Mid-Term Evaluation Report

EEG-based P300 Detection using Machine Learning

Project Summary

The main objective of this project is to study EEG signals and detect the P300 response using machine learning techniques. EEG (Electroencephalography) is used to measure the electrical activity of the brain through electrodes placed on the scalp. Among various EEG responses, the P300 is an important Event-Related Potential (ERP) that occurs when a subject identifies a rare or meaningful stimulus. This response plays a key role in Brain-Computer Interface (BCI) systems such as P300 spellers.

In this project, EEG data is first preprocessed and filtered to reduce noise. The signals are then segmented into epochs around stimulus events. Relevant features are extracted from these epochs and used to train supervised machine learning models to classify target and non-target events. The overall aim is to build a basic end-to-end pipeline that combines EEG signal processing with machine learning-based classification.

Work Completed

EEG Signal Understanding and ERP Analysis

The project began with a detailed study of how EEG signals are generated and recorded. Brain activity originates from neurons communicating through electrical impulses. When large groups of cortical neurons fire synchronously, they generate electric fields that propagate through brain tissue, skull, and scalp. These weak electric fields are captured by electrodes placed on the scalp following standardized electrode placement systems. The recorded signals are then amplified, digitized, and stored as EEG signals for further processing. This complete pipeline can be summarized as: brain activity leads to neuronal firing, which creates electric fields that are detected by electrodes, amplified, and finally converted into digital signals.

Different EEG frequency bands were studied to understand their relevance to brain states. Delta waves (0.5–4 Hz) are mainly associated with deep sleep, theta waves (4–8 Hz) correspond to drowsiness and light sleep, alpha waves (8–13 Hz) are prominent during

relaxed states with eyes closed, beta waves (13–30 Hz) are related to active thinking and problem-solving, and gamma waves (above 30 Hz) are linked to fast cognitive processing. Since the P300 response is associated with attention and cognitive evaluation, frequency components in the alpha and beta ranges are especially important for this project.

Event-Related Potentials (ERPs) were then studied as time-locked brain responses to specific stimuli. The P300 component is a positive voltage peak that typically occurs around 300 milliseconds after a target stimulus is detected. This delay does not represent sensory perception, but rather the time taken by the brain to evaluate and recognize a meaningful event. In P300 speller systems, this timing is crucial because it allows differentiation between target and non-target stimuli based on attentional response, making the P300 a reliable marker for decision-making and stimulus recognition.

Data Preprocessing and Filtering

In general, data preprocessing is a crucial step in any machine learning pipeline because raw data is often noisy, inconsistent, and unsuitable for direct model training. Preprocessing helps in removing irrelevant information, reducing noise, and transforming data into a structured and meaningful form. Without proper preprocessing, even well-designed machine learning models can perform poorly.

In the context of this project, preprocessing is especially important because EEG signals are extremely low in amplitude (in the order of microvolts) and are highly sensitive to various types of noise such as eye blinks, muscle movement, and electrical interference. Since the goal of this project is to detect the P300 response, which is a subtle cognitive signal, careful preprocessing is necessary to enhance the signal-to-noise ratio and preserve meaningful brain activity.

The preprocessing pipeline used in this project consists of the following steps:

Step 1: Loading EEG Data

EEG data was loaded using specialized neurophysiological signal processing libraries that support standard EEG file formats. Libraries such as `MNE-Python` were used because they provide built-in functions for handling EEG signals efficiently.

```
import mne
raw = mne.io.read_raw_fif('data.fif', preload=True)
```

Step 2: Band-pass Filtering

Band-pass filtering was applied to retain only the frequency components relevant to cognitive activity and ERP analysis. This helps remove slow drifts and high-frequency noise that do not contribute to the P300 response.

```
raw.filter(l_freq=0.1, h_freq=30)
```

Step 3: Epoch Extraction

The continuous EEG signal was segmented into smaller time windows called epochs, centered around stimulus events. Each epoch captures the brain response before and after a stimulus, allowing analysis of event-related potentials such as the P300.

```
epochs = mne.Epochs(raw, events, tmin=-0.2, tmax=0.8)
```

Step 4: Baseline Correction

Baseline correction was performed to normalize EEG signals by subtracting the mean of the pre-stimulus period. This helps in reducing variability across trials and improves comparability between epochs.

```
epochs.apply_baseline((None, 0))
```

These preprocessing steps ensured that the EEG data used for feature extraction and model training was clean, structured, and focused on the relevant cognitive responses required for P300 classification.

Feature Extraction

Feature extraction is the process of transforming raw or preprocessed data into a set of meaningful numerical features that can be used by machine learning models. Since EEG signals are time-series data with high dimensionality, directly feeding raw signals into classical machine learning models is inefficient and often ineffective. Feature extraction helps in reducing dimensionality while preserving the most discriminative information.

In this project, feature extraction was guided by the known characteristics of the P300 response. The P300 appears as a positive voltage deflection approximately 300 ms after a target stimulus. Therefore, features were extracted from specific time windows around this region, allowing the model to distinguish between target and non-target EEG epochs.

The primary steps involved in feature extraction are described below:

Step 1: Selecting the P300 Time Window

A time window centered around the expected P300 latency was selected from each epoch. This window captures the most relevant portion of the EEG signal for classification.

```
p300_window = epochs.copy().crop(tmin=0.25, tmax=0.45)
```

Step 2: Amplitude-based Feature Extraction

Mean amplitude values were computed from the selected time window across relevant EEG channels. These values serve as direct indicators of the presence or absence of the P300 component.

```
import numpy as np
features_mean = np.mean(p300_window.get_data(), axis=2)
```

Step 3: Statistical Feature Extraction
In addition to mean amplitude, simple statistical features such as variance were calculated to capture signal variability within the P300 window.

```
features_var = np.var(p300_window.get_data(), axis=2)
```

Step 4: Feature Vector Construction
The extracted features were concatenated to form a final feature vector for each epoch, which was then used as input to the machine learning classifiers.

```
X = np.concatenate((features_mean, features_var), axis=1)
```

This feature extraction approach effectively bridges domain knowledge from neuroscience with data-driven machine learning. By focusing on biologically meaningful time windows and simple statistical measures, the extracted features enabled the classification models to learn patterns associated with the P300 response while keeping the overall system computationally efficient.

Machine Learning models

Machine learning models can be broadly categorized into supervised, unsupervised, and reinforcement learning. In supervised learning, the model is trained using labeled data, where both input features and corresponding output labels are known. The objective is to learn a mapping that can correctly predict labels for unseen data. Common examples include classification and regression tasks.

Unsupervised learning, on the other hand, works with unlabeled data. The model attempts to discover inherent patterns or structures in the data, such as clustering or anomaly detection. While useful for exploratory analysis, unsupervised methods do not directly provide class predictions. Reinforcement learning involves an agent interacting with an environment and learning through rewards or penalties based on its actions, making it suitable for sequential decision-making problems.

In this project, supervised learning was chosen because labeled EEG data was available, where each epoch was marked as either a target (P300) or non-target response. The objective was to classify EEG segments based on the presence of the P300 component, which naturally fits a supervised classification framework. Models such as Logistic Regression and Support Vector Machines were used, as they are well-suited for binary classification and provide interpretable results. The implementation involved extracting features from preprocessed EEG epochs, splitting the data into training and testing sets, and evaluating model performance using standard classification metrics.

Results and Evaluation

Evaluating a machine learning model is essential to understand how well it performs on unseen data. In classification problems, especially those involving imbalanced datasets such as EEG-based P300 detection, relying on a single metric can be misleading. Therefore, multiple evaluation metrics were used in this project to obtain a comprehensive view of model performance.

Accuracy

Accuracy measures the proportion of correctly classified samples out of the total number of samples. It is defined as:

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

where TP represents true positives, TN true negatives, FP false positives, and FN false negatives.

In this project, accuracy indicates how many EEG epochs were correctly classified as either target (P300) or non-target. However, since non-target samples are much more frequent than target samples in a P300 speller paradigm, accuracy alone is not sufficient to judge performance.

Precision

Precision measures how many of the samples predicted as positive are actually positive. It is defined as:

$$Precision = \frac{TP}{TP + FP}$$

In the context of this project, precision reflects how many of the EEG epochs predicted to contain a P300 response truly correspond to target stimuli. High precision indicates that the model produces fewer false alarms.

Recall

Recall, also known as sensitivity, measures how many of the actual positive samples are correctly identified. It is defined as:

$$Recall = \frac{TP}{TP + FN}$$

For P300 detection, recall represents the model's ability to correctly detect target stimuli that evoke a P300 response. A low recall would mean that many true P300 events

are missed, which is undesirable in BCI applications.

F1 Score

The F1 score is the harmonic mean of precision and recall, providing a balance between the two. It is defined as:

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

In this project, the F1 score was particularly useful because it accounts for both false positives and false negatives. This makes it a more reliable metric than accuracy for evaluating P300 classification performance on imbalanced data.

Area Under the ROC Curve (AUC)

The Area Under the Receiver Operating Characteristic (ROC) Curve measures the model's ability to distinguish between classes across different classification thresholds. It can be interpreted as the probability that the model ranks a randomly chosen positive sample higher than a randomly chosen negative sample.

An AUC value close to 1 indicates strong discriminative ability, while a value close to 0.5 suggests random guessing. In the context of this project, a higher AUC indicates better separation between target and non-target EEG epochs, regardless of the chosen decision threshold.

Project Evaluation Summary

Using the above metrics, the trained models achieved moderate but promising performance. F1 and AUC scores indicating reasonable discrimination between target and non-target classes. These results demonstrate that even simple supervised learning models can capture meaningful patterns in EEG data when combined with appropriate preprocessing and feature extraction. The evaluation also highlights areas for improvement, such as handling class imbalance and refining feature selection, which will be addressed in future work.

Code Links

The project code and experiments are organized in a GitHub repository. The important files are listed below:

- Preprocessing and filtering Notebook: `https://github.com/electricalengineersiitk/Winter-projects-25-26/blob/main/EEG-Based%20P300%20Speller/assignments/assignment_2/240330_Deepak_assgn2.ipynb`

```
===== Model Comparison === Development Mode Activated!!!
LDA                  | Acc: 0.583 | F1: 0.320 | Prec: 0.219 | Recall: 0.591
Logistic Regression  | Acc: 0.567 | F1: 0.323 | Prec: 0.218 | Recall: 0.622
SVM                  | Acc: 0.519 | F1: 0.267 | Prec: 0.179 | Recall: 0.528
Random Forest        | Acc: 0.830 | F1: 0.000 | Prec: 0.000 | Recall: 0.000
Gradient Boosting    | Acc: 0.593 | F1: 0.254 | Prec: 0.183 | Recall: 0.417
```

Figure 1: Result on model training

-
- Model Training Notebook: `https://github.com/deepakchuahan/Winter-projects-25-26/blob/main/EEG-Based%20P300%20Speller/assignments/assignment_3/EEG_assignment3_Deepak_240330.ipynb`

Results and Observations

Results

The following results were obtained from classical machine learning models:

- **LDA Accuracy:** 58.3%, **F1 Score:** 0.32, **Recall:** 0.59
- **Logistic Regression Accuracy:** 56.7%, **F1 Score:** 0.32, **Recall:** 0.62
- **SVM Accuracy:** 51.9%, **F1 Score:** 0.27
- **Random Forest Accuracy:** 83.0% (but F1 and Recall = 0)
- **Gradient Boosting Accuracy:** 59.3%, **F1 Score:** 0.25

Observations

- Linear models such as LDA and Logistic Regression performed more consistently for P300 detection compared to tree-based models.
- Random Forest achieved high accuracy due to class imbalance but failed to detect target P300 trials, resulting in zero recall and F1 score.
- Recall values were higher than precision, indicating that models detected many target trials but also produced false positives.
- Overall performance was limited by low signal-to-noise ratio and class imbalance inherent in EEG P300 data.

What Worked Well

- Band-pass filtering and epoch averaging enhanced the P300 component.
- ERP-based temporal features were effective for linear classifiers.
- Supervised learning models were able to learn meaningful patterns after preprocessing.

What Did Not Work Well

- Tree-based models struggled with highly imbalanced EEG data.
- Accuracy alone was misleading and did not reflect true model performance.
- Feature representation requires further refinement for improved classification.

Name: Deepak Kumar Chauhan
Roll Number: *(240330)*