MTC-AIC-3 System Description: BCI Processor

Team Name: Synaptic Squad
Team Leader: Abdelrahman Ahmed Mahmoud Eldaba
Contact Email: abdelrahmaneldaba123@gmail.com

Abstract

This document describes our submission for the MTC-AIC-3 competition, presenting a Brain-Computer Interface (BCI) processor for classifying Motor Imagery (MI) and Steady-State Visually Evoked Potential (SSVEP) tasks. Our system leverages advanced preprocessing, feature extraction, and machine learning techniques to achieve robust performance. We detail the system architecture, preprocessing steps, feature extraction methods, challenges faced, and solutions implemented.

Introduction

The MTC-AIC-3 competition requires classifying EEG signals for MI (Left vs. Right) and SSVEP (Forward, Backward, Left, Right) tasks. Our solution, implemented in the BCI-Processor Python package, combines Filter Bank Common Spatial Patterns (FBCSP), Independent Component Analysis (ICA), extended Canonical Correlation Analysis (CCA), and robust machine learning models to achieve high accuracy.

System Architecture

Our system is modular, comprising preprocessing, feature extraction, and modeling components, integrated via the EnhancedBCIProcessor class. Key components include:

- Preprocessing: Data loading, subject-specific normalization, ICA-based artifact removal, notch and bandpass filtering.
- Feature Extraction: Time-domain, frequency-domain, wavelet, and spatial features for MI; frequency, CCA, SNR, and phase features for SSVEP.
- Modeling: Linear Discriminant Analysis (LDA) for MI and Gradient Boosting with hyperparameter tuning for SSVEP.

The system is configured via a “config.yaml” file, allowing flexible parameter adjustments.

Preprocessing

We preprocess EEG data as follows:

- Data Loading: Load trial-specific EEG data from CSV files, handling variable trial lengths.
- Normalization: Apply subject-specific normalization to standardize channel data.
- ICA: Use FastICA to remove artifacts, zeroing out the first component.
- Notch Filter: Remove 50 Hz power line interference.
- Bandpass Filtering: Apply multiple bands (4-8, 8-12, 12-16, 16-20, 20-30 Hz) for MI and 5-50 Hz for SSVEP using a 6th-order Butterworth filter.

Feature Extraction

Motor Imagery (MI)

We extract features from motor cortex channels (C3, CZ, C4):

- Time-Domain: Mean, variance, skewness, kurtosis, percentiles, mobility, complexity.
- Frequency-Domain: Power spectral density (PSD) in alpha (8-12 Hz), beta (13-30 Hz), and mu (8-13 Hz) bands, spectral entropy.
- Wavelet: Daubechies-4 wavelet coefficients (mean, std, energy ratios).
- FBCSP: Simplified Common Spatial Patterns using covariance-based spatial filters.
- Cross-Channel: C3-C4 power ratio for lateralization. Total features: 60 (selected via SelectKBest).

SSVEP

We focus on occipital channels (PO7, OZ, PO8):

- Frequency-Domain: Power at target frequencies (7, 8, 10, 13 Hz) and harmonics, harmonic ratios.
- Extended CCA: Correlation with reference signals (sine, cosine, phase-shifted).
- SNR: Signal-to-noise ratio at target frequencies.
- Phase: Phase-locking value and circular variance. Total features: 120 (selected via SelectKBest).

Modeling

MI Model: Linear Discriminant Analysis with SVD solver and auto-shrinkage.
SSVEP Model: Gradient Boosting Classifier with RandomizedSearchCV for hyperparameter tuning (nestimators, maxdepth, learningrate, subsample). Training uses robust scaling and feature selection to handle high-dimensional data.

Challenges and Solutions

- **Challenge:** CSP indexing error due to label mismatch.
  Solution: Simplified CSP to use trial-specific covariance matrices, removing label dependency during testing.

- **Challenge:** Noisy EEG data and artifacts.
  Solution: ICA-based artifact removal and robust scaling.

- **Challenge:** Low SSVEP accuracy.
  Solution: Extended CCA with phase-shifted references and hyperparameter tuning.

Reproducibility

The submission includes:

- Code: Modular Python package (bci_processor) with preprocessing, feature extraction, and modeling modules.
- Configuration: config.yaml for all parameters.
- Scripts: run_bci_processor.py for end-to-end execution.
- Dependencies: requirements.txt for environment setup.
- Tests: Unit tests in tests/ for validation.

Performance

Validation accuracies:
MI: 0.6000
SSVEP: 0.4600
Overall: 0.5300

Conclusion

Our BCI-processor provides a robust, modular solution for EEG classification, addressing the MTC-AIC-3 requirements with advanced signal processing and machine learning techniques. Future improvements could include Task-Related Component Analysis (TRCA) or deep learning models for SSVEP.

Look at the GitHub Repo:
https://github.com/Abdelrahman47-code/bci-processor-mtc-aic3