UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE

# Bachelor thesis

**Dan Yu Wang (jcx263) and Emma Cathrine Liisborg Leschly (sgk962)**

# Error detection in P300-BCI

**Advisor: Tuukka Ruotsalo**

**Handed in: June 11, 2021**

# Abstract

Error-related potentials (ErrP) are changes in brain activity recorded using electroencephalography (EEG) following a subject's perception of erroneous feedback from a brain-computer interface (BCI). The ability to detect such errors based on the alterations of EEG signals is relevant as it enables the computer to correct the mistakes and thereby improve the overall performance and usability of BCI.

The P300 speller experiment provides an opportunity to explore the detection of errors. In the experiment, the subjects perform a spelling task using a BCI by means of their recorded EEG signals. Sometimes, the computer misinterprets the intention of the subjects resulting in erroneous feedback. Previous research has shown that these erroneous feedback can be detected from the recorded EEG signals.

The objective of this research is to detect errors in a P300 speller experiment and compare the performance of error detection of neural network architectures against a state-of-the-art linear model. Here, we will compare the performance of a Linear Discriminant Analysis (LDA) with shrinkage to the performance of a Convolutional Neural Network with Long Short-Term Memory (CNN-LSTM) and an UNet architecture based on several Convolutional Neural Network.

Our findings show, that LDA and CNN-LSTM achieve statistically similar performances and that both models perform significantly differently than UNet, which was found to have a stronger bias towards the majority label (non-errors).

Our findings confirm that error detection in EEG-based BCI is possible. Furthermore, our findings support the idea, that modern neural network architectures can be used for detecting errors in EEG signals with results comparable to those of more conventional linear classification models. Finally, our findings revealed a trade-off between sensitivity and specificity depending on the model architecture. Further work still remains, especially in understanding how tuning the architecture and the hyperparameters in neural network architectures might improve performance.

# Contents

# 1 Introduction

## 1.1 Motivation

Being able to interact with an environment through one's mind may sound like science fiction. However, decoding a user's intentions from brain signals and translating them into commands for a computer system is an active area of research and has already been implemented in several applications from assistive technologies for motor-impaired users to prosthetic rehabilitation devices, wheelchair control systems, and gaming devices [12] [15] [29] [34].
These systems are referenced as brain-computer interfaces (BCIs). In many other human-computer interaction systems (HCIs), users need to perform deliberate actions such as typing, touching, or gazing to produce input for the system. With BCIs, users can interact with the system through their measured brain activity alone. This allows users who would otherwise not be able to control such devices due to physical and motoric limitations to communicate and interact with the system and hopefully gain more autonomy [29], while also improving accessibility for ordinary users.
However, measurements of brain activity are often noisy [5] and the BCI will sometimes misinterpret the user's intention resulting in an erroneous command [22]. This can be frustrating for the user and prolong the time it takes to perform the desired action [22]. Indeed, Lotte et al. [16] identified the low reliability and performance of BCIs as some of the main challenges in the field today.
Overall, there are two ways in which error detection can improve the BCI: First, error detection can be used in a corrective manner to correct the mistake that has been made by the BCI. This could for example be by replacing the erroneous initial guess with the second-best suggestion or preventing the command from being fully executed [4]. Secondly, it can be used in a adaptive manner where the BCI system is continuously re-calibrated and retrained using its mistakes in order to improve future predictions [15] [4]. Therefore, being able to reliably detect when these errors are made is important as it enables the system to correct the mistakes and improve the overall performance and usability of the BCI.

## 1.2 Research questions

Our research will focus on the detection of errors in BCI systems as we first ask our primary research question:

***How can measurements of electroencephalography signals be used to detect errors in a BCI?***

While our primary research question focuses on how electroencephalography (EEG) signals can be used to detect errors, through our second research question, we want to investigate whether the recent progress in the field of machine learning and neural networks can improve upon the classification performance of a conventional classifier which we consider our baseline model. Hence, we investigate the application of newer classification models on the same experimental data and evaluate their performance asking the question:

***Can neural network architectures be used to improve the performance of error detection in BCI compared to conventional classifier methods?***

## 1.3 Structure

This thesis will first provide a theoretical background for our research and outline the related work which has already been done in the field in the *Background* section. We will then introduce our data as well as our pipeline for preprocessing, feature extraction, and feature selection in the *Methods* section. In this section, we will also introduce our classification models: Shrinkage Lin-

ear Discriminant Analysis (LDA), Convolutional Neural Network with Long Short-Term Memory architecture (CNN-LSTM), and a UNet neural network architecture. In the *Results* section, we will present a descriptive analysis of the data as well as the performance of our models in the error detection task. Lastly, in the *Discussion and conclusions* section, we will discuss the implications and limitations of our findings, present suggestions for future research and finally conclude with the implications of our work.

# 2 Background

In this section, we will provide a theoretical background on how EEG is used in BCIs, how event-related potentials (ERP) can be used as markers for error detection, and how P300 speller is used within the context of ERP and BCIs. Lastly, we will provide an outline of the related work in the field of error detection in BCIs.

## 2.1 Using EEG for brain-computer interfaces

A widely used technique to measure brain activity is EEG. EEG measures the electrical potentials through electrodes that are positioned on different locations on the scalp for example via an EEG cap. The changes in electrical potentials over time reflect the underlying brain activity. However, this reflection is not perfect as the electrodes measure the summed activity from up to millions of neurons, and thus, it is difficult to know exactly from which part of the brain the signals arise [20]. Measurements are also easily affected by artifacts such as muscle movements, eye blinks, [5] and the attention and engagement of the user [24]. These factors cause high inter- and intra-variability in EGG which poses a challenge for building reliable classification methods with high performances and generalization abilities [16].
While EEG generally has a poor spatial resolution compared to other techniques such as functional magnetic resonance imaging (fMRI), the temporal resolution is high. This makes it suitable for tasks such as determining responses to stimuli which often unfold within the scope of milliseconds. Along with its non-invasive nature and the relatively low cost of the experimental setup, this makes EEG a popular technique in neuroscience studies [5] [35].

The variation in the brain activity that can be observed in response to a stimulus is referred to as an event-related potential (ERP). The ERP can be represented as a waveform with peaks and troughs that represent the brain activity and thereby the underlying cognitive processes such as attention and perception [35]. As ERPs are time-locked to a stimulus, it is possible to measure the changes in the brain activity in response to an event. Averaging over multiple time-locked trials allows for observation of overall patterns in cognitive responses to different stimuli despite the previously mentioned drawbacks of EEG measurements.

Different temporal parts of the ERP are referred to as *components*. The components are often denoted by a polarity ('P' for positive or 'N' for negative) and a latency (often in milliseconds) [26]. An example of this is the perhaps most well-known component: the P300 which is a positive deflection around 300 ms post stimulus. It is evoked in response to unexpected or rare stimuli [27] and can for example be used to select items in a BCI [22]. Other well-known components are the N400 which is used in language research to understand semantic processing [2], the Mismatch Negativity (MMN) which is a negative deflection in response to mismatching auditory stimuli [24] and early components such as the C1, P1 and N1 which are associated with the first, quick selective attention processes in response to stimuli [26]. The early components are referred to as exogenous meaning that they depend on the physical features of the stimulus. In contrast, later components such as the P300 are endogenous. This means they do not depend on the physical features of the

stimulus but instead on the attention and information processing of the user [31] making them useful as markers in studies of cognitive processes and intentionality [32].

An error-related potential (ErrP) is a subtype of ERP which can be used as a marker in error detection. It was discovered in the 1990s when researchers Falkenstein et al. [6] and Gehring et al. [9] discovered a difference between the brain activity of subjects during correct and incorrect trials of simple choice reaction tasks.
Although many variations of ErrP have been observed [12], it has been found to consist of two components: A negative component generally elicited around 50-100 ms post-stimulus followed by a positive component elicited between 200-500 ms post-stimulus [27]. These components have shown to be reliable over time and across tasks [4]. The negative component is referred to as error-related negativity (ERN).
In our research, we are interested in a specific type of ERN namely feedback-ERN (FRN) which is observed when erroneous feedback is presented to the subject. We are interested in detecting errors made by the computer and not the errors made by the subjects. Since it requires information processing for the user to not only perceive the physical feature of the feedback but also interpret it as erroneous, this type of FRN is elicited later around 200-300 ms post feedback [4]. In contrast, the ERN that Falkenstein et al. [6] and Gehring et al. [9] observed, was elicited in experiments where the errors were committed by the subjects themselves in a speed choice reaction task.

Figure 1 from Perrin et al. [22] shows an example of the electrical potentials during trials with correct and incorrect feedback in a BCI experiment as well as their difference waveform. The graphs show the grand average which means that that the results have been averaged across subjects and trials to obtain a robust ERP [22]. The figure shows a FRN (here denoted 'Neg-ErrP') around 300 ms post-stimulus and a following positive component (here denoted 'Pos-ErrP') around 450 ms post-stimulus. As these components show, there is an observable significant difference between the two types of trials (correct and incorrect feedback) when averaged across trials. This makes them useful for error detection in a BCI setting.

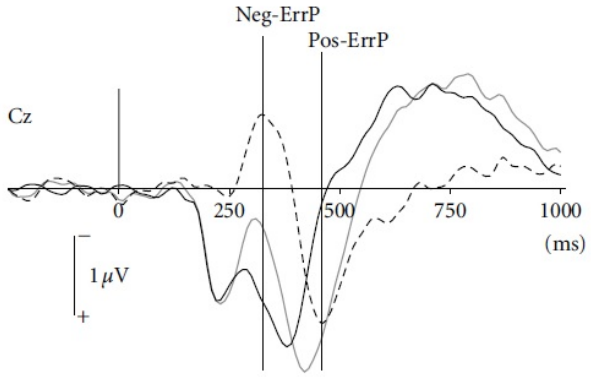



**Figure 1:** Perrin et al. [22] Grand average ERP (250 ms to 1000 ms, channel Cz) associated with responses to correct feedbacks (black solid line), responses to incorrect feedbacks (grey solid line), and the difference between the two which is the ErrP waveform (black dashed line).

The application context for the BCI that we will consider in this thesis is the P300 speller. It is an assistive technology developed to restore communication abilities for motor-impaired users by allowing them to spell letters and words [22] [27]. The first P300 speller was introduced by Farwell and Donchin [7] in 1988. Since then, various designs have been developed [21] but the main idea remains the same. It is based on the rapid serial visual presentation (RSVP) paradigm in

which items are displayed rapidly and successively to the user [14]. The stream of items consists mostly of non-target items mixed with a few target items [14].
With the P300 speller, the user sits in front of a screen and watches groups of letters flashing rapidly as illustrated in Figure 2. The user is instructed to pay attention to the target item - the letter that he or she wants to spell - and ignore the flashing non-target items. Meanwhile, the user's brain activity is recorded typically through electrodes placed on an EEG cap. Based on the brain activity, the BCI can decode which letter the user intends to spell based on the assumption that only the flashing groups which include the target letter will elicit an ERP response. The response involves the P300 component which has given name to the speller. The combinations of groups of letters are designed such that the BCI can decode the target letter that is consistent across the groups which elicited an ERP response. However, noise and other factors such as low attention and fatigue may mean that the basis for decoding is not ideal. This can result in the BCI predicting the wrong letter as the target letter. It is in this case that we want the BCI to be able to detect the error and apply a corrective procedure such as making a new guess and possibly also learn from the error to improve future predictions using adaptive learning methods.


**(a)** A P300 spelling setup where rows or columns of letters flashes as employed by Spüler et al. [29]

**(b)** A P300 spelling setup where groups of pseudo-random letters flashes as employed by Perrin et al. [22].

**Figure 2:** Two examples of common P300 speller systems.

## 2.2 Related work in error detection in brain-computer interfaces

The interest in BCIs and their applications is increasing and a wide range of different methods for classification as well as the preliminary steps of data preparation, feature extraction, and feature selection have been researched [16]. Early research largely applied standard machine learning classification models such as linear classifiers like Support Vector Machines (SVMs). Later research has focused on identifying classification models that tackle some of the challenges faced in BCIs such as the low signal-to-noise ratio (SNR) and the intra- and inter-variability of EEG signals [16] [3] that prevent the trouble-free use of BCIs [29].

Some of the early findings on error detection in a P300 speller were discovered by Visconti et al. [33] who in their study from 2008 achieved a sensitivity of 0.71 and specificity of 0.82 using a SVM model and sensitivity of 0.64 and sensitivity of 0.77 averaged across subjects using a Linear Discriminant Analysis (LDA) model. These models (SVM and LDA) were and still are some of the most popular classifiers for BCIs and adaptions of them are continuously being developed [16]. In 2012, Perrin et al. [22] achieved a sensitivity of 0.63 and a specificity of 0.88 averaged across subjects using a Gaussian Mixture Model (GMM) for error detection.

In another P300 speller study from 2012, Spüler et al. [29] applied SVM and LDA and reported an average sensitivity of 0.35 among a group of *amyotrophic lateral sclerosis* (ALS) patients while the sensitivity for the non-ALS participants ranged from 0.40 to 0.50. One of the reasons why these

sensitivity scores were low compared to other studies is the fact that Spüler et al. [29] only considered algorithms with specificity above 0.90. They considered it more important that a correct letter was not wrongly detected as an error (specificity) than that an incorrect letter was detected (sensitivity). This is because they assumed that the first would be more frustrating for users and could potentially affect their motivation and thus the performance of the BCI resulting in a vicious circle [29]. Perrin et al. [22] also considers it important to retain a high specificity while attempting to improve sensitivity as they label a false positive (a correct letter wrongly detected as an error) the worst case [22].
The research of Spüler et al. [29] offers an interesting and important perspective because it actually includes subjects from one of the patient groups that the P300 speller is aimed at. In contrast, the majority of the research in the area is only conducted on able-bodied subjects and thus relies on the assumption that the knowledge gained from these subjects can be transferred viably to the patient groups.
In a study from 2016, Zeyl et al. [37] achieved a sensitivity of 0.87 and specificity of 0.97 using a variation of LDA called bidirectional stepwise LDA (SWLDA) and by incorporating information about the confidence of the P300 speller based on a continuous update of the posterior probabilities of the rows and columns in the P300 speller being the target into the predictions.

Given the progress in the field of machine learning and neural networks and their high performance on other classification tasks such as images and text [5] [23] [16], these models have also been explored for the task of EEG classification in general and error detection in particular. However, according to the review by Lotte et al. [16] from 2018, neural networks have yet to show superiority to other state-of-the-art methods. One of the challenges is the large number of parameters that require many training samples - something that is generally not possible in a BCI setting [16]. On the other hand, an advantage of neural networks is that raw EEG signals can be fed directly into the model without the need for preprocessing and handcrafted features [5].

In 2017, Torres et al. [32] used a Convolutional Neural Network (CNN) for error detection and achieved ErrP accuracies[1] in the range of 0.75-0.80 and non-ErrP accuracies of 0.84 which improved the performance of both GMM and SVM. Recently, neural networks have been used in other EEG classification tasks such as sleep stage scoring [30] [23] and epileptic seizure recognition [36]. In 2017, Supratak et al. [30] proposed DeepSleepNet which is a neural architecture that combines a CNN with a Long Short Term Memory (LSTM). Incorporating the LSTM allows the model to learn the temporal nature of the data which proved useful for classifying sleep stages as they are temporally ordered [30]. Their model achieved accuracies between 0.81 - 0.86 on different datasets and performed as good as state-of-the-art methods while being able to take the raw EEG data as input and thus not rely on handcrafted features [30]. Xu et al. [36] proposed a CNN-LSTM in an epileptic seizure recognition task based on EEG recordings. They achieved an accuracy of 0.99 on a binary task and 0.82 on a five-class task [36]. In 2019, building on the U-Net architecture from Ronneberger et al. [25], Perslev et al. [23] proposed U-Time which achieved performance beyond that of DeepSleepNet [30] on several datasets [23].

From the above summary of the work which has been done in the field of error detection and EEG classification in general, we note that there are significant methodological differences and variations from study to study which can make the results difficult to compare. Variations are observed in both the experimental setup of the P300 speller, the numbers of subjects (ranging from

[1]Torres et al. [32] reports accuracies without specifying how accuracy is calculated. Since accuracy in statistical terms is not the same as sensitivity or specificity, comparison with other results should take this into account. The general accuracy measure can only be considered valid if the classes are balanced [16] which is rarely the case in error detection where the number of errors is usually smaller than the number of non-errors.

only 2 to more than 20), differences in how the data is being preprocessed (feature extraction, standardization etc.), the choice of classification models, and the measures that the models are evaluated on. This makes comparison difficult which is a common problem in the field of EEG research [5].
Lotte et al. [16] point out that many studies on neural networks do not compare their performance to state-of-the-art models or that when they do, they select suboptimal parameters for the state-of-the-art models or do not justify the parameters of the neural network which leads to biased conclusions. This is of course not true for all studies using neural networks, although we did observe ambiguous and unjustified choices of measures.
Another potential issue is that EEG studies generally are conducted on non-public or non-published experimental data which also complicates replication and comparison of results [5]. However, we do note that in the studies in which different classification model are used on the same dataset such as Craik et al. [5], Kumar and Vinod [13], Torres et al. [32], Supratak et al. [30], Perslev et al. [23], applying a neural network framework leads to a significant improvement in error detection compared to the previously more popular classifiers such as LDA and SVM.

# 3 Methods

In this section, we introduce the methodology of our research including the dataset that we use and the experimental procedure of the study from which it was obtained. We describe our pipeline for the preprocessing steps including augmentation of the data as well as our pipeline for predictive modeling. We also describe the measures that we have selected to evaluate our models, which largely draw inspiration from the studies mentioned in the previous section. Lastly, we explain how we test for the statistical significance of the results.

## 3.1 Data

We use a dataset made available in a Kaggle competition that was part of the IEEE Neural Engineering Conference in 2015[2]. The data set contains EEG recordings from 26 subjects who took part in a P300 speller experiment. In the experiment, the subjects spell words letter by letter while their brain activity is recorded with 56 passive EEG electrodes placed around the scalp as illustrated in Figure 3.
Each subject goes through a total of five sessions of spelling. Sessions 1 - 4 consist of spelling 12 predefined five-letter words while the last session consists of spelling 20 five-letter words chosen by the subjects themselves. This corresponds to 60 letters pr. session in sessions 1 to 4 and 100 letters in session 5. In total, we have 340 trials per subject and 8,840 trials in total from the 26 subjects. These trials are also known as *epochs* in the context of EEG research. Every spelling event is denoted as feedback '0' ('incorrect' feedback corresponding to the computer choosing a non-target letter) or '1' ('correct' feedback corresponding to the computer choosing the target letter).

[2]BCI Challenge @ NER 2015 [1]

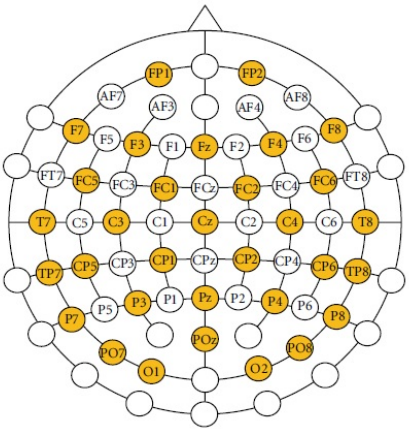



**Figure 3:** Placement of electrodes for EEG recording following the 10-20 system. Figure from Perrin et al. [22].

The course of spelling a single letter is illustrated in Figure 4. First, the target letter is indicated by a green circle [phase 1] for 1 second. Letters are then flashed in groups of 6 for 60 ms applying a pseudo-random stimulation procedure [phase 2]. The subject is told to focus on the target letter and ignore the rest. As explained in the previous Section 2.1, the target letter can then be decoded by the BCI as only groups containing the target letter are expected to elicit a P300 response. Based on this, the computer can make a prediction of the letter as intended by the subject to spell after a delay of 2.5 - 4 seconds. The letter chosen by the computer is displayed for 1.3 seconds [phase 3], in which the subjects were instructed not to blink during this feedback presentation as this would introduce artifacts in the measurements. Finally, if the letter is the last letter in the word, a 4.5 second break occurs during which a display indicates the next word [phase 4] and the difficulty level (slow or fast spelling mode) [phase 5]. If not, there is a short 0.5 second break before the next green circle appears [phase 1] Perrin et al. [22]. As our research is focused on the topic of error detection, it is during [phase 3] in which the predicted letter is revealed, where we expect the subjects to elicit error-related potentials when a wrong letter is revealed.
Two spelling conditions were used: A fast, more error-prone one and a slower, less error-prone one. Perrin et al. [22] used the faster method to force the generation of more errors among the subjects in order to obtain a sufficient number of errors for subsequent model training. In order to simplify our classification task, we assume that the overall EEG patterns between these two methods are similar, though it remains an assumption that needs to be considered.

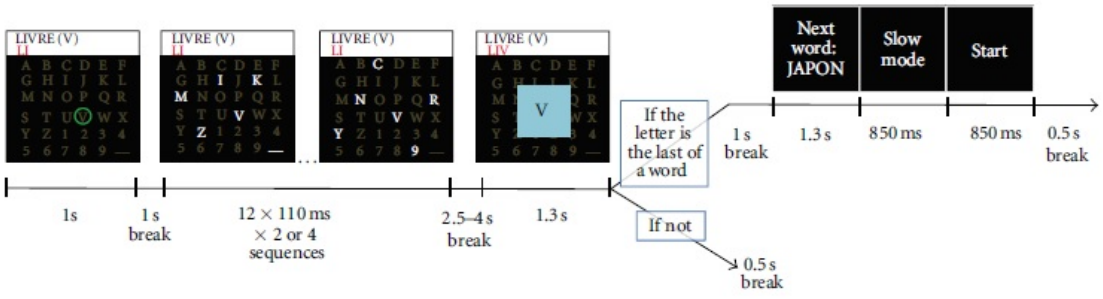



**Figure 4:** The process of spelling a letter. Figure from Perrin et al. [22].

According to the description on Kaggle[3], "*most of the data for this competition come from this study [Perrin et al. [22]*]". However, since Perrin et al. [22] only reported 16 subjects in their study, ten or more subjects must come from different studies. It is stated that all subjects followed

[3]BCI Challenge @ NER 2015 [1]

the same procedure during the data collecting process which we assume to be true. The Kaggle competition data has been downsampled at 200 Hz from the initial 600 Hz which is the sample frequency of the original experiment in Perrin et al. [22]. A sample rate of 200 Hz means that for every second of the experiment, 200 recordings are made.

## 3.2 Preprocessing

EEG data is inherently noisy due to the fact that the EEG electrodes also pick up many signal noises and artifacts such as nerve impulses from eye blinks, muscle movements, or electrode displacement when the subject moves between trials [5]. Furthermore, the signal strength is also affected by psychological factors such as fatigue, motivation, and stress level [28].
To illustrate the volatility of EEG recordings, a plot of the raw recordings during a single session of the experiment for a randomly chosen subject is shown in Figure 13 (Appendix). It illustrates that with the same subject and within the same session, the electrical potentials are volatile and exhibit both global trends and local peaks. For the same randomly chosen subject and session, the raw recordings during the spelling of the first word are illustrated in Figure 14 (Appendix) for multiple EEG channels. It shows a general downward trend during the spelling of the word (which consists of five letters) which could be due to fatigue, demotivation, or simply familiarization with the task. Furthermore, we see that the shift in signal strength is persistent across the included EEG channels. It shows that the channels differ in the intervals in which they record signals e.g. P08 records signals at a high voltage compared to Pz.

When analyzing EEG, a significant amount of work and consideration needs to be put into the preprocessing steps before the data can be given as input to a classification model due to mentioned volatile nature of EEG signals. The overview of our data preprocessing is shown in Figure 5. A wide array of different preprocessing methods are common during the preprocessing steps of EEG feature extraction [5, 17, 16, 19]. However, as a comparison of these are not the focus of this thesis, we will not be exploring pros and cons of different preprocessing methods. Instead, we will largely follow the procedure described and applied by Perrin et al. [22], where the data originated from, and use this procedure for all the classification models.

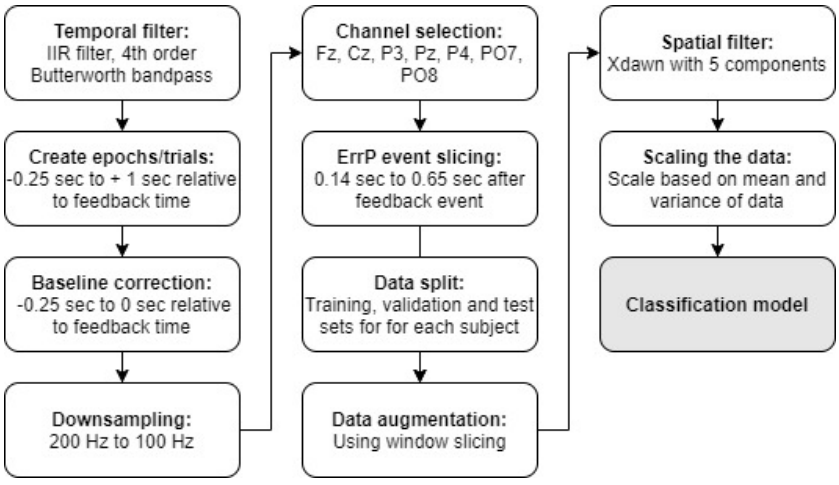



**Figure 5:** Data preprocessing, feature extraction and feature selection to prepare data for subsequent use in error detection.

We first apply a temporal bandpass filter on the raw data in the range of 1 - 20 Hz using an IIR fourth-order Butterworth filter to increase the signal-to-noise ratio (SNR). In order to reduce the

data size, we extract epochs i.e. trials from the filtered raw data defining an epoch as lasting from 0.25 seconds prior to feedback event until 1 second post feedback event. Then we apply a baseline correction to all epochs based on the means of the signals of the pre-stimulus time frame[4]. Lastly, the epochs are then downsampled from 200 Hz to 100 Hz (100 timestamps per second).
Following the procedures of Lotte [17] and Krusienski et al. [11], we have chosen to focus on the following seven channels: $[Fz, Cz, P3, Pz, P4, PO7, PO8]$, as some of the most discriminable EEG signals relevant to the P300 spelling task, occurs at these locations as shown by Krusienski et al. [11]. Krusienski et al. [11] also recommend including the Oz channel which was not measured in our data and thus omitted. After selecting the channels, we extracted the time steps between 0.14 - 0.65 seconds after the feedback event. The extraction is inspired by Perrin et al. [22] who used the interval 0.2 - 0.6 post feedback. Our slightly larger time interval is chosen in order to facilitate data augmentation.
Afterward, we split the data for every subject into three sets: training, validation, and test. Elaborations on the choice of data splits are further explained in Section 3.3. As we have a limited amount of data which is especially relevant for the neural network models, we use a window slicing method to augment our data with more samples. Our initial extracted data lies in the interval between 0.14 - 0.65 seconds post feedback. From this, we can generate artificial yet realistic data by further sub-extracting samples in the range of 0.14 - 0.61 seconds to 0.18 - 0.65 seconds for every epoch as illustrated in Figure 6. This augmentation strategy allows us to generate four additional samples from every epoch, thus greatly augmenting the available data. As the grand average ERP plot in Figure 1 shows, the average distance between the Neg-ErrP and Pos-ErrP are around 150 ms, while we only augment the data within an interval of size 50 ms. Thus, we assume that the augmented data retains many of the same signal characteristics as the original data while still providing some additional variability within the classes themselves.

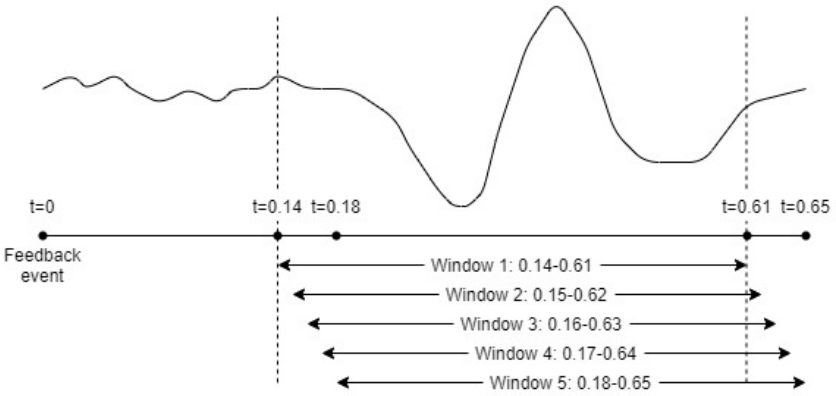



**Figure 6:** Data augmentation with multiple signal windows.

Lastly, we apply a spatial filter using Xdawn followed by scaling of every epoch around the respective means and variance before we use the data as input to the classification models.

## 3.3 Predictive modeling

In order to examine whether EEG signals can be used for error detection, we use the preprocessed and augmented data described in Section 3.2 as input to the predictive modeling pipeline shown in Figure 7. Because of the high inter-subject variability in EEG recordings, the general procedure in EEG-based BCI such as the P300 speller is to create and train a subject-specific model for every subject [10]. In practice, this means that the BCI needs to be calibrated for every subject through a calibration session [18]. This is different from the common machine learning pipeline. Generally,

[4] 0.25 seconds before the feedback stimulus up until the feedback stimulus time.

a single model is trained using all the available data and similarly, performance is evaluated on a single test set.

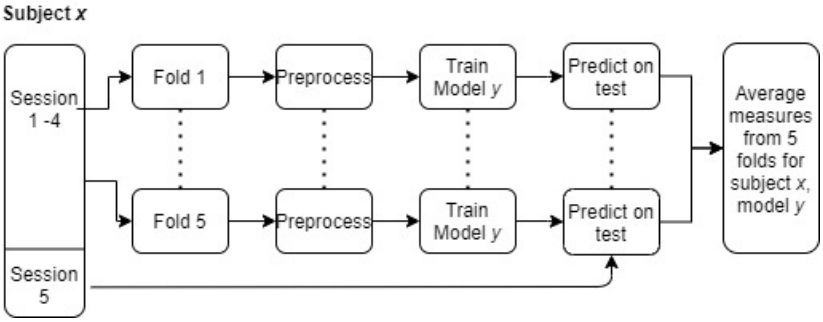



**Figure 7:** Predictive modelling pipeline for a given subject *x* using model *y*

We use the first four sessions (each consisting of 60 epochs resulting in a total of 240 epochs per subject) as the training set keeping the entirety of session 5 with 100 epochs as our test set. This splitting strategy is chosen such that it most closely corresponds to a real-life scenario in which the BCI is first calibrated and trained followed by a *real-life* use of the speller system represented by testing the performance on an unknown test set.
In order to reliably estimate the performance of the model, we furthermore apply five-fold cross-validation on the data. The training data from sessions 1 to 4 for each subject is split into five folds of data with each fold contain 80 % of the data for training and the remaining 20 % for validation. Note that we create these folds before preprocessing to avoid data leakage between the training and validation sets. This is particularly relevant since we apply data augmentation by window slicing meaning that some samples are close to identical. Each of the folds is trained on a new instance of the classification model. This is done in order to test for the variance of the model performance i.e. if the model performs consistently or not when provided with different sets of data as input for training.
After training and validation, each of the five models is evaluated on the test set (session 5). We compute sensitivity, specificity, error precision, non-error precision, AUC, and F-score which are further described in Section 3.4. To evaluate the overall performance of each subject-specific model, we average across the measures for each of the five folds.

### 3.3.1 Linear Discriminant Analysis (LDA)

As our baseline model, we use a Linear Discriminant Analysis (LDA) model which is one of the most popular choices of classification models with respect to ERP studies and has shown state-of-the-art performance in many previous studies both earlier and recent [5] [16]. We will apply regularization in the form of shrinkage (shrinkage-LDA) which has been shown to improve generalization performance [3] and is also recommended by Lotte et al. [16].

### 3.3.2 CNN-LSTM architecture

The second model which we have chosen to implement is a Convolutional Neural Network with Long Short-Term Memory architecture (CNN-LSTM) that largely follows the architecture suggested in Xu et al. [36] as shown in Figure 8. The main idea behind this model is to first use a series of one-dimensional convolutional neural networks (CNN) to extract the most characteristic features of the input signals. This is followed by two layers of Long Short-Term Memory (LSTM) which can learn the temporal relations between different parts of the signal. In the end, three fully connected (FC) layers and a softmax layer make the final predictions of the label.

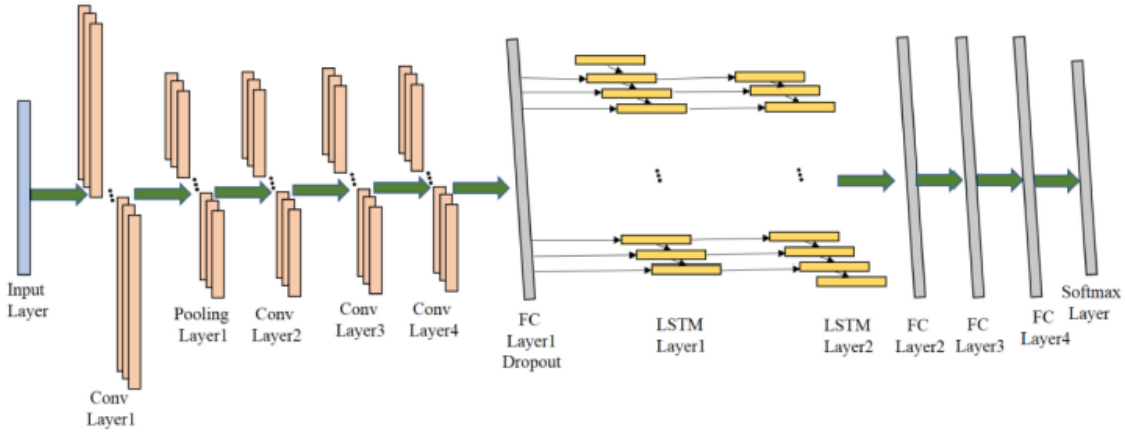



**Figure 8:** The structure of the one-dimensional CNN-LSTM model for signal classification. Figure from Xu et al. [36]

### 3.3.3 UNet architecture

As our third model, we have implemented a neural network architecture inspired by the UNet model proposed by Ronneberger et al. [25] in 2015. The architecture is shown in Figure 9. UNet is a type of CNN but instead of requiring high numbers of samples for proper training, an advantage of the UNet architecture is that it makes more efficient use of limited samples of data for classification through data augmentation [25]. This is achieved through a series of convolutions with max-pooling (encoding) followed by a series of symmetric up-convolutions (decoding), in which the input for every step of the up-convolutions is a concatenation of the encoded and the decoded results as shown in the figure. Thus the decoding steps maintain the localized features from the correspondingly encoding steps on the same level.

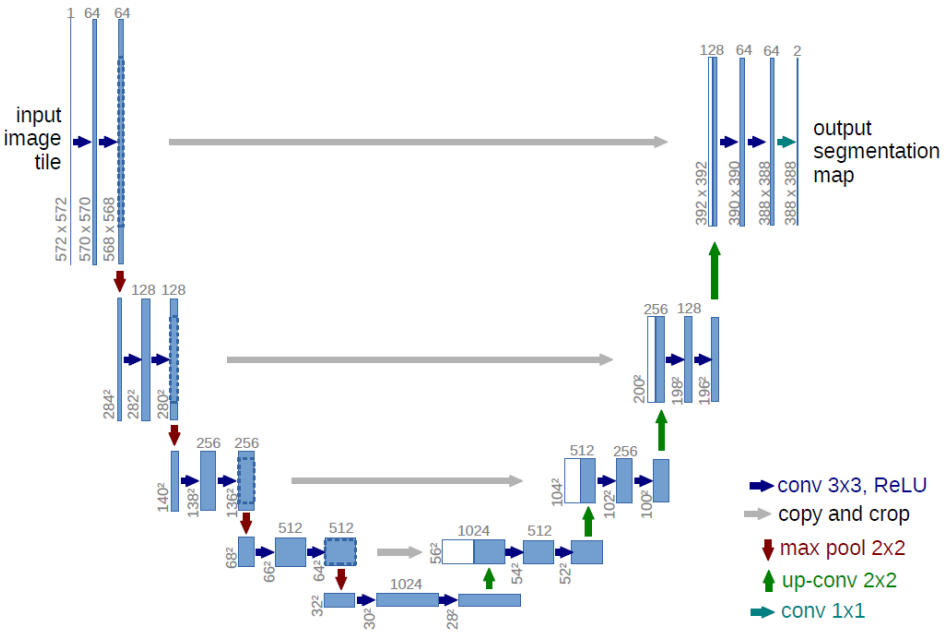



**Figure 9:** Illustration of UNet from Ronneberger et al. [25], showing a downward contracting path (encoding) and an upward expanding path (decoding) forming the U-shape.

### 3.3.4 Technical implementation

Our project is completed in *Python (v3.6)* using the *Google Colab* environment. For the initial processing of the EEG data, we have used the *MNE (v0.22)* Python package. For LDA, we have used the implementation from the *scikit-learn (v0.24.2)* Python module. Our neural networks are implemented using *Pytorch (v1.8.0)*. The implementation and related results can be found at our Github repository: `https://github.com/ArcticMooncake/BA_thesis_DYW_ECLL`.

## 3.4 Measures

In order to evaluate and compare the classification models, we use a set of different measures. These measures are *sensitivity* (error recall), *specificity* (non-error recall), error, and non-error *precision*, *Area Under the Curve* (AUC) and *F-score*. Note that we consider errors as the *positive* class since these are the cases that we want to detect, while the non-errors are considered the *negative* class.

*Sensitivity* is also known as True Positive Rate (TPR) as defined in Eq. 1 measures the ratio of the number of samples that we correctly detected as errors out of all the samples that are indeed errors. In other words, how good the model is at finding all the erroneous letters that exist in the dataset. Low sensitivity means that the model is bad at recalling i.e. bad at detecting all the erroneous letters we have in the data. *Specificity* is also known as True Negative Rate (TNR) as defined in Eq. 2 and measures the number of samples that we correctly detect as non-errors out of all the samples which are indeed non-errors in our data.

$$\text{Sensitivity (error recall)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{Eq. 1}$$

$$\text{Specificity (non-error recall)} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{Eq. 2}$$

*Error precision* describes the Positive Predictive Rate (PPR) as defined in Eq. 3. When considering errors as positive cases, error precision describes how many of the samples, which we detect to be errors, are indeed errors when the model tries to detect the class of an unknown signal. Similarly, non-error precision describes how many of the signals, which we detect to be non-errors, that are indeed non-errors as defined in Eq. 4.

$$\text{Error precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{Eq. 3}$$

$$\text{Non-error precision} = \frac{\text{TN}}{\text{TN} + \text{FN}} \tag{Eq. 4}$$

*F-score* is a measure of the harmonic mean between precision and recall. For errors, it is defined in Eq. 5. The F-score measures how well the model performs on finding all the errors in the data (sensitivity) and how many non-errors the model mistook as errors (error precision). F-score for non-errors is equivalently defined in Eq. 6. We calculate the F-score for both errors and non-errors and calculates the unweighted average of them. Consequently, a high (unweighted) F-score is only possible when the model manages to achieve high recall and precision on both errors and non-errors.

$$\text{Error F-score} = \frac{2}{\text{error precision}^{-1} + \text{sensitivity (error recall)}^{-1}} \quad \text{(Eq. 5)}$$

$$\text{Non-error F-score} = \frac{2}{\text{non-error precision}^{-1} + \text{specificity (non-error recall)}^{-1}} \quad \text{(Eq. 6)}$$

*AUC* describes the model's ability to distinguish between the classes - in our case, between errors and non-errors. It is computed as the area under the *Receiver Operating Characteristic* (ROC) curve which plots the (1 - specificity) against the sensitivity. An AUC score of 1 means that all predictions are correct while an AUC of 0 means that all predictions are incorrect.

### 3.5 Test of statistical significance of the results

In order to compare the performance of LDA, CNN-LSTM, and UNet for every subject, we apply the Mann-Whitney U test which tests for significance of the cross-validated averages pairwise between the three models for each of the six included measures. We decided on using the Mann-Whitney U test since the preliminary test using the Shapiro-Wilk test revealed that a number of the five-fold cross-validated scores across different models and subjects were not normally distributed and thus the assumptions of a t-test were not met.

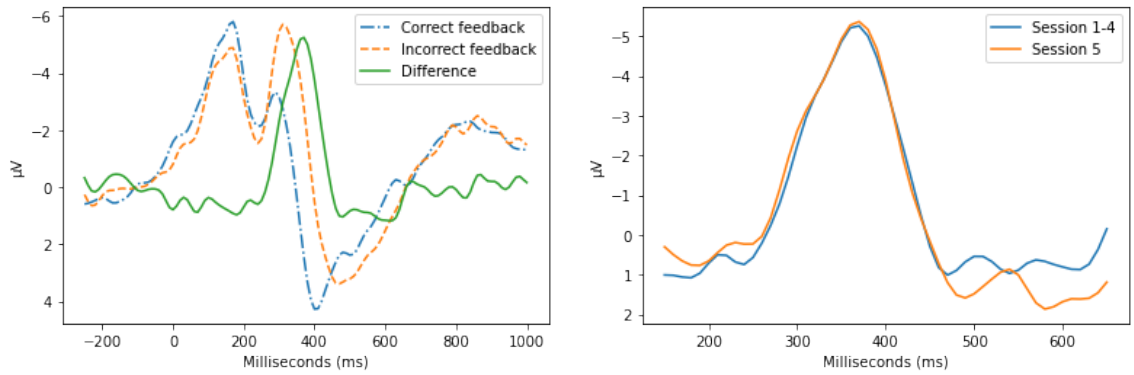
## 4 Results

### 4.1 ErrP and exploratory analysis

As described in Section 2.1, incorrect feedback from the BCI can be detected in the subjects' recorded brain signals as it is expected to elicit an ErrP response consisting of two components: The FRN around 200-300 ms post-stimulus and a subsequent positive deflection. To investigate whether we can observe these components in our data and whether there is a difference between the brain signals in trials with correct and incorrect feedback, we compute the grand average of the signals across all epochs from all five sessions for all 26 subjects as shown in Figure 10(a). We plot the results for the correct and incorrect feedback as well as their difference waveform. By convention [22] [29], we report the signals from the Cz-channel which occupies the central position on the EEG cap as shown previously in Figure 3.

Figure 10(a) shows that there is indeed a difference between the trials with correct and incorrect feedback. We also note that the waveforms show a negative deflection followed by a positive deflection. For the incorrect trials, the negative deflection is larger and appears a few milliseconds later resulting in a large difference between the trials around 0.3 - 0.4 seconds post-stimulus. We also observe an early negative deflection around 0.05 - 0.2 seconds post-stimulus which is not present in Figure 1 from Perrin et al. [22]. This can be due to differences in the measurements and choices regarding preprocessing such as baseline correction. However, since the deflection is very similar for the correct and incorrect trials, we do not expect it to cause problems in the subsequent classification into the two classes.

Furthermore, when only looking at the difference waveforms for session 1-4 compared with session 5, Figure 10(b) shows that the time and shape of the Neg-ErrP response is similar. This supports the underlying assumption of the predictive modeling pipeline in which the models trained on sessions 1-4 can be used for predicting errors and non-errors in session 5.



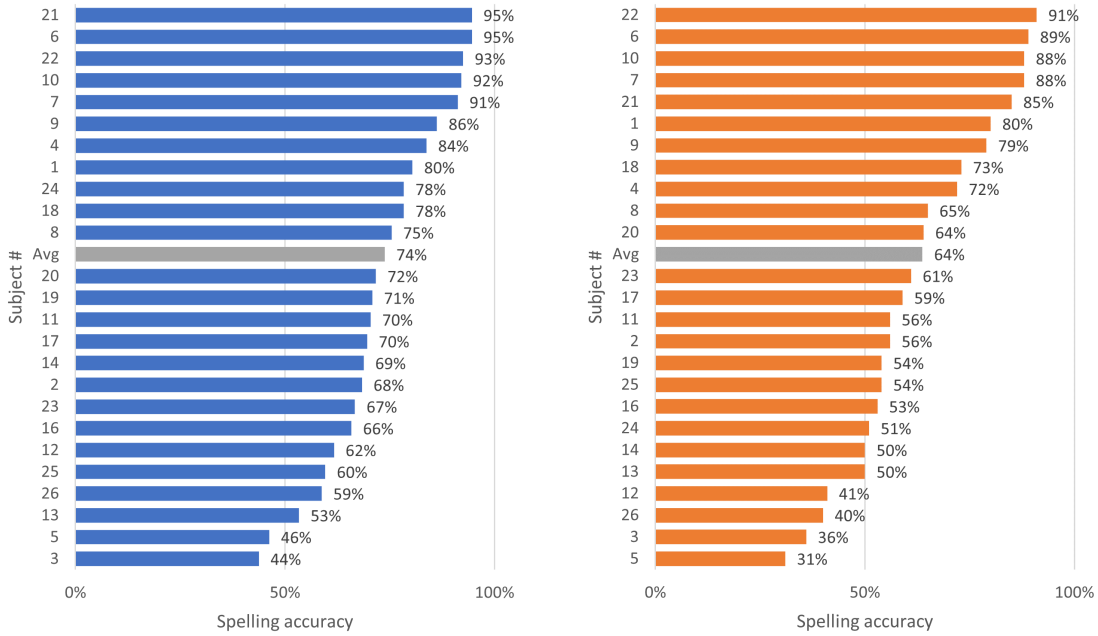

**(a)** Grand average of signals from Cz-channel associated with responses to correct and incorrect feedback as well as their difference waveform. Feedback event at ms = 0. Sessions 1-5.

**(b)** Grand average difference waveform between correct and incorrect trials. Sessions 1-4 (training set) against session 5 (test set). 150 ms - 650 ms post-feedback.

**Figure 10:** Grand average ERP responses.

According to Perrin et al. [22], spelling accuracy is correlated with specificity and could possibly also be correlated with other measures in our analysis. To investigate the distribution of spelling accuracy over subjects, we compute the spelling accuracy for all subjects for sessions 1-4 and session 5. On average, the spelling accuracy for sessions 1-4 is 74 % (with 11 subjects performing above average), while the averaged spelling accuracy for session 5 is 64% (also with 11 subjects performing above the average).

Figure 11(a) shows that the spelling accuracy varies a lot between the subjects. We note that five subjects achieve spelling accuracies above 90% in sessions 1-4. This corresponds to less than 24 trials with incorrect feedback prior to data augmentation. For subjects with this high spelling accuracy, the small number of trials with incorrect feedback could potentially lead to poor performance on error detection (and thus low sensitivity) compared to subjects with a lower spelling accuracy as the latter case will have more trials with incorrect feedback to learn from and be less biased towards trials with correct feedback. There is little variation within subjects across sessions, however Figure 15 (Appendix) shows the correlation between the spelling accuracy of sessions 1-4 and the spelling accuracy of session 5 for each subject. We find an $R^2$ score of 0.88 which shows that the spelling accuracies for the subjects are generally consistent across sessions 1-4 and session 5.




**(a)** Spelling accuracy of sessions 1-4.

**(b)** Spelling accuracy of session 5.

**Figure 11:** Spelling accuracy for all 26 subjects. Sorted in descending order. The average spelling accuracies are indicated as grey bars.

## 4.2 Predictive modeling results

Table 1 shows the average performance of the predictions from the three classification models (LDA, CNN-LSTM, UNet) and the respective standard deviations among the five folds. All of the subject-specific results are listed in Table 2 - 7 (Appendix). We observe similar performances in sensitivity between LDA (0.46) and CNN-LSTM (0.48) while UNet achieves a much lower sensitivity (0.25). Inversely, we see that for specificity, LDA (0.83) and CNN-LSTM (0.84) achieve similar performances while UNet achieves a higher specificity (0.93).

For error precision, we observe similar performances across models with CNN-LSTM (0.69) performing only slightly better than LDA (0.66) and UNet (0.67). Similarly for non-error precision, CNN-LSTM (0.74) also performs only slightly better than LDA (0.73) and UNet (0.70). It should be noted that the standard deviation of the five folds of error precision is notably higher compared to the standard deviations of the other measures indicating a larger variability of results between the folds on this measure.

The performances of the models on AUC are also similar with UNet (0.77) performing only slightly better than LDA (0.73) and CNN-LSTM (0.75). Finally, looking at the unweighted F-score, we observe that LDA (0.65) and CNN-LSTM (0.66) achieve similar performances while UNet performs notably worse (0.55), thus the lower performances of UNet on sensitivity and precision are not counterbalanced by the higher specificity.

| | Average of all subjects | | | Average st.dev. (five folds) | | |
|---|---|---|---|---|---|---|
| | LDA | CNN-LSTM | UNet | LDA | CNN-LSTM | UNet |
| Sensitivity | 0.46 | 0.48 | 0.25 | 0.06 | 0.09 | 0.05 |
| Specificity | 0.83 | 0.84 | 0.93 | 0.03 | 0.05 | 0.02 |
| Error precision | 0.66 | 0.69 | 0.67 | 0.08 | 0.09 | 0.15 |
| Non-error precision | 0.73 | 0.74 | 0.70 | 0.02 | 0.03 | 0.02 |
| AUC | 0.73 | 0.75 | 0.77 | 0.03 | 0.05 | 0.03 |
| Unweighted F-score | 0.65 | 0.66 | 0.55 | 0.04 | 0.05 | 0.04 |

**Table 1:** Average scores of the three models across all 26 subjects each with five separately trained models (five-fold cross validation). The standard deviations between the five folds for each model on each measure are reported in the three right-most columns.

Figure 12(a)-(c) shows the numbers of subjects for which the subject-specific models perform either significantly better or worse or are not significantly different from each other for each of the selected measures. Comparing LDA to CNN-LSTM, Figure 12(a) shows that among the 26 subject-specific models, only a few of the LDA models perform either significantly ($p < 0.05$) better or worse than CNN-LSTM on all six measures, thus for the majority of the subject-specific comparisons, the two models appear to achieve similar performances across the subjects.

Comparing LDA with UNet as shown in Figure 12(b), we observe that for sensitivity, 21 of the subject-specific LDA models perform statistically better ($p < 0.05$) than their UNet counterparts. On the other hand, 18 of the UNet models perform statistically better ($p < 0.05$) than LDA when measured on specificity with only one LDA model performing statistically better ($p < 0.05$) than a UNet model on this measure. Similarly, we note that 10 of the UNet models achieve significantly better ($p < 0.05$) performance on error precision compared to the LDA models while 15 of the LDA models on the other hand perform significantly better ($p < 0.05$) than UNet on non-error precision. For both error and non-error precision, however, there is also notable numbers of pairs where the models perform equally well. Finally, we see that the F-scores, which is a combined measure based on both the two recall and precision measures, are generally significantly better ($p < 0.05$) for LDA compared to UNet.

Similar results can be observed in Figure 12(c) which compares CNN-LSTM against UNet. Here, 19 CNN-LSTM models perform significantly better ($p < 0.05$) than UNet on sensitivity while 18 UNet models perform significantly better ($p < 0.05$) than CNN-LSTM on specificity. As before, we can observe a bias of the UNet to generally perform better on error precision but worse on non-error precision - with the caveat that for many subjects, the precision measures are not significantly different. We observe that 15 of the CNN-LSTM models perform significantly better than their UNet counterparts whereas no UNet model performs better than its CNN-LSTM counterpart on this measure.

These results support the idea of a trade-off between sensitivity and specificity for which the UNet architecture appears to focus on optimizing predictions on non-errors (specificity) at the cost of worsened prediction on errors (sensitivity) compared to the other two models.


**(a)** LDA vs. CNN-LSTM. LDA significantly better (blue), no significant difference (grey) or CNN-LSTM significantly better (green).

**(b)** LDA vs. UNet. LDA significantly better (blue), no significant difference (grey) or UNet significantly better (orange).

**(c)** CNN-LSTM vs. UNet. CNN-LSTM significantly better (green), no significant difference (grey) or UNet significantly better (orange).

**Figure 12:** Comparison of the performances of the models. The models are compared pairwise resulting in three combinations: **(a)** LDA vs. CNN-LSTM, **(b)** LDA vs. UNet and **(c)** CNN-LSTM vs. UNet. The 26 subject-specific models for each of the model types in the combination are compared on all six measures. The figures show how many of the subject-specific models perform significantly better than their counterparts and how many of the pairs do not differ significantly in their performances. No difference is marked with grey whereas the number of LDA models performing better are marked with blue, the number of CNN-LSTM models performing better are marked with green and the number of UNet models performing better are marked with orange.

# 5 Discussion and conclusions

## 5.1 Contributions

In this thesis, we have demonstrated how EEG recordings can be used for the detection of errors and non-errors in BCI systems. We have performed multiple preprocessing steps, extracted and selected relevant features, and employed data manipulation techniques such as data augmentation using window slicing before passing the recordings as input to classification models. Using two distinct neural network architectures and a state-of-the-art conventional model, we have shown that all three models can be used for error detection in BCIs although they show significant differences in performance on some measures and appear to employ different optimization strategies to improve performance. Our findings show that neural network architectures do not necessarily perform better than simpler state-of-the-art conventional methods, in our thesis represented by a Linear Discriminant Analysis (LDA) model despite their increased complexity and ability to recognize complex and potentially useful relationships between features. This suggests that more work remains in order for neural networks to consistently match and exceed the performance of state-of-the-art conventional methods for the task of error detection.

## 5.2 Empirical findings

With our second research questions, we set out to compare the performances of two neural network architectures (CNN-LSTM and UNet) with that of a state-of-the-art conventional model (LDA) which we considered our baseline, asking whether the neural networks could improve the performance on error detection.

Our results show that CNN-LSTM overall performed on par with the baseline performance of

LDA. For the majority of the subjects (between 22 and 25 out of 26), we observed no significant differences ($p < 0.05$) between the performances of LDA and CNN-LSTM on any of the selected measures. Thus, we find these two models to perform equally well on error detection in our selected BCI setup. While we found the performances of LDA and CNN-LSTM to be similar, it should be noted that neural networks in general have a much higher number of parameters to optimize and require extensive computational resources to train. Thus, it can be argued that the simpler and potentially more explainable model, LDA, is preferred over the more complex model, CNN-LSTM, given that they achieve similar performances.

Our results show that the second selected neural network architecture, UNet, performs differently than both LDA and CNN-LSTM on a number of measures. Overall, UNet appears to have a stronger bias towards predicting non-errors which is the majority label for all but a few subjects. The stronger bias results in a significantly higher ($p < 0.05$) specificity (non-error recall) for 18 out of 26 of the subject-specific models both when compared to LDA and CNN-LSTM as very few non-errors are misclassified as errors (False Positives). However, the results show that the increase in specificity comes at a cost of a decrease in sensitivity as well as a significantly lower non-error precision compared to both LDA and CNN-LSTM as a larger amount of the errors are misclassified as non-errors (False Negatives).

Comparing the unweighted F-scores between UNet and the other two models, we observe that the majority (16 and 15 out of 26, respectively) of the subject-specific models for LDA and CNN-LSTM achieve significantly higher ($p < 0.05$) F-scores than their UNet counterparts. This shows that, when evaluated on the combined performance for error and non-error detection, UNet performs worse than the other two models for most subjects. This shows that the observed decrease in sensitivity and non-error precision is not offset by a correspondingly large increase in specificity.

While the lower unweighted F-score of UNet suggests an overall inferior performance on error detection, it is important to note that UNet, besides the superior performance on specificity, performs better on error precision for ten out of 26 subjects compared to LDA and six out of 26 subjects compared to CNN-LSTM ($p < 0.05$). Thus, we cannot ultimately conclude whether UNet performs better or worse compared to the other two models, yet we have shown that it performs *differently*.

As it has been pointed out earlier in the thesis, in the case of the P300 speller, both Perrin et al. [22] and Spüler et al. [29] consider a low specificity the "worst-case" as it means misclassifying and hence wrongly correcting an actually correct letter. On this measure, the UNet performs significantly better than both LDA and CNN-LSTM. In conclusion, when selecting the best model for error detection in BCI, our findings suggest that the decision must be viewed in the context of the specific use case and the measures that it is consequently considered most important to optimize. If maximizing the probability of correct predictions on non-errors is considered most important and a relatively higher number of errors misclassified as non-errors (False Negatives) can be tolerated, the UNet architecture could prove to be most useful, while in the cases in which a more balanced focus on the detection of both errors and non-errors is prioritized, LDA or CNN-LSTM might prove to be better choices.

## 5.3 Limitations

The findings of this thesis should be considered alongside several important limitations which can have had an influence on the results. The limitations relate both to the data which we have used and to the applied methodological choices along with unanswered questions.

The origin of the used data is from an external source. Therefore, the data is affected by all the limitations and potential pitfalls caused by using an external dataset without being able to derive thorough and exact knowledge about the data generation process. As mentioned in Section 3.1, the data which we are using in this thesis comes from at least two different experiments: One set of data comes from an experiment done by Perrin et al. [22] and the other set comes from an unknown source (potentially several unknown sources). All of the data is reported to be collected in the same way[5]. However, due to the noisy nature of EEG experiments, even if the framework of the experiment is the same, other changes in the experimental setup such as the wiring of the EEG electrodes and the instruction of the subjects might result in potential artifacts. We are not able to specify or correct these as we are able unable to confidently classify data points as belonging to either Perrin et al. [22] or some other source.

In our work, we have employed a range of preprocessing methods to the initial raw EEG data. Our pipeline follows the methodology [22] to the extent that we were able to decode it from the descriptions. The preprocessing pipeline causes some significant changes to the initial data through a series of transformations, manipulations, feature extractions, and selections in order to increase the signal-to-noise ratio (SNR) of the otherwise noisy raw EEG recordings. In the scope of this thesis, we have not explored the consequences of these transformations or fine-tuned the preprocessing steps to improve performance. It is likely that different choices of filters, channel selections, or methods for data augmentation as well as the associated hyper-parameters could change how the data is transformed and thereby also the final predictive modeling results.

During the process of working with the data, we have observed a significant imbalance between the two classes (errors and non-errors) among some of the subjects who in the most extreme cases had less than 10 % (24 samples) trials with incorrect feedback prior to data augmentation. The high imbalance poses a challenge in training. Spüler et al. [29] recommends a minimum of 50 trials with incorrect feedback in order for the model to be able to reliably predict this class. A high imbalance during training can lead to a bias towards the majority label which transfers to the following prediction task. We observed this in all three models. Despite the addition of more error samples through data augmentation[6], they perform better at identifying the majority class (non-errors) than the minority class (errors). This is also reflected in the performance of the different measures. For all three models, specificity (non-error recall) is much higher than sensitivity (error recall) and non-error precision is slightly higher than error precision.

Another challenge imposed by the noisy and volatile nature of EEG recordings is transferring features and knowledge of the classification models across domains. In a BCI context, this could mean across subjects, across sessions for the same subject, or across different technical setups. In our predictive modeling setup, we train a model for each of the subjects. For each subject-specific model, we allocate sessions 1-4 for training and session 5 for testing. This division relies on the assumption that the data in session 5 follows the same probability distribution as the training data from sessions 1-4. As EEG recordings are sensitive to the mental state of the subject, which is naturally expected to vary throughout the course of the sessions or change on a day-to-day basis, this assumption is not always met. An example is fatigue and familiarity with the task which is expected to increase over the sessions and attention which in turn is expected to decrease [22] [29]. It raises the question of whether a model that has been trained on data from one subject at one point in time can be applied on data from another subject or even on data from the same subject

[5]BCI Challenge @ NER 2015 [1]

[6]More non-errors samples are also generated and added as described in Section 3.2. The distribution of the two classes remains the same.

but collected at a future point in time.

A limitation of neural networks is that a large number of training samples is usually needed to tune these models well and obtain superior performance over the existing state-of-the-art classification models [16] due to their large number of parameters. In BCI experiments, this is often not possible. The data used in this thesis consists of 340 recorded trials for each of the 26 subjects. After applying data augmentation and splitting the data into training, validation and test sets, we have 960 trials[7] for each subject-specific neural network models to train on. This may be too few for training and optimizing these complex neural network architectures. Both the CNN-LSTM and UNet would likely benefit from more training samples.

The number of hyper-parameters to tune rises with the complexity of the chosen neural network architecture. In much of the literature, the chosen neural architecture and tuning of hyper-parameters such as number and size of hidden layers, optimizer, and loss function is not justified [16] and sometimes not reported in detail. This leaves the implementation of neural networks as a black box. In our thesis, we have not investigated tuning the deep neural network models extensively. Instead, we have chosen to re-implement the CNN-LSTM and UNet models based on the descriptions in the original articles by Ronneberger et al. [25] and Xu et al. [36]. Without the fine-tuning to the specifics of our data, it is likely that highly complex models such as CNN-LSTM or UNet will achieve worse performance compared to a simpler model [16] such as a 2-layered CNN model as suggested by Craik et al. [5]. Therefore, a case could be made that the reported results for both CNN-LSTM and UNet could likely be improved with tuning. The task is further complicated by the subject-specific models in the predictive setup as a set of parameters that works well for one subject might results in inferior performance when applied to data from other subjects.

## 5.4 Implications

In our thesis, we have shown that EEG recordings can be used in a BCI system to detect erroneous feedback. We reach this conclusion with a small number of samples per subject-specific model by employing a series of preprocessing steps and data augmentation.

Our findings suggest that even for an out-of-box neural network model without any fine-tuning, the models perform on par with conventional methods yet the our implementations of the neural network models did not improve upon state-of-the-art conventional methods used in EEG research. Our findings support the idea that neural networks show promise within the fields of EEG and BCI research and contribute to their promising results across a wide range of scientific disciplines. However, it should be emphasized that our findings also reveal that neural networks that perform well on one task have limitations when applied on other tasks. Our findings suggest that they require careful considerations regarding the choice of architecture and extensive tuning of hyper-parameters before they can exceed the current conventional methods within EEG and BCI research.

In addition, our findings reveal a performance trade-off between error detection and non-error detection. It remains subject to future research why this trade-off happens and whether it should be attributed to elements of the training process such as over- or under-fitting, the preprocessing steps, the hyper-parameters used in the implementation of the models or something innate in the model architectures which results in the different neural network architectures choosing different optimization strategies and thereby prioritizing performance on certain measures higher than others.

[7] Trials $\times$ number of augmentations $\times$ training set size $= 240 \times 5 \times 0.8 = 960$ trials

## 5.5 Future work

Overcoming the challenges of inter- and intra-subject variability in EEG recordings remains a major challenge and topic of future research [8, 16, 18]. It remains an open question whether the subject-specific models could be used for detecting errors and non-errors in a cross-domain context. Furthermore, it could be interesting to investigate how the data could be used in a transfer learning context in which one could exploit all the available data from all subjects to make a pre-trained model which could be followed by a subject-specific fine-tuning in order to improve the performance on the specific domain. This could solve both the long calibration time and the small amount of available data which still today pose major challenges in EEG research [16].

In our work, we have observed that the two different neural network architectures exhibit biases towards optimizing certain measures. As it has already been mentioned, the cause might be contributed to the specific neural network architecture itself. However, it remains a subject for future research to investigate if certain architectures show certain class preferences when faced with imbalanced data and how changes in the model architecture may shift these preferences.

Our work also demonstrates that the initial raw EEG data require a considerable amount of preprocessing, feature extraction and feature selection before they can be used as input to a classification model. It can be argued that an intrinsic ability and advantage of neural network models is to find the most significant patterns in the input data and thus that they can perform meaningful feature extraction and artifact removal between the hidden layers which would reduce or entirely remove the need for some of the initial data preprocessing steps. As mentioned by Craik et al. [5], it remains an unanswered question in EEG research whether a CNN-LSTM model (and potentially other neural network architectures) can retain the performance measures if provided with raw unprocessed data as input compared to the current methodical preprocessing of EEG signals.

## 5.6 Conclusion

The main goals of this thesis were to investigate how EEG signals can be used for detection of errors in BCI systems and to compare the classification performance on error detection between different models. We have demonstrated how EEG recordings are preprocessed in order to improve the signal-to-noise ratio and how features are extracted and used as input in different classification models. We achieved this by following existing strategies for preprocessing of EEG signals and implementation of machine learning pipelines.
Furthermore, we have shown that neural network architectures can match a conventional classification method (LDA) in the field of EEG research. However, our findings suggest that further work remains in order to improve the performance of these architectures.

# 6 Appendix

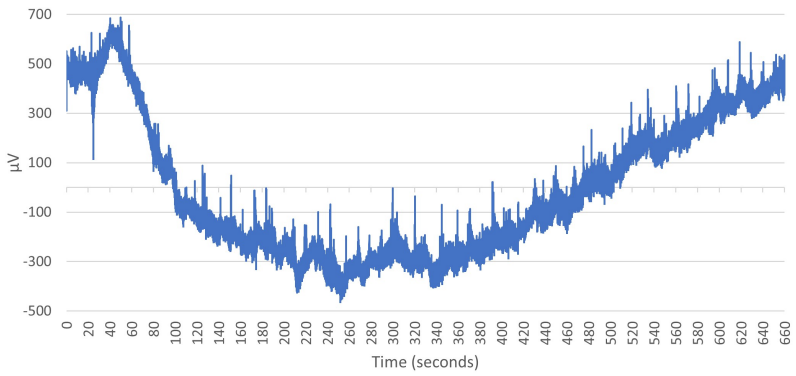



**Figure 13:** Example: Randomly chosen (subject 2, session 1). Plot of raw Cz recordings of the entire session (660 seconds duration).

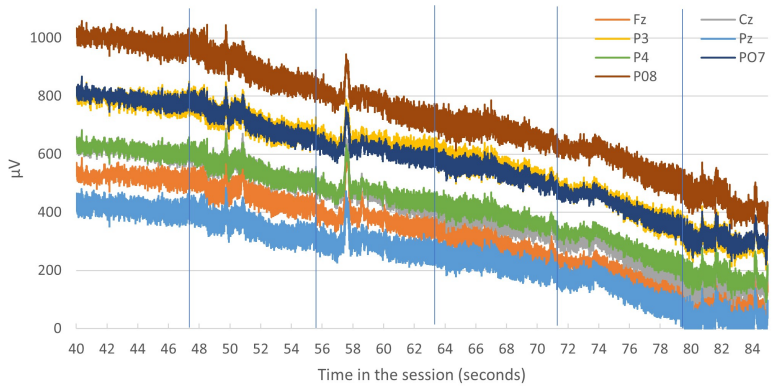



**Figure 14:** Example: Randomly chosen (subject 2, session 1, 1st word). Raw signals measured on 7 channels (Fz, Cz, P3, Pz, P4, PO7 and P08). The blue vertical lines mark the start of each epoch (corresponding to the appearance of each of the five letters of the word).

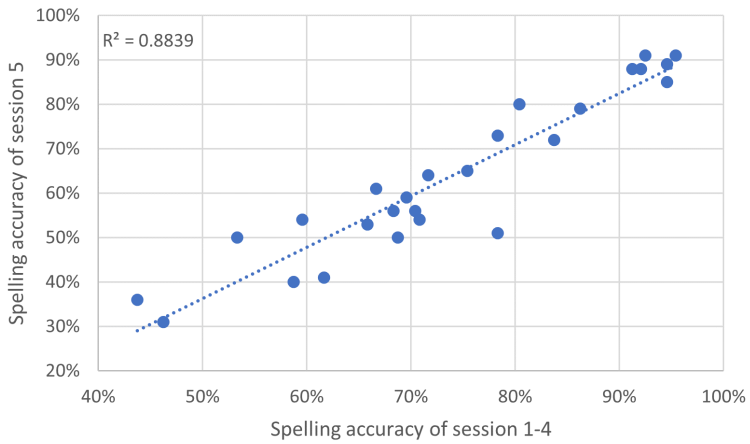



**Figure 15:** Spelling accuracy of sessions 1-4 vs. session 5 and the $R^2$ correlation score

| | Average of 5-fold cross validation | | | St.dev of 5-fold cross validation | | | P-value | | |
|---|---|---|---|---|---|---|---|---|---|
| | LDA | CNN-LSTM | UNet | LDA | CNN-LSTM | UNet | LDA vs CNN-LSTM | LDA vs UNet | UNet vs CNN-LSTM |
| Subject 1 | 0.59 | 0.64 | 0.38 | 0.043 | 0.077 | 0.087 | 0.210 | 0.023 | 0.031 |
| Subject 2 | 0.11 | 0.15 | 0.00 | 0.076 | 0.081 | 0.000 | 0.977 | 0.036 | 0.010 |
| Subject 3 | 0.62 | 0.63 | 0.15 | 0.151 | 0.095 | 0.091 | 1.000 | 0.016 | 0.016 |
| Subject 4 | 0.48 | 0.36 | 0.14 | 0.053 | 0.124 | 0.030 | 0.138 | 0.018 | 0.041 |
| Subject 5 | 0.47 | 0.55 | 0.39 | 0.047 | 0.097 | 0.049 | 0.369 | 0.087 | 0.031 |
| Subject 6 | 0.68 | 0.70 | 0.69 | 0.042 | 0.058 | 0.094 | 0.588 | 0.885 | 1.000 |
| Subject 7 | 0.48 | 0.46 | 0.36 | 0.036 | 0.142 | 0.038 | 1.000 | 0.018 | 0.214 |
| Subject 8 | 0.50 | 0.48 | 0.17 | 0.059 | 0.102 | 0.049 | 1.000 | 0.018 | 0.018 |
| Subject 9 | 0.59 | 0.60 | 0.40 | 0.065 | 0.056 | 0.061 | 1.000 | 0.017 | 0.018 |
| Subject 10 | 0.33 | 0.37 | 0.04 | 0.071 | 0.108 | 0.017 | 0.894 | 0.014 | 0.015 |
| Subject 11 | 0.40 | 0.48 | 0.03 | 0.064 | 0.174 | 0.023 | 0.372 | 0.018 | 0.017 |
| Subject 12 | 0.36 | 0.24 | 0.01 | 0.060 | 0.101 | 0.030 | 0.124 | 0.013 | 0.014 |
| Subject 13 | 0.16 | 0.29 | 0.00 | 0.099 | 0.061 | 0.000 | 0.086 | 0.008 | 0.010 |
| Subject 14 | 0.59 | 0.60 | 0.19 | 0.084 | 0.047 | 0.059 | 1.000 | 0.018 | 0.018 |
| Subject 15 | 0.36 | 0.40 | 0.09 | 0.023 | 0.069 | 0.037 | 0.431 | 0.018 | 0.018 |
| Subject 16 | 0.43 | 0.45 | 0.25 | 0.025 | 0.062 | 0.078 | 1.000 | 0.018 | 0.018 |
| Subject 17 | 0.21 | 0.25 | 0.01 | 0.055 | 0.127 | 0.022 | 1.000 | 0.013 | 0.015 |
| Subject 18 | 0.68 | 0.79 | 0.88 | 0.072 | 0.061 | 0.036 | 0.069 | 0.018 | 0.054 |
| Subject 19 | 0.31 | 0.56 | 0.10 | 0.090 | 0.187 | 0.069 | 0.051 | 0.031 | 0.018 |
| Subject 20 | 0.79 | 0.83 | 0.85 | 0.036 | 0.052 | 0.017 | 0.496 | 0.023 | 0.897 |
| Subject 21 | 0.53 | 0.53 | 0.32 | 0.048 | 0.144 | 0.079 | 1.000 | 0.017 | 0.089 |
| Subject 22 | 0.66 | 0.74 | 0.43 | 0.021 | 0.054 | 0.048 | 0.038 | 0.014 | 0.017 |
| Subject 23 | 0.35 | 0.17 | 0.03 | 0.037 | 0.118 | 0.046 | 0.013 | 0.013 | 0.145 |
| Subject 24 | 0.56 | 0.49 | 0.13 | 0.136 | 0.099 | 0.145 | 0.469 | 0.023 | 0.022 |
| Subject 25 | 0.30 | 0.34 | 0.08 | 0.091 | 0.062 | 0.050 | 0.897 | 0.018 | 0.018 |
| Subject 26 | 0.39 | 0.38 | 0.31 | 0.089 | 0.039 | 0.086 | 1.000 | 0.260 | 0.214 |
| Means | 0.46 | 0.48 | 0.25 | 0.064 | 0.092 | 0.052 | 0.672 | 0.064 | 0.115 |

**Table 2:** Subject-specific results of **sensitivity/error recall**. Prediction on test set based on model training on 5-fold training set. P-values are based on Mann-Whitney U test and Bonferroni-corrected.

| | Average of 5-fold cross validation | | | St.dev of 5-fold cross validation | | | P-value | | |
|---|---|---|---|---|---|---|---|---|---|
| | LDA | CNN-LSTM | UNet | LDA | CNN-LSTM | UNet | LDA vs CNN-LSTM | LDA vs UNet | UNet vs CNN-LSTM |
| Subject 1 | 0.93 | 0.87 | 0.99 | 0.020 | 0.053 | 0.008 | 0.028 | 0.013 | 0.014 |
| Subject 2 | 0.96 | 0.98 | 1.00 | 0.023 | 0.030 | 0.000 | 0.365 | 0.011 | 0.270 |
| Subject 3 | 0.95 | 0.97 | 1.00 | 0.027 | 0.012 | 0.000 | 0.756 | 0.011 | 0.010 |
| Subject 4 | 0.75 | 0.83 | 0.98 | 0.033 | 0.072 | 0.016 | 0.133 | 0.016 | 0.016 |
| Subject 5 | 0.80 | 0.76 | 0.96 | 0.037 | 0.105 | 0.027 | 0.791 | 0.016 | 0.016 |
| Subject 6 | 0.78 | 0.72 | 0.75 | 0.071 | 0.061 | 0.041 | 0.374 | 0.695 | 0.789 |
| Subject 7 | 0.86 | 0.86 | 0.94 | 0.017 | 0.100 | 0.026 | 1.000 | 0.016 | 0.365 |
| Subject 8 | 0.68 | 0.67 | 0.93 | 0.052 | 0.083 | 0.041 | 1.000 | 0.018 | 0.018 |
| Subject 9 | 0.79 | 0.83 | 0.95 | 0.015 | 0.025 | 0.014 | 0.038 | 0.016 | 0.017 |
| Subject 10 | 0.87 | 0.91 | 0.99 | 0.020 | 0.036 | 0.008 | 0.171 | 0.016 | 0.016 |
| Subject 11 | 0.81 | 0.76 | 0.98 | 0.016 | 0.062 | 0.016 | 0.514 | 0.017 | 0.018 |
| Subject 12 | 1.00 | 0.96 | 1.00 | 0.006 | 0.010 | 0.000 | 0.016 | 0.266 | 0.011 |
| Subject 13 | 0.99 | 1.00 | 1.00 | 0.009 | 0.005 | 0.000 | 0.348 | 0.106 | 0.636 |
| Subject 14 | 0.70 | 0.73 | 0.95 | 0.030 | 0.063 | 0.024 | 0.791 | 0.018 | 0.018 |
| Subject 15 | 0.84 | 0.89 | 1.00 | 0.029 | 0.016 | 0.009 | 0.050 | 0.014 | 0.014 |
| Subject 16 | 0.69 | 0.76 | 0.90 | 0.058 | 0.065 | 0.056 | 0.248 | 0.017 | 0.029 |
| Subject 17 | 0.87 | 0.86 | 1.00 | 0.042 | 0.050 | 0.007 | 1.000 | 0.016 | 0.016 |
| Subject 18 | 0.56 | 0.48 | 0.47 | 0.034 | 0.123 | 0.050 | 0.433 | 0.040 | 1.000 |
| Subject 19 | 0.94 | 0.93 | 1.00 | 0.008 | 0.033 | 0.000 | 0.781 | 0.010 | 0.011 |
| Subject 20 | 0.66 | 0.66 | 0.63 | 0.059 | 0.087 | 0.074 | 1.000 | 0.787 | 0.689 |
| Subject 21 | 0.90 | 0.92 | 0.99 | 0.035 | 0.032 | 0.008 | 0.897 | 0.016 | 0.016 |
| Subject 22 | 0.98 | 0.97 | 0.99 | 0.011 | 0.023 | 0.000 | 0.651 | 0.270 | 0.108 |
| Subject 23 | 0.97 | 0.99 | 1.00 | 0.016 | 0.010 | 0.000 | 0.080 | 0.010 | 0.036 |
| Subject 24 | 0.99 | 0.99 | 1.00 | 0.011 | 0.009 | 0.000 | 1.000 | 0.106 | 0.036 |
| Subject 25 | 0.82 | 0.78 | 0.97 | 0.031 | 0.061 | 0.021 | 0.433 | 0.017 | 0.018 |
| Subject 26 | 0.64 | 0.74 | 0.75 | 0.031 | 0.036 | 0.059 | 0.018 | 0.068 | 0.687 |
| Means | 0.83 | 0.84 | 0.93 | 0.029 | 0.049 | 0.019 | 0.559 | 0.100 | 0.202 |

**Table 3:** Subject-specific results of **specificity/non-error recall**. Prediction on test set based on model training on 5-fold training set. P-values are based on Mann-Whitney U test and Bonferroni-corrected.

| | Average of 5-fold cross validation | | | St.dev of 5-fold cross validation | | | P-value | | |
|---|---|---|---|---|---|---|---|---|---|
| | LDA | CNN-LSTM | UNet | LDA | CNN-LSTM | UNet | LDA vs CNN-LSTM | LDA vs UNet | UNet vs CNN-LSTM |
| Subject 1 | 0.87 | 0.80 | 0.96 | 0.038 | 0.057 | 0.026 | 0.055 | 0.018 | 0.018 |
| Subject 2 | 0.25 | 0.75 | 0.00 | 0.186 | 0.348 | 0.000 | 0.066 | 0.038 | 0.010 |
| Subject 3 | 0.65 | 0.74 | 0.80 | 0.126 | 0.089 | 0.447 | 1.000 | 0.195 | 0.197 |
| Subject 4 | 0.60 | 0.62 | 0.82 | 0.049 | 0.140 | 0.124 | 1.000 | 0.018 | 0.089 |
| Subject 5 | 0.77 | 0.78 | 0.93 | 0.033 | 0.059 | 0.039 | 1.000 | 0.018 | 0.018 |
| Subject 6 | 0.76 | 0.72 | 0.73 | 0.061 | 0.058 | 0.027 | 0.796 | 0.605 | 1.000 |
| Subject 7 | 0.77 | 0.79 | 0.87 | 0.009 | 0.119 | 0.044 | 1.000 | 0.018 | 0.605 |
| Subject 8 | 0.58 | 0.57 | 0.70 | 0.040 | 0.038 | 0.126 | 1.000 | 0.139 | 0.090 |
| Subject 9 | 0.66 | 0.72 | 0.85 | 0.023 | 0.038 | 0.036 | 0.054 | 0.018 | 0.018 |
| Subject 10 | 0.49 | 0.61 | 0.83 | 0.052 | 0.062 | 0.236 | 0.041 | 0.039 | 0.298 |
| Subject 11 | 0.54 | 0.52 | 0.50 | 0.043 | 0.057 | 0.373 | 0.796 | 1.000 | 1.000 |
| Subject 12 | 0.94 | 0.52 | 0.20 | 0.085 | 0.088 | 0.447 | 0.015 | 0.087 | 0.190 |
| Subject 13 | 0.72 | 0.95 | 0.00 | 0.298 | 0.112 | 0.000 | 0.291 | 0.011 | 0.008 |
| Subject 14 | 0.55 | 0.59 | 0.71 | 0.047 | 0.066 | 0.096 | 0.796 | 0.055 | 0.090 |
| Subject 15 | 0.68 | 0.77 | 0.97 | 0.048 | 0.016 | 0.064 | 0.032 | 0.014 | 0.015 |
| Subject 16 | 0.67 | 0.74 | 0.80 | 0.049 | 0.052 | 0.060 | 0.141 | 0.032 | 0.216 |
| Subject 17 | 0.29 | 0.31 | 0.10 | 0.055 | 0.131 | 0.224 | 1.000 | 0.195 | 0.157 |
| Subject 18 | 0.73 | 0.73 | 0.75 | 0.016 | 0.043 | 0.015 | 1.000 | 0.090 | 1.000 |
| Subject 19 | 0.64 | 0.75 | 1.00 | 0.087 | 0.079 | 0.000 | 0.139 | 0.011 | 0.011 |
| Subject 20 | 0.84 | 0.85 | 0.83 | 0.021 | 0.029 | 0.028 | 1.000 | 1.000 | 0.796 |
| Subject 21 | 0.75 | 0.78 | 0.96 | 0.070 | 0.043 | 0.050 | 1.000 | 0.016 | 0.016 |
| Subject 22 | 0.90 | 0.87 | 0.90 | 0.049 | 0.078 | 0.010 | 1.000 | 1.000 | 1.000 |
| Subject 23 | 0.60 | 0.62 | 0.40 | 0.142 | 0.375 | 0.548 | 0.594 | 1.000 | 1.000 |
| Subject 24 | 0.84 | 0.79 | 0.60 | 0.171 | 0.148 | 0.548 | 1.000 | 1.000 | 1.000 |
| Subject 25 | 0.58 | 0.58 | 0.58 | 0.100 | 0.083 | 0.333 | 0.794 | 0.305 | 0.442 |
| Subject 26 | 0.47 | 0.55 | 0.51 | 0.061 | 0.038 | 0.023 | 0.090 | 0.603 | 0.141 |
| Means | 0.66 | 0.69 | 0.67 | 0.075 | 0.094 | 0.151 | 0.711 | 0.328 | 0.424 |

**Table 4:** Subject-specific results of **error precision**. Prediction on test set based on model training on 5-fold training set. P-values are based on Mann-Whitney U test and Bonferroni-corrected.

| | Average of 5-fold cross validation | | | St.dev of 5-fold cross validation | | | P-value | | |
|---|---|---|---|---|---|---|---|---|---|
| | LDA | CNN-LSTM | UNet | LDA | CNN-LSTM | UNet | LDA vs CNN-LSTM | LDA vs UNet | UNet vs CNN-LSTM |
| Subject 1 | 0.74 | 0.75 | 0.67 | 0.021 | 0.034 | 0.032 | 0.605 | 0.032 | 0.032 |
| Subject 2 | 0.90 | 0.90 | 0.89 | 0.007 | 0.006 | 0.000 | 0.307 | 0.178 | 0.010 |
| Subject 3 | 0.95 | 0.95 | 0.90 | 0.019 | 0.012 | 0.010 | 1.000 | 0.017 | 0.017 |
| Subject 4 | 0.65 | 0.63 | 0.59 | 0.030 | 0.047 | 0.010 | 0.796 | 0.018 | 0.214 |
| Subject 5 | 0.51 | 0.54 | 0.52 | 0.023 | 0.033 | 0.019 | 0.315 | 0.603 | 0.442 |
| Subject 6 | 0.71 | 0.71 | 0.71 | 0.036 | 0.052 | 0.056 | 0.796 | 1.000 | 0.796 |
| Subject 7 | 0.62 | 0.62 | 0.59 | 0.012 | 0.052 | 0.012 | 0.603 | 0.018 | 1.000 |
| Subject 8 | 0.60 | 0.60 | 0.56 | 0.027 | 0.030 | 0.014 | 1.000 | 0.055 | 0.090 |
| Subject 9 | 0.74 | 0.75 | 0.70 | 0.030 | 0.028 | 0.021 | 0.313 | 0.089 | 0.032 |
| Subject 10 | 0.78 | 0.80 | 0.74 | 0.018 | 0.023 | 0.003 | 0.216 | 0.017 | 0.017 |
| Subject 11 | 0.71 | 0.73 | 0.64 | 0.023 | 0.054 | 0.004 | 0.605 | 0.018 | 0.018 |
| Subject 12 | 0.90 | 0.88 | 0.85 | 0.008 | 0.014 | 0.004 | 0.054 | 0.014 | 0.015 |
| Subject 13 | 0.92 | 0.93 | 0.91 | 0.008 | 0.005 | 0.000 | 0.067 | 0.011 | 0.011 |
| Subject 14 | 0.73 | 0.74 | 0.65 | 0.044 | 0.032 | 0.015 | 0.901 | 0.018 | 0.018 |
| Subject 15 | 0.58 | 0.61 | 0.53 | 0.015 | 0.025 | 0.009 | 0.055 | 0.018 | 0.018 |
| Subject 16 | 0.44 | 0.48 | 0.44 | 0.028 | 0.030 | 0.014 | 0.141 | 1.000 | 0.142 |
| Subject 17 | 0.81 | 0.82 | 0.80 | 0.008 | 0.024 | 0.003 | 0.794 | 0.060 | 0.208 |
| Subject 18 | 0.49 | 0.57 | 0.70 | 0.049 | 0.080 | 0.054 | 0.216 | 0.018 | 0.055 |
| Subject 19 | 0.78 | 0.85 | 0.74 | 0.023 | 0.055 | 0.015 | 0.032 | 0.031 | 0.018 |
| Subject 20 | 0.59 | 0.64 | 0.65 | 0.034 | 0.063 | 0.040 | 0.315 | 0.142 | 1.000 |
| Subject 21 | 0.78 | 0.79 | 0.73 | 0.020 | 0.046 | 0.024 | 1.000 | 0.032 | 0.141 |
| Subject 22 | 0.91 | 0.93 | 0.87 | 0.005 | 0.012 | 0.010 | 0.030 | 0.017 | 0.018 |
| Subject 23 | 0.92 | 0.90 | 0.88 | 0.005 | 0.013 | 0.005 | 0.016 | 0.015 | 0.296 |
| Subject 24 | 0.96 | 0.95 | 0.92 | 0.013 | 0.009 | 0.012 | 0.369 | 0.032 | 0.032 |
| Subject 25 | 0.58 | 0.58 | 0.55 | 0.030 | 0.026 | 0.009 | 1.000 | 0.216 | 0.142 |
| Subject 26 | 0.55 | 0.58 | 0.56 | 0.041 | 0.017 | 0.015 | 0.519 | 0.796 | 0.090 |
| Means | 0.73 | 0.74 | 0.70 | 0.022 | 0.032 | 0.016 | 0.503 | 0.177 | 0.207 |

**Table 5:** Subject-specific results of **non-error precision**. Prediction on test set based on model training on 5-fold training set. P-values are based on Mann-Whitney U test and Bonferroni-corrected.

| | Average of 5-fold cross validation | | | St.dev of 5-fold cross validation | | | P-value | | |
|---|---|---|---|---|---|---|---|---|---|
| | LDA | CNN-LSTM | UNet | LDA | CNN-LSTM | UNet | LDA vs CNN-LSTM | LDA vs UNet | UNet vs CNN-LSTM |
| Subject 1 | 0.87 | 0.84 | 0.86 | 0.025 | 0.039 | 0.015 | 0.605 | 0.796 | 1.000 |
| Subject 2 | 0.67 | 0.65 | 0.71 | 0.036 | 0.133 | 0.189 | 1.000 | 0.216 | 0.444 |
| Subject 3 | 0.91 | 0.91 | 0.91 | 0.018 | 0.020 | 0.029 | 1.000 | 0.796 | 0.796 |
| Subject 4 | 0.65 | 0.65 | 0.70 | 0.056 | 0.045 | 0.029 | 1.000 | 0.260 | 0.216 |
| Subject 5 | 0.68 | 0.70 | 0.77 | 0.020 | 0.024 | 0.031 | 0.216 | 0.018 | 0.032 |
| Subject 6 | 0.81 | 0.80 | 0.79 | 0.041 | 0.033 | 0.021 | 1.000 | 0.796 | 1.000 |
| Subject 7 | 0.74 | 0.74 | 0.75 | 0.017 | 0.053 | 0.023 | 1.000 | 0.605 | 0.605 |
| Subject 8 | 0.62 | 0.59 | 0.56 | 0.039 | 0.058 | 0.044 | 0.605 | 0.142 | 1.000 |
| Subject 9 | 0.78 | 0.80 | 0.85 | 0.031 | 0.027 | 0.018 | 0.605 | 0.032 | 0.032 |
| Subject 10 | 0.64 | 0.72 | 0.75 | 0.021 | 0.015 | 0.039 | 0.018 | 0.018 | 0.315 |
| Subject 11 | 0.70 | 0.71 | 0.68 | 0.011 | 0.036 | 0.038 | 1.000 | 0.901 | 0.796 |
| Subject 12 | 0.88 | 0.86 | 0.89 | 0.032 | 0.054 | 0.014 | 1.000 | 1.000 | 0.605 |
| Subject 13 | 0.78 | 0.78 | 0.83 | 0.048 | 0.094 | 0.039 | 1.000 | 0.315 | 0.695 |
| Subject 14 | 0.72 | 0.69 | 0.66 | 0.020 | 0.034 | 0.027 | 0.216 | 0.018 | 0.796 |
| Subject 15 | 0.68 | 0.73 | 0.70 | 0.027 | 0.041 | 0.028 | 0.216 | 0.444 | 0.796 |
| Subject 16 | 0.60 | 0.60 | 0.59 | 0.045 | 0.046 | 0.053 | 1.000 | 1.000 | 1.000 |
| Subject 17 | 0.50 | 0.56 | 0.54 | 0.038 | 0.073 | 0.023 | 0.315 | 0.142 | 1.000 |
| Subject 18 | 0.66 | 0.73 | 0.74 | 0.028 | 0.048 | 0.019 | 0.090 | 0.018 | 1.000 |
| Subject 19 | 0.80 | 0.87 | 0.91 | 0.037 | 0.033 | 0.011 | 0.032 | 0.018 | 0.216 |
| Subject 20 | 0.80 | 0.86 | 0.84 | 0.019 | 0.027 | 0.018 | 0.032 | 0.055 | 0.315 |
| Subject 21 | 0.80 | 0.82 | 0.87 | 0.057 | 0.040 | 0.017 | 1.000 | 0.018 | 0.142 |
| Subject 22 | 0.91 | 0.94 | 0.97 | 0.018 | 0.031 | 0.012 | 0.315 | 0.018 | 0.142 |
| Subject 23 | 0.78 | 0.76 | 0.79 | 0.034 | 0.037 | 0.013 | 1.000 | 0.796 | 0.216 |
| Subject 24 | 0.93 | 0.95 | 0.98 | 0.043 | 0.035 | 0.007 | 0.444 | 0.055 | 0.139 |
| Subject 25 | 0.62 | 0.63 | 0.68 | 0.029 | 0.046 | 0.042 | 1.000 | 0.055 | 0.142 |
| Subject 26 | 0.55 | 0.63 | 0.57 | 0.050 | 0.052 | 0.037 | 0.090 | 0.605 | 0.216 |
| Means | 0.73 | 0.75 | 0.77 | 0.032 | 0.045 | 0.032 | 0.735 | 0.357 | 0.575 |

**Table 6:** Subject-specific results of **AUC**. Prediction on test set based on model training on 5-fold training set. P-values are based on Mann-Whitney U test and Bonferroni-corrected.

| | Average of 5-fold cross validation | | | St.dev of 5-fold cross validation | | | P-value | | |
|---|---|---|---|---|---|---|---|---|---|
| | LDA | CNN-LSTM | UNet | LDA | CNN-LSTM | UNet | LDA vs CNN-LSTM | LDA vs UNet | UNet vs CNN-LSTM |
| Subject 1 | 0.77 | 0.76 | 0.67 | 0.03 | 0.04 | 0.05 | 1.000 | 0.032 | 0.054 |
| Subject 2 | 0.54 | 0.57 | 0.47 | 0.05 | 0.03 | 0.00 | 0.307 | 0.178 | 0.010 |
| Subject 3 | 0.79 | 0.82 | 0.60 | 0.06 | 0.04 | 0.08 | 0.695 | 0.017 | 0.017 |
| Subject 4 | 0.61 | 0.58 | 0.49 | 0.04 | 0.09 | 0.03 | 1.000 | 0.018 | 0.214 |
| Subject 5 | 0.60 | 0.63 | 0.61 | 0.03 | 0.04 | 0.03 | 0.444 | 1.000 | 1.000 |
| Subject 6 | 0.73 | 0.71 | 0.72 | 0.04 | 0.05 | 0.04 | 1.000 | 1.000 | 1.000 |
| Subject 7 | 0.66 | 0.64 | 0.62 | 0.01 | 0.07 | 0.02 | 1.000 | 0.032 | 1.000 |
| Subject 8 | 0.58 | 0.57 | 0.49 | 0.03 | 0.03 | 0.04 | 0.796 | 0.018 | 0.018 |
| Subject 9 | 0.69 | 0.72 | 0.67 | 0.03 | 0.03 | 0.04 | 0.313 | 1.000 | 0.142 |
| Subject 10 | 0.61 | 0.65 | 0.47 | 0.04 | 0.05 | 0.01 | 0.142 | 0.017 | 0.017 |
| Subject 11 | 0.61 | 0.62 | 0.42 | 0.04 | 0.07 | 0.02 | 0.605 | 0.018 | 0.018 |
| Subject 12 | 0.73 | 0.62 | 0.47 | 0.04 | 0.06 | 0.03 | 0.054 | 0.014 | 0.015 |
| Subject 13 | 0.60 | 0.70 | 0.48 | 0.06 | 0.04 | 0.00 | 0.067 | 0.011 | 0.011 |
| Subject 14 | 0.64 | 0.66 | 0.54 | 0.05 | 0.05 | 0.04 | 0.796 | 0.018 | 0.018 |
| Subject 15 | 0.58 | 0.62 | 0.42 | 0.02 | 0.04 | 0.03 | 0.090 | 0.018 | 0.018 |
| Subject 16 | 0.53 | 0.57 | 0.48 | 0.03 | 0.04 | 0.05 | 0.214 | 0.141 | 0.032 |
| Subject 17 | 0.54 | 0.56 | 0.45 | 0.02 | 0.07 | 0.02 | 0.794 | 0.016 | 0.030 |
| Subject 18 | 0.61 | 0.64 | 0.68 | 0.03 | 0.06 | 0.02 | 0.444 | 0.032 | 0.216 |
| Subject 19 | 0.63 | 0.75 | 0.51 | 0.06 | 0.08 | 0.06 | 0.032 | 0.031 | 0.018 |
| Subject 20 | 0.72 | 0.74 | 0.74 | 0.02 | 0.04 | 0.04 | 0.796 | 1.000 | 1.000 |
| Subject 21 | 0.73 | 0.73 | 0.66 | 0.03 | 0.06 | 0.05 | 0.901 | 0.089 | 0.141 |
| Subject 22 | 0.85 | 0.87 | 0.75 | 0.01 | 0.02 | 0.03 | 0.136 | 0.017 | 0.018 |
| Subject 23 | 0.69 | 0.60 | 0.50 | 0.04 | 0.09 | 0.04 | 0.051 | 0.015 | 0.296 |
| Subject 24 | 0.82 | 0.78 | 0.59 | 0.07 | 0.06 | 0.11 | 0.514 | 0.032 | 0.032 |
| Subject 25 | 0.54 | 0.55 | 0.42 | 0.06 | 0.04 | 0.04 | 1.000 | 0.055 | 0.018 |
| Subject 26 | 0.51 | 0.55 | 0.51 | 0.05 | 0.03 | 0.03 | 0.444 | 1.000 | 0.090 |
| Means | 0.65 | 0.66 | 0.55 | 0.04 | 0.05 | 0.04 | 0.574 | 0.263 | 0.230 |

**Table 7:** Subject-specific results of **Unweighted F-score**. Prediction on test set based on model training on 5-fold training set. P-values are based on Mann-Whitney U test and Bonferroni-corrected.

# 7 References


[1] BCI Challenge @ NER 2015. A Brain Computer Interface challenge about detecting Error Potentials. 2015. URL https://www.kaggle.com/c/inria-bci-challenge.

[2] Anna Beres. Time is of the essence: A review of electroencephalography (eeg) and event-related brain potentials (erps) in language research. *Applied Psychophysiology and Biofeedback*, 42, 12 2017. doi: 10.1007/s10484-017-9371-3.

[3] Benjamin Blankertz, Steven Lemm, Matthias Treder, Stefan Haufe, and Klaus-Robert Müller. Single-trial analysis and classification of ERP components - A tutorial. *NeuroImage*, 56(2): 814–825, may 2010. doi: 10.1016/j.neuroimage.2010.06.048.

[4] Ricardo Chavarriaga, Aleksander Sobolewski, and Jose del R. Millan. Errare machinale est: The use of error-related potentials in brain-machine interfaces. *Frontiers in Neuroscience*, 8, 07 2014. doi: 10.3389/fnins.2014.00208.

[5] Alexander Craik, Yongtian He, and Jose L Contreras-Vidal. Deep learning for electroencephalogram (EEG) classification tasks: a review. *Journal of Neural Engineering*, 16(3): 031001, apr 2019. doi: 10.1088/1741-2552/ab0ab5. URL https://doi.org/10.1088/1741-2552/ab0ab5.

[6] Michael Falkenstein, J Hohnsbein, J Hoormann, and L Blanke. Effects of crossmodal divided attention on late erp components: Ii. *Electroencephalography and clinical neurophysiology*, 78:447–55, 07 1991. doi: 10.1016/0013-4694(91)90062-9.

[7] Lawrence Farwell and Emanuel Donchin. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical neurophysiology*, 70:510–23, 01 1989. doi: 10.1016/0013-4694(88)90149-6.

[8] Monica Fira and Goras Liviu. Comparison of inter-and intra-subject variability of p300 spelling dictionary in eeg compressed sensing. *International Journal of Advanced Computer Science and Applications*, 7, 10 2016. doi: 10.14569/IJACSA.2016.071049.

[9] William Gehring, Brian Goss, Michael Coles, David Meyer, and Emanuel Donchin. A neural system for error detection and compensation. *Psychological Science*, 4, 11 1993. doi: 10.1111/j.1467-9280.1993.tb00586.x.

[10] Parisa Ghane and Ulisses Braga-Neto. Comparison of classification algorithms towards subject-specific and subject-independent bci. 12 2020.

[11] Dean Krusienski, Eric Sellers, Dennis Mcfarland, Theresa Vaughan, and Jonathan Wolpaw. Toward enhanced p300 speller performance. *Journal of neuroscience methods*, 167:15–21, 02 2008. doi: 10.1016/j.jneumeth.2007.07.017.

[12] Akshay Kumar, Lin Gao, Elena Pirogova, and Qiang Fang. A review of error-related potential-based brain-computer interfaces for motor impaired people. *IEEE Access*, PP, 10 2019. doi: 10.1109/ACCESS.2019.2944067.

[13] Parashivappagol Kumar and A.P. Vinod. Improving classification accuracy of detecting error-related potentials using two-stage trained neural network classifier. pages 1–5, 12 2020. doi: 10.1109/iCAST51195.2020.9319482.

[14] Stephanie Lees, Natalie Dayan, Hubert Cecotti, Paul Mccullagh, Liam Maguire, Fabien Lotte, and Damien Coyle. A review of rapid serial visual presentation-based brain-computer interfaces. *Journal of Neural Engineering*, 15, 11 2017. doi: 10.1088/1741-2552/aa9817.

[15] Catarina Lopes-Dias, Andreea-Ioana Sburlea, and Gernot Müller-Putz. Online asynchronous decoding of error-related potentials during the continuous control of a robot. *Scientific Reports*, 9, 11 2019. doi: 10.1038/s41598-019-54109-x.

[16] F Lotte, L Bougrain, A Cichocki, M Clerc, M Congedo, A Rakotomamonjy, and F Yger. A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update. *Journal of Neural Engineering*, 15(3):031005, apr 2018. doi: 10.1088/1741-2552/aab2f2.

[17] Fabien Lotte. A tutorial on eeg signal processing techniques for mental state recognition in brain-computer interfaces. 10 2014. doi: 10.1007/978-1-4471-6584-2_7.

[18] Shijian Lu, Cuntai Guan, and Haihong Zhang. Learning adaptive subject-independent p300 models for eeg-based brain-computer interfaces. pages 2461–2465, 06 2008. doi: 10.1109/IJCNN.2008.4634141.

[19] Dennis Mcfarland, Charles Anderson, Klaus-Robert Müller, Alois Schlögl, and Dean Krusienski. Bci meeting 2005—workshop on bci signal processing: Feature extraction and translation. *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 14:135–8, 07 2006. doi: 10.1109/TNSRE.2006.875637.

[20] Christoph Michel and Denis Brunet. Eeg source imaging: A practical review of the analysis steps. *Frontiers in Neurology*, 10, 04 2019. doi: 10.3389/fneur.2019.00325.

[21] Jiahui Pan, Yuanqing Li, Zhenghui Gu, and Zhuliang Yu. A comparison study of two p300 speller paradigms for brain–computer interface. *Cognitive Neurodynamics*, 7, 12 2013. doi: 10.1007/s11571-013-9253-1.

[22] Margaux Perrin, Emmanuel Maby, Sébastien Daligault, Olivier Bertrand, and Jérémie Mattour. Objective and Subjective Evaluation of Online Error Correction during P300-Based Spelling. *Advances in Human-Computer Interaction*, 2012, dec 2012. doi: 10.1155/2012/578295.

[23] Mathias Perslev, Michael Jensen, Sune Darkner, Poul Jennum, and Christian Igel. U-time: A fully convolutional network for time series segmentation applied to sleep staging. 10 2019.

[24] Gerard Remijn, Emi Hasuo, Haruna Fujihira, and Satoshi Morimoto. An introduction to the measurement of auditory event-related potentials (erps). *Acoustical Science and Technology*, 35:229–242, 09 2014. doi: 10.1250/ast.35.229.

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. 05 2015.

[26] Carolina Saavedra and Laurent Bougrain. Processing stages of visual stimuli and event-related potentials. 10 2012.

[27] Bernardo Seno, Matteo Matteucci, and Luca Mainardi. Online detection of p300 and error potentials in a bci speller. *Comp. Int. and Neurosc.*, 2010, 01 2010.

[28] Martin Spüler and Christian Niethammer. Error-related potentials during continuous feedback: using eeg to detect errors of different type and severity. *Frontiers in Human Neuroscience*, 9:155, 2015. doi: 10.3389/fnhum.2015.00155.

[29] Martin Spüler, Michael Bensch, Sonja Kleih, Wolfgang Rosenstiel, Martin Bogdan, and Andrea Kübler. Online use of error-related potentials in healthy users and people with severe motor impairment increases performance of a P300-BCI. *Clinical Neurophysiology*, 123(7): 1328–1337, jul 2012. doi: 10.1016/j.clinph.2011.11.082.

[30] Akara Supratak, Hao Dong, Chao Wu, and Yike Guo. Deepsleepnet: a model for automatic sleep stage scoring based on raw single-channel eeg. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, PP, 03 2017. doi: 10.1109/TNSRE.2017.2721116.

[31] Shravani Sur and Vinod Sinha. Event-related potential: An overview. *Industrial psychiatry journal*, 18:70–3, 01 2009. doi: 10.4103/0972-6748.57865.

[32] Juan Mayor Torres, Tessa Clarkson, Evgeny Stepanov, Cristrian Luhmann, Matthew Lerner, and Giuseppe Riccardi. Enhanced error decoding from error-related potentials using convolutional neural networks. *Conference proceedings: Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2018, 07 2018. doi: 10.1109/EMBC.2018.8512183.

[33] G Visconti, B Seno, Matteo Matteucci, and Luca Mainardi. Automatic recognition of error potentials in a p300-based brain-computer interface. *Proceedings of the 4th International Brain-Computer Interface Workshop  Training Course*, pages 238–243, 01 2008.

[34] Nile R. Wilson, Devapratim Sarma, Jeremiah D. Wander, Kurt E. Weaver, Jeffrey G. Ojemann, and Rajesh P. N. Rao. Cortical topography of error-related high-frequency potentials during erroneous control in a continuous control brain–computer interface. *Frontiers in Neuroscience*, 13:502, 2019. doi: 10.3389/fnins.2019.00502. URL `https://www.frontiersin.org/article/10.3389/fnins.2019.00502`.

[35] G. Woodman. A brief introduction to the use of event-related potentials in studies of perception and attention. *Attention, perception  psychophysics*, 72 8:2031–46, 2010.

[36] Gaowei Xu, Tianhe Ren, Yu Chen, and Wenliang Che. A one-dimensional cnn-lstm model for epileptic seizure recognition using eeg signal analysis. *Frontiers in Neuroscience*, 14, 12 2020. doi: 10.3389/fnins.2020.578126.

[37] Timothy Zeyl, Erwei Yin, Michelle Keightley, and Tom Chau. Adding real-time bayesian ranks to error-relatedpotential scores improves error detection and auto-correction in ap300 speller. *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society*, 24, 08 2015. doi: 10.1109/TNSRE.2015.2461495.