Machine Learning approach to classifying quench antenna signals

2022 Italian Summer Students program at FNAL

Final report

Supervisor:
PhD Emanuela Barzi
Cosupervisor:
PhD Reed Teyber (LBNL)

Final report of:
Alberto Plebani

Internship dates: 08/01/2022 - 09/30/2022

Abstract

In this report is present the work I have performed during the time I spent an intern at the Fermilab National Accelerator Laboratory (FNAL) as a part of the Italian Summer Students program. During this internship, I have worked at the Technical Division inside the Fermilab laboratory under the supervision of Emanuela Barzi and the co-supervision of Reed Teyber from the Lawrence Berkeley National Laboratory (LBNL). I also received help from Joseph DiMarco, Maxime Marchevsky and Steve Krave, whom I would like to personally thank. My job has been to analyse the data provided by LNBL's magnets looking for events that resembled the quench, hoping to have a better understanding of the quench and hoping to see if we could find other events generated by a current redistribution that looked similar to the quench.

Contents

Chapter 1

Theoretical introduction

1.1 Superconducting magnets

A superconducting material is a material which achieved the state of superconductivity, a state of matter that causes the electrical resistance in the material to reduce to 0. Therefore, the current inside superconductors flows without resistance, making it possible to generate higher magnetic fields, reaching also ~ 15 T.

Superconductors are of extremely importance in the physics world, especially in the High Energy Physics department, because the strong magnetic fields generated by the superconductors can provide a strong focusing and bending of the beam. Superconducting magnets are used in particle accelerators, because in a circular collider the beam has to be accelerated, and for that RF cavities are used, but it also needs to be bent and focused, and this is where the superconducting magnets come in. Magnets are used also in other HEP experiments, for example the $g-2$ experiment at Fermilab, where a muon beam has to be bent in order to run through a circular ring.

Despite all the pros of using a superconducting material in experiment, there is a big con. Superconductors behave in a superconducting way only in a certain range of temperature, current density and magnetic field, which is called the critical surface. The range of temperatures inside the critical surface is very low, as can be seen in Figure 1.1. The magnets used for the Large Hadron Collider require 1.9 K to provide a magnetic field of 6.5 T, whereas the world record magnet built at Fermilab (Figure 1.2) provides at the same temperature a magnetic field of 14.6 T

1.2 Quench

As previously stated, a superconducting material exists only inside a certain range of temperature, current density and magnetic field. During the lifetime of the magnet, it happens that one of these three parameters exits the allowed range of

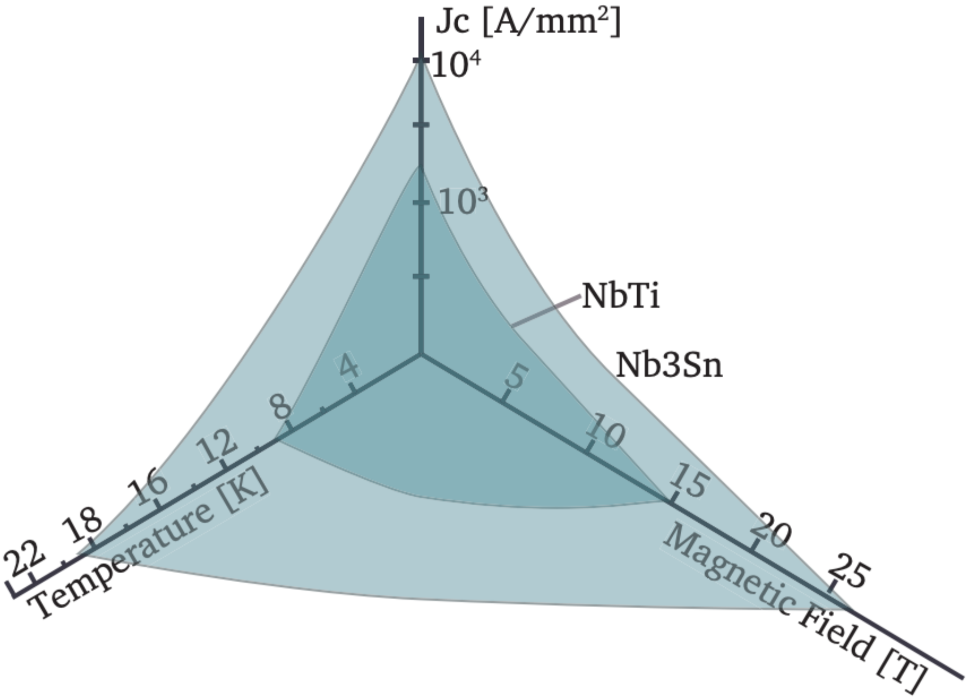


Figure 1.1: Critical surface of Nb−Ti and Nb_3Sn conductors

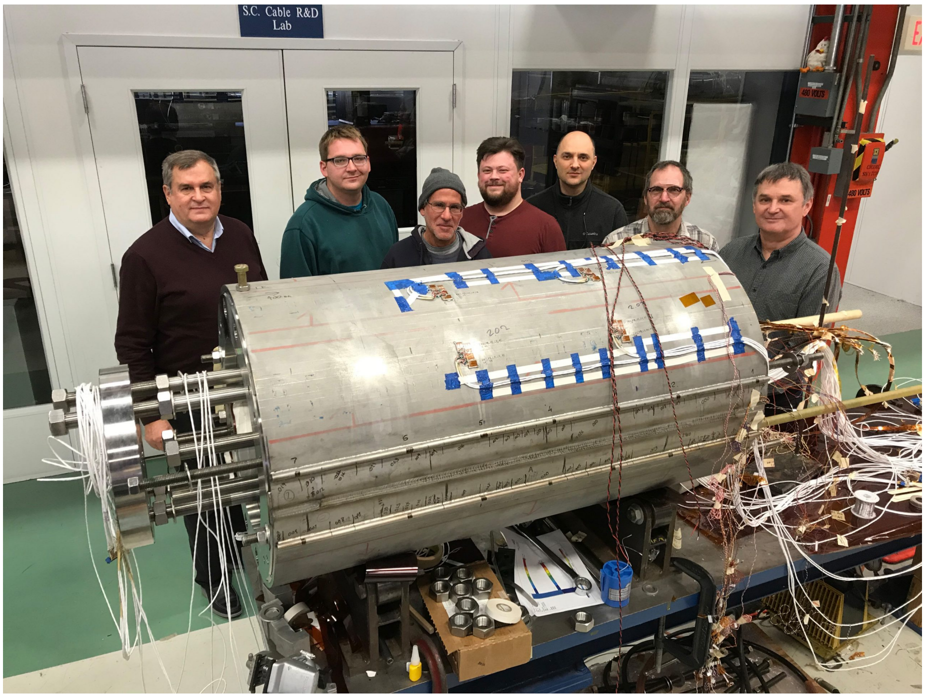


Figure 1.2: The world record Nb_3Sn magnet built at Fermilab, which provides 14.6 T at 1.9 K

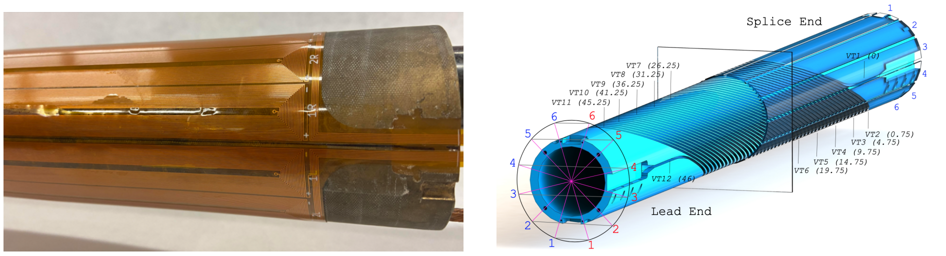


Figure 1.3: A photo of the Quench Antenna that provided me the data for the analysis on the left; a scheme of the same antenna on the right

the critical surface (usually the temperature rises above the threshold). When the material exits the critical surface, quench happens. The quench is a sudden and irreversible transition of the superconductor into the normal conducting state. During the quench, the global temperature increases in the cable, and thus the magnet requires re-cooling before normal operation can be resumed. The energy stored in the magnet must be dissipated in order to protect it, because once the quench happens, the temperature rises quickly and this would cause the magnet to break. In order to avoid that, a resistor is switched on to bring the current down. This process must be done immediately, otherwise the magnet would be permanently damaged. Therefore, it is essential to predict the quench. For this purpose, quench antennas have been developed since the 1980s.

1.2.1 Quench antennas

Quench antennas (QA) are pick-up coil array which are sensitive to changes in the magnetic flux, and can therefore provide both quench identification and localization. The sensitivity of quench antennas extends to both dynamic cable events, such as strand slip-stick motion, and stationary current redistribution, which may or may not be preceded by a mechanical event.

In my work at Fermilab I analysed the data provided by the quench antennas at the Lawrence Berkeley National Laboratory (LNBL), which can be seen in Figure 1.3. It is a printed-circuit-board (PCB) antenna integrated into the coil structure of the Nb_3Sn superconducting magnet. Since the antenna is close to the magnet, the sensitivity to current redistribution, strand motion and cable motion is increased.

During this analysis, I will be presenting data from 9 different ramps all the way to quench.

1.3 Machine Learning and clustering

Machine Learning (ML) is a branch of Artificial Intelligence (AI) devoted at building algorithms that learn, which means that this methods leverage data to improve performance on some set of tasks. Machine Learning techniques are divided into two categories: supervised ML (SML) and unsupervided ML (UML). SML algorithms require a training dataset, in which each entry of the dataset has an assigned label, which specify to which category each entry corresponds. UML is more complicated than SML, because with UML you don't know each entry in which category belongs, and the goal of the algorithm is to find the different categories. For my work I had to perform Unsupervised ML, because I do not know both the number of categories and consequently the category in which each event belong. The goal of my analysis was to classify the different events into clusters which shared the same features. In order to do that, I decided to use the K-Means clustering algorithm.

1.3.1 K-Means algorithm

K-Means is a partitioning algorithm, that is an algorithm which aims at partitioning a database D of n objects into a set of K clusters, such that the sum of squared distances is minimized. In K-Means, each cluster is represented by the center of the cluster. This means that k-means can be applied if we have all objects described by numerical features. The mean value of the points within the cluster id defined as the centroid. This algorithm works as follows:

- Choose a number of centroids k
- Arbitrarily choose k objects from the dataset D as the initial cluster centers
- Assign each object to the cluster to which the object is the most similar based on the mean value of the objects in the clusters
- Calculate the mean value of the objects for each cluster
- Repeat until there's no change in the cluster centers

This algorithm is really efficient, as it is $\mathcal{O}(tkn)$, where t is the number of iterations and n is the number of objects, because it is linear with k, whereas other algorithms like PaM are $\mathcal{O}(k(n-k)^2)$. The clusters have a spherical shape, because we similarity between points is evaluated using the euclidean distance. The weakness of this method is that we need to specify k, so we include a bias, because we force the algorithm to select a specific number of clusters. We could think of executing k-means a number of times, always increasing the value of k, and selecting the value for which the cost function is the lowest. The problem with this method is that the minimum is found when k=n, i.e., we have each event clustered alone. Therefore, this approach cannot be used. Luckily, we can perform different tests to find the optimal value of k, which are:

- k-elbow method

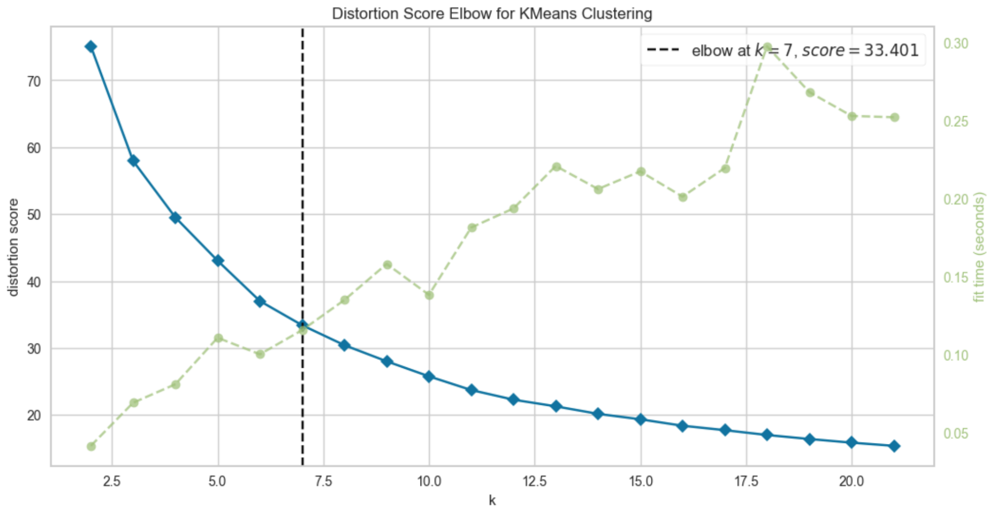


Figure 1.4: Example of an elbow plot, which shows the optimal value $k = 7$

- Silhouette analysis
- WSS cross validation
- Consistency test

k-elbow

The k-elbow method states that increasing the number of clusters can help to reduce the sum of within-cluster variance of each cluster. We consider as the optimal value of k the value for which the slope of the cost function changes the most, as we can see from Figure 1.4.

Silhouette

The silhouette analysis is an intrinsic method, which evaluates how well the clustered are separated and how compact they are. For each object i in the dataset D we compute $a(i)$ as in Equation (1.3.1)

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} \text{dist}(i, j) \tag{1.3.1}$$

The silhouette coefficient of every object tests the compactness of the cluster in which each object is located. We want $a(i)$ to be as low as possible, because it means that the cluster is compact.

We then evaluate $b(i)$ as in Equation (1.3.2)

$$b(i) = \min_{j \neq i} \frac{1}{|C_j|} \sum_{j \in C_j} \text{dist}(i, j) \tag{1.3.2}$$

The silhouette coefficient of the object i is $s(i)$ (Equation (1.3.3))

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{1.3.3}$$

The single silhouette coefficient, which is the coefficient for each object, ranges from -1 to $+1$. When the coefficient is negative, it means that the object is closer to objects in other clusters than to objects in its own cluster, whereas when the coefficient is close to 1 it means that the objects are well clustered. For all the objects we evaluate $s(i)$ and then we take the average on all the clusters. We call this average the silhouette score, and we decide the optimal value of k as the value for which the score is higher and the number of negative silhouette coefficients is lower.

WSS cross validation

The Within-cluster Sum of Squares Cross Validation is a method that consists on dividing the dataset into $m < n$ parts, and then using $m - 1$ parts to obtain a clustering model. The remaining parts are used to test the quality of the clustering. For each point in the test set we find the closest centroid and then we use the sum of squared distance between all points in the test set and the closest centroid to measure how well the model fits the test set. We repeat this analysis m times for each k, and we select the optimal k as the value for which the slope of the cost function changes the most, same as we did for the elbow method.

Consistency

Finally, the consistency test is a way to measure if the clusters are reproducible. For each k we try to do the same clustering N different times, with N usually in the order of 100. We evaluate the standard deviation of the number of events in the biggest cluster, and the optimal value of k is the one for which the standard deviation is the lowest, because it means that at each iteration we generate similar clusters.

Chapter 2

Event and feature selection

The purpose of my analysis was to find events along the ramp which shared some features with the quench. I couldn't work on a time series data, because any Machine Learning algorithm requires single separate events. In this section I will be explaining how we selected those events and how from those events we extracted some peculiar features, which will then be fed to the K-Means clustering algorithm which will help on identifying the representative events.

2.1 Event Selection

The data was recorded with a sampling frequency of 25 kHz, and we defined an event as follows:

A peak in the voltage is considered an event only if a spike above a certain threshold is matched to a spike under a negative threshold within 40 μs.

The picture below (Figure 2.1) shows what I mean. With a chosen threshold, we can see that using this definition for the events we eliminate any possible noise fluctuation.

2.1.1 Filter

I started my analysis looking at one single ramp, the one I will call *Antenna 9*. The voltage recorded along the ramp is presented in Figure 2.2. A closer look on the voltage allows to observe the frequency of the noise oscillation. This background noise has a frequency of roughly 4.54 kHz, calculated averaging over the 9 periods shown in the Figure 2.3. In order to reduce this background, we decided to implement a 6th order bandstop Butterworth filter. For this filter, we used the `butter` function provided by the `scipy` package of Python [1]. The cutoff frequencies were chosen to be 4 kHz for the lowest and 6 kHz for the highest, so that the frequencies in the $4 - 6$ kHz range have been reduced. The effect the filter has on the signal can be seen in Figure 2.4. We can note that the amplitude

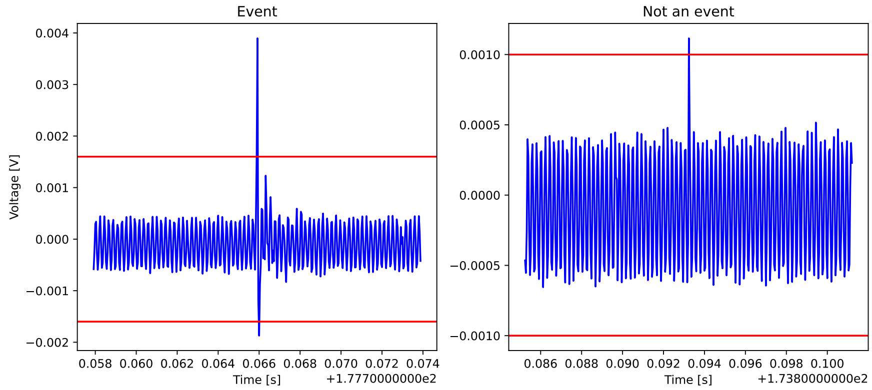


Figure 2.1: The comparison between an event (on the left) and a fluctuation which is not considered to be an event (right)

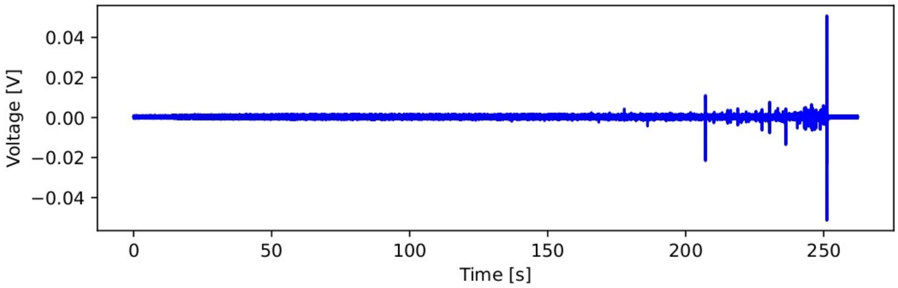


Figure 2.2: Voltage along the ramp by Antenna 9

of the oscillations has been reduced, because most of the waves oscillating at the background noise frequency has been eliminated. In the following, both the raw and the filtered signal has been analysed.

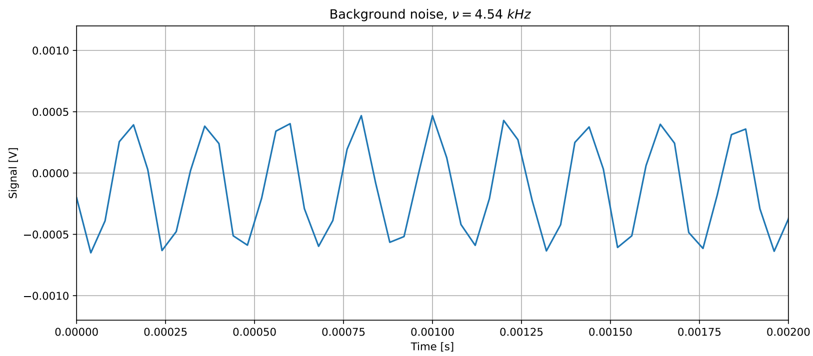


Figure 2.3: A closer look on the voltage along the ramp, which allows to observe the background noise oscillation

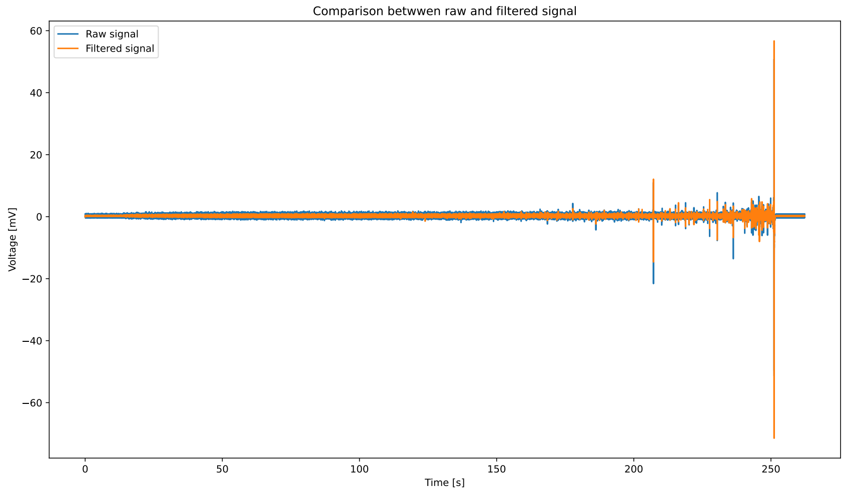


Figure 2.4: Comparison between the unfiltered (blue) and the filtered signal (orange) after applying the 6th order Butterworth filter

2.1.2 Threshold analysis

As I stated before, we considered a spike in the voltage to be an event only if a peak above a positive threshold was matched to a peak under the negative threshold within 40 µs. Since we did not know in advance what was the optimal threshold, I decided to test different values of threshold, between 1 mV and 4 mV, using steps of 0.2 mV. Once again, I will be presenting the results only for Antenna 9, and these results are in Table 2.1. Figure 2.5 shows the events selected by each threshold, in the 190 − 260 s region of the ramp.

As expected, the yields are higher for the raw signal and for lower thresholds. Since a Machine Learning technique requires a large number of events, we decided

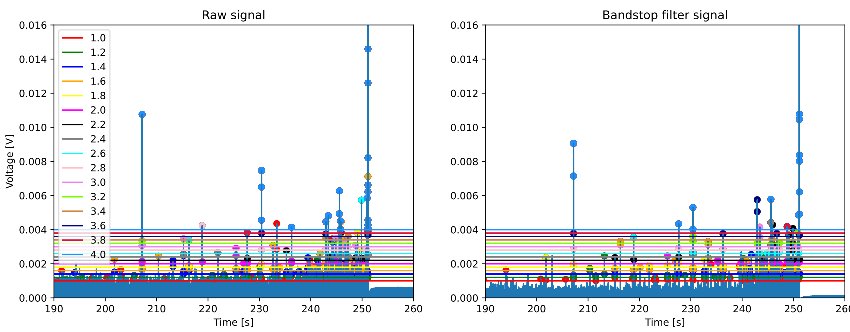


Figure 2.5: A zoom in the voltage distribution of Antenna 9, showing the different events, for both raw (left) and filtered (right) signal. Each colour represents a threshold, and the coloured dots represent the events

Threshold (mV)	Raw data	Bandstop filter data
1	147	82
1.2	90	66
1.4	67	51
1.6	51	45
1.8	36	35
2	31	32
2.2	27	25
2.4	24	18
2.6	22	17
2.8	20	14
3	17	14
3.2	13	13
3.4	13	11
3.6	12	10
3.8	11	7
4	10	6

Table 2.1: Table presenting the number of events found for each threshold, for both the filtered and the raw signal

to use the lowest threshold, in order to have the highest number of events. This same selection was used for all the ramps. We also decided to use the filtered signal. In the end, from the nine ramps we obtained 2491 events.

Once I have selected all the events, I generated a window around each one of them. All the windows have the same length, which is 5.2 ms, and lasts from 2.4 ms before the peak to 2.8 ms after the peak. An example of a windowed event and a quench event can be seen in Figure 2.6. The 9 quench events are selected in a different way, and this can be noted by looking at the plot in Figure 2.6. Since after the quench the current must be dissipated, we can't consider too much time after the quench into the windowed event, because otherwise we would be analysing non-physical data. Therefore, the window around each quench is generated in a different way, keeping the windows of the same length (5.2 ms) but changing the time at which each window starts. We can see all the nine different quenches in Figure 2.7.

2.2 Features

The K-Means clustering algorithm requires a matrix whose entries are values which will then be used to distinguish between the different clusters. These values are what I will call *features*.

The features must be representative of each event, showing some characteristics that will help on generating clusters. In the following I will give a brief description

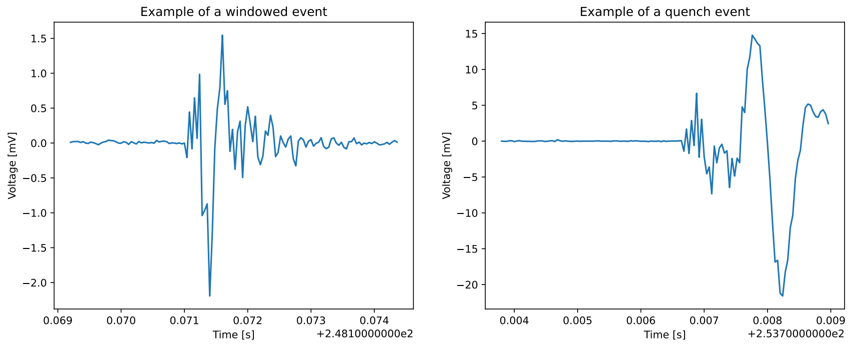


Figure 2.6: An example of a windowed event (left) and an example of a quench event (right)

of the features selected.

2.2.1 Voltage features

The first features we selected were the features regarding the voltage distribution. Figure 2.6 shows the voltage distribution for one event, and from that distribution many features can be extracted.

i) Maximum voltage: it is the value of the maximum voltage in the windowed event. It can be useful because the quench events have a higher peak in the voltage distribution, therefore this feature could help identifying events that look similar to the quench

ii) Maximum voltage: it is the value of the minimum voltage in the windowed event. It can be used instead of the maximum value because sometimes the negative spike is higher in absolute value than the positive spike

iii) Absolute maximum: it is the absolute maximum value of the voltage in the windowed event. It's better than both maximum and minimum

iv) Maximum integrated voltage: it is the highest value of the integral of the voltage. The integral of the voltage is a useful variable which can give informations on the magnetic flux received by the quench antennas. The integral is evaluated using the `cumulative_trapezoid` method provided by the `scipy` package of Python [2]

v) Minimum integrated voltage: it is the lowest value of the integral of the voltage

vi) Absolute maximum integrated voltage: it is the absolute highest value of the integral of the voltage

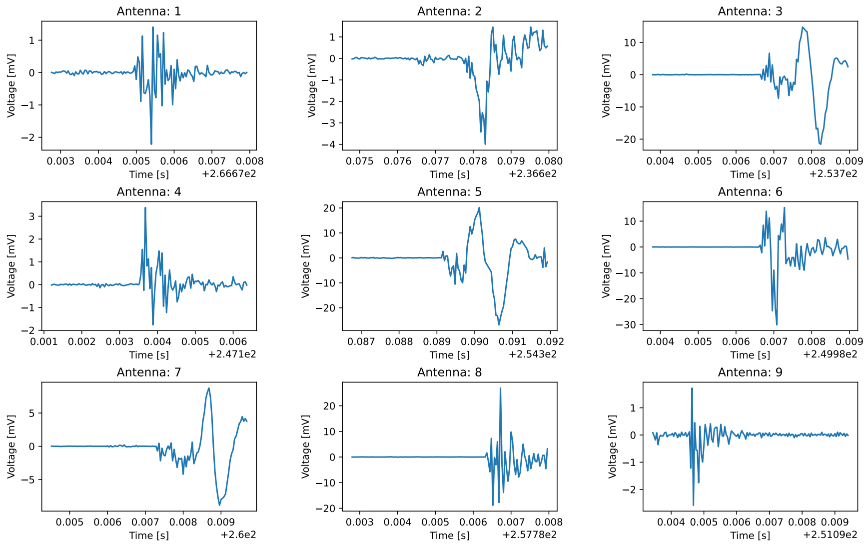


Figure 2.7: The 9 quench events, one for each ramp

vii) Norm of the voltage: it is the euclidean norm of the voltage array, evaluated as $\|\mathbf{v}\| = \sqrt{\sum_i v_i^2}$, using the `linalg.norm` function implemented in the `numpy` package [3]

viii) Norm of the integrated voltage

ix) Definite integral of the voltage: it provides information on the total amount of magnetic flux received by the quench antennas

2.2.2 Signal shape features

We then focused on features regarding the shape of the signal. These four features can help us understanding how fast a signal is.

i) Time 80: it is the difference between the time the voltage reaches 80% of the peak and the time of the absolute maximum. The 80% mark is selected after the peak, so that we look only at how fast the voltage restore itself to its normal value after the spike

ii) Time 50: it is the difference between the time the voltage reaches 50% of the peak and the time of the absolute maximum

iii) Time 30: it is the difference between the time the voltage reaches 30% of the peak and the time of the absolute maximum

iv) Time 20: it is the difference between the time the voltage reaches 20% of the peak and the time of the absolute maximum

2.2.3 Frequency features

Finally, we focused on features regarding the frequency distribution of each event. The frequency distribution can be useful because some of the quench events, as can be seen in Figure 2.7, show a peculiar 1 kHz oscillation. We want to find features that can provide information on whether a 1 kHz oscillation occurs in the signal distribution.

In order to extract features on the frequency distribution, we decided to use the Continuous Wavelet Transform (CWT). We did not use the Fourier Transform (FFT) because the FFT is localized only in frequency and not in time. Therefore, we lose all the information on when a specific frequency occurs in the distribution, and since the quenches show a 1 kHz oscillation at the time in which the voltage is maximum, it is important to have as much informations as possible on both the time and the frequency. This is where the CWT comes in handy, because the wavelet measurement tells something about the temporal extent of the signal, as well as something about the frequency spectrum of the signal. Since Heisenberg's uncertainty principle states that one cannot measure both the frequency and the time at the same time, the CWT provides a lesser precise measurement of both time and frequency, whereas the FFT provides a theoretically infinitely precise measurement of the frequency, losing all the informations on the time. We have selected five features:

i) Lead frequency from the CWT analysis: it is the frequency with the highest number of occurrences in the frequency distribution

ii) Number of peaks in the frequency distribution at the time when the voltage was the highest: it is a feature that uses the property of the CWT to extract information on both time and frequency. It evaluates the frequency distribution at the time in which the peak in the voltage occurs, and it looks for the peaks in the distribution using the function `find_peaks` of the `scipy.signal` package [4]

iii) Highest peak in the frequency distribution at the time when the voltage is higher: it's not the highest frequency when the voltage was higher, because otherwise it would mostly coincide with the Lead frequency, but it is the highest peak, therefore it requires the frequency distribution to have at least one peak. If no peaks are found, this feature returns 0

iv) Standard deviation of the frequency distribution at the time when the voltage is the highest

v) Number of times the frequency of 1 kHz is found in the frequency distribution at the time when the voltage is the highest: it gives an idea on how much the 1 kHz frequency is present in the distribution, which we have seen to be a peculiar characteristic of the quench. We consider all the frequencies in the $0.75 - 1.25$ kHz, because the frequency of the quenches isn't exactly 1 kHz

The features $ii) - v)$ are all features which consider the frequency distribution at the time when the voltage reaches its peak. It is important to consider that time between the signal is very short, lasting usually no more than a couple of milliseconds, and it's in those milliseconds that we can extract informations in order to match some events with the quenches.

2.2.4 Plots displaying some features

Non-quench event

The first example is shown in Figure 2.8. The top left plot shows the voltage distribution, in which the absolute maximum and its time have been highlighted in yellow. The red dotted line represent the voltage at 20% of its peak and its corresponding time. The difference between the red and the yellow vertical lines is the feature called `time_20`. The other time features (Section 2.2.2) are not represented. The top right plot displays the integral of the voltage, in which the dotted yellow line represents the absolute maximum value of the integrated voltage. The bottom left plot shows the CWT analysis. The white plot above the background is the voltage, which allows us to see the frequency distribution at the time when the voltage was the higher. The lead frequency is the frequency which appears with the most intense shade of red. In this example is around 7.6 kHz. Finally, the bottom right plot shows the frequency distribution at the time in which the voltage was the highest. It can be seen that the

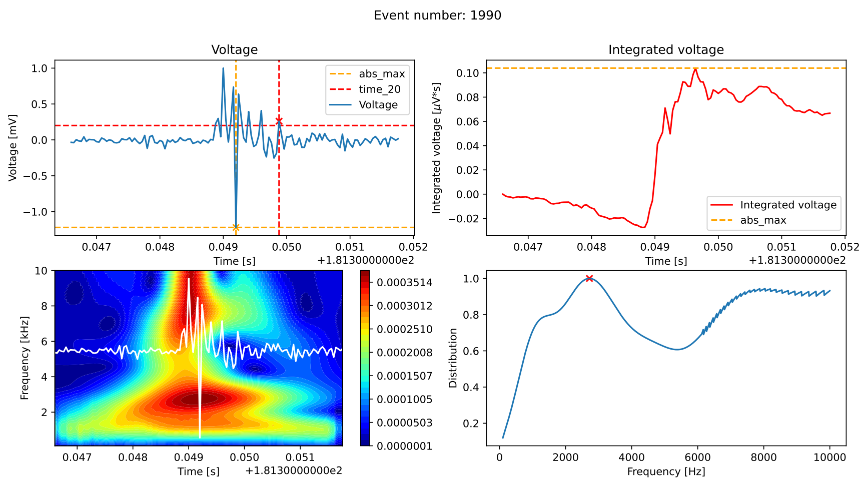


Figure 2.8: Example of some selected features for an event

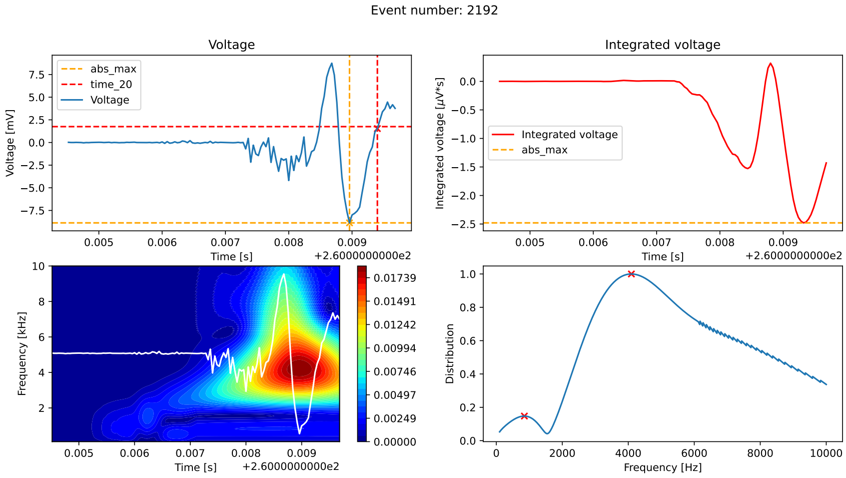


Figure 2.9: Example of some selected features for a quench event

Quench event

An example of a quench event can be found in Figure 2.9. We can recognize the same features discussed in the earlier section for the non-quench event. There are a couple of things worth discussing. The first one is the absolute maximum value of voltage (and consequently of integrated voltage). We can see that for the quench, this values are much larger than they are for the non-quench event. The second one can be seen from the frequency distribution. The lead frequency is around 4.5 kHz, which is the frequency of the noise. However, we can observe that the frequency distribution at the time when the voltage was the highest shows a peak at a frequency close to 1 kHz. This is what we expect to see with those quenches and other events that show current redistribution.

Chapter 3

Machine Learning analysis

In this section I will present the clustering analysis we performed on the feature matrix generated as per Section 2.2. We have selected 18 features, but it is clear that we can't use all those features for the clustering. The first reason is that many features are strongly correlated to each other, for example maximum, minimum and absolute maximum. For the clustering, we don't want to use too many features, because otherwise the algorithm wouldn't have any freedom to learn, and we want to use only features which don't have a strong correlation to each others. We tested two combinations of features, one with many features and one with few features. I will present both analysis above

3.1 7-features analysis

We decided first to perform the analysis using eight features. The eight features selected are `abs_max` (the absolute maximum value of voltage), `def_int` (the definite integral of the signal), `time_20` (the time between the maximum and 20% of the peak), `lead_freq` (the lead frequency from the CWT analysis), `num_peaks` (the number of peaks in the frequency distribution at the time when the voltage was maximum), `std_freq` (the standard deviation of the frequency distribution at the time when the voltage was maximum) and `occ_1000` (the number of times the frequency of 1 kHz is found in the frequency distribution at the time when the voltage is the highest).

3.1.1 Data visualization

In Figure 3.1 we can see the correlation between all the seven features. We have removed the features that showed a strong linear correlation: for example, we noted that `high_peak` and `lead_freq` had a strong linear correlation. Therefore, we decided to use only the latter of those. Figure 3.2 shows the distribution of the features along the 9 different ramps, with the dotted red lines representing

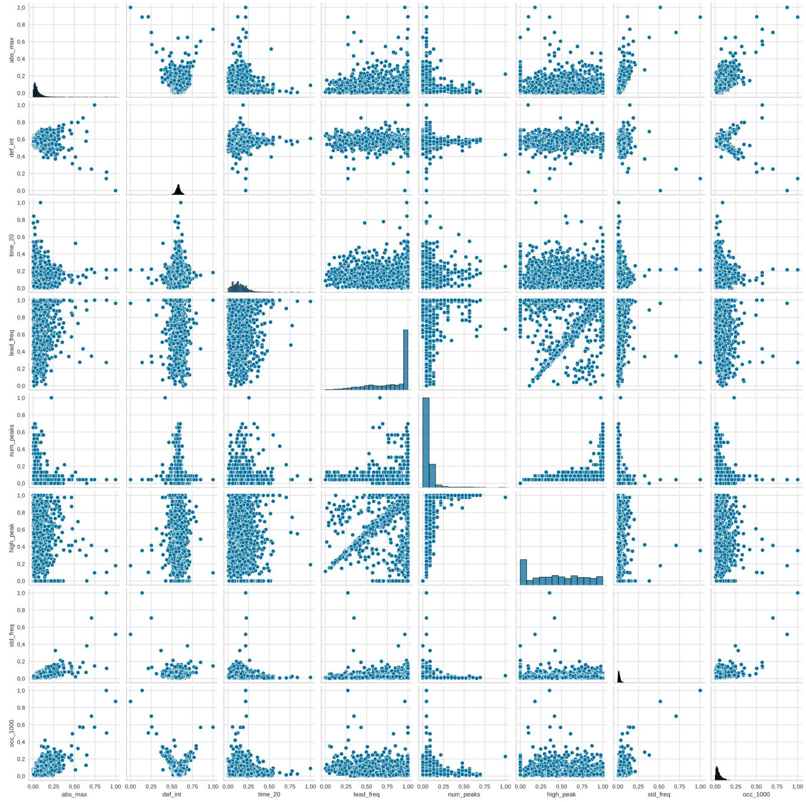


Figure 3.1: Correlation plot for all the features. The main diagonal represents the normalized distribution for each feature

the quench events. We can see that the lead frequency distribution shows many events with a frequency of 10 kHz. This happens because we are performing the CWT analysis using frequencies only in the $0.1 - 10$ kHz. This suggests that `lead_freq` will represent a "biased" feature, because we are forcing the frequency to be lower than a certain threshold. Nevertheless, this feature is useful because it tells us he events in which the lead frequency is high. Furthermore, we note that some of the quenches have an absolute maximum higher than the other events, as well as a high value of `occ_1000`, as expected.

PCA

In order to better visualize the data, a Principal Component Analysis (PCA) was implemented. The PCA is a tool that allows to visualize the data coming from many variables (i.e, the features) using less variables, in a way such that all the variance of the original variables is still explained. We decided to use 3 variables to describe all the variance of our 7 features, having noticed that 3 variables explained more than 90% of the variance of all the data. Figure 3.3 shows the correlation between the new variables, whose decomposition can be

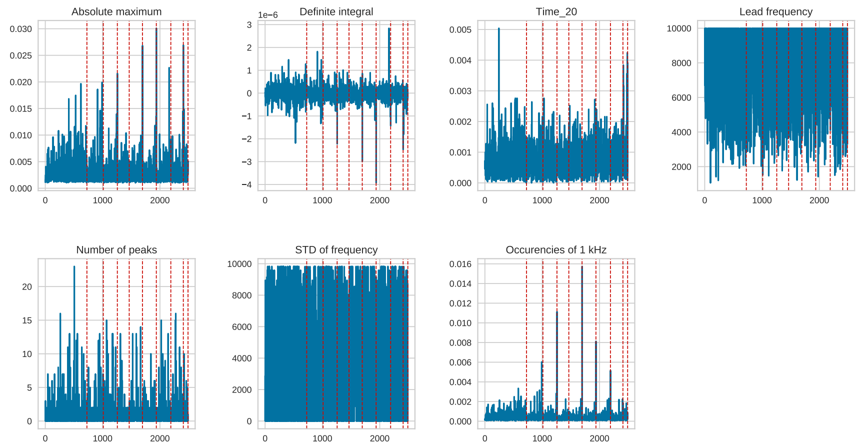


Figure 3.2: Distribution of features along the ramps, with each ramp ending with a dotted red line. The x axis represents the number of the event

seen in Table 3.1. We can see that the first variable (PC1) is made of mostly `lead_freq`, PC2 is made mostly of `time_20`, which is the dominant component also of PC3. We see an almost flat distribution in Figure 3.3 for PC1, which is also the component with the most variance.

Variable	PC1	PC2	PC3
`abs_max`	0.021	0.537	0.55
`def_int`	-0.007	-0.084	-0.062
`time_20`	0.006	-0.703	0.687
`lead_freq`	-0.998	0.027	0.040
`num_peaks`	0.011	-0.171	0.089
`std_freq`	0.005	0.191	0.212
`occ_1000`	0.054	0.378	0.410

Table 3.1: Decomposition of the variables after the PCA

Hopkins statistic test

The Hopkins statistic test [5] is a test which shows if a dataset is fit to be clustered. Given a set X of n data points, considering a random sample of $m \ll n$ points with members x_i, if we generate a set Y of m uniformly randomly distributed data points and we define u_i as the distance of $y_i \in Y$ from its nearest neighbour in X and w_i as the distance of m number of randomly chosen $x_i \in X$ from its

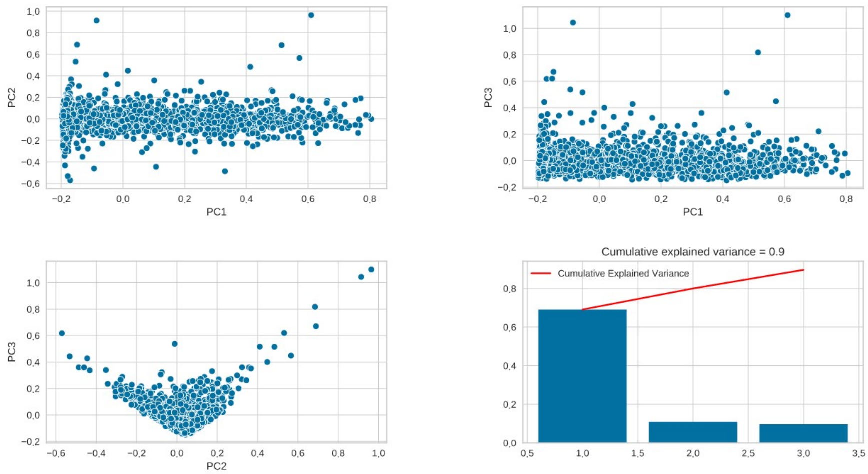


Figure 3.3: PCA using 3 variables (first three plots) and cumulative explained variance (bottom right plot)

nearest neighbour in X, then by defining

$$H = \frac{\sum_{i=1}^{m} u_i^d}{\sum_{i=1}^{m} u_i^d + \sum_{i=1}^{m} w_i^d} \tag{3.1.1}$$

d is the dimension of the dataset, $1 - H$ is the null-hypothesis. If it's close to 0 (i.e., if H is close to 1), then the data are likely to be suited for clustering. The Hopkins statistic test for this dataset returned $H = 0.81$, which means that this dataset could be suitable for a clustering algorithm. However, this does not guarantee that the dataset is clusterizable, but it only states that the dataset is not likely to be generated by a regularly spaced distribution.

3.1.2 Optimization: finding the best K

As explained in Section 1.3.1, there are many ways to test which is the optimal number of centroids K. We tested all the four different methods, and the results will be presented in the following

i) k-elbow: optimal value is $K = 6$, which is the value for which the slope of the cost function changes the most. See Figure 3.4

ii) Silhouette (Figure 3.5(a))

- Score: optimal value is$K = 3$, because it is the highest
- Negative values: optimal value is $K = 3$, because it is the lowest

iii) Consistency: optima value is $K = 3$, because it is the lowest (Figure 3.6)

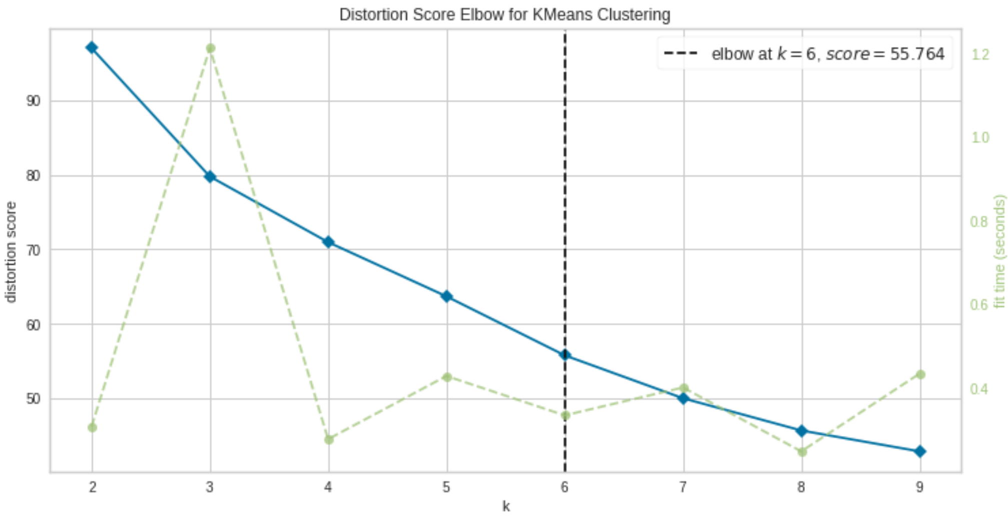


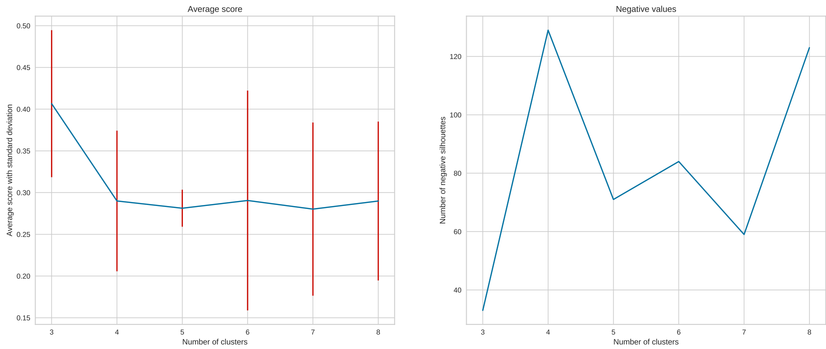
Figure 3.4: k-elbow analysis

Method	Best-K	Score
Elbow	6	56
Silhouette scores	3	0.4
Silhouette negative values	3	33
Consistency	3	3.23
WSS	6	50

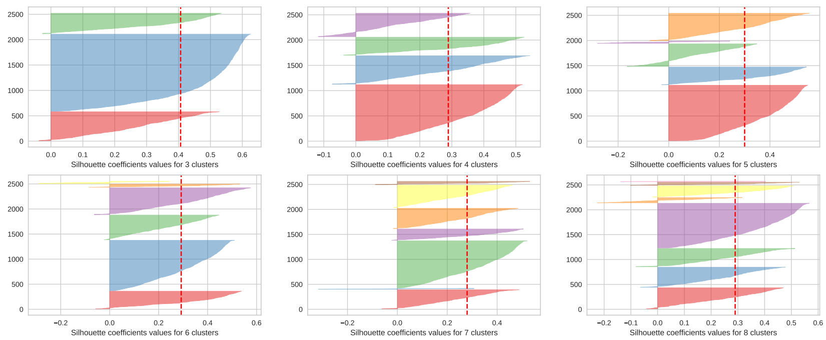
Table 3.2: Summary of the results for the 7-features analysis

iv) WSS cross validation: $K = 6$, because it is the value for which the slope changes the most(Figure 3.6)

Table 3.2 summarizes the optimization of K. We have two different results, and of these two we choose $K = 3$. We make this choice because looking at the silhouette (Figure 3.5(b)) for $K = 6$ we have one cluster made mostly of negative elements. This suggests that the clustering has not been optimal, and this suggestion is also confirmed by the average score, which is 0.29, much less than the score for $K = 3$. Using 3 clusters instead we have a high score for all the other tests, and also the silhouette visualizer shows all 3 clusters having average score well distributed with respect to the score of each event. I will now present my analysis using 3 centroids.



(a) Silhouette score (left) and negative values (right) analysis



(b) Silhouette visualization

Figure 3.5: Silhouette test: score (3.5(a)) and visualization (3.5(b))

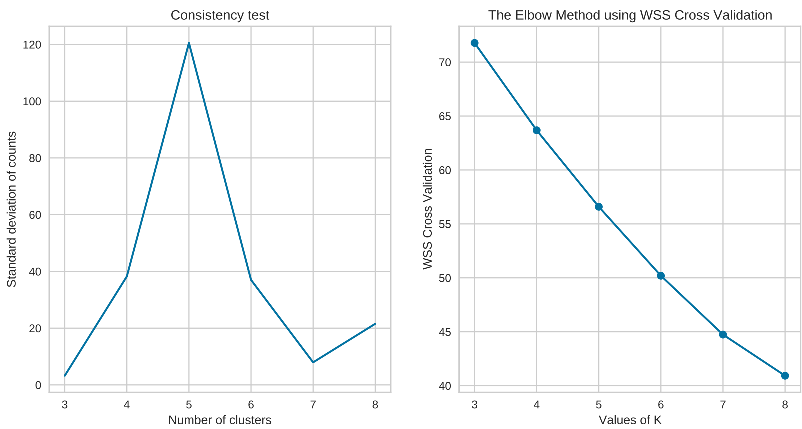


Figure 3.6: Consistency (left) and WSS cross validation (right) analysis

3.1.3 $K = 3$ analysis

Having chosen 3 centroids, we performed k-means using $K = 3$. The size of the clusters is as follows:

- Cluster 0 (color: Red) : 1520 events, 3 quenches
- Cluster 1 (color: Blue): 570 events, 1 quench
- Cluster 2 (color: Green): 401 events, 5 quenches

The centroid distribution along the ramp can be seen in Figure 3.7. We note that the nine quenches aren't clustered all together, but we see 3 quenches in cluster 0, 1 quench in cluster 1 and 5 quenches in cluster 2. Figure 3.8 shows where each quench has been clustered. We can see that quenches 3, 5 and 7 are clustered together, and this makes sense because they are very similar to each other, whereas the first quench is clustered alone, and this makes less sense because its features make it similar to quench 9. Quench 2 and quench 6 are also very similar, but they are clustered in different clusters.

To check the validity of our results, we also looked at the PCA, the parallel plot and the box plots.PCA and parallel plots are shown in Figure 3.9. It can be seen that the clusters are generated basing only on the first variable of the PCA (PC1), which, as we had seen from Table 3.1, is made almost completely of `lead_freq`. We note this because the first two plots of the PCA show the clusters well separated, whereas the third plot, which shows PC2 vs PC3, has all the events spread randomly. The parallel plot and the box plots (Figure 3.10) also give us the same information, because we see that the only feature for which there is a clear separation between the clusters is `lead_freq`. From the box plots we can also see that there are many events in the clusters that are far away from the mean values. To have so many outliers means that the clustering has not been efficient.

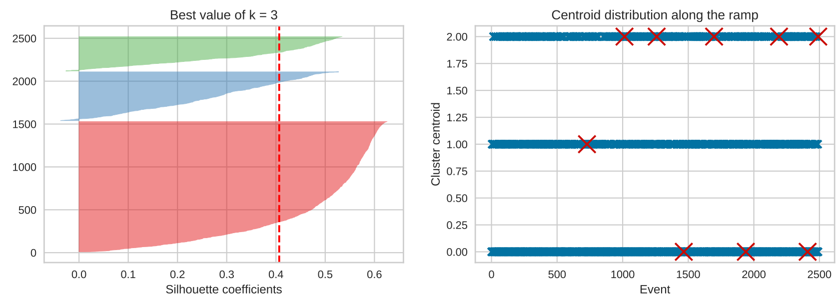


Figure 3.7: Silhouette (left) and centroid distribution along the ramp (right) for k-means using 3 centroids. The red crosses are the quench events

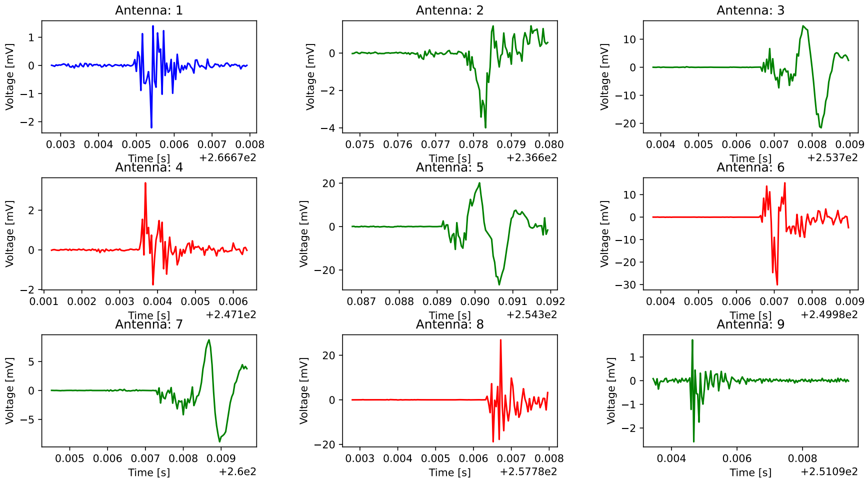


Figure 3.8: Plots representing in which cluster each quench belongs to

Summary

From all the considerations made above, we can say that the clustering with 3 centroids didn't provide acceptable results. The clustering of the quenches looks random, because we would expect some of them to be clustered together, and the whole clustering is based only on one feature, which is `lead_freq`. This feature isn't probably the best feature to describe the events because it is a "biased" feature, as we discussed in Section 3.1.1. Therefore, this whole clustering analysis is affected by that bias, and consequently the obtained results could not have a true physical meaning.

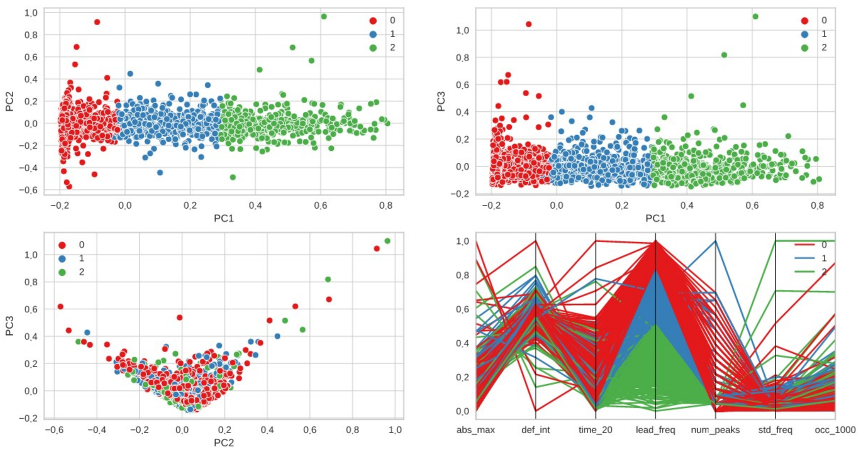


Figure 3.9: PCA and parallel plot for the three clusters

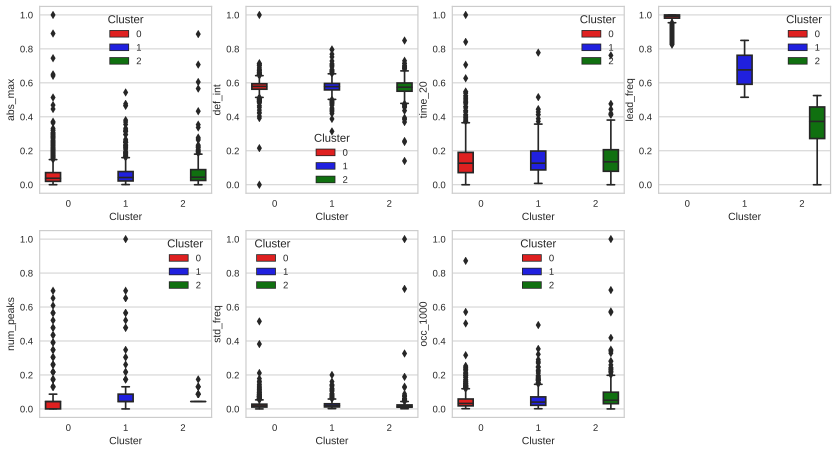


Figure 3.10: Box plots for the three clusters

3.2 3-features analysis

Unsatisfied with the results from the previous analysis, we decided to perform an analysis using only three features, one regarding the voltage, one regarding the shape and one regarding the frequency. The features that best captured the properties of the different events are `abs_max` (the absolute maximum value of voltage), `time_20` (the time between the maximum and 20% of the peak) and `occ_1000` (the number of times the frequency of 1 kHz is found in the frequency distribution at the time when the voltage is the highest). We performed the analysis the same way we did for the 7 features.

3.2.1 Data visualization

In Figure 3.1 we can see the correlation between the three features. We note that the events with a high value of absolute maximum voltage tend to be have a strong component of 1 kHz in the frequency distribution. Figure 3.12 shows the distribution of the three features along the ramp.

Since we now have only three features, we don't need to use a PCA. This is a great advantage compared to the prior analysis, because now we can link the events in the clusters directly to the features.

Hopkins statistic test

The Hopkins statistic test (Equation (3.1.1)) for this dataset returns a value $H = 0.97$. Therefore, this dataset is more suited to a clustering than the dataset generated by the 7 features.

3.2.2 Optimization: finding the best K

Same as we did for the previous analysis, we tested the four different methods to evaluate the optimal value of K.

i) k-elbow: optimal value is $K = 6$, as can be seen from Figure 3.13

ii) Silhouette (Figure 3.14):

- Score: optimal value is $K = 3$
- Negative values: optimal value is $K = 3$

iii) Consistency: optimal value is $K = 3$

iv) WSS Cross Validation: optimal value is $K = 6$

Table 3.3 summarizes the optimization of K. We have two different results, and of these two we choose $K = 3$. This choice was made because for six centroids

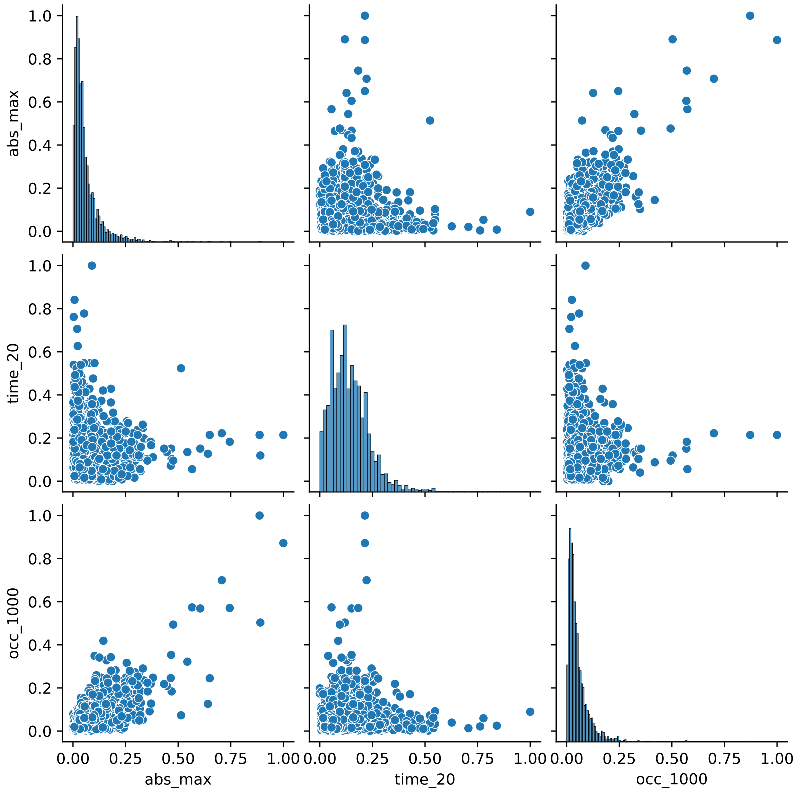


Figure 3.11: Correlation plot for all the features. The main diagonal represents the normalized distribution for each feature

we have the highest number of negative values and the lowest silhouette score. Therefore, we decided to use for this analysis $K = 3$ as well.

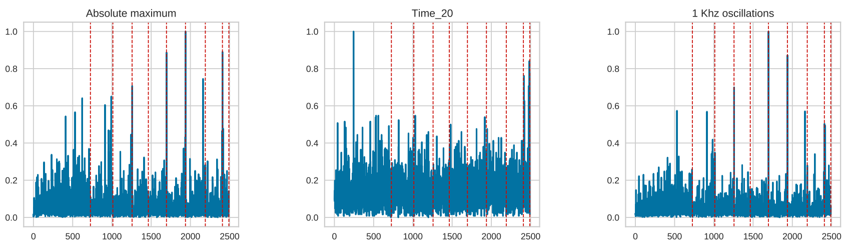


Figure 3.12: Distribution of features along the ramps, with each ramp ending with a dotted red line. The x axis represents the number of the event

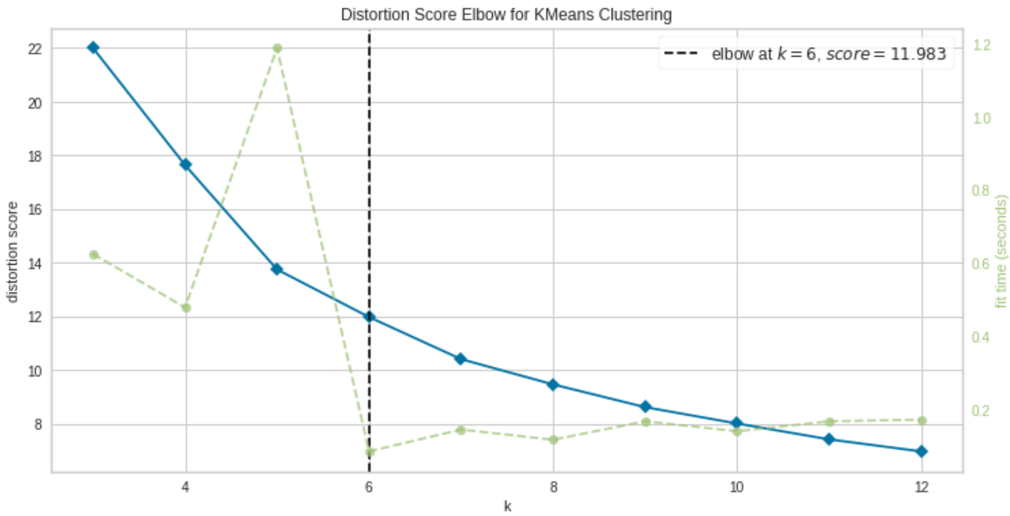
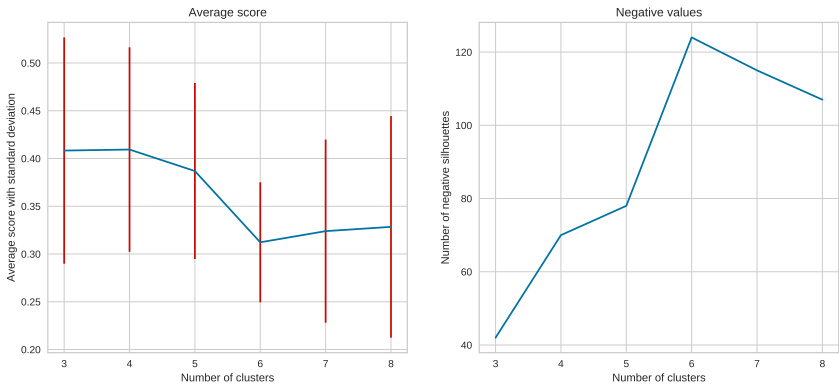
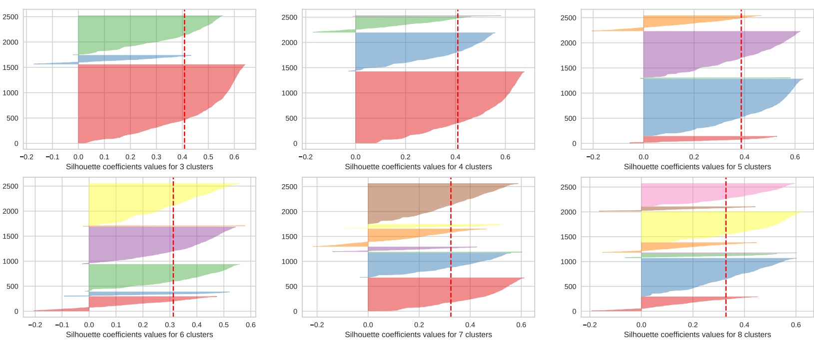


Figure 3.13: k-elbow analysis



(a) Silhouette score (left) and negative values (right) analysis



(b) Silhouette visualization

Figure 3.14: Silhouette test: score (3.5(a)) and visualization (3.5(b))

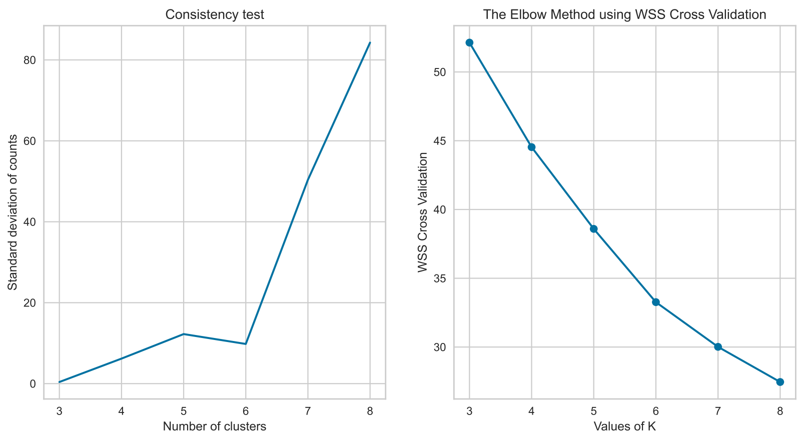


Figure 3.15: Consistency (left) and WSS cross validation (right) analysis

Method	Best-K	Score
Elbow	6	11.98
Silhouette scores	3	0.41
Silhouette negative values	3	42
Consistency	3	14.6
WSS	6	33

Table 3.3: Summary of the results for the 3-features analysis

3.2.3 $K = 3$ analysis

Having chosen 3 centroids, we performed k-means using $K = 3$. The size of the clusters is as follows:

- Cluster 0 (color: Red) : 1510 events, 3 quenches
- Cluster 1 (color: Blue): 813 events, no quenches
- Cluster 2 (color: Green): 168 events, 6 quenches

The centroid distribution along the ramp can be seen in Figure 3.16. We note that the nine quenches aren't clustered all together, but we see 3 quenches in cluster 0 and 6 quenches in cluster 2. Figure 3.17 shows where each quench has been clustered. The nine quenches are clustered as we would expect, because we have three low-voltage quenches clustered together, whereas the remaining six quenches show a strong component of the kHz oscillation.

To check the validity of our results, we also looked at the pair plot, the parallel plot and the box plots. As previously stated, there is no need for a PCA, therefore we can visualize the events simply in the pair plot. We note that the first two plots, which show `time_20` vs the other two features, show the events

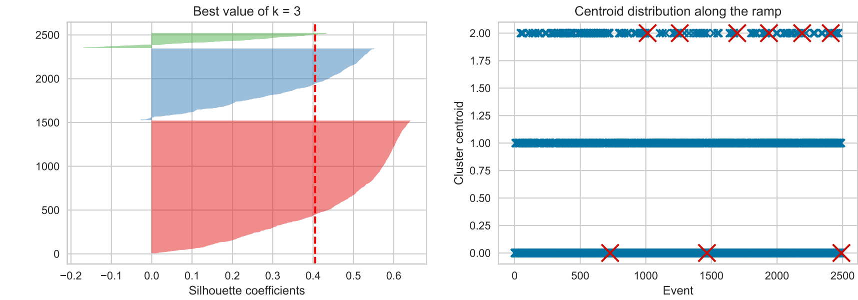


Figure 3.16: Silhouette (left) and centroid distribution along the ramp (right) for k-means using 3 centroids. The red crosses are the quench events

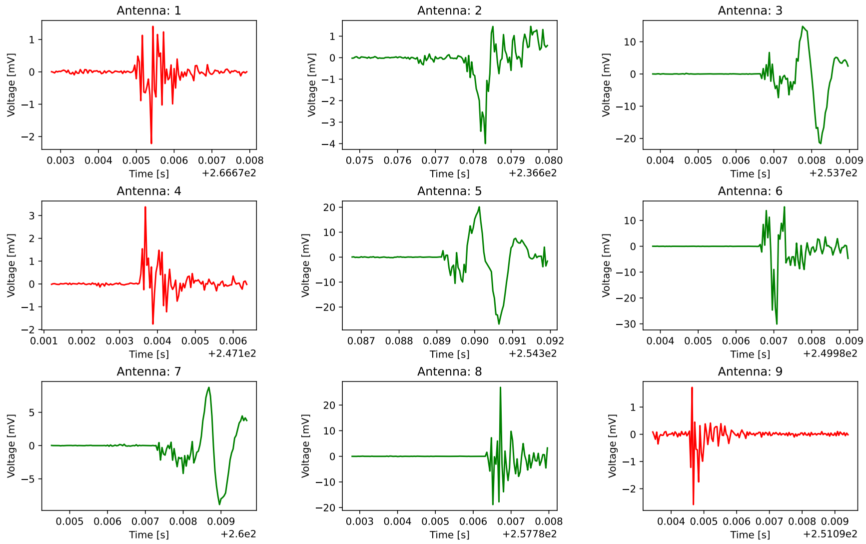


Figure 3.17: Plots representing in which cluster each quench belongs to

well separated. The third plot (`occ_1000` vs `abs_max`) doesn't distinguish well between the first two clusters (0 and 1), but it shows that the events with a high value of maximum voltage and of `occ_1000` are all clustered together, in the same cluster where the majority of quenches is (cluster 2). The parallel plot and the box plots (Figure 3.19) show the features well separated: cluster 0 has events with low `abs_max`, low `time_20` and low `occ_1000`; cluster 1 is made of events with the lowest value of `abs_max` and `occ_1000`, but with the highest vale of `time_20`; cluster 2 has events with the highest value of `abs_max` and `occ_1000`, with its `time_20` value is higher than cluster 0. The fact that we don't have as many outliers as we did for the previous analysis shows that this clustering has been more efficient.

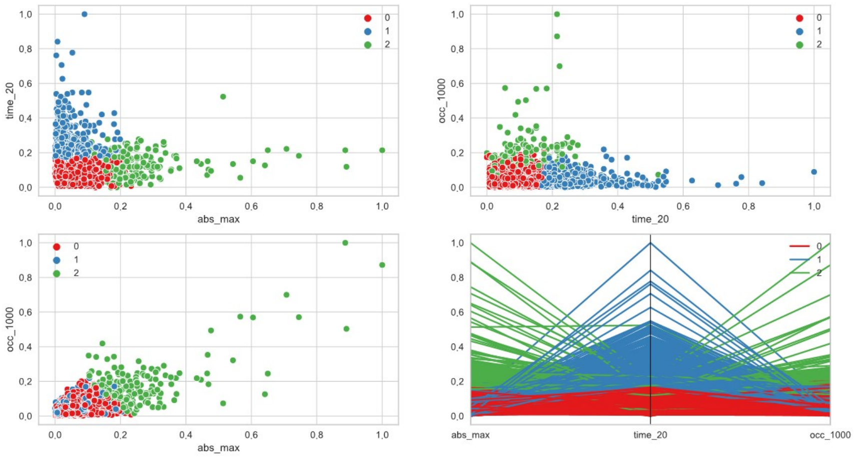


Figure 3.18: Correlation and parallel plot for the three clusters

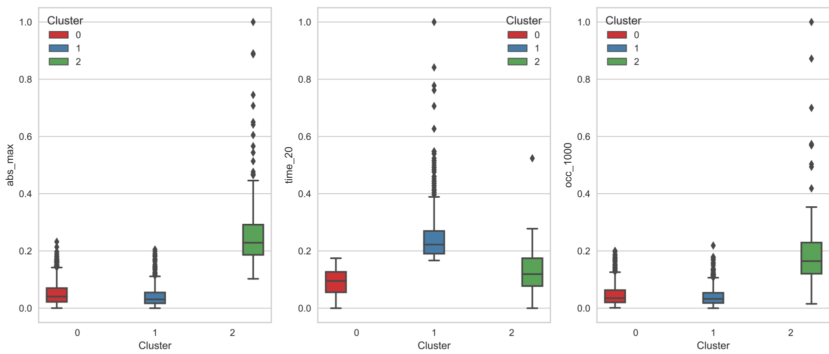


Figure 3.19: Box plots for the three clusters

Plots

Having validated the results of the clustering, we plotted many random events for each cluster to see if we could gain some information on what each cluster represents.

Figures 3.20 to 3.22 show 25 random event for each cluster. We can see some common characteristics for each cluster:

- Cluster 0 (Figure 3.20): they all look like mechanical events, because they show a spike followed by a fast decrease. Three quenches are found in this cluster
- Cluster 1 (Figure 3.21): mostly low-amplitude events, no quenches
- Cluster 2 (Figure 3.22): events which look like they could be generated by

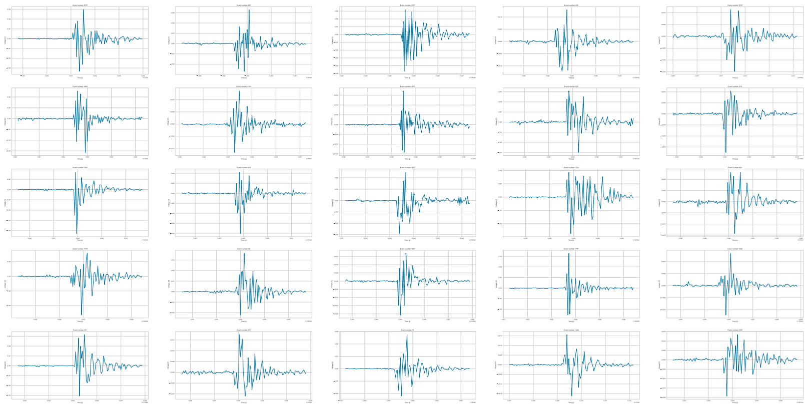


Figure 3.20: 25 random events for cluster 0

a current redistribution within the cable, much as the quenches

From this considerations, we can say that current redistribution within the cable looks to happen along the ramp. These events are similar to the quenches, and that is why they have been clustered together with the quenches. The quench is an event in which the current redistributes within the cable, but the different between the quench and the other events is that the quench events don't recover themself, and the voltage doesn't return to the normal value after.

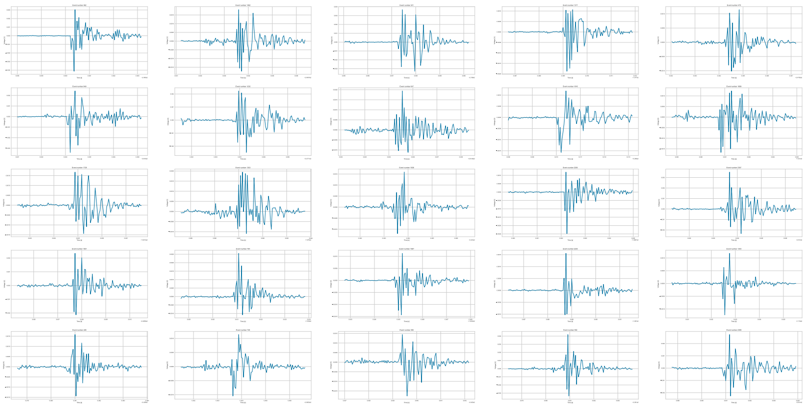


Figure 3.21: 25 random events for cluster 1

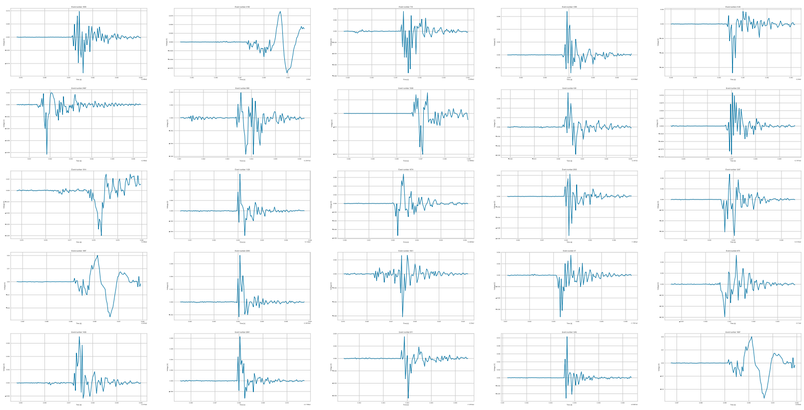


Figure 3.22: 25 random events for cluster 2

3.3 Summary and conclusions

Both the 3-feature and the 7-feature analysis showed that there are some events clustered together with the quenches. However, only the 3-features result can be considered, because the 7-features clustering wasn't optimal. From this analysis, we can then affirm that current redistribution appears to happen along the ramp, and sometimes it is recovered, whereas other times quench happens. This result could be useful in the future, because it will inform on the performance limits of the Nb_3Sn magnets.

Bibliography

[1] *Scipy.signal documentation.* https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.butter.html.

[2] *Scipy.integral documentation.* https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.cumulative_trapezoid.html.

[3] *Scipy.signal.find_peaks documentation.* https://numpy.org/doc/stable/reference/generated/numpy.linalg.norm.html.

[4] *Scipy.signal.find_peaks documentation.* https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html.

[5] A. Banerjee and R.N. Dave. "Validating clusters using the Hopkins statistic". In: *2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542).* Vol. 1. 2004, 149–153 vol.1. DOI: 10.1109/FUZZY.2004.1375706.