university of groningen / faculty of science and engineering / artificial intelligence

We All Make Mistakes: Classification of Error-Related Potentials In A Virtual Reality Environment For Brain-Computer Interfaces

Bachelor's Project Thesis

Niclas Brand, s3563944, n.c.brand@student.rug.nl,
Supervisors: Dr A.I. Sburlea & Prof Dr S. Enriquez Geppert

Abstract: Error-related potentials (ErrPs) are specific brain activity patterns occurring when a person experiences an error while interacting with their environment. These patterns can be observed by electroencephalographic (EEG) signals in a variety of tasks and feedback types. Virtual Reality technology allows for such multi-modal sensory feedback in an interactive environment. In this paper, two error types (Tracking and Feedback errors) of two participant groups (Attention Deficit Hyperactivity Disorder (ADHD) and the neurotypical control) will be classified by means of various machine learning classifiers. Data from 20 participants were collected, aggregated and analysed. The study investigates to what extent the classifiers can effectively discriminate between different Error-related potentials in EEG data when participants are exposed to simulated errors. After training, classification was possible across all tested classifiers and performed significantly better than a dummy classifier. These results further support the potential design of self-corrective applications by directly exploiting brain information.

1 Introduction

Humans are fallible. But this does not always need to be a disadvantage. Error-related Potentials (ErrPs) are neurophysiological signals associated with error processing (Pires et al., 2022). With the help of Machine Learning (ML) algorithms, we can use these brain signals to our advantage, for example in Brain-Computer Interface (BCI) applications (Kübler, 2020). BCIs are tools to directly communicate between the brain and an external, technical device and can be used in a variety of contexts.

ErrPs can be studied through Electroencephalography (EEG) — a method to record the electrical activity of the brain. They are linked to their subpart, Error-related negativity (ERN), which has been studied in neuroscience for a few decades (Holroyd & Coles, 2002), whereas ErrPs in conjunction with BCIs have only been investigated recently (e.g., Chavarriaga et al., 2014). Although, more and more task conditions are being explored (e.g., P. W. Ferrez & del R. Millan, 2008; Omedes et al., 2015, Spüler & Niethammer, 2015, Dias et al., 2018), the populations in combination with realistic tasks have been somewhat neglected. One contribution of this study is to investigate the existence of ErrPs and ERN in a population with Attention Deficit Hyperactivity Disorder (ADHD) tendencies.

The previously mentioned BCIs have been explored on therapeutic, supplementary but also on recreational levels. The devices range from neuroprosthetics to replace lost limbs (Pfurtscheller et al., 2008; Müller-Putz et al., 2022) over more exploratory, scientific applications such as reconstructing dreams and thoughts of people (Nishimoto et al., 2011; Chen, Qing, Xiang, et al., 2023; Chen, Qing, & Zhou, 2023), enhancing cognitive abilities (Zander & Kothe, 2011) to different and faster input methods for info- and entertainment purposes (Scherer et al., 2011; Marshall et al., 2013, for a comprehensive overview of the range of applications see Brunner et al., 2015). Combining both, ErrPs and BCI applications, allows us to improve BCI accuracy and reliability in general (Ahkami & Ghassemi, 2021), create intuitive, and self-corrective applications (Freudenburg et al., 2021) that also adapt to the individual user over time (Zander et al., 2016). For example, Freudenburg et al. (2021) initiated resets in problematic situations in a BCI-controlled prosthetic arm for patients suffering from amyotrophic lateral sclerosis (ALS) before further escalation. These resets would be based on ErrP detection in the patient's brain signals.

Virtual Reality (VR) environments enable the facilitation of experiments which would normally be hard or impossible to create in reality while also allowing for multi-modal sensory feedback. This broadens the range for ErrP detection (e.g.: visuo-haptic conflicts in Gehrke et al., 2019). Hence, a VR environment was chosen. The paradigm is related to the implementation of Si-Mohammed et al. (2020).

Literature shows that VR environments increase the sense of agency or immersion (Jeunet et al., 2018). The authors investigate a theory that focuses on the feeling and judgement of agency and apply them to virtual environments. The participants had to perform simple hand movements such as counting and tapping while simulation manipulations were performed. For example, a lag of 1, 1.5 and 2s was introduced to interfere with the principle of conscious intention before performance. They concluded that their modulations significantly interfered with the sense of agency within VR. Following this paper, a VR environment with high immersion and an intrusive experimental condition, the "Tracking Error" were constructed. As mentioned, Si-Mohammed et al. (2020) created a simple pick-and-place task in their VR setting which served as the foundation of this study's experimental paradigm. In the present study, however, some adjustments to the environment and task have been made. The environment was altered to increase immersion while the task has been modified to keep participants' suspicions low and attention high. Specifically, the task has been changed from a simple pick-and-place task to one that resembles the Wisconsin Card Sorting Task (Grant & Berg, 1993). This test is used to measure frontal lobe dysfunction for people with a variety of brain injuries, neurodegenerative or mental disorders. For the purpose of this study, however, its main purpose was to keep the participants attentive to the task to elicit valid ErrPs (Falkenstein et al., 2000).

To make use of the ErrPs in BCI applications in the manner described above, some form of automatic classification is paramount. Advancements in ML have given researchers the tools necessary for this challenge. Some machine learning classifiers make use of statistical properties of data (Fisher, 1936) to reliably discriminate data into a number of predetermined classes. The linear classifiers make use of the values of linear combinations of the object's features to classify the object into the classes (Yuan et al., 2012). These classifiers reach comparable accuracy levels to non-linear classifiers which are more flexible and can usually adjust better to the data at hand. The great advantage of linear classifiers is that they take a lot less training time. Thus, another contribution of this study is to determine whether Linear Discriminant Analysis (LDA) among Random Forest (RF) and AdaBoost (AB) qualify as valid classification methods for ErrP types.

Combining the present literature and goals of the study, the research question of interest is **to what extent can LDA, RF and AB accurately discriminate between different ErrPs in EEG data when participants are exposed to simulated errors**. Previous findings of related and similar approaches (Lopes-Dias et al., 2021) have shown great promise for this technique. Relying on their conclusions it is hypothesised that **LDA, RF and AB will demonstrate a statistically significant ability to accurately discriminate between different error-related potentials in EEG data and can predict error types robustly**.

To test this research question, a data set is collected where participants perform a simple sorting task that has resemblances with the Wisconsin Card Sorting task in a VR environment. After preprocessing of the data, LDA, RF and AB classifiers are trained and tested. Multiple classifiers are chosen to gain a more complete insight. Balanced accuracies are reported. Additionally, other statistical metrics such as Precision and Recall, F1-scores as well as *p*-values in comparison to a random uniform classifier are evaluated. The chosen approach for the statistical analysis is inspired by Dias et al. (2018). The authors also created a linear classifier distinguishing between correct and error trials in EEG data which coincided with this study's analyses. Albeit binary classification in their case, the approach still largely corresponds.

In the following sections details on the methodology, experimental setup and procedure as well as the trained classifiers will be given. After the description, the grand average brain waves, classifier performances, and results of the statistical analysis will be presented which will be built upon in the discussion of the study's insights, limitations and implications. These will be discussed in light of real-world applications, as well as opportunities for future research.

2 Methods

2.1 Participants

Twenty-one persons participated in the experiment. The age of the participants ranged from 18 to 28 years old with a mean age of 22 years ($SD =$ 3.5). The majority of the participants were first-year Psychology students selected through the University of Groningen's internal recruitment portal (SONA). Other students became aware of our study through flyers in student messaging groups. The remaining participants were contacts of the researchers and fall under availability sampling. The inclusion criteria were either particularly high or low ADHD tendency scores from a previous questionnaire round. These groups also make up the intervention and control group of another study which makes use of the same data set, but will be treated as demographic data in the study at hand. As our feedback was usually displayed in the colours green and red, participants with colour blindness were accommodated by changing the feedback colours to cyan and yellow for positive and negative feedback, respectively.

2.2 Ethical approval

The study has been approved by the ethics committee of the Faculty of Behavioural and Social Sciences of the University of Groningen (Ethical request number PSY-2223-S-0104). All participants read and signed informed consent before the start of the experiment and could refuse their participation at any time.

2.3 Materials

2.3.1 Hardware and EEG setup

For displaying the environment in VR, a computer (see Appendix A for the hardware specifications) was connected to the *Oculus Rift* (Meta Platforms, Menlo Park, United States of America) which is a commercial virtual reality device available to the general public. EEG signals were recorded with a sampling rate of 500 Hz. The *BrainAmp* amplifier as well as the *ActiCap Snap* cap (Brain Products, Munich, Germany) with 20 active channels were used. The specific layout can be found in Appendix A. The reference electrode was placed on the right mastoid, whereas the ground electrode was placed on the left mastoid. Due to the VR headset interfering, no explicit electrooculogram (EOG) electrodes could be placed. Artefact removal was based on the foremost EEG channels (AF3, AFz, AF4). The VR headset was mounted on the head by tightening the straps at the side rather than on the top to avoid too much pressure on the front-central electrodes which is our area of interest.

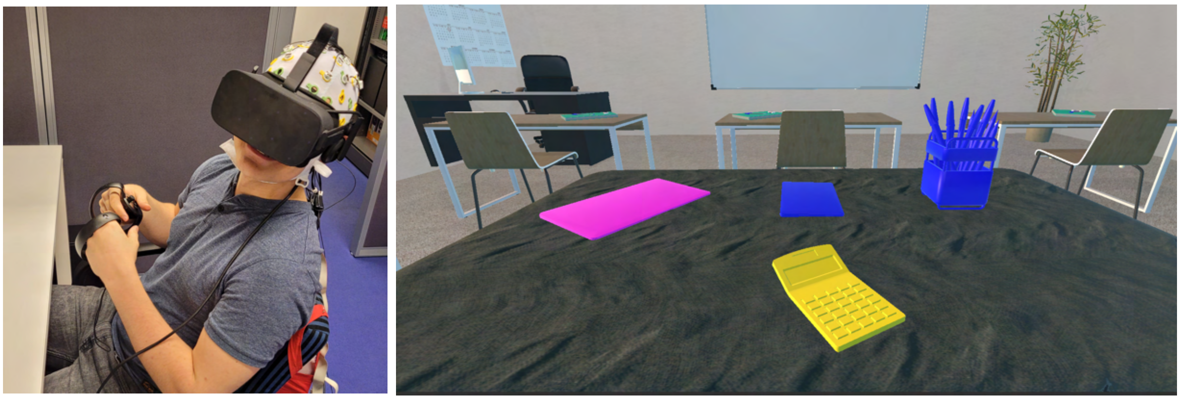
2.3.2 Software

For the complete experiment to run, a multitude of programs were used. The *Oculus software* (version 3.1.13) is the driver of the VR headset and is required to display and run applications in VR. The virtual environment has been created with the *Unity Game Engine* (version 2022.2.12f1). To run the virtual environment in a VR setting, *Steam VR* (version 1.25.8) was used as an interface. The EEG data recording is a delicate matter that requires multiple programs for proper collection. The first step is to run the *BrainVision recorder* (version 1.25.001) to set up the cap and measure the impedances of the specific electrodes. After proper setup, the amplifier was linked with the computer via *Liveamp* (version 1.23.5) and the live brain signal recording could be inspected using *BrainVision LSL viewer* (version 0.9.5). Finally, the LSL stream as well as the marker streams were recorded through Lab Recorder (version 1.16.3). The files were converted from XDF file format to FIF format and analysed with *Python* (version 3.11) scripts.

2.4 Experimental Protocol and Structure

The participants were welcomed into a simple lab room which was secluded from other people and provided no other distractions (see Appendix A). After filling out an ADHD tendency questionnaire*

*See Conners et al., 2012



(a) Participant wearing both an EEG cap and VR headset **(b) VR environment with targets at the sides and game object (here: a blue notebook) in the top middle**

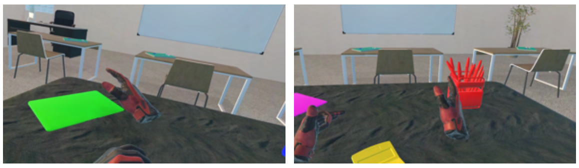
Figure 2.1: Experimental setup

for another study, they signed the informed consent. Then, the prepared EEG cap was put on and set up (Figure 2.1a). After mounting the VR glasses, the participant was sat close to the real table such that it coincided with the virtual one. The experiment followed a simple procedure in which the participant was guided through the experiment by means of a trial experiment. The recordings were started and the experiment commenced.

The virtual scene was inspired by Si-Mohammed et al. (2020) and consisted of a classroom setting with multiple tables, a whiteboard and a distinct table in front of the participant (Figure 2.1b). This virtual table was associated with a real table to provide the participant with passive haptic feedback for further immersion. Participants were sat at both, the virtual and real, table and were asked to perform a centre-out and place task. Specifically, the game object spawning in the top middle of the table was movable and could be picked up by means of the VR controllers. Then, the object had to be drawn onto one of the targets (i.e., a pink notebook to the top left, a blue pen holder to the top right or a yellow calculator to the bottom centre), either to the corresponding colour or shape. Each experiment run consisted of 400 iteration trials. To accommodate breaks for the participants, the experiment was split into 20 blocks with 20 iteration trials. Before running the experiment, a trial experiment with ten iterations was done to familiarise the participant with the technical equipment, the handling of the control devices, the virtual environment, the rules, the feedback and the information canvas indicating the end of blocks and break times. Within the experiment, 120 iteration trials (i.e.: 30%) were manipulated to be erroneous. Each iteration trial falls into one of the three experimental conditions. Sixty iterations showed a Tracking error (Te) and 60 iterations showed a Feedback error (Fe), leaving 280 iterations for unmodified trials (Ne). The errors were allocated semi-randomly through a sequence file. Some padding between errors was forced to avoid overly confusing the participants with too many back-to-back errors. The resulting data set was analysed offline after conducting the experiment.

2.5 Ruleswaps and Experimental Conditions

The experiment followed one game rule. The game object had to be either drawn into the target object corresponding to the same shape or colour. For example, a rule swap may change from shape-based targeting to a colour-based one. This game rule for targeting switched every four to six iterations without any cues to the participant. The change rate was based on previous literature (Monsell, 2003; Kiesel et al., 2010). The pilot experiment confirmed that this rate kept participants' attention without overwhelming or frustrating them. This uncued rule swap resembles that of the Wisconsin Card Sorting Test (Grant & Berg, 1993). The experiment consisted of three experimental conditions. The normal



(a) Correct feedback indicated by the target lighting up green (here: the notebook to the left)

(b) Wrong feedback indicated by the target lighting up red (here: the penholder to the right)

Figure 2.2: Different feedback types indicated by highlighting the targets visually after each iteration of the task

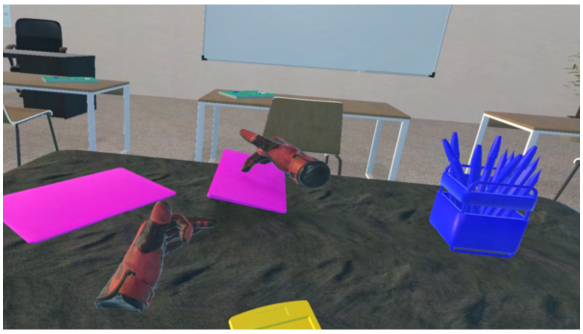


Figure 2.3: The Tracking Error preventing the game object from being dragged further

"No Error" condition, the "Tracking Error" and the "Feedback Error" condition.

The No Error condition (Ne) was the unmodified, normal trial procedure. Once the game object was grabbed and attached to the virtual hand, the participant would continue to draw the object onto a target object. Then, they were shown the correct feedback (see Figure 2.2a).

The Tracking Error condition (Te) corresponds to trials where the simulation lost track of the game object before it could be drawn onto a target object. At a random point between 15% and 65% of the initial game object position and the target destination, the game object froze and stuck in midair (Figure 2.3). The percentages were taken to allow the game object to fall onto the table while preventing it to reach a target in time. The object was detached from the virtual hand and after two seconds, a new trial started. This error condition tries to elicit the interaction ErrP – an error-related potential that is being triggered by an unexpected (and wrong) interaction with the environment.

The Feedback Error condition (Fe) corresponds to the trials where it initially proceeded without any alteration. After completing the task, however, erroneous feedback was given (Figure 2.2b): even when sorting correctly, the feedback turned red (or yellow for colourblind participants). This condition is aimed at the feedback ErrP – an ErrP linked to realising that the previous choice was incorrect through feedback.

2.6 Data preprocessing

Before data analysis, the collected data was preprocessed. The data of one participant was corrupted as a result of a computer crash during the data collection. It could not be opened. Furthermore, another participant's data remains relatively small after dismissing them due to another crash on the same day. This leaves the whole dataset with a little more than 19 complete data sets. The data was filtered between 1 and 40 Hz with a two-pass forward and reverse, zero-phase finite impulse response (FIR) bandpass filter. The lower and upper transition bandwidths were, both, 0.50 Hz with a length of 5000 samples. It uses the window method and is characterised by non-causality. Afterwards, Independent Component Analyses (ICAs) were applied to all participant data sets. The identified components were manually inspected and discarded if they matched common patterns of other artefacts. For example, eye blink artefacts were removed after checking the front electrode channels and matching their average evoked patterns. On average, around 15% of the samples were omitted this way. After the exclusion of the components, the individual data sets were saved. Inspection of ErrPs per individual or group (after pooling) was now possible. Then, the data was refiltered with a similar filter to obtain the frequency range of interest (1 to 10 Hz) for ErrPs (Lopes-Dias et al., 2019; P. Ferrez & Millán, 2007; Spüler & Niethammer, 2015). Finally, individual and group data analysis could be done.

2.7 Epoch creation

Albeit that some trials were short in duration, they were sufficiently long to create epochs to classify the three conditions. For the "Tracking Error" and "Feedback Error" conditions, 1300ms epochs were considered. They lasted from $300ms$ to $1000ms$ with respect to the error onset. The difference between these classifications is the actual error onset. The Te condition had its onset during the trial; once the object froze midair whereas the Ne and Fe conditions marked the errors once feedback was shown, i.e. at the end of a given trial. For the classification, all EEG channels were used whereas our area of interest ('Fz', 'FC1', 'FCz', 'FC2', and 'Cz') was averaged to show the grand average of the various conditions. These epochs will function as samples for the training of the classifiers.

2.8 Machine Learning Classifiers

2.8.1 Linear Discriminant Analysis

A model based on statistical classification was trained for the automatic categorisation of the trial types. Specifically, Linear Discriminant Analysis (LDA) is used as a foundation. LDA is based on Fisher's Criteria which maximises between-class mean distance and minimises within-class spread (Fisher, 1936). LDA assumes the same within-class variance which leads to linear cutoff thresholds. These cutoff thresholds are computed by linear combination of the feature vectors of the input. When the LDA classifier receives a real vector $\vec{x}$ as a feature input, then the output score is

$$y = f(\vec{w} \cdot \vec{x}) = f\left(\sum_j w_j x_j\right). \tag{2.1}$$

Here $\vec{w}$ is a vector of weights in the function f that is similar to a threshold function. If the output falls within a certain area with threshold boundaries, it will be classified as that specific class, but if it falls outside of it, the prediction changes (see Figure 2.4).

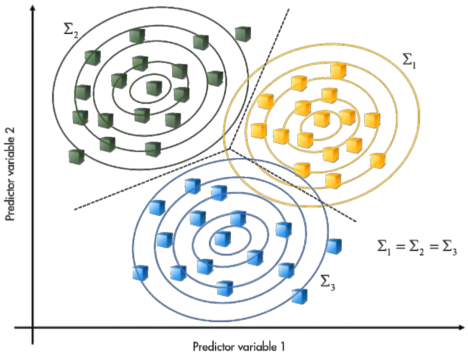


Figure 2.4: Visualisation of LDA with three classes to discriminate[†]

2.8.2 Random Forest

The next two classifiers can be used on the same format of data and were investigated because they use a different approach to classification. Their classification is based on decision trees and, in our case, classification trees. In those, the trees are constructed such that the final leaves of a tree represent class labels (i.e.: Ne/Te/Fe) and the branches a conjunction of data point features such as minima or maxima. These trees commonly overfit to the training data which is circumvented by the Random (Decision) Forest (RF) ensemble learning method. It constructs multiple decision trees and bases the final classification on the class predicted by the majority (see Figure 2.5).

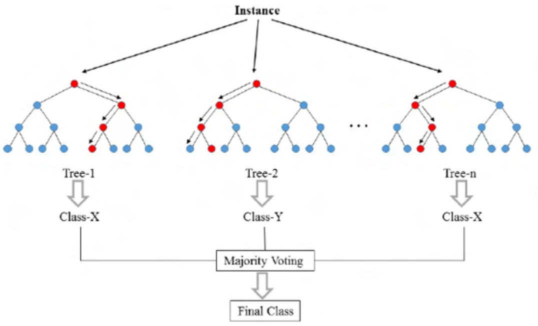


Figure 2.5: Visualisation of the Random Forest ensemble learning method[‡]**(e.g., class $X = Te$, class $Y = Fe$)**

2.8.3 AdaBoost

The AdaBoost (AB) method is short for "Adaptive Boosting" and increases the performances of "weak learners" by aggregation. These weak learners are models built with an imposed complexity limit, which perform slightly better than random

[†] Image taken from a video by MATLAB (Academy)

[‡] Image taken from Pawar, 2020

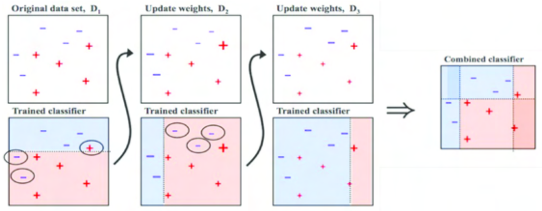


Figure 2.6: Visualisation of the adaptive tweaking of weak learners to combine to a strong learner[§]

guessing. The final output of this method is the weighted average of all the small base estimators. To preserve computing time, the base estimator type chosen was the Binary Tree Classifier with a depth of one. It is a simple classification tree with one branch. Fifty base estimators were considered by AdaBoost and the classifiers were tweaked whenever a class has been misclassified by a previous classifier. This adaptive property gives the name to the method. The final classifier is a combination of these tweaked weak learners. An exemplary adaptation can be seen in Figure 2.6.

The algorithms were implemented using the SciKit Learn Python library (Pedregosa et al., 2011) which allows for simple training once data is fed and formatted properly. The chosen implementation can be found on GitHub. For an inspection of the hyperparameters, consult Appendix A.

2.9 Metrics

Each classifier was validated with a ten-fold cross-validation. Evaluation of the classifier was done with a variety of metrics.

Accuracy represents how many correct predictions a model has produced across all test data.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.2}$$

where TP are true positives; correctly identified error trials and TN are true negatives; correctly identified no-error trials. FP are false positives; incorrectly identified error trials and FN are false negatives; incorrectly identified no-error trials. Unfortunately, this metric - along with precision and recall outlined below - is prone to unbalanced data sets and may be bloated by always selecting the most frequent class. Nonetheless, they are helpful metrics depending on the context but also constitute the larger F1 score which will be described later. Understanding both is paramount to grasping the reported F1 scores. Precision and Recall will also be reported separately.

Precision represents the number of true positives among the total number of "guessed" positives. In other words, it is a measure of how many positive guesses were correct.

$$Precision = \frac{TP}{TP + FP}. \tag{2.3}$$

Recall is, essentially, a measure of how many of the total number of true positives were successfully detected - this is because false negatives can be interpreted as true positives that were misidentified or missed.

$$Recall = \frac{TP}{TP + FN}. \tag{2.4}$$

Finally, the F1 scores are weighted combinations of precision and recall. F1 scores are a useful metric to show the performance of classification in one glance.

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}. \tag{2.5}$$

The metrics presented up to now will be calculated on the classifiers' test performances on a test data set and will solely be used for the construction of the confusion matrix. The test data set (30% of the data) was separated from the training data set (70% of the data) before the classifier training.

These common metrics are the foundation of the more appropriate weighted metrics. These alongside their standard deviation from the ten-fold cross-validation will be reported.

Balanced accuracy is accuracy weighted by class imbalance. In the default binary case, it is calculated as follows:

$$Balanced\ Accuracy = \frac{TPR + TNR}{2} \tag{2.6}$$

where TPR and TNR are computed by TP over all positives and TN over all negatives. Balanced Accuracy is a more appropriate measure when the data set has imbalanced classes while retaining the intuitive property.

[§] Image taken from Alto, 2020

The Receiver Operating characteristic (ROC) curve is a graphical plot that shows the diagnostic ability of a binary classifier. It plots TPR or $Recall$ against TNR and, essentially, shows intuitively how well the classifier balances them. The corresponding Area Under the Curve (AUC) can be computed to wrap this information into a single number. As we are dealing with multilabel classification, some further adjustments are necessary. The one-vs-rest (OVR) algorithm computes the average of the ROC AUC scores for each class against all other classes. The grand average of all pairwise combinations is the final metric.

Balanced accuracy as well as the weighted variants and the ROC-AUC metrics are calculated based on the average scores resulting from the tenfold cross-validation.

To get an additional idea of how well the classifier performs, the mean balanced accuracy will be compared to a random classifier's mean accuracy by means of a paired t-test. The t statistics are accommodated by the corresponding p-values. The paired t-tests are computed with

$$t = \frac{\bar{x}_{\text{diff}}}{s_{diff}/\sqrt{n}} \tag{2.7}$$

where $\bar{x}_{\text{diff}}$ is the sample mean of the differences between the classifier's predictions and a general uniform dummy classifier which predicts all classes equally. The s_{diff} term denotes the sample standard deviation of the differences and n the number of pairs.

3 Results

In the following, the results of the EEG analysis as well as the classifier performances will be presented.

3.1 Electrophysiological analysis

The EEG signal of the grand average for one epoch of the "No Error", "Tracking Error" and "Feedback Error" conditions can be seen in Figures 3.1, 3.2 and 3.3, respectively. The different coloured lines indicate the signal behaviour across the different channels of interest; Fz, FC1, FCz, FC2, Cz[¶]. The Figure in the Appendix (A.3) shows the brain signals across all EEG channels.

3.1.1 No error condition

The "No Error" condition is not aligned with any event which is why no event-related potential is expected. The signal shows no extreme peaks and remains relatively flat (see Figure 3.1). The signal fluctuates between low amplitudes. It has a negative peak at $t = 0ms$ with a negative amplitude of $-1.15\mu V$ followed by a positive peak at $t = 268ms$ time with $6.07\mu V$ amplitude. The topoplot shows the scalp activity throughout the whole epoch from $t = -300ms$ to $t = 1000ms$ in intervals of $100ms$.

3.1.2 Tracking error condition

The signal behaviour of the "Tracking Error" condition can be seen in Figure 3.2. The tracking error is characterised by a negative peak at $t = 135ms$ with an amplitude of $-1.97\mu V$. This negative peak is followed by a positive peak at $t = 203ms$ of $0.70\mu V$. After another negative peak at $t = 256ms$ of $-1.73\mu V$, the signal peaks positively at $t = 334ms$ with an amplitude of $3.90\mu V$. The signal then levels off with a final negative peak at $t = 454ms$ with an amplitude of $1.70\mu V$ and positive peak at $t = 587ms$ with an amplitude of $3.31\mu V$. Again, the topoplot shows the scalp activity throughout the whole epoch from $t = -300ms$ to $t = 1000ms$ in intervals of $100ms$.

3.1.3 Feedback error condition

The "Feedback Error" condition behaves fairly similarly to the "No Error" condition (see Figure 3.3). A negative peak at $t = -51ms$ with an amplitude of $-1.24\mu V$ is followed by a positive peak at $t = 256ms$ with an amplitude of $6.31\mu V$. In contrast to the "No Error" signal, this "Feedback Error" signal shows one more pair of peaks. Namely, a negative peak at $t = 346ms$ with an amplitude of $3.26\mu V$ and a positive peak at $t = 415ms$ with an amplitude of $4.97\mu V$. Afterwards, the signal levels off. The topoplot shows the average epoch from $t = -300ms$ to $t = 1000ms$ in intervals of $100ms$.

[¶] in the figures: Fz=*lime*, FC1=*green*, FCz=*teal*, FC2=*pink*, Cz=*purple*

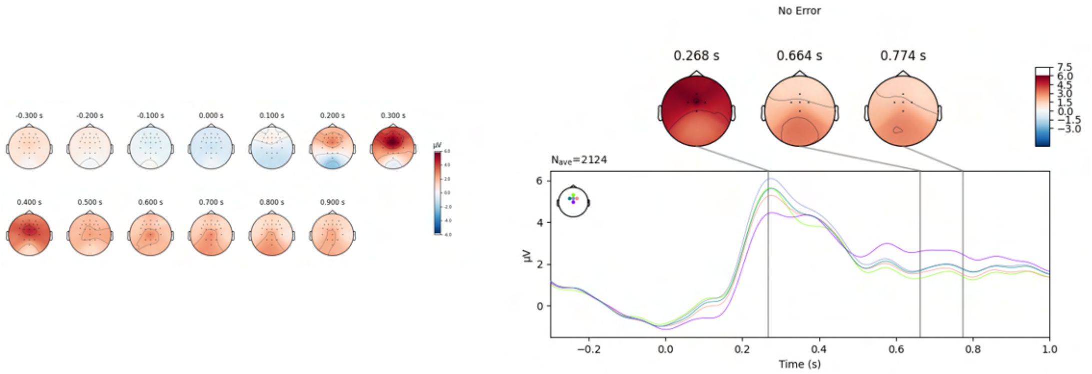


Figure 3.1: Grand average and topoplot of the "No Error" Condition

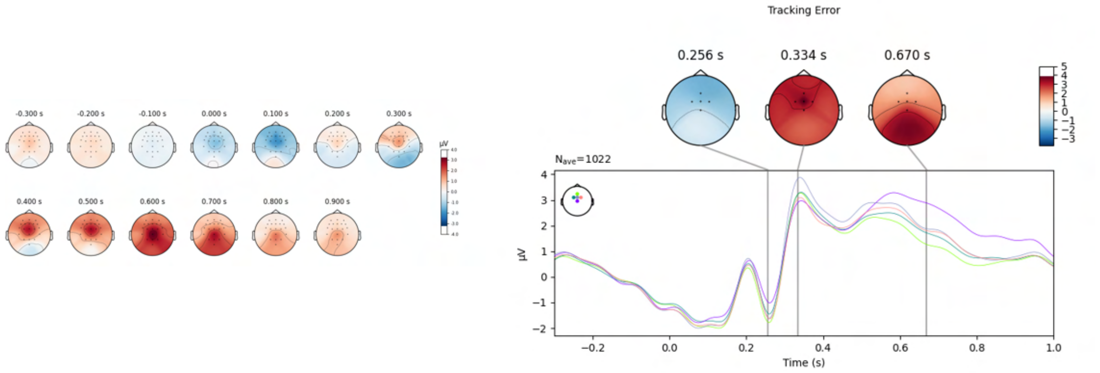


Figure 3.2: Grand average and topoplot of the "Tracking Error" Condition

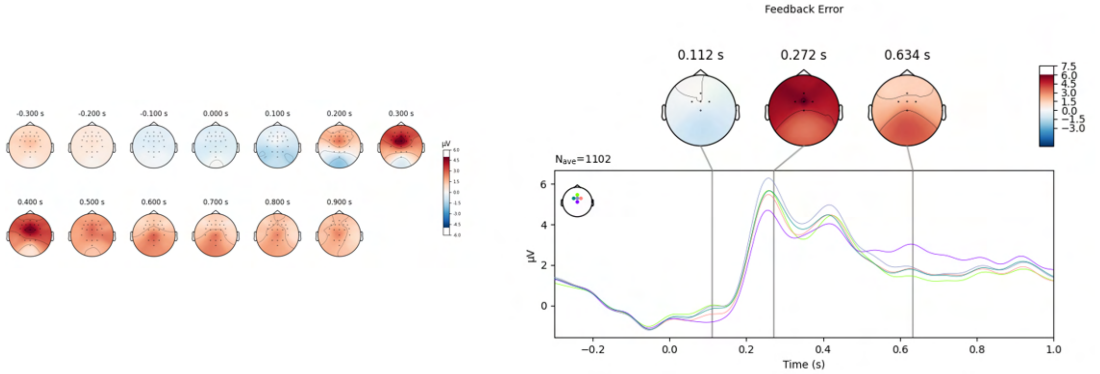


Figure 3.3: Grand average and topoplot of the "Feedback Error" Condition

3.2 Trial Classification

After initial training, the imbalance of the data set became apparent. Thus, all epochs were equalised to balance the samples in the data set ($N = 4892$) before another training commenced. Afterwards, the epochs count was such that the "No Error" condition constituted one half of the data set ($N_{Ne} = 2108$) and the error conditions the other ($N_{Te} = 1006, N_{Fe} = 1102$). This data split leaves out 676 invalid samples which could not be properly identified by the function and had multiple labels from the markerstream attached to them. Nonetheless, the balancing allows all the chosen metrics to be appropriate.

3.2.1 Performances and Significance Levels

All classifier performances can be inspected using Table 3.1. Additionally, the confusion matrices (Figure 3.4 for LDA, Figure 3.5 for RF, and Figure 3.6 for AdaBoost) of the various classifiers give more details of each method's behaviour which are outlined in the following. The confusion matrices are based on the classifier being trained with 70% of the complete data set. The remaining 30% were used as test data. By using the paired t-test and calculating the corresponding p-values, the statistical significance of each classifier's performance could be evaluated. The statistical results can be seen in Table 3.2.

The LDA classifier performed the best out of the tested classifiers. The average balanced accuracy was .38 ($SD = .06$) with an F1-score of .42 ($SD = .07$). The confusion matrix shows some tendencies to predict 'Ne' when uncertain. The LDA classifier yielded significantly better results than the uniform dummy classifier.

The RF classifier shows an average balanced accuracy of .34 ($SD = .06$) and an F1-score of .39 ($SD = .05$). The confusion matrix shows strong tendencies to predict 'Ne' when uncertain. Inspecting the feature importances using Mean Decrease in Impurity (see Figure A.4 in Appendix A) shows the important features according to mean decrease in impurity. The RF performed significantly better than the dummy classifier.

The AB classifier performs the worst out of the three with a balanced accuracy of .33 ($SD = .03$) and an F1-score of .37 ($SD = .03$). The confusion matrix shows more spread indicating that the classifier predicts 'Ne' less than the RF classifier, but still shows preferences for that condition. The low standard deviation is a perk that should not be neglected. AB also had significantly better results than the uniform dummy classifier.

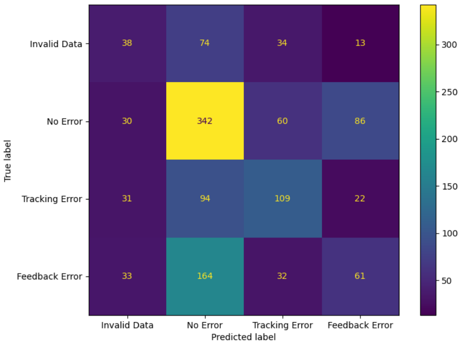


Figure 3.4: Classification performance of the LDA classifier on a balanced data set

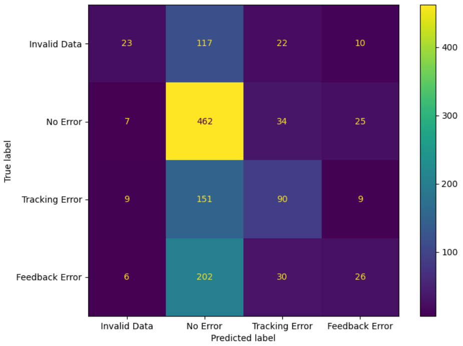


Figure 3.5: Classification performance of the Random Forest classifier on a balanced data set

These results allow for the rejection of the null hypothesis and provide evidence in favour of out alternative hypothesis; LDA, RF and AB demonstrate significantly better abilities to discriminate between different ErrPs in EEG data.

4 Discussion

The purpose of this study was to gain a better understanding of the potential automatic classifi-

Classifier	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC AUC**
LDA*	.38 ± .06	.42 ± .06	.44 ± .08	.42 ± .07	.65 ± .06
RF*	.34 ± .06	.42 ± .05	.47 ± .04	.39 ± .05	.64 ± .05
AB*	.33 ± .03	.39 ± .06	.43 ± .04	.37 ± .03	.62 ± .04
Dummy	.23 ± .02	.28 ± .02	.23 ± .02	.24 ± .02	.50 ± .00

Table 3.1: Cross-validated classifier performances ($M \pm SD$). All metrics are weighted by class imbalance. All classifiers significantly better than the dummy classifier are marked with an '*'

Classifier	t-value	p-value
LDA	5.27	$< .001$
RF	4.44	$< .001$
LDA	6.01	$< .001$

Table 3.2: Significance levels in comparison to the uniform dummy classifier

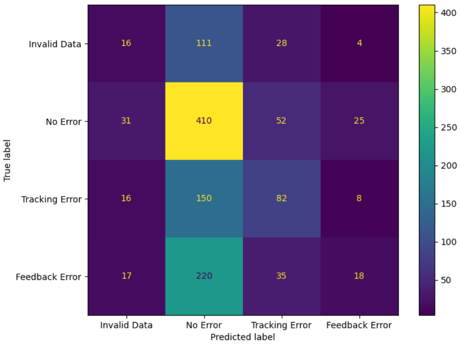


Figure 3.6: Classification performance of the AdaBoost classifier on a balanced data set

cation of error-related potentials. The results of the present study support the hypothesis that machine learning can do so successfully. There are two key findings of the present research. First, the results provide evidence that a multitude of different machine learning classifiers are able to distinguish reliably between the different types of ErrPs and experimental conditions. In the research at hand, these are characterised by continuous time series EEG data dealing with the brain activity patterns elicited by perceived errors in a virtual reality environment. Second, the tracking error condition seems to be more distinct than the feedback error condition. This pattern of results is consistent with some literature (see Si-Mohammed et al., 2020), but needs further investigation. Some potential reasons are discussed later. The study also revised the VR environment to be more immersive according to suggestions outlined in Jeunet et al. (2018).

4.1 Theoretical and Practical implications

The theoretical implications coincide partly with these key insights. ErrP classification is possible through various means, albeit LDA classification is by far the fastest to train and cross-validate with a training and testing time of about half an hour by exploiting shrinkage methods. In contrast, the Random Forest Classifier with an infinite number of features took around eight hours to finish the ten-fold cross-validation on the full data set and AdaBoost around one and a half hours. The cross-validation was done on higher-end consumer hardware[‖].

The LDA confusion matrix shows that the classifier cannot always discriminate between the "No Error" and "Feedback Error" conditions. Nonetheless, the results seem to be the best and most realistic out of the three tested classifiers.

Yielding moderate results, the RF classifier is slightly worse than LDA. However, RF can typically be used without much worries about overfitting due to the characteristics of ensemble methods. Random forests are also usually remarkably good out-of-the-box and a good initial approach. Another such out-of-the-box is the AdaBox which also showed mediocre results. From the confusion

[‖] AMD Ryzen 5 7600X

matrices of the Random Forest and AdaBoost classifiers, it becomes clear that they exhibit a common ML algorithm pitfall. Namely, the tendency to maximise performance by always predicting the most common class. It becomes apparent that the RF and AdaBoost classifiers fell victim to challenge and did not learn the features properly.

The study also built on previous studies (Si-Mohammed et al., 2020) and further confirmed that tracking and feedback errors are apparent in a similar VR environment and task. The errors persevered through the increase in error trials and remained distinct from the control 'Ne' condition. Another interesting insight with practical application is that ErrPs are also strongly present in neurodiverse populations. The high number of participants with ADHD tendencies did not impede our findings strongly while the results simultaneously indicate that reliable classification is also possible for these populations. This insight provides supporting evidence that promising applications of ErrP classification, such as resetting a robotic limb replacement, can also be used by people who have attention deficits. To overcome the presumed difficulties of scarce training data, transfer learning may be an interesting approach that shows potential. The concept will be described in Section 4.3 in more detail.

4.2 Potential limitations

Before taking these findings at face value, several limitations should be recognised.

One limitation that needs addressing is that the computer crashed multiple times towards the end of the data collection. Three participants' data sets had to be stitched together and only approximated the targeted 60 error trials. This is due to a random sequence file being generated at the beginning of the experiment. If the progress is interrupted and needs restarting, a new sequence file is generated and the error distribution might deviate from the targets. Nonetheless, these deviations are minor and were deemed to not affect the grand data set too much. Another consequence of the crashes was a data set being inaccessible**. The origin of the crashes could not be identified, however, the computer completely freezing without any black- or bluescreens may indicate a memory problem of the lab computer.

Another limitation of this study is that the Feedback error did not seem striking enough. It is possible that low engagement was prevalent in participants after a while or that the rule swaps may have been too frequent and made participants uncertain. Another explanation could be that the error could only be reliably determined with matching shape and colour while other erroneous feedback was ascribed to a rule swap during that trial. Hence, no explicit Feedback ErrP was elicited. Rather, only cognitive dissonance was experienced during these few completely matching trials and their feedback (Falkenstein et al., 2000). Although the present research cannot rule out these explanations, it seems useful to point out issues that may conflict with these results. This feedback ambiguity was not spotted during initial tests as the rule swap was one of the last parts to be implemented.

Despite the reported statistical significance, the practical significance remains questionable. Yielding AUC scores of about 65% and confusing matrices indicating predictions of the most frequent class do not seem to be sufficient for the practical implementation of the ErrP classifiers at hand. Some adjustments are necessary. To further improve and evaluate the performances, the data needs to be more thoroughly processed. In contrast to other literature (e.g., Si-Mohammed et al., 2020; Dias et al., 2018; Lopes-Dias et al., 2021), the "Tracking Error" signal remained relatively flat. During preprocessing and, specifically, during the ICA of the data, components were not removed conservatively. It is expected that too much brain activity was removed that way. Additional tweaks to the data set will most likely yield better results by making the class features more distinct from one another. Fine-tuning the classifier settings may also bring about improved results, but might make them overfit the data set. Hence, the performances may increase while overfitting remains controlled when preprocessing expertly and removing the correct ICA components.

It is noteworthy that the first two participants did not encounter the trial experiment, but were introduced to the task in the actual experiment. Hence, their first trials may differ from the other participants, but the impact on the grand data set

**The integer indicating the file size in bytes did not correspond to a valid length of 1, 4 or 8. Hence, no standard function could unravel the corrupted file.

should be minimal. Some complications[††] during the lab setup delayed initial data collection which cut the time for pilot studies short.

Finally, there were some known (unintended) bugs and errors in the simulation that may have made the experience less immersive than expected. After picking up the game object, dragging it into the target object (rather than dropping it onto it) made the next game object stuck to the hand once picked up until drawing it into the next target. Additionally, if the participant picked up a game object with both controllers simultaneously and dragged the object into a target, one hand was deactivated afterwards. These bugs could not be resolved in time before the data collection. The scripts are closely intertwined with the official SteamVR scripts and were difficult to search for errors. They were deemed minor enough to not postpone the data collection onset.

4.3 Future Research

As mentioned before, further research may shed light on some areas the study at hand fails to explore. It would be useful to extend the current findings by examining the performance of another predictive model. Namely, the autoregressive moving average (ARMA) approach (Box et al., 2015). It is a model specialised in dealing with time series data and is used to understand past values, but also predict future ones. ARMA is split into two parts. The autoregressive part (AR) regresses a variable on the time series' own past values. The moving average part (MA) models the error term as a linear combination of the error terms occurring at various times in the past. This statistical analysis can also be used to train a machine learning classifier by feeding, for example, 60% of an epoch into the model for training and using the remaining 40% for prediction. Then, the difference between the predicted and actual values is taken. With this difference, a predictability score can be computed. This approach also allows for the computation of interesting metrics such as confidence intervals which, in turn, could be used to further streamline the final application of a BCI by only committing to a reset if the confidence is high enough.

If neurodiverse populations show somewhat different ErrP signals, as the present study suggests, then there might be a need for research that explores the possibilities of efficiently training classifiers when data is scarce. The EEG signals did not greatly impede automatic classification in the paper at hand, however, studies suggest that there are subtle differences (Lopes-Dias et al., 2021) in various populations. At the same time, these studies highlight the promise of transfer learning (Iturrate et al., 2014; Dias et al., 2018). Transfer learning is the approach of reusing knowledge from a previous task to boost performance on a task related to the original, somewhat similar to the focus on weak learners in the AdaBoost algorithm. Transfer learning, however, describes the more general exploitation of knowledge from previous data and fully trained classifiers. For instance, an ErrP classifier could be trained on a data set such as ours and tested on an exclusively neurodivergent population. If insufficient performances are reached, the original classifier can be supplemented by the exclusive test data set and retrained. Then, the process is repeated with a new neurodivergent test sample. The process repeats until satisfactory results are achieved. This approach circumvents the training of a classifier on a large neurodivergent data set and relies on some previously learned weights. In theory, the current data set could be used for transfer learning as eight of the 20 participants showed ADHD tendencies and could be used as the neurodivergent sample. As the performances sufficed here for the study's objectives and the statistical analysis of the ErrP difference between the neurotypical and neurodivergent group was not finished at the time of writing (Nowicki, 2023, in writing), we continued without exploring the promised benefits of transfer learning.

The present study extended the research of Lopes-Dias et al. (2021) by investigating an additional neurodivergent population. Nonetheless, much more work remains to be done before a complete understanding of the extent of ErrP classification can be established. Other populations such as those with neurological dispositions (Autism/ASD, Fetal Alcohol Spectrum Disorder), motor disorders such as tic disorders (Tourette's) or developmental coordination disorder, as well as mental disorders (e.g., depressive, manic or psychotic people) need to be examined. It might also be of interest

[††] The first reference electrode broke before any use which made troubleshooting quite difficult.

to investigate older people, considering the demographic change and that older people being more prone to illness, especially, fall into the category of potential BCI users. To supplement the ErrP research even further, different tasks demand exploration. For example, text-based tasks in combination with dyslexic people, hearing-based tasks and people with hearing loss or people who have lost their haptic or optical sensory inputs. Once it has been confirmed that ErrPs are present in all these populations and tasks, tailored BCI implementation can be investigated (Volosyak et al., 2011).

5 Conclusion

Despite the mentioned limitations, the present study has enriched our understanding of the classification of error-related potentials. In summary, the research contributes to a growing body of evidence suggesting that ErrPs transcend the typical distinction between populations. Albeit the ErrPs slightly differ, this difference does not seem to create an obstacle that machine learning algorithms cannot surpass. ErrPs can be classified using machine learning methods regardless of the task and population involved. Although the results seem promising, the reported metrics need to be treated carefully. The reported weighted variants of precision, recall, f1-scores, and accuracies need to be evaluated on the background of the task. What metrics are useful depends on the actual goal the classifier is supposed to fulfil or even replace. For example, a classifier for automatic cancer detection might be interested in high recall rates which will be double-checked by medical professionals while automatic botany classification might want high precision after uncertainties arose among the experts. In the case of a BCI-controlled neuroprosthetic, precision seems to be more critical: resetting the robotic arm when something goes really wrong, rather than resetting too often even when no actual ErrP was processed seems to be appropriate to the general task of medical BCI applications. Nevertheless, these findings raise confidence in BCI-based rehabilitation in the healthcare sector. Yet, the generality of the current results needs to be confirmed and established. Further exploration of classification algorithms yielding high precision and transfer learning for practical feasibility remains necessary. Once we have a more coherent understanding of ErrPs, the amazing recovery options of mind and body through BCI applications (Lorach et al., 2023; Wagner et al., 2018) can be further accelerated. We hope that the promising results of the current research stimulate further investigation of this important area. Let us use our errors to our advantage.

References

Ahkami, B., & Ghassemi, F. (2021). Adding tactile feedback and changing ISI to improve BCI systems' robustness: An error-related potential study. *Brain topography*, *34*(4), 467–477.

Alto, V. (2020, Jan). *Understanding adaboost for decision tree.* Towards Data Science.

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control.* John Wiley & Sons.

Brunner, C., Birbaumer, N., Blankertz, B., Guger, C., Kübler, A., Mattia, D., ... Müller-Putz, G. R. (2015). BNCI Horizon 2020: towards a roadmap for the BCI community. *Brain-Computer Interfaces*, *2*(1), 1-10. doi: 10.1080/2326263X.2015.1008956

Chavarriaga, R., Sobolewski, A., & Millán, J. d. R. (2014). Errare machinale est: the use of error-related potentials in brain-machine interfaces. *Frontiers in Neuroscience*, *8*. doi: 10.3389/fnins.2014.00208

Chen, Z., Qing, J., Xiang, T., Yue, W. L., & Zhou, J. H. (2023). Seeing Beyond the Brain: Conditional Diffusion Model with Sparse Masked Modeling for Vision Decoding.

Chen, Z., Qing, J., & Zhou, J. H. (2023). Cinematic Mindscapes: High-quality Video Reconstruction from Brain Activity.

Conners, C. K., Erhardt, D., & Sparrow, E. (2012). Conners' adult ADHD rating scales. *PsycTESTS Dataset*.

Dias, C. L., Sburlea, A. I., & Müller-Putz, G. R. (2018, apr). Masked and unmasked error-related potentials during continuous control and feedback. *Journal of Neural Engineering*, *15*(3), 036031. doi: 10.1088/1741-2552/aab806

Falkenstein, M., Hoormann, J., Christ, S., & Hohnsbein, J. (2000). ERP components on reaction errors and their functional significance: a tutorial. *Biological Psychology*, *51*(2), 87-107. doi: https://doi.org/10.1016/S0301-0511(99)00031-9

Ferrez, P., & Millán, J. (2007). EEG-based brain-computer interaction: Improved accuracy by automatic single-trial error detection. *Advances in neural information processing systems*, *20*.

Ferrez, P. W., & del R. Millan, J. (2008). Error-Related EEG Potentials Generated During Simulated Brain–Computer Interaction. *IEEE Transactions on Biomedical Engineering*, *55*(3), 923-929. doi: 10.1109/TBME.2007.908083

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, *7*(2), 179–188.

Freudenburg, Z., Kohneshin, K., Aarnoutse, E., Vansteensel, M., Branco, M., Leinders, S., ... Ramsey, N. (2021, NOV). The dorsolateral prefrontal cortex bi-polar error-related potential in a locked-in patient implanted with a daily use brain-computer interface. *Control Theory and Technology*, *19*(4, SI), 444-454. doi: 10.1007/s11768-021-00062-y

Gehrke, L., Akman, S., Lopes, P., Chen, A., Singh, A. K., Chen, H.-T., ... Gramann, K. (2019). Detecting visuo-haptic mismatches in virtual reality using the prediction error negativity of event-related brain potentials. In *Proceedings of the 2019 chi conference on human factors in computing systems* (pp. 1–11).

Grant, D. A., & Berg, E. A. (1993). Wisconsin Card Sorting Test. *Journal of Experimental Psychology*.

Holroyd, C. B., & Coles, M. G. (2002). The neural basis of human error processing: reinforcement learning, dopamine, and the error-related negativity. *Psychological review*, *109*(4), 679.

Iturrate, I., Chavarriaga, R., Montesano, L., Minguez, J., & Millán, J. (2014, apr). Latency correction of event-related potentials between different experimental protocols. *Journal of Neural Engineering*, *11*(3), 036005. doi: 10.1088/1741-2560/11/3/036005

Jeunet, C., Albert, L., Argelaguet, F., & Lécuyer, A. (2018). "Do you feel in control?": Towards Novel Approaches to Characterise, Manipulate and Measure the Sense of Agency in Virtual Environments. *IEEE transactions on visualization and computer graphics*, *24*(4), 1486–1495.

Kiesel, A., Steinhauser, M., Wendt, M., Falkenstein, M., Jost, K., Philipp, A. M., & Koch, I. (2010). Control and interference in task switching—A review. *Psychological bulletin*, *136*(5), 849.

Kübler, A. (2020). The history of BCI: From a vision for the future to real support for personhood in people with locked-in syndrome. *Neuroethics*, *13*(2), 163–180.

Lopes-Dias, C., Sburlea, A. I., Breitegger, K., Wyss, D., Drescher, H., Wildburger, R., & Müller-Putz, G. R. (2021, mar). Online asynchronous detection of error-related potentials in participants with a spinal cord injury using a generic classifier. *Journal of Neural Engineering*, *18*(4), 046022. doi: 10.1088/1741-2552/abd1eb

Lopes-Dias, C., Sburlea, A. I., & Müller-Putz, G. R. (2019). Online asynchronous decoding of error-related potentials during the continuous control of a robot. *Scientific reports*, *9*(1), 17596.

Lorach, H., Galvez, A., Spagnolo, V., Martel, F., Karakas, S., Intering, N., ... others (2023). Walking naturally after spinal cord injury using a brain–spine interface. *Nature*, 1–8.

Marshall, D., Coyle, D., Wilson, S., & Callaghan, M. (2013). Games, Gameplay, and BCI: The State of the Art. *IEEE Transactions on Computational Intelligence and AI in Games*, *5*(2), 82-99. doi: 10.1109/TCIAIG.2013.2263555

Monsell, S. (2003). Task switching. *Trends in cognitive sciences*, *7*(3), 134–140.

Müller-Putz, G. R., Kobler, R. J., Pereira, J., Lopes-Dias, C., Hehenberger, L., Mondini, V., ... Sburlea, A. I. (2022). Feel Your Reach: An EEG-Based Framework to Continuously Detect Goal-Directed Movements and Error Processing to Gate Kinesthetic Feedback Informed Artificial Arm Control. *Frontiers in Human Neuroscience*, *16*. doi: 10.3389/fnhum.2022.841312

Nishimoto, S., Vu, A., Naselaris, T., Benjamini, Y., Yu, B., & Gallant, J. (2011). Reconstructing Visual Experiences from Brain Activity Evoked by Natural Movies. *Current Biology*, *21*(19), 1641-1646. doi: https://doi.org/10.1016/j.cub.2011.08.031

Omedes, J., Iturrate, I., Minguez, J., & Montesano, L. (2015). Analysis and asynchronous detection of gradually unfolding errors during monitoring tasks. *Journal of neural engineering*, *12*(5), 056001.

Pawar, U. (2020, Dec). *Lets open the black box of random forests.* Analytics Vidhya.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pfurtscheller, G., Müller-Putz, G. R., Scherer, R., & Neuper, C. (2008). Rehabilitation with Brain-Computer Interface Systems. *Computer*, *41*(10), 58-65. doi: 10.1109/MC.2008.432

Pires, G., Castelo-Branco, M., Guger, C., & Cisotto, G. (2022). Editorial: Error-related potentials: Challenges and applications. *Frontiers in Human Neuroscience*, *16*. doi: 10.3389/fnhum.2022.984254

Scherer, R., Friedrich, E. C. V., Allison, B., Pröll, M., Chung, M., Cheung, W., ... Neuper, C. (2011). Non-invasive Brain-Computer Interfaces: Enhanced Gaming and Robotic Control. In J. Cabestany, I. Rojas, & G. Joya (Eds.), *Advances in Computational Intelligence* (pp. 362–369). Berlin, Heidelberg: Springer Berlin Heidelberg.

Si-Mohammed, H., Lopes-Dias, C., Duarte, M., Argelaguet, F., Jeunet, C., Casiez, G., ... Scherer, R. (2020). Detecting System Errors in Virtual Reality Using EEG Through Error-Related Potentials. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (p. 653-661). doi: 10.1109/VR46266.2020.00088

Spüler, M., & Niethammer, C. (2015). Error-related potentials during continuous feedback: using EEG to detect errors of different type and severity. *Frontiers in human neuroscience*, *9*, 155.

Volosyak, I., Valbuena, D., Luth, T., Malechka, T., & Graser, A. (2011). BCI Demographics II: How Many (and What Kinds of) People Can Use a High-Frequency SSVEP BCI? *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *19*(3), 232-239. doi: 10.1109/TNSRE.2011.2121919

Wagner, F. B., Mignardot, J.-B., Le Goff-Mignardot, C. G., Demesmaeker, R., Komi, S., Capogrosso, M., ... others (2018). Targeted neurotechnology restores walking in humans with spinal cord injury. *Nature*, *563*(7729), 65–71.

Yuan, G.-X., Ho, C.-H., & Lin, C.-J. (2012). Recent advances of large-scale linear classification. *Proceedings of the IEEE*, *100*(9), 2584–2603.

Zander, T. O., & Kothe, C. (2011). Towards passive brain–computer interfaces: applying brain–computer interface technology to human–machine systems in general. *Journal of neural engineering*, *8*(2), 025005.

Zander, T. O., Krol, L. R., Birbaumer, N. P., & Gramann, K. (2016). Neuroadaptive technology enables implicit cursor control based on medial prefrontal cortex activity. *Proceedings of the National Academy of Sciences*, *113*(52), 14898–14903.

A Appendix

A.1 Lab computer specification

- Operating System: Windows 10
- Graphics Processing Unit: NVIDIA GeForce GTX1060 6GB
- Central Processor Unit: 11th Gen Intel(R) Core(TM) i7-11700 2.50GHz (16 cores)

A.2 Hyperparameters

A.2.1 LDA Classifier:

- Least squares solution (`solver='lsqr'`),
- automatic shrinkage of features (`shrinkage='auto'`),
- no prior class probabilities (`priors=None`),
- no specified number of components for dimensionality reduction (`n_components=None`),
- and the default covariance estimator (`covariance_estimator=None`)

Consult the documentation for further information.

A.2.2 Random Forest Classifier:

- One hundred (100) trees in the forest (`n_estimators=100`),
- using the 'Gini' impurity as the split quality measure (`criterion="gini"`),
- default tree properties (i.e.: no maximum depth, at least two (2) samples for a valid split, one (1) sample for a leaf),
- the square root of the total number of features as the number of features to consider when looking for the best split (`max_features="sqrt"`)

Consult the documentation for further information.

A.2.3 AdaBoost Classifier:

- Decision Trees as base estimators (`estimator=None`)
- Fifty (50) base estimators (`n_estimator=50`),
- a learning rate of one (`learning_reate=1.0`),
- using the 'SAMME.R' real boosting algorithm

Consult the documentation for further information.

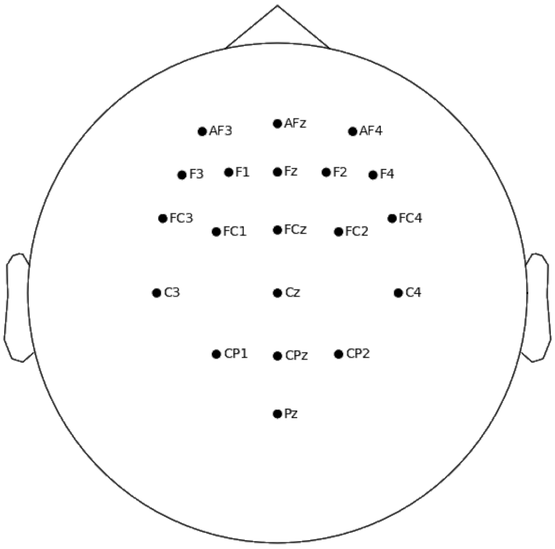


Figure A.1: Sensor cap layout for ActiCap Snap with all 20 channels used. The reference electrode was placed on the right mastoid and the ground electrode was placed on the left mastoid

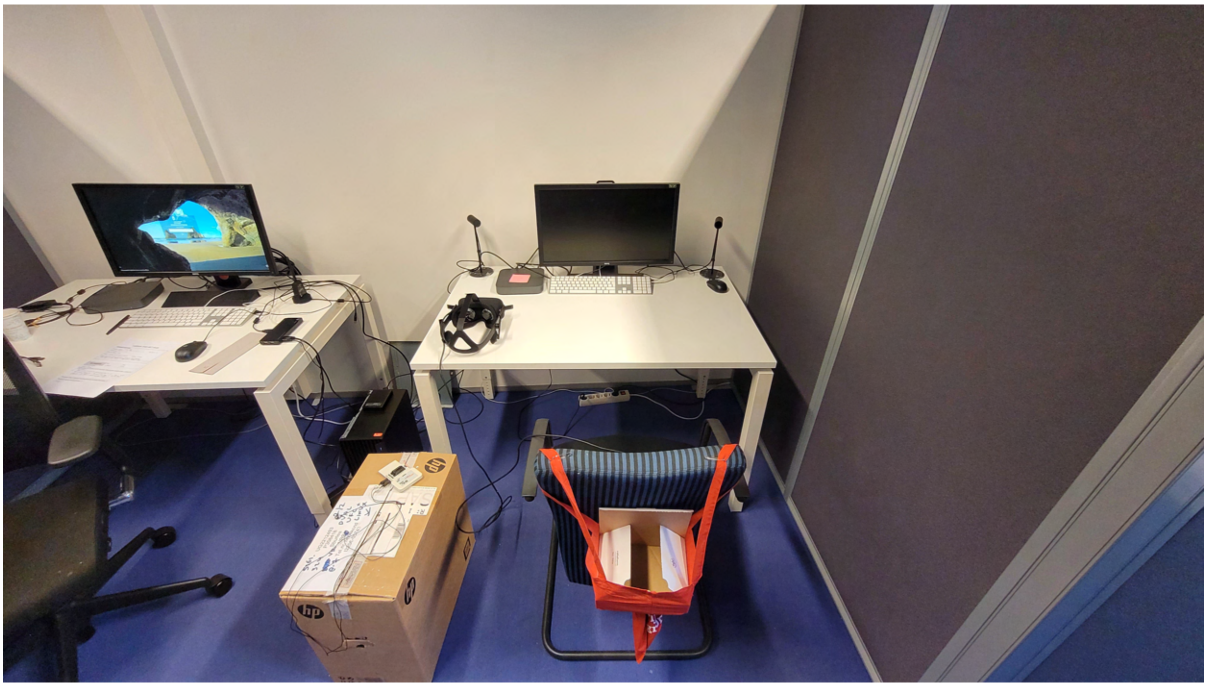


Figure A.2: Lab setup. The display on the left was used to monitor the EEG- and VR activity. The height of the virtual table coincided with the real table on the right

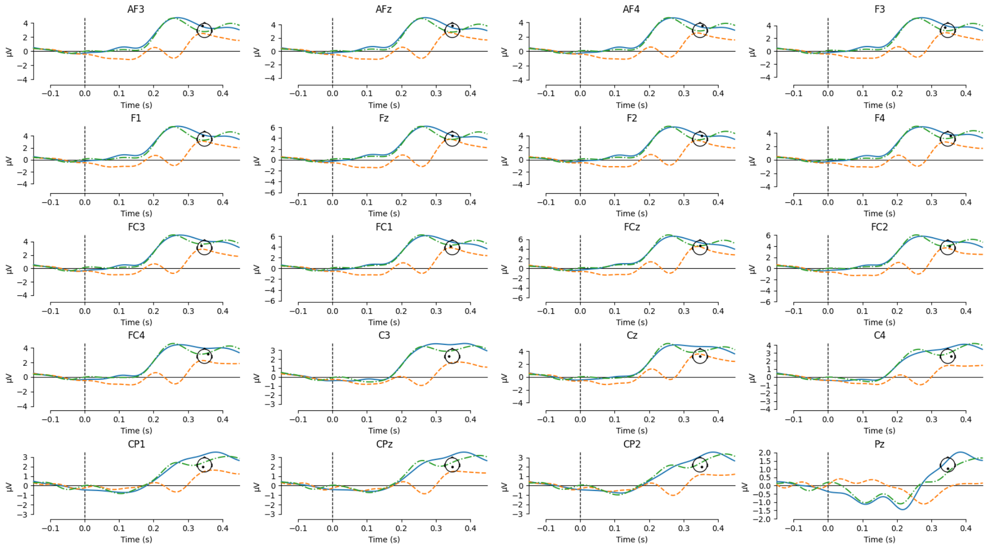


Figure A.3: Grand error signals across all channels

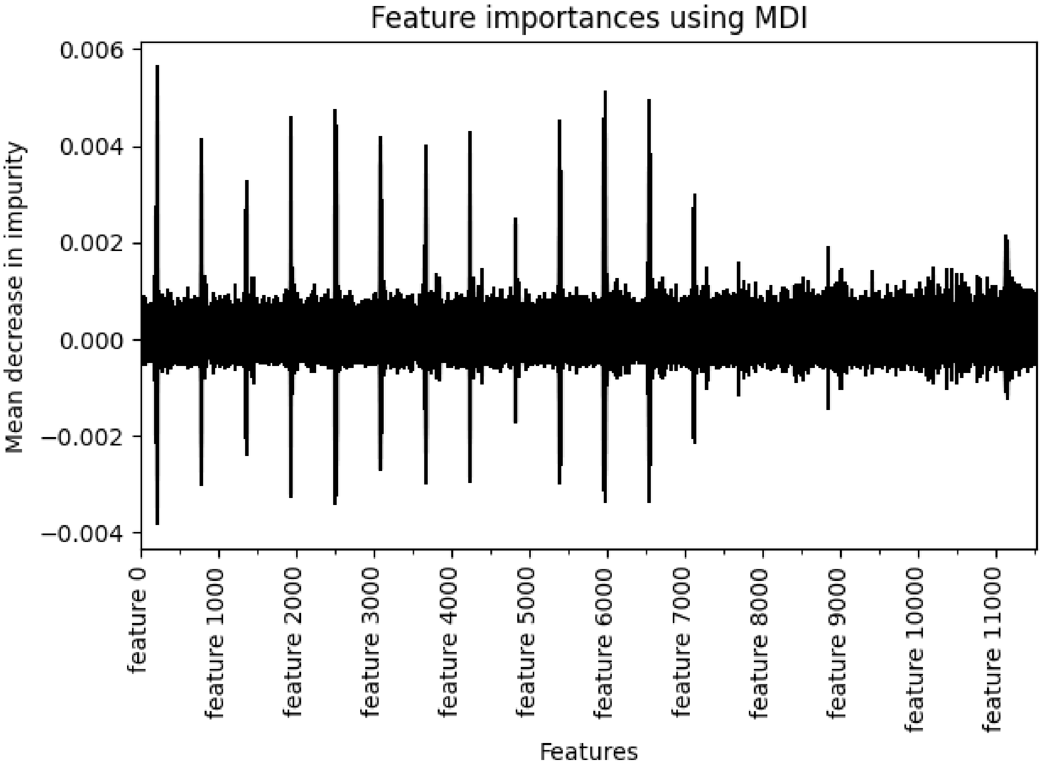


Figure A.4: Feature Importance of the RF Classifier using Mean Decrease in Impurity. The larger the bars, the more important that specific feature is