2023 Italian Summer Students program at FNAL
Final Report

Machine Learning for superconducting magnets application

Supervisor:
PhD Emanuela Barzi (FNAL)

Co-supervisor:
PhD Reed Teyber (LBNL)

Elisa Stabilini
Intern N° 49956N

Abstract

This report documents the work I conducted during my internship as part of the Italian Summer Students program at the Fermilab National Accelerator Laboratory (FNAL). Throughout my internship, I was stationed in the Technical Division within the Fermilab laboratory, where I was under the guidance of Emanuela Barzi and co-supervised by Reed Teyber from the Lawrence Berkeley National Laboratory (LBNL).
The aim of the project is the study and characterization of quench antenna signals in order gain understanding on the mechanical and electromagnetic phenomena happening inside superconductive magnets. The results presented here identify some potential areas of investigation and potential directions for improving the construction of magnetic superconductors, particularly within the framework of the CCT subscale program at LBNL.

Acknowledgments

I would like to express my sincere gratitude to my supervisors, Emanuela Barzi and Reed Teyber, for their patience and guidance throughout my summer project. Many thanks to Alberto Plebani for his availability on answering all my many questions. A particular thanks goes also to Silvia Picchi and Sara Gamba who constantly supported me during these months.

Contents

1 Superconducting Magnets **5**
1.1 Superconductive materials . . . 5
1.2 Quench . . . 5
1.2.1 Magnets training . . . 6
1.2.2 Quench antenna . . . 6

2 K-means clustering **8**
2.1 Algorithm description . . . 8
2.2 Clustering evaluation . . . 8

3 First data analysis **11**
3.1 Code study and development . . . 11
3.2 Analysis in Feature Space . . . 11
3.3 K-Means clustering . . . 14

4 Second data analysis **15**
4.1 Data preprocessing . . . 15
4.2 Event selection & statistical analysis . . . 16
4.3 Event analysis . . . 16
4.4 Feature analysis . . . 18
4.4.1 Statistical analysis on the events . . . 19
4.5 Principal Components Analysis . . . 19
4.6 Clustering results . . . 21

5 Conclusions **25**

Bibliography **26**
5.1 Appendix A - code documentation and organization . . . 28
5.2 Appendix B - examples of nonphysical data . . . 30
5.3 Appendix C - Features distribution over time . . . 30
5.4 Appendix D - Distribution of cluster's feature . . . 31

Chapter 1

Superconducting Magnets

In the last decades, superconducting magnets played an important role in advancing experimental sciences and engineering [15]. They find a particularly important application in modern high energy physics (HEP) research [3] as they are a key element in circular accelerators, where the accessible scale of investigation strongly depends on increasing the magnetic bending of the particle beams.

There are different reasons behind the choice of using superconducting magnets instead of their classical counterparts firstly they conduct high electrical current density with zero resistance. This allows magnets to produce substantially higher magnetic field strengths while avoiding the significant energy losses associated with standard electromagnets. They are also more cost-effective and smaller in size than their traditional counterparts.

Magnets are deployed at various points within the accelerator: they are used to guide and collimate the beam on a quasi-circular orbit, to transfer beams between accelerator rings in so-called transfer lines, to direct or separate beams for the accelerating radio-frequency cavities, and to focus beams for collision at the interaction points where the experiments reside. Due to their various applications, the magnet system of an accelerator is composed of a number of different magnets, mainly dipoles and quadrupoles. [16]

Despite the various positive aspects of using superconductors within magnets, there are also some drawbacks, primarily due to quenching. To enhance the performance of magnets and expand their operational range in the United States, an important research and development program, the US Magnet Development Program (US-MDP), has been established. This work is focused on a model of a canted-cosine-theta (CCT) superconducting magnet currently under study at LBNL [13], [2].

1.1 Superconductive materials

Superconductive materials are materials that, under particular conditions of current density J, temperature T, and magnetic field B, can reach a zero-resistance conductive state. The superconductive state is delimited by a critical surface in the phase space, defined by the critical values of the same three quantities, J_C, T_C, and B_C. The values of these critical surfaces are highly dependent on the mechanical and electromagnetic properties of the materials. In Figure 1.1, we illustrate the critical surfaces of two widely utilized superconducting technology alloys: namely, Niobium-Tin and Niobium-Titanium.

When a superconductor transitions from a state below the critical surface to one above it, it experiences a phase change, converting it into a typical resistive conductive state. This transformation poses a potential hazard in the context of superconducting magnets.

1.2 Quench

The term quench indicates the *sudden* and *irreversible* transition of a portion of the conductor to a standard resistive state inside a superconducting magnet. Quenching phenomena can be caused by both electromagnetic and mechanical events that happen inside the magnets. For example, quenching phenomena can be due to mechanical events such as epoxy cracks and fractures, debonding, friction, and motion at layer-to-layer and cable-in-groove interfaces. or electromagnetic phenomena such as current redistributions. One of the goals of current research is to identify

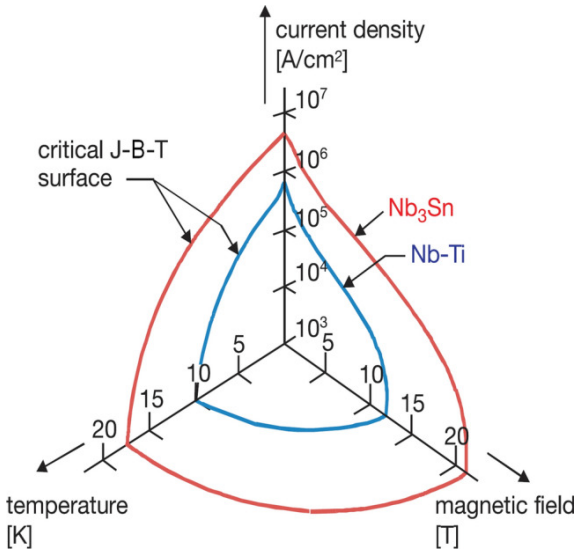


Figure 1.1: Source: [8]. Representation of the critical surfaces for a Nb_3Sn and NbTi alloy in the phase space.

differences in data of differently originated events in order to improve intervention and develop higher-performance magnets.

1.2.1 Magnets training

Before reaching their working conditions, namely the magnet field, current density, and temperature the magnet was designed for, magnets undergo a training phase. During the training, the current density inside the superconductor is progressively and slowly increased until quenching occurs (current ramp). At this point, the Quench Protection System activates in order to dissipate the energy and prevent irreversible damage to the magnets.
Magnets undergo various training cycles, commonly known as thermal cycles, during which the magnet's response is studied, and measurements are taken accordingly. Each thermal cycle consists of multiple current ramps, with each ramp halting upon the occurrence of a quenching event.
Furthermore, between different thermal cycles, it is possible to identify issues in the magnet's mechanical or electronic components.

1.2.2 Quench antenna

Quench antennas or QAs, are arrays of pickup coils designed to detect changes in magnetic flux. They consist of a printed-circuit-board (PCB) antenna integrated into the Nb_3Sn superconducting magnet's coil structure. Due to its close proximity to the magnet, this antenna offers enhanced sensitivity to current redistribution, strand motion, and cable motion.

Quench Antennas serve a dual purpose by identifying and pinpointing the location of quenches in superconducting systems. These antennas exhibit sensitivity not only to dynamic cable events, such as strand slip-stick motion but also to stationary current redistribution, which may or may not be preceded by a mechanical disturbance. Figure 1.2 shows QAs positioned in the magnet.

Each QA measures the change in voltage according to Faraday-Neumann-Lenz's low as

$$V = \sum_{i=1}^{n} \frac{\mathrm{d}}{\mathrm{d}t} \left(\int B \mathrm{d}A_i \right); \tag{1.1}$$

where V represents the voltage variation measured by the antenna, the index n is the count of coil loops wrapped around the magnet, B is the magnetic field inside the magnet, and A_i represents the area of each individual loop of the antenna oriented perpendicular to the magnetic field.

More in detail, within each of the CCT subscale magnets developed at LBNL, a configuration of six antennas is arranged as illustrated in Figure 1.3. The labeling system employs two sets of numbers, denoted as R and L, ranging from 1 to 6. These designations represent the right and left

edges of each antenna, respectively. Each antenna is designed to measure the voltage difference between the corresponding symmetric edges of the coil. For instance, antenna QA6 measures the voltage difference between the points 6R and 6L.

(a) Edge of the antenna on the magnet (b) Antenna positioned on the magnet

Figure 1.2: Source: unknown. The images show the antennas positioned on the magnets at LBNL

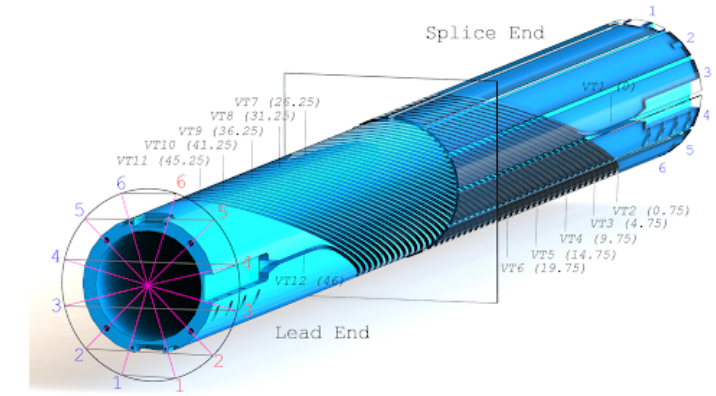


Figure 1.3: Source: unknown. The image shows a scheme of the antenna's positions around a CCT magnet at LBNL

Chapter 2

K-means clustering

Clustering is one of the most common exploratory data analysis techniques used to get an intuition about the structure of the data. The goal of performing clustering is to identify subgroups in data such that data points in the same cluster are very similar to other data points inside the cluster and different from data points in other sub-groups.

2.1 Algorithm description

As I was continuing the work described in [12], I decided to apply K-means clustering for my data-analysis.
The K-means algorithm is an iterative algorithm that tries to partition the dataset into a pre-defined number K of distinct and non-overlapping clusters, where each data point belongs to only one group. The goal of the algorithms is to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible.

The K-means algorithm works as follows, first, the number K of clusters is specified, and then it proceeds in two iterative steps:

1. in the *assignment step* each data point is assigned the closest cluster's centroid. Initially, the centroids are initialized by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement.

2. in the *update step* the centroids are adjusted to match the sample means of the data points that they are responsible for

The two steps are repeated iteratively until there is no change to the centroids and the assignment of data points to clusters is not changing [9].
The algorithm assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid is at the minimum, meaning that the objective function 2.1 is minimized.

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} ||x^i - \mu_k||^2, \tag{2.1}$$

where $w_{ik} = 1$ for data point x_i if it belongs to cluster k, otherwise $w_{ik} = 0$ and μ_k is the centroid of x_i's cluster.
The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

2.2 Clustering evaluation

Despite being straightforward to use, the K-Means algorithm encounters some limitations when employed with large datasets. First of all, the algorithm complexity in time is $\mathcal{O}(tkn)$ where t is the clock time for what we assume to be an iteration of the algorithm, n is the number of objects and k is the number of clusters, this results in impractical computational demands for extensive datasets.
Another issue is related to the necessity of specifying the number of clusters and the sensitivity of

the algorithm to the initialization. Selecting the optimal number of clusters is often an intricate and subjective task, and making an incorrect choice can yield sub-optimal results. Moreover, K-means is sensitive to the initial placement of cluster centroids. Different starting points for centroids can lead to divergent cluster assignments, causing instability in the results and making it difficult to guarantee the robustness of the clustering outcome [7] [5].

To address these problems I chose to evaluate the algorithm's performance while varying the number of clusters. This evaluation helps in determining an optimal number of clusters that best represent the underlying structure in the data. In the following paragraphs I describe the evaluation techniques I used.

K-elbow The elbow method is a graphical method for finding the optimal K value in a k-means clustering algorithm. The elbow graph shows the within-cluster-sum-of-square (WCSS) values on the y-axis corresponding to the different values of K (on the x-axis). The optimal K value is the point at which the graph forms an elbow. Fig 2.1 shows an example of a K-Elbow plot.

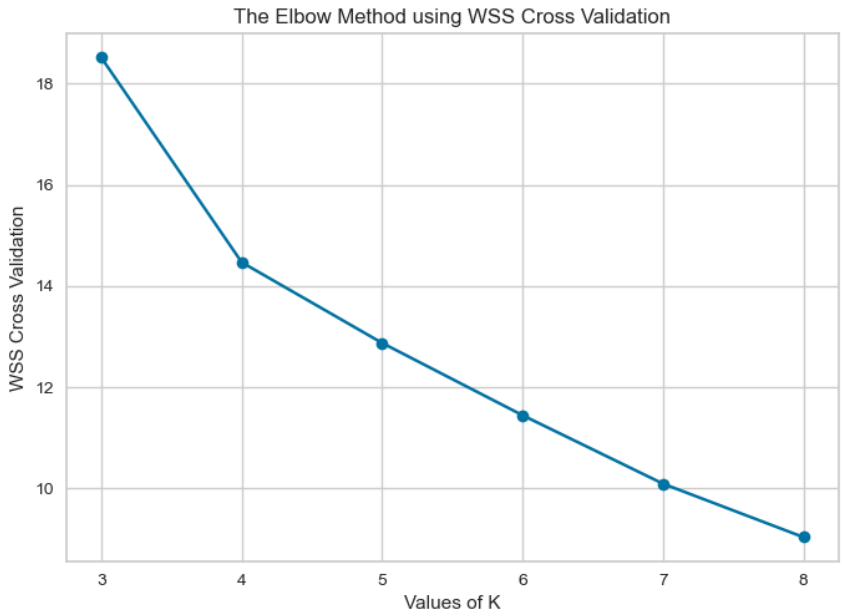


Figure 2.1: Elbow plot used to determine the best K for QA1 data for the first thermal cycle

Silhouette analysis Silhouette analysis is an intrinsic method used to assess the quality of clustering results by examining how well the clusters are separated and how tightly grouped their members are. For each data-point i in the cluster C_I we compute the mean distance between itself and the other data points within the same cluster using equation 2.2

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} \text{dist}(i, j) \tag{2.2}$$

where $|C_I|$ is the number of points belonging to the cluster $|C_I|$ and $d(i, j)$ is the distance between the data-points i and j.
Then we define the mean dissimilarity of the data-point i to the other clusters C_J, with $I \neq J$. So for each point $i \in C_I$

$$b(i) = \min_{j \neq i} \frac{1}{|C_J|} \sum_{j \in C_J} \text{dist}(i, j) \tag{2.3}$$

The cluster characterized by the smallest average dissimilarity is referred to as the "adjacent cluster" to object i, as it represents the most suitable alternative cluster for the data-point i.
Then, we compute the silhouette value of the data-point i as

$$s(i) = \begin{cases} \frac{b(i)-a(i)}{\max a(i), b(i)}, & \text{if } |C_I| > 1 \\ 0, & \text{if } |C_I| = 1 \end{cases} \tag{2.4}$$

subsequently $-1 \leq s(i) \leq 1$. A negative coefficient indicates that an object is more proximate to objects in different clusters than to those within its own cluster, signifying a sub-optimal clustering arrangement. On the other side, when the coefficient approaches 1, the objects are tightly grouped within well-defined clusters, reflecting a favorable clustering outcome.
For every object in the dataset, we compute the silhouette coefficient $s(i)$, and then, we calculate the average silhouette score across all clusters. This collective average is termed the silhouette score. To determine the optimal number of clusters k, we select the value that maximizes the silhouette score while simultaneously minimizing the count of negative silhouette coefficients. This approach ensures the identification of a clustering configuration that results in well-separated and internally coherent clusters.

WCSS cross validation The Within-Cluster Sum of Squares (WCSS) Cross Validation is a technique involving the division of the dataset into m subsets, where $m < n$ and n represents the total number of data points. Subsequently, a clustering model is constructed using $m - 1$ of these subsets, leaving one subset aside for evaluating the quality of the clustering.
In the evaluation phase, for each data point within the reserved subset, the closest centroid is determined based on the clustering model. Then, the quality of the model's fit to the test set is quantified by calculating the sum of the squared distances between all data points in the test set and their respective closest centroids.
This process is repeated m times for each candidate value of k (the number of clusters), similar to the approach employed in the elbow method. The optimal value of k is determined by identifying the point at which the rate of change in the cost function, often represented as the WCSS, exhibits the most significant transition.

Consistency test The consistency test is a method employed to assess the reproducibility of clusters obtained through clustering analysis. When determining the optimal number of clusters, denoted as k, a consistency test is conducted by repeatedly applying the clustering algorithm N times, typically around 100 iterations. During each iteration, clustering is performed with the same k value.
The assessment involves calculating the standard deviation of the number of data points within the largest cluster for each iteration. The rationale behind this procedure is to identify the optimal k value, which is characterized by the lowest standard deviation. A lower standard deviation signifies that across multiple iterations, similar clusters are consistently generated, indicating greater stability and reproducibility in the clustering results.

Chapter 3

First data analysis

As I mentioned in previous sections, during my internship I worked with data coming from a quench antenna located inside CCT magnets at LBNL.

Datasets description My work on these data was divided into two phases, a first phase during which I mostly worked on the algorithm and on the code using data already analyzed by Alberto Plebani [12], and a second phase during which I used the previously developed code and routine-analysis to analyze data coming from new magnets.

For this reason, I will often refer to these two different data sets, the first smaller one which contains data regarding a single antenna for four different current ramps to quenches. This dataset was mostly used to run tests on the code and optimize the data analysis algorithm; for this reason, I analyzed the first one. In the present chapter, I will focus on the approach I used and developed to analyze this first set of data.
The second dataset, as I will discuss more in-depth in section 4, is instead a wide dataset including data coming from different thermal cycles and even different magnets. This difference obviously had consequences on the choice of the data analysis approach.

The voltage data I analyzed are saved in `.npy` format with a standard name convention. Each file is saved as `X00_00000.npy` where the first three symbols identify the ramp along the training where data come from, the second number series is the code that associates the raw filenames from the data acquisition system (DAQ) to the quench.

3.1 Code study and development

As first step, I reorganized and developed the code, written and used by Alberto Plebani in [12], aiming to simplify its usage and tried to build some documentation on the usage of the main functions for signal analysis, event detection, and study. A complete description of this work and of the code organization is available in appendix 5.1. Here I want to stress that I kept the same event definition used in [12]: *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* $40\mu s$. Regarding the threshold, I used a 1 mV threshold in order to have more events possible and therefore more statistics.

At this point, I tested the new function and the new data analysis routine on data registered by QA6 during the current ramp `A04_00010`. In Figure 3.1 I show a few examples of events extracted from these data.

3.2 Analysis in Feature Space

The first step of feature analysis consisted of extracting interesting features from event data. In particular, as Alberto did, I identified three main groups of features interesting to analyze that could help us characterize voltage signals.

Voltage features

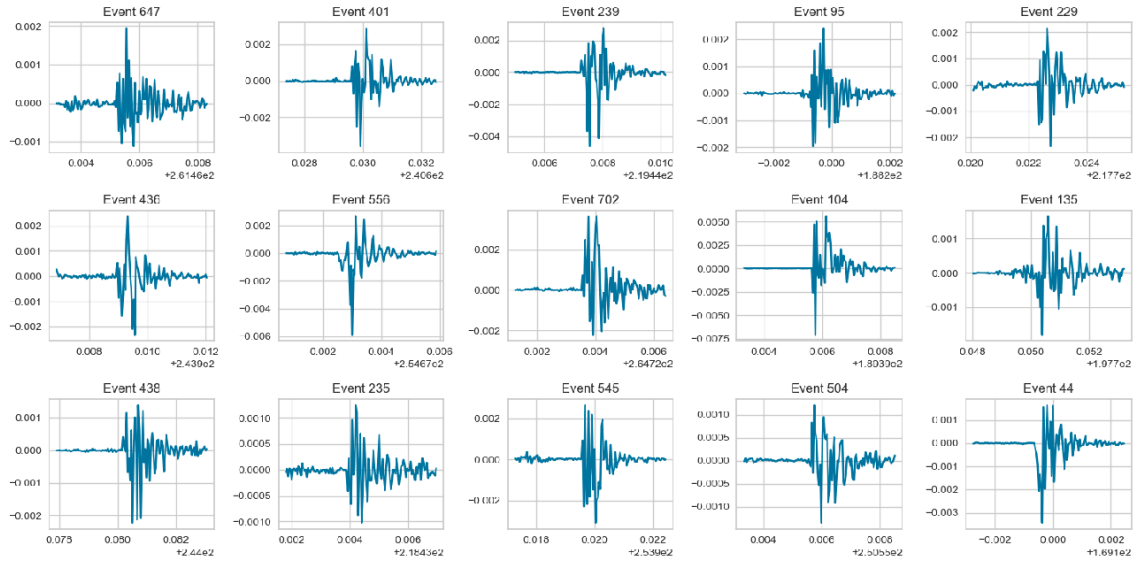


Figure 3.1: Examples of the events detected in `A04_00010` using the described algorithm

- 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.
- Minimum voltage: it is the value of the minimum voltage in the windowed event. It can be used instead of the maximum value as sometimes the negative spike is higher in absolute value than the positive spike
- Absolute maximum: it is the absolute maximum value of the voltage in the windowed event.
- Maximum integrated voltage: it is the highest value of the integral of the voltage. The integral of the voltage is a useful variable that can give information 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 [14].
- Time difference between maximum and minimum: measures the difference in time between the maximum and minimum in voltage
- norm of the voltage: the Euclidean norm of the voltage array, evaluated using the `linalg.norm` function implemented in the `numpy` package[4]

Signal features

- 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 restores itself to its normal value after the spike
- Time 50: it is the difference between the time the voltage reaches 50% of the peak and the time of the absolute maximum
- Time 30: it is the difference between the time the voltage reaches 30% of the peak and the time of the absolute maximum
- Time 20: it is the difference between the time the voltage reaches 20% of the peak and the time of the absolute maximum

Frequency features Following the path traced by Alberto Plebani in his work [12], the frequency analysis of the signal was made using a Continuous Wavelet Transformation (CWT), for more details on the analysis see for example [1]. In particular, I considered the following features resulting from the CWT:

- CWT leading frequency: The leading frequency I obtain from the CWT frequency analysis

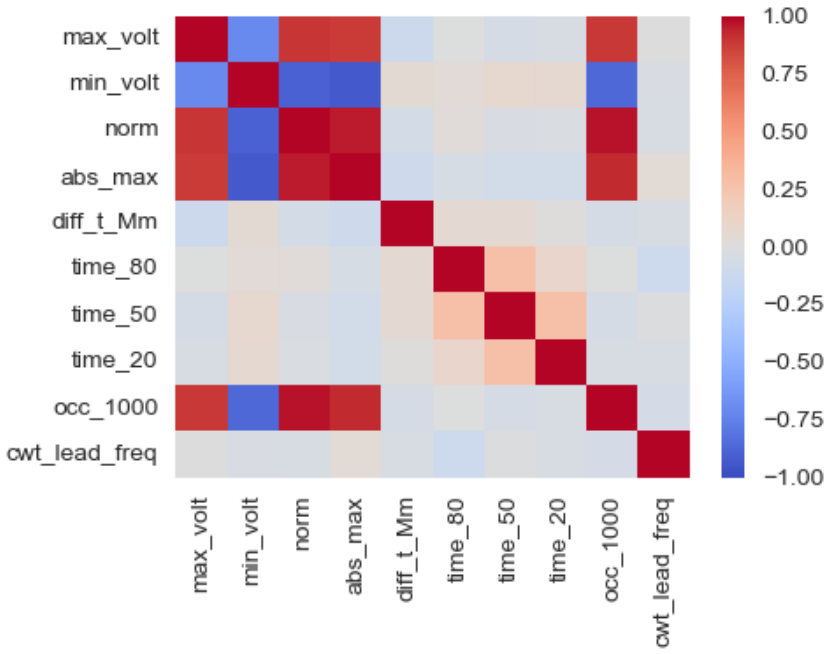


Figure 3.2: Correlation matrix for the mentioned features. We are looking for non-correlated features.

- Occurrence 1000: Number of times the frequency of 1 kHz is found in the frequency distribution at the time when the voltage is the highest.

Feature correlation

Starting the analysis from the results of Alberto's work I first considered the single features trying to understand which of them could be used to achieve an efficient clustering. For this reason, I started with a correlation study of selected features. In Figure 3.2 I show the heat map for the correlation plot.

Using this heat map it's immediate to see which features are correlated. In particular, we see as expected, that all the voltage features are correlated. The most interesting information we get about the feature correlation is that the time difference between the maximum and minimum voltage is independent of every other feature. Upon conducting further analyses, we observed that the absence of any significant correlation between this particular feature and others is attributed to its behavior resembling white noise. After further discussions, we concluded that the sampling frequency employed for data collection is inadequate for extracting meaningful information from this feature. On the other side, the plot shows a strong (positive or negative) correlation between the occurrence of the 1000 frequency and the voltage features.

Studying feature correlations before applying K-means clustering is crucial for enhancing efficiency. When features are highly correlated, they provide redundant information to the algorithm, potentially leading to inaccurate clustering. Analyzing feature correlations allows for the identification and elimination of redundant features, improving computational efficiency and ensuring that the clustering process is based on the most relevant, independent features. This results in more accurate and meaningful clusters, optimizing the data analysis process.

3.3 K-Means clustering

Starting from the previous considerations on features correlation, I decided to apply the clustering algorithm to the events identified using three features: `abs_max`, `time_20`, and `cwt_lead_freq`. The result obtained using the new code is similar to that obtained in [12]: the best choice has been to use three clusters that differentiate the events based on the values of the aforementioned three features. In Figure 3.3, I illustrate how the events shown in Figure 3.1 have been divided into clusters.

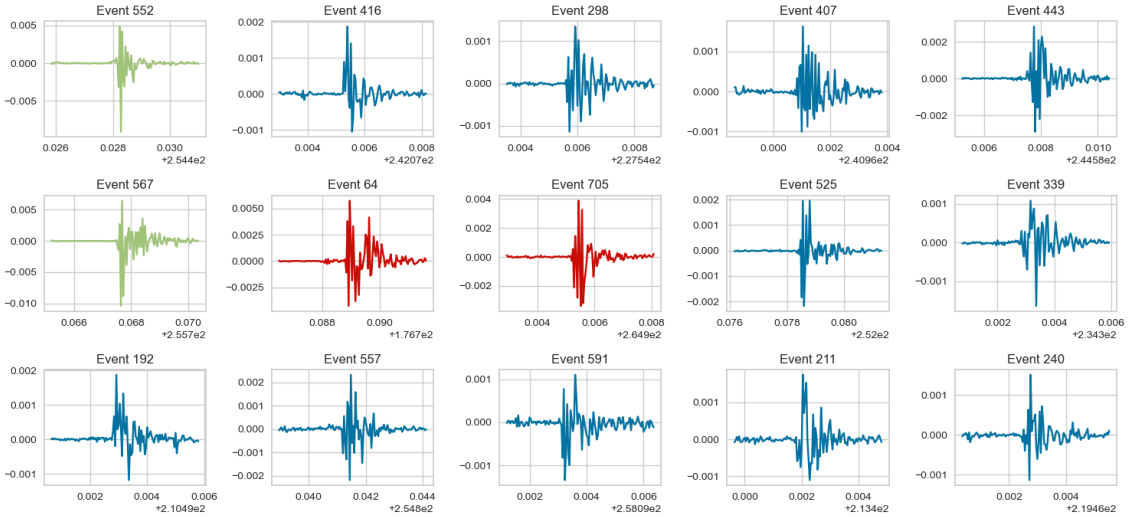


Figure 3.3: Event samples associated to each cluster

Chapter 4

Second data analysis

New dataset

In this chapter, I will focus on the analysis I carried out on the second slot of data I used during the second part of the analysis.
The main difference between the new dataset and the test one consists of the dimension. In the new dataset, I have much more data to analyze, moreover, it's raw data that needs pre-processing in order to be cleaned from underlying noise. The new dataset consists of data from two magnets:

- sub_2: The subscale 2 magnet which is particularly important as it provides baseline data for other magnets analysis, is impregnated with standard epoxy (NHMFL mix-61).
- sub_4: The subscale 4 magnet is impregnated with ultra-tough, or non-epoxy, epoxy (CTD 701X)

For the sub_2 magnet I analyzed data coming from three different thermal cycles, identified respectively as `thermal_1`, `thermal_2`, `thermal_3`. The thermal cycles I analyzed were characterized by a different number of ramps to quench: 23 ramps for the first thermal cycle, 32 ramps for the second thermal cycle, and 31 ramps for the third thermal cycle. For the sub_4 magnet, I only analyzed a single thermal cycle, `thermal_1`, consisting of 13 ramps to quench.
It is also important to stress that the data along these different thermal cycles were sampled with different frequencies. For magnet sub_2 the sampling frequency was 10 kHz for the first thermal cycle, 20 kHz for the second, and 25 kHz for the third thermal cycle. Meanwhile, the sampling frequency for magnet sub_4 was 25 kHz.

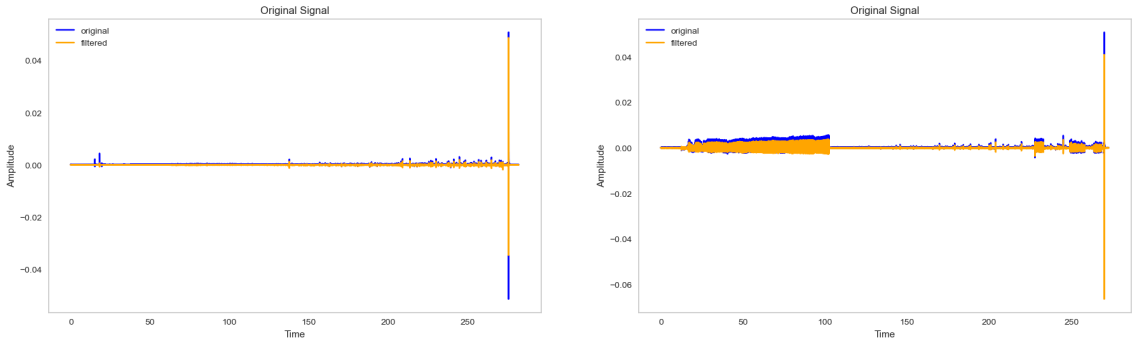
This second dataset I got to analyze was substantially different from the first one, main practical difference consisted in the amount of data I had to analyze, for this reason, I had to change the data analysis approach.

4.1 Data preprocessing

The new dataset consisted mainly of raw data from different quench antennas, in particular, each file contains data regarding the voltage measured by each antenna and voltage decay due to the Hall effect measured by Hall sensors. An example of data from the new datasets is represented in Figure 4.1. In Figure 4.1, the first 12 columns contain the voltage values measured by the leading (e.g. QALd_1) and the splice (e.g. QASp_1) edges of the six antennas.

	QALd_1 [V]	QALd_2 [V]	QALd_3 [V]	QALd_4 [V]	QALd_5 [V]	QALd_6 [V]	QASp_1 [V]	QASp_2 [V]	QASp_3 [V]	QASp_4 [V]	QASp_5 [V]	QASp_6 [V]	volt_IL [V]	Hall_V [V]	volt_OL [V]
0	0.000209	-0.000857	0.000211	0.000192	0.000234	0.000005	0.000289	0.000052	0.001146	-0.000483	0.000025	-0.000142	0.003746	-0.004563	0.010307
1	0.000214	-0.000857	0.000212	0.000197	0.000234	0.000003	0.000292	0.000053	0.001146	-0.000482	0.000025	-0.000146	0.002810	-0.004544	0.010307
2	0.000209	-0.000853	0.000208	0.000194	0.000234	0.000006	0.000289	0.000052	0.001140	-0.000482	0.000025	-0.000144	0.004058	-0.004556	0.010307
3	0.000212	-0.000856	0.000212	0.000190	0.000242	0.000009	0.000290	0.000050	0.001143	-0.000478	0.000023	-0.000146	0.003434	-0.004538	0.010619
4	0.000215	-0.000853	0.000214	0.000195	0.000237	0.000006	0.000289	0.000053	0.001144	-0.000478	0.000028	-0.000146	0.002810	-0.004563	0.010619

Figure 4.1: Example of how data are organized in the data file and uploaded



(a) Example of QA signal

(b) Example of nonphysical signal

Figure 4.2: Comparison between a physical (first subfigure, current ramp A23_00011) and a non-physical signal (second subfigure, current ramp A11_00022)

To adapt the previously developed code for the analysis of new data, I organized the data in a structure similar to the one used in the prior analysis. Given the specific focus of this work, for each current ramp, I concentrated on collecting and storing relevant information only for the leading edge of the antenna. For this purpose, I employed the pandas library [10] to create and manage data frames. I set up these data frames to include information from all six antennas for every current ramp in each thermal cycle.

Looking ahead, it could be a good idea to also include data from splice edges and Hall sensors in future analyses. These extra data sources might give us more helpful insights and make our clustering techniques work even better, especially when we're trying to understand events connected to how current moves in the superconductors.

Data filtration

At this point, I conducted a preliminary analysis to gain insights into the signals under examination. This basic analysis helped me spot and filter out data associated with current ramps that had anomalies, mostly linked to issues in the Data Acquisition (DAQ) system.

In Figure 4.2, I present a visual comparison between a physical signal and other signals that exhibit irregularities. It's worth noting that the code I used proved to be quite robust; it reliably identifies and separates events, even in tricky or unusual cases (see Appendix 5.2)

Figure 4.2 also shows the result of the application of a high-pass filter in order to clean the signal. In particular, I used the function `signal.butter` from the `scipy` library [14] applying a 6th order filter, with a cutoff frequency of 400 Hz.

4.2 Event selection & statistical analysis

Threshold selection In the new data analysis, one of the key considerations is choosing the right thresholds for our event detection algorithm. The threshold used in the previous data analysis was too high for the initial current ramp during the first thermal cycle. So, for the analysis of 'magnet sub_2,' we did it twice – first with a 1 mV threshold and then with a lower 0.5 mV threshold. On the other hand, I stuck with a 1 mV threshold for the analysis of 'magnet sub_4.' This adjustment in threshold values is important to make sure we get accurate results when identifying events in our data. After selecting the correct threshold I proceeded to identify the events in all current ramps of the different thermal cycles and magnets.
In Figure 4.3 I show 20 events randomly extracted from all the events identified in the different ramps of the first thermal cycle.

4.3 Event analysis

I then concentrated on the statistical analysis of the events I had identified. Unlike the previous dataset, the amount of data in this case was much larger. That's why I started with some general

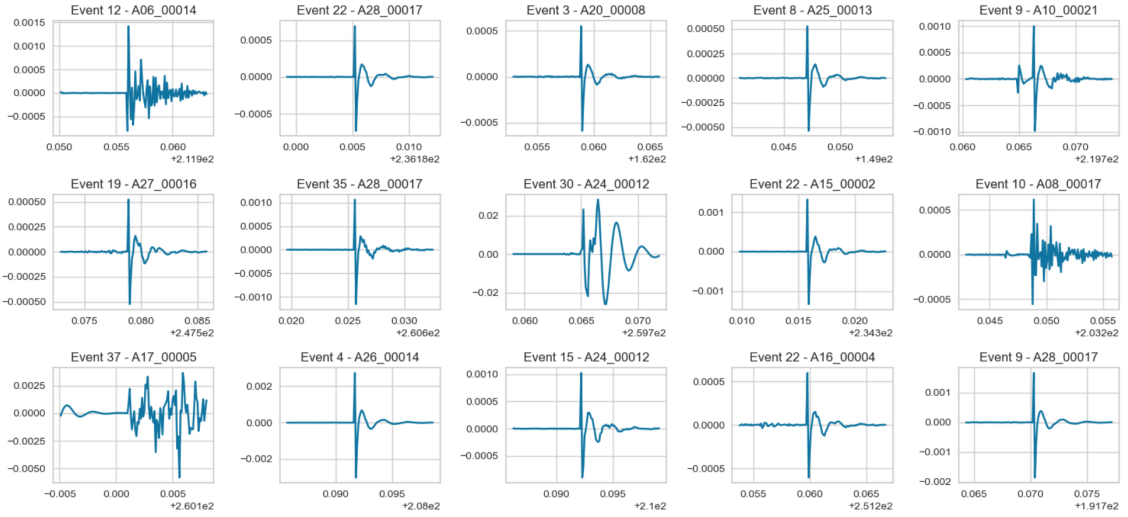


Figure 4.3: Examples of the identified events

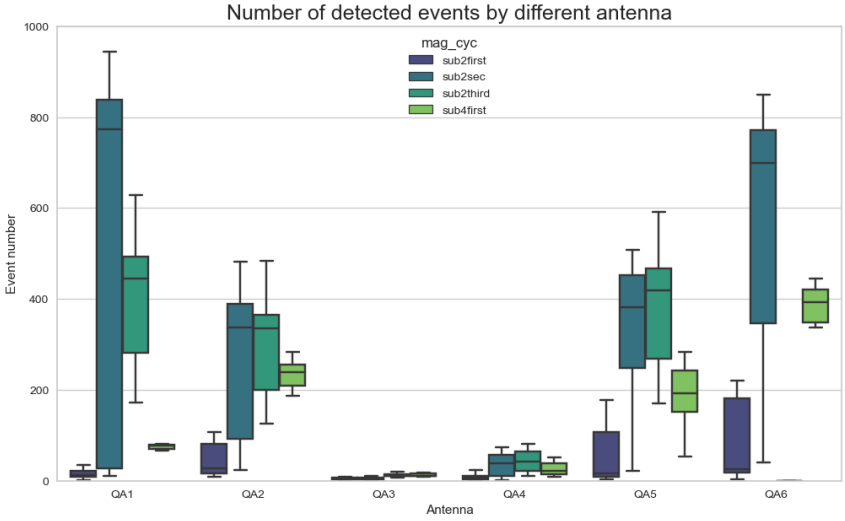


Figure 4.4: Number of events comprehensively detected by different antennas along the first, second and third thermal cycle for magnet `sub_2` and first thermal cycle for magnet `sub_4`

statistical analyses to investigate how events were distributed within the thermal cycle and within each individual current ramp. Furthermore, I found it interesting to compare how the distribution and numerosity of events differed among the various antennas for the same ramp and within the same thermal cycle. This comparative analysis gave insights into how events varied across different sensors and provided a more comprehensive view of the dataset.

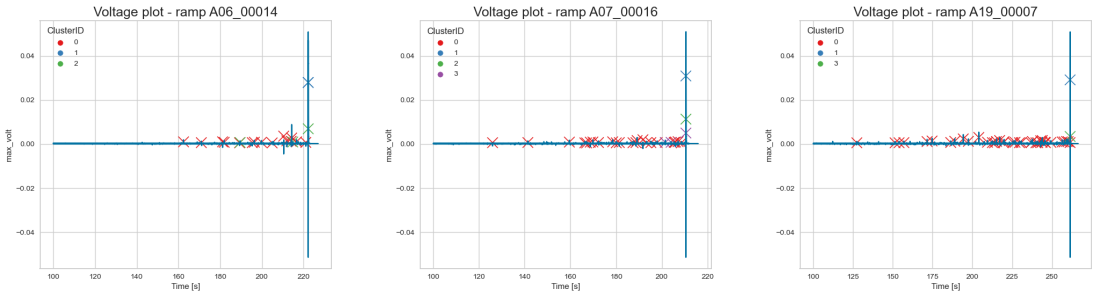
The graph in Figure 4.4 gives much information about the event distribution. One primary observation is the significant disparity in event detection between antennas QA3 and QA4 in comparison to the other antennas. This trend is consistently observed across various current ramps, thermal cycles, and magnets. The reasons behind this observation may be various. Initially, I thougth that these antennas might necessitate a lower detection threshold; however, experimenting with lower thresholds did not alter the outcome. Notably, this phenomenon is not tied to any specific magnet, as it is evident for both the sub_2 and sub_4 magnets. It is also possible that the position of the antennas around the magnet causes the magnetic field flux through their coil loops to be generally lower than that of the other antennas (resulting in a parallel magnetic field). Consequently, they are less sensitive to what occurs within the magnet itself. This interesting observation is currently an object of study in other research groups.

Figure 4.4 also highlights the absence of data for antenna QA6 in the context of the sub_4 magnet. This is because the signal from this antenna was degraded, and subsequently the event

extraction algorithm required too many computational resources. Consequently, we were unable to identify events for this antenna.

Furthermore, a compelling observation is that, overall, the virgin sub_4 exhibits more events than the virgin sub_2. This aspect is worth looking at more closely, and it's a good idea to investigate it further. It could be helpful to try different ways of defining events to see if the results are affected by the criteria we've chosen or if the same thing keeps happening.

Looking at results from the event detection one of the first things that can be noticed is the increasing number of events along different ramps but most importantly along different ramps of the same thermal cycle as I show with the three plots in Figure 4.5.



(a) A06_00014 (b) A07_00016 (c) A19_00007

Figure 4.5: Example of event distribution along the first thermal cycle for magnet sub_2

Since the events in each ramp are relatively few, a fact that would not allow us to make an effective statistical analysis, and exhibit substantial similarity among them (see Figure 4.6), in an effort to conduct a more comprehensive study and gain a deeper understanding of the magnet's behavior, I opted to examine all events originating from a single thermal cycle collectively. Specifically, I focused my attention on the initial thermal cycle of the sub_2 magnet.

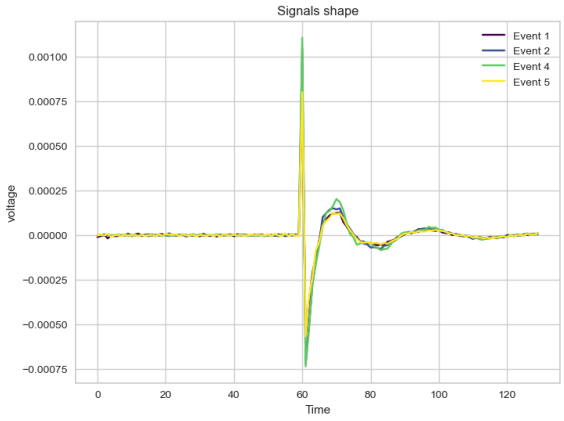


Figure 4.6: write a real caption and say what are the events I am considering

4.4 Feature analysis

Once the initial statistical considerations regarding the events were completed, I returned to studying their characteristics. In an effort to improve the analysis, unlike what was done with the first dataset, additional features were introduced, specifically the arc length and the number of peaks identified after applying the CWT analysis.

Once again, the initial analysis conducted on these features involved studying their correlations to gain insight into which ones were most significant. A result of this analysis is presented in Figure 4.7. The first observation that naturally arises is that the newly added features should not,

in any way, improve the results compared to the previous analysis because, being correlated with the existing features, they are not expected to provide additional information. Another interesting aspect is that as the volume of data to be analyzed increased, the intensity of (anti-)correlation measured among different features also intensified.

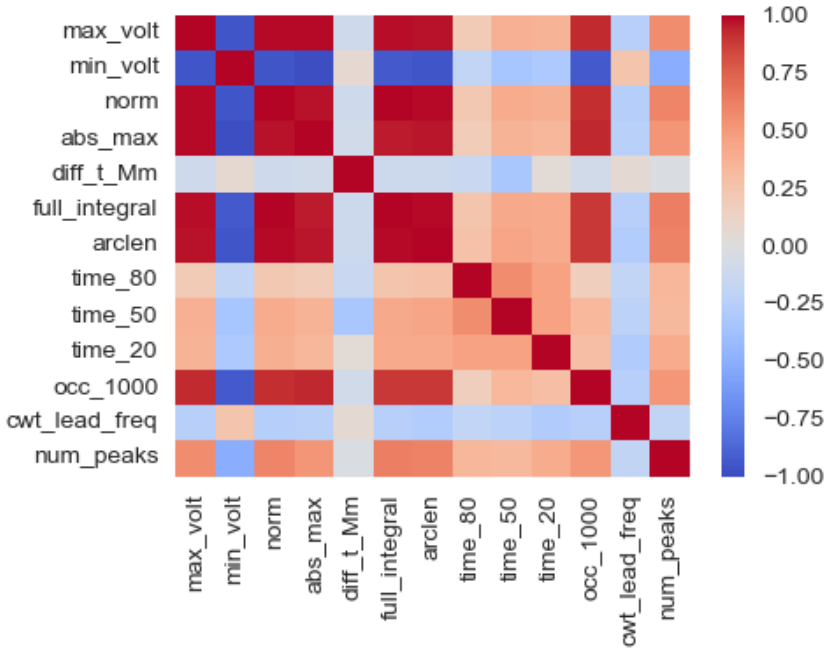


Figure 4.7: The image shows a heat map representing the correlation matrix of the different features considered

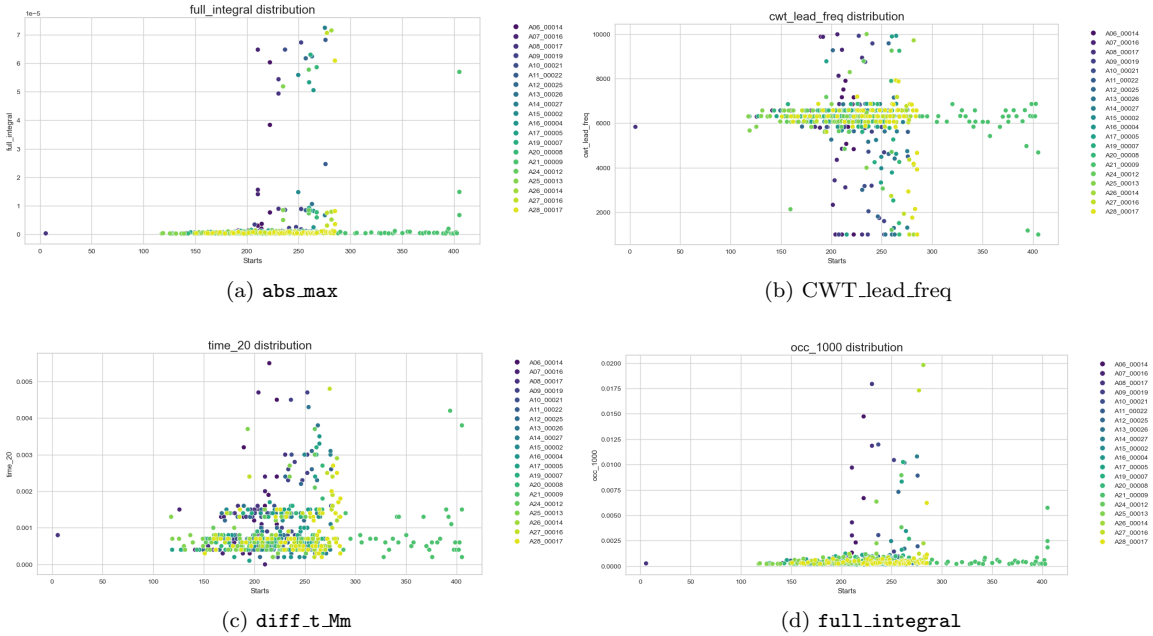
4.4.1 Statistical analysis on the events

As an initial qualitative analysis of the events, I created plots to visualize the evolution of various features over time to observe if there was a systematic variation in their values. In Figure 4.8, I present the plots for the features that were considered most relevant, while the remaining plots are included in Appendix 5.3.

The image above shows the temporal distribution of all features for the identified events during the first thermal cycle of magnet sub_2. The dataset pertains to antenna QA1. The x-axis represents the event's sampling index. Since events are stored as time windows in which the event may or may not be present, I have chosen to designate the start of the time window as the time reference for identifying the event. For this reason, the x-axis label reads "start." On the y-axis, the values associated with the feature under consideration in the plot are displayed.
The plots shown in Figure 4.8 do not display any systematic variation of the features over time.

4.5 Principal Components Analysis

One of the main results of the data analysis is the efficient application of Principal Component Analysis (PCA) on our data set.
PCA is a widely used technique for analyzing large datasets with a high number of features per observation. It helps in simplifying data while retaining as much information as possible and enables the visualization of multidimensional data. PCA reduces the dimensionality of a dataset by transforming it into a new coordinate system where most of the data's variance can be described with fewer dimensions than the original dataset [6].
In PCA, a collection of data points in a real coordinate space is represented by a sequence of unit vectors called principal components. Each principal component represents a direction in which the data best fits while being orthogonal to the previous components. These directions form an orthonormal basis in which different data dimensions are linearly uncorrelated.



(a) `abs_max` (b) CWT_lead_freq

(c) `diff_t_Mm` (d) `full_integral`

Figure 4.8: Scatterplot of the events feature over time. The colors reflect the cluster to which the event belongs.

In data analysis, the first principal component is a derived variable that explains the most variance in a set of variables presumed to be jointly normally distributed. Subsequent principal components explain the most remaining variance after accounting for the previous components. This process continues until all the variance is explained. PCA is particularly useful when many variables are highly correlated, and reducing their number to an independent set is desired.
Mathematically, principal components are eigenvectors of the data's covariance matrix, which can be computed through eigen-decomposition or singular value decomposition. PCA is a straightforward eigenvector-based multivariate analysis and is closely related to factor analysis, which incorporates domain-specific assumptions.

For the purpose of my analysis I applied PCA using `scikit-learn` decomposition library [11], in particular by setting a 90% variance retention threshold, I aimed to strike a balance between dimensionality reduction and information preservation. This ensured that the reduced dataset retained a substantial amount of the original data's variability, making it a suitable basis for further analysis and interpretation.

Component	EVR
Comp_1	0.669398
Comp_2	0.143993
Comp_3	0.084768
Comp_4	0.030355

Table 4.1: Explained Variance Ratio for the PCA

In table 4.1 I report the Explained Variance Ratio (EVR) for the PCA analysis. EVR gives us a measure of the portion of the total variance in the original dataset that is explained by each principal component; as expected, the first component alone explains more than 60% of the variance.

As one of the goals of the analysis is to gain an understanding of the physical phenomena happening in the magnets, is interesting to understand which features are more important and give a great contribution to each principal component for this reason, I made the plot 4.9.

One of the most significant observations that can be made regarding the content depicted in

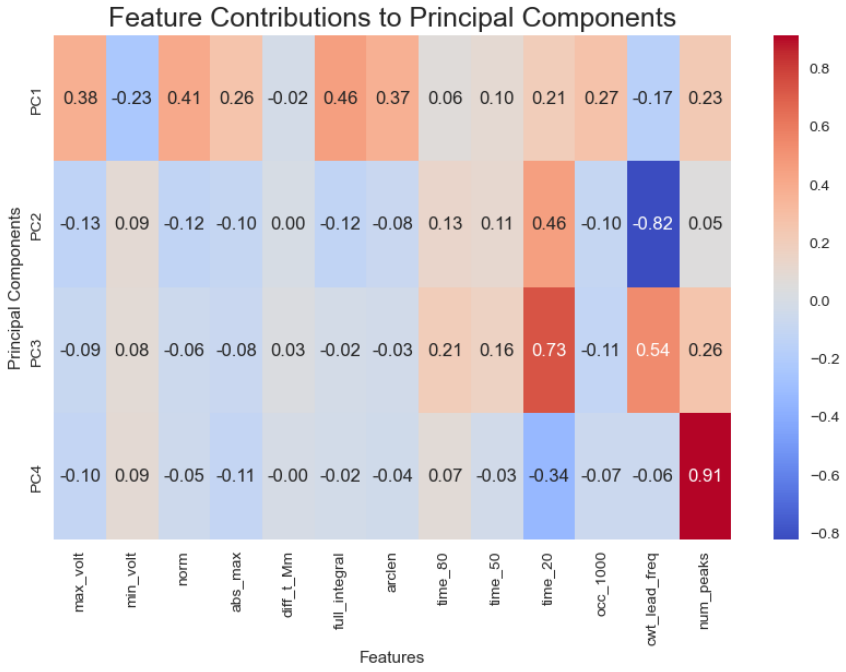


Figure 4.9: The image shows a heat map representing the contribution of each analyzed feature to each of the principal components used in the analysis

this plot pertains to the fact that the different features contributing to each component are, in fact, correlated with each other. Nevertheless, they make comparable contributions to the overall variance, as exemplified in the case of the first component. This implies that the outcome achieved through clustering based on these components will be superior to that obtained by considering uncorrelated components alone, as it will be grounded in a richer information substrate.

4.6 Clustering results

Clustering was performed using the first 4 components obtained from PCA. In this case, as well, I used the K-Means clustering approach described in Section 2.1. In particular, the various evaluation techniques provided the following results:

- Silhouette Score: 4 clusters with a silhouette score of 0.748 (see Fig. 4.10)
- Number of negative silhouette score: 4 clusters (5 points)
- K-Elbow: 4 clusters (the plot of the K-Elobow cluster obtained is shown in Figure 2.1)
- WCSS cross-validation: 4 clusters with WCSS score of 18.16

Therefore, the optimal result is the one that involves dividing the data points into 4 clusters.

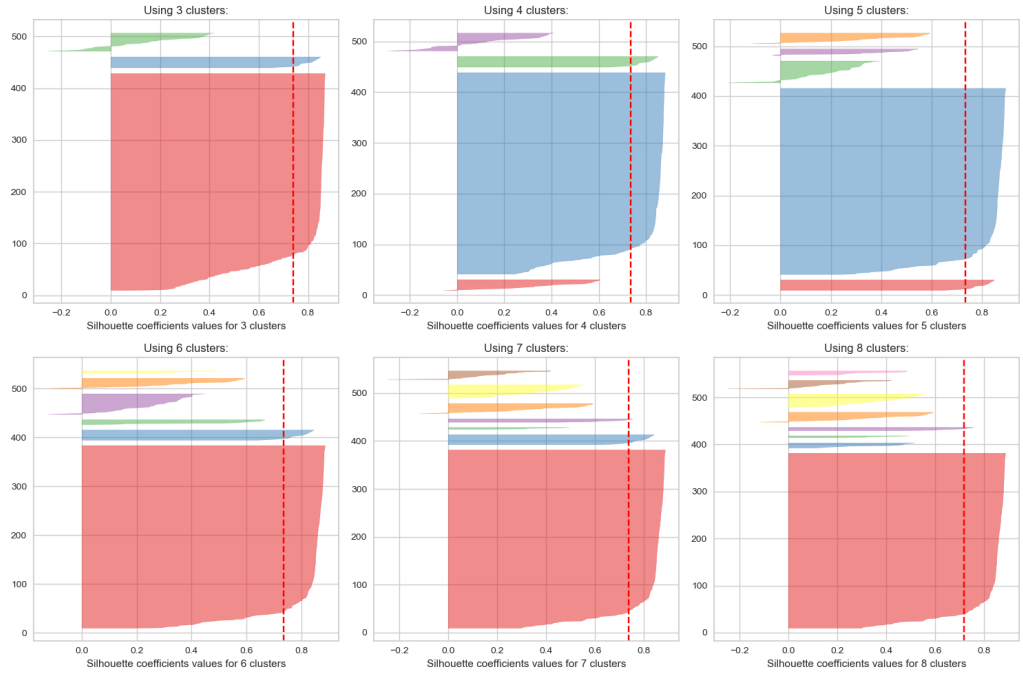


Figure 4.10: Silhouette plot for different numbers of clusters

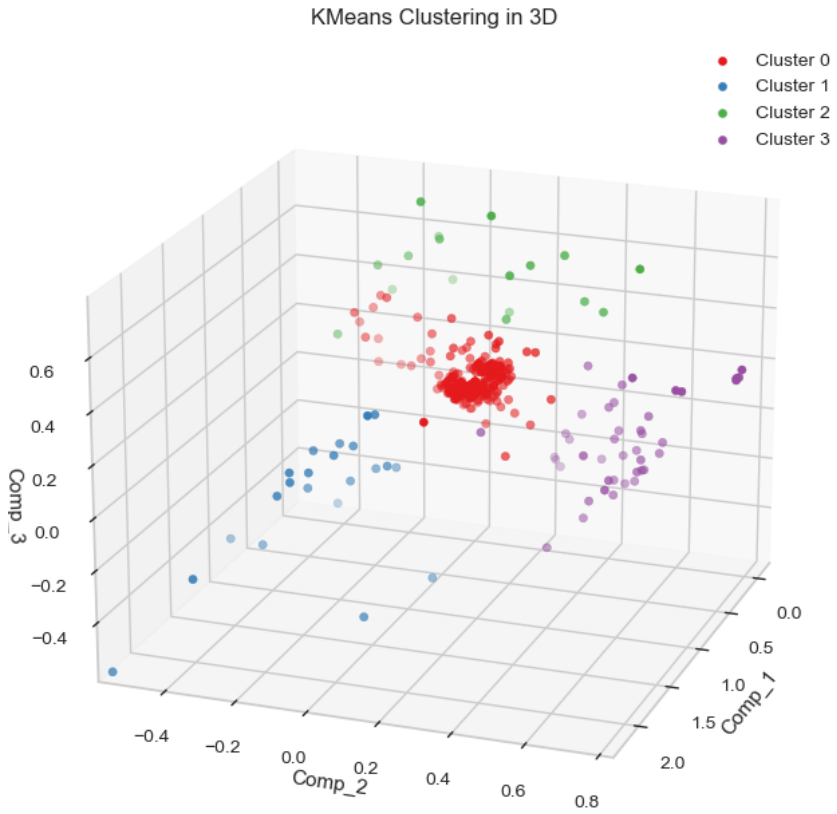
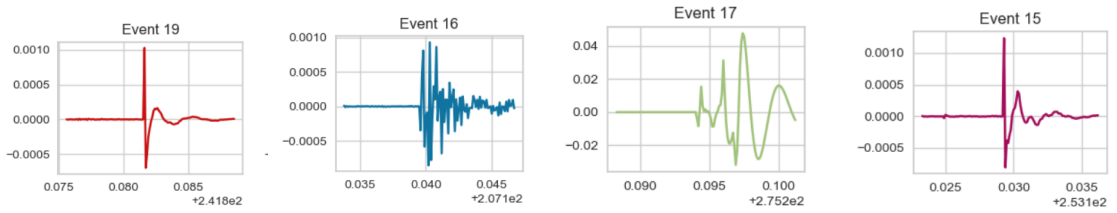


Figure 4.11: Representation of the data-points distribution in the 3-principal-components space

In picture 4.11 I show the results of the K-means clustering analysis using a number $K = 4$ of clusters.

The largest cluster is undoubtedly the central red one, characterized by events like those shown in Figure 4.12 - `(a)`. The specific shape of this signal and the frequency with which it is detected

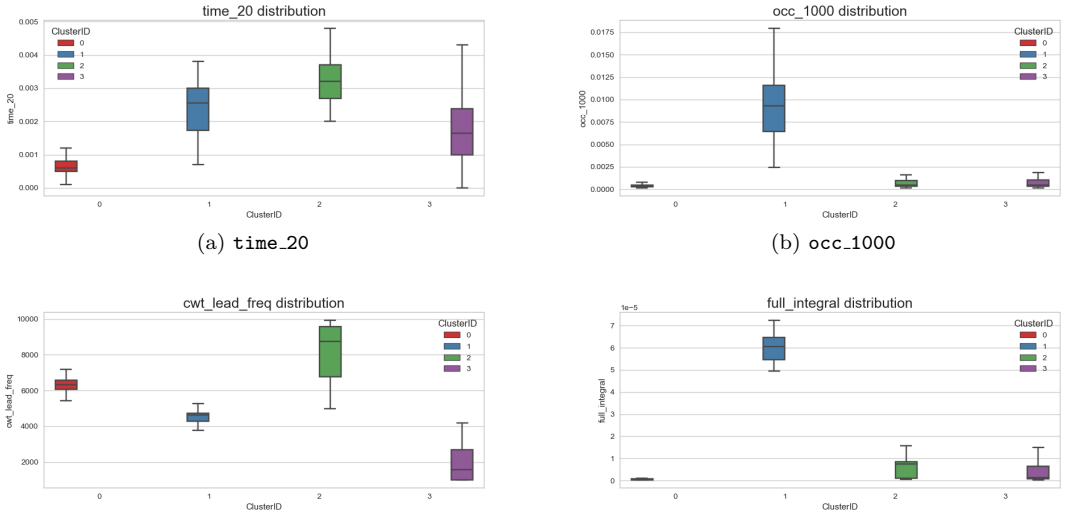
suggest that it may be related to redistribution currents, one of the possible candidates among electromagnetic phenomena that can cause quenching. Another possible source of this signal is a mechanical phenomenon related to the superconductor cables, which repeat within the magnet.
In addition to this, three other clusters are identified. Cluster 1, represented in blue in Figure 4.11, is predominantly composed of events similar to the one shown in Figure 4.12 - `(b)`. The type of signal seems more like an acoustic signal, characteristic of a mechanical event, such as cable friction or an epoxy crack inside the magnet.
Cluster 2, represented in green in Figure 4.11, is characterized by events mostly similar to those represented in Figure `(c)`. These events, mostly recognized and grouped by the algorithm, are not so much physical signals as they are signals resulting from the application of a filter at the edges of the signal.
Cluster 3, on the other hand, consists of events that have a signal shape similar to events in Cluster 0, as I show in Figure 4.12 - `(d)` but are characterized by different values of the PCA components.



(a) Example from cluster 0 (b) Example from cluster 1 (c) Example from cluster 2 (d) Example from cluster 3

Figure 4.12: Example of events that best represents the event in each cluster

Once the clustering was completed, I returned to the space of physical features associated with individual events to observe if there were specific values of these features that could be associated with each cluster. In Figure 4.13, I present the boxplots related to the characteristics that have proven to be the most significant and well-distinguished in the analysis of the different clusters.



(a) `time_20` (b) `occ_1000`

(c) `cwt_lead` (d) `full_int`

Figure 4.13: Boxplot representing the value of the main feature in each cluster

In particular, in Figure 4.13 - `(a)`, it can be observed that the time instant at which the voltage reaches 20% of the peak intensity is very similar for clusters 0 and 3. These clusters were characterized by a very similar signal shape. Furthermore, in Figures `(b)` and `(d)`, it is evident that cluster 1 takes on values for some features that deviate significantly from those of the other clusters, as reflected in the scatterplot in Figure 4.11.
Additionally, in accordance with what was observed in Figure 4.11, the features of events in cluster

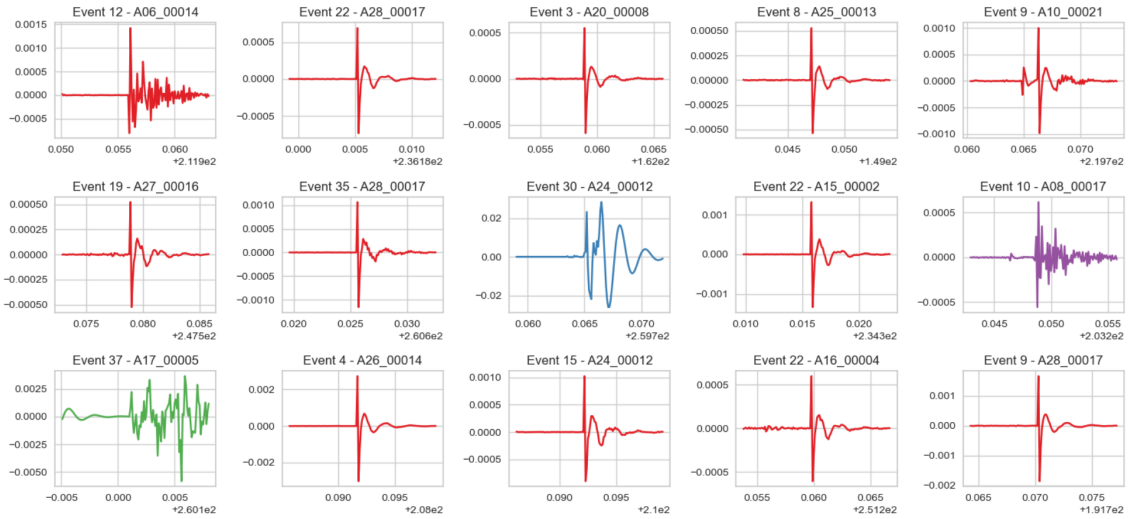


Figure 4.14: Examples of the identified events

0 tend to assume very similar values, making the cluster very compact.

These results suggest that the data analysis with clustering has been effective, although there are still areas for improvement. One of the aspects that certainly needs to be reevaluated is the application method of the CWT analysis. It has been noted that the values reported for the lead frequency of the CWT analysis are not compatible with the signal trend shown in Figure 4.12 - (a).

Chapter 5

Conclusions

With this work, we have been able to develop a tool that provides consistent event labeling and allows for the detection of similarities between events. Though this is a notable accomplishment, the significance of these findings is more moderate: we observed a significant increase in events in sub_2 following the first thermal cycle, a phenomenon that requires further investigation to better understand its implications. This increased event occurrence may have implications for the magnet's performance, although more research is needed to draw concrete conclusions.

Our analysis also revealed the presence of remarkably similar events within the dataset, which raises questions about the underlying disturbances affecting the system, and further study is required to gain a deeper understanding of their significance.

The observation of a lower number of events in QA3 and QA4, is likely due to antenna orientation with respect to the magnetic field direction, in agreement with what was observed and studied by other researchers.

Additionally, we noticed that the sub_4 virgin ramp has shown a higher number of events compared to the baseline magnet, possibly related to the mechanics of impregnation materials. This observation provides insights into the behavior of these materials under different conditions, but the precise implications for magnet design and long-term performance remain uncertain.

Some possible steps forward are the exploration of the variation in clusters across the magnet data and efforts to pinpoint the sources of these signals, to do this would be useful and interesting to incorporate acoustic and quench antenna data to refine our results and improve the model's efficiency. Comparative studies between sub2 and sub4 will provide additional context for the observed events. Regarding the algorithm would certainly be interesting to test different clustering techniques and develop a proper hyper-parameter tuning routine, this could the model's flexibility. This comprehensive approach will help us to better understand the system and contribute to its reliability and efficiency in a more detailed manner.

Bibliography

Bibliography

[1] What is a continuous wavelet transform? https://www.collimator.ai/reference-guides/what-is-a-continuous-wavelet-transform. Accessed: 2023-10-12.

[2] D. Arbelaez, T. Bogdanof, L. Brouwer, S. Caspi, D. Dietderich, J. L. Rudeiros Fernández, P. Ferracin, S. Gourlay, R. Hafalia, M. Krutulis, M. Marchevsky, M. Maruszewski, C. Myers, S. Prestemon, M. Reynolds, T. Shen, J. Swanson, R. Teyber, M. Turqueti, G. Vallone, and X. Wang. Status of the nb_3sn canted-cosine-theta dipole magnet program at lawrence berkeley national laboratory. *IEEE Transactions on Applied Superconductivity*, 32(6):1–7, 2022.

[3] Daniel Brandt, editor. *Proceedings, 2009 CAS-CERN Accelerator School: Specialised course on Magnets: Bruges, Belgium, June 16 - 25, 2009*, CERN Yellow Reports: School Proceedings, Geneva, 2010. CERN.

[4] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[5] Trevor Hastie, Robert Tibshirani, and J H Friedman. *The elements of statistical learning*. Springer series in statistics. Springer, New York, NY, 2 edition, dec 2009.

[6] Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philos. Trans. A Math. Phys. Eng. Sci.*, 374(2065):20150202, April 2016.

[7] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data*. Probability & Mathematical Statistics S. John Wiley & Sons, Nashville, TN, 99 edition, April 1990.

[8] Mark E. Ladd, Harald H. Quick, Oliver Speck, Michael Bock, Arnd Doerfler, Michael Forsting, Jürgen Hennig, Bernd Ittermann, Harald E. Möller, Armin M. Nagel, Thoralf Niendorf, Stefan Remy, Tobias Schaeffter, Klaus Scheffler, Heinz-Peter Schlemmer, Sebastian Schmitter, Laura Schreiber, N. Jon Shah, Tony Stöcker, Michael Uder, Arno Villringer, Nikolaus Weiskopf, Moritz Zaiss, and Maxim Zaitsev. Germany's journey toward 14 tesla human magnetic resonance. *Magnetic Resonance Materials in Physics, Biology and Medicine*, 36(2):191–210, April 2023.

[9] David J C MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, Cambridge, England, sep 2003.

[10] The pandas development team. pandas-dev/pandas: Pandas. -, feb 2020.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[12] Alberto Plebani. Machine learning approach to classifying quench antenna signals. *FNAL final report*, 2022.

[13] Soren Prestemon, Kathleen Amm, Lance Cooley, Steve Gourlay, David Larbalestier, George Velev, and Alexander Zlobin. The 2020 updated roadmaps for the us magnet development program. -, 2020.

[14] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[15] Martin N Wilson. Superconducting magnets. -, 1983.

[16] Alexander V. Zlobin and Daniel Schoerling. *Superconducting Magnets for Accelerators*, pages 3–22. Springer International Publishing, Cham, 2019.

Appendix

5.1 Appendix A - code documentation and organization

I write this section with the purpose of making it easier for other people who want to used the code developed by Alberto and me to understand the purpose and the usage of the different functions. Here I will make a list of the main functions used in the data analysis explaining their arguments and their output.

Class definition `class Event_Reduced():`
`def __init__(self, name, time, data)`

Event selection

1. `IdentifyEvent_Reduced(time_vec, volt_vec, antenna, dtype, threshold)`:

   - `time_vec`: `numpy` vector containing the time information of the signal
   - `volt_vec`: `numpy` vector containing the voltage information of the signal
   - `antenna`: string for the antenna name (eg. `'QA1'`
   - `dtype`: string saying if the data are filtered or unfiltered (after code modification useful only for standard saving routine)
   - `threshold`: threshold selected for the event identification

   The function identifies and saves all the events in a given signal. Events are defined as explained in the previous part of this work

Feature extraction

1. `get_subfolder_names(folder_path)`: takes as argument the path to the folder containing subfolders. Is important to extract features from more current ramps with a single call to the function.
2. `ReadEvents_Reduced(path)`: takes as argument the path to the event I want to get the information of
3. `def plotCWT(event, cwt_lead, freq, save, path)`:

   - `event`: `Event` - or `Event_Reduced` object
   - `cwt_lead`: lead frequency of the CWT analysis
   - `freq`: sampling frequency
   - `save`: boolean, 1 for saving 0 for not saving the plot
   - `path`: string, path to save location

   The function returns the plot of the CWT analysis

4. `SignalFeatures(event, perc)`:

   - `event`: `Event` - or `Event_Reduced` object
   - `perc`: float, percentage of the peak we are interested in (eg. 0.8 to get `time_80`)

The function returns the signal feature asked as input

5. `FreqFeatures(event, plot)`:
   - `event`: `Event` - or `Event_Reduced` object
   - `path`: boolean, 1 to get 0 not to get the CWT plot. In this case the plot is not saved.

   The function returns the following features: `occ_1000, cwt_lead_freq, num_peaks`
6. `VoltageFeatures(event)`: `Event` - or `Event_Reduced` object
   The function returns the following features: `max_volt, min_volt, norm, abs_max, diff_t_Mm, full_integral`
7. `ExtractFeatures(event)`: `Event` - or `Event_Reduced` object
   The function calls all the previously defined functions and returns a list with the following features `max_volt, min_volt, norm, abs_max, diff_t_Mm, time_80, time_50, time_20, occ_1000, cwt_lead_freq, num_peaks, full_integral`
8. `SaveFeatures(events, path, n)`:
   - `events`: list of `Event` or `Event_Reduced`
   - `path`: path to save location
   - `n`: a string representing the number of features we are extracting, (eg. `'all_` for all features), this is important only for saving reasons.

   The function saves the feature in an `n_features.txt` file.
9. `BuildDataframe(path)`: takes as argument the path to the saving location to read the feature. This function is written to read the `all_features.txt` file and can be modified if needed.

K-Means clustering and clustering evaluation

1. `ScaleData(df_model)`: The function takes as input a pandas data frame containing all the features useful to clustering. Returns a rescaled version of the data frame.
2. `show_silhouette_scores(clusters, k, df_model, save)`:
   - `clusters`: cluster object obtained as output from `km.fit_predict()` function of the `scikit-learn`
   - `k`: number of clusters we want
   - `df_model`: pandas data frame with data on which we have to perform clustering
   - `save`: boolean, 1 for saving 0 for not saving the plot
3. `show_single_silhouette(model, k, df_model)`:
   - `model`: output of `scikit-learn` function `KMeans`
   - `k`: number of clusters we want
   - `df_model`: pandas data frame with data on which we have to perform clustering
4. `SilohuetteTest(kmin, kmax, df_model)`:
   - `kmin`: minimum value of k we want to test
   - `kmax`: maximum value of k we want to test
   - `df_model`: pandas data frame with data on which we have to perform clustering
5. `KFoldTest(ks, n_splits, df_model, path, save)`:
   - `ks`: values of k we want to test
   - `n_splits`: number of folds we want to use
   - `df_model`: pandas data frame with data on which we have to perform clustering

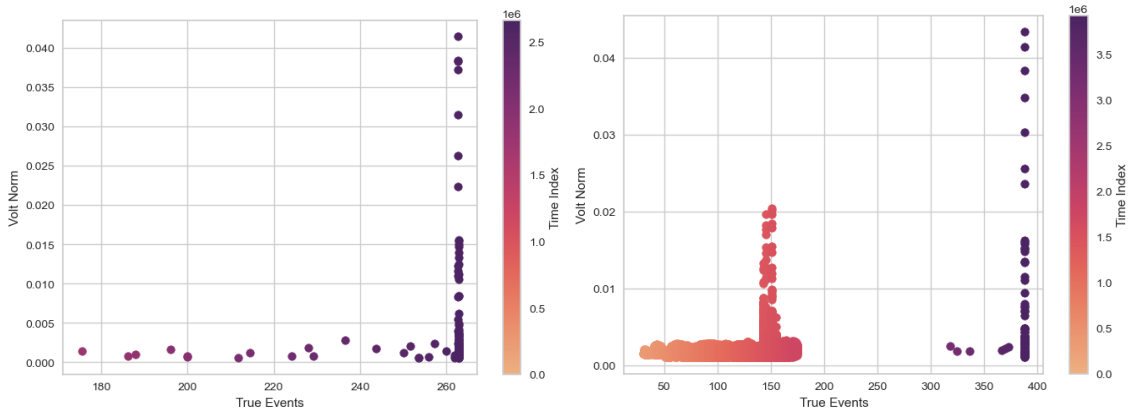
- `path`: string, path to save location
- `save`: boolean, 1 for saving 0 for not saving the plot

6. `show_parallel_coordinates(clusters, df_model, bestK, save)`:

- `clusters`: cluster object obtained as output from `km.fit_predict()` function of the `scikit-learn`
- `df_model`: pandas data frame with data on which we have to perform clustering
- `bestK`: number of clusters
- `save`: boolean, 1 for saving 0 for not saving the plot

All functions exist not only in the `Reduced` version but also in a "standard" version. This "standard" version is intended to be identical to the "reduced" one but is designed to be used with objects of type `Event` rather than `Event_Reduced`. The reason for developing two sets of functions in parallel is that the test data and the data at the core of the analysis were saved with different formats and information.

5.2 Appendix B - examples of nonphysical data



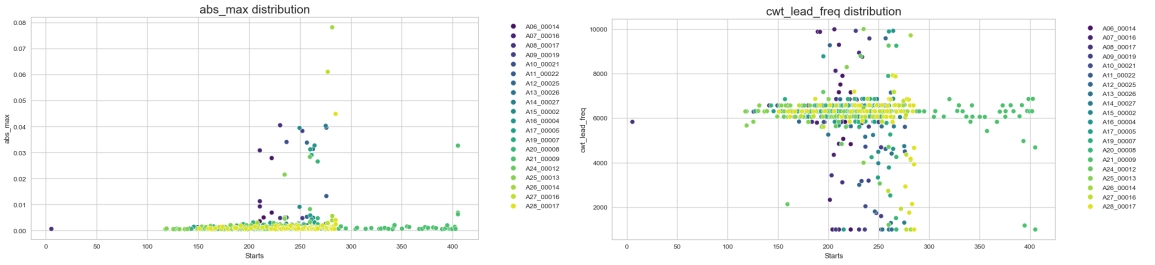
(a) A11_00022 - QA1 (b) A23_00011 - QA1

Figure 5.1: EVents detected by the algorithm on two different signals

The two images show how the algorithm detects events on both clean and noisy - probably non-physical - signals.

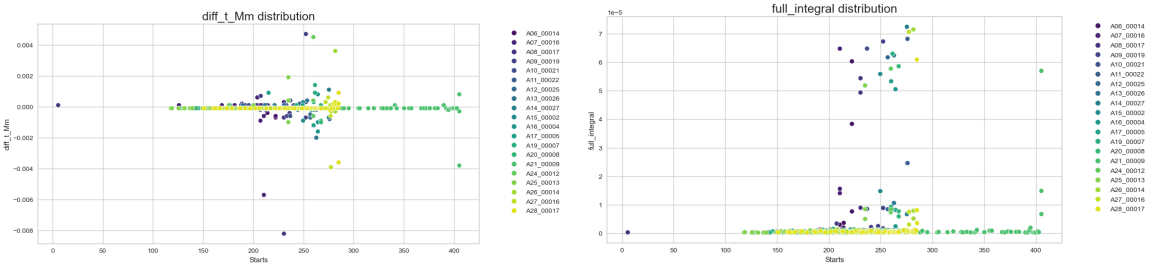
5.3 Appendix C - Features distribution over time

Here, I present the temporal distribution of all features for the identified events during the first thermal cycle of magnet sub_2. The dataset pertains to antenna QA1. The x-axis represents the event's sampling index. Since events are stored as time windows in which the event may or may not be present, I have chosen to designate the start of the time window as the time reference for identifying the event. For this reason, the x-axis label reads "start." On the y-axis, the values associated with the feature under consideration in the plot are displayed.



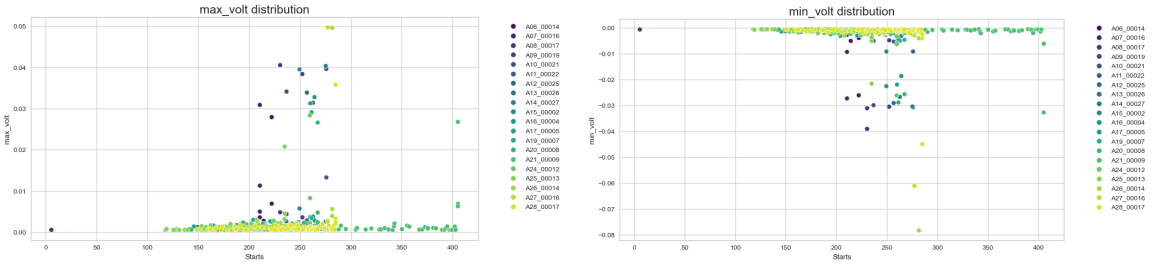
(a) `abs_max`

(b) cwt_lead_freq



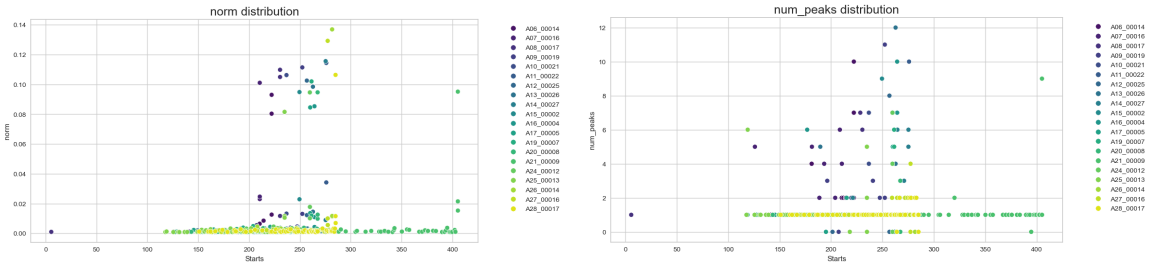
(c) `diff_t_Mm`

(d) `full_integral`



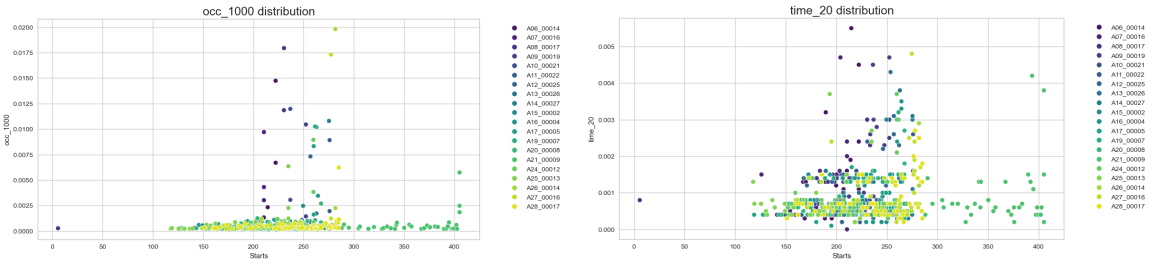
(e) `max_volt`

(f) `min_volt`



(g) `volt_norm`

(h) `num_peaks`

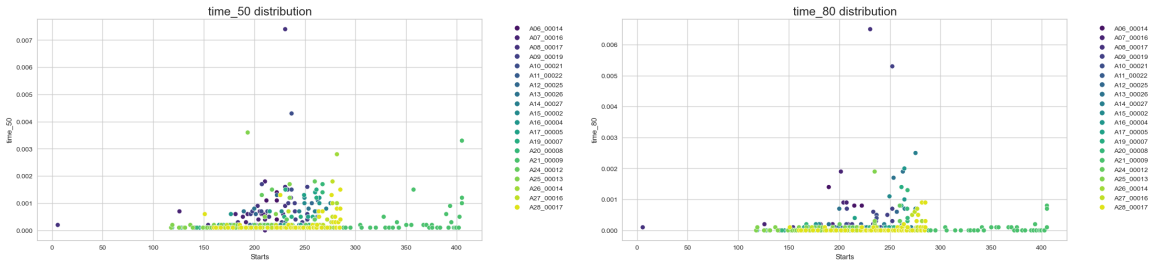


(i) `1000 frequency occurance`

(j) `time_20`

5.4 Appendix D - Distribution of cluster's feature

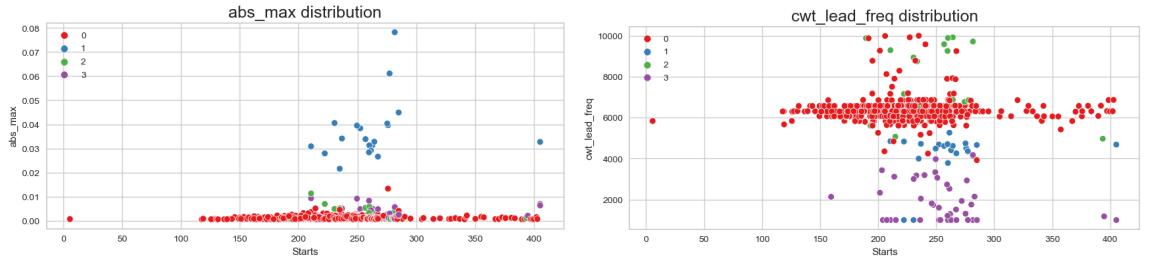
Here, I present the temporal distribution of all features for the identified events during the first thermal cycle of magnet sub_2. The dataset pertains to antenna QA1. The x-axis represents the event's sampling index. Since events are stored as time windows in which the event may or may



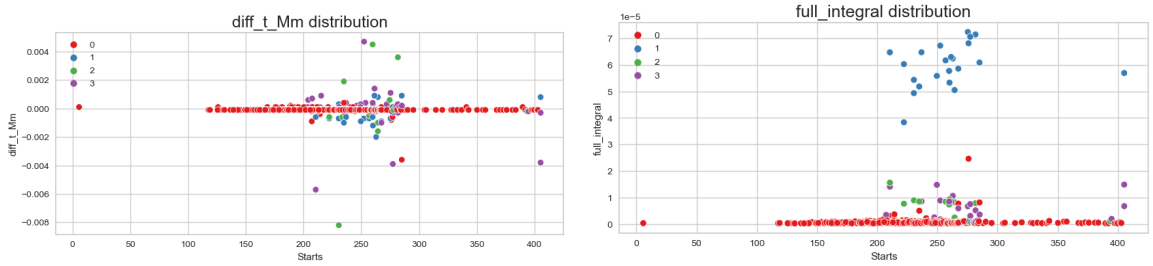
(a) `time_50` (b) `time_80`

Figure 5.2: Feature distribution over time

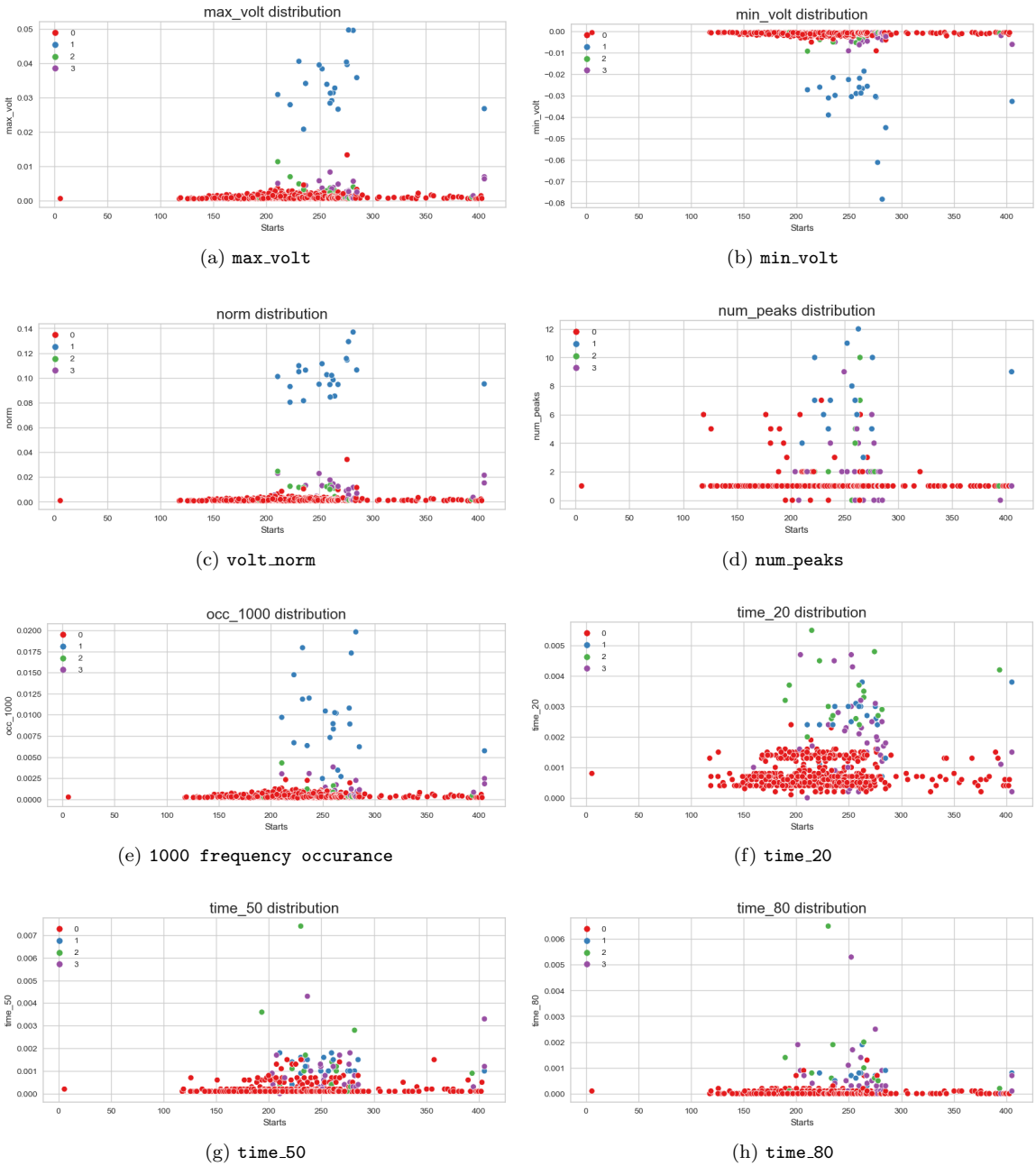
not be present, I have chosen to designate the start of the time window as the time reference for identifying the event. For this reason, the x-axis label reads "start." On the y-axis, the values associated with the feature under consideration in the plot are displayed. The colors represent the assigned cluster to which each data point belongs.



(a) `abs_max` (b) CWT_lead_freq



(c) `diff_t_Mm` (d) `full_integral`



(a) `max_volt` (b) `min_volt`

(c) `volt_norm` (d) `num_peaks`

(e) `1000 frequency occurance` (f) `time_20`

(g) `time_50` (h) `time_80`

Figure 5.3: Feature distribution over time