Implementation Plan

Project Name:

Unsupervised Learning of EEG States for the Application of Seizure Detection in Epilepsy

Project Number:

2022659

Group members:

Mehak Kalra

Fatima Siddiqui

Khalil Scott

Mina Assaad

1.0 Executive Summary

This document overviews the three different prototypes the team is currently working on to design a software toolbox. The tool will utilize machine learning (ML) capabilities with digital signal processing (DSP) to preprocess unlabelled electroencephalogram (EEG) data. The three prototypes being implemented are stacked autoencoder, convolutional neural network (CNN), and clustering ML models. Currently, the clustering prototype is getting the most accurate result of around 50% training and 82% testing accuracy.

The design specification of the final design should aim to have an accuracy of at least 80% with a labelling speed faster than 256 Hz. Two of the key risks seen were converting the EEG data from the Neurodata Without Borders (NWB) into raw signals that could be used as inputs to each of the respective prototypes, and having machines capable of running these compute-heavy ML algorithms.

The team initially struggled to communicate with the supervisor and deliver on the initial goals set out. This was resolved by tracking the project's progress with weekly reports, as well as using project management tools, Asana and MarkWhen; to help get the team back on track for meeting project goals.

2.0 Project Status and Report and Changes

Currently, the implementation is estimated to be halfway complete. Initially, there was a lack of communication with the supervisors, but through a system of weekly emailed reports, and the project management tools MarkWhen and Asana, we have regular communication. The inclusion of weekly reports and project management tools marked a major change in coordination and organization of our project. The team has implemented three approaches: clustering, deep learning, and stacked autoencoder.

3.0 Possible Solutions and Design Alternatives

The team looked into three different ML approaches to label EEG data. The first is deep learning using a CNN. The second is using a convolutional stacked autoencoder. The third method uses clustering algorithms, such as K-means. The following subsections will present overviews of the three approaches.

3.1 Deep Learning

The CNN model was selected because it is the most popular deep learning method pursued to automate seizure detection [1]. A 2-Dimensional CNN architecture was used to preprocess 2-Dimensional Time-Frequency (2D-TF) representations of the EEG data [1]. The 2D-TF signals of the EEG were generated using the continuous wavelet transformation [1]. After preprocessing the signals using the

2D-CNN, three fully connected layers were used under the classification layer to label the signals as either seizure or normal brain activity. Figure 3 showcases the accuracy and loss for the train, validation, and test input data for a batch size of 1, a learning rate of 0.01, and 14 epochs. It can be seen that the accuracies oscillate between a range of 44% and 56% for the validation and 48% and 52% for training. Although the model is learning, its accuracy is not meeting the required 80%.

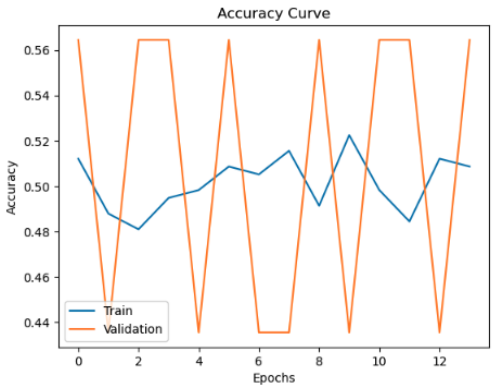

Figure 1. Accuracy curve of CNN model.

3.2 Stacked Autoencoder

Another approach the team looked at was using a convolutional stacked autoencoder, which operates on input in a similar manner to the CNN model. The main difference is unsupervised learning is used to improve feature extraction on the input by reducing it to a smaller size at the hidden layers. By using autoencoder layers to reconstruct the input signal and minimize the loss between the input and its reconstructed output, it is able to optimize the extraction of features.

Inputting these features into a classifier should then improve the accuracy of this model compared to a regular CNN. However, the training and validation accuracies yielded little improvement compared to the CNN model with oscillations between 48% and 57% training accuracy, and 44% and 56% validation accuracy. Figure 2 shows the training and validation accuracies of this model over the course of 25 training epochs.

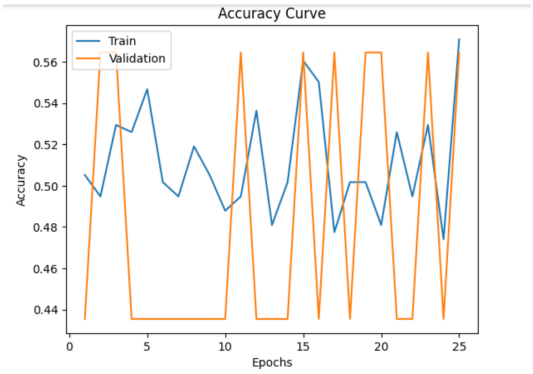

Figure 2. Accuracy curve of the convolutional stacked autoencoder model.

3.3 Clustering

The team looked into clustering as a viable option for labelling EEG data. We decided to test an implementation of the K-means clustering algorithm. From our research, we found that it is possible to achieve an accuracy of around 90% using K-means for EEG signal analysis [2]. Since K-means cannot process raw signals due to their high complexity, we used discrete wavelet transform (DWT) to decompose the signal and obtain its wavelet coefficients.

Using the wavelet coefficients, we calculated the average power, mean, standard deviation, and entropy of the signal [2]. These values would be the input features for the K-means algorithm. Using the CHB-MIT dataset, we tuned the K-means algorithm and computed the Rand Index (RI) to evaluate the accuracy [3]. RI is the percentage of true positives and true negatives resulting from the clustering algorithm. RIs of 0.50 and 0.82 were achieved on the training and testing sets, respectively.

4.0 Technical Design and Implementation

From our test implementations of the three approaches, the team has decided to select clustering as the chosen solution. The following two subsections will detail the high-level and low-level overviews and descriptions of the clustering algorithm approach.

4.1 System Level Overview

The clustering approach will require a number of steps to implement fully. First, incoming EEG signals are extracted from files using Python. Feature extraction is then performed on the input signals in order to reduce their complexity. Once a set of features has been extracted, the K-means algorithm will be used to annotate the data as normal brain activity or epileptic seizure activity. A high-level block diagram detailing the workflow of the approach can be found in Figure 3.

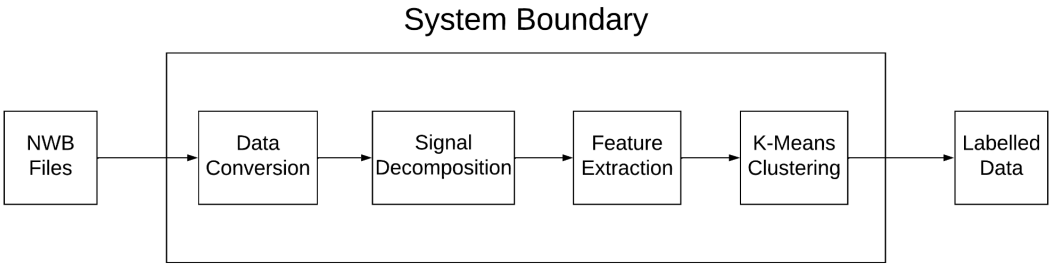

Figure 3. High-level block diagram of clustering approach.

4.2 Module Level Description

Table 1 below details the input, outputs, and overall function of each block seen in Figure 3.

Table 1. Module level description of K-means clustering approach

Module	Input	Output	Function	Evaluation
Data Conversion	NWB files	Array of EEG signals	Read NWB files and process them using the PyNWB library [4]. This will allow us to use the EEG signals in our Python scripts and split the data into a training, validation, and testing set.	N/A
Signal Decomposition	Array of EEG signals	Array of EEG wavelet coefficients per signal	Perform DWT on the EEG signals to obtain the wavelet coefficients.	N/A
Feature Extraction	Array of EEG wavelet coefficients	Array containing the average power, mean, standard deviation, and entropy of each signal	Compute the average power, mean, standard deviation, and entropy of each signal to use as the input features for the K-means algorithm.	N/A
K-means Clustering	Array of signal features	Data labels	Run K-means on the extracted features and cluster them. The cluster assignments will be used to determine the label of the signal.	Compute the distortion and RI of the algorithm

5.0 Design Specifications

This section highlights the full specifications of the project including the primary functions, objectives and constraints of the design. The functions detail what the software toolbox must do to label the EEG

data. Objectives provide the goals the design should meet when the final solution is tested. Constraints provide the limits that the software toolbox is restricted to to be a valid design.

Table 2. Detailed requirements of software toolbox

No #	Specification	Type	Metric	Goal	Reason
1	Pre-process input data for ML algorithms	Primary function	N/A	N/A	Stated by supervisor
2	Integrate with off-the-shelf supervised ML algorithms for training	Primary function	N/A	N/A	Stated by supervisor
3	Display training results and overall performance data in a visualization interface	Primary function	N/A	N/A	Stated by supervisor
4	Accuracy	Objective	Percentage of data labelled correctly	Should be greater than 80% [5]	Stated by supervisor
5	Labeling speed	Objective	Amount of data that can be labelled per unit time	Should be faster than the sampling speed of 256 Hz [6], as requested by supervisor	Stated by supervisor
6	Must have a visual interface	Constraint	Not Measurable	N/A	Stated by supervisor
7	Must accept .nwb files as input	Constraint	Not Measurable	N/A	Stated by supervisor
8	Must follow the Personal Information Protection and Electronic Documents Act	Constraint	Whether the toolbox follows the principles laid out by the code, including the collection, use, and retention of personal information from patients	N/A	An organization is legally obligated to collect and use personal information in fair and legal ways, and only with consent [7]

6.0 Project Management

The project management tool for tracking the current deadlines is MarkWhen, showcased in Figure 4. The individual tasks are delegated using Asana, as seen in Figure 5.

Figure 4. Timeline of deliverables.

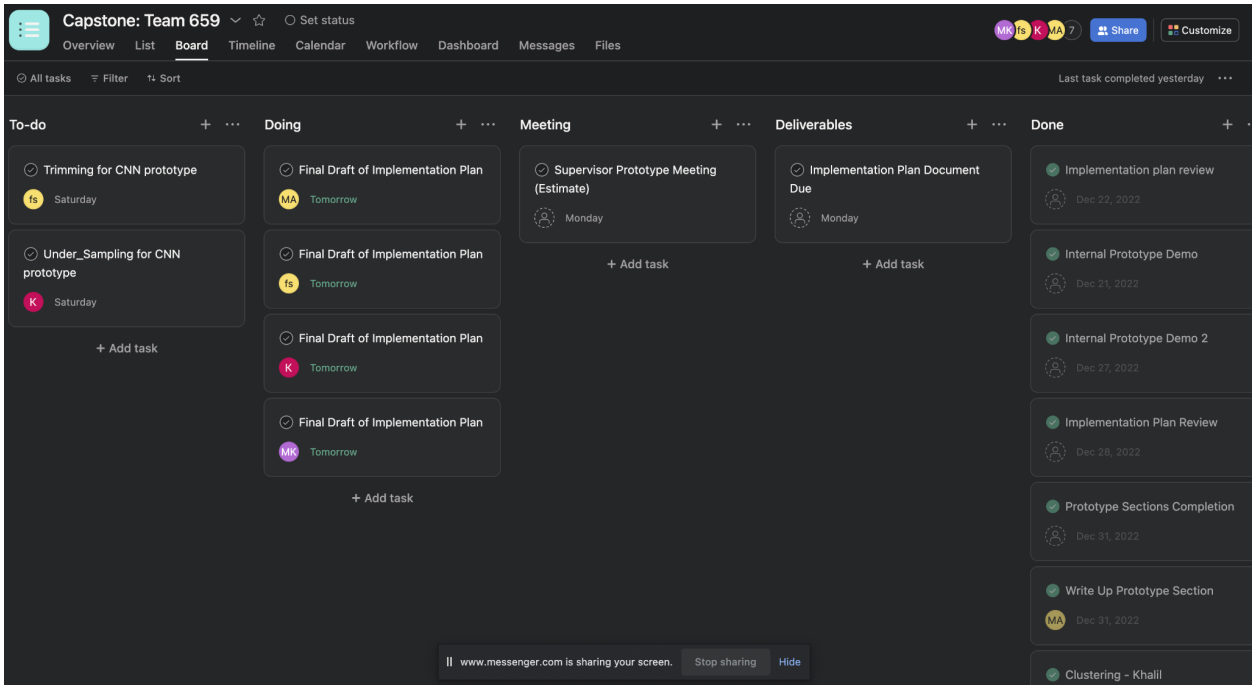

Figure 5. The board on Asana used to delegate individual tasks.

7.0 Risk Assessment

One identifiable risk is the possible unavailability of a software library tool that can read raw EEG signals from files in the NWB format. For preprocessing techniques used in our ML approaches, a raw signal is the only acceptable input form for the wavelet transformations. If there is no available Python library to read raw signals from NWB files, then the plan will be to convert the incoming NWB files to European Data File (EDF) format because there exists a popular built-in Python library tool to read signals from EDF formatted files [8].

A second identifiable risk is that training the ML models could potentially take too long. Currently, training the CNN and stacked autoencoder models takes anywhere between 1-8 hours. For models training beyond 2 hours, it can be quite difficult to keep the machine, on which the model is running, awake, and it is challenging for the individual training the model to monitor over long durations of time. If this becomes a continuous struggle, especially as we enter the second semester after the winter break, the mitigation plan will be to use fewer input data and fewer training epochs. In the long term, we might need to invest in a computer that has a high enough GPU to train the models at a faster rate [9].

8.0 References and Appendices

[1] X. Wang, T. Ristaniemi, and F. Cong, “One and two dimensional convolutional neural networks for seizure detection using EEG signals,” *2020 28th European Signal Processing Conference (EUSIPCO)*, 2021.

[2] C. K. Pradhan, S. Rahaman, M. Abdul Alim Sheikh, A. Kole, and T. Maity, “EEG signal analysis using different clustering techniques,” *SpringerLink*, 01-Jan-1970. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-13-1498-8_9. [Accessed: 05-Jan-2023].

[3] C. D. Manningm, P. Raghavan and H. Schütze, Introduction to Information Retrieval, Cambridge: Cambridge University Press, 2008.

[4] “NWB for python¶,” NWB for Python - PyNWB unknown documentation. [Online]. Available: https://pynwb.readthedocs.io/en/stable/. [Accessed: 05-Jan-2023].

[5] Jin Jing, P. D. (2020, January 1). Interrater reliability of experts in identifying interictal epileptiform discharges in eegs. JAMA Neurology. Retrieved September 23, 2022, from https://jamanetwork.com/journals/jamaneurology/fullarticle/2752670

[6] CHB-MIT Scalp Eeg Database. CHB-MIT Scalp EEG Database v1.0.0. (2010, June 9). Retrieved September 23, 2022, from https://physionet.org/content/chbmit/1.0.0/

[7] Branch, L. S. (2022, September 15). Consolidated federal laws of canada, Personal Information Protection and Electronic Documents act. Personal Information Protection and Electronic Documents Act. Retrieved September 23, 2022, from https://laws-lois.justice.gc.ca/eng/acts/p-8.6/page-7.html

[8] "EDF/BDF toolbox in python¶," *PyEDFlib*. [Online]. Available: https://pyedflib.readthedocs.io/en/latest/#download.

[9] "Choose the Colab plan that's right for you," *Google colab*. [Online]. Available: https://colab.research.google.com/signup. [Accessed: 06-Jan-2023].