

NeuroPAC: A Pre-Processing, Analysis, and Classification Pipeline Platform for EEG Signals

Selin Uygun

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Bachelor of Science in Software Engineering

Department of Software Engineering
Mugla Sitki Kocman University

Abstract

(Abstract text goes here...)

Contents

List of Tables

List of Figures

Chapter I

Introduction

I.1 Problem Statement

The core difficulty in Electroencephalography (EEG) data analysis lies in the trade-off between power/flexibility and usability/accessibility. Here are some limitations of two existing and most popular state of the art tools that are used for EEG pre-processing and analysis:

- **Fieldtrip, Complexity of Code-Based Toolboxes:** A high-performance, flexible toolbox like FieldTrip is essential for advanced analysis but rely on MATLAB code and scripting. This necessitates a high level of programming expertise, creating a significant barrier to entry and slowing down the research process for those whose primary focus is neuroscience or clinical application rather than coding and data structuring.
- **EEGLAB, Existing GUI Complexity:** While EEGLAB offers a Graphical User Interface (GUI), this interface can present a dense, overwhelming array of parameters, menus, and advanced options to a novice. This complexity often fails to make the fundamental, necessary workflow—preprocessing, processing, and classification—feel less intimidating or more welcoming to beginners.

There is a clear and persistent need for a dedicated, simplified interface that explicitly targets the essential and initial steps of EEG analysis, thereby democratizing the process.

- **Need for Simplified Workflow (Engineering Goal):** The study addresses the need for a streamlined, intuitive Graphical User Interface (GUI) designed to guide users through

the crucial, initial steps of preprocessing, processing, and machine learning classification in a less intimidating manner. By building on the powerful computational back-end of toolboxes like FieldTrip, this GUI will abstract away the code, presenting only the core configuration options needed for a reproducible, foundational analysis.

- **Need for Clinical Validation and Application (Research Goal):** The efficacy of this simplified pipeline must be demonstrated in a real-world, high-stakes context. There is a need to validate this new streamlined analysis tool by successfully applying it to challenging neurological datasets, specifically in generating robust and clinically meaningful machine learning and deep learning classification results for conditions such as Parkinson’s Disease and mild Traumatic Brain Injury (mTBI).

In summary, the basic difficulty is the lack of a user-friendly tool that simplifies the essential EEG analysis pipeline (preprocessing, feature extraction analysis, and classification). Existing tools are either highly code-dependent (FieldTrip) or still too complex for beginners (EEGLAB). The perceived need is thus the development of a simplified, welcoming user interface that leverages existing powerful toolboxes to make foundational EEG analysis accessible, efficient, and reproducible, with its utility confirmed through successful application and validation in classifying clinical cohorts like Parkinson’s Disease and mTBI patients.

I.2 Purpose of the Study

The purpose of this study is two-fold: to address the usability gap in existing neuroscientific software and to validate the resulting platform through compelling clinical application.

- **Engineering Goal (Product):** To develop a user-centric Graphical User Interface (GUI) using Python that provides a streamlined, accessible, and intuitive workflow for the fundamental steps of EEG data analysis (preprocessing, processing, and classification). This GUI will abstract the complexities of code-based toolboxes like FieldTrip, thereby lowering the technical barrier to entry and welcoming new users to the field of neuroscience and EEG analysis.

- **Research Goal (Practical Outcomes):** To validate the utility and efficacy of the developed GUI and its integrated machine learning classification pipeline by generating meaningful, publishable findings. This includes providing example EEG data outcomes, specifically robust analysis results and accurate classification metrics for clinical datasets concerning Parkinson’s Disease and mild Traumatic Brain Injury (mTBI).

The ultimate goal is to deliver a practical product that enhances the efficiency of the EEG analysis pipeline and produces high-quality, reproducible research outcomes.

I.3 Research Questions

- **Usability and Accessibility:** Can a dedicated Python-based Graphical User Interface (GUI) be successfully developed to simplify the complex, code-based commands of established EEG toolboxes (e.g., FieldTrip), thereby providing an intuitive and streamlined workflow for the fundamental steps of preprocessing and processing?
- **Comparative Advantage:** How does the proposed streamlined GUI workflow compare to existing EEG analysis interfaces (e.g., EEGLAB) in terms of reducing complexity and improving user efficiency for individuals new to neuroscience and EEG data analysis?
- **Classification Efficacy (Parkinson’s Disease):** What level of classification accuracy can the integrated machine learning classification pipeline achieve in reliably distinguishing EEG data from individuals with Parkinson’s Disease compared to healthy controls?
- **Clinical Differentiation (mTBI):** Can the complete EEG analysis and classification platform successfully extract meaningful neurophysiological features and generate differentiating results to classify patients with mild Traumatic Brain Injury (mTBI) from control subjects?
- **Data Outcome Utility:** Does the developed platform effectively output clear, reproducible analysis results and visualization products (e.g., graphs, classification metrics) that aid neuroscientists and clinicians in interpreting the EEG data outcomes from clinical populations?

I.4 Definition of Terms

(Content...)

I.5 Assumptions and Limitations of the Study

- **Essential Software Environment:** The core functionality of the GUI relies on proprietary and third-party software being present. It is fundamentally assumed that the end-user has a valid license for MATLAB and that both MATLAB and the FieldTrip toolbox are correctly installed, configured, and accessible within the operating system environment. Without these prerequisites, the computational functions handled by the GUI cannot execute.
- **FieldTrip and MATLAB Reliability:** It is assumed that the underlying algorithms and functions within the FieldTrip toolbox are accurate, well-tested, and functioning correctly as documented by their developers. The study focuses on the interface and workflow, not on re-validating the internal processing logic of these established tools.
- **Data Quality:** It is assumed that the clinical EEG datasets used for validation (Parkinson's Disease and mTBI) were collected using standardized, ethical procedures and possess sufficient signal quality to yield meaningful, artifact-reduced data after the initial preprocessing steps.
- **Machine Learning Framework Stability:** It is assumed that the Python-based Machine Learning libraries (e.g., scikit-learn, etc.) that power the classification module will function stably and reliably throughout the study.
- **User Competency:** For any usability evaluation, it is assumed that users participating possess a basic familiarity with desktop computing and will apply a consistent effort when using the developed GUI compared to other analysis tools.

I.6 Overview

(Content...)

Chapter II

Literature Review

II.1 Next Heading

(Content...)

II.2 Next Heading

(Content...)

II.3 Summary

(Content...)

Chapter III

Methodology

III.1 Research Design and Procedures

III.2 System Design and Modules

III.2.1 Preprocessing Module

Overview

Preprocessing

Preprocessing is the very first step of the EEG analysis pipeline. It includes adjusting the raw EEG data to make it suitable for further analysis. To analyze a dataset, we must first initialize a cfg data structure from the FieldTrip toolbox, which serves as our data container and provides numerous configurable properties. The preprocessing module of NeuroPAC streamlines this process by allowing users to set cfg structure properties directly through the GUI, automating the workflow without requiring manual code configuration. The preprocessing steps currently implemented in NeuroPAC as base preprocessing components are as follows:

- `cfg.trialfun = 'ft_trialfun_general'` : Defines trials based on specified events and time windows.
- `cfg.trialdef.eventtype = 'stimulus'`: Specifies the type of event to use for trial definition.

- `cfg.trialdef.eventvalue = {'S200', 'S201', 'S202'}` : Sets the specific event value to identify trials (conditions).
- `cfg.channel = 'all'` : Selects all available EEG channels for analysis.
- `cfg.demean = 'yes'` : Removes the mean value from each channel.
- `cfg.baselinewindow = [-0.2 0]` : Defines the time window for baseline correction.
- `cfg.dftfilter = 'yes'` : Enables the use of a notch filter to remove line noise.
- `cfg.dftfreq = [50 60]` : Specifies the frequencies to be removed by the notch filter (e.g., 50 Hz and its harmonics).

These steps can be deleted, modified, or new ones can be added from the top menu as needed. These are individual lines that get converted from simple MATLAB script lines to specific QML template items by the parameter parser, you can read further about the parameter parser and template items themselves on sections 3.3.2 and 3.3.3. Before moving on to running pre-processing, the user has to make sure that:

- Their fieldtrip path is correctly set from the top menu,
- and the file browser is pointing to the correct directory where the raw EEG data is located.

After these configurations are set, the user can run the "Preprocess and Run ICA" button, which executes a Fieldtrip function called `ft_preprocessing` that applies the defined preprocessing steps to the raw EEG data. This operation will require to run on MATLAB, therefore the MATLAB will open to ensure a smooth execution and will provide any warnings from its console that the user might need to be aware of. After the preprocessing is done, the preprocessed data will be saved in a fix named "data.mat" file in the same directory as the raw data. After preprocessing is done, the button will fulfill its second purpose, which is to run ICA on the preprocessed data. ICA is a computational dimension reduction method used to separate a multivariate signal into additive, independent components. In the context of EEG data, ICA is commonly employed to identify and remove artifacts (unwanted signals) such as eye blinks, muscle activity, and other noise sources that can contaminate the EEG recordings. In our case ICA is chosen as

`fast_ica` algorithm provided by Fieldtrip, and in this study's case only the eye artifacts were considered and removed. After running ICA, the resulting components will be saved in a file named `data_ICApplied.mat` in the same directory as the raw data for further inspection and artifact removal. The user, if ready to proceed, can then move on to the artifact removal step by clicking the "`data_ICApplied.mat`" file from the file browser, which will open a Matlab window displaying a slightly

Independent Component Analysis (ICA) & Artifact Removal

Event/Condition Decomposition

III.2.2 Analysis Module

Overview

ERP Analysis

Time-Frequency Analysis

Spectral Analysis

Channel-Wise Connectivity Analysis

Inter-Trial Coherence Analysis

III.2.3 Classification Module

Overview

EEGNet Classifier

EEG-Inception Classifier

Riemann Classifier

III.3 Other Application Components

III.3.1 Matlab Executor

III.3.2 Template Items

III.3.3 Matlab Parameter Parser

III.3.4 File Browser Window

III.3.5 Top Menu

Parameter Adding & Editing

Chapter IV

Results

IV.1 Data Analysis

(Content...)

IV.2 Summary

(Content...)

Chapter V

Summary, Conclusions, and Recommendations

V.1 Summary of the Results

(Content...)

V.2 Conclusions

(Content...)

V.3 Recommendations

(Content...)

Appendix A

Assessment Activity

(Content...)

Appendix B

Assessment Rubric

(Content...)