Logo%20Main%20200

**Boston University**

**Electrical & Computer Engineering**

**EC464 Capstone Senior Design Project**

User's Manual

Neuron Spike Identification with Machine Learning



Submitted to

Chen Yang, PhD

Yang Research Laboratory

cheyang@bu.edu

by

Team 2

Spike Sorters

Team Members

Victoria Carlsten [carlsten@bu.edu](mailto:email1@bu.edu)

Hao Chen [ha0chen@bu.edu](mailto:email1@bu.edu)

Claire Cropper [ccropper@bu.edu](mailto:email1@bu.edu)

Shi Gu [bengushi@bu.edu](mailto:email1@bu.edu)

Submitted: 4/19/2024

#### User Manual

#### Table of Contents

[Executive Summary 3](#_heading=h.30j0zll)

[Introduction 1](#_heading=h.1fob9te)

[System Overview and Installation 3](#_heading=h.3znysh7)

[Overview Block Diagram 3](#_heading=h.2et92p0)

[User Interface 3](#_heading=h.tyjcwt)

[Installation, Setup, and Support 4](#_heading=h.1t3h5sf)

[Operation of the Project 5](#_heading=h.4d34og8)

[Operating Mode 1: Normal Operation 5](#_heading=h.2s8eyo1)

[Operating Mode 2: Abnormal Operations 6](#_heading=h.17dp8vu)

[Safety Issues 6](#_heading=h.3rdcrjn)

[Technical Background 7](#_heading=h.26in1rg)

[Relevant Engineering Standards 9](#_heading=h.lnxbz9)

[Cost Breakdown 11](#_heading=h.35nkun2)

[Appendices 12](#_heading=h.1ksv4uv)

[Appendix A - Specifications 12](#_heading=h.44sinio)

[Appendix B – Team Information 12](#_heading=h.2jxsxqh)

# 

# Executive Summary

Spike Sorters

Team 02

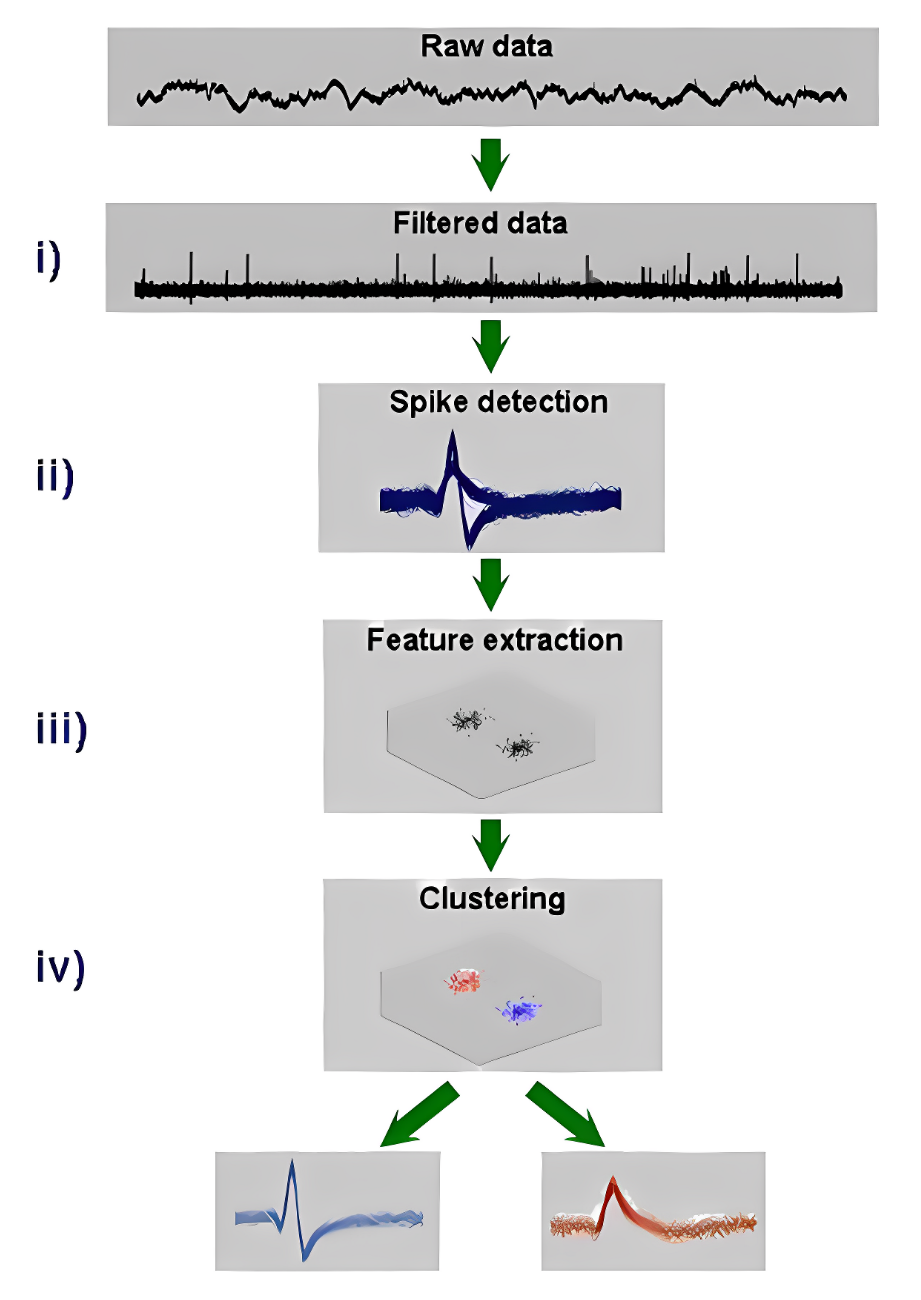
Epilepsy, a prevalent neurological disorder, impacts millions globally. Despite the array of treatments available, a significant number of patients remain non-responsive. This project involves the development of a machine learning algorithm using PCA and optimized K-means that will accurately detect and identify the brain activity, or neuron spikes, that indicate active seizure activity. The creation of an app front end allows for the seizure activity to be visualized and for any relevant parameters to be adjusted as a method for analysis following the collection.

# Introduction

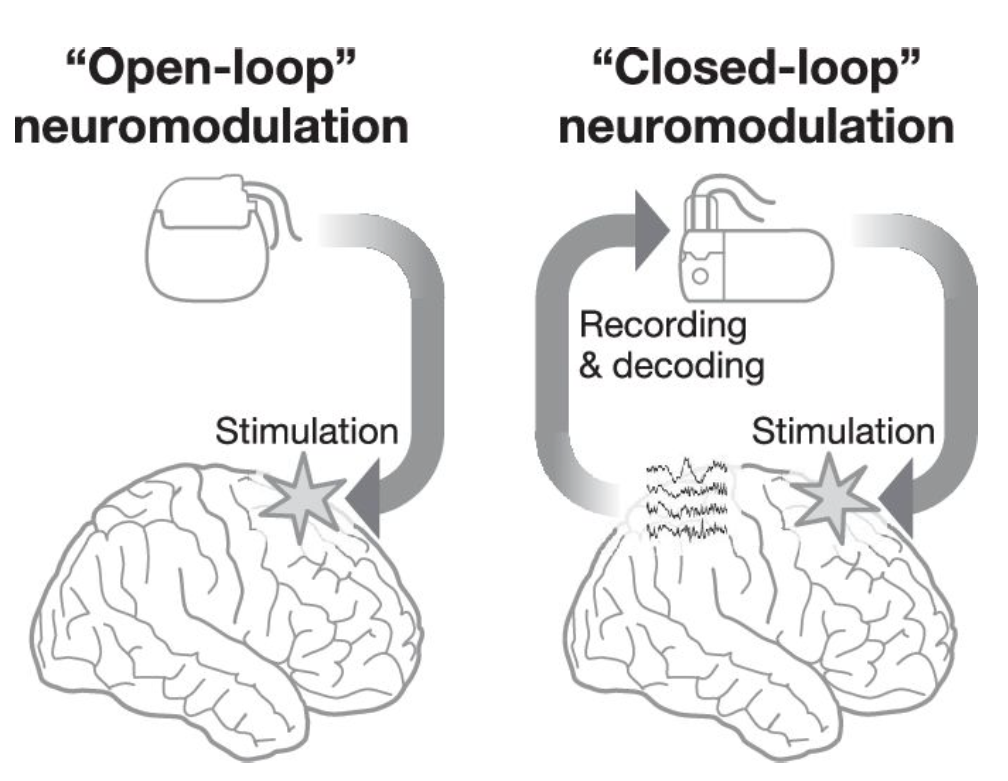
The customer's problem is the management of drug-resistant epilepsy, a condition where patients do not respond to conventional medications and are ineligible for surgical interventions. This leaves them vulnerable to frequent and potentially debilitating seizures. To comprehend this project, it's important to understand the role of neurostimulation in epilepsy treatment, specifically the limitations of current open-loop neurostimulators which deliver constant electrical stimulation without responding to the immediate physiological state of the brain.

The purpose of our team's project is to develop a more sophisticated treatment option for epilepsy through machine learning algorithms that facilitate closed-loop neuromodulation (CLN). This system delivers targeted electrical stimulation in response to specific neural activity associated with impending seizures, thus offering personalized seizure management. Our approach involves the creation of two Python algorithms. One processes electrophysiological data retrospectively to identify neural spikes and clusters, and the other does so in real-time (RTsr). These algorithms will enable precise detection and reaction to epileptic seizures as they occur.

This approach will address the user's need for an effective seizure management system by providing personalized, on-demand neurostimulation only when necessary, reducing the frequency of seizures with minimal side effects. Through automating spike detection, we offer a solution that adjusts to the individual's neural patterns, creating a tailored treatment plan that is both dynamic and responsive. The highlights of our project include the integration of cutting-edge machine learning with neurostimulation technology to produce an innovative treatment for epilepsy. This includes the development of a user-friendly app interface that allows for real-time visualization and parameter adjustment for individualized therapy. This represents a transformative step in epilepsy care, prioritizing patient-specific treatment and maximizing therapeutic efficacy.



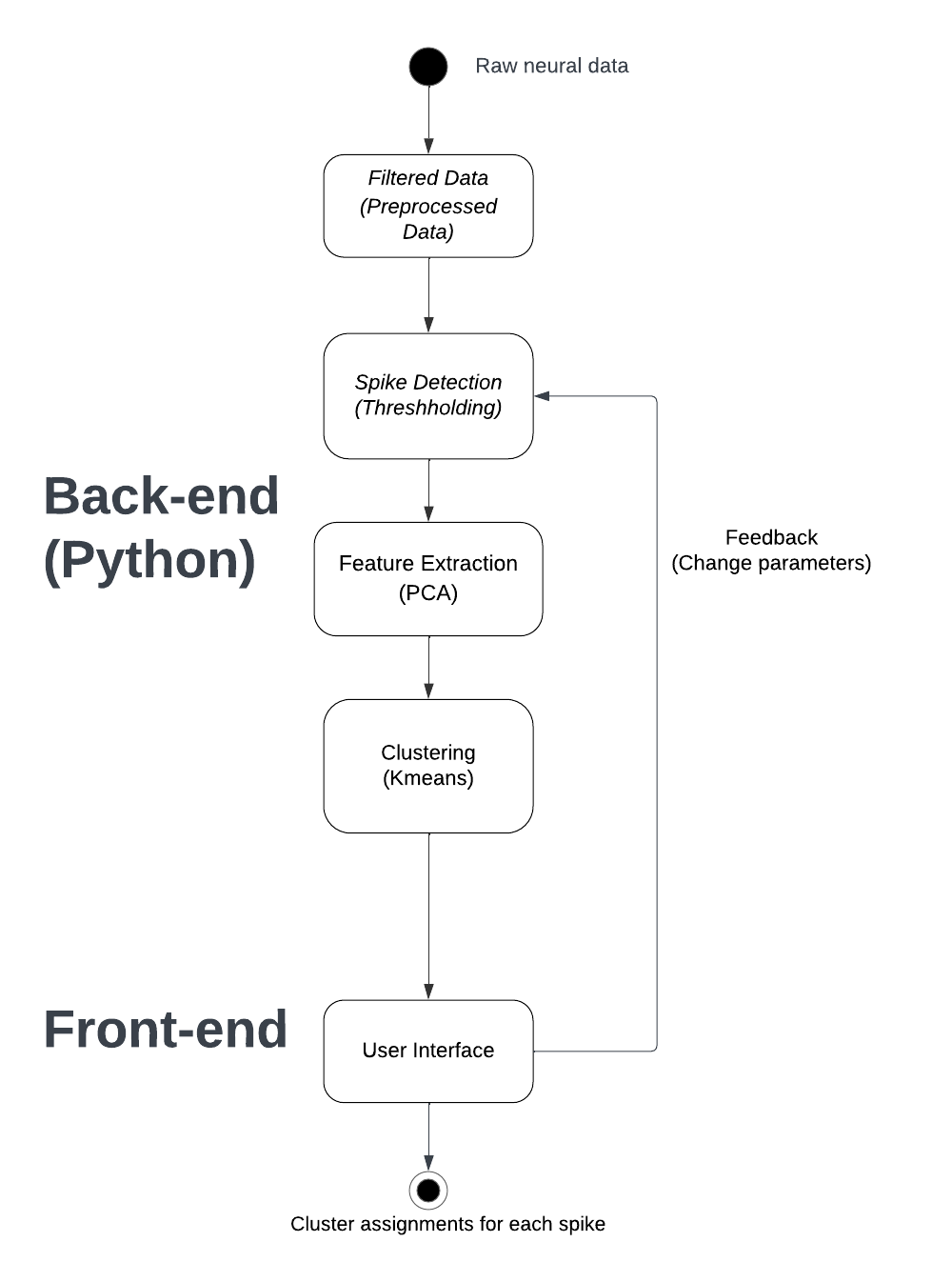
*Figure 1. Shows the basic steps for spike sorting, beginning with the recorded raw data, and ending with clustering of spikes.*



*Figure 2. Diagrams of the two various neuromodulation approaches.*

# System Overview and Installation

## Overview block diagram



*Figure 3. Block diagram of the whole project.*

Our application is structured into two primary segments: the backend and the frontend. The backend, written in Python, accepts and processes raw electrophysiology data from detection devices. It filters noise, detects spikes using thresholding, extracts pivotal features, and employs machine learning techniques, like PCA, for neuron cluster classification. This refined data is then channeled to the frontend, which visualizes labeled spike-trains and neuron clusters with confidence of labeling.

Additionally, an interactive user interface in the frontend lets users tweak processing parameters, offering tailored visualization results.

## User Interface

We will generate a web-based user interface. The user interface for the neural signal analysis project features a clean and intuitive design, focusing on ease of use for researchers. The main screen displays a dashboard with interactive graphs showing k-means clustering results, PCA scatter plots, and time-series plots of neuron signals.

## Installation, setup, and support

To begin, the users must install the python environment in their own personal computer. In order to run our code correctly, we provide a ‘requirements.txt’ file content that users can use to install all the necessary libraries for the Python environment. If users are installing these in a new environment, they can use this requirements.txt file with pip: pip install -r requirements.txt. Moreover, once users are ready to analyze their data, they can proceed by running the program and will be able to visualize all the graphs and the corresponding code.

# Operation of the Project

## Operating Mode 1: Normal Operation

**Starting the Analysis**

* User Action: Begin by loading your electrophysiological recording data into the system. This can be done through the interface by selecting either a .mat file or an ABF file.
* Response: The system performs initial checks on the file format and data integrity, then loads the selected segment of the recording based on predefined start and end times.
* Interface Options: File selection dialog, start/end time input fields.
* Normal Consequences: The selected data segment is loaded and ready for preprocessing.

**Data Preprocessing**

* User Action: Apply bandpass filtering to isolate spikes and LFP from the raw signal.
* Response: The system filters the data according to specified frequency bands for spikes and LFP.
* Interface Options: Frequency band selection, filter type, and parameters.
* Normal Consequences: Display of filtered signals in the graphical interface.

**Spike Detection and Clustering**

* User Action: Run the spike detection algorithm followed by clustering.
* Response: The system identifies spikes based on threshold criteria and groups them into clusters using K-means based on PCA features.
* Interface Options: Threshold adjustment, PCA component selection, number of clusters.
* Normal Consequences: Spikes are detected and clustered, with each cluster potentially representing a different neuron.

**Analyzing and Visualizing Results**

* User Action: Review and analyze the clustering results, inspect individual spike waveforms, and compare against optimized templates.
* Response: The system displays various plots and metrics, including raw and filtered signals, PCA space, and clusters.
* Interface Options: Selection of clusters to view, template optimization, PCA component plots.
* Normal Consequences: Insight into neuronal activity and spike patterns.

**Exiting the Analysis**

* Instructions: Save the session, export results, and close the application.
* Stopping Operations: Confirm saving and exporting before closure to prevent data loss.

## Operating Mode 2: Abnormal Operations

Anticipated Abnormal States:

* **Data Integrity Issues:** Corrupted or unsupported file formats can prevent data from being loaded correctly.

Recovery: Verify the file format and integrity before loading. The system should provide clear error messages and troubleshooting tips.

* **Clustering Anomalies:** Incorrect number of clusters or PCA components can lead to poor separation of neuronal signals.

Recovery: Implement diagnostic tools to assess cluster quality (e.g., silhouette score) and allow easy modification of clustering parameters.

## Safety Issues

Given the sensitive nature of electrophysiological data, which could potentially include identifiable human subject information, data security is a critical concern:

* **Confidentiality:** Measures should be taken to ensure that personal data is kept confidential and is only accessible to authorized personnel. Encryption of data both at rest and in transit can help protect against unauthorized access.
* **Integrity:** The accuracy and consistency of data must be maintained throughout its lifecycle. This includes safeguards against unauthorized alterations and ensuring robust data backup and recovery procedures are in place.
* **Availability:** Systems and data should be readily available when needed by authorized users. This includes implementing redundancy and failover systems to prevent data loss or downtime.

# Technical Background

**Signal Processing for Spike Detection**

The project starts with raw electrophysiological data, which consists of voltage signals recorded from neurons over time. Given the nature of these signals, they contain a mixture of desired neural spike information (action potentials) and undesired noise (e.g., electrical noise, other biological signals). To isolate neural spikes, we apply bandpass filtering, which allows frequencies within a specific range (300 Hz to 3000 Hz, typical for spike activity) to pass through while attenuating frequencies outside this range. This process is based on the Fourier transform, a mathematical technique that transforms a signal into its constituent frequencies.

**Principal Component Analysis (PCA)**

PCA serves as the backbone of our feature extraction process. By applying PCA to the spike waveforms, we aim to reduce the dimensionality of our data. Spike waveforms, inherently high-dimensional due to the sampling rate required to accurately capture their shape, present a challenge for direct analysis and clustering. PCA addresses this by transforming the original data into a new set of variables, the principal components (PCs), which are linearly uncorrelated and ordered so that the first few retain most of the variation present in the original data. This transformation is achieved mathematically through the eigendecomposition of the data covariance matrix or singular value decomposition of the data matrix itself.

The key principles that allow PCA to work effectively for spike sorting include:

* **Dimensionality Reduction:** By focusing on the principal components that account for the most variance, PCA reduces the complexity of the data, simplifying subsequent clustering steps.
* **Feature Highlighting:** The process enhances differences between spike waveforms, making it easier to distinguish spikes originating from different neurons.

**K-means Clustering**

Following PCA, K-means clustering is applied to the reduced-dimensionality data. K-means is an iterative algorithm that partitions the data into K distinct clusters based on feature similarity. Each cluster is defined by its centroid, the mean of the points assigned to the cluster. The algorithm seeks to minimize the total within-cluster variance, effectively grouping spikes that share similar characteristics (as highlighted by PCA) into the same cluster.

The operation of K-means in our project relies on the following steps:

* **Initialization:** Select K initial centroids (usually randomly from the dataset).
* **Assignment:** Assign each data point (spike) to the nearest centroid based on Euclidean distance in the feature space.
* **Update:** Recalculate centroids as the mean of all points assigned to each cluster.
* **Iterate:** Repeat the assignment and update steps until convergence is reached, meaning the assignments no longer change.

This process assumes that each cluster represents spikes from a single neuron, enabling the separation of overlapping signals from multiple neurons.

**Optimized Template Matching**

To enhance the accuracy and reliability of spike sorting, our project incorporates an optimized template matching step. After initial clustering, we generate a template for each cluster by averaging the waveforms assigned to it. These templates represent the typical spike waveform for each presumed neuron.

# Relevant Engineering Standards

**Software Development Standards**

**PEP 8 - Style Guide for Python Code:**

Ensure that all Python code conforms to PEP 8 guidelines to maintain consistency in coding style, which includes indentation, variable naming, and line length. This standardization not only improves readability but also facilitates easier maintenance and debugging.

**PEP 257 - Docstring Conventions:**

Enforce the use of docstrings for all modules, classes, functions, and methods. Docstrings should clearly describe what the function/class does, its parameters, and what it returns, following PEP 257 guidelines. This practice improves code documentation and assists in automated documentation generation.

**Software Design Descriptions**

**IEEE Std 1016-2009:**

Adhere to the IEEE Std 1016-2009 for documenting software designs. This includes detailed descriptions of software architecture, data design, interface specifications, and development environment. Your documentation should be clear, structured, and detailed, allowing new team members and stakeholders to understand the system quickly.

**Testing and Quality Assurance**

**Unit Testing and Integration Testing:**

Develop comprehensive unit tests for each component or function, using frameworks like PyTest. Follow this with integration testing to ensure that combined parts of the application function together as expected.

**Performance Standards**

**Algorithm Efficiency:**

Optimize machine learning algorithms for speed and efficiency, particularly for real-time applications like spike detection. Use profiling tools to identify bottlenecks in the code.

**Scalability:**

Design the system to handle increases in workload gracefully. This can involve optimizing algorithms, increasing hardware resources, or employing scalable cloud services.

**Reproducibility and Open Science**

**Documentation of Experimental Procedures:**

Maintain detailed records of all experiments and simulations, including parameters, version numbers, and environments used. This documentation is crucial for reproducibility.

**Open Source Libraries and Tools:**

Where possible, use open-source libraries and tools to encourage transparency and facilitate collaboration within the research community.

**User Interface Design**

**Accessibility and Usability:**

Design user interfaces that are intuitive and accessible to users with varying levels of technical expertise. Consider user feedback for UI improvements and conduct usability testing sessions.

# Cost Breakdown

Our project is purely software and has no cost.

# Appendices

## Appendix A - Specifications

Team # 2 Team Name: Spike Sorters

Project Name: Neuron Spike Identification with Machine Learning

*Table 1. Engineering Requirements*

| **Requirement** | **Value, range, tolerance, units** |
| --- | --- |
| Bandpass Filter | 300 Hz - 3000 Hz |
| Threshold | -20 microvolts - 100 microvolts |
| Spike Detection | 60 data points in 30 milliseconds |

## Appendix B – Team Information

Victoria Carlsten is a student at Boston University majoring in Computer Engineering with a concentration in Technology Innovation. 857.423.9961, [carlsten@bu.edu](mailto:carlsten@bu.edu)

Hao Chen is a student at Boston University majoring in Computer Engineering with a concentration in Machine Learning. 617.943.8368, [ha0chen@bu.edu](mailto:.ha0chen@bu.edu)

Claire Cropper is a student at Boston University majoring in Computer Engineering with a concentration in Technology Innovation. 817.917.4964, [ccropper@bu.edu](mailto:ccropper@bu.ed)

Shi Gu is a student at Boston University majoring in Electrical Engineering. 617.309.0639, [bengushi@bu.edu](mailto:bengushi@bu.edu)