Instructions

* **raw\_data** contains EDF and CSV files. The EDF files are EEG waveforms in European Data Format, while the CSV files contain timestamps corresponding to the abnormalities present in corresponding EDF files. EDF files have been categorized into Normal and Abnormal subfolders
* **utils.py** contains code for functions needed to extract windows and labels from raw data. EDF files are broken into 50% overlapping windows of size 400 samples. This size corresponds to 2 second as the sampling frequency of the EDF files were 200 Hz
* **abnormal\_img\_extractor.py** transforms windows into cwt scalograms corresponding to each of the label classes. A window is considered abnormal if there is at least 25% abnormality present in it.
* **Normal\_img\_extractor.py** Produces random normal images from all of the normal edf files. Normal class scalograms are undersampled to around 432k images by randomly taking 445 windows from each normal EDF file.
* **deeplearning\_models.ipynb** contains deep learning models, namely vgg16, Efficientnet and Googlenet. This code processes the cwt image scalograms, passes them through the models and produces deep learning results, classifying each window
* **window\_tracker.py** traces the original windows that are transformed into cwt scalograms and saves the corresponding EDF file name, channel number, window number and label into a **window\_tracker.txt file**
* **window\_tracker.txt** file is a record of every single window being used in the cwt images data set. This file is then used to reproduce the same training and test sets for the machine learning models. Since machine learning models are not utilizing a validation dataset, validation data has been added to the training set and we now have a 85% training and 15% test data split. Both machine learning dwt statistical feature data points and deep learning model cwt image data test sets have been produced from the same windows using this file.
* **machinelearning\_models.ipynb** contains machine learning models, namely a decision tree classifier, random forest classifier and support vector classifier. This code selects windows mentioned in the **window\_tracker.txt** file and extracts them from EDF files using utils.py. It is important to state the position of the ‘train’, ‘validation’ and ‘test’ keywords in the window\_tracker.txt file in the train\_pos, validation\_pos and test\_pos variables in this notebook. The windows are transformed into dwts of 1 to 4 scales using ‘db2’ as the mother wavelet. Statistical features including mean, maximum value, minimum value, standard deviation and variance are extracted from these data points. We then apply PCA of dimensionality of 2 to these features and split them into numpy arrays of 85% training and 15% test split. These data sets are then passed through the models to produce results of classification of windows

note: Use random seed of 444 for reproducing more similar results