 ± 0.09
ShallowConvNet	**0.88 ± 0.12**	0.72 ± 0.18	**0.91 ± 0.11**	0.72 ± 0.17
DeepConvNet	0.87 ± 0.11	0.72 ± 0.19	0.88 ± 0.14	0.34 ± 0.08
EEGNet 8 2	0.85 ± 0.16	0.69 ± 0.20	0.90 ± 0.12	0.61 ± 0.21
EEG ITNet	0.70 ± 0.18	0.65 ± 0.15	0.71 ± 0.17	0.34 ± 0.05
EEG TCNet	0.73 ± 0.20	0.69 ± 0.20	0.76 ± 0.19	0.40 ± 0.14
EEGNeX 8 32	0.70 ± 0.21	0.67 ± 0.17	0.72 ± 0.20	0.45 ± 0.16

Table 7: Performance Offline Cross-Session Evaluation using the package MOABB changing the scoring to use nMCC. Results for the DL architecture are listed after the two line.

Pipeline	BNCI2014004	BNCI2015001	BNCI2014001
MDM	0.79 ± 0.14	0.87 ± 0.11	0.59 ± 0.14
Cov + EN	0.81 ± 0.14	0.90 ± 0.10	0.64 ± 0.12
FgMDM	0.80 ± 0.14	0.89 ± 0.10	0.63 ± 0.13
TANG + SVM	0.81 ± 0.14	0.90 ± 0.10	0.62 ± 0.13
AUG + TANG + SVM	**0.85 ± 0.14**	**0.94 ± 0.07**	**0.73 ± 0.13**
CSP + LDA	0.81 ± 0.14	0.89 ± 0.10	0.60 ± 0.14
CSP + RF	0.76 ± 0.15	0.86 ± 0.12	0.56 ± 0.13
CSP + SVM	0.81 ± 0.14	0.89 ± 0.10	0.61 ± 0.13
FBCSP+LDA	0.82 ± 0.14	0.91 ± 0.08	0.66 ± 0.13
FBCSP+SVM	0.83 ± 0.14	0.91 ± 0.08	0.66 ± 0.13
FBCSP+MLP	0.82 ± 0.14	0.92 ± 0.08	0.65 ± 0.12
FBCSP+RF	0.80 ± 0.15	0.90 ± 0.09	0.63 ± 0.11
ShallowConvNet	0.73 ± 0.19	0.92 ± 0.10	0.70 ± 0.16
DeepConvNet	0.75 ± 0.17	0.90 ± 0.11	0.37 ± 0.10
EEGNet 8 2	0.75 ± 0.16	0.90 ± 0.12	0.59 ± 0.19
EEG ITNet	0.71 ± 0.15	0.79 ± 0.15	0.43 ± 0.16
EEG TCNet	0.74 ± 0.20	0.84 ± 0.17	0.44 ± 0.14
EEGNeX 8 32	0.71 ± 0.16	0.76 ± 0.19	0.46 ± 0.16

Table 8: Performance Pseudo Online Cross-Session Evaluation. Results for the DL architecture are listed after the two line.

Pipeline	BNCI2014004	BNCI2015001	BNCI2014001
MDM	0.61 ± 0.06	0.67 ± 0.07	0.65 ± 0.05
Cov + EN	0.59 ± 0.07	0.69 ± 0.07	0.67 ± 0.06
FgMDM	0.62 ± 0.06	0.69 ± 0.06	0.67 ± 0.06
TANG + SVM	0.58 ± 0.07	0.68 ± 0.07	0.66 ± 0.06
AUG + TANG + SVM	**0.64 ± 0.08**	**0.73 ± 0.06**	**0.70 ± 0.06**
CSP + LDA	0.58 ± 0.07	0.62 ± 0.07	0.57 ± 0.05
CSP + RF	0.58 ± 0.06	0.61 ± 0.07	0.57 ± 0.04
CSP + SVM	0.60 ± 0.07	0.61 ± 0.07	0.57 ± 0.05
FBCSP+LDA	0.62 ± 0.07	0.70 ± 0.07	0.68 ± 0.06
FBCSP+SVM	0.61 ± 0.07	0.70 ± 0.08	0.68 ± 0.05
FBCSP+MLP	0.63 ± 0.07	0.69 ± 0.08	0.68 ± 0.06
FBCSP+RF	0.61 ± 0.07	0.68 ± 0.08	0.66 ± 0.06
ShallowConvNet	0.55 ± 0.05	0.71 ± 0.07	0.69 ± 0.06
DeepConvNet	0.56 ± 0.05	0.64 ± 0.06	0.62 ± 0.07
EEGNet 8 2	0.56 ± 0.06	0.70 ± 0.07	0.67 ± 0.08
EEG ITNet	0.55 ± 0.05	0.68 ± 0.07	0.64 ± 0.07
EEG TCNet	0.55 ± 0.06	0.70 ± 0.07	0.66 ± 0.06
EEGNeX 8 32	0.56 ± 0.05	0.66 ± 0.07	0.61 ± 0.06