# Department of Electronic and Telecommunication Engineering

University of Moratuwa

## Biosignal Processing

### Paper Implementation

# BCI Competition III: Dataset II
# Ensemble of SVMs for BCI P300 Speller

Bandara D.M.D.V.
Undergraduate
Dept. Electronic and Telecommunications
Faculty of Engineering
University of Moratuwa

Amarasekara A.T.P.
Undergraduate
Dept. Electronic and Telecommunications
Faculty of Engineering
University of Moratuwa

Date: December 18, 2024

# Table of Content


**1 Introduction and Background** **2**

**2 Overview** **3**
2.1 Selected Paper . . . 3
2.2 Dataset . . . 3

**3 Methodology** **5**
3.1 Data Preprocessing . . . 5
3.2 Model Training . . . 5
3.3 Character Prediction . . . 6

**4 Implementation Details** **7**
4.1 Data - Preprocessing . . . 7
4.1.1 Windowing . . . 7
4.1.2 Filtering . . . 7
4.1.3 Feature Vectoring . . . 8
4.2 Data - Partitioning . . . 8
4.3 Model Training . . . 8
4.4 Channel Ranking . . . 8
4.5 Channel Selection . . . 9
4.6 C Parameter Fine-Tuning . . . 9
4.7 Character Prediction . . . 9

**5 Implementation Results** **10**

**6 Conclusion and Future Improvements** **13**


# List of Figures


1 Example time course of an average signal waveform vs. a signal waveform with P300 EPR (at Cz) [1]. . . . 2
2 The user display for this paradigm. In this example, the user's task is to spell the word "SEND" (focusing on one character at a time) [1]. . . . 4
3 Filtered and Decimated dataset . . . 7
4 Concatenated Feature vector . . . 8
5 SVM accuracy before fine tuning . . . 10
6 Top 12 channels for each 15 partitions in order. . . . 10
7 Top 12 channels of first 5 partitions according to the paper . . . 10
8 $C_{cs}$ accuracy vs. Number of channels. Result from our implementation (left) and from paper (right) . . . 11
9 $C_{cs}$ accuracy vs C parameter . . . 11
10 SVM 1 accuracy after fine-tuning . . . 11
11 Character Prediction . . . 12

# 1 Introduction and Background

Brain-Computer Interfaces (BCIs) enable direct communication between the human brain and external devices. The technology holds promise for neurological research, in addition to assisting people with severe motor impairments by allowing them to interact with their environment through brain signals. The P300 speller is a special paradigm of BCI that uses the P300 brain wave for character recognition. It uses the P300 event-related potential (ERP), a positive deflection in the electroencephalogram (EEG) approximately 300 milliseconds after the occurrence of a rare or significant stimulus, to identify the characters flashed on a screen displayed to the user.

The P300 speller relies on the "oddball paradigm," where a user focuses on a target in a 6×6 character matrix while rows and columns are flashed in a random sequence. When the row or column containing the desired character is highlighted, the brain generates a P300 response, which can be detected and classified. However, the accurate detection of P300 signals poses a significant challenge due to the low signal-to-noise ratio (SNR) of EEG data, the variability across sessions and subjects, and the redundancy introduced by multiple EEG channels.

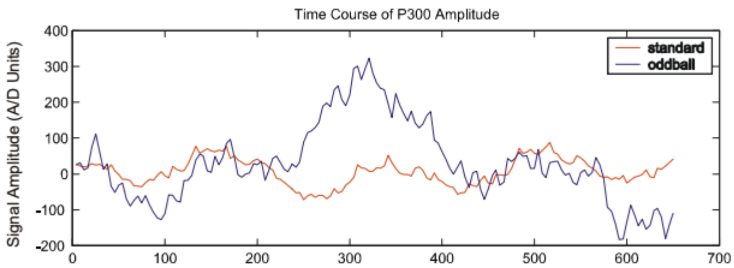


Figure 1: Example time course of an average signal waveform vs. a signal waveform with P300 EPR (at Cz) [1].

To address these challenges, we implement the methodology described in the paper "BCI Competition III: Dataset II - Ensemble of SVMs for BCI P300 Speller" by Rakotomamonjy and Guigue [2]. The paper proposes an approach combining an ensemble of support vector machines (SVMs) and recursive channel elimination to optimize P300 detection. The key goals of the paper are as follows

1. Enhance classification performance by identifying the most relevant EEG channels through recursive elimination.

2. Mitigate the variability in EEG data by using ensemble learning, which aggregates predictions from multiple classifiers trained on subsets of data.

The dataset used in the study, BCI Competition III Dataset II [1], comprises EEG recordings from 64 scalp electrodes during P300 speller experiments.

This report documents the implementation of the methodology outlined in the paper along with a validation of the approach, analysis of performance, and identification of areas of potential improvement. The subsequent sections in the report provide an overview of the paper and dataset, a detailed explanation of the methods and implementation, results achieved, and a discussion on challenges faced and potential improvements.

# 2 Overview

## 2.1 Selected Paper

The paper, "BCI Competition III: Dataset II - Ensemble of SVMs for BCI P300 Speller" by Rakotomamonjy and Guigue [2], presents an approach to addressing the two critical challenges in EEG-based P300 classification. First, being the variability of EEG signals, which poses a significant obstacle, as P300 responses could differ considerably between subjects, sessions, and experimental conditions. Second, being the high dimensionality of EEG data, with 64 channels and multiple time points per sample, which introduces redundancy that could negatively impact classifier performance.

To overcome these challenges, the authors proposed a methodology that combines an ensemble of support vector machines (SVMs) with recursive channel elimination. Recursive channel elimination systematically identifies and retains the most relevant EEG channels, which benefits both classification accuracy and computational efficiency. By removing irrelevant or noisy channels, the dimensionality of the feature space and classifier focus is reduced, thus the accuracy is improved.

The second core component, the ensemble of SVMs, mitigates the variability in EEG signals by training multiple SVM classifiers on different subsets of the data. The predictions from these classifiers are aggregated, which improves both the robustness and generalization by reducing the impact of noisy or outlier samples. Both aforementioned methodologies are discussed in detail in the following section.

The authors validated their approach on Dataset II [1] from the BCI Competition III and demonstrated superior performance, where their method was able to achieve the best classification results in the competition.

## 2.2 Dataset

The dataset used in this implementation is Dataset II [1] from the BCI Competition III, which was recorded during P300 speller experiments. The P300 speller paradigm is based on the "oddball paradigm," where rare and expected stimuli elicit a positive deflection in the EEG approximately 300 milliseconds after stimulus onset. This deflection, known as the P300 component, is a reliable feature present in nearly all humans.

The P300 speller was first introduced by Farwell and Donchin [3], who developed a protocol that presents a 6×6 character matrix to the subject. To spell a character, each of the 12 rows and columns of the matrix is sequentially intensified in a random order, referred to as a sequence. The subject is instructed to focus on the desired character, and when the row or column containing the target character is intensified, a P300 ERP is generated in response. To improve the reliability of the spelling process, the intensification sequence is repeated 15 times for each character.

This dataset contains signals collected from two subjects, referred to as Subject A and Subject B, across five different spelling sessions. During each session, the subjects were asked to spell multiple words, and all EEG signals were recorded from a 64-channel scalp montage. The bandpass-filtering from 0.1–60 Hz was applied on the EEG signals before digitization at a sampling rate of 240 samples per second. Each recorded signal is referred to as a "post-stimulus signal," representing the EEG activity following the intensification of a row or column. The dataset provides both labelled training data and unlabelled test


Figure 2: The user display for this paradigm. In this example, the user's task is to spell the word "SEND" (focusing on one character at a time) [1].

data for this classification task.

The training data consists of signals corresponding to 85 spelled characters for each subject. Each signal associated with the spelling of a single character begins with the flashing of a row or column, lasting 100 ms (24 samples), followed by a blank period of 75 ms (18 samples) during which no row or column is flashed. This pattern continues until all 12 rows and columns are flashed in sequence. The entire sequence is then repeated 15 times for each character spelling, amounting to a total of 15,300 labelled post-stimulus signals per subject.

The goal is the classification of these signals into two categories: those containing the P300 ERP and those without it. This binary classification serves as the first step toward identifying the target character. The second part of the problem involves the prediction of the desired character based on the fewest sequences, which is a 36-class classification problem corresponding to the 6×6 character matrix.

The dataset has the following variables:

- Signal: The EEG data is structured as 85 characters × 7794 samples × 64 channels. Each character's data spans 7794 samples, capturing 15 repetitions of the row/column intensification sequence.

- Flashing: A sequence marking when rows or columns of the matrix were intensified. The sequence alternates between 24 and 18 samples for the flashing and inter-flash periods, with the final chunk consisting of 252 samples.

- StimulusCode: Codes ranging from 1–12, identifying which row or column was intensified at any given time.

- StimulusType: Binary labels indicating the presence (1) or absence (0) of the P300 ERP.

- TargetChar: The 85 target characters spelled during the experiment.

The true labels for the testing dataset are not provided in the dataset, limiting the ability to evaluate performance on unseen data. As a result, the training dataset was used for both training and testing in this implementation.

# 3 Methodology

The methodology for P300 classification is divided into three main stages: data preprocessing, model training, and character prediction, each stage designed to address specific challenges in P300 detection.

## 3.1 Data Preprocessing

The preprocessing of EEG signals is a crucial step for extracting relevant features that highlight the P300 component while minimizing noise. For each intensification, the interval from stimulus onset to approximately 667 ms post-stimulus is selected, as this period is known to contain the P300 signal. This ensures that all required time features are captured for an efficient classification, as this window is large enough. With EEG signals recorded from 64 channels, this approach captures both spatial and temporal dimensions of the data.

To prepare the data for analysis, the raw EEG signals are passed through a Chebyshev bandpass filter. This type of filter is particularly effective for retaining the frequencies associated with the P300 signal (which typically lies within the delta (0.5–4 Hz) and theta (4–8 Hz) bands [4]), while attenuating noise from other frequency bands. Once filtered, the data is decimated to reduce the sampling rate, condensing the high-dimensional signals into a manageable format without losing important information.

Feature vectors are constructed by concatenating the decimated samples across all channels into a single dimension. This representation ensures that spatial and temporal information is preserved in a format suitable for machine learning models.

## 3.2 Model Training

Model training involves the use of support vector machines (SVMs), a robust classification technique known for its efficacy in high-dimensional datasets. SVMs work by finding the hyperplane that best separates data points belonging to different classes. In the context of P300 detection, this is used to distinguish between EEG signals with and without the P300 component.

To improve generalization and mitigate variability in the data, the training data is divided into partitions, and multiple SVM classifiers are trained independently on these subsets. This approach, inspired by the "Bagging" ensemble learning technique (Bootstrap aggregation), helps to reduce the impact of noise and outliers.

The tuning of the regularization parameter C is an essential part of model training. Parameter C controls the trade-off between achieving a low error on the training data and maintaining a simpler decision boundary that generalizes well to unseen data or the test data. A smaller C value places greater emphasis on generalization, tolerating some misclassifications in the training data to avoid overfitting. Conversely, a larger C value focuses on minimizing training error, which can sometimes lead to overfitting. By systematically testing different values of C, the model is fine-tuned to achieve an optimal balance between bias and variance.

Channel selection is a key step in reducing the dimensionality of the feature space while maintaining classification performance. Each channel's contribution to classification accuracy is evaluated, and channels are ranked accordingly.

$$C_{cs} = \frac{tp}{tp + fp + fn} \tag{1}$$

Here, $tp$, $fp$, and $fn$ represent the number of true positives, false positives, and false negatives, respectively. The accuracy score ($C_{cs}$) is used for channel ranking to address the imbalance between true positives and true negatives in the dataset. This ranking provides the relative importance of each channel, indicating the subset of channels that are necessary for reliable P300 detection.

### 3.3 Character Prediction

The final stage of the methodology focuses on using the predictions from the SVM classifiers to accurately identify the focused character. For each character, multiple sequences of row and column intensifications are analyzed, producing post-stimulus decision scores for each intensification. These decision scores indicate the likelihood of a P300 response and are key to identifying the target row and column.

For robustness, a double-averaging strategy is employed. First, decision scores from all classifiers are averaged for each intensification, mitigating the impact of poor scores from individual classifiers. Then, the decision scores across multiple sequences are averaged for each row and column, further reducing noise and variability.

The row and column with the highest decision scores are identified, and their intersection in the 6×6 character matrix determines the predicted character. The detailed implementation of this methodology, along with the numerical values used, is presented in the following section.

# 4 Implementation Details

The implementation of the paper begins with data preprocessing. The raw dataset for Subject A (or B) is in the format (85, 7794, 64), where 85 represents different focused characters and 7794 are samples obtained across 64 EEG channels. Each sample contains 12 row/column intensifications repeated 15 times, interspersed with blank samples.

## 4.1 Data - Preprocessing

### 4.1.1 Windowing

First, these blanks are removed, resulting in a dataset of size (85, 180, 64). For each of the 180 intensifications of each character, the interval from 0 to 667 ms after each intensification is extracted per 64 channels, as this interval is expected to contain the P300 signal. With a sampling rate of 240 Hz, this interval contains 160 samples per channel.

### 4.1.2 Filtering

An 8th-order Chebyshev bandpass filter (0.1–10 Hz) mentioned in the paper was initially used to denoise the signal. However, during implementation, we encountered an issue where the higher-order filter caused transient outputs. To resolve this, the filter order was reduced to 4, which eliminated the problem. As described in the paper, the samples were then decimated, reducing the data from 160 samples to 14 samples per channel.

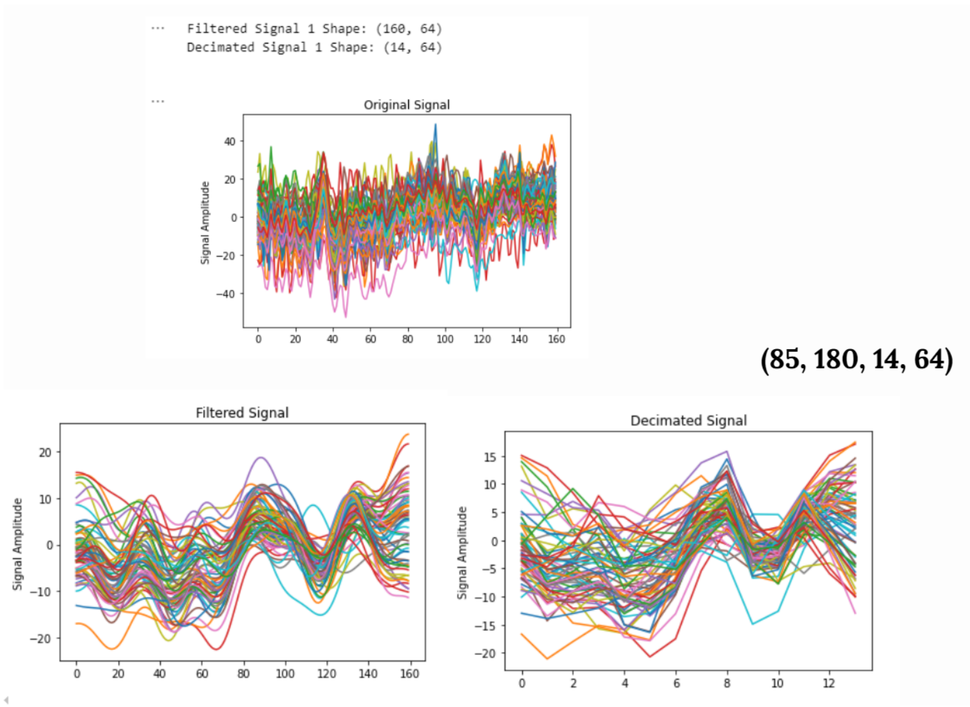


Figure 3: Filtered and Decimated dataset

#### 4.1.3 Feature Vectoring

The feature vector was created by combining the 14 samples from all 64 channels into a single dimension, resulting in a feature size of (896,).

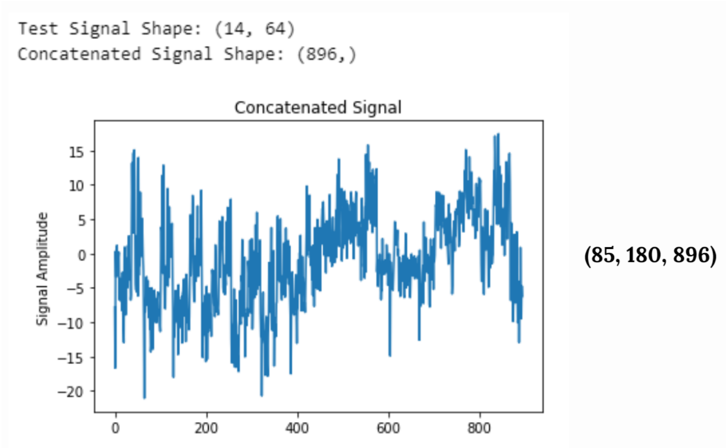


Figure 4: Concatenated Feature vector

Following preprocessing, model training was performed. However, the test data provided in the original paper could not be used for testing our model performance as it was unlabeled. The competition may have used this test data for evaluation purposes. Therefore, we had to partition the training data into separate training and testing sets.

### 4.2 Data - Partitioning

The dataset of 85 character epochs was split into 75 epochs for training (13,500 samples) and 10 epochs for testing (1,800 samples). For the 75 training character epochs, the data was further divided into 15 partitions, each consisting of 5 consecutive character epochs.

### 4.3 Model Training

The partitioned training data is used to train Support Vector Machine (SVM) classifiers. Fifteen SVM classifiers are trained, each on one of the 15 partitions. During training, the C parameter is treated as a hyperparameter. The paper indicates that not all 64 channels are required to detect the P300 signal, so hyperparameter tuning is performed to fine-tune the SVM classifiers.

### 4.4 Channel Ranking

To determine the optimal number of channels needed, each channel is removed individually, and its effect on accuracy is evaluated. After recording the accuracy values, the least significant channel is removed, and the process is repeated with the remaining subset of channels. This iterative process ranks the channels from least to most effective. However, as noted in the paper, this method does not account for the interactions of channels removed earlier and is computationally expensive.

We encountered the same problem and addressed it by implementing a simplified channel ranking cycle. Instead of iterative removal, the ranking process was run once, and thus obtained accuracy values were used to rank each channel. This reduced the computational complexity from O(n!) to O(n), where n is the no.of channels, though it still required more computational resources than anticipated.

## 4.5 Channel Selection

Once the channels are ranked, the most effective channels are added one by one, and the model's accuracy is evaluated at each step. A graph is plotted to identify the optimal number of channels that yields the maximum accuracy. However, we were unable to produce a graph with a clear peak as described in the paper.

## 4.6 C Parameter Fine-Tuning

To determine the optimal value for the C parameter, a matrix of values, C = [ 0.01, 0.05, 0.1, 0.5, 1.0], is used. The value of C that results in the highest accuracy is selected for the final model.

## 4.7 Character Prediction

For each character epoch, 180 post-stimulus vectors (12 rows/columns $\times$ 15 sequences) are analyzed. Each SVM assigns a decision score to each post-stimulus vector. These scores are then processed through a double-averaging approach.

First, the decision scores from all classifiers for each post-stimulus vector are averaged, improving robustness and correcting poor scores from individual classifiers. Next, the decision scores of multiple sequences for each row and column are averaged, resulting in a single score for each row and column.

The final character prediction is made by selecting the row and column with the highest scores. The intersection of these two identifies the predicted character.

# 5 Implementation Results

As we tested our data using a partition from the training data itself, we were unable to directly compare our results with those presented in the paper. However, we successfully trained 15 SVMs using the given dataset. By ensembling the SVMs, we achieved an accuracy of 80.5% (tested with limited data) without any fine-tuning (fixing the C parameter as 0.01 and using all 64 channels).

| Subject A | Subject B |
|---|---|
| `Ensemble SVM accuracy on test data: 80.50%` | `Ensemble SVM accuracy on test data: 81.00%` |

Figure 5: SVM accuracy before fine tuning

Using the channel ranking method, we ranked the most effective channels, as illustrated in the figure below for the 15 SVMs. The top 12 ranked channels are shown. In the paper, this analysis was only performed for 5 partitions. Notably, our results showed different ranked channels compared to the paper. This discrepancy may be attributed to using a smaller data sample from the training set for testing.

```
['PO8', 'CP5', 'Cz', 'CPz', 'POz', 'TP8', 'F7', 'FC1', 'F3', 'F4', 'Pz', 'Fz']
['Pz', 'F2', 'P7', 'T8', 'POz', 'FC3', 'PO7', 'AF3', 'C1', 'F1', 'CPz', 'F7']
['T7', 'T8', 'C2', 'O2', 'C3', 'C5', 'Oz', 'P6', 'F8', 'P7', 'CP6', 'PO7']
['P7', 'P8', 'F2', 'Oz', 'C3', 'Iz', 'POz', 'CP6', 'F7', 'Pz', 'PO7', 'FC2']
['C4', 'Fp1', 'PO7', 'POz', 'P7', 'T7', 'CP3', 'FCz', 'Pz', 'O1', 'F8', 'Cz']
['AF4', 'POz', 'Fp2', 'CP4', 'Iz', 'FC2', 'T10', 'FT8', 'CP1', 'F5', 'P7', 'FT7']
['FC6', 'O1', 'C2', 'CP3', 'F7', 'Fz', 'PO7', 'O2', 'T7', 'P6', 'Oz', 'Fp2']
['AF4', 'F7', 'P5', 'F1', 'CP4', 'O1', 'AF3', 'T7', 'PO8', 'F6', 'P7', 'P6']
['FT8', 'POz', 'PO8', 'FC1', 'PO3', 'F6', 'P6', 'FC5', 'T10', 'PO4', 'AF4', 'P3']
['P7', 'C5', 'AF8', 'CP6', 'AF4', 'F5', 'C3', 'CP3', 'Pz', 'F8', 'FT8', 'FCz']
['P1', 'C2', 'C1', 'Pz', 'Fp2', 'Cz', 'POz', 'PO7', 'CP2', 'FC3', 'CP3', 'F5']
['Cz', 'POz', 'TP7', 'F7', 'C3', 'PO8', 'CP1', 'Iz', 'F5', 'O1', 'Fp1', 'TP8']
['P2', 'T10', 'F4', 'Pz', 'FT7', 'P7', 'F8', 'CP5', 'P5', 'P6', 'PO3', 'FC6']
['PO8', 'PO7', 'POz', 'F5', 'P4', 'PO3', 'F8', 'O2', 'FC5', 'FCz', 'P8', 'T9']
['PO7', 'P7', 'F6', 'AF3', 'F8', 'Cz', 'CP5', 'FC1', 'AF7', 'TP8', 'CP2', 'C6']
```

Figure 6: Top 12 channels for each 15 partitions in order.

| Data | 12 Top Ranked Channels | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A1 | $FC_1$ | $C_2$ | $CP_3$ | $CP_z$ | $F_z$ | $F_4$ | $F_6$ | $P_5$ | $P_z$ | $P_8$ | $PO_7$ | $PO_8$ |
| A2 | $C_1$ | $CP_z$ | $CP_4$ | $AF_7$ | $AF_z$ | $F_z$ | $F_8$ | $P_5$ | $P_z$ | $PO_7$ | $PO_z$ | $PO_8$ |
| A3 | $FC_2$ | $CP_5$ | $CP_1$ | $F_1$ | $F_z$ | $FT_8$ | $T_7$ | $P_7$ | $P_5$ | $P_z$ | $PO_7$ | $PO_8$ |
| A4 | $C_3$ | $C_1$ | $F_{P1}$ | $F_2$ | $F_4$ | $F_6$ | $TP_7$ | $P_7$ | $P_5$ | $P_z$ | $PO_7$ | $PO_8$ |
| A5 | $C_z$ | $CP_5$ | $CP_2$ | $F_7$ | $F_8$ | $P_7$ | $P_z$ | $P_4$ | $P_8$ | $PO_7$ | $PO_4$ | $PO_8$ |

Figure 7: Top 12 channels of first 5 partitions according to the paper

Next, we added channels one by one in the order of their priority and plotted the $C_{cs}$ accuracy (equation 1). However, we were only able to perform this analysis for one partition due to computational limitations. In the first partition, accuracy peaked when 51 channels were used (a sufficient peak was observed when 12 channels were used).

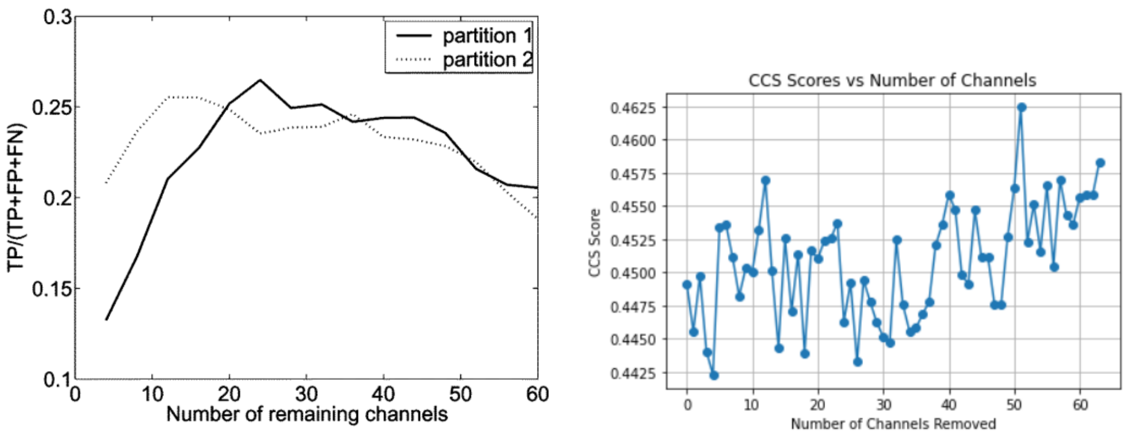


Figure 8: $C_{cs}$ accuracy vs. Number of channels. Result from our implementation (left) and from paper (right)

The twelve channels were selected, and with the optimized set of channels, we determined the optimal value of parameter C by generating a similar plot while varying the parameter C to different pre-determined values ([0.01, 0.05, 0.1, 0.5, 1.0]). However, for each C value, we got almost the same accuracy values.

```
CCS scores for each C values are,
[[1240  260]
 [ 204   96]]
CCS for C=0.01: 0.44956405097246505
[[1240  260]
 [ 204   96]]
CCS for C=0.05: 0.44956405097246505
[[1240  260]
 [ 204   96]]
CCS for C=0.1: 0.44956405097246505
[[1240  260]
 [ 204   96]]
CCS for C=0.5: 0.44956405097246505
[[1240  260]
 [ 204   96]]
CCS for C=1: 0.44956405097246505
```

Figure 9: $C_{cs}$ accuracy vs C parameter

Finally, the fine-tuned model achieved an accuracy of 85.4%. This is lower than the 96.5% accuracy reported in the paper. We believe this discrepancy was raised due to the limited data available for testing which also resulted in different sets of ranked channels obtained during our implementation.

```
SVM 1 accuracy on test data after fine tuning: 85.40%
```

Figure 10: SVM 1 accuracy after fine-tuning

We were unable to calculate the ensemble SVM accuracy because running the algorithm on the entire dataset required substantial computational power. In addition to the provided results, the paper compares their results with those of other algorithms, fixed number of channel vs. the channel ranking method etc.

For character prediction, we utilized the full 64-channel system with the ensemble SVM. Using this setup, we achieved the following results. Note that only 10 characters were in the testing dataset we created.

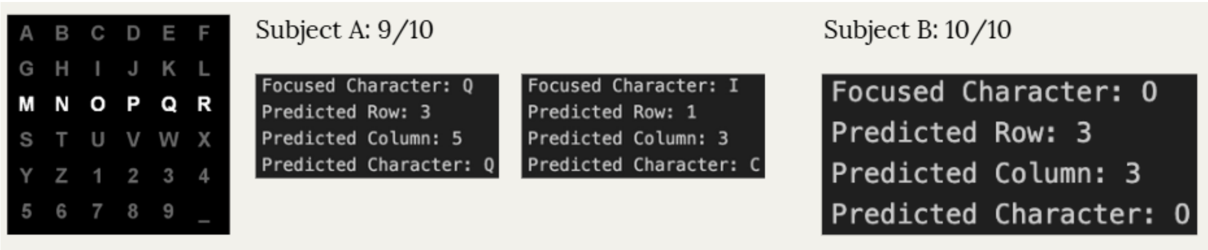


Figure 11: Character Prediction

This completes the set of results obtained from our implementation. Despite some limitations, such as computational constraints and the use of a smaller testing dataset, we were able to replicate key aspects of the methodology. By utilizing the full 64-channel system with the ensembled SVM, we achieved meaningful insights and comparable outcomes for the character prediction task, even though the testing was limited to 10 characters.

# 6 Conclusion and Future Improvements

The methodology implemented for P300 detection and character recognition integrates signal processing, machine learning, and ensemble techniques to address the major challenges associated with EEG-based classification. Data preprocessing ensures the extraction of meaningful features, model training utilises SVMs and ensemble methods for robust classification, and the character prediction process provides a reliable framework for identifying target characters. The approach demonstrated strong performance on the provided training dataset, showcasing its potential for use in P300 speller systems. However, there are several areas where the methodology can be refined to improve efficiency, accuracy, and generalizability.

**Future Improvements:**

**Channel ranking procedure**

The channel ranking process iteratively removes the least significant channels but does not revisit previously eliminated channels. This limitation fails to account for potential interactions between channels, which might change their relative importance during the process. Future work could explore dynamic ranking methods that re-evaluate all channels at each iteration.

**Testing accuracy derived from training set**

Due to the lack of true labels for the provided testing dataset, accuracy was evaluated using a partitioned subset of the training data. This limits the ability to assess the model's generalization capabilities on unseen data. Future work could incorporate additional datasets with labelled test samples to provide a more robust evaluation.

**Impact of dataset partitioning on ensemble classifier performance**

The performance of the ensemble classifiers is heavily influenced by how the training set is partitioned. Training classifiers on specific subsets of the data space versus diverse samples covering the entire space could lead to different levels of performance. Investigating and optimizing the partitioning method could enhance the method's ability to generalize.

**Inter-subject variability**

The current algorithm is trained and tested exclusively on data from a single subject without addressing inter-subject variability. Since EEG signals vary significantly across individuals, this limits the algorithm's applicability to new users. Tackling inter-subject learning by incorporating transfer learning techniques or domain adaptation methods is essential.

By addressing these areas, the methodology can be enhanced to become more efficient, accurate, and general. These improvements will pave the way for broader applications of P300 speller systems, making them more practical for real-world scenarios and diverse user populations.

# References

[1] "BCI Competition III." [Online]. Available: https://www.bbci.de/competition/iii/#data_set_ii

[2] A. Rakotomamonjy and V. Guigue, "BCI Competition III: Dataset II- Ensemble of SVMs for BCI P300 Speller," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 3, pp. 1147–1154, 2 2008. [Online]. Available: https://doi.org/10.1109/tbme.2008.915728

[3] L. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalography and Clinical Neurophysiology*, vol. 70, no. 6, pp. 510–523, 12 1988. [Online]. Available: https://doi.org/10.1016/0013-4694(88)90149-6

[4] V. Kolev, T. Demiralp, J. Yordanova, A. Ademoglu, and Isoglu-Alkaç, "Time–frequency analysis reveals multiple functional components during oddball P300," *Neuroreport*, vol. 8, no. 8, pp. 2061–2065, 5 1997. [Online]. Available: https://doi.org/10.1097/00001756-199705260-00050