EE 385V Term Project: Classifying Error-Related Potentials

Deland Liu, Jackson Lightfoot, Jonathan Madera

May 12, 2021

1 Introduction

1.1 Project objective

This project aims to decode error-related potentials (ErrPs) in a BCI speller application built to provide an alternative communication avenue to patients with severe motor disabilities. ErrPs can be exploited to improve BCI performance. For instance, detection of ErrPs can trigger the BCI to take corrective actions. Using ErrP feedback, BCI decoders can also adapt and learn from their mistakes, such as in reinforcement learning [5].

1.2 Experimental Design and Dataset Description

The BCI spelling experiment is described in detail in [4]. In summary, subjects monitored the speller cursor, which moved between adjacent characters every 1000 ms. After each cursor movement, 16-channel EEG was decoded to detect the presence of an ErrP. When the cursor moves away from the intended character, the subject's perception of this error can elicit an ErrP. Based on the decoder output and a natural language model, the reinforcement-learning algorithm re-estimates the next, most probable character. EEG signals were recorded using 16 channels on four healthy subjects during the experiment.

The EEG dataset of each subject contains three sessions: one offline and two online (S2, S3). Each session contains four runs, each corresponding to the spelling of a single word and containing a sequence of trials. Individual trials are extracted according to triggers, which mark the time of cursor movements. Each trial can be a 'correct' or 'error' trial, depending on whether an ErrP was generated during the trial. Hence, ErrP classification is a binary classification problem.

1.3 ErrP

Error-related potentials (ErrP) are a systematic, temporal response in the anterior cingulate cortex of the human brain to a stimulus that indicates an error was made or observed. The ErrP is characterized by a negative deflection (error-related negativity) in EEG over the fronto-central scalp areas after the onset of an error-causing stimulus, followed by a positive deflection (error positivity) [18, 6]. ErrP is also characterized by frequency modulations, specifically an increase of theta band activity (4-8 Hz) [5]. To illustrate the characteristics of ErrPs, Fig. 1a shows the grand averages (GA) of trials in Subject 1's Offline session FCz channel. Fig. 1b depicts topological plots of the 'error' and 'correct' trials at 484 ms, further highlighting the ErrP negative deflection. Compared to 'correct' trials, the 'error' trials have a delayed and more significant negative deflection on average. The topological plots suggest that central and fronto-central areas are the main areas involved in producing ErrPs.

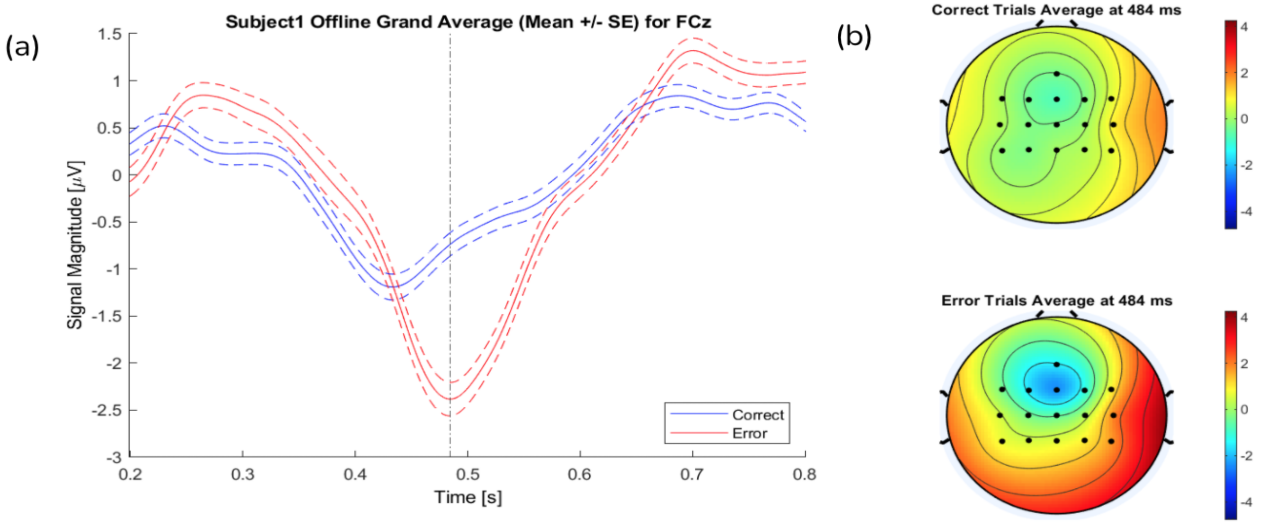

Figure 1: (a) GA plot of 'error' and 'correct' trials in Subject 1 Offline session FCz channel, showing the differences in the temporal waveforms of the two classes. (b) The topological plots of the 'error' and 'correct' trials of the same session at 484 ms. Frontal negative pattern is more significant in the 'error' trials than in the 'correct' trials.

2 Methods

2.1 EEG pre-processing

Before feature extraction and selection, the ErrP dataset was pre-processed using temporal and spatial filtering techniques. ErrPs are often characterized by modulations and high power in the theta band (4-8 Hz), hence the dataset was band-pass filtered in the 1-10 Hz band with a 3rd order Butterworth filter to improve the signal-to-noise ratio. ErrP originates from a broad source, so common average reference spatial filtering was applied to increase the spatial resolution between the EEG channels.

2.2 Feature Extraction

In this project, both temporal and spectral-domain features were extracted from all 16 channels. Temporal features may be useful to capture distinct EEG waveform patterns of ErrPs in the temporal domain. As illustrated in Fig.1, ErrPs tend to have high temporal resolutions and are characterized by error-related negativity followed by error-related positivity. These maxima and minima components can be captured by temporal domain features. Spectral features may capture frequency modulations that characterize ErrPs, e.g. an increase in theta band activity. Spectral features are also reported to be more robust to temporal jitter across trials [5]. A combination of temporal and spectral-domain features may also improve ErrP classification performances, as reported in [15].

In each trial, features were extracted from a window of 200-800 ms after the trigger, where most information of the ErrP is expected. Seven different types of features commonly reported in ErrP classification research were extracted from the window: **(1)** Time-stamps **(2)** Interval features **(3)** Wavelet coefficients **(4)** Power Spectral Density **(5)** Signal power **(6)** Maximum and minimum of voltage and **(7)** Standard deviation of signal.

2.2.1 Time-stamps

Time-stamps are a popular feature extracted for ErrP classification [5]. To extract time-stamps, signals in the window of each trial were down-sampled to 64 Hz from 512 Hz. Time-stamp features may have several drawbacks when used for ErrP classification. For instance, individual time-stamps are unlikely to be stable across different sessions, since the exact times of error negativity will vary across different trials and sessions. In addition, time-stamps deemed discriminant by the Fisher algorithm (high Fisher scores) may be highly correlated, since the Fisher algorithm computes the Fisher score for each feature independently. It can be argued that some of these features may be 'redundant' - disregarding some of them may improve classification performance by reducing overfitting risks. Therefore, time-stamp features alone may not lead to good classification performances.

2.2.2 Interval features

Interval features are potentially more stable than time-stamps. Incoming trials were first segmented into non-overlapping, consecutive intervals, and the means of these intervals were extracted as interval features, as shown below:

$$Y[n] = \frac{1}{T}\sum_{t=0}^{T-1} X[t+nT]$$

$$n = 0, \cdots, \frac{N}{T} - 1$$

where Y is the vector of interval features computed from each trial X, N is the length of trial X, and T is the interval length, which was set to 12 samples [2].

2.2.3 Power Spectral Density (PSD)

PSD features were extracted to capture error-related frequency modulations. For frequency domain features, the power spectrum was estimated using the Welch method at window lengths of 400 ms, 60% overlap, and frequency resolution of 0.25 Hz [15]. The first 40 bins of the periodogram were extracted as features. The Welch method can compute a more consistent estimator of PSD compared to standard periodogram because it takes the ensemble average of the periodogram for each window. Compared to simply extracting signal power or band power as features, the PSD features can provide more specific information of power distributed at individual frequency bins.

2.2.4 Wavelet coefficients

Wavelet transform is a multi-resolution analytical technique that offers both time and frequency localization. Wavelet coefficients have been used in ErrP classification [12], but are less popular compared to features described above. Wavelet coefficients were extracted as features in this report because wavelet transforms have been widely applied in other EEG applications (e.g. denoising, compression) and have proven useful in capturing the spectral-temporal characteristics of EEG [9, 8]. The Daubechie 8 wavelet function at five levels of decomposition was selected in this study. The Daubechie 8 function has been widely used in EEG denoising and is reported to better conserve decomposed EEG signals [14].

Other features extracted from each trial are **(1)** Signal power, computed as $(\text{rms}(\text{signal}))^2$, **(2)** Max and Min voltage, and **(3)** Standard deviation of signal.

2.3 Feature Selection

After extracting all of these 406 individual features across 16 EEG channels, the total number of features for each subject was 6,496. However, using a feature space this large to train the classifiers would clearly overfit to the training set. Thus, a feature selection algorithm was implemented to choose smaller feature spaces that contain the most stable and discriminant features.

To estimate the discriminant power (DP) of each extracted feature, the Fisher algorithm was used. Fisher score is a popular supervised, filter-based feature selection method. This metric finds features of which the instances from different classes are as far apart as possible, and instances from the same class are as close as possible. Therefore, a feature with a larger Fisher score should theoretically be more discriminant. The Fisher score of the i$^{\text{th}}$ feature is computed as follows:

$$S(x_i) = \frac{\sum_{j=1}^{c} n_j (u_{ij} - u_i)^2}{\sum_{j=1}^{c} n_j \sigma_{ij}^2} \tag{1}$$

Here, u_{ij} and σ_{ij} are the mean and standard deviation of the i$^{\text{th}}$ feature in the j$^{\text{th}}$ class, u_i is the mean of the i$^{\text{th}}$ feature across whole dataset, n_j is the size of the j$^{\text{th}}$ class, and c is the number of classes [7].

Applying the Fisher score algorithm to the extracted feature space results in an ordered list of the features from most discriminant to least discriminant. However, this metric does not reveal much information about the stability of these features. In order to ensure the selected features generalize to the online sessions, an additional method must be used to evaluate the stability of the features. Since each offline session is divided into four runs of 125 trials each, the Fisher algorithm was applied to each individual run. This allows Fisher scores to be compared across runs, and a feature with consistently high Fisher scores across all four runs would be considered stable.

Since the optimal feature space size for each subject was unknown, a thresholding technique was used to vary the percentage of features considered when evaluating stability. For example, with a threshold (T) of 25%, the top 1,624 features (based on Fisher score) of each run were considered. Then, the intersection of these sets was taken to keep only the stable features. Put differently, only the features that show up in the top 1,624 features of all four runs were deemed stable and subsequently selected.

12 different threshold values were used, varying from 2% to 50%. It should be noted that even though the same T was used for each subject, this algorithm will result in a different number of selected features for each subject. When training the classifiers, T is used as a hyperparameter to be tuned. This optimizes the feature space size for each individual classifier. Note that this hyperparameter does not affect the classifier itself, but only the size of the inputted feature space.

2.3.1 Feature Discussion

Table 1 shows the top six stable, discriminant features selected from each subject using the procedure described above. Most of the electrodes shown here, such as FCz and Cz, belong to the fronto-central areas. This is expected as the fronto-central region is the main source for error-related activity. Across different subjects, many different categories of features showed up as the most stable and discriminant. Subject 1's best features were primarily time stamps (TS), subjects 2 and 4's were PSD bins, and Subject 5's contained many different categories.

The time-stamp and PSD features selected for Subjects 1, 2, and 4 definitely contain redundant features that could be removed. An idea for future work would be to remove the redundant features by using a multivariate feature selection algorithm. More focus was put on the thresholding technique with cross-validation to optimize the feature space size.

Subject1	**Subject2**	**Subject4**	**Subject5**
FCz TS 497ms	FC2 SP	Cz PSD bin 31	Cz SP
FCz TS 513ms	FC2 Std	Cz PSD bin 32	Cz Std
FCz TS 528ms	FC1 PSD bin 33	Cz PSD bin 33	C1 TS 450ms
C1 TS 559ms	FCz PSD bin 30	Cz PSD bin 34	FC1 Inter 7
FCz Inter 7	FCz PSD bin 31	Cz PSD bin 35	C1 DWT 14
FCz DWT 16	FCz PSD bin 32	Cz PSD bin 36	Cz DWT 20
Notation: (Channel, Type, Number); TS: Time-stamp; SP: Signal power; Inter: Interval; DWT: Wavelet coeff; Std: Standard deviation			

Table 1: Top six stable, discriminant features selected for each subject. For wavelet and interval features, the number (X) following feature type refers to the X*th* feature of the respective type extracted from each trial.

3 ErrP Classification

Once features were extracted, selected, and analyzed, the next step was classification. Four supervised classifiers were chosen based on previous literature, including two linear (Support Vector Machine and Linear Discriminant Analysis) and two non-linear (Neural Network and Random Forest). To evaluate these classifiers, five different classification metrics were recorded: **(1)** Accuracy, **(2)** Sensitivity (true positive rate or TPR), **(3)** Specificity (true negative rate or TNR), **(4)** False positive rate (FPR), and **(5)** Kappa. Since the dataset has more 'correct' trials than 'error' trials, Kappa was chosen as the primary metric, as it performs well with unbalanced datasets [17]. The formula for Kappa can be seen below:

$$P_e = \frac{(TN + FP)(TN + FN) + (FP + TP)(FN + TP)}{N_{total}^2} \tag{2}$$

$$Kappa = \frac{Accuracy - P_e}{1 - P_e} \tag{3}$$

where N_{total} refers to total number of trials classified. True positive here refers to an 'error' trial being correctly classified. Additionally, each performance metric was compared to that of chance-level decoding. To determine the chance-level for each metric, the test set labels were randomly permuted 10,000 times. During each iteration, every metric was evaluated on the permuted labels compared to ground truth. Averaging all iterations produced an estimate for chance-level decoding. For example, Kappa's chance-level decoding value is around 0.

3.1 Cross-Validation

Cross-validation (CV) is useful for **(1)** evaluating a classifier's offline performance and **(2)** tuning the hyperparameters of the classifier to improve generalization performance. In order to respect the temporality of non-stationary EEG signals, cross validation was performed run-wise. This means that each run of the offline session (four runs in total) corresponded to a validation fold. Each validation fold was evaluated using the five metrics described above.

For each of the four classifiers studied, the Grid Search algorithm was used for hyperparameter tuning. This algorithm looks for the best learning model by testing all combinations of the hyperparameters in their predefined ranges via cross validation. The threshold (T) used in the feature selection algorithm was used as a hyperparameter for all four classifiers to change the feature space size of the input data. Consistent across the four classifiers, 12 T values including [2% 3% 6% 8% 9% 10% 15% 20% 22% 30% 40% 50%] were evaluated, each corresponding to a feature space size. Other hyperparameters were tuned depending on the specific classifier. Note that this section only attempts to myopically optimize the classifiers. Due to time constraints, the hyperparameters were tuned over a small range.

In order to evaluate each classifier's performance, the mean Kappa across validation folds for the optimal set of hyperparameters was used. In an actual experimental setting, this metric would be used to choose the best classifier for each subject to use in the online sessions. Because the online data is already available in this project, online performance can be evaluated for every classifier as an additional metric. First, each classifier was trained on the entire offline session using the optimal hyperparameters (including feature space size) found in the Grid Search algorithm. Then, the trained classifiers were used to make predictions on the online sessions, and performance was evaluated using the same five metrics.

3.1.1 Support Vector Machine (SVM)

SVM (with linear kernel) has been used as an ErrP classifier in literature [15, 19]. SVM selects a linear classifier over the feature space with largest margin, so the data can be separated to the maximum extent. The linear SVM requires the tuning of C, which controls the maximum penalty on margin-violating observations, potentially helping prevent overfitting. Following the Grid Search algorithm described above, the T parameter controlling feature space size and C were tuned using cross-validation. C was selected from the set of values [0.01, 0.1, 1, 10, 100].

Fig. 2 shows the mean kappa across cross-validation folds over different feature space sizes and optimal C. The maxima were the optimal combinations of T and C returned from Grid Search. Note that at a given T, the number of features selected will vary across different subjects, therefore the plots varied in lengths across subjects. The cross-validation plots below are purely a visual demonstration of the optimal feature space size and do not reflect the methodology of Grid Search, which was performed over all combinations of hyperparameters.

3.1.2 Linear Discriminant Analysis (LDA)

Linear discriminant analysis is a classifier that assumes the feature space is generated from a multivariate Gaussian. All classes are assumed to have the same covariance matrix, but the mean vector is unique for each class. Given this probabilistic model, LDA chooses the class with the highest posterior probability. Previous BCI studies have successfully used LDA for general EEG decoding [13] and error-related potentials [11].

The Gamma hyperparameter, which controls the amount of regularization, was optimized over [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]. At maximum regularization of 1, the covariance matrix is assumed to be diagonal. Fig. 3 is similar to that for SVM, and it should be noted that subjects 1 and 5 required smaller feature space sizes compared to subjects 2 and 4. Additionally, subjects 1 and 2 preferred higher regularization, while subjects 4 and 5 preferred lower regularization.

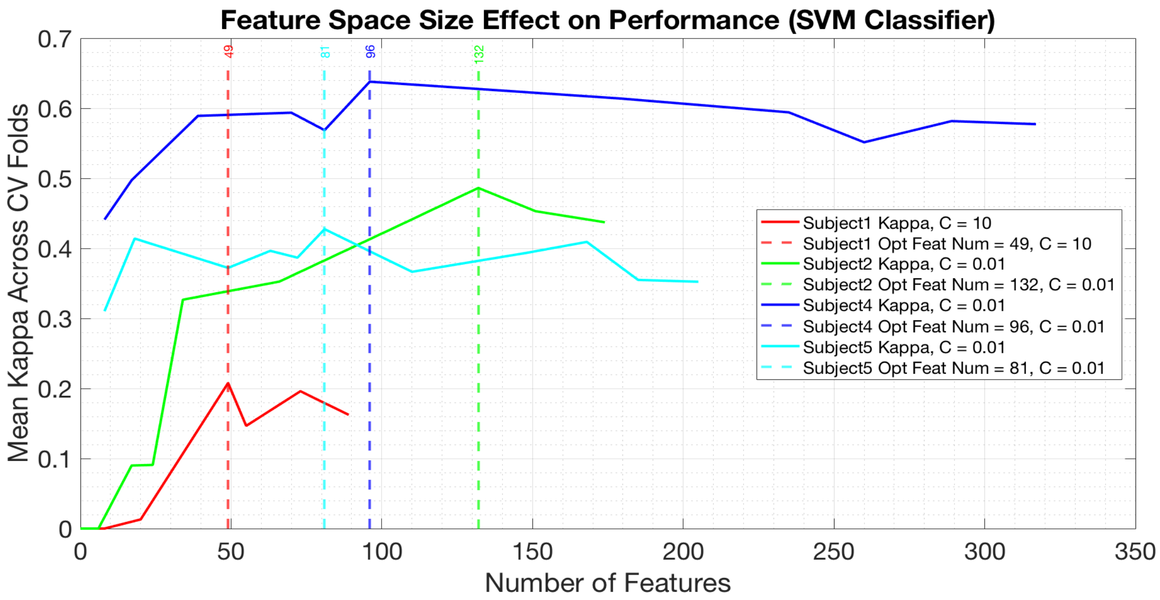

Figure 2: The optimal feature space size and C of SVM were selected using Grid Search with cross-validation. The labelled maxima mark the optimal combinations of T and C returned from Grid Search.

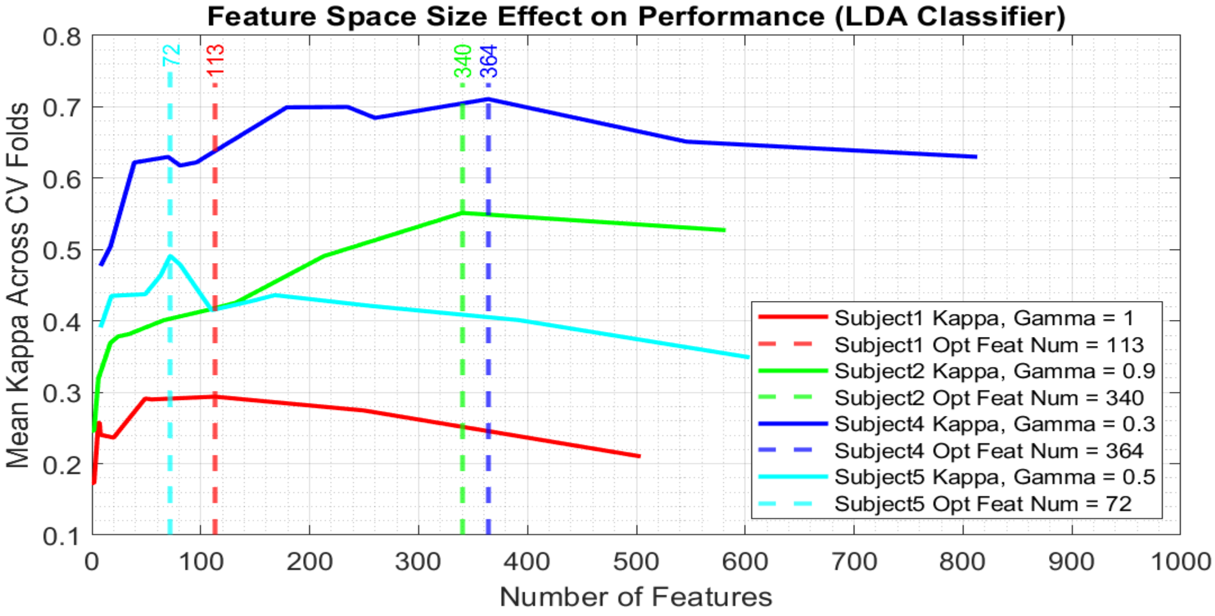

Figure 3: The optimal feature space size and Gamma of LDA were selected using Grid Search with cross-validation. The labelled maxima mark the optimal combinations of T and Gamma returned from Grid Search.

3.1.3 Neural Network (NN)

Several different kinds of NNs have been used in EEG-based BCI applications [16, 20]. For simplicity, this project employed a three-layer NN with a sigmoid activation function for classification. The second (hidden) layer size was allowed to vary. It was found that the larger the hidden layer size, the higher the risk of the NN overfitting to the training data. The last layer output a probability of classification from 0 to 1, where values less than 0.5 meant no ErrP was present in the trial, and values greater than 0.5 indicated the subject did elicit an ErrP.

In order to achieve the best performance of the NN, the hyperparameters hidden layer size HL, feature regularization constant lambda λ, and threshold parameter T were optimized using the Grid Search cross validation method. λ was selected from [1e-6, 1e-3, 1, 1e3, 1e6], and HL was selected from [10, 25, 50, 100]. Fig. 4 plots the mean kappa across validation folds over different feature space sizes and optimal hyperparamters. One thing to note is the NN had the best cross-validation performance when the input feature space size was larger compared to the other algorithms.

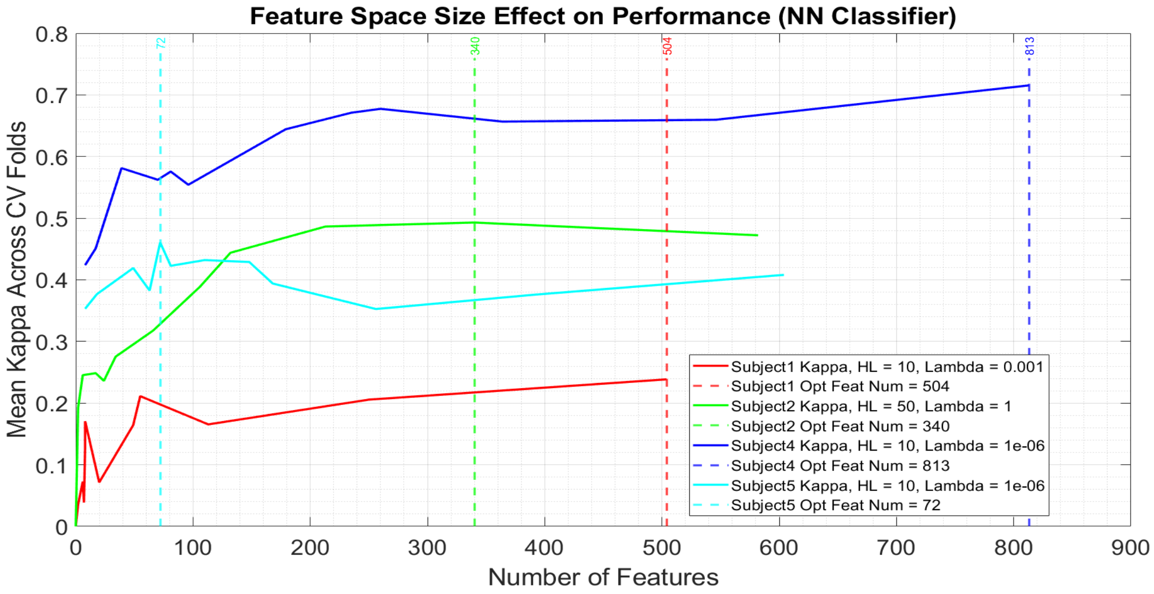

Figure 4: Optimal T, λ, and HL of NN were selected using Grid Search with cross-validation. The labelled maxima mark the optimal combinations of HL, λ, and T returned from Grid Search.

3.1.4 Random Forest (RF)

Similar to other classifiers studied, RF has also been used in ErrP classification in literature [10]. RF is a non-linear classifier that constructs multiple decision trees and combines them together to get a more accurate prediction. The Grid Search algorithm only searched for the optimal $numTrees$ (number of trees in forest), with other hyperparameters $MinLeafSize$ (min number of observations per tree leaf) and $NumPredictorsToSample$ (number of features considered randomly for a decision split) kept constant at 1 and square root of the feature space size, respectively. $numTrees$ was selected from the range [5, 50, 100, 200, 300]. Fig. 5 plots the mean Kappa across validation folds over different feature space sizes and optimal $numTrees$.

As mentioned, the online performances of BCIs will not always be available in real-world applications. Thus, classifiers' offline performances will be important in this aspect. The offline performances of classifiers are compared in Fig. 6, which shows the mean kappa across validation folds for the 4 subjects. The optimal hyperparameters returned from Grid Search were used in each classifier. As shown, LDA and NN slightly outperformed RF and SVM in Subjects 1, 2, 4.

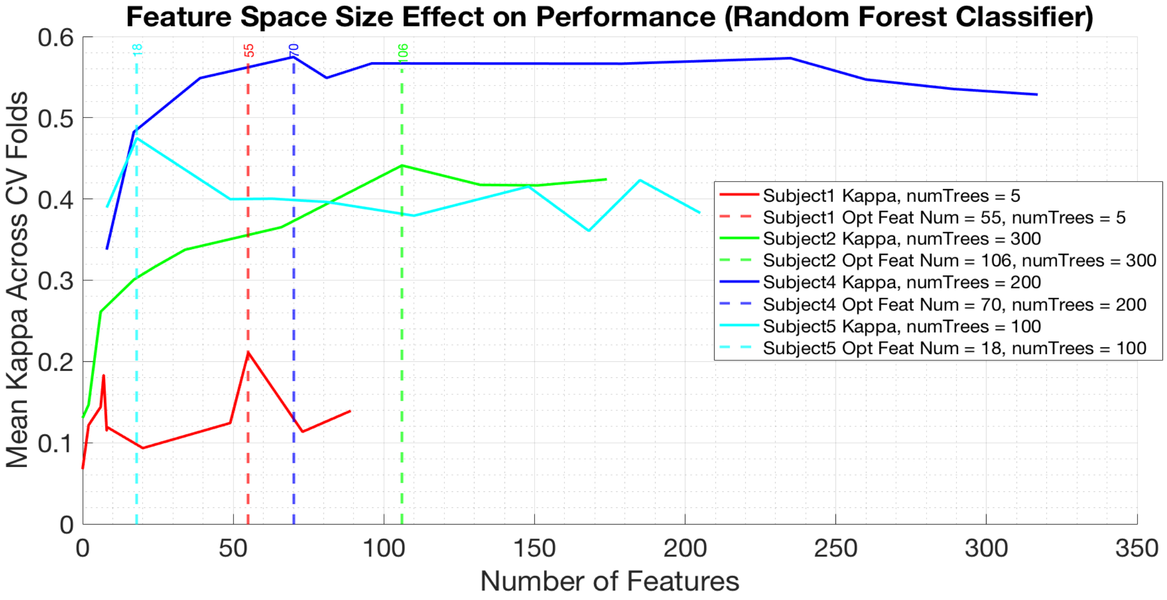

Figure 5: The optimal feature space size and $numTrees$ of RF were selected using Grid Search with cross-validation.

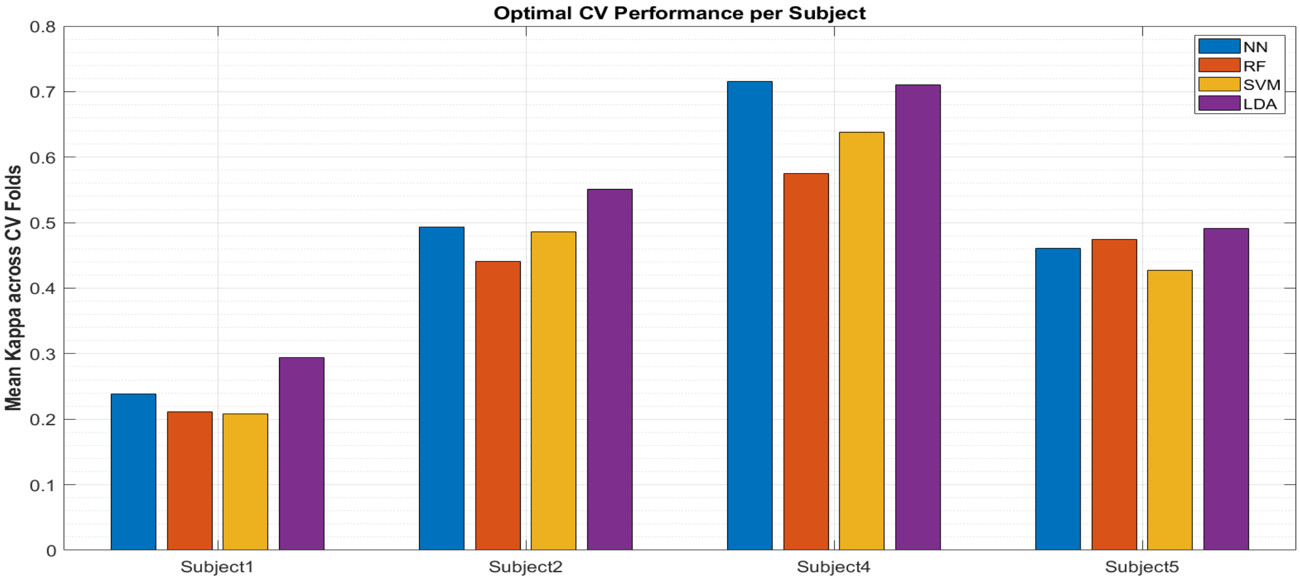

Figure 6: A comparison of the classifiers' mean Kappa across validation folds suggest that NN and LDA slightly outperformed the other classifiers in Subjects 1, 2, 4's offline sessions.

4 Results

As mentioned in section 3.1, after optimal hyperparameters were selected via cross-validation, the classifiers were retrained on the entire offline session and tested for online performance. Fig. 7, 8, 9, and 10 present the Kappa values and TPR vs. FPR of the online sessions of each subject using SVM, LDA, NN and RF classifiers, respectively.

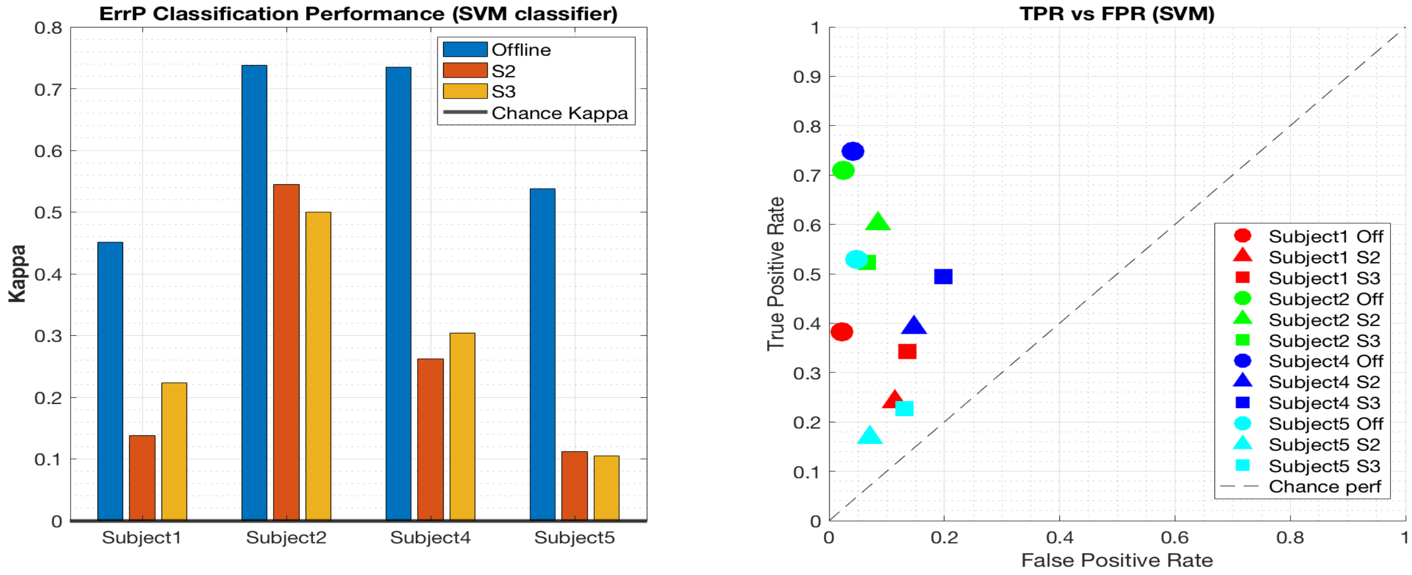

Figure 7: ErrP classification performance with SVM

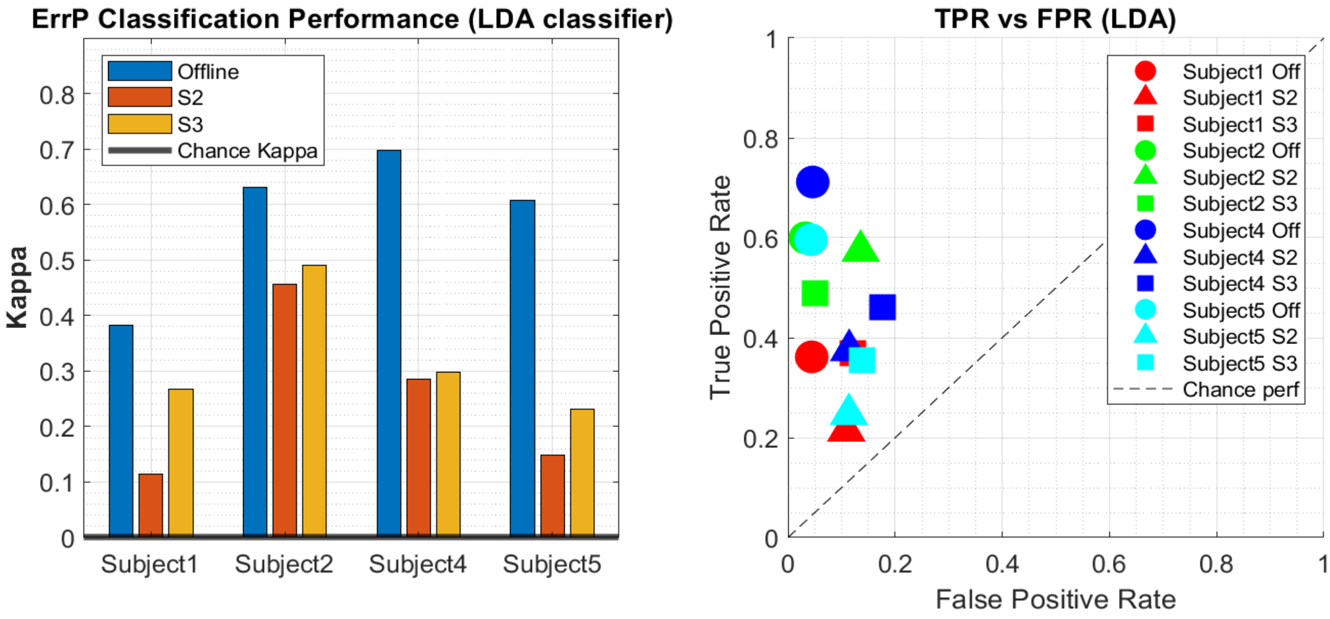

Figure 8: ErrP classification performance with LDA

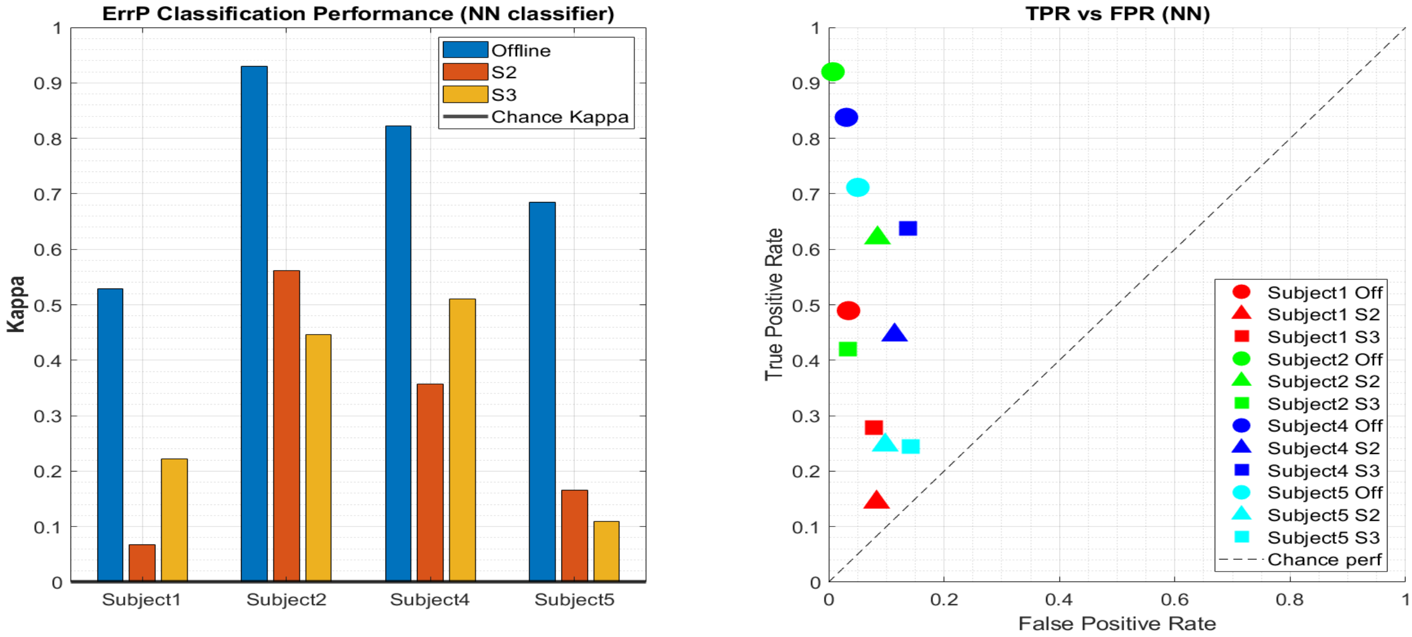

Figure 9: ErrP classification performance with NN

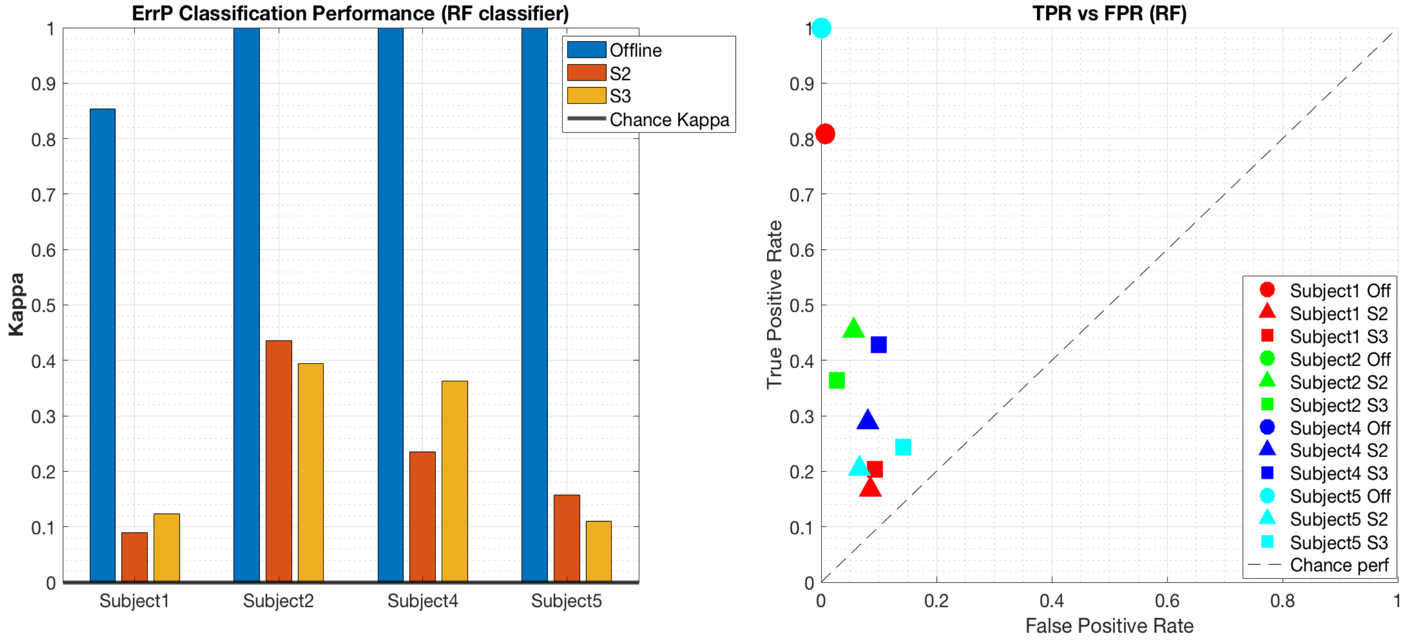

Figure 10: ErrP classification performance with RF

5 Discussion

As shown in Fig. 7, 8, 9, and 10, all four classifiers' Kappa exceeded chance level in the online sessions. Shown in Fig. 11, LDA and NN result in the highest Kappas in Subjects 4 and 5, whereas LDA and SVM show the highest Kappas in Subjects 1 and 2. Consistent across both

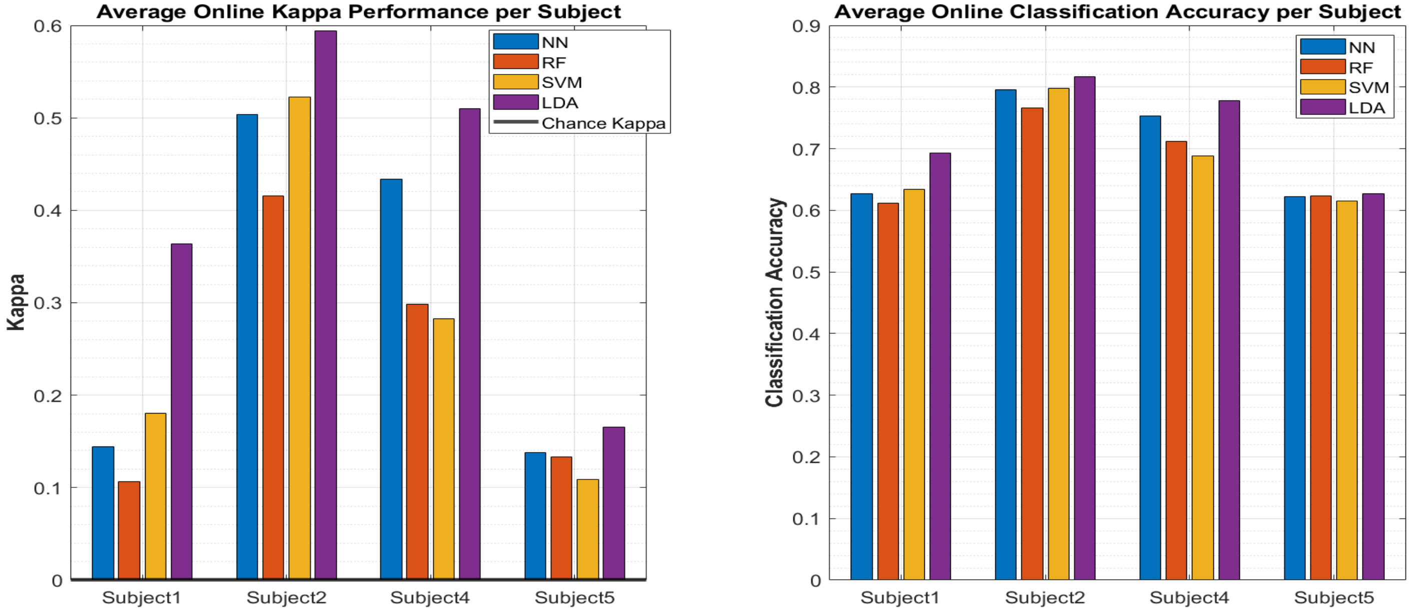

Figure 11: Average online performance metrics for Session 2 and 3 (Kappa, classification accuracy). The variability in the Kappa performance gives better indication of the best performing classifier compared to classification accuracy. Consistent with offline performance in Fig. 6, the LDA classifier had the best online performance.

offline and online performances in Fig. 6 and 11 respectively, LDA was the best performing classifier. Interestingly, the classification accuracy, biased towards the more frequently occurring class, did not show large discrepancies amongst the four classifiers. Subject wise, consistent across Fig. 6 and 11, Subjects 2 and 4 performed the best. Performance variability across subjects and sessions can be explained by fatigue, changes in signal quality, adaptation to protocol, etc. Looking at Fig. 2, 3 and 5, Subjects 2 and 4 had the largest number of stable and discriminant features selected, which may have contributed to better classification performances.

Across all classifiers, a significant drop in sensitivity (TPR) is observed in online performances compared to offline, especially in RF (Fig. 10). This suggests that more trials are classified as negatives (‘correct’). Discrepancy in offline and online performances suggests overfitting. The classifiers were only optimized myopically, with some hyperparameters tuned over a small range and others simply kept constant (e.g. in the case of RF). A limitation here is that the cross-validation Grid Search algorithm implemented is a brute force approach to optimization and can be computationally expensive. Its effectiveness relies on choosing a good range of hyperparameters for each classifier before the Grid Search. In the case of NN cross-validation (Fig. 4), the mean Kappas in Subjects 1 and 4 do not reach their maxima until the largest tested feature space sizes. This suggests that the range was not large enough to cover the optimal feature space size for NN. In the future, Bayesian optimization could be explored, in which the hyperparameter values are selected and updated according to a probabilistic mapping of the cross-validation performance. Bayesian optimization has been shown to outperform Grid Search in [1].

Fig. 11 suggests that the NN performed well compared to RF and SVM in Subject 4 and 5’s online performance. Based on this observation, it is compelling to further exploit the non-linearity of NNs and employ Deep Neural Networks (DNNs), which have become popular in speech recognition, bioinformatics, etc. due to their high level of performance. However, in this project the authors did not implement variants of DNNs due to several reasons. Firstly, DNNs are data hungry algorithms, and the authors believe there is not enough data provided in the offline sessions to achieve a significant performance increase over the other classifiers [3, 16]. A potential future direction is to explore augmenting datasets for DNN implementation. Secondly, a DNN's huge computational complexity and long training time may carry limitations in real-world BCI applications when the decoder has to be recalibrated constantly. Lastly, DNNs learn and extract features via a 'black box'. Hence, it is difficult to interpret the relationship between learned features and task-related neurophysiological patterns, which is important in BCI engineering [20].

In conclusion, this project studied into ErrP decoding in BCI applications. Based on literature, the authors extracted seven categories of features and selected discriminant, stable features based on Fisher score and a thresholding technique. Four classifiers commonly used in ErrP decoding research were implemented and optimized via Grid Search with cross-validation. Overall, the authors found that majority of stable and discriminant features selected belonged to the fronto-central regions, but the types of features selected were heavily dependent on subject. For example, amongst the top six stable and discriminant features, PSD features were predominant in two of the subjects, whereas time-stamp features were predominant in one subject. The authors also discovered that amongst the studied classifiers, LDA consistently showed the best decoding performance (w.r.t Kappa) in both offline and online sessions.

References

[1] James Bergstra et al. "Algorithms for Hyper-Parameter Optimization". In: *Advances in Neural Information Processing Systems*. Ed. by J. Shawe-Taylor et al. Vol. 24. Curran Associates, Inc., 2011, pp. 2546–2554. URL: https://proceedings.neurips.cc/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

[2] Benjamin Blankertz et al. "Boosting bit rates and error detection for the classification of fast-paced motor commands based on single-trial EEG analysis". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 11.2 (2003), pp. 127–131.

[3] R. Chavarriaga and J. d. R. Millan. "Learning From EEG Error-Related Potentials in Non-invasive Brain-Computer Interfaces". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 18.4 (2010), pp. 381–388. DOI: 10.1109/TNSRE.2010.2053387.

[4] Ricardo Chavarriaga, Inaki Iturrate, and José del R Millán. "Robust, accurate spelling based on error-related potentials". In: *Proceedings of the 6th International Brain-Computer Interface Meeting*. CONF. 2016.

[5] Ricardo Chavarriaga, Aleksander Sobolewski, and José del R Millán. "Errare machinale est: the use of error-related potentials in brain-machine interfaces". In: *Frontiers in neuroscience* 8 (2014), p. 208.

[6] Michael Falkenstein et al. "ERP components on reaction errors and their functional significance: a tutorial". In: *Biological Psychology* 51.2 (2000), pp. 87–107. ISSN: 0301-0511. DOI: https://doi.org/10.1016/S0301-0511(99)00031-9. URL: http://www.sciencedirect.com/science/article/pii/S0301051199000319.

[7] Quanquan Gu, Zhenhui Li, and Jiawei Han. "Generalized fisher score for feature selection". In: *arXiv preprint arXiv:1202.3725* (2012).

[8] G. Higgins et al. “EEG compression using JPEG2000: How much loss is too much?” In: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. Aug. 2010, pp. 614–617. DOI: 10.1109/IEMBS.2010.5628020.

[9] Garry Higgins et al. “The effects of lossy compression on diagnostically relevant seizure information in EEG signals.” In: *IEEE J. Biomedical and Health Informatics* 17.1 (2013), pp. 121–127.

[10] Anwar Isied and Hashem Tamimi. “Using random forest (rf) as a transfer learning classifier for detecting error-related potential (errp) within the context of p300-speller”. In: *Bernstein Conference*. 2015.

[11] Akshay Kumar, Elena Pirogova, and John Q Fang. “Classification of error-related potentials using linear discriminant analysis”. In: *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*. IEEE. 2018, pp. 18–21.

[12] Ludmila I Kuncheva and Juan J Rodrıguez. “Interval feature extraction for classification of event-related potentials (ERP) in EEG data analysis”. In: *Progress in Artificial Intelligence* 2.1 (2013), pp. 65–72.

[13] Nikolay V Manyakov et al. “Comparison of classification methods for P300 brain-computer interface on disabled subjects”. In: *Computational intelligence and neuroscience* 2011 (2011).

[14] Noor Al-Qazzaz et al. “Selection of mother wavelet functions for multi-channel EEG signal analysis during a working memory task”. In: *Sensors* 15.11 (2015), pp. 29015–29035.

[15] Martin Spüler and Christian Niethammer. “Error-related potentials during continuous feedback: using EEG to detect errors of different type and severity”. In: *Frontiers in human neuroscience* 9 (2015), p. 155.

[16] S. A. Swamy Bellary and J. M. Conrad. “Classification of Error Related Potentials using Convolutional Neural Networks”. In: *2019 9th International Conference on Cloud Computing, Data Science Engineering (Confluence)*. 2019, pp. 245–249. DOI: 10.1109/CONFLUENCE.2019.8776901.

[17] Eoin Thomas, Matthew Dyson, and Maureen Clerc. “An analysis of performance evaluation for motor-imagery based BCI”. In: *Journal of neural engineering* 10.3 (2013), p. 031001.

[18] Christos E Vasios et al. “Classification of event-related potentials associated with response errors in actors and observers based on autoregressive modeling”. In: *The Open Medical Informatics Journal* 3 (2009), p. 32.

[19] Errikos M Ventouras et al. “Classification of error-related negativity (ERN) and positivity (Pe) potentials using kNN and support vector machines”. In: *Computers in biology and medicine* 41.2 (2011), pp. 98–109.

[20] Xiang Zhang et al. “A survey on deep learning-based non-invasive brain signals: recent advances and new frontiers”. In: *Journal of Neural Engineering* (2020).