BCI Project 1

Sotirios Moschos, DMCI: 131

April 26, 2025

VEP responses (Q.1)

The script demonstrates a vector-ordering approach used to detect outlying single-trial responses in a dataset. The dataset consists of 110 single-trial visual evoked potentials (VEPs), some of which are contaminated by artifacts (outliers).
The signal is measuring the raw voltage outputs of the MEG sensors. In this script they're simply plotted in volts (with "a.u." indicating arbitrary units), so the y-axis is sensor voltage (V).
The MEG signal is simply the time-varying magnetic flux density recorded over auditory cortex by your MEG sensors. Physically, it reflects the tiny magnetic fields generated by synchronous postsynaptic currents in populations of cortical pyramidal neurons in left and right auditory areas in response to an auditory stimulus.
A VEP is a Visual Evoked Potential—that is, the time-locked electrical response of the visual cortex to a visual stimulus, as recorded on the scalp with EEG electrodes.
In this VEP context, an artifact refers to any large-amplitude, non-physiological deflection in a single-trial recording that does not originate from the brain's visual response. Common sources include:

- Ocular activity (e.g. eye blinks)
- Muscle contractions (e.g. facial or neck muscles)
- Electrode pops or drifts (e.g. poor contact, cable movement)
- Environmental electrical noise (e.g. line interference, equipment artifacts)

Because these artifacts are typically much larger than the genuine VEP, they appear as outliers in the trial-by-trial waveforms. When present, they:

- Increase the apparent noise power (np in the script's SNR calculation), lowering the single-trial SNR (sp/np).
- Skew the ensemble average, producing spurious peaks or baseline shifts in what should be a clean VEP waveform.
- Bias distance-based rankings, since artifact-contaminated trials stand out with large vector distances to all other trials, which is precisely why the script uses vector-ordering to flag them as outliers.

The goal is to remove the detected outliers and then compare how their removal affects the signal-to-noise ratio (SNR).
The sampling frequency is 1000 Hz, meaning each sample represents 1/1000 of a second.
We'll go through the script step-by-step utilizing at first all the 110 single-trials. As a second step, we're going to utilize only the 50 first trials. We're going to compare the results of these 2 scenarios.

0.0.1 Plotting the Averaged Response

As we can see on the figures above, with only 50 trials, we end up with more residual noise in the average, whereas with the 110 trials the averaged response is smoother. In the 50 trials case, a couple of big artifacts can have a big effect on the averaged response. In contrast, with 110 trials, those same few artifacts are "diluted" by the larger pool of clean trials, so our ensemble average is more robust. VEP components (e.g. N75, P100) appear as sharp inflection points in the 110-trial trace; in the 50-trial trace they're merged with background fluctuations. After the main P100, the 50-trial average retains small ripples (line noise, ocular micro-potentials). Those ripples are visibly attenuated in the 110-trial average. Additional trials act as a low-pass ensemble filter, suppressing residual high-frequency noise.

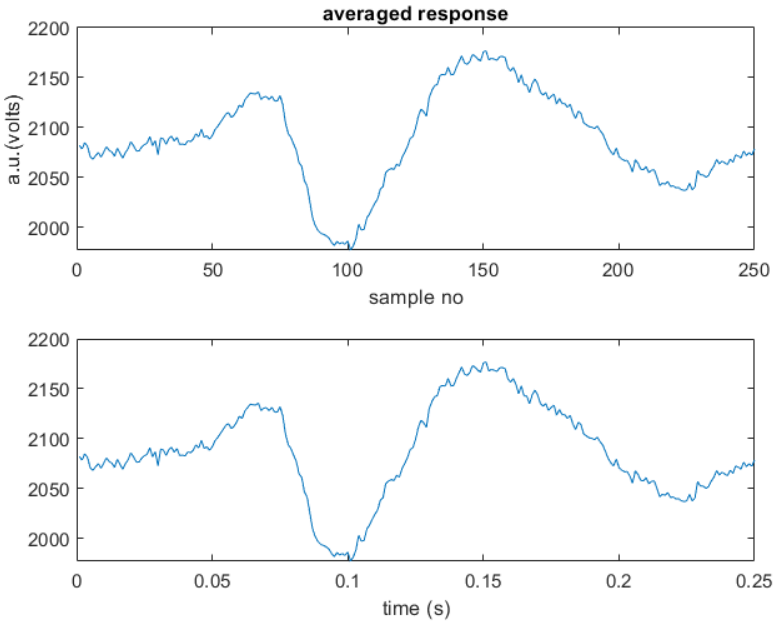

Figure 1: Averaged response across all 110 trials

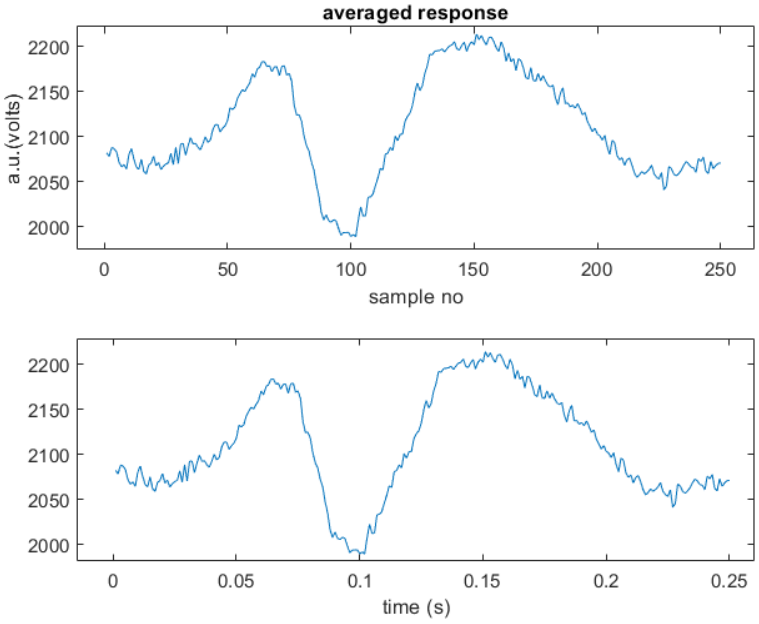

Figure 2: Averaged response across first 50 trials

0.0.2 Detrending the data (DC-Offset removal)

The detrend function removes any constant (DC) offset (or even linear trends) from the data. The transpose operation (veps') is used so that each trial (originally a row) becomes a column because detrend operates on columns by default. After detrending, the data is transposed back to its original orientation and stored in VEPS.
The DC component (Direct-Current component) of a time-series is its average (mean) value. If the average is not exactly zero, the whole waveform sits on a vertical pedestal—that constant part is the "DC-offset".
We need to remove this DC-offset, because:

- Peaks are mis-estimated; polarity comparisons become unreliable.
- Spectral leakage: A big constant term appears as a spike at 0 Hz and bleeds into the lowest frequency bins, distorting PSD or band-pass filtering.
- Algorithms that rely on Euclidean distance or variance (like the outlier ranking in the script) treat the offset as extra "signal," artificially inflating noise power.

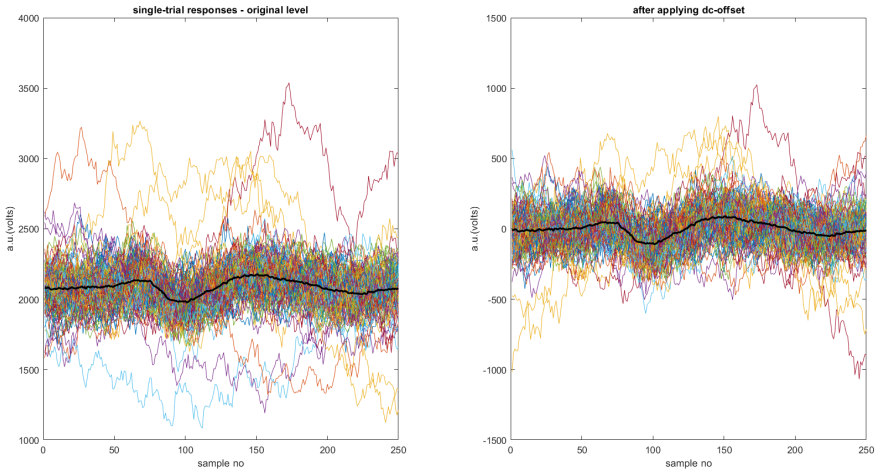

Figure 3: Comparing original and detrended waveforms across all 110 trials

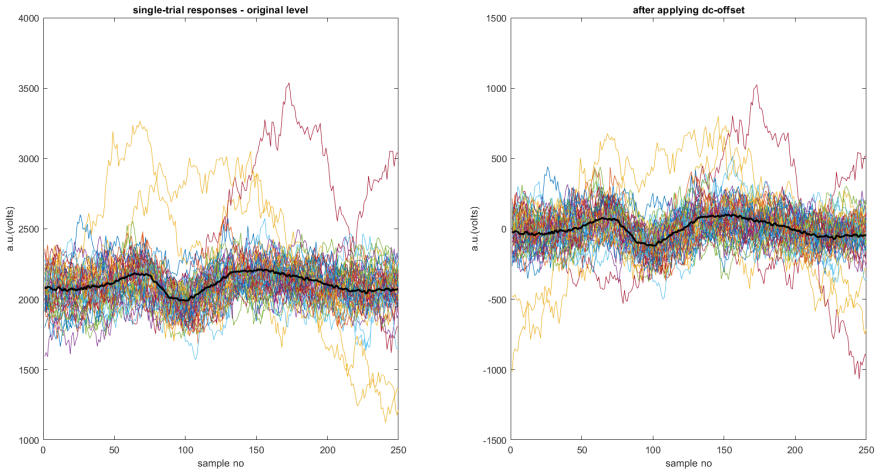

Figure 4: Comparing original and detrended waveforms across first 50 trials

0.0.3 Measuring Signal-to-Noise Ratio (SNR)

dmatrix.m produces an N×N matrix of squared Euclidean distances between every pair of trials. The unbiased variance across trials is:

$$\sigma_t^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_{i,t}-\mu_t)^2, \qquad \mu_t = \frac{1}{N}\sum_i x_{i,t}.$$

A standard identity rewrites this variance in pair-distance form:

$$\sigma_t^2 = \frac{1}{2N(N-1)}\sum_{i<j}(x_{i,t}-x_{j,t})^2.$$

Averaging that variance over all p time points gives exactly the noise power calculated at snr-sample.m.

$$\mathrm{NP} = \frac{1}{p}\,\frac{1}{N(N-1)}\sum_{i<j}\|x_i-x_j\|^2$$

Hence NP is the mean (over time) of the across-trial variance—i.e. the "noise" that remains when you line up trials but don't average them.
SP is the mean-squared amplitude of the grand average, corrected for the finite-sample noise term $(1/N)*NP$. The term $(1/N)*NP$ subtracts the finite-sample bias: when you estimate signal power from a limited number of noisy trials, a fraction 1/N of the noise leaks into the average. Removing that fraction yields an unbiased estimate of the true evoked-response power. At last:

$$\mathrm{SNR}_{\mathrm{trial}} = \frac{\text{signal power (mean waveform)}}{\text{noise power (trial-to-trial variance)}}$$

Because independent noise falls with 1/N when you average N trials, the script simply multiplies this single-trial SNR by N to predict the SNR of the ensemble-average.

SNR for an individual trial for the 110 single-trials scenario: **0.0950**
SNR for the averaged response for the 110 single-trials scenario: **10.4510**

SNR for an individual trial for the 50 first single-trials scenario: **0.1199**
SNR for the averaged response for the 50 first single-trials scenario: **5.9973**

The single-trial SNR higher in the 50-trial subset. With only 50 samples the variance of the estimate is larger; we happened to draw a "good" half. Despite the slightly worse per-trial quality, adding 60 more trials boosts the averaged SNR by 0.74. Early epochs happen to be the cleanest, but the sheer number of recordings remains the dominant factor. With 110 trials we achieve a smoother, higher-fidelity VEP than with only the first 50, even though each of those first 50 looks slightly better in isolation.

0.0.4 Computing and utilizing Pairwise Distances

MATLAB's pdist computes the pairwise Euclidean distances between all trials (each trial is treated as a vector).
Dmatrix is num-trials x num-trials (110x110 or 50x50).
sum(Dmatrix) computes an aggregate distance for each trial by summing the distances of that trial to all other trials.

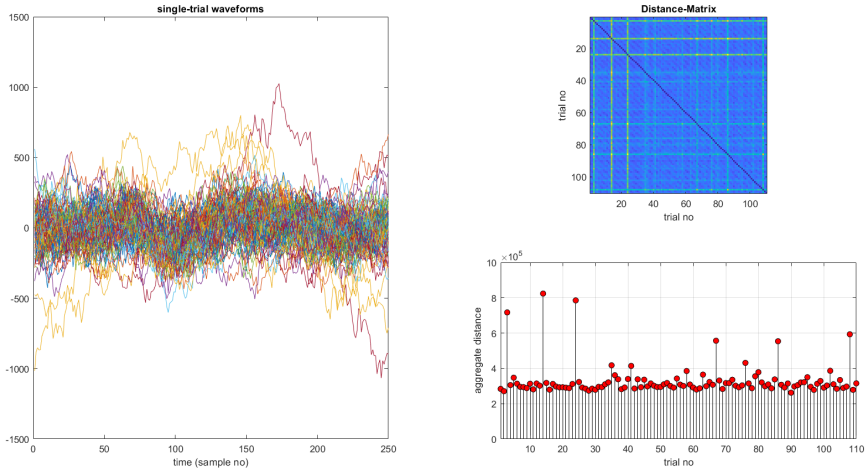

Figure 5: Visualizing single-trial data and distance matrix across all 110 trials

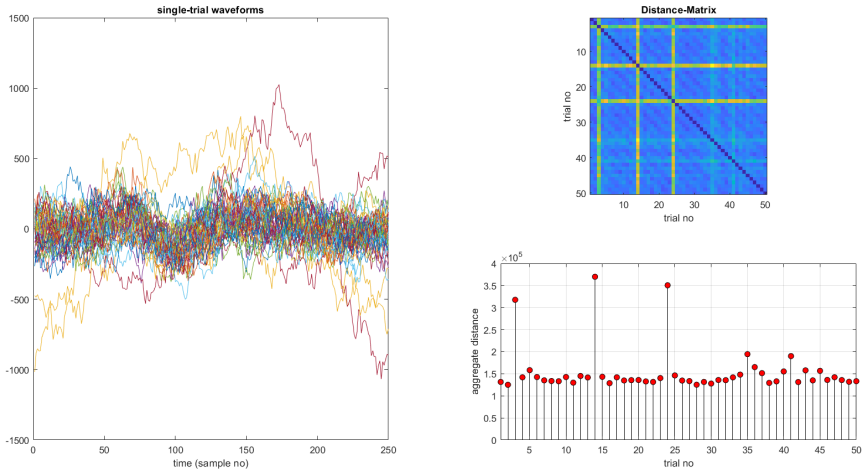

Figure 6: Visualizing single-trial data and distance matrix across first 50 trials

0.0.5 Ranking and identifying outliers

The sort function sorts the aggregate distance scores in ascending order.
The sorted-list contains the original trial indices rearranged in the order of increasing aggregate distance. Trials with low scores are more "typical" while those with high scores (at the end of the list) are candidate outliers.
In the case of the scenario with the 110 trials, we will select the last six entries from sorted-list in descending order (i.e. those with the highest aggregate distance scores).
In the case of the scenario with the first 50 trials, we will select the last three entries.

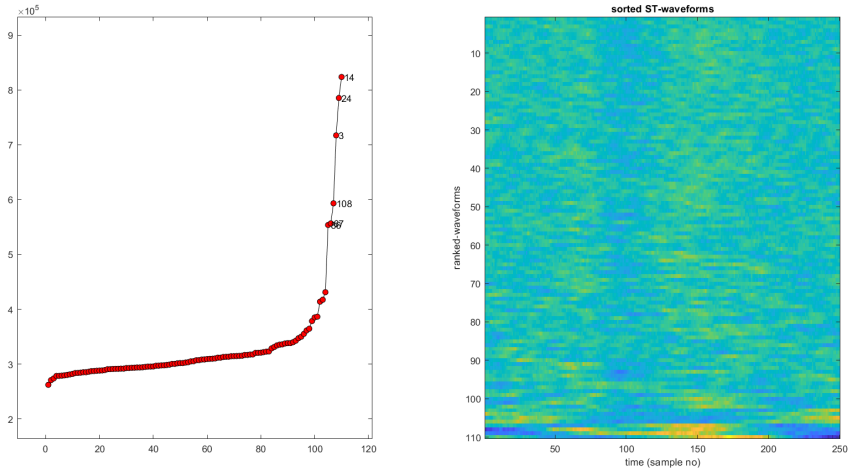

Figure 7: Ranking distances and identifying outliers across all 110 trials

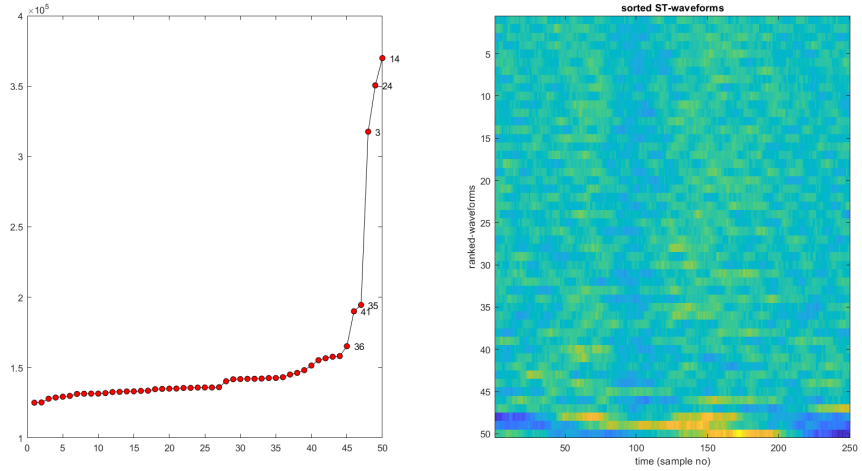

Figure 8: Ranking distances and identifying outliers across first 50 trials

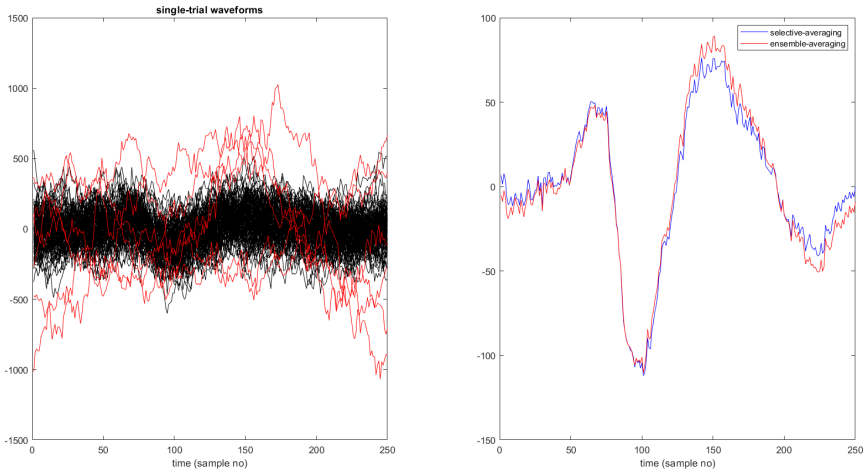

Figure 9: Visualizing single-trial data and averaged responses across all 110 trials

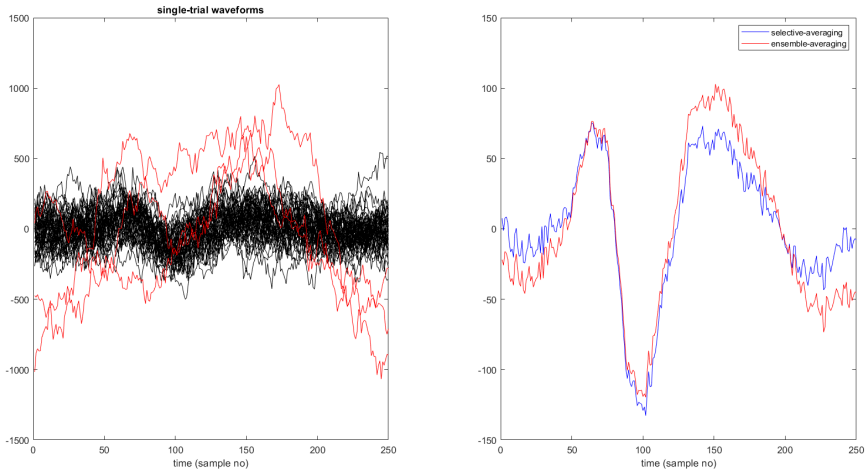

Figure 10: Visualizing single-trial data and averaged responses across first 50 trials

0.0.6 Re-calculating and Reporting SNR Improvements

The final calculation compares the SNR after outlier removal (selective-trial-SNR) with the original single-trial SNR (trial-SNR).
The result, expressed as a relative change (or percentage improvement if multiplied by 100), quantifies how much the SNR has improved due to the removal of artifact-contaminated trials.

Selective-trial-SNR for an individual trial for the 110 single-trials scenario: **0.1234**
Improvement: **0.2983**

Selective-trial-SNR for an individual trial for the 50 first single-trials scenario: **0.1489**
Improvement: **0.2413**

Epilepsy (Q.2)

This script is designed to demonstrate a feature–engineering and subsequent data–learning (classification) process for multi–channel intracranial EEG (iEEG) data recorded during epileptic events. It distinguishes between the pre–ictal and ictal states and then uses Hjorth descriptors as features and the t-test criterion for ranking in order to classify these states.
The two goals of this script are state–detection (distinguishing between pre–ictal and ictal conditions) and localizing the seizure focus.

0.0.7 Ictal data normalization

Z-normalization is essential for several reasons:

- Different gains and noise floors: In multi-channel iEEG, each electrode and amplifier chain can have its own intrinsic gain, offset, and noise characteristics. If you compared raw voltages across sensors, channels with higher gain or more baseline noise would dominate any statistics or classifiers. By subtracting each channel's own mean and dividing by its own standard deviation (both computed over the pre-ictal segment), you recast all channels onto a common, unit-variance scale. Now a "+2" in z-units means exactly "two standard deviations above baseline" no matter which sensor you look at.

By converting each channel's ictal trace into

$$z(t) = \frac{x_{\text{ictal}}(t) \; - \; \mu_{\text{pre-ictal}}}{\sigma_{\text{pre-ictal}}}$$

we ensure that

- every sensor contributes equally to downstream analyses,
- only the relative deviations that truly signify seizure onset are highlighted, and
- statistical and machine-learning steps rest on solid, standardized footing.

0.0.8 Feature extraction using Hjorth descriptors

Hjorth parameters are common in EEG analysis because they provide concise descriptors of signal properties relevant to brain dynamics. For each event in the pre–ictal and ictal state, the script computes the variance (Activity) per sensor and the first and second derivatives (for calculating Mobility and Complexity). The loop iterates over events and organizes the computed features into a 3–D array where each sensor has three attributes for each event. This structure is necessary for later feature vector construction and classification.
Activity is simply the variance, capturing the signal's overall power or amplitude spread. In

EEG, large variance can indicate either high-amplitude oscillations (e.g., seizures) or high-noise levels.
Mobility approximates the mean frequency content of the signal—higher values indicate faster average oscillations. Mobility functions as a rough proxy for the dominant frequency: if a signal oscillates more rapidly, its first difference will have larger variance relative to the raw signal. Thus, Mobility grows with higher-frequency content.
Complexity tells you whether the frequency content itself is changing rapidly. A simple rhythmic oscillation yields low Complexity, while rapidly shifting or spiky signals (such as seizure discharges) increase Complexity.

0.0.9 Feature ranking

By evaluating which features differ most significantly between the two groups, the script determines which sensors and attributes are most discriminative for the classification task.

0.0.10 Extraction of the most (and least) informative sensor and attribute

The feature ranking provides a one–dimensional index (1 to 228). Since there are 3 features per sensor, dividing by 3 and applying the ceiling function determines which sensor corresponds to the best–ranked feature. The remainder when dividing the feature index by 3 indicates which attribute (Activity, Mobility, or Complexity) is most discriminative.
The optimal sensor is sensor 10 and the most discriminative attribute is Mobility.

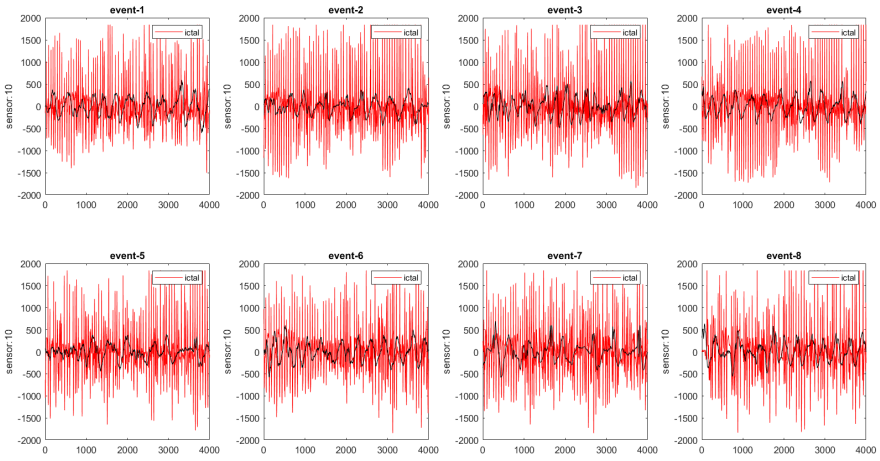

Figure 11: pre-ictal and ictal waveforms from best sensor

The least informative sensor is sensor 29 with the less discriminative attribute being Mobility.

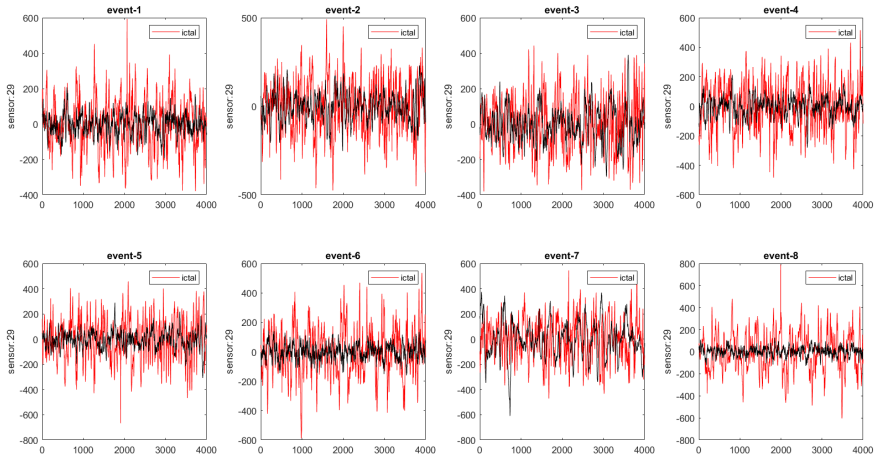

Figure 12: pre-ictal and ictal waveforms from worst sensor

0.0.11 Classification and performance evaluation

This final step is to test the performance of a Linear Discriminant Analysis (LDA) classifier. For that, we'll uses only the top 5 ranked features to classify the 16 feature vectors (8 ictal + 8 pre–ictal). For the training and testing process, we opted out for LOO cross-validation. Using LDA provides a simple benchmark to assess whether the top features yield a clear separation between classes.

LOO cross-validation accuracy (best 5 features): **1.0**

As a second proof-of-concept, we'll going to utilize the 3 features from the least informative sensor (29) and repeat the process mentioned above in order to extract a new correct classification rate.

LOO cross-validation accuracy (3 features from the worst sensor): **1.0**

Accuracy is still optimal. One reason for that is that all eight seizures in our data look very similar from event to event. Even the "worst" sensor still shows a reliably different pattern before versus during the seizure—enough that LDA can draw a hyperplane separating the two classes.