Christoph Obermair, BSc.

Extension of Signal Monitoring Applications with Machine Learning

Master's Thesis

submitted to

Graz University of Technology

Supervisors

Assoc.Prof. Dipl.-Ing. Dr.mont. Franz Pernkopf
Dr. Michał Maciejewski

Signal Processing and Speech Communication Laboratory

in cooperation with

CERN
Geneva, Switzerland

Graz, February, 2020

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

date

(signature)

Acknowledgements

Finishing this thesis would not have been possible without the help of many people. In particular, I would like to thank my CERN supervisor Dr. Michał Maciejewski for giving me so much incredible support. Not only did he constantly give me professional advise in his role as my supervisor and mentor, but we also spent a great time together outside of CERN by organizing BBQs, climbing mountains and discussing the true meaning of life. I also want to thank my supervisor of the Technical University of Graz Dr. Franz Pernkopf. With his knowledge, he supported me very well, even though I did most of my work in Geneva and not in Graz. I also want to thank the TE-MPE-PE section leader Dr. Arjan Verweij, for giving me the opportunity to give proof of my skills. Special thanks to Dr. Zinour Charifoulline who provided me a great insight into his previous work at CERN and supported me a lot while implementing it. Furthermore, I want to thank all the people from my office, the SPSC institute and the members of the LHC monitoring project, who helped me out. Last but not least, I want to thank my family and my friends from Austria for their mental support and for keeping me up to date with all the local news during my absence.

Abstract (English)

The Large Hardron Colider (LHC) is the world's largest particle accelerator. It is 27-km long and contains a wide range of superconducting circuits for controlling the shape and trajectory of particles. During operation, the nominal designed current (for 7 TeV) in the main bending dipole circuit is 11 850 A, which is equivalent to the current of about 120 single-family households. In order to prevent failures during operation, there are several protection systems installed. Furthermore, each of the magnets is checked during the Hardware Commissioning (HWC) powering test, which take place prior to each operation following an extended technical stop. Especially, because of the high complexity of the LHC and the requirement of high reliability during operation, those safety measures have a huge responsibility. Many protection systems have taken care of this responsibility in the past, which led to several years of successful operation. The data gathered during these years, allows the characterisation of the protection systems and the usage of the obtained values as reference for the monitoring during operation.

The "LHC Signal Monitoring Project" has been founded to unite existing analysis tools. This thesis shows how the logged signals of the different databases can be used in order to implement new and extend existing monitoring applications into the development environment of the project. Several LHC component features are calculated and their significance is discussed. Since the LHC consists of several copies of similar circuits, the distribution of those parameters is studied and compared over both time and circuit. In particular, the implementation of two existing LHC analysis modules from the past are presented in this thesis. The busbar resistance analysis and the quench heater analysis. For both methods, this thesis provides a generic analysis which can be applied to any signal of the LHC systems. It covers the data analysis steps of acquisition, exploration, modelling, and monitoring. During modelling a unique approach is introduced, which uses supervised machine learning to extend existing signal monitoring applications.

Abstract (German)

Mit einer Länge von 27 km ist der Large Hardron Collider (LHC) der größte Teilchenbeschleuniger der Welt. Eine Vielzahl an supraleitenden Stromkreisen sorgt dafür, dass die Laufbahn der hochenergetischen Teilchen eingehalten wird. Dabei führt beispielsweise der Stromkreis mit den supraleitenden Dipolmagneten einen Strom von bis zu 11 850 A, was einem Äquivalent von rund 120 durchschnittlichen Einfamilienhäusern entspricht. Um jegliche Störungen zu vermeiden, sind eine Vielzahl an Sicherungssysteme installiert. Des Weiteren werden die Bauteile im Rahmen der Hardware Inbetriebnahme Tests in regelmäßig Abständen von Experten gewartet und geprüft. Besonders wegen der Komplexität und der großen Anforderung an die Zuverlässigkeit der Maschine, haben diese Sicherungssysteme eine enorme Verantwortung zu tragen. Die gewissenhafte Arbeit zahlreicher Experten und die Zuverlässigkeit der Sicherheitssysteme haben diese Verantwortung in der Vergangenheit erfolgreich gemeistert, was den erfolgreichen Betrieb der letzten Jahre sichergestellt hat. Die dabei gesammelten Daten erlauben die Charakterisierung dieser Sicherungssysteme und die Verwendung der Maschinenparameter als Referenz für die Überwachung des Betriebsverhaltens.

Das "LHC Signal Monitoring Project" wurde gegründet um existierende Sicherheitssysteme zu vereinen. Diese Arbeit wird zeigen, wie die gespeicherten Signale der verschiedenen Datenbanken genutzt werden können um existierende Sicherheitssysteme in das "LHC Signal Monitoring Project" zu implementieren und zu erweitern. Dazu werden eine Vielzahl an Betriebsparameter verschiedener Maschinenkomponenten berechnet und diskutiert. Der LHC besteht aus mehreren Kopien mit gleichen Stromkreisen, weshalb die Verteilung dieser Betriebsparameter in Abhängigkeit der Zeit und der Stromkreise verglichen wird. Insbesondere werden zwei existierende LHC Analyse Module in dieser Arbeit vorgestellt: die Busbar-Wiederstands Analyse und die Quench-Heizer Analyse. Zur einheitlichen Implementierung dieser Module wurde ein allgemeines Konzept entwickelt, zu dem weitere Analyse Module einfach hinzugefügt werden können. Dieses Konzept umfasst die Analyseschritte: Data-Acquisition, Data-Exploration, System-Modellierung und Condition-Monitoring. Im Rahmen der System-Modellierung wird ein neues Konzept vorgestellt, in dem Supervised Machine Learning genutzt wird um existierende Signal Monitoring Anwendungen zu erweitern.

Contents

1 Introduction

The name Large Hardron Colider (LHC) precisely describes the properties of the most powerful particle accelerator ever built. It is a *Large* 27 km long ring of superconducting magnets, accelerating cavities, beam instruments, detectors, etc. The purpose of this complex system is to accelerate protons or heavy ions, which belong to the group of *Hadrons* (heavy particles). Those particles are accelerated to nearly the speed of light and travel in the opposite directions inside two beam pipes enclosed in superconducting magnets. Once the particles reach the desired energy, they are made to *Collide* at four points of the LHC. With huge detectors (see Figure 1.1), those collisions are analyzed and the origin of our Universe can be explored [1].

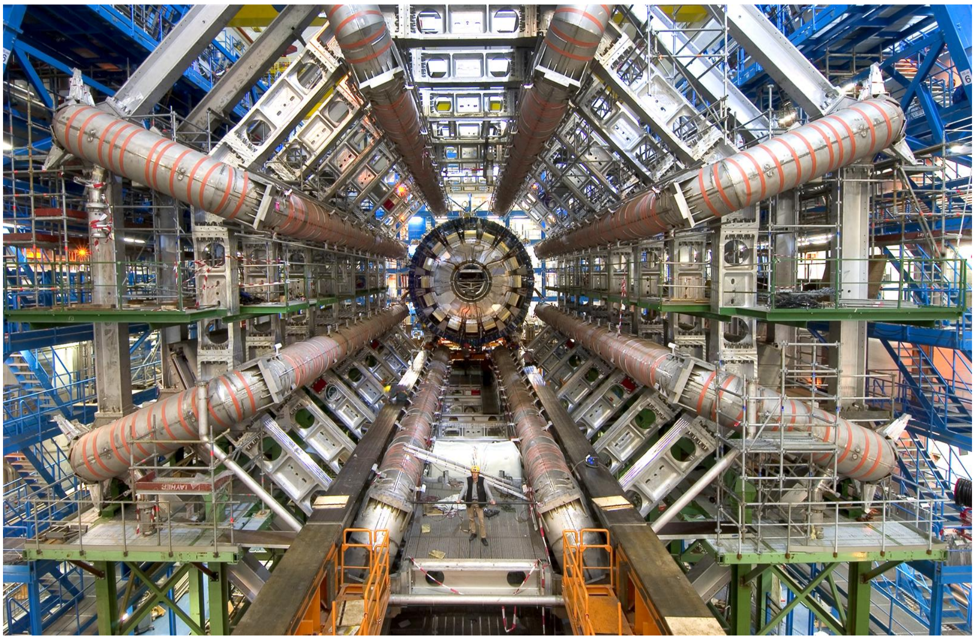


Figure 1.1: ATLAS, one of the four detectors at the collision points [2].

Before the beams are accelerated in the LHC they are pre-accelerated in a sequence of several other smaller particle accelerators (see Figure 1.2). Once a desired energy threshold is reached, the beam is injected into the next accelerator of the chain. After injecting the beam into the last element of the chain, the LHC, the beams have such high energy that it is possible to keep them in the given trajectory only by using superconducting dipole magnets. Beyond the dipole magnets there are also quadrupole magnets which focus/defocus the beam, and corrector magnets which compensate imperfections in the magnetic field [3].

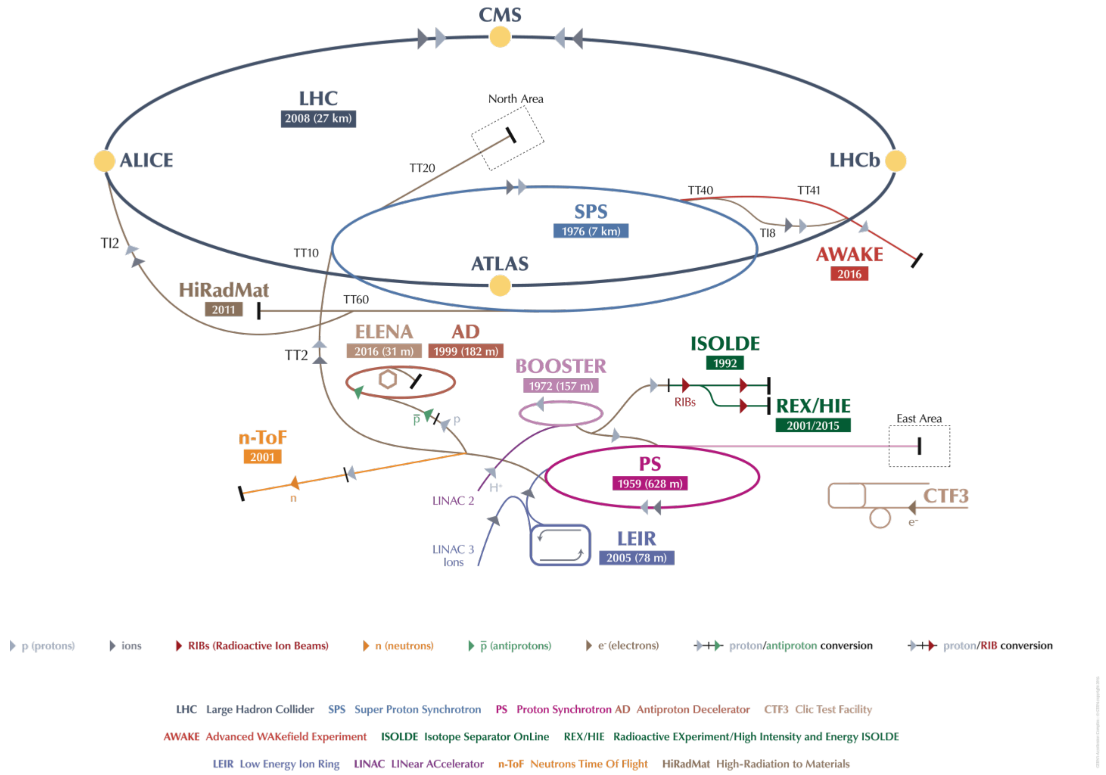


Figure 1.2: Conseil Européen pour la Recherche Nucléaire (CERN) accelerator complex map [1].

1.1 The LHC Main Dipole Magnet

The cross-section of the LHC main dipole magnets is shown in Figure 1.3. The magnet consists of two apertures powered in series with each aperture containing two superconducting coils (i.e., an upper and a lower pole), made of Niobium-Titanium (Nb-Ti) cables. This material becomes superconducting, which means that the resistance of the cable becomes zero, below a certain temperature, magnetic field, and current density. In order to additionally ensure a save operation and to profit from the properties of the superfluid helium, which is used to cool down the magnet, the operating temperature is around 1.9 K (–271.3 °C). It is therefore possible to run the LHC main dipole magnets with their nominal magnetic field of 8.3 T and their nominal current of 11.85 kA, which is required to bend the 7 TeV proton beams of the LHC [4].

However, if ether the temperature, the magnetic field, or the current in the magnet exceeds the critical value, a so-called quench occurs. This means that a part of the superconducting coil turns into a normal conducting coil with a non-zero resistance at this specific position. Since the rest of the magnet still operates without losses, all the energy stored in the magnet (7.1 MJ) is dissipated at this particular point and could melt up to 14 kg of cable if the protection systems are not functioning. Additionally, the high Lorenz forces, the excessive voltages, and huge temperature gradients can further destroy the magnet [4], [6].

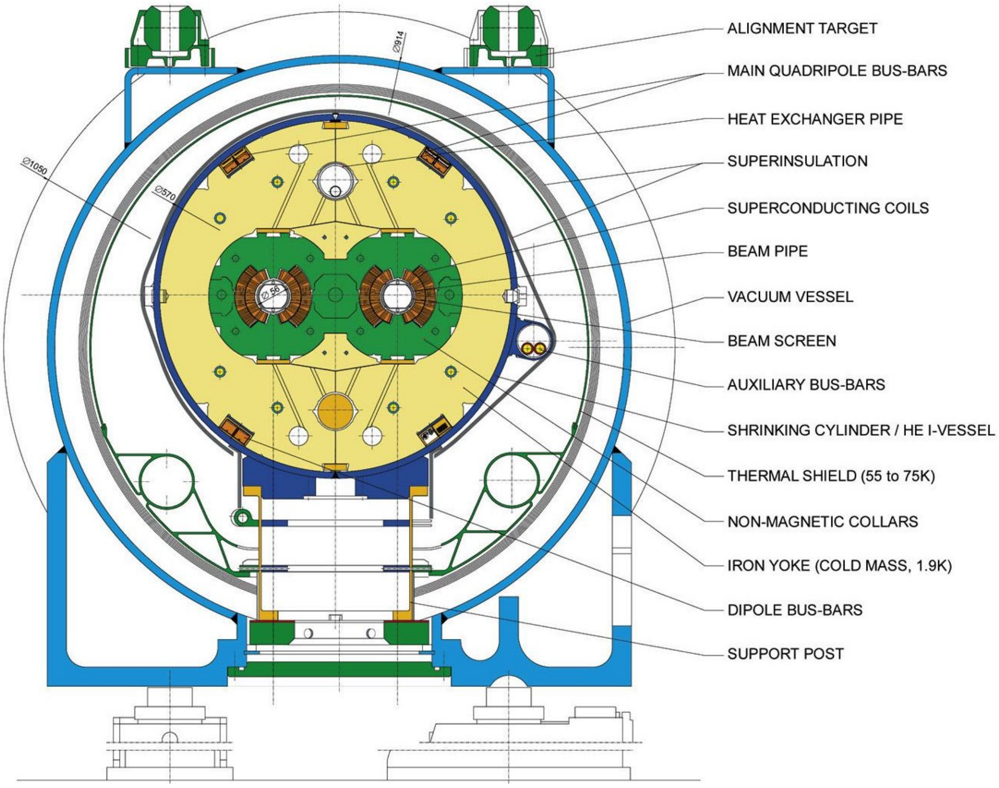


Figure 1.3: Cross-section of the LHC main dipole magnet [5].

LHC characteristics		
Parameter	Value	Unit
Total construction costs	6.51	BCHF
Circumference	26.659	km
Dipole operating temperature	1.9 (-271.3)	K (°C)
Number of main dipoles	1232	-
Number of main quadrupoles	392	-
Number of corrector magnets	6000	-
Number of circuits	8	-
Nominal magnetic dipole field	8.3	T
Nominal current per circuit	11.85	kA
Energy stored in one dipole magnet	7.1	MJ
Energy stored in one circuit	1.1	GJ
Beam Energy	362	MJ
Energy consumption (Run 1)	650	GWh
Data flow from experiments	30	PB/a

Table 1.1: LHC characteristics [1], [7]

1.1.1 Magnet Protection

The superconducting cable in the LHC main dipole magnet consists of superconducting filaments surrounded by a copper matrix. Due to the temperature increase after a quench, the resistivity of superconductor is larger than the one of copper. Therefore, the current commutes from the superconducting filaments to the copper matrix. Accordingly, there is a resistive voltage rise in the magnet which is composed of the current in the magnet multiplied by the copper resistance.

In order to detect this resistive voltage rise in the magnet, quench detection systems are installed.

The Quench Protection System (QPS) protects a magnet in case of a quench and consists of two subsystems among others: the quench heater strips (see Section 4.1) and a cold bypass diode. The purpose of the quench heaters is to expand the region of the quench, by heating up the entire magnet. This increases resistance in the whole magnet coil which leads to a bigger region of energy dissipation. The electrical circuit in Figure 1.4 shows that the heating strips (R_{HDS}) are connected with a 900 V capacitor bank discharge power supply ($C_{\mathrm{QH1}}, C_{\mathrm{QH2}}$). In case of a quench the whole magnet coil is, therefore, heated up with an energy of 2.86 kJ and a maximum current (I_{HDS}) of about 80 A. The purpose of the cold bypass diode, on the other hand, is to redirect the current in case of a quench. Once the increasing resistive voltage reaches the forward voltage of the diode (6 V), it is switched on. Therefore, the cold diode D creates a loop in which the magnet current is discharged about half a second after the quench occurred. Afterwards, it conducts the circuit current until the circuit is fully discharged. [8]

1.2 The LHC Main Dipole Circuit

The LHC is divided into eight sectors with one dipole (RB) and two quadrupole circuits (RQD and RQF) in each sector. The 154 dipole magnets and the 47 or 51 quadrupole magnets in each circuit are powered in series. Figure 1.4 shows the simplified electrical protection scheme of the LHC main dipole magnets. The corresponding circuit parameters and symbols are introduced in Table 1.2. Thereby, the magnets M001 - M154 are not the only superconducting components. Also the interconnection of the magnets, called busbar R_{BB} and the temperature transition between the room temperature and the liquid helium environment, called Current Leads (CL) are in a superconducting state and need to be protected by the QPS. The circuit is powered by a 13 kA Power Converters (PC), with a crowbar in parallel for bypassing the current in case the PC is switched off. There are two Energy Extraction (EE) units which consist of a electromechanical Switches (SW), an extraction resistor R_{EE}, and a snubber capacitor C_{SN}. Furthermore, there is a resistor R_{p} connected in the parallel path of each dipole magnet, in order to smoothen transient voltage oscillations. [9]

Parameter	Symbol	Value	Unit
Crowbar	CB		
Power Converter	PC		
Current Leads	CL		
Magnet equivalent model	M001 - M154		
Busbar Resistance (at 1.9 K)	R_{BB}	0 - 1.2	$n\Omega$
Magnet Inductance per Aperture	AP_1, AP_2	51	mH
Diode	D		
Parallel Resistance	R_{p}	100	Ω
Quench Heater Resistance (at 1.9 K)	R_{QH}	11	Ω
Quench Heater Discharge Power Supply	DQHDS		
Quench Heater Trigger	$T_{1,2}$		
Quench Heater Power Supply Capacitor	$C_{\mathrm{QH1}}, C_{\mathrm{QH2}}$	7.05	mF
Fuse	F		
Energy Extraction	EE		
Switch	SW		
Snubber Capacitor	C_{SN}	53	mF
Energy Extraction Resistance	R_{EE}	74	$m\Omega$

Table 1.2: Circuit parameter of Figure 1.4 with nominal values.

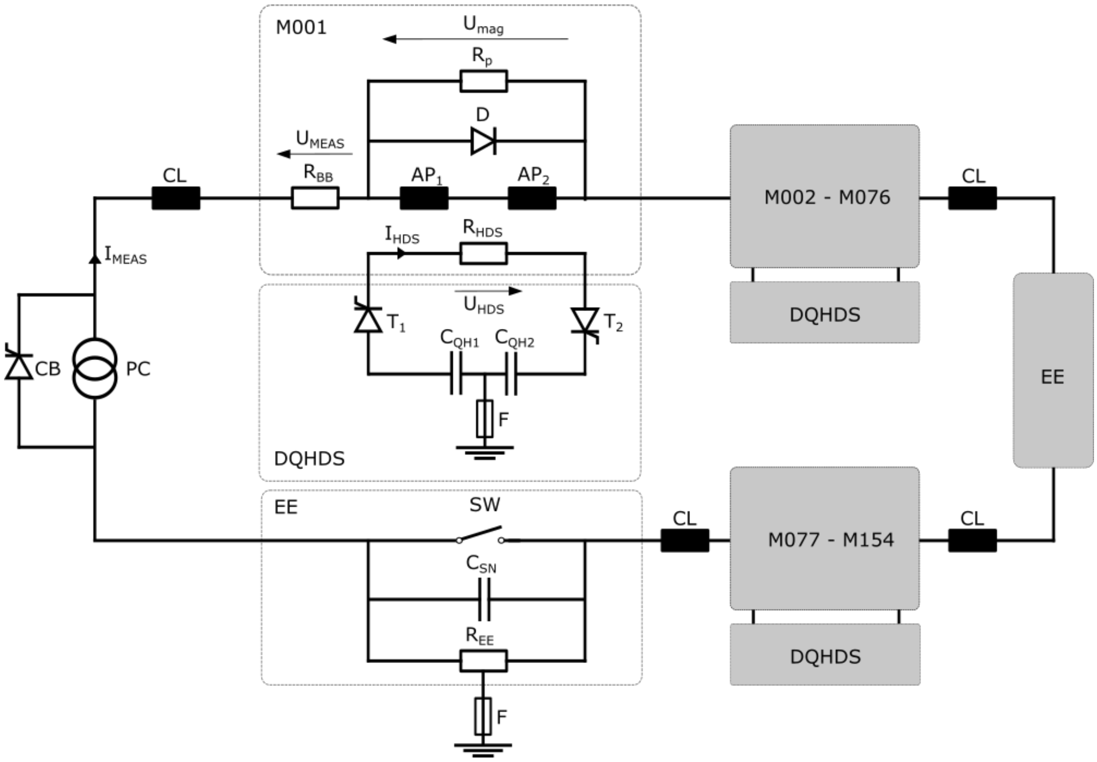


Figure 1.4: Simplified electrical protection scheme of the LHC main dipole magnets [4], [9], [10].

1.2.1 Circuit Protection

The main task of the circuit protection, is to extract the huge energy stored in the magnetic field of a magnet in a safe way. The destructive potential of this energy was has been experienced in 2008 during the commissioning of the LHC. A faulty interconnection between two of the superconducting magnets led to an arc, which caused a considerable damage (see Figure 1.5). The LHC characteristics summarized in Table Table 1.1, show the immense responsibility of the circuit protection.

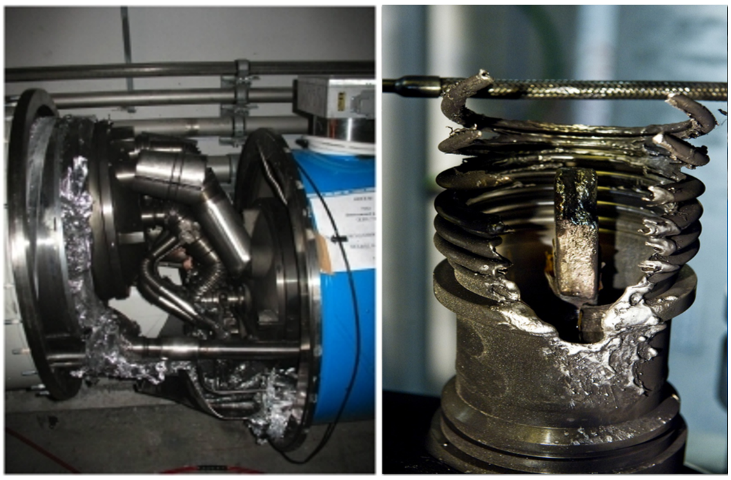


Figure 1.5: Damage of the LHC main dipole magent after the 2008 incident [11].

In general, the sequence of events in case of a quench can be described in the following steps:

1. *Beam extraction*
   The high energy proton beam (350 MJ) inside the magnet is dumped into a 7 m long water-cooled graphite block, leading to a temperature rise of up to 800 °C at the impact zone [12].

2. *Magnet protection*
   As discussed in Section 1.1.1, the combination of the quench heaters and the cold bypass diode change the path of the magnet current. Consequently, the quenched magnet is circumvented of the destructive energy of the circuit within half a second.

3. *Energy Extraction*
   With a certain delay needed to avoid spurious triggering of the detection devices, the energy extraction is activated. Consequently, the energy extraction resistor R_{EE} is switched on and the power converter is detached. Thus, the energy of the magnet is discharged over the internal resistance of the two apertures AP_1 and AP_2 which leads to a current decreases. This current decrease enhances the quench propagation due to the induction of eddy-current losses in the magnet coils. The whole process takes only a few hundreds of seconds[1] until the circuit is completely switched off [12].

1.3 Data Logging Systems

Several data logging systems are dedicated to access and store the high-volume, high-velocity and high-variety data of the LHC hardware components discussed in the previous sections. The architecture and the function of several data logging systems is summarized below.

- *Post Mortem (PM) Database [13]*
  After a failure occurred, the PM (from Latin "after death") Framework records the transient data of the Machine Protection System (MPS) equipment. The MPS devices host a circular buffer, which is frozen and sent to the PM data collection in case of a failure. The main purpose is to help experts understand the reason of the failure. For example, in case of a quench, the experts can look at the signals of the magnet, deciding whether a safe restart of the accelerator is possible again. While the signal duration during such an event is relatively short, it can contain up to 50 GB of data since the sampling frequency is very high. The PM Framework organizes this dense information and offers several resources, like the PM REST Application Programming Interface (API), database/reference access, analysis configurations, etc. A simplified architecture of the PM Framework is shown in Figure 1.6. Once the data is collected, the event builder forms groups of PM buffers and the type of event is presumed (e.g. magnet quench). Contingent on the assumption, the Post Mortem Analysis (PMA) server then starts to analyse the event. The triggering of the PMA is performed twice, after about 30 seconds (preliminary analysis results) and after about 8 minutes (finalized analysis results), in order to provide an LHC operator with up-to-date information.

- *CERN Accelerator Logging Service (CALS) [14], [15]*
  CALS is a system for continuous storage of LHC signals. With about five million data extraction requests on average per day, by more than 1000 users, CALS is a commonly used CERN database. Figure 1.7 shows the CALS architecture, which is divided into

[1] The time constant of the EE process can actually be estimated by $\tau_{EE} = L/R = 154 \cdot (L_{AP1} + L_{AP2})/(2 \cdot R_{EE}) = 154 \cdot (51\ \mathrm{mH} + 51\ \mathrm{mH})/(2 \cdot 74\ \mathrm{m\Omega}) \approx 106s$

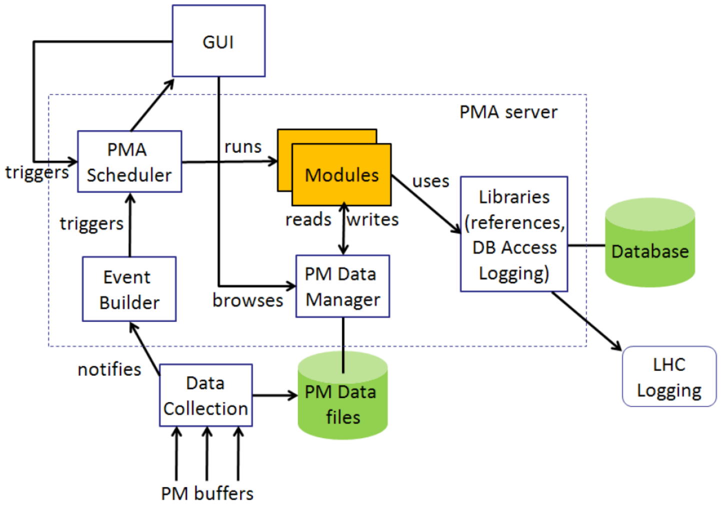


Figure 1.6: Overview of the LHC Post Mortem (PM) Framework architecture [13].

three layers: providers, persistence, and consumers. Different hardware systems provide the data and send them to the next layer. The persistence layer consists of two Oracle Real Application Clusters (RAC) databases. At the short-term Measurement Database (MDB), raw data from Java processes and other Oracle databases is stored for seven days. On the other hand, at the long-term Logging Database (LDB), a sub-set of MDB data and pre-filtered data from the Supervisory Control And Data Acquisition (SCADA) system are stored permanently. A dedicated Java API enables to extract data from the described databases for the users. Optionally a generic Java Graphical User Interface (GUI) called TIMBER or a Python wrapping of the CALS API called PyTimber can be used to extract logged data.

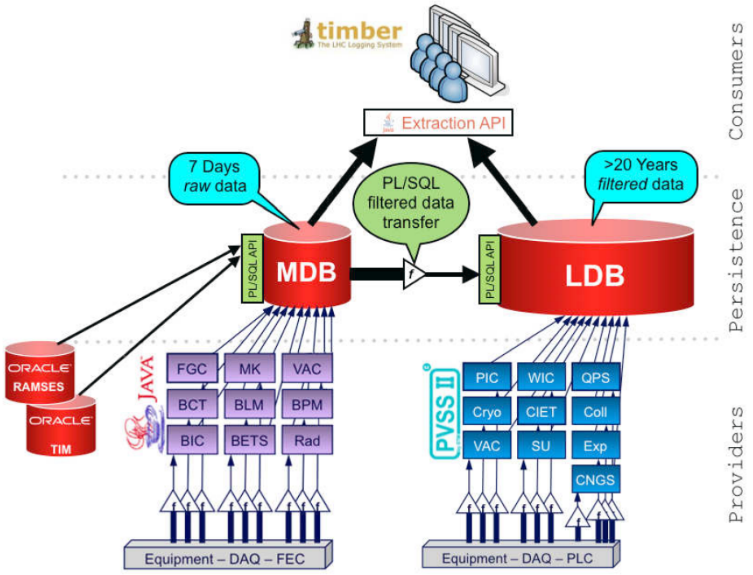


Figure 1.7: Overview of the CALS architecture [14].

- *Next Generation CERN Accelerator Logging Service (NXCALS) [14], [15]*
  CALS is currently reaching its limits in terms of performance, scalability, integration with heterogeneous analytics tools (Python, Matlab, R, Java,...), etc. Therefore, CERN recently developed the NXCALS ecosystem, which is based on Apache Spark, Hadoop, and Kafka frameworks. Furthermore, a new Java and Python API for distributed computing, built via Apache Spark, is available. This API offers a unified analytics engine for large scale data processing. Algorithms can be implemented into the data query and only the result will be returned. While most of the MPS applications still use CALS instead of NXCALS, it is already possible to discuss the advantages and applications of distributed computing during the development of the analysis algorithms in this thesis.
- *INFLUXdb [16]*
  INFLUXdb is a time-series database, optimized for fast, high-availability storage and fetching of time-series data. Since it is open-source, it offers a widely accessible solution for use cases that are using large amounts of time-stamped data, like signal monitoring or real-time analytics. It is not specifically designed for CERN, however, it is still used in many CERN projects such as monitoring of accelerator systems, experiments and data centers. Specifically, one of the four LHC detectors, called ALICE uses INFLUXdb to monitor data with a data flow of 3.4 TB/s.

1.4 Research Goals and Motivation

During several years of successful machine operation, a lot of data has been acquired from the superconducting circuits and their protection equipment. However, since the LHC safety system consists of many subsystems, the LHC signal monitoring project aims to unite, analyse, and correlate the data from the different subsystems [17]. While there has been already a lean API for time-series data acquisition of the main machine protection databases (see Chapter 2 and [18]) developed, this thesis aims to further analyse and correlate data from the machine protection subsystems. Therefore, already existing hardware monitoring applications, which have been proven to be reliable in several years of operation, should be implemented into the LHC signal monitoring environment.

The goal of this thesis is to use data-driven models to extend and supplement the implemented protection methods by learning from existing data. In particular, supervised machine learning techniques are used to evaluate whether a hardware deterioration can be predicted and to incorporate expert knowledge into a classification process. The approach used for both the implementation and the extension of existing signal monitoring applications are as general as possible in order to lay the foundation for further signal monitoring of LHC components.

1.5 Thesis Structure

This thesis provides a general overview in the fields of LHC machine protection (Chapter 2) and data science (Chapter 3, 4). Therefore, it is structured as follows:

In Chapter 2 a general introduction to the LHC machine protection measures is given. Mainly, this chapter focuses on the structure of the machine protection system and the LHC signal monitoring project. Furthermore, the reader is introduced to the used application development approach, following steps of acquisition, exploration, model creation, and monitoring.

In Chapter 3 the busbar resistance, which is the resistance of the magnet interconnection, is calculated and analyzed in order to manually detect a possible deterioration of the soldered joints over time. Additionally, new features are derived which describe the confidence of the busbar resistance calculation. Furthermore, a GUI is be presented, which enables the users to browse through the calculated features across time and circuit.

In Chapter 4 the quench heater discharge is analyzed. The characteristics of the quench heaters is described by extracting features from the quench heater signals, which can then be used to detect possible damages of the heater. Supervised machine learning is used in combination with the already existing threshold-based classification method, to propose a hybrid classification system. While manual interventions have been necessary in the past, this classification system makes it possible to incorporate expert decisions into the classification process semi-automatically.

Finally, Chapter 5 provides a discussion about the results of this work and possible future steps to continue the development of LHC signal monitoring applications beyond this thesis.

2 The LHC Signal Monitoring Project

The LHC signal monitoring project was started in April 2018 with the goal of providing a unified monitoring approach for the LHC superconducting circuits, that eventually could be extended to other LHC hardware. Since this thesis will provide applications within the LHC signal monitoring environment, this section will give an insight into the project architecture, the software stack and the application development approach of the project.

2.1 Project Architecture

The LHC signal monitoring project has been created by a team of developers within the TE-MPE group to ensure that there is an environment where it is easy to implement LHC hardware monitoring applications. Therefore, several works have been established to create such an environment [17], [18]. The architecture of the LHC signal monitoring project consists of several layers, including but not limited to the elements stated bellow.

DbSignal class

The DbSignal class brings several simplifications and generalizations. First, it simplifies data acquisition. Different databases, discussed in Section 1.3, have different commands to query data and also return different formats. The DbSignal class unifies database access, which makes it possible to gather data from all data sources with one method. This function returns a DataFrame (DF) which is a special data structure provided by an external library [19]. Secondly the DbSignal class generalizes the time conversion. Depending on the database and the application, different timestamp formats are used (see Table 2.1). The DbSignal class allows the easy transition between those formats. Finally, it is also possible to process the queried signal.

Different time classes		
Format	Example	Field of application
unix time (integer with ns precision)	1525125600000000000	PM key format
string	“2018-05-01 00:00:00+01:00”	Human readable
datetime	datetime.datetime(2018, 5, 1, 0, 0, tzinfo=<DstTzInfo 'Europe/Zurich' CEST+2:00:00 DST>)	CALS key format
pandas datetime	Timestamp('2018-05-01 00:00:00+0200', tz='Europe/Zurich')	Used in analysis modules, INFLUXdb
unix time (float with μs precision)	15251256e9	Returned by Pytimber, Scientific notation

Table 2.1: Different time classes in the LHC signal monitoring environment [18].

Metadata

In order to query the right signals, it is necessary to provide the exact system (e.g. PC, EE, CL), the exact signal name (e.g. U_{MEAS}, I_{MEAS}) and the exact circuit name (e.g. for the main dipole circuit: RB.A12, RB.A23, RB.A34 etc.). The LHC signal monitoring metadata module provides a complete list of all those parameters. In addition, there is a mapping from a circuit name to magnet names, voltage feelers, etc. This simplifies the process of finding the right signals for the user.

Reference

Several hardware components have reference signals. Those signals can be acquired during HWC, which is a detailed hardware testing campaign, performed prior to restart of operation after an extended stop. Furthermore, some logged signals can be defined as reference signals, once experts have checked them. For some analysis methods (see Chapter 4) those reference signals can be used for comparison with the current signal.

Applications

Applications within the LHC signal monitoring project get implemented in several stages. A detailed explanation of each analysis stage is stated bellow [17].

1. *Acquisition*
   The acquisition step is devoted to analysing a single event. Therefore, the first step of the acquisition is the definition of a specific hardware component, the user wants to analyse during a specific time. The LHC signal monitoring API helps to find the right naming and the right timestamps in the correct format. Afterwards a specific event can be filtered out, since some hardware behaviours, only occurs during certain operating states, which are further described in Section 3.1. If there is any problem with the logged signal, like missing data or a wrong name, those signals can be filtered out in the next step with the preprocessing functions of the API. Certain dedicated features can then be calculated in order to compress signals and store only the most important information from them. Finally, the results of the acquisition step are stored and documented. Overall, the acquisition step integrates existing experience encoded in other analysis modules (e.g. LabVIEW) as well as new modules.

2. *Exploration*
   In the exploration step, the goal is to perform analysis over an extended period of time in order to obtain a distribution of features. To do so, certain algorithms are applied to the signals and the features from the acquisition phase. Those features are then tracked over time to see how they evolve and they provide a tool for experts to gain a better understanding of the current machine state.

3. *Model Creation*
   In general, mathematical models can be used to represent a system to better understand its performance. Consequently, this step is devoted to use the data from the exploration step in order to extract a statistical model which describes the system. Such a model can then be employed for performance evaluation and optimization, availability and reliability studies, predictive maintenance etc.

4. *Monitoring*
   Since the LHC is currently in its second Long Shutdown (LS), this thesis will mainly focus on the acquisition and the exploration of the past signals. However, before the start of

run 3 in 2021, the next step of online hardware monitoring should be implemented as well in order to use the knowledge from the past signals and compare the current values of the features to their historical distribution. This enables to determine whether a hardware component is subject to deterioration

2.2 Software Stack

In order to make certain figures and code snippets in the next chapters to be understood more easily, the software stack of the LHC signal monitoring project is stated below.

- *Programming Language: Python*
  While many programming languages are capable of applying certain mathematical operations, Python is particularly often used in other signal monitoring projects. The reason is its huge amount of statistical libraries (e.g. pandas, SciPy) and its big community.
- *Code Execution: SWAN [20]*
  The CERN Service for Web-based ANalysis (SWAN) is an implementation of Jupyter notebooks [21] embedded in the CERN environment. This means, it is a web application for execution, documentation, and visualisation of analysis code within every browser. All the necessary libraries are pre-installed, which makes the usage very user friendly. Since it is possible for people with a CERN computer account to access those notebooks from any device with internet access, it is easy to distribute them via the SWAN Galleries or via GitLab. SWAN uses CERNBox, the CERN cloud data storage, as a repository.
- *Versioning: GitLab [22]*
  GitLab is a web-based, open-source version control system which offers repository hosting as well as code review and collaboration features (e.g. merge requests workflow). An especially important GitLab feature is the Continuous Integration pipeline, which allows developers to build, test, and validate their new code before merging it into the master repository. GitLab was introduced at CERN in spring 2015 as an alternative to Apache Subversion (SVN) by the CERN IT department.
- *Static Code Analysis: SonarQube [23]*
  SonarQube is a web-based, open-source platform for continuous inspection of code quality. It offers automatic source code checks to find errors, security leaks, and bad structure. It is used in the LHC signal monitoring project in order to produce codes with unified coding standard, low code complexity, and little code redundancy.
- *Storage: INFLUXdb, CSV*
  INFLUXdb, which is further explained in section 1.3, will be used to store the time-series data of the LHC signal monitoring analysis in the future. Until it is set up, CSV files offer a simple solution to temporary store the widely used tabular data within this project (e.g. the pandas.DataFrame format; see Section 2.1).

3 Busbar Resistance Calculation

As an initial development of an application within the LHC signal monitoring project, the busbar resistance R_{BB} is analyzed. The objective of this initial development is to provide a proof of concept for applications within the LHC signal monitoring project. Therefore, the opportunities and advantages of signal acquisition, graphical visualization, and data analysis within the LHC signal monitoring project environment is investigated. However, this use case is only described briefly, as the focus of this thesis lies in the analysis of the quench heater signals in Chapter 4. For further information a separate report has been published as a documentation of the busbar resistance calculation [24].

3.1 Overview

Two main magnets of the LHC are interconnected by soldering together the superconducting Nb-Ti Rutherford cable with an Sn-Ag alloy (see Figure 3.1). Additionally, the overlap of 120 mm is stabilized with a copper busbar [25]. The lower the resulting resistance of this solder joint is, the better is the quality of this interconnection. Thereby, a deterioration of the solder joint condition can be detected, by monitoring this resistance.

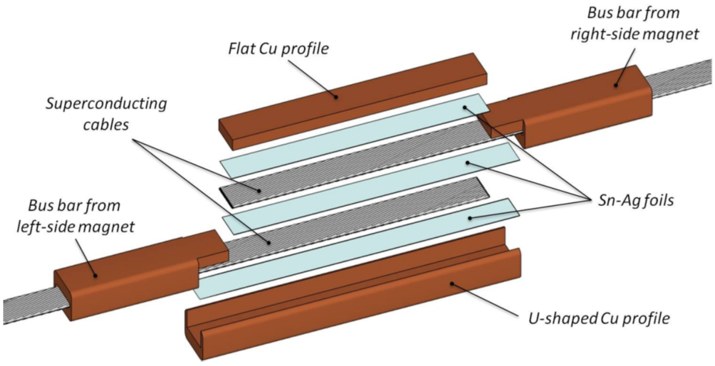


Figure 3.1: Illustration of a typical magnet-to-magnet interconnection [26].

Busbar Resistance Signals

In this chapter, the time discrete signals $\boldsymbol{z}_m$, with the corresponding timestamps $\boldsymbol{t}^{\mathrm{z}}$, are defined as vectors of the form:

$$\boldsymbol{z}_m = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} \in \mathbb{R}^N, \qquad \boldsymbol{t}^{\mathrm{z}} = \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix} \in \mathbb{R}^N,$$

for $n \in [1, ..., N]$ and $m \in [1, ..., M]$, where N is the amount of sample points and M is the number of magnet interconnections. Therefore, the electrical quantities U_{MEAS} and I_{MEAS} from the circuit shown in Figure 1.4, are defined as:

$$\boldsymbol{u}_m = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} \in \mathbb{R}^N, \qquad \boldsymbol{i}_m = \begin{bmatrix} i_1 \\ \vdots \\ i_N \end{bmatrix} \in \mathbb{R}^N,$$

The signals $\boldsymbol{Z}_d$ of all magnets can be written as:

$$\boldsymbol{U}_d = \{\boldsymbol{u}_1, ..., \boldsymbol{u}_M\} \in \mathbb{R}^{N \times M}, \qquad \boldsymbol{I}_d = \{\boldsymbol{i}_1, ..., \boldsymbol{i}_M\} \in \mathbb{R}^{N \times M},$$

for $d \in [1, ..., D]$, where D the number of machine cycles (events). Overall, this leads to a tensor $\boldsymbol{Z}$ of the form:

$$\boldsymbol{U} = \{\boldsymbol{U}_1, ..., \boldsymbol{U}_D\} \in \mathbb{R}^{N \times M \times D}, \quad \boldsymbol{I} = \{\boldsymbol{I}_1, ..., \boldsymbol{I}_D\} \in \mathbb{R}^{N \times M \times D}.$$

Furthermore, there is vector $\boldsymbol{b}$ for the beam mode of the LHC which also has a corresponding time stamp $\boldsymbol{t}^{\mathrm{b}}$:

$$\boldsymbol{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_K \end{bmatrix} \in \mathbb{N}^K, \qquad \boldsymbol{t}^{\mathrm{b}} = \begin{bmatrix} t_1 \\ \vdots \\ t_K \end{bmatrix} \in \mathbb{R}^K,$$

for $k \in [1, ..., K]$, where K is the amount of sample points. Thereby, each value b_k is logged only in case the beam mode changes. The values of $\boldsymbol{b}$ are assigned to the following semantic content:

$$b_k = \begin{cases} 1 & \text{"no mode"} \\ 2 & \text{"setup"} \\ 3 & \text{"pilot injection"} \\ 4 & \text{"intermediate injection"} \\ 5 & \text{"nominal injection"} \\ 6 & \text{"before ramp"} \\ 7 & \text{"ramp"} \\ 8 & \text{"flat top"} \\ 9 & \text{"squeeze"} \\ 10 & \text{"adjust beam on flat top"} \\ 11 & \text{"stable beam for physics"} \\ 12 & \text{"unstable beam"} \\ 13 & \text{"beam dump"} \\ 14 & \text{"ramp down"} \\ 15 & \text{"recovering"} \\ 16 & \text{"inject and dump"} \\ 17 & \text{"circulate and dump"} \\ 18 & \text{"recovery after a beam permit flag drop"} \\ 19 & \text{"pre-cycle before injection"} \\ 20 & \text{"warning beam dump"} \\ 21 & \text{"no beam or preparation for beam."} \end{cases}$$

Existing Analysing Methods

Several existing works have been dedicated to the analysis of the busbar resistance [25], [27]. These methods use data from the QPS, stored in the CALS database. With this data, the calculation is triggered if the LHC is working approximately one hour at injection level $2 \leq b_n < 6$ and one hour with a stable beam $8 \leq b_n < 14$. The methods for calculating this resistance are further described in Section 3.2.

Goals

The goal of the busbar resistance calculation is to analyze the condition of the quadrupole magnet interconnections by using the same features as in the existing analysing methods. Therefore, the busbar resistance should be calculated periodically during the period of 2015-2018 in order to detect a possible deterioration of the soldered joints over time. It should further be examined if any patterns within the historical development of these features can be identified.

3.2 Acquisition

The CALS database logs the QPS signals U_{MEAS} and I_{MEAS} continuously with a 10 Hz sampling frequency and a resolution of 1.5 nV for the voltage and 2 ppm for the current [25]. A sequence of acquisition steps, described in Section 2.1, is necessary to extract features from those events, which will then be used for further analysis.

Input for Data Query

Each signal is exactly defined once the location of the event (magnet) and the corresponding time period is given. The LHC signal monitoring API will be used together with the LHC signal monitoring metadata to load the data from the CALS database (see Section 2.1).[2]

Search Events

Similar to the existing methods, the signals are queried for time range in which the machine operates with a constant current, i.e., at the beam injection and stable beams. Hence, the calculations uses $\boldsymbol{Z}_d^{\mathrm{inj}}$ where $2 \leq b_n < 6$ and $\boldsymbol{Z}_d^{\mathrm{sb}}$ where $8 \leq b_n < 14$ for each machine cycle $d \in [1, ..., D]$.

Filter Events

Due to a high signal to noise ratio of the voltage signals, it is necessary to define a minimum signal duration over which the signal is averaged in order to obtain reasonable results. Consequently, only events for which $\boldsymbol{T}_d^{\mathrm{inj}} = t^{b=6} - t^{b=2} > 30$ min and $\boldsymbol{T}_d^{\mathrm{sb}} = t^{b=14} - t^{b=8} > 30$ min are further pre-processed.

Preprocessing

Once the signals $\boldsymbol{z}_m$ for $m \in [1, ..., M]$ are queried for a event d, they are further pre-processed. By assuming N normal distributed sample points $z_n \sim \mathcal{N}\left(\mu^{\mathrm{z}_m}, (\sigma^{\mathrm{z}_m})^2\right)$ for $m \in [1, ..., N]$, it is

[2] As the NXCALS database is evolving, also the busbar resistance calculation has already been adjusted to use NXCALS data.

possible to calculate the values μ^{z_m} and σ^{z_m} by:

$$\mu^{z_m} = \boldsymbol{\phi}_{\mu}(\boldsymbol{z}_m) = \overline{z}_m = \frac{1}{N}\sum_{n=1}^{N} z_n, \tag{3.1}$$

$$\sigma^{z_m} = \boldsymbol{\phi}_{\sigma}(\boldsymbol{z}_m) = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(z_n - \mu^{z_m})^2}. \tag{3.2}$$

The Gaussian density function is defined as:

$$\mathcal{N}\left(\mu^{z_m},\, (\sigma^{z_m})^2\right) = \frac{1}{\sigma^{z_m}\sqrt{2\pi}} e^{-(z_n - \mu^{z_m})^2 / 2(\sigma^{z_m})^2}. \tag{3.3}$$

In particular, the values μ^{z_m} and σ^{z_m} are calculated for the signals $\boldsymbol{u}_m^{\text{inj}}$,$\boldsymbol{i}_m^{\text{inj}}$,$\boldsymbol{u}_m^{\text{sb}}$ and $\boldsymbol{i}_m^{\text{sb}}$.

Feature Engineering

The busbar resistance r_m is a measure for the condition of the magnet interconnection $m \in [1, ..., M]$. Consequently, this resistance is calculated by:

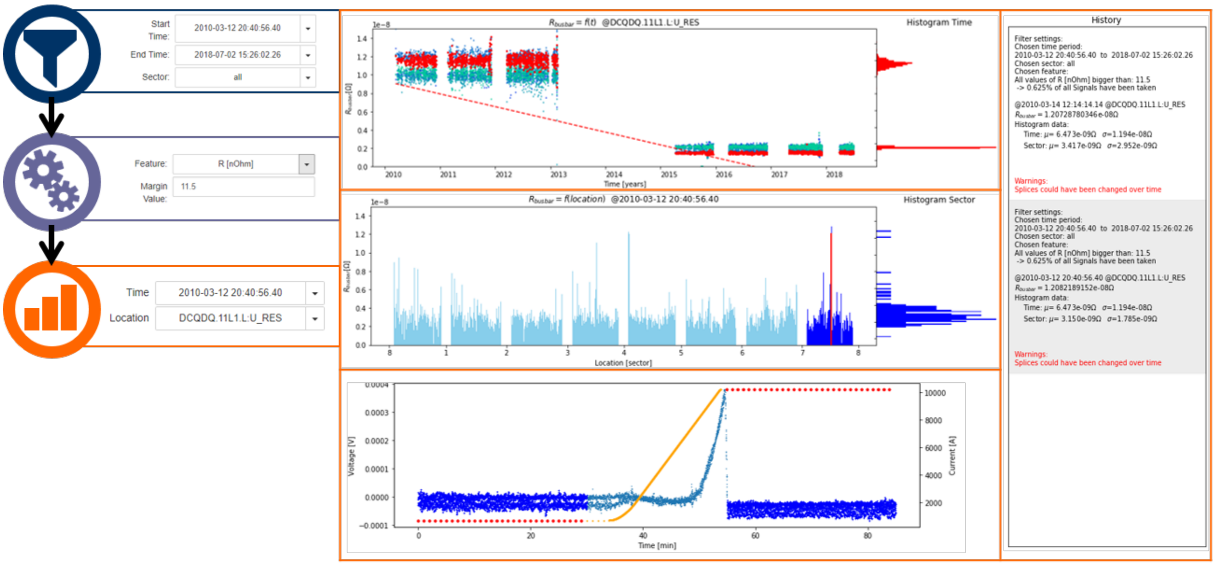
$$r_m = \left| \frac{\overline{u}_m^{\text{sb}} - \overline{u}_m^{\text{inj}}}{\overline{i}_m^{\text{sb}} - \overline{i}_m^{\text{inj}}} \right| \tag{3.4}$$

In addition, the signal to noise ratio is calculated as an additional feature for the signals $\boldsymbol{u}_m^{\text{inj}}$,$\boldsymbol{i}_m^{\text{inj}}$,$\boldsymbol{u}_m^{\text{sb}}$ and $\boldsymbol{i}_m^{\text{sb}}$ according to:

$$s^{z_m} = \frac{|\mu^{z_m}|}{|\sigma^{z_m}|}. \tag{3.5}$$

3.3 Exploration

In order to validate the successful implementation of the busbar resistance calculation, the busbar resistance is calculated during the time period of 2015-2018 for the main quadrupole magnets. During this time period $D = 2537$ events could be detected for the $M = 800$ main quadrupole magnet interconnections.



Start Time: 2010-03-12 20:40:56.40

End Time: 2018-07-02 15:26:02.26

Sector: all

Feature: R [nOhm]

Margin Value: 11.5

Time: 2010-03-12 20:40:56.40

Location: DCQDQ.11L1.L:U_RES

History

Filter settings:
Chosen time period:
2010-03-12 20:40:56.40 to 2018-07-02 15:26:02.26
Chosen sector: all
Chosen feature:
All values of R [nOhm] bigger than: 11.5
-> 0.625% of all Signals have been taken

@2010-03-14 12:14:14.14 @DCQDQ.11L1.L:U_RES
R_{busbar} = 1.20728780346e-08Ω
Histogram data:
Time: μ= 6.473e-09Ω σ=1.194e-08Ω
Sector: μ= 3.417e-09Ω σ=2.952e-09Ω

Warnings:
Splices could have been changed over time

Filter settings:
Chosen time period:
2010-03-12 20:40:56.40 to 2018-07-02 15:26:02.26
Chosen sector: all
Chosen feature:
All values of R [nOhm] bigger than: 11.5
-> 0.625% of all Signals have been taken

@2010-03-12 20:40:56.40 @DCQDQ.11L1.L:U_RES
R_{busbar} = 1.2082189152e-08Ω
Histogram data:
Time: μ= 6.473e-09Ω σ=1.194e-08Ω
Sector: μ= 3.150e-09Ω σ=1.785e-09Ω

Warnings:
Splices could have been changed over time

Figure 3.2: Graphically modified representation of the developed GUI for feature analysis.

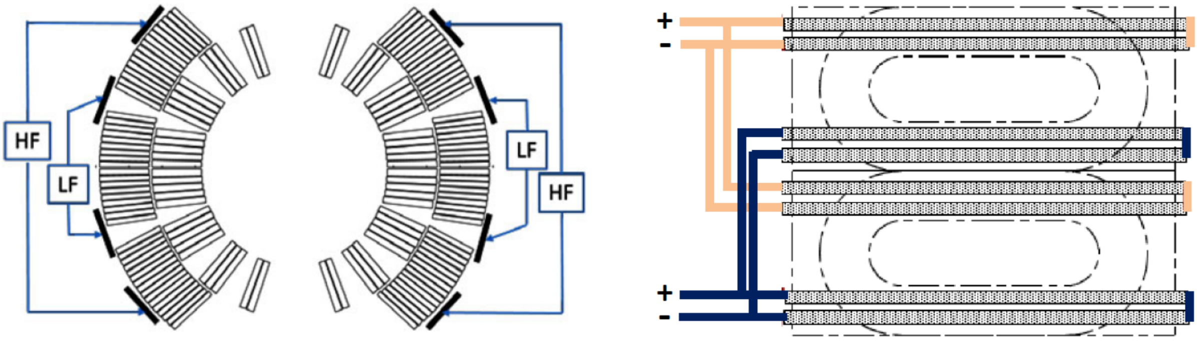
Figure 3.2 shows a graphically augmented representation of the developed GUI, which is developed in SWAN, for browsing through this data across time and location.[3] The user can chose the desired time and location with the input parameter on the right hand side. Then the corresponding signals and features are displayed. In particular the resistance is plotted as a function of time in the upper plot and as a function of the location in the middle plot. In the lower plot the origin signals z_m of the chosen feature, are plotted as a function of time. In order to analyse those features, the GUI is able to calculate the linear regression (see Section 4.2.5), within a given time period. This linear regression is visualized as a dotted line in the upper plot. With this GUI the same outliers of quadrupole busbars, as mentioned in existing methods [29], could be detected.

[3] A detailed explanation of this GUI is presented in [28].

4 Quench Heater Monitoring

4.1 Overview

As discussed in Chapter 1, the Quench Heaters (QHs) are a safety measure to mitigate the negative impacts of a magnet quench. They are made out of austenitic stainless steel strips which are partially plated with copper and they are attached to the outer layer of the magnet coils as shown in Figure 4.1. The QH strips are about 15 mm wide and 25 µm thick, which leads to a resistance of about 1.3 Ω/m at a temperature of 1.9 K [8]. The steel strips are surrounded by electrical insulation which is strong enough to withstand the high voltages, low temperatures, high compression forces, and ionizing radiation [29].



(a) QH layout in magnet cross section. The HF and the LF thereby refer to the field region of the heater circuit [29].

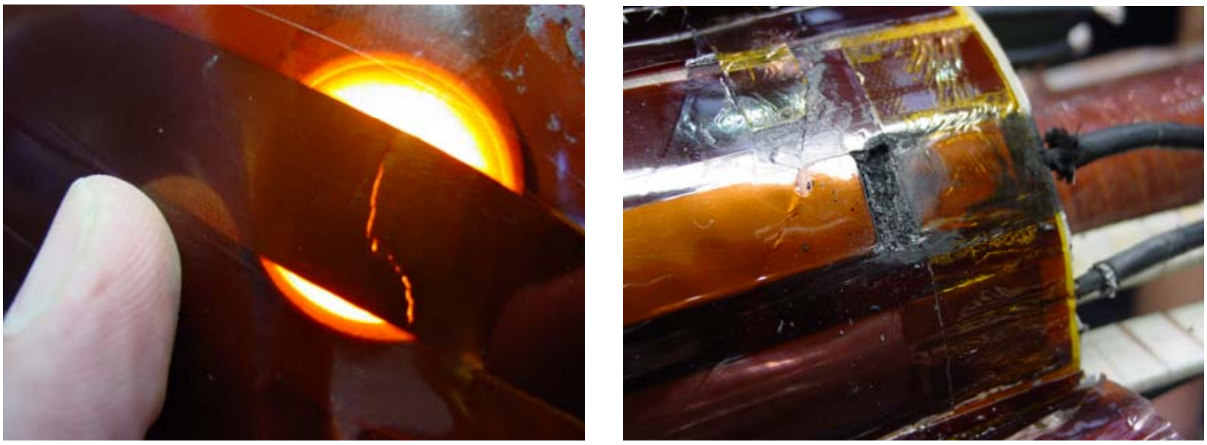
(b) QH HF connection scheme in side-view [30].

Figure 4.1: QH layout and connection scheme of one main dipole aperture.

The schematic shown in Figure 4.1(a) pictures one dipole magnet aperture with two High Field (HF) QH circuits and two Low Field (LF) QH circuits. A double-aperture dipole magnet, therefore, contains 16 heating strips connected in 8 circuits. However, only the HF QH circuits are actively powered during operation. The LF QH circuits are kept as spares in case of failures. This means that there are 4928 active QH circuits in dipole magnets on which this chapter will focus on. Additionally, there are 1148 QH circuits in other LHC superconducting magnets (main quadrupoles, inner triplets, etc.), leading to a total of 6076 QH circuits (including the main dipole magnets).

During the HWC all QH circuits are extensively tested in order to validate their flawless functionality. This test consists of a resistance measurement, a high voltage qualification test (at 1.9 K and room temperature), and a discharge test (at 1.9 K) [29].

Typically, a failure can occur in two regions: (i) the straight section of the magnet (see Figure 4.2(b)), which is most likely due to a failure in the quench heater fabrication: (ii) or at the strip turn in the extremities of the magnet, which is most likely due to an increased pressure in the coil ends (see Figure 4.2(b)). In the period from October 2007 to May 2016, 10 "faulty" QHs in the main dipole magnets have been repaired or replaced. Those failures



(a) QH steel strip crack which could not have been detected during the HWC tests but was seen after disassembling the magnet.

(b) QH short-to-ground in the coil head.

Figure 4.2: Pictures of damaged QHs strips [31].

were detected as short-to-ground during a discharge or as an electrical insulation fault after a successful discharge test [31]. However, the problem is that a failure like this can be very hard to detect under certain circumstances. Even if a steel strip is opened up to 90%, it is possible that there is no abnormality during the resistance test, and that the heater withstands the high voltage qualification test before it burns through (see Figure 4.2(a)). Therefore, the QH discharges are analyzed during machine operation as well in order to find failures or precursors of failures as soon as possible. However, also the discharge analysis during HWC and during operations can only make estimates about the QH condition. The actual state of the QH only reveals after disassembling the magnet. The shape of a QH discharge during a failure or a precursor is further discussed in Section 4.2.5

4.1.1 Quench Heater Signals

In this chapter the time discrete signals $\boldsymbol{z}_c$ with the corresponding timestamps $\boldsymbol{t}$ are defined as vectors of the form:

$$\boldsymbol{z}_c = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} \in \mathbb{R}^N, \qquad \boldsymbol{t} = \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix} \in \mathbb{R}^N,$$

for $n \in [1, ..., N]$ and $c \in [1, ..., C]$, where N is the amount of sample points and C is the amount of QH circuits ($C = 4$ in case of the main dipoles). Therefore, the electrical signals $U_{\text{HDS}}, I_{\text{HDS}}$, and R_{HDS} from the circuit shown in Figure 1.4, are defined as:

$$\boldsymbol{u}_c = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} \in \mathbb{R}^N, \qquad \boldsymbol{i}_c = \begin{bmatrix} i_1 \\ \vdots \\ i_N \end{bmatrix} \in \mathbb{R}^N, \qquad \boldsymbol{r}_c = \begin{bmatrix} u_1/i_1 \\ \vdots \\ u_N/i_N \end{bmatrix} = \begin{bmatrix} r_1 \\ \vdots \\ r_N \end{bmatrix} \in \mathbb{R}^N,$$

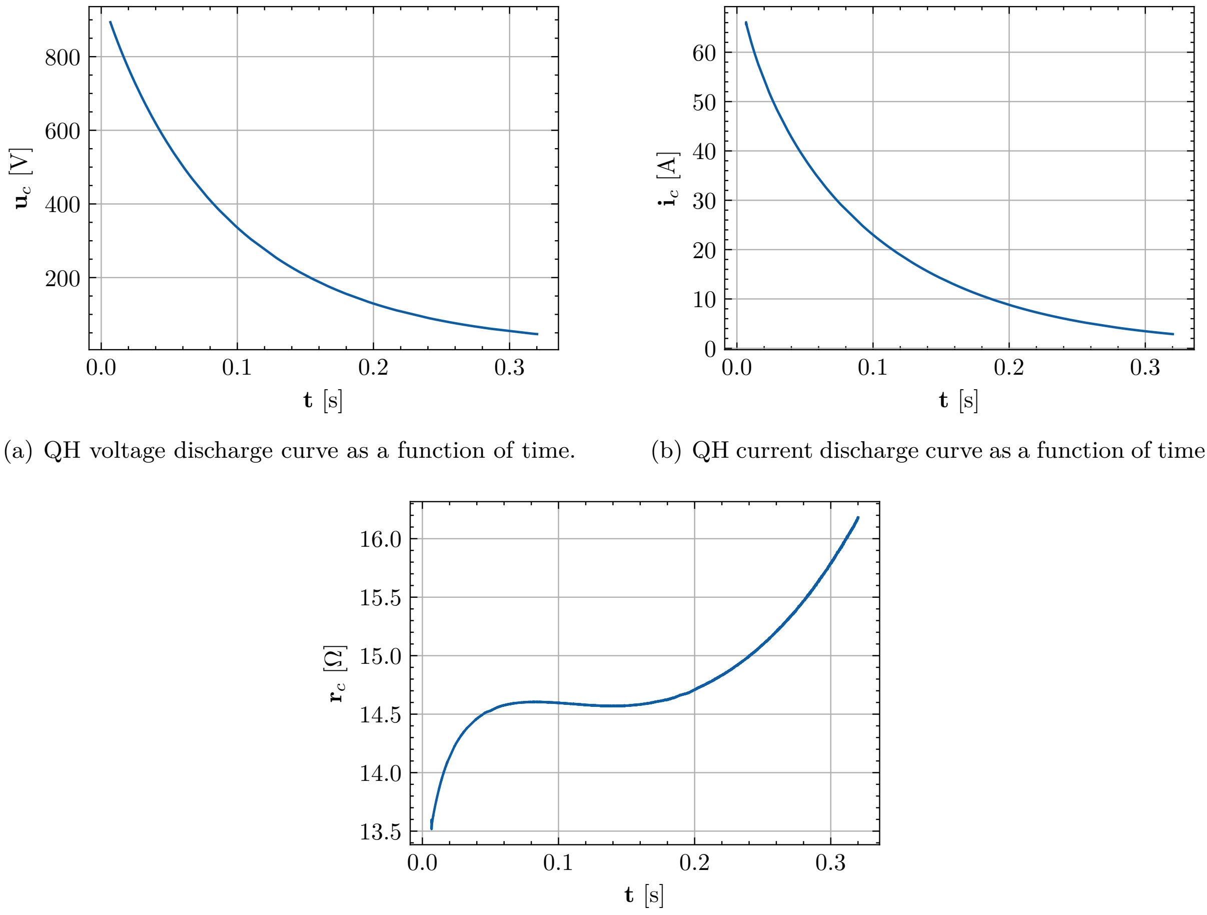
where the resistance $\boldsymbol{r}_c$ is calculated for all $i_n \neq 0$. The signals $\boldsymbol{Z}_d$ from a magnet can be written as:

$$\boldsymbol{U}_d = \{\boldsymbol{u}_1, ..., \boldsymbol{u}_C\} \in \mathbb{R}^{N \times C}, \qquad \boldsymbol{I}_d = \{\boldsymbol{i}_1, ..., \boldsymbol{i}_C\} \in \mathbb{R}^{N \times C}, \qquad \boldsymbol{R}_d = \{\boldsymbol{r}_1, ..., \boldsymbol{r}_C\} \in \mathbb{R}^{N \times C},$$

for $d \in [1, ..., D]$, where D the number of events during 2014-2018. Overall, this leads to a tensor $\boldsymbol{Z}$ of the form:

$$\boldsymbol{U} = \{\boldsymbol{U}_1, ..., \boldsymbol{U}_D\} \in \mathbb{R}^{N \times C \times D}, \quad \boldsymbol{I} = \{\boldsymbol{I}_1, ..., \boldsymbol{I}_D\} \in \mathbb{R}^{N \times C \times D}, \quad \boldsymbol{R} = \{\boldsymbol{R}_1, ..., \boldsymbol{R}_D\} \in \mathbb{R}^{N \times C \times D}.$$

Furthermore, $\boldsymbol{Z}^{\star}$ indicates a reference signal selected by experts, $\boldsymbol{Z}^{\circ}$ a signal before pre-processing, and $\boldsymbol{Z}^{\diamond}$ a normalized signal.



(a) QH voltage discharge curve as a function of time.

(b) QH current discharge curve as a function of time.

(c) QH resistance curve as a function of time.

Figure 4.3: Typical QH discharge signals.

In Figure 4.3 the pre-processed signals $\boldsymbol{u}_c$, $\boldsymbol{i}_c$, and $\boldsymbol{r}_c$ are plotted on the vertical axis and the time vector t on the horizontal axis.[4] Due to the ohmic-capacitive characteristic of the circuit discussed in Section 1.1.1, voltage and current decays exponentially during a QH discharge and can be calculated by:

$$\boldsymbol{u}_c = \widehat{u}_c \mathrm{e}^{-\frac{\boldsymbol{t}}{\tilde{\boldsymbol{\tau}}}}, \quad \boldsymbol{i}_c = \widehat{i}_c \mathrm{e}^{-\frac{\boldsymbol{t}}{\tilde{\boldsymbol{\tau}}}}, \tag{4.1}$$

where $\widehat{u}_c$ and $\widehat{i}_c$ is the maximum value of the decay and $\tilde{\boldsymbol{\tau}} \in \mathbb{R}^N$ denotes the characteristic time of the pseudo-exponential decay.[5]

[4] In this particular chapter, the algorithms will be explained with signals from the first QH inside the dipole magnet A32L5 for a PM timestamp at "2015-03-18 05:33:55.135000 (GMT+1)".

[5] As the resistance of the QH strips changes over time due to the increasing temperature, the discharge curve of the voltage and the current is pseudo-exponential. Consequently, the decay cannot be described with a time constant τ, but with a characteristic time $\tilde{\boldsymbol{\tau}}$.

The QH characteristic features are denoted by:

$$\boldsymbol{x}_f^{\mathrm{z}} = \boldsymbol{\phi}_f(\boldsymbol{Z}_d) \in \mathbb{R}^L, \quad \boldsymbol{x}_f^{\mathrm{z}\star} = \boldsymbol{\phi}_f(\boldsymbol{Z}_d^{\star}) \in \mathbb{R}^L, \quad \boldsymbol{x}_f^{\mathrm{z,c2c}} = \boldsymbol{\phi}_f(\boldsymbol{Z}_d, \boldsymbol{Z}_d^{\star}) \in \mathbb{R}^L,$$

for $f \in [1, ..., F]$ and $l \in [1, ..., L]$, where F^{z} is the amount of different features per signal. Furthermore, L is the dimension of each feature, which is similar to the amount of circuits $C = 4$ except for the calculation of the similarity matrix where it is $(C-1)! = 6$. This leads to a feature matrix of the form:

$$\boldsymbol{X}^{\mathrm{z}} = \{\boldsymbol{x}_1^{\mathrm{z}}, ..., \boldsymbol{x}_F^{\mathrm{z}}\} \in \mathbb{R}^{F^{\mathrm{z}} \times L}, \quad \boldsymbol{X}^{\mathrm{z}\star} = \{\boldsymbol{x}_1^{\mathrm{z}\star}, ..., \boldsymbol{x}_F^{\mathrm{z}\star}\} \in \mathbb{R}^{F^{\mathrm{z}} \times L} \ .$$

Those signals can then be compared to the reference signals, which leads to a matrix:

$$\check{\boldsymbol{X}}^{\mathrm{z}} = \boldsymbol{X}^{\mathrm{z}} - \boldsymbol{X}^{\mathrm{z}\star} = \{\boldsymbol{x}_1^{\mathrm{z}} - \boldsymbol{x}_1^{\mathrm{z}\star}, ..., \boldsymbol{x}_F^{\mathrm{z}} - \boldsymbol{x}_F^{\mathrm{z}\star}\} \in \mathbb{R}^{F^{\mathrm{z}} \times L}.$$

Furthermore, one can reduce the dimension of the feature matrix by taking the mean value over L of each feature vector $\boldsymbol{x}_f^{\mathrm{z}}$:

$$\overline{\boldsymbol{x}}^{\mathrm{z}} = \{\overline{x}_1^{\mathrm{z}}, ..., \overline{x}_F^{\mathrm{z}}, \overline{x}^{\mathrm{z,c2c}}\} \in \mathbb{R}^{F^{\mathrm{z}}+1}.$$

All those features can be put in one feature matrix $\boldsymbol{X}_d$, which is given as:

$$\boldsymbol{X}_d^{\mathrm{z}} = \{\boldsymbol{X}^{\mathrm{z}}, \boldsymbol{X}^{\mathrm{z}\star}, \check{\boldsymbol{X}}^{\mathrm{z}}, \boldsymbol{x}^{\mathrm{z,c2c}}, \overline{\boldsymbol{x}}^{\mathrm{z}}\} \in \mathbb{R}^{(4F^{\mathrm{z}}+2) \times L}.$$

For all three QH signals this leads to $F = F^{\mathrm{u}} + F^{\mathrm{i}} + F^{\mathrm{r}}$ features which can be merged to.

$$\boldsymbol{X}_d = \{\boldsymbol{X}_d^{\mathrm{u}}, \boldsymbol{X}_d^{\mathrm{i}}, \boldsymbol{X}_d^{\mathrm{r}}\} \in \mathbb{R}^{(4F+2) \times L}.$$

For D events during 2014-2018, this leads again to a tensor $\boldsymbol{X}$ in which a discrete class label y_d is assigned to each matrix $\boldsymbol{X}_d$:

$$\boldsymbol{X} = \{\boldsymbol{X}_1, ..., \boldsymbol{X}_d\} \in \mathbb{R}^{(4F+2) \times L \times D}, \qquad \boldsymbol{y} = \{y_1, ..., y_d\} \in \mathbb{Z}^D,$$

In this thesis the class labels are assigned to the following semantic content:

$$y_d = \begin{cases} 4 & \text{"wrong name"} \\ 3 & \text{"insufficient voltage variation"} \\ 2 & \text{"voltage variation outside range"} \\ 1 & \text{"healthy"} \\ -1 & \text{"faulty".} \end{cases}$$

Thereby, a "faulty" label is marked with a negative number, due to its harmful impact.

4.1.2 Existing Analysing Methods

The Quench Heater Discharge Analysis (QHDA) tool is an existing QH analysis module, which is part of the PMA Framework (see Section 1.3). It is executed in LabVIEW, which is a graphical development environment for integrating measurement hardware and algorithms for data analysis and for building a user-specific GUI [32]. The goal is to find abnormalities in the QH discharge signal which could be a sign of existing or upcoming QH damages. The existing workflow is summarized in Figure 4.4.

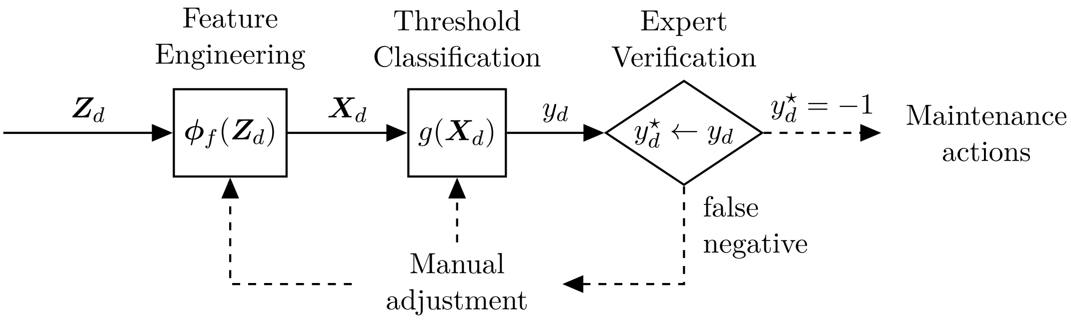


Figure 4.4: Current workflow of the QHDA tool.

First, the time discrete signals $\boldsymbol{Z}_d$, provided by the PM database, are analyzed by extracting features $\boldsymbol{X}_d$ which characterize the QH properties. In particular, the QHDA validates the QH discharges with the following features [29]:

1. *Steady state voltage level:* Calculation of the initial and final value of the voltage U_{HDS}.
2. *Characteristic time of the pseudo-exponential decay:* The characteristic time of the exponential decay is calculated for both the current signal and the reference signal.
3. *Steady state resistance level:* The initial resistance of the QH strip is computed.
4. *Signal comparison:* The voltage, current, and resistance signals are compared pointwise to the corresponding reference signals.

With these features, the QHDA tool then makes a statement if the current QH condition is "healthy" ($y_d = 1$) or "faulty" ($y_d = -1$). The classification uses a threshold-based approach:

$$y_d = g(\boldsymbol{X}_d) = \begin{cases} 1 & \text{if } \check{\boldsymbol{K}} < \boldsymbol{X}_d < \hat{\boldsymbol{K}} \\ -1 & \text{otherwise,} \end{cases} \tag{4.2}$$

in which $\check{\boldsymbol{K}}$ is the minimum threshold vector and $\hat{\boldsymbol{K}}$ is the maximum threshold vector, which are both determined by the know-how of experts and the experience from past QH analyses.

In case the QH condition is rated "faulty", experts have to verify this result. The automatically labeled event therefore becomes a reference label $y_d^\star$, i.e. 1 if $y_d = 1$ and -1 if the expert verifies that $y_d = -1$. In this thesis, a "healthy" QH is defined to be positive and a "faulty" QH is defined to be negative. Consequently, if the experts decide that the prediction of the classification algorithm was true negative ($y_d^\star = -1$), they can initiate further actions, like the activation of the redundant LF QH circuits. Nevertheless, if the experts decide that the automatic classification of the QH condition does not reflect the actual QH condition (false negative), the machine operation continues as usual. Furthermore, the experts then adjust the classification algorithm, such that this specific "faulty" classification does not occur in the future any more. However, since the experts only get notified in case of a "faulty" classification, they only intervene if the classification was false negative (no QH damage). A false positive error of the classification algorithm only emerges if the QH damage is fatal and triggers any machine protection systems.

4.1.3 Goals

The goal is to first analyze the QH discharges in the environment of the LHC signal monitoring project by using the same features and thresholds as in the existing QHDA tool. The results of both calculations should be cross-checked in order to examine possible enhancements. By using the classification thresholds from the QHDA tool, it should then be possible to classify past QH discharges with a similar result as in the QHDA tool. Any divergence should be further analyzed with experts. Once the QHDA tool is implemented, it should be investigated if there are further features which can describe the QH characteristic with additional accuracy. If so, the effect of those features on the QH discharge classification should be verified. The thresholds of the current QH discharge classification require manual interventions from experts in case there are any inaccuracies in the classification. Therefore, it should be analyzed whether common machine learning classifiers can use the data from the past in order to incorporate expert decisions into the classification process automatically. Finally, it should be examined whether it is possible to predict any changes in the QH signals by learning from past QH signal changes.

4.2 Acquisition

In case of an event during operation or HWC test, the signals $\boldsymbol{U}_d$ and $\boldsymbol{I}_d$ are stored in the PM database (see Section 1.3) with a variable resolution and sampling frequency. During the Run 1 of the LHC, from 2008-2012, only the QH voltage was stored with a resolution of 300 mV and a sampling frequency of 125 Hz - 500 Hz depending on the circuit. This measurement then got improved for the main dipole circuit in order to enable enhanced QH diagnostics for LHC Run 2 from 2015-2018. In addition to the QH voltage, also a measurement for the QH current has been implemented. Their resolution is 20 mV, and 2 mA, respectively with sampling frequency up to 192 kHz.[10] [29]

Several steps of acquisition, described in Section 2.1, are necessary to extract features from those events, which will then be used for further stages of analysis. These steps are first developed and tested on one signal from a single event before they get applied on a broader scale to a certain period of machine operation [17].

4.2.1 Input for Data Query

Each event is exactly defined once the location of the event (magnet) and the corresponding PM timestamp is given. Thereby, a PM timestamp, refers to the first data point in the time array $\boldsymbol{t}$. The LHC signal monitoring API will be used together with the LHC signal monitoring metadata to load the data from the PM database (see Section 2.1).

4.2.2 Search Events

With the LHC signal monitoring API it is possible to search for all quench events during a certain period. In order to apply the algorithms on a broad scale, the calculations are ran over the period of 2014-2018 for the main dipole magnets. This allows a more accurate characteristic of the QH condition with features than before 2014, due to the enhanced measurements in the dipole magnets during this period. Furthermore, the QH signals have been stored differently before 2014 and the queries from LHC signal monitoring are not yet supported by the PM REST API. Even though the LHC was in the first long shutdown from 2013-2014, there were HWC tests at the end of this period, which could also be interesting for analysis. In total there are $D = 30150$ QH events in the PM database for which the unprocessed signals $\boldsymbol{U}_d^\circ$ are queried.

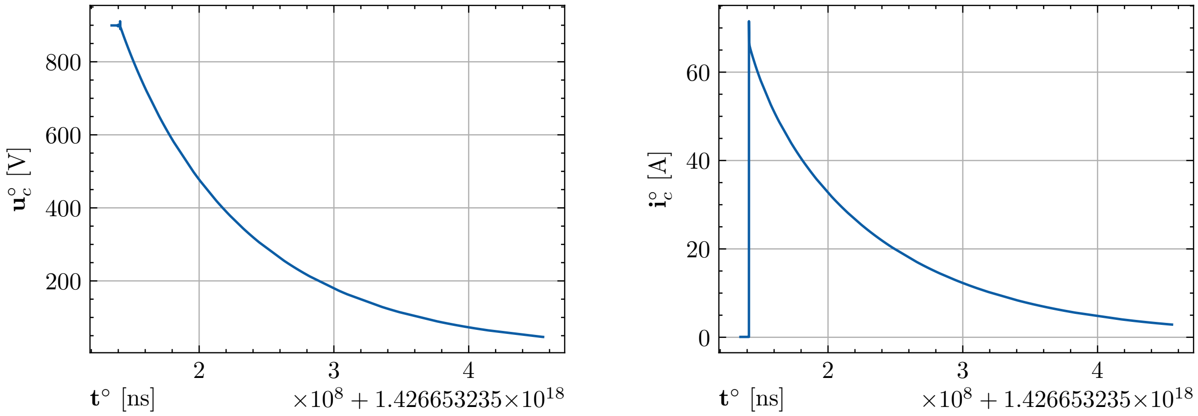
4.2.3 Filter Events

During operation and the HWC tests, sometimes data is stored into the PM database with a different signal name for test purposes. Accordingly, the data entries are rejected in case the name of the data entry does not match with the expected magnet name (comparison with regular expression). Furthermore, a spurious trigger of the PMA or a different machine setup can lead to data entries that cannot be used to make a statement about the current QH condition. Therefore, all events which are not eligible for further processing are filtered out depending on the name, the minimum value $\check{u}_c$, and the maximum value $\hat{u}_c$ of the signal $\boldsymbol{U}_d$ (see Equation 4.18 and 4.19). In addition, a class label y_d, which is introduced in Section 4.1.1, is assigned to each event $d \in [1, ..., D]$:

$$y_d(\boldsymbol{U}_d) = \begin{cases} 4 & \text{if name does not match pattern} \\ 3 & \text{if } 20V > \hat{u}_c - \check{u}_c \quad \forall c \in [1, ..., C] \\ 2 & \text{if } 15V > \check{u}_c > 75V \quad \forall c \in [1, ..., C] \\ 2 & \text{if } 780V > \hat{u}_c > 980V \quad \forall c \in [1, ..., C] \\ \text{continue to pre-processing} & \text{otherwise.} \end{cases} \tag{4.3}$$

4.2.4 Pre-processing

After the filtering of the events, also the unprocessed current $\boldsymbol{I}_d^\circ$ is queried. However, the signals have a variable sampling rate, decay start time, amplitude, and contain overlapping white noise and spikes (see Figure 4.5). Thus, it is necessary to pre-process the signals in order to make them comparable (the pre-processed signals can be seen in Figure 4.3).



(a) Unprocessed QH voltage discharge curve as a function of the unprocessed time.

(b) Unprocessed QH current discharge curve as a function of the unprocessed time.

Figure 4.5: Unprocessed QH discharge signals.

1. *Time synchronization*
   During the feature extraction, each event is treated individually. Therefore, the start time is set to zero by subtracting each element of the QH signal array with the first element of the array:

$$\boldsymbol{t}^{\text{sync}} = \boldsymbol{t}^\circ - t_1^\circ. \tag{4.4}$$

2. *Resampling [33]*
A typical QH signal has $M = 16384$ data points and is $T = 320036808$ ns long. However, the sampling time of the QH signal, stored in the PM database, is not constant. This can be seen in Figure 4.6, where $n \in [1, ..., N]$ is plotted next to the corresponding timestamp t_n.[6]

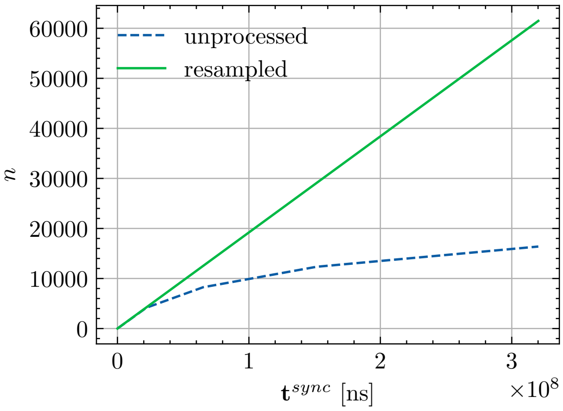


Figure 4.6: Comparison of time distribution.

Since most of the algorithms for feature engineering require discrete signals with fixed sampling frequency, the signals are resampled with the highest resolution ($T^{\mathrm{s}} = 5208$ ns). Therefore, the signal z_m is first extended to z_n^{gap} by setting the missing values to zero:

$$z_n^{\mathrm{gap}} = \begin{cases} z_{mK} & \text{if } n \in mK \\ 0 & \text{otherwise,} \end{cases} \tag{4.5}$$

for $n \in [1, ..., N]$ and $m \in [1, ..., M]$, where $K \in \mathbb{N}$ is the magnification factor, M is the amount of sample points of the unprocessed signal and $N = T/T^{\mathrm{s}}$ is the amount of sample points of the resampled signal. Considering a typical QH signal, this leads to $N = 320036808/5208 = 61451$ data points. In order to interpolate the missing values of the new data points $z_n^{\mathrm{rs}} \in \boldsymbol{z}_c^{\mathrm{rs}}$, linear interpolation is used [33]:

$$z_n^{\mathrm{rs}} = \sum_{i=n-(K-1)}^{n+(K-1)} z_i^{\mathrm{gap}} h_{n-i}, \tag{4.6}$$

with h as a triangularly shaped impulse response[7]:

$$h_{n-i} = \begin{cases} 1 - \frac{|n-i|}{K} & \text{if } |n-i| \leq K \\ 0 & \text{otherwise.} \end{cases} \tag{4.7}$$

3. *Time conversion*
Each data point stored in the PM database is precisely defined by an integer timestamp in unix time (see. Table 2.1). Unix time is the amount of nanoseconds passed after "1970-01-01 00:00:00.000000 (GMT+1)", which is why the numbers during the period from 2014-

[6] In fact, each signal is divided in four sections, in which the sampling time is doubling w.r.t. the previous one. While the sampling time of the signals is 5208 ns in the first section, it is doubled three times, leading to a sampling time of 41667 ns in the last section of the signal.

[7] Note that linear interpolation, which is described in this thesis, is the simplest form of interpolation. However, the interpolation methods used in the corresponding programs do not necessarily always use linear interpolation and might have a more complex triangularly shaped impulse response h.

2018 have 19 digits. Due to the high numbers, the visual representation of the signals can be confusing. Accordingly, the LHC signal monitoring API can be used to convert the time into a human-readable string date or into unix time in seconds:[8]

$$\boldsymbol{t} = 10^9 \boldsymbol{t}^{\mathrm{sync}}. \tag{4.8}$$

4. *Filtering*
   Spikes and noise can be filtered out with several denoising filtering methods. In this thesis two of them are used in particular. Figure 4.7 compares how these two methods work on a signal with noise and spikes.

   a) *Low-pass filter [33], [35], [36]*
   The goal of the low-pass filter is to eliminate high frequency harmonic components from a given time continuous signal. The ideal low-pass characteristic, therefore, is defined as:

$$H^{\mathrm{ideal}}(\omega) = \begin{cases} 1 & \text{if } |\omega| < \Omega^{\mathrm{c}} \\ 0 & \text{otherwise,} \end{cases} \tag{4.9}$$

   where $H(\omega)$ is the transfer function, ω is the angular frequency, and Ω^{c} is the angular cutoff frequency.

   In order to approximate this ideal low-pass behavior, the Butterworth approximation is used in this thesis. The square magnitude of the transfer function is given as follows:

$$|H^{\mathrm{but}}(\omega)|^2 = \frac{1}{1 + (\frac{\omega}{\Omega^{\mathrm{c}}})^{2K}}, \tag{4.10}$$

   with K as the order of the filter. Accordingly, the filter gain decreases by $20 \times K$ dB/decade for high frequencies.

   By using the identity $s = j\omega$ and the bilinear transformation:

$$s = \frac{2}{T^{\mathrm{s}}} \frac{z-1}{z+1}, \tag{4.11}$$

   with T^{s} as the sampling time of the signal and z as the delay operator, the transfer function can be converted into the discrete-time domain. This results in a transfer function of the form:

$$H^{\mathrm{but}} = \frac{\sum_{k=0}^{K} b_k z^{-k}}{\sum_{k=1}^{K} a_k z^{-k}}. \tag{4.12}$$

   The values of the coefficients b_k and a_k, thereby, depend on the order K of the filter, the sampling time T^{s}, and the cutoff frequency Ω^{c}. This transfer function can then

[8] It is important to keep in mind that during the conversion from nanoseconds into seconds the 64 bit integer timestamp is converted into a 64 bit float timestamp. A 64 bit integer uses one bit for the sign and 63 bits for the mantissa which allows the storage of numbers from -9223372036854775808 to 9223372036854775807. A 64 bit float on the other hand uses one bit for the sign, 52 bits for the mantissa, and 11 bits for the exponent (e.g. -9.223372036854776e+18 to 9.223372036854776e+18) [34]. Accordingly, a 64 bit float can store much higher numbers but it necessarily has to round a 19 digit number on its last digits. A conversion from a unix timestamp in seconds, back to a unix timestamp in nanoseconds is therefore not eligible without the prior time synchronisation.

be applied to a time discrete signal z_n by using:

$$z_n^{\mathrm{lp}} = \sum_{k=0}^{K} b_k z_{n-k} - \sum_{k=1}^{K} a_k z_{n-k}^{\mathrm{lp}}. \tag{4.13}$$

Furthermore, the function $\boldsymbol{\phi}_{lp}(\boldsymbol{z})$ is defined to calculate all $z_n^{\mathrm{lp}} \in \boldsymbol{z}^{\mathrm{lp}}$ for $n \in [1, ..., N]$.[9]

b) *One-dimensional median filter [37]*
The one-dimensional median filter is a nonlinear signal processing technique that replaces the value of a moving window in a sequence with the median value of the window. Unlike the low-pass filter it preserves sharp edges in a signal and is, therefore, especially suited for the smoothing of high-frequency noise.

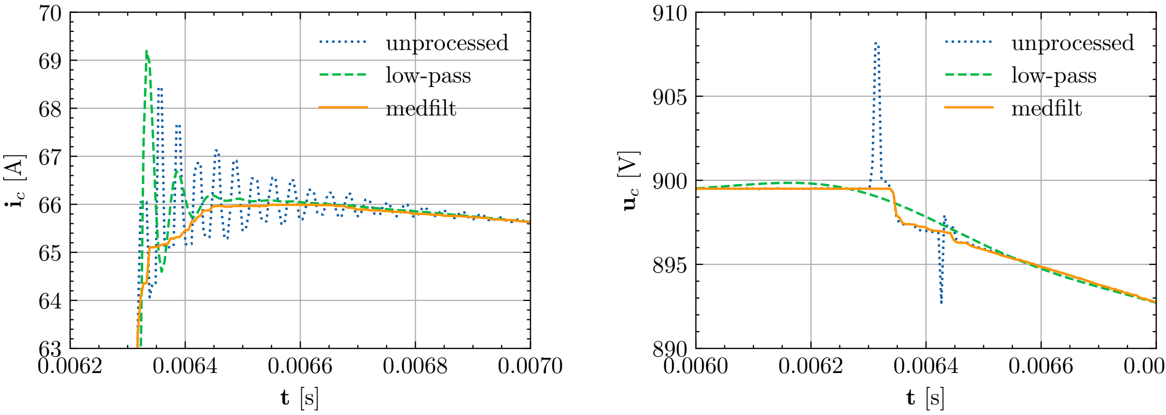
Considering a sorted sequence z_n for $n \in [1, ..., N]$ with N as the number of odd data points, the median value $\mathrm{med}(z_1, ..., z_N)$ is defined as the middle value $z_{\left(\frac{N+1}{2}\right)}$ of the sequence. A median filter of the odd window size K^{w} applied on z_n, therefore, is notated as:

$$z_n^{\mathrm{med}} = \mathrm{med}\left(z_{\left(n-\frac{K^{\mathrm{w}}-1}{2}\right)}, ..., z_{\left(n+\frac{K^{\mathrm{w}}-1}{2}\right)}\right). \tag{4.14}$$

Again, the function $\boldsymbol{\phi}_{\mathrm{med}}(\boldsymbol{z}, K^{\mathrm{w}})$ is further defined to calculate all $z_n^{\mathrm{med}} \in \boldsymbol{z}^{\mathrm{lp}}$ for $n \in [1, ..., N]$.

Figure 4.7 shows the comparison of the two filters, applied to the start of the exponential decay of the QH signals. One can see that the median filter is less prone to oscillations and spikes, which is why the signal $\boldsymbol{z}_c^{\mathrm{rs}}$ is further pre-processed with a median filter of the size $K^{\mathrm{w}} = 51$:

$$\boldsymbol{z}_c^{\mathrm{med}} = \boldsymbol{\phi}_{\mathrm{med}}(\boldsymbol{z}_c^{\mathrm{rs}}, K^{\mathrm{w}}). \tag{4.15}$$



(a) Filtered section of the QH current. The cutoff frequency of the low-pass is set to $f_{\mathrm{c}} = 1/10T_{\mathrm{s}}$ and the window size of the one-dimensional median filter to K=51.

(b) Filtered section of the QH voltage. The cutoff frequency of the low-pass is set to $f_{\mathrm{c}} = 1/100T_{\mathrm{s}}$ and the window size of the one-dimensional median filter to $K = 51$.

Figure 4.7: Visual representation of the applied denoising filters with given filter parameter.

[9] In practice the coefficients b_{k} and a_{k} are automatically generated the python function "scipy.signal.butter()" with the cutoff frequency $f^{\mathrm{c}} = \omega^{\mathrm{c}}/2\pi$ as an input parameter.

5. *Decay extraction*
   Within the PM buffer the start of a discharge is not always synchronised to the PM timestamp. Therefore, in order to compare the discharge profiles it is necessary to synchronise the start of a discharge. In order to compare the QH signals with each other, two different methods are used to find the start timestamp T^{d} of the QH decay:

   a) *Finding a window with a given mean value and standard deviation in the QH current.[29]*
      At the start of the QH current decay there is a transient oscillation due to the inductance in the QH heating strips (see Figure 4.7(a)). The mean value and the standard deviation of this oscillation are consistent. It is, therefore, possible to detect the last data point E of the oscillation by finding a signal of length L with a given mean value $K^{\mu} = 50$ and standard deviation $K^{\sigma} = 0.1$ (see Algorithm 1). Thereby, the samples of the decay $\boldsymbol{z}_c^{\mathrm{d1}}$ are z_e^{d1} for $e \in [E, ..., N]$ and the decay start time is t_E. While this method can find the start of the decay very accurately, it can only be used for the normal operation of the QHs, since the oscillation varies with the initial value of the QH discharge current (in fact the operating voltage). Furthermore, the QH current is only logged in the main dipole magnet which also limits the scope of application for this function.

Algorithm 1: Find the start of the decay with given mean and standard deviation

```
Result: $E$
assume input values $z_n^{\mathrm{med}}$ for $n \in [1, ..., N]$ to be known ;
assume input values $L, K^{\mu}, K^{\sigma}$ to be known ;
for $m = 1, ..., N - L$ do
    set $\mu = \boldsymbol{\phi}_{\mu}(z_m^{\mathrm{med}}, ..., z_{m+L}^{\mathrm{med}})$ with Equation (3.1) ;
    set $\sigma = \boldsymbol{\phi}_{\sigma}(z_m^{\mathrm{med}}, ..., z_{m+L}^{\mathrm{med}})$ with Equation (3.2) ;
    if $\mu \geq K^{\mu}$ and $\sigma \leq K^{\sigma}$ then
        $E \leftarrow m$;
    end
end
```

   b) *Finding the first value with a certain deviation from the initial value of the QH voltage.*
      If the function above does not return a decay start time, it is also possible to take all samples e in which the signal z_n for $n \in [1, ..., N]$ is below a certain voltage threshold:

$$e = \begin{cases} n & \text{if } \boldsymbol{\phi}_{lp}(z_n^{\mathrm{med}}) \leq \hat{z}_n^{\mathrm{med}} K^p \\ 0 & \text{otherwise.} \end{cases} \tag{4.16}$$

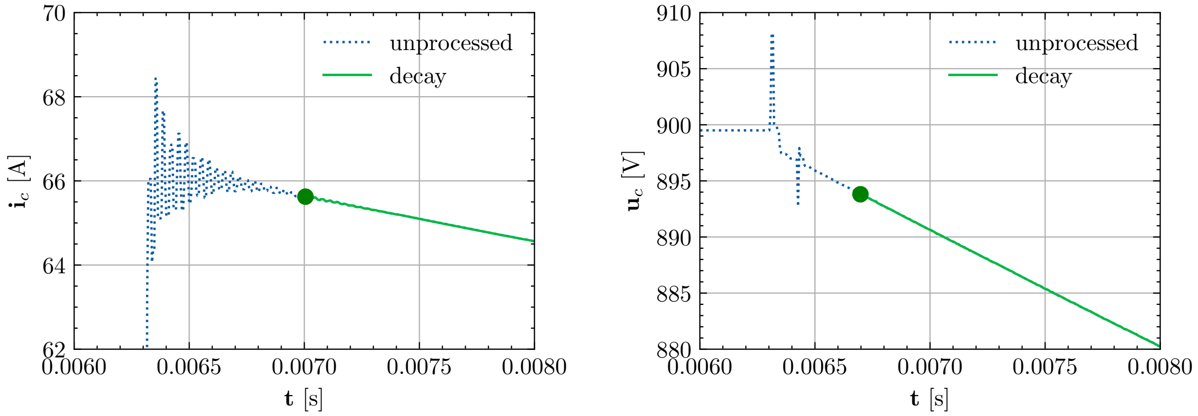
      Thereby, the voltage threshold is calculated by taking a certain percentage $K^p = 0.98$ of the initial value $\hat{z}_n^{\mathrm{med}}$. The samples of the decay $\boldsymbol{z}_c^{\mathrm{d2}}$ then, again, are z_e^{d2} for $e \in [E, ..., N]$ and the decay start time is t_E. This approach is much more general and can, therefore, be used for all kinds of exponential signal analysis. However, in Figure 4.8(b) it can be seen that there are overlapping spikes in the voltage, which can lead to several complications and inaccuracies.

      First, it is hard to find the initial value $\hat{z}_n$. This value is computed by taking the mean value of the first 20 QH voltage data points (see Equation (4.18)). A voltage spike in the beginning can, therefore, influence this value drastically. Furthermore, it is possible that a spike before the start of the decay reaches the deviation threshold and is classified as the beginning of the decay by mistake. By filtering the signal with the denoising methods described in the previous pre-processing step (low-pass and median filter), it is possible to smooth out the signal and get rid of the spikes in it. Nevertheless, if the spike takes place over several data points the impact of the spike can only be reduced by the filter and not completely eliminated. The accuracy of this decay start time calculation is therefore not as high as the previous one. For this reason, it is only executed in case the previous one cannot find the start of the decay.

Figure 4.8 shows the comparison of the both methods to find the start of the decay. The first approach proved to be more accurate, the second one, however, is more stable. Thus, the signal $\boldsymbol{z}_c^{\text{med}}$ is further pre-processed, by combining both methods, as stated below:

$$\boldsymbol{z}_c = \begin{cases} \boldsymbol{z}_c^{\text{d1}} & \text{if decay found} \\ \boldsymbol{z}_c^{\text{d2}} & \text{otherwise,} \end{cases} \tag{4.17}$$

where $\boldsymbol{z}_c$ is the pre-processed signal, used for feature engineering.



(a) Finding a window with a given mean value and standard deviation in the QH current

(b) Finding the first value with a certain deviation from the initial value of the QH voltage.

Figure 4.8: Visual representation of methods for finding the start of decay. The green dot represents the decay start time t_E

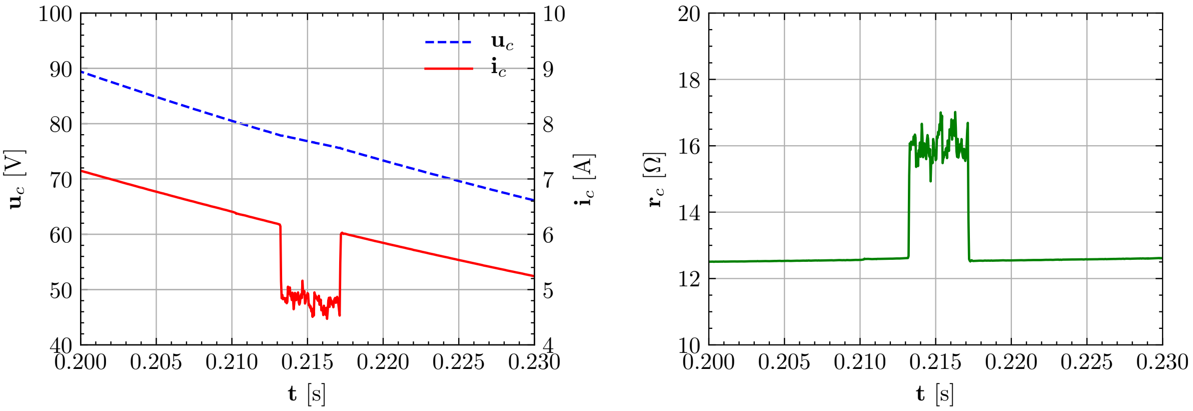
4.2.5 Feature Engineering

Quench Heater Fault Patterns

After pre-processing, the QH resistance $\boldsymbol{R}_d$ can be calculated with $\boldsymbol{U}_d$ and $\boldsymbol{I}_d$ (see Section 4.1.1) for $d \in [1, ..., D]$. A change in the QH hardware condition can affect those QH signals in several ways. As an example, one QH fault or precursor which occurred as a spike, can be seen in Figure 4.9. In this particular case the following actions have been taken by experts:[10]

1. Once the spike was detected, the HF QH was deactivated and the LF QH was activated instead.
2. Furthermore, the magnet was planned to be replaced during the long shutdown two.
3. A test discharge, with the LF QH, was initialized a few hours after the occurrence of the spike. This discharge was taken as a new reference signal $\boldsymbol{Z}_d^\star$.
4. Finally, the magnet was replaced during the long shutdown two.

In addition to a spike, any other deviation from a QH signal compared to the reference signal can also be an indicator for a change in the QH hardware condition (e.g. change in the characteristic time of the pseudo-exponential decay).



(a) Measured QH voltage and current discharge.

(b) Calculated resistance change.

Figure 4.9: QH fault or precursor, which occurred in the dipole magnet A26R8 (HDS3) on "2015-09-07 14:59:39.113000 (GMT+1)"

Quench Heater Characteristic Features

The QH characteristic features should describe the QH hardware condition, such that it is easy to find any QH fault patterns, mentioned in the last section. Therefore the following features are extracted.

1. *Initial and final value [29]*
   The initial and the final values of a QH provide partial information about the LHC operation (see Section 4.2.3) and make it possible to identify events with no QH discharge (see Section 4.2.3). They are calculated for the decay of the median filtered ($K^{\mathrm{w}} = 3$) voltage,

[10] Information from Z. Charifoulline, private communication

the current, and the resistance by taking the mean value of the first/last $K^{\text{init}} = 20$ data points:

$$x_1^z = \boldsymbol{\phi}_1(\boldsymbol{z}) = \boldsymbol{\phi}_\mu(\boldsymbol{\phi}_{med}(z_1, ..., z_{K^{\text{init}}}, \text{K}^{\text{w}})), \tag{4.18}$$

$$x_2^z = \boldsymbol{\phi}_2(\boldsymbol{z}) = \boldsymbol{\phi}_\mu(\boldsymbol{\phi}_{med}(z_{N-K^{\text{init}}}, ..., z_N, K^{\text{w}})). \tag{4.19}$$

2. *Characteristic time of the pseudo-exponential decay [29]*
In a linear RC circuit, the characteristic time of the pseudo-exponential decay $\tilde{\boldsymbol{\tau}}$ is equal to the time constant $\tau = RC$. However, as the resistance $\boldsymbol{r}_c$ of the heater strip is time-dependent due to the changing temperate level, also the $\tilde{\boldsymbol{\tau}}$ is variable. However, in order to minimize the number of features, the characteristic time $\tilde{\boldsymbol{\tau}}$ is approximated with a time constant τ.[11] This leads to the general form of an exponential decay:

$$\boldsymbol{z} = z_1 e^{-\frac{\boldsymbol{t}}{\tau}}, \tag{4.20}$$

assuming the data points z_n with the corresponding time t_n for $n \in [1, ..., N]$ and the initial value z_1 is known.

Furthermore, it is important to mention that the QH current cannot rise instantaneously, as the QH strips have non-zero inductance. However, as the rise time of the current is about thousand times smaller than the decay time of the current, the exponential rise of the current was not further considered in this thesis.

a) *Charge Approach[12] [29]*
Considering a time continuous signal:

$$f(t) = f_0 e^{-\frac{t}{\tau}}, \tag{4.21}$$

by integrating Equation (4.20) within $\boldsymbol{t} \in [t_1, ..., t_N]$ the function can be written as [29]:

$$\int_{\theta_0}^{\theta_1} f(t)d\theta = \int_{\theta_0}^{\theta_1} f_0 e^{-t/\tau} d\theta = -\tau f_0 e^{-t/\tau}\Big|_{\theta_0}^{\theta_1} = -\tau f_0 e^{-\theta_1/\tau} - \left(-\tau f_0 e^{-\theta_0/\tau}\right). \tag{4.22}$$

Since $f_0 e^{-t_1/\tau} = f(t_1)$ and $f_0 e^{-t_0/\tau} = f(t_0)$, this equation can be simplified:

$$\int_{\theta_0}^{\theta_1} f(t)d\theta = \tau(f(t_0) - f(t_1)). \tag{4.23}$$

Consequently, the time constant of an exponential decay $f(t)$, calculated with the charge approach, can be determined as:

$$\tau = \frac{\int_{\theta_0}^{\theta_1} f(t)d\theta}{f(t_0) - f(t_1)}. \tag{4.24}$$

By using the trapezoidal rule [33]:

$$\int_{\theta_1}^{\theta_N} f(t)dt \approx \sum_{n=1}^{N} \frac{z_n + z_{n-1}}{2}(t_n - t_{n-1}), \tag{4.25}$$

[11] Considering the nominal values of the main dipole circuit (see Table 1.2), the time constant has estimated to be $\tau = RC = 11\ \text{m}\Omega \cdot 7.05\ \text{mF} = 78$ ms before the actual calculation.

[12] It is named "Charge Approach", as the voltage, which is proportional to the charge in a capacitor, is integrated.

Equation (4.24) can be transformed into the time-discrete domain leading to:

$$x_3^{\mathrm{z}} = \boldsymbol{\phi}_3(\boldsymbol{z}) = \frac{\sum_{n=1}^{N} \frac{z_n + z_{n-1}}{2}(t_n - t_{n-1})}{z_0 - z_N}. \tag{4.26}$$

b) *Energy Approach*[13]*[29]*
The time integral of the squared Equation (4.21) for $t \in [t_0, t_1]$ is given as:

$$\int_{\theta_0}^{\theta_1} f^2(t) d\theta = \int_{\theta_0}^{\theta_1} f_0^2 e^{-2t/\tau} d\theta = -\frac{\tau}{2} f_0^2 e^{-2t/\tau}\Big|_{\theta_0}^{\theta_1} = -\frac{\tau}{2} f_0^2 e^{-2\theta_1/\tau} - \left(-\frac{\tau}{2} f_0^2 e^{-2\theta_0/\tau}\right). \tag{4.27}$$

As in the charge approach, this equation can be simplified by substituting $f_0^2 e^{-2t_1/\tau}$ with $f^2(t_1)$ and $f_0^2 e^{-2t_0/\tau}$ with $f^2(t_0)$. After rearranging the equation, the time constant of an exponential decay $f(t)$, calculated with the energy approach, is given as:

$$\tau = 2\frac{\int_{\theta_0}^{\theta_1} f^2(t) d\theta}{f^2(t_0) - f^2(t_1)}. \tag{4.28}$$

Again, the trapezoidal rule from Equation (4.25) can be used to transform Equation (4.28) into the time-discrete domain:

$$x_4^{\mathrm{z}} = \boldsymbol{\phi}_4(\boldsymbol{z}) = \frac{2\sum_{n=1}^{N} \left(\frac{z_n + z_{n-1}}{2}(t_n - t_{n-1})\right)^2}{z_0^2 - z_N^2}. \tag{4.29}$$

c) *Linear Regression*
By taking the natural logarithm of the exponential decay $\boldsymbol{z}$, Equation (4.20) can be rewritten as:

$$\ln(\boldsymbol{z}) = \ln(z_1) - \frac{t}{\tau}. \tag{4.30}$$

Given a signal z_n with the corresponding time t_n for $n \in [1, ..., N]$ the goal is to minimize the squared error of the hypothesis:

$$h_{\boldsymbol{\theta}}(t_n) = \theta_0 + \theta_1 t_n, \tag{4.31}$$

and the output z_n, by optimizing the parameters $\boldsymbol{\theta} = (\theta_0, \theta_1)$. This leads to the following optimization problem:

$$\min_{\boldsymbol{\theta}} \left(\frac{1}{N}\sum_{i=1}^{N} (h_{\boldsymbol{\theta}}(t_n) - z_n)^2\right). \tag{4.32}$$

After comparing the coefficients from Equation (4.30) with the hypothesis in Equation (4.31), it is possible to see that τ corresponds to:

$$x_5^{\mathrm{z}} = \boldsymbol{\phi}_5(\boldsymbol{z}) = -1/\theta_1. \tag{4.33}$$

d) *Exponential Fit*
The optimization problem in Equation (4.32) can be further applied on the parameters

[13] It is named "Energy Approach", as the squared voltage, which is proportional to the energy in a capacitor, is integrated.

$\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2)$ and the hypothesis:

$$h_{\boldsymbol{\theta}}(z_n) = \theta_0 \mathrm{e}^{\theta_1 (z_n - \theta_2)}. \tag{4.34}$$

By comparing the coefficients of the exponential decay $\boldsymbol{z}$ from Equation (4.20) it is again possible to see that:

$$x_6^{\mathrm{z}} = \boldsymbol{\phi}_6(\boldsymbol{z}) = -1/\theta_1. \tag{4.35}$$

e) *Mean Value of Characteristic Time of the Pseudo-Exponential Decay*
Another way to calculate the time constant of an exponential decay, is by dividing the decay with its derivative. Considering a pseudo-exponential decay, this results in a vector with the characteristic time $\tilde{\boldsymbol{\tau}}$:

$$\frac{\boldsymbol{z}}{\boldsymbol{z}'} = -\tilde{\boldsymbol{\tau}} \frac{z_0 e^{-\frac{t}{\tilde{\boldsymbol{\tau}}}}}{z_0 e^{-\frac{t}{\tilde{\boldsymbol{\tau}}}}} \Rightarrow \tilde{\boldsymbol{\tau}} = -\frac{\boldsymbol{z}}{\boldsymbol{z}'}. \tag{4.36}$$

Thereby, the derivative of the discrete signal is defined as:

$$\boldsymbol{z}' \triangleq \frac{z_n + z_{n-1}}{t_n - t_{n-1}} \quad \forall n. \tag{4.37}$$

By assuming normally distributed sample points of the characteristic time of the pseudo-exponential decay (see Equation (4.20)) $\tilde{\boldsymbol{\tau}} \sim \mathcal{N}(\mu, \sigma^2)$ it is possible to derive an estimation of a time constant. In addition, the standard deviation σ can be taken as a feature for the variation of the data points:

$$x_7^{\mathrm{z}} = \boldsymbol{\phi}_7(\boldsymbol{z}) = \boldsymbol{\phi}_\mu(\tilde{\boldsymbol{\tau}}), \tag{4.38}$$

$$x_8^{\mathrm{z}} = \boldsymbol{\phi}_8(\boldsymbol{z}) = \boldsymbol{\phi}_\sigma(\tilde{\boldsymbol{\tau}}). \tag{4.39}$$

Table 4.1 shows the characteristics of the different approaches for the approximation of the time constant. Thereby, the mean values where derived from all features $\boldsymbol{X}$ within D events from 2014-2018. One can see that the calculation time is higher for both methods which contain optimization functions (i.e. $\boldsymbol{x}_5^{\mathrm{z}}$ and $\boldsymbol{x}_6^{\mathrm{z}}$).

Feature	Description	Average Calculation Time	Average Value $\boldsymbol{\phi}_{\mu}(\boldsymbol{X})$	Unit
$\boldsymbol{x}_3^{\mathrm{z}}$	Charge Approach	10	80.7	ms
$\boldsymbol{x}_4^{\mathrm{z}}$	Energy Approach	10	79.2	ms
$\boldsymbol{x}_5^{\mathrm{z}}$	Linear Regression	50	96.7	ms
$\boldsymbol{x}_6^{\mathrm{z}}$	Exponential Fit	330	89.5	ms
$\boldsymbol{x}_7^{\mathrm{z}}$	Mean Value of CT	10	93.9	ms

Table 4.1: Comparison of the different approaches for the approximation of the time constant, considering all features from D events.

Figure 4.10 further varifies the results from Table 4.1 as it shows the error made by the assumption of a constant characteristic time of the pseudo-exponential decay of the QH voltage. This error is calculated with a typical QH signal by:[10]

$$\boldsymbol{u}^{\mathrm{e}} = \boldsymbol{u} - u_1 e^{-\frac{\boldsymbol{t}}{\tau}}, \tag{4.40}$$

One can see that the error of the energy approach is lower in the beginning and higher in the end while it is the other way around with the linear regression approach. Consequently, both features can characterise the QH condition. For example, in case $\boldsymbol{x}_4^{\mathrm{z}}$ rises while $\boldsymbol{x}_6^{\mathrm{z}}$ stays the same, it is a indication that the characteristic time of the QH signals changed in the beginning.

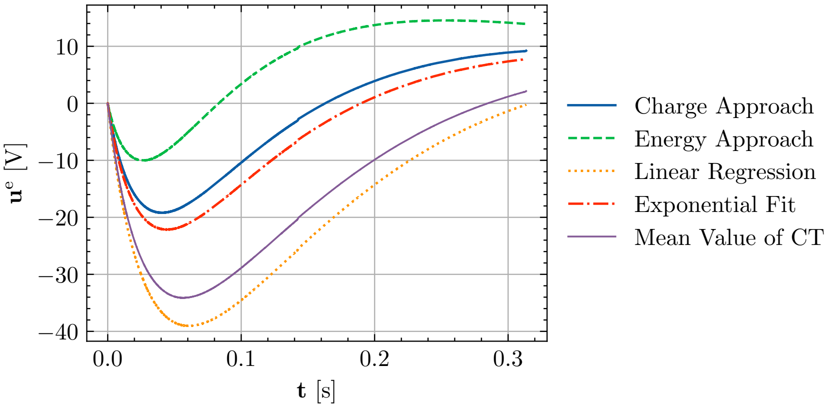


Figure 4.10: Error made by the different approaches, due to the assumption of a constant Characteristic Time (CT) of the pseudo-exponential decay.

3. *Similarity matrix*
Two data-sets $\boldsymbol{x} = (x_1, ..., x_N)^{\mathrm{T}}$ and $\boldsymbol{y} = (y_1, ..., y_N)^{\mathrm{T}}$ of equal length N can be compared by calculating the Euclidean distance:

$$s = \|\boldsymbol{x} - \boldsymbol{y}\| = \sqrt{\sum_{n=1}^{N} (x_n - y_n)^2}. \tag{4.41}$$

As discussed in the previous section, the initial and final values of the QH signals can change due to several reasons. In order to make the signals comparable, independent from any offset, it is necessary to normalize each signal separately by using min/max normalization:

$$\boldsymbol{z}^{\diamond} = \boldsymbol{\phi}_{norm}(\boldsymbol{z}) = \frac{\boldsymbol{z} - \check{z}_n}{\hat{z}_n - \check{z}_n}, \tag{4.42}$$

where $\check{z}_n$ is the minimum value of the signal and $\hat{z}_n$ is the maximum value of the signal. The following similarity matrix results by comparing the $C = 4$ QHs signals with this approach:

$$\boldsymbol{S} = \begin{bmatrix} s_0 & s_1 & s_2 & s_3 \\ s_4 & s_5 & s_6 & s_7 \\ s_8 & s_9 & s_{10} & s_{11} \\ s_{12} & s_{13} & s_{14} & s_{15} \end{bmatrix} = \begin{bmatrix} 0 & \|\boldsymbol{z}_1^{\diamond} - \boldsymbol{z}_2^{\diamond}\| & \|\boldsymbol{z}_1^{\diamond} - \boldsymbol{z}_3^{\diamond}\| & \|\boldsymbol{z}_1^{\diamond} - \boldsymbol{z}_4^{\diamond}\| \\ \|\boldsymbol{z}_2^{\diamond} - \boldsymbol{z}_1^{\diamond}\| & 0 & \|\boldsymbol{z}_2^{\diamond} - \boldsymbol{z}_3^{\diamond}\| & \|\boldsymbol{z}_2^{\diamond} - \boldsymbol{z}_4^{\diamond}\| \\ \|\boldsymbol{z}_3^{\diamond} - \boldsymbol{z}_1^{\diamond}\| & \|\boldsymbol{z}_3^{\diamond} - \boldsymbol{z}_2^{\diamond}\| & 0 & \|\boldsymbol{z}_3^{\diamond} - \boldsymbol{z}_4^{\diamond}\| \\ \|\boldsymbol{z}_4^{\diamond} - \boldsymbol{z}_1^{\diamond}\| & \|\boldsymbol{z}_4^{\diamond} - \boldsymbol{z}_2^{\diamond}\| & \|\boldsymbol{z}_4^{\diamond} - \boldsymbol{z}_3^{\diamond}\| & 0 \end{bmatrix}. \tag{4.43}$$

Since this matrix is symmetrical $\boldsymbol{S} = \boldsymbol{S}^{\mathrm{T}}$, only the six values of the upper triangular matrix are taken as features.

$$\boldsymbol{\phi}_9(\boldsymbol{Z}) = (s_1, s_2, s_3, s_6, s_7, s_{11})^{\mathrm{T}}. \tag{4.44}$$

4. *Curve to curve comparison [29]*
In order to limit the deviation of a signal $\boldsymbol{z}$ from a reference signal $\boldsymbol{z}^{\star}$, two signals are subtracted and a time-dependent threshold $\boldsymbol{k}^{\mathrm{th}}$ is set which the subtracted signals cannot exceed:

$$w_n = \begin{cases} 1 & \text{if } |z_n^{\diamond} - z_n^{\star\diamond}| \geq k_n^{\mathrm{th}}, \\ 0 & \text{otherwise}, \end{cases} \tag{4.45}$$

where $z_n^{\diamond}$ is the normalized signal for $n \in [1, ..., N]$. Since each QH signal has a corresponding reference signal, this approach is applied to the voltage, the current, and the resistance. The threshold function is, thereby, chosen as an exponential decay:

$$\boldsymbol{k}^{\mathrm{th}} = K^{\mathrm{th}} \mathrm{e}^{-\boldsymbol{t}/2\boldsymbol{\phi}_4(\boldsymbol{z}^{\star})}. \tag{4.46}$$

The initial value of the exponential decay K^{th} is chosen depending on the QH signal (see Table 4.2). Finally the curve to curve comparison feature is calculated by:

$$\boldsymbol{\phi}_{c2c}(\boldsymbol{z}, \boldsymbol{z}^{\star}) = \frac{\sum_{n=1}^{N} w_n}{N}. \tag{4.47}$$

Signal	Name	K^{th}	Unit
Voltage	$\boldsymbol{U}_d$	20	V
Current	$\boldsymbol{I}_d$	5	A
Resistance	$\boldsymbol{R}_d$	0.5	Ω

Table 4.2: Initial value of the exponential decay K^{th} for the curve to curve comparison threshold [29].

4.3 Exploration

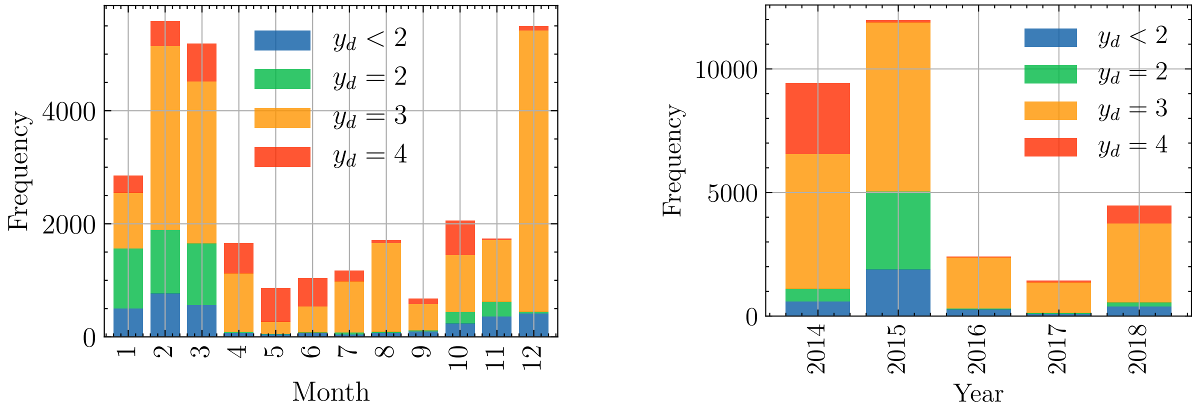
4.3.1 Event Exploration

This section is dedicated to get an overview of the available data and the related opportunities for further analysis and prediction. Table 4.3 shows the amount of events remaining after the filtering of events in Section 4.2.3. In the filtering of events, the class labels $y_d = 2, 3, 4$ are already defined, as the margins are clearly specified and the results from the existing QHDA tool match exactly with the results presented in this thesis. In the next section, the previously calculated features will be used for the classification of class labels $y_d < 2$.

Event Characteristic	Number of Events D	Difference	Class Label y_d
All Events	30150	4087	4 - "wrong name"
Events with matching names	26063	18923	3 - "insufficient voltage variation"
Events with 20V voltage variation	7140	3894	2 - "voltage variation outside range"
Events within voltage range	3246	3246	<2 - "healthy" or "faulty"

Table 4.3: Number of QH events after applying certain criteria.

Figure 4.11 shows the time in which the events, shown in the table above, occurred. The distribution of events in these histograms can be explained with several justifications. For example one can see in Figure 4.11(b) that most of the events with $y_d = 4$ - "wrong name" happened in 2014. In this year the LHC was in the second long shutdown in which the PM data logging system was still under development.



(a) Frequency of QH events per month.

(b) Frequency of QH events per year.

Figure 4.11: Distribution of QH events over a certain time period.

By looking at Figure 4.12, one can see the distribution of QH events over the circuits. Events with $y_d = 4$ - "wrong name" cannot be linked to a specific location, as this information is provided by the name.

Figure 4.13 shows the frequency of the number of events per magnet. For $y_d < 2$ this means there are for example 275 QH magnets with one data entry, 388 QH magnets with two data entries, and so on (in total there are 1232 main dipole magnets). It is important to mention that not all events for $y_d = 3$ are plotted in this histogram as it would exceed the scope. However, there is an interesting outlier, which is magnet "B9L7", as it has 1356 data entries with a

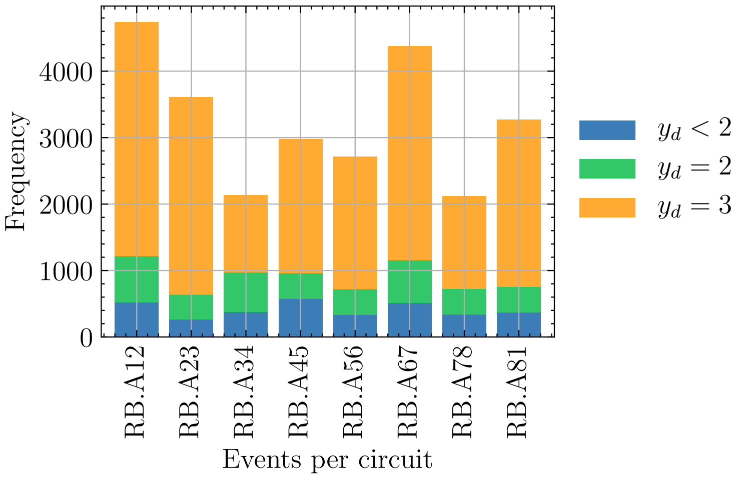


Figure 4.12: Frequency of QH events per circuit.

"voltage variation outside range". After further research it was found that those events occurred during the time period of 2014-12-17 to 2014-12-28 at about 12 minute intervals. Most likely the reason was either a faulty detection board or a faulty crate controller.[10] This histogram is particularly useful, when considering the analysis and prediction opportunities. Prediction is especially interesting for events with $y_d < 2$, as they generally occur arbitrary without any manual changes in the LHC. One can see that the average number of data entries per magnet with $y_d < 2$ is about two. For the easiest type of prediction, which is linear regression (see Equation (4.31)), at least two data points are necessary for the training of the hyper-parameters. Therefore, events of selected magnets, like the one described in Section 4.2.5, have been further examined. However, no precursors of faults could be detected in prior events from the current features so far.

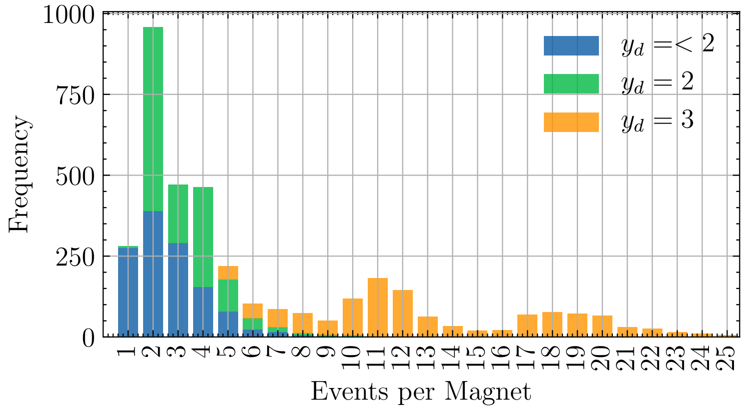


Figure 4.13: Frequency of QH events per magnet.

4.3.2 Feature Overview

Table 4.4 shows an overview of the calculated QH characteristic features. In total the feature space consists of 180 values. In order to get a better overview, those features can further be compared and compressed, as described in Section 4.1.1, leading to the matrices $\check{\boldsymbol{X}}^{\mathrm{z}}$ and $\overline{\boldsymbol{x}}^{\mathrm{z}}$. However, only specific features, further discussed in the next section, are selected for comparison and compression.

Feature $\boldsymbol{x}_f$	Description	Input Parameter Z_d	Feature Dimension L	Nr. Features per Magnet
Features $\boldsymbol{X}^{\mathrm{z}}$				
$\boldsymbol{x}_1^{\mathrm{z}} = \boldsymbol{\phi}_1(\boldsymbol{Z}_d)$	Initial Value	$\boldsymbol{U}_d, \boldsymbol{I}_d, \boldsymbol{R}_d$	4	12
$\boldsymbol{x}_2^{\mathrm{z}} = \boldsymbol{\phi}_2(\boldsymbol{Z}_d)$	Final Value	$\boldsymbol{U}_d, \boldsymbol{I}_d, \boldsymbol{R}_d$	4	12
$\boldsymbol{x}_3^{\mathrm{z}} = \boldsymbol{\phi}_3(\boldsymbol{Z}_d)$	τ Charge Approach	$\boldsymbol{U}_d, \boldsymbol{I}_d$	4	8
$\boldsymbol{x}_4^{\mathrm{z}} = \boldsymbol{\phi}_4(\boldsymbol{Z}_d)$	τ Energy Approach	$\boldsymbol{U}_d, \boldsymbol{I}_d$	4	8
$\boldsymbol{x}_5^{\mathrm{z}} = \boldsymbol{\phi}_5(\boldsymbol{Z}_d)$	τ Linear Regression	$\boldsymbol{U}_d, \boldsymbol{I}_d$	4	8
$\boldsymbol{x}_6^{\mathrm{z}} = \boldsymbol{\phi}_6(\boldsymbol{Z}_d)$	τ Exponential Fit	$\boldsymbol{U}_d, \boldsymbol{I}_d$	4	8
$\boldsymbol{x}_7^{\mathrm{z}} = \boldsymbol{\phi}_7(\boldsymbol{Z}_d)$	Mean of $\tilde{\boldsymbol{\tau}}_c$	$\boldsymbol{U}_d, \boldsymbol{I}_d$	4	8
$\boldsymbol{x}_8^{\mathrm{z}} = \boldsymbol{\phi}_8(\boldsymbol{Z}_d)$	Std of $\tilde{\boldsymbol{\tau}}_c$	$\boldsymbol{U}_d, \boldsymbol{I}_d$	4	8
$\boldsymbol{x}_9^{\mathrm{z}} = \boldsymbol{\phi}_9(\boldsymbol{Z}_d)$	Similarity Matrix	$\boldsymbol{U}_d, \boldsymbol{I}_d, \boldsymbol{R}_d$	6	18
Reference features $\boldsymbol{X}^{\mathrm{z}\star}$				
$\boldsymbol{x}_1^{\mathrm{z}\star} = \boldsymbol{\phi}_1(\boldsymbol{Z}_d^\star)$	Initial Value	$\boldsymbol{U}_d^\star, \boldsymbol{I}_d^\star, \boldsymbol{R}_d^\star$	4	12
$\boldsymbol{x}_2^{\mathrm{z}\star} = \boldsymbol{\phi}_2(\boldsymbol{Z}_d^\star)$	Final Value	$\boldsymbol{U}_d^\star, \boldsymbol{I}_d^\star, \boldsymbol{R}_d^\star$	4	12
$\boldsymbol{x}_3^{\mathrm{z}\star} = \boldsymbol{\phi}_3(\boldsymbol{Z}_d^\star)$	τ Charge Approach	$\boldsymbol{U}_d^\star, \boldsymbol{I}_d^\star$	4	8
$\boldsymbol{x}_4^{\mathrm{z}\star} = \boldsymbol{\phi}_4(\boldsymbol{Z}_d^\star)$	τ Energy Approach	$\boldsymbol{U}_d^\star, \boldsymbol{I}_d^\star$	4	8
$\boldsymbol{x}_5^{\mathrm{z}\star} = \boldsymbol{\phi}_5(\boldsymbol{Z}_d^\star)$	τ Linear Regression	$\boldsymbol{U}_d^\star, \boldsymbol{I}_d^\star$	4	8
$\boldsymbol{x}_6^{\mathrm{z}\star} = \boldsymbol{\phi}_6(\boldsymbol{Z}_d^\star)$	τ Exponential Fit	$\boldsymbol{U}_d^\star, \boldsymbol{I}_d^\star$	4	8
$\boldsymbol{x}_7^{\mathrm{z}\star} = \boldsymbol{\phi}_7(\boldsymbol{Z}_d^\star)$	Mean of $\tilde{\boldsymbol{\tau}}_c$	$\boldsymbol{U}_d^\star, \boldsymbol{I}_d^\star$	4	8
$\boldsymbol{x}_8^{\mathrm{z}\star} = \boldsymbol{\phi}_8(\boldsymbol{Z}_d^\star)$	Std of $\tilde{\boldsymbol{\tau}}_c$	$\boldsymbol{U}_d^\star, \boldsymbol{I}_d^\star$	4	8
$\boldsymbol{x}_9^{\mathrm{z}\star} = \boldsymbol{\phi}_9(\boldsymbol{Z}_d^\star)$	Similarity Matrix	$\boldsymbol{U}_d^\star, \boldsymbol{I}_d^\star, \boldsymbol{R}_d^\star$	6	18
Curve to curve comparison feature $\boldsymbol{X}^{\mathrm{z,c2c}}$				
$\boldsymbol{x}^{\mathrm{z,c2c}} = \boldsymbol{\phi}_{c2c}(\boldsymbol{Z}_d, \boldsymbol{Z}_d^\star)$	C2C comparison	$\boldsymbol{U}_d, \boldsymbol{I}_d, \boldsymbol{R}_d$ $\boldsymbol{U}_d^\star, \boldsymbol{I}_d^\star, \boldsymbol{R}_d^\star$	4	18
Total number of features				180

Table 4.4: Overview of QH characteristic features.

4.3.3 Feature Exploration

In order to gain a better overview and avoid over-fitting of a classification model, the number of features is reduced. Accordingly, redundant features are sorted out and compressed, with the goal to lose as little information as possible.

Feature Correlation

In order to detect redundant features, the Pearson correlation coefficient is evaluated for each feature [38]. Considering two data-sets $\boldsymbol{x} = (x_1, ..., x_N)^{\mathrm{T}}$ and $\boldsymbol{y} = (y_1, ..., y_N)^{\mathrm{T}}$ of equal length N, which are compared pairwise, this coefficient can be calculated by:

$$r = \frac{cov(\boldsymbol{x}, \boldsymbol{y})}{\boldsymbol{\phi}_{\sigma}(\boldsymbol{x})\boldsymbol{\phi}_{\sigma}(\boldsymbol{y})}. \tag{4.48}$$

Therefore, it is first necessary to describe the covariance [39]:

$$cov(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}\left[(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])(\boldsymbol{y} - \mathbb{E}[\boldsymbol{y}])\right], \tag{4.49}$$

where $\mathbb{E}$ denotes the expected value:

$$\mathbb{E}[\boldsymbol{x}] = \sum_{n=1}^{N} x_n p_n, \tag{4.50}$$

and where p_n is the probability with which the sample x_n occurs. Assuming two normal distributed data-sets $x_n, y_n \sim \mathcal{N}(\mu, \sigma^2)$ the expected values $\mathbb{E}[\boldsymbol{x}]$ and $\mathbb{E}[\boldsymbol{y}]$ are equal to μ. Therefore, the covariance can be calculated like [38]:

$$cov(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{N} \sum_{n=1}^{N} \left(x_n - \boldsymbol{\phi}_{\mu}(\boldsymbol{x})\right)\left(y_n - \boldsymbol{\phi}_{\mu}(\boldsymbol{y})\right). \tag{4.51}$$

The Pearson correlation coefficient is a measure for the linear dependence of the two variables. It has a value between 1 and -1, whereby the sign denotes a positive or a negative linear correlation. The closer to 0 r is the less relationship is between the two variables.

This behavior is used to sort out the redundant QH characteristic features. Therefore, r is calculated for all features leading to a Matrix $\boldsymbol{R} \in \mathbb{R}^{F \times F}$. This matrix is then analyzed and only features with low linear correlation are selected and further processed. In Figure 4.14 the Pearson correlation matrix of selected features, which are further described bellow, is illustrated.

Feature Selection an Compression

After analysis of the Pearson correlation matrix, several patterns stand out:

1. The initial value $\boldsymbol{x}_1^{\mathrm{z}}$ is linearly dependent with the final value $\boldsymbol{x}_2^{\mathrm{z}}$.

2. The different approaches for the approximation of the time constant τ are linearly dependent.

3. The same features $\boldsymbol{x}_f^{\mathrm{z}}$ within a circuit correlate with the corresponding mean value $\overline{x}_f^{\mathrm{z}}$, as the mean value is a linear operator.

Accordingly, redundancy is removed by selecting only one of the dependent features. Thereby, several aspects are taken into consideration for this selection, including the significance and the application frequency of the features in the existing QH analysis tool. Furthermore, regarding

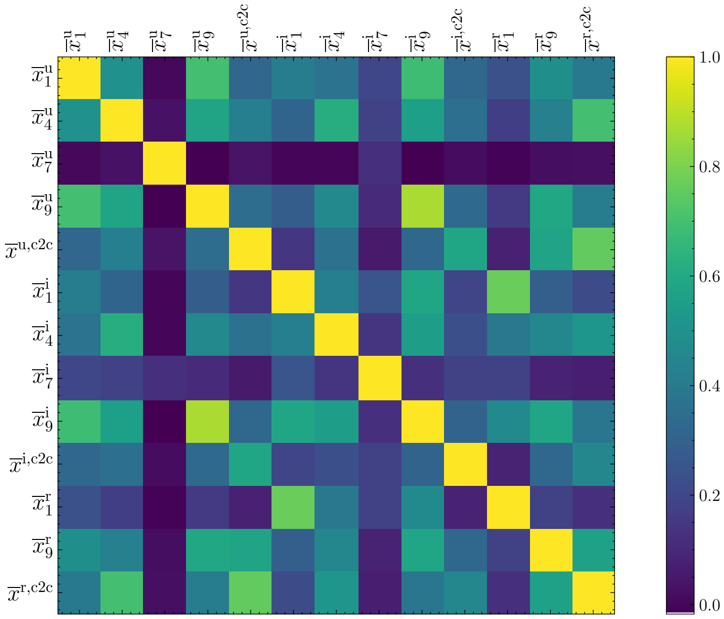


Figure 4.14: Pearson correlation of the compressed features.

the characteristic time of the exponential decay, an in-depth comparison is done in Section 4.2.3. Finally, $\boldsymbol{x}_1^{\mathrm{z}}$ and x_4^{z} is chosen for further processing.

After this selection, the compared features, shown in Table 4.5, and the compressed features, shown in Table 4.6, remain for further processing. It is important to mention that the values of the compressed features $\overline{x}^{\mathrm{z}}$ are cutted at an empirically chosen value (cap value) in order to enhance the clarity of visualization in the following plots.

Feature $\breve{\boldsymbol{x}}_f^{\mathrm{z}}$	Description	Input Parameter Z_d	Feature Dimension L	Nr. Features per Magnet
		Compared features $\breve{\boldsymbol{X}}^{\mathrm{z}}$		
$\breve{\boldsymbol{x}}_1^{\mathrm{z}} = \boldsymbol{x}_1^{\mathrm{z}} - \boldsymbol{x}_1^{\mathrm{z}\star}$	Initial Value	$\boldsymbol{U}_d, \boldsymbol{I}_d, \boldsymbol{R}_d$	4	12
$\breve{\boldsymbol{x}}_4^{\mathrm{z}} = \boldsymbol{x}_4^{\mathrm{z}} - \boldsymbol{x}_4^{\mathrm{z}\star}$	τ Energy Approach	$\boldsymbol{U}_d, \boldsymbol{I}_d$	4	8
$\breve{\boldsymbol{x}}_7^{\mathrm{z}} = \boldsymbol{x}_7^{\mathrm{z}} - \boldsymbol{x}_7^{\mathrm{z}\star}$	Mean of $\tilde{\boldsymbol{\tau}}_c$	$\boldsymbol{U}_d, \boldsymbol{I}_d$	4	8
$\breve{\boldsymbol{x}}_8^{\mathrm{z}} = \boldsymbol{x}_8^{\mathrm{z}} - \boldsymbol{x}_8^{\mathrm{z}\star}$	Std of $\tilde{\boldsymbol{\tau}}_c$	$\boldsymbol{U}_d, \boldsymbol{I}_d$	4	8
$\breve{\boldsymbol{x}}_9^{\mathrm{z}} = \boldsymbol{x}_9^{\mathrm{z}} - \boldsymbol{x}_9^{\mathrm{z}\star}$	Similarity Matrix	$\boldsymbol{U}_d, \boldsymbol{I}_d, \boldsymbol{R}_d$	6	18
Total number of features				54

Table 4.5: Overview of compared QH features.

Feature compression				
Feature $\overline{x}_f^z$	Description	Nr. Features per Magnet	Cap Value	Unit
Compressed features $\overline{\boldsymbol{x}}^{\mathrm{u}}$				
$\overline{x}_1^{\mathrm{u}}$	Initial Value	1	25	V
$\overline{x}_4^{\mathrm{u}}$	τ Energy Approach	1	0.0025	ms
$\overline{x}_8^{\mathrm{u}}$	Std of $\tilde{\boldsymbol{\tau}}_c$	1	2	s
$\overline{x}_9^{\mathrm{u}}$	Similarity Matrix	1	1	
$\overline{x}^{\mathrm{u,c2c}}$	C2C comparison	1		
Compressed features $\overline{\boldsymbol{x}}^{\mathrm{i}}$				
$\overline{x}_1^{\mathrm{i}}$	Initial Value	1	25	A
$\overline{x}_4^{\mathrm{i}}$	τ Energy Approach	1	0.0025	ms
$\overline{x}_8^{\mathrm{i}}$	Std of $\tilde{\boldsymbol{\tau}}_c$	1	0.005	s
$\overline{x}_9^{\mathrm{i}}$	Similarity Matrix	1	1	
$\overline{x}^{\mathrm{i,c2c}}$	C2C comparison	1		
Compressed features $\overline{\boldsymbol{x}}^{\mathrm{r}}$				
$\overline{x}_1^{\mathrm{r}}$	Initial Value	1	5	Ω
$\overline{x}_9^{\mathrm{r}}$	Similarity Matrix	3	1	
$\overline{x}^{\mathrm{r,c2c}}$	C2C comparison		2	
Total number of features		13		

Table 4.6: Overview of compressed QH features.

4.4 Modelling

Generally, modelling methods can be divided into model-based and data-driven. While the model-based methods rely on the physical laws representing a failure model, the data-driven methods use the data provided from sensors to fit a model that represents the behavior of a system. The complexity of model-based methods and the availability of large amounts of data make the data-driven approach an attractive solution, widely adopted in the industry. Data driven modeling can be implemented through machine learning, a sub-field of artificial intelligence, which is further divided in three categories :

1. *Supervised learning [39]*
   Each input vector $\boldsymbol{x}_d$ for $d \in [1, ..., D]$, where D is the amount of data samples, has a corresponding output value $\langle \boldsymbol{x}_1, y_1 \rangle, ..., \langle \boldsymbol{x}_D, y_D \rangle$. The goal of supervised learning is to train a decision function, derived from the training data in order to make it possible to predict the output of the validation data. With linear regression, a supervised learning technique has already been used in this thesis (see Section 4.2.5). It has been called regression, as each value of the output vector y_d could consist of a linear combination of the input vector $\boldsymbol{x}_d$ and the hyper parameter $\boldsymbol{\theta}$. In contrast, in classification, each value of the output value y_d consists of a finite value of discrete categories. This kind of supervised learning will be used in this section.

2. *Unsupervised learning [39]*
   Contrary to supervised learning, unsupervised learning the training data consists of input values $\boldsymbol{x}_d$ without corresponding output values. Therefore, the goal of unsupervised learning is to detect a structure from data (for example similar segments), or to project data from a high-dimensional space to a space with less dimensions for the purpose of visualization.

3. *Reinforcement learning [39]*
   In reinforcement learning there are no given samples which describe the best possible output, but feedback which can reward a system in a process of trial and error. Therefore, a problem is tried to be solved by maximizing the rewards to suitable actions for given situations. This means that a system, trained with reinforcement learning, typically learns through interaction with its environment in a series of trials.

Before it is possible to implement a data-driven model, it is first necessary to define a data-set from which the decision function can learn from. Therefore, it is necessary to assign an output label y_d to each input feature matix $\boldsymbol{X}_d$. As some of the events have already been classified (see Section 4.1.1), this section will only focus on all events with $y_d < 2$, as their classification is less univocal and not necessarily linked to manual changes in the LHC. The labeled data-set is created as follows:

1. First the threshold-based classification approach from the QHDA tool is implemented. This also helps to compare and validate the feature calculation from this thesis with the feature calculation from the QHDA tool.
2. All events which are classified differently are then manually labeled with the help of experts.
3. The resulting data-set $\langle \boldsymbol{X}_1, y_1^\star \rangle ... \langle \boldsymbol{X}_D, y_D^\star \rangle$, with D as the number of events, is used as a reference for all classification methods, described in this section.[14]

4.4.1 Threshold-based Classification

For the threshold-based classification the output label $y_d \in \{-1, 1\}$ is defined as:

$$y_d = g(\boldsymbol{x}_f^{\mathrm{z}}) = \begin{cases} 1 & \text{if } \check{\boldsymbol{K}} < \boldsymbol{x}_f^{\mathrm{z}} < \hat{\boldsymbol{K}}. \\ -1 & \text{otherwise} \end{cases} \tag{4.52}$$

Therefore, the thresholds are chosen as stated in Table 4.7. While the thresholds from the "characteristic time of the pseudo-exponential decay" are the same as in the QHDA tool (see Section 4.1.2), the thresholds are chosen differently in the "signal comparison" step. This is because the features of the signal comparison differ from the QHDA features. For the existing QHDA tool the signals of the four circuits are merged together and are then directly compared with a threshold envelope. However, in the approach described in this thesis, the signals are compared within each other (calculation of M) and are then compared to the reference signal (calculation of $C2C$) in order to gain more features for the data-driven model (see Section 4.4.3). Furthermore, the QHDA features, of the time period 2014-2018, are not calculated with the normalized signals.[15]

In addition to the different feature calculation method also the signal pre-processing of this approach slightly differs from the pre-processing of the QHDA tool, due to the different programming environment. As a result, the classification of the threshold-based (TB) classification deviates by about 0.6% to the classification of the QHDA tool. While both approaches could find all events with critical indicators of QH faults or precursors (see Figure 4.9), signals with less critical indicators (i.e. variance in the characteristic time of the pseudo-exponential decay) sometimes are classified differently. In order to compare the performance of both TB approaches, all events which have been classified differently, are further examined by experts. The resulting manually labeled data-set $\langle \boldsymbol{X}_1, y_1^\star \rangle ... \langle \boldsymbol{X}_D, y_D^\star \rangle$ contains 3130 "healthy" events and 116 "faulty"

[14] In fact $\boldsymbol{X}_d$ is vectorized to get rid of one dimension, leading to a vector $\boldsymbol{x}_d$ of the size $\mathbb{R}^{\check{F}L+\overline{F}}$. However, for the sake of continuity, $\boldsymbol{X}_d$ is kept as a notation.

[15] Normalization has been implemented in the QHDA tool during the second Long Shutdown in 2019

Feature Name	Corresponding Signals	Lower threshold $\check{\boldsymbol{K}}$	Upper threshold $\hat{\boldsymbol{K}}$	Unit
Characteristic time of the pseudo-exponential decay				
$\breve{\boldsymbol{x}}_4^{\mathrm{u}}$	$U_{\mathrm{HDS}}, U_{\mathrm{HDS,REF}}$	-0.003	0.003	s
$\breve{\boldsymbol{x}}_4^{\mathrm{i}}$	$I_{\mathrm{HDS}}, I_{\mathrm{HDS,REF}}$	-0.003	0.003	s
Signal comparison				
$\breve{\boldsymbol{x}}_7^{\mathrm{u}}$	U_{HDS}	-	1.2	-
$\breve{\boldsymbol{x}}_7^{\mathrm{i}}$	I_{HDS}	-	1	-
$\breve{\boldsymbol{x}}_7^{\mathrm{r}}$	R_{HDS}	-	30	-
$\breve{\boldsymbol{x}}^{\mathrm{u,c2c}}$	$U_{\mathrm{HDS}}, U_{\mathrm{HDS,REF}}$	-	0.00001	%
$\breve{\boldsymbol{x}}^{\mathrm{i,c2c}}$	$I_{\mathrm{HDS}}, I_{\mathrm{HDS,REF}}$	-	0.00001	%
$\breve{\boldsymbol{x}}^{\mathrm{r,c2c}}$	$R_{\mathrm{HDS}}, R_{\mathrm{HDS,REF}}$	-	0.00001	%

Table 4.7: Feature acceptance limits from the current QHDA tool [29].

events. The performance of both TB approaches is then measured by the intersection of each classification to the manually labeled data-set (see Section 4.4.5).

4.4.2 Feature Distribution

In order to chose the right data driven model for classification, it is first necessary to analyze the distribution of the features. Accordingly, the following plots (see Figure 4.15, Figure 4.16, and Figure 4.17), show the distribution of the compressed features $\overline{x}^{\mathrm{z}}$. In order to visualize the data, each feature is plotted against each other in a scatter matrix. Each of the diagonal plots show two histograms with the "healthy" and "faulty" class labels. One can see that the labels of those features are partially overlapping (i.e. red and green) and are, therefore, mostly not clearly separable by a linear threshold. Consequently, an alternative approach for classification is developed in Section 4.4.3 in addition to the implementation of the current QHDA classification.

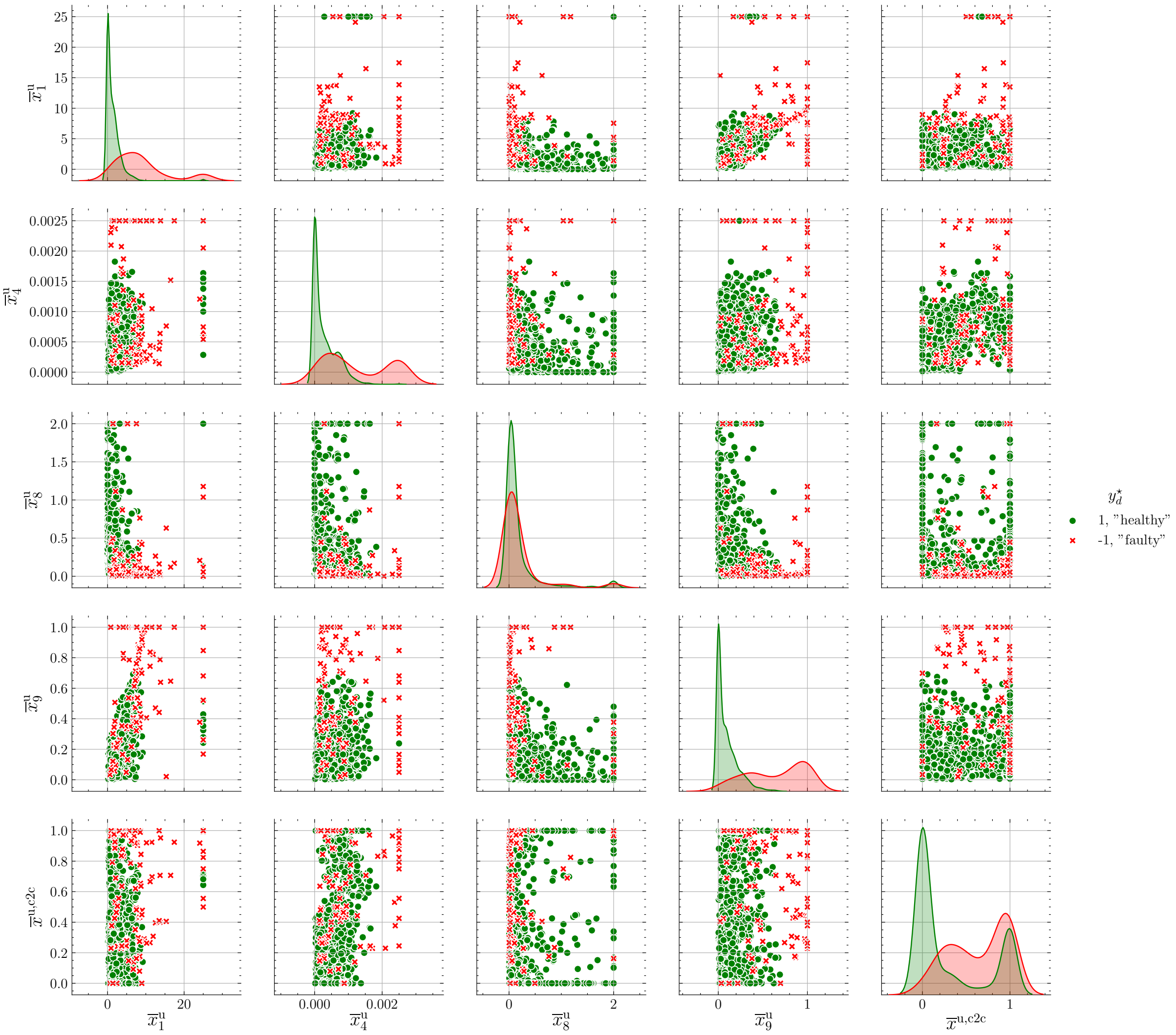


Figure 4.15: Distribution of the compressed QH voltage features $\overline{x}^{\mathrm{u}}$.

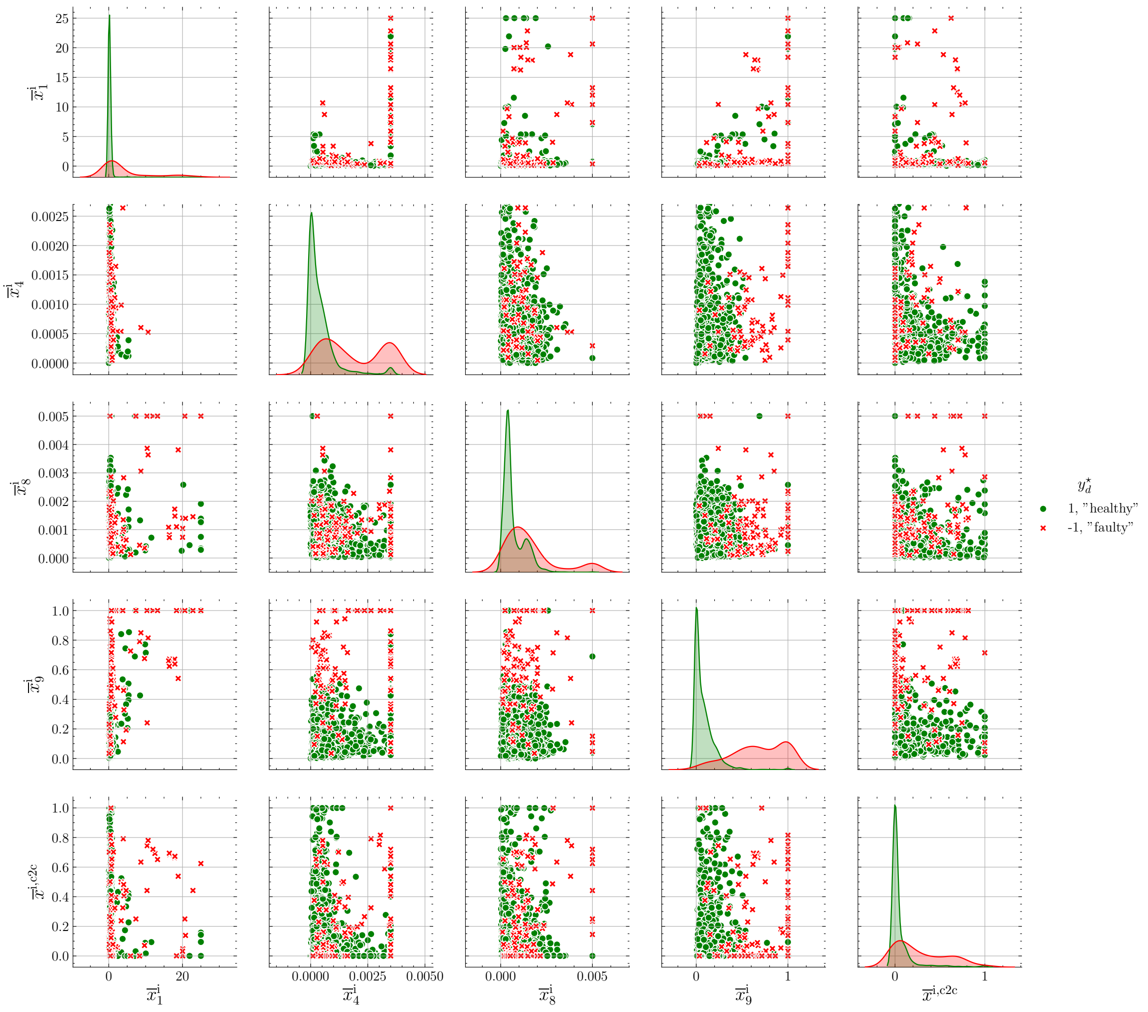


Figure 4.16: Distribution of the compressed QH current features $\overline{x}^{\mathrm{i}}$.

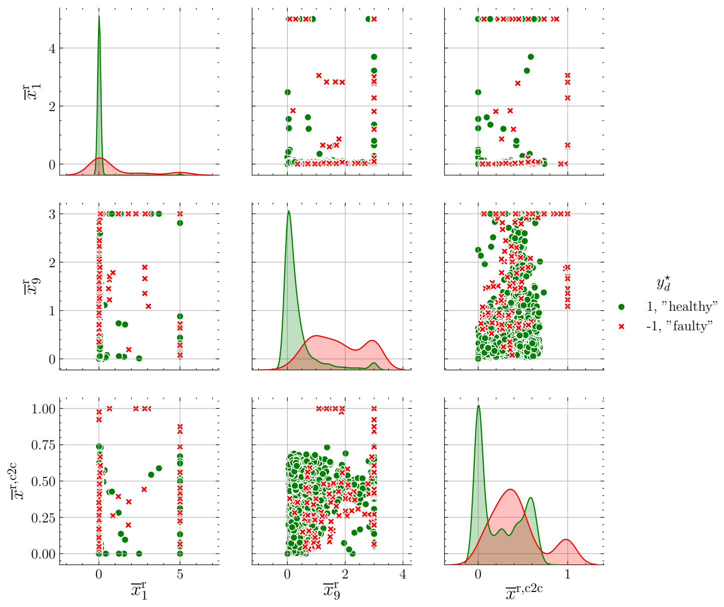


Figure 4.17: Distribution of the compressed QH resistance features $\overline{x}^{\mathrm{r}}$.

4.4.3 Classification with a Support Vector Machine

A Support Vector Machine (SVM) is a classification model for making decisions based on its input parameters, given a training data-set $\langle \boldsymbol{x}_1, y_1\rangle ... \langle \boldsymbol{x}_D, y_D\rangle$. Thereby, a two-class classification $y_d \in \{-1, 1\}$ problem with non-overlapping class distribution is assumed for the moment. This data-set can be separated by the separation hyperplane or decision boundary of the form [39]:

$$h(\boldsymbol{x}_d) = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_d) + b \tag{4.53}$$

where $\phi(\boldsymbol{x})$ is a fixed feature space transformation and b denotes a bias parameter. There exists at least one hyperplane for which:

$$h(\boldsymbol{x}_d) = \begin{cases} > 0 & \text{if } y_d = +1 \\ < 0 & \text{if } y_d = -1, \end{cases} \tag{4.54}$$

such that $y_d h(\boldsymbol{x}_d) \geq 0$ for all training data points. Since there might as well exist many decision boundaries for which Equation (4.54) is true, the SVM determines $\boldsymbol{w}$ and b such that the margin, which is the smallest perpendicular distance between the decision boundary $h(\boldsymbol{x}_d)$ and any sample $\boldsymbol{x}_d$, is minimized. For all solutions where the data points are classified correctly, the perpendicular distance is given by:

$$r = \frac{y_d h(\boldsymbol{x}_d)}{||\boldsymbol{w}||}, \tag{4.55}$$

where $||\boldsymbol{w}||$ is the Euclidean norm of $\boldsymbol{w}$ (see Equation (4.41)). This perpendicular distance can be used later on as a measure of confidence for the classification of each new value y_d.

The samples with the smallest perpendicular distance to the decision boundary are called support vectors which intersect with the margin boundaries (see Figure 4.18). Since scaling of $\boldsymbol{w}$ and b does not change the decision boundary, the support vectors are chosen, such that:

$$y_d(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_d) + b) = 1. \tag{4.56}$$

As a result of the scaling, the optimization problem, therefore, only requires to maximize $||\boldsymbol{w}||^{-1}$, and can be rewritten as [39]:

$$\arg\min_{\boldsymbol{w}} \quad \frac{1}{2}||\boldsymbol{w}||^2 \tag{4.57}$$

$$\text{subject to: } y_d(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_d) + b) \geq 1, \quad n = 1, ..., N. \tag{4.58}$$

Figure 4.18 shows a synthetic data-set which is divided by the separation hyperplane. In this example the data-set is linearly separable, which is why the fixed feature space transformation is linear as well $\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{x}$.

The optimization problem 4.57 can be solved with the Lagrange multipliers $\boldsymbol{a} = (a_1, ..., a_d)^{\mathrm{T}}$ where each element $a_d \geq 0$ [39]:

$$L(\boldsymbol{x}, b, \boldsymbol{a}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{d=1}^{D} a_d \{y_d(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_d) + b) - 1\}. \tag{4.59}$$

By setting the derivative of $L(\boldsymbol{x}, b, \boldsymbol{a})$ with respect to $\boldsymbol{x}$ and b to zero and substitute the result back into Equation (4.59), the dual representation of the maximum margin problem can be

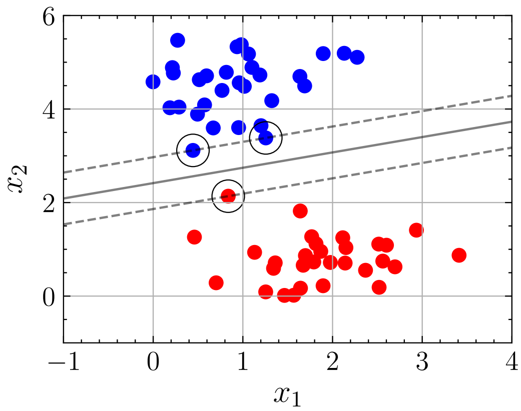


Figure 4.18: Illustration of a two dimensional synthetic data-set which is separated by a SVM. Thereby, the continuous line represents the decision boundary, the dotted lines represent the margin boundaries, and the encircled samples the support vectors.

obtained [39]:

$$\max_{\boldsymbol{a}} \qquad \widetilde{L}(\boldsymbol{a}) = \sum_{d=1}^{D} a_d - \frac{1}{2}\sum_{d=1}^{D}\sum_{m=1}^{D} a_d a_m y_d y_m k(\boldsymbol{x}_d, \boldsymbol{x}_m) \tag{4.60}$$

$$\text{subject to: } a_d \geq 1, \quad d = 1, ..., D \tag{4.61}$$

$$\sum_{d=1}^{D} a_d y_d = 1, \tag{4.62}$$

where the kernel function is defined as $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}')$. Equation (4.62) can be optimized with a quadratic programming algorithm, which returns $\boldsymbol{a}$. The Lagrange multipliers can be used to obtain the parameters of the separation hyperplane (4.53):

$$\boldsymbol{w} = \sum_{d=1}^{D} a_d y_d \boldsymbol{\phi}(\boldsymbol{x}_d), \tag{4.63}$$

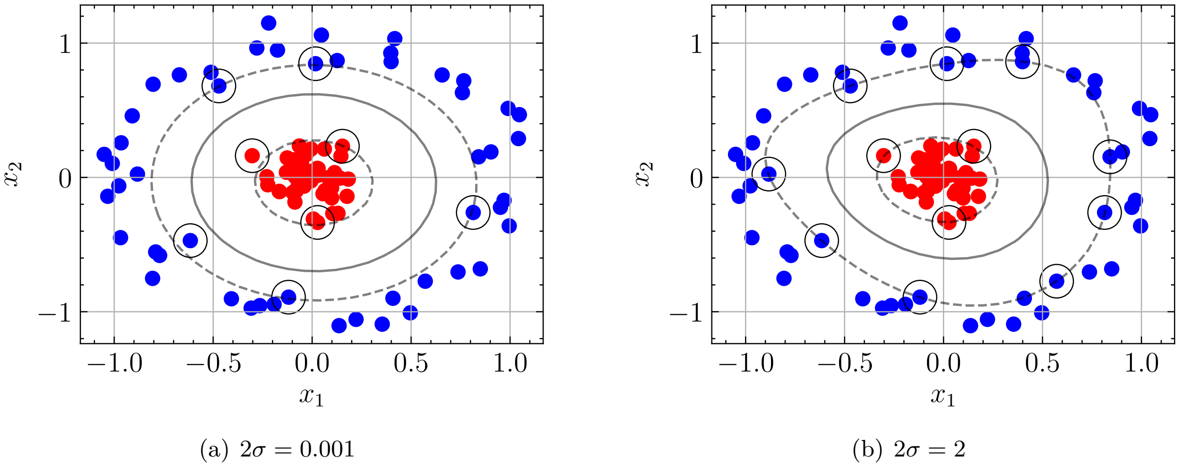
$$b = y_d - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_d). \tag{4.64}$$

Figure 4.19 shows an example of a data-set which is not linearly separable. However, with the commonly used Gaussian kernel function:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \mathrm{e}^{-\frac{||\boldsymbol{x}-\boldsymbol{x}'||^2}{2\sigma}}, \tag{4.65}$$

it is possible to transform the input parameter into a linear separable feature space. Therefore, the Gaussian kernel function can be used as an additional tool, which can be helpful for certain classification problems. Figure 4.19 further shows the impact of the parameter 2σ, which can be used as a regularization term in order to avoid over- or under-fitting.

However, not every feature space $\boldsymbol{\phi}(\boldsymbol{x})$ might be linearly separable, as there can be outliers or intersections in certain classes. Therefore, soft margins make it possible to allow mislabeled samples. With a slack variable $\xi_d \geq 0$ for each data point d, a regularization term is appended



(a) $2\sigma = 0.001$

(b) $2\sigma = 2$

Figure 4.19: Example of classification with a support vector machine using a Gaussian kernel function with different parameter settings.

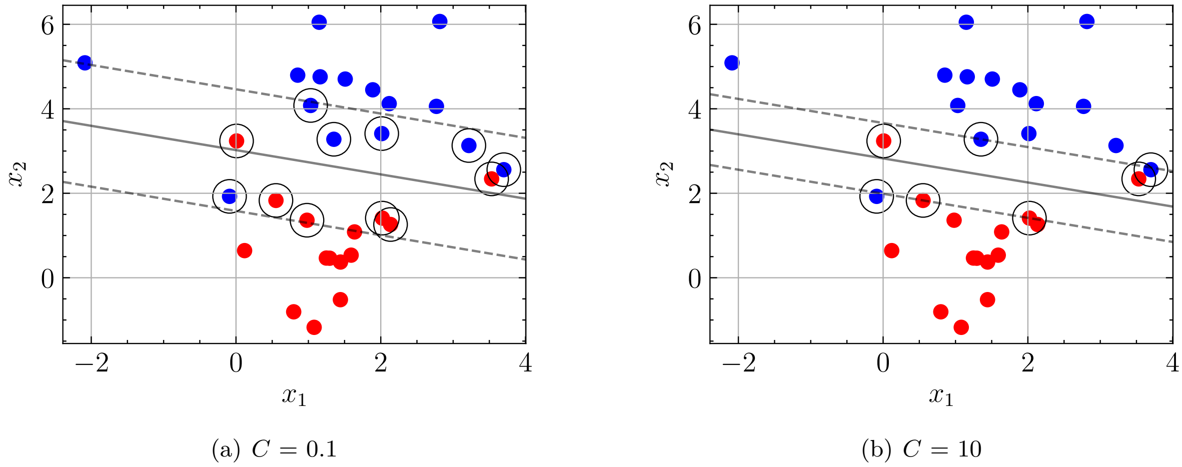
to the optimization problem:

$$\underset{\boldsymbol{w}}{\arg\min} \quad \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{d=1}^{D}\xi_d \tag{4.66}$$

$$\text{subject to: } y_d(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_d) + b) \geq 1 - \xi_d, \quad d = 1, ..., D \tag{4.67}$$

$$\xi_d \geq 0. \tag{4.68}$$

Thereby, $\xi_d = 0$ for data points which are on the correct side of the margin boundary and $\xi_d = |y_d - h(\boldsymbol{x}_d)|$ for other points. This means that data points with $\xi_d > 1$ are misclassified. The free parameter C regulates the importance of outliers to the classification. The smaller this free parameter is chosen, the more impact do outliers have on the classification process. Figure 4.20 shows the separation of a data-set with overlapping class distribution, by using the soft margin classification method with different settings for the parameter C. Similar to the algebraic transformations from 4.59 to 4.64, Lagrange multipliers are used to solve the optimization problem 4.66.



(a) $C = 0.1$

(b) $C = 10$

Figure 4.20: Figure of a SVM with soft marging classification with different parameter settings.

Implementation

The SVM is trained on reference data-set $\langle \boldsymbol{X}_1, y_1^\star \rangle ... \langle \boldsymbol{X}_D, y_D^\star \rangle$. Thereby, $\boldsymbol{X}_d$ includes the the normalized compared features (see Table 4.5) and the normalized compressed features (see Table 4.6):

$$\boldsymbol{X}_d = \{\breve{\boldsymbol{X}}^{\mathrm{z}\diamond}, \overline{\boldsymbol{x}}^{\mathrm{z}\diamond}\} \in \mathbb{R}^{\overline{F} \times (L+1)},$$

for $d \in [1, ..., D]$ where D is the amount of events for which $y_d^\star < 2$ (two classes). Furthermore, it is important to mention, that for the training of the SVM, $\boldsymbol{X}_d$ is vectorized to get rid of one dimension. As the Euclidean norm is used as a distance function for the SVM, normalization (see Equation (4.42)) is necessary to give the same weight to each feature. The parameters of the SVM are chosen empirically in order to keep the amount of false-negative labels relatively low ($2\sigma = 0.03$ and $C = 5$).

The class imbalance of the data-set $y_d^\star$ (3130 "healthy" and 116 "faulty" events) can cause several problems. First, it leads to a stronger consideration of the more dominant label in the optimization function (see Equation (4.66)) during training of the SVM. This problem is covered by replicating the smaller class of the training set until the class labels are balanced [40]. Secondly, the classification significantly depends on the separation of the data-set into training and validation set. In case the features of the less dominant label differ too much within the two data-sets, the decision boundary cannot be adjusted accordingly. In order to reduce the impact of the separation significance, K-fold cross-validation is used. Therefore, the data-set is randomly split into $K = 5$ folds.[16] For each split the SVM, the parameters $\boldsymbol{w}$ and b are calculated for the training set and the target values y_d are determined with the input parameter $\boldsymbol{X}_d$ from the validation set. The performance of the SVM is then measured as the classification accuracy of each cross-validation fold K (see Section 4.4.5).

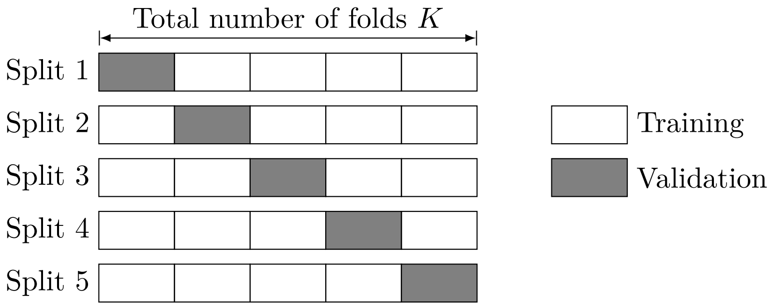


Figure 4.21: Illustration of K-fold cross-validation, where $K = 5$.

One can see that the TB classification and the SVM classification work with the same inputs, which is why it is further possible to combine both methods as a hybrid classification approach, introduced in the next section.

[16] As the data-set is "shuffled" before splitting, the results in Section 4.4.5 represent the average values over 100 iterations.

4.4.4 Hybrid Classification Approach

With the hybrid classification approach, the workflow from the current QHDA tool (see Figure 4.4) is extended, such that the amount of false-negative classified labels is minimized and that there is less manual adjustment necessary by experts. The workflow of the hybrid classification approach can be seen in Figure 4.22. Thereby, the SVM makes it possible to incorporate expert decisions into the classification process.

In the initial phase, the parameters $\boldsymbol{w}$ and b of the SVM separation hyperplane are determined with the past data by using Equation (4.66). Then, for each new event d, the TB classification evaluates y_d^{tbc} (see Equation (4.2)) and the SVM classification calculates y_d^{svm} (see Equation (4.53)) with the features $\boldsymbol{X}_d$. The merged output parameter y_d is then determined by:

$$y_d = \begin{cases} 1 & \text{if } y_d^{\text{tbc}} = 1 \wedge y_d^{\text{svm}} = 1 \\ -1 & \text{otherwise,} \end{cases} \tag{4.69}$$

which is checked by experts. Once the experts have evaluated the QH condition, the corresponding label is put into training set and the SVM parameters $\boldsymbol{w}$ and b are recalculated. Accordingly, the new label is used as feedback for future decisions of the SVM.

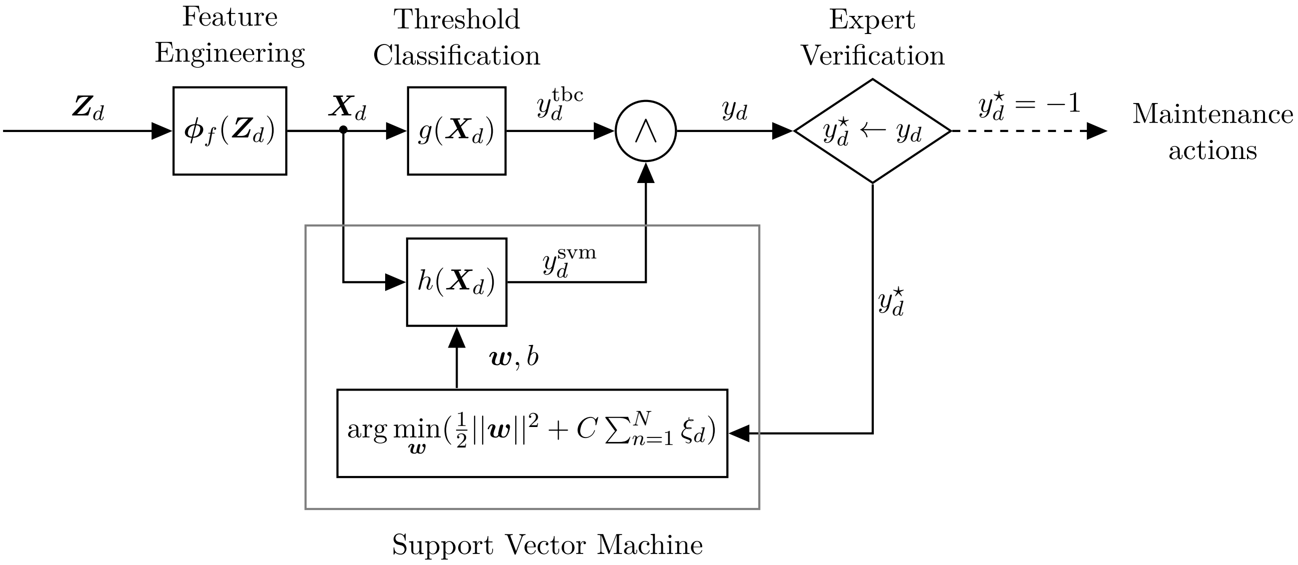


Figure 4.22: Workflow of the hybrid classification approach.

In order to simulate the performance of the hybrid classification approach, K-fold cross-validation is used again. For each fold, the SVM is trained on the training set and the output parameter y_d is evaluated on the validation set.

4.4.5 Results

In order to compare all classification approaches, a confusion matrix is used [41], which helps to visualize the performance of each approach. Figure 4.23 shows how a confusion matrix is structured. In this case, true positive events include all "healthy" events and true negative events all "faulty" events, which are classified correctly. In contrast, labels which are wrongly predicted as "healthy" are defined as false positive, and labels which are wrongly labeled as "faulty" are defined as false negative. Furthermore, the values from this matrix can be used to calculate several common performance metrics [41]:

		True Class	
		p	n
Hypothesized Class	p′	**T**rue **P**ositive	**F**alse **P**ositive
	n′	**F**alse **N**egative	**T**rue **N**egative
		P=TP+FN	N=FP+TN

Figure 4.23: Structure of a confusion matrix.

$$\text{Accuracy} = \frac{TP+TN}{P+N} \qquad \text{TP Rate} = \frac{TP}{P} \qquad \text{TN Rate} = \frac{TN}{N} \tag{4.70}$$

It is important to mention that in both classification workflows presented in this thesis, experts only verify negative predicted values n′. Therefore, a false positive prediction can be crucial to the operation of the LHC. Thus, a high true negative rate is especially important to ensure the high availability of the LHC.

As stated before, the reference data-set $y_d^\star$ consists of 3130 healthy labels and 116 negative labels. Since K-fold cross-validation, with $K = 5$, is applied to all approaches, this leads to an average number of $P = 626$ positive labels and $N = 23.2$ negative labels in the validation set. Table 4.8 shows the resulting confusion matrix after comparing the predicted labels y_d from the different approaches with the reference labels $y_d^\star$.

		True Class		
		Method	p	n
Hypothesized Class	p′	QHDA tool classification	625.80	3.40
		TB classification	621.80	1.80
		SVM classification	620.57	3.05
		Hybrid classification	618.57	0.55
	n′	QHDA tool classification	0.20	19.80
		TB classification	4.20	21.40
		SVM classification	5.43	20.15
		Hybrid classification	7.43	22.65
		$\sum$	626.00	23.20

Table 4.8: Confusion matrix, which compares the predicted labels y_d with the reference labels $y_d^\star$.

Furthermore, the performance metrics, defined in Equation (4.70), are calculated in Table 4.9. One can see that the QHDA tool classification has the highest accuracy as the amount of true positive labels is especially high. However, there are also the most false positive labels. In contrast, the hybrid classification method has the best performance on "faulty" labels as it contains "faulty" labels from the TB classification and the SVM classification. Consequently, however, this leads to more false negative labels which affect the accuracy of the hybrid classification approach negatively. Due to the class imbalance the true negative rate has less effect on the accuracy than the true positive rate.

	Accuracy	True Positive Rate	True Negative Rate
QHDA tool classification	99.45%	99.97%	85.34%
TB classification	99.07%	99.33%	92.24%
SVM classification	98.69%	99.13%	86.87%
Hybrid classification	98.77%	99.88%	97.61%

Table 4.9: Results of different performance measures in %.

Figure 4.24 shows the Region Of Convergence (ROC) graph for the four different classifiers. This graph visualizes the trade-off between benefits (true positive) and costs (false positive)[41]. Therefore, the true positive rate is plotted on the vertical axis and the false positive rate (FPR =1-TNR) is plotted on the horizontal axis. Consequently, the point (0,1) represents the ideal classifier. This means that the closer a classifier is to the upper left side of the ROC graph, the better it is. The ROC graph can be further extended by plotting the performance of each classifier for different parameter inputs (e.g. threshold values).

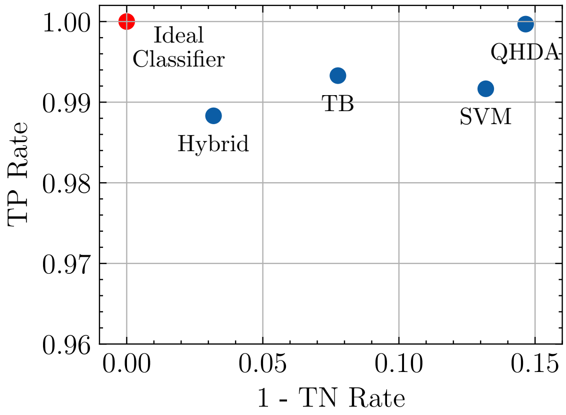


Figure 4.24: ROC visualisation of the different classifiers.

5 Conclusion

The goal of this thesis was to develop applications within the LHC signal monitoring project, with a method adaptable to further use cases. Therefore, a general application development approach is introduced and documented.

The busbar resistance analysis was a first proof of concept application within the LHC signal monitoring project. It was shown how the general application development approach from Chapter 2 can be applied to this use case. Therefore, a possible approach for graphical visualization and data analysis within the LHC signal monitoring project, further described in [24], [28], was illustrated.

Subsequently, the QHs were analyzed with this approach. During the implementation of this use case, several goals were achieved. First, the signals from a QH were treated with a sequence of preprocessing methods. In that context, the advantages and disadvantages of existing filtering methods were tested and displayed. New filtering methods, like the extended method to find the start of the decay, were implemented to generalize the scope of application.

Secondly, several functions were defined to embed the features from the existing QHDA tool. As the approximation of the characteristic time of the exponential decay is a major component of the existing QHDA tool, different methods to obtain this value were compared by stating their properties. Considering the subsequent implementation of a data driven-model, new functions to calculate features were defined. For example, the Euclidean distance was used to calculate the similarity matrix between the QH signals of one magnet. In order to avoid overfitting of a data-driven model, all events and features were statistically analyzed and filtered. Therefore, the Pearson correlation coefficient was used as a measure for the linear dependency of the features.

Thirdly, the threshold based classification approach from the existing QHDA tool was implemented with the acquired features. With the help of experts the difference of both approaches was analyzed and a manually labeled database was created. This manually labeled database was used to train a data-driven classification model. In particular, a SVM was adjusted to the specific properties of the data-set with a Gaussian kernel function and soft margins. Furthermore, the impact of the class-imbalance was reduced by replicating the less dominant class labels during training and splitting the data-set into K-folds during validation. Finally, a new workflow for the classification of QH discharges was presented which allows the automatic incorporation of expert decision into the classification process. The key strengths of this hybrid classification approach is the combination of the existing model-based approach with the new data-driven approach. Therefore, it combines the advantages of the threshold based classification, with the advantages of the SVM classification. On the one hand, it promises to detect QH faults and precursors with a similar accuracy as in the existing QHDA tool, and on the other hand e.g. the perpendicular distance to the decision boundary is available for each label, which can be used as a measure of confidence for the classification. The main advantage of the hybrid classification approach was the better score on the true negative rate which can improve the availability of the QHs.

5.1 Outlook

The approach presented in this thesis, including the mathematical notations, the defined functions, and the chosen modeling methods, provide a foundation for additional use cases within the LHC signal monitoring project. The scope of new use cases includes but is not limited to:

- Power Converters
- Current leads
- Grounding networks
- Circuit and magnet protection systems

Additionally, the QH condition analysis can be further explored by several future research avenues:

Since the data-driven model adjusts the decision boundary automatically, further features can be easily appended without the need for initial threshold development. Thereby, additional features for measuring the point-wise similarity of the QH signals could be especially interesting.

While the introduced hybrid classification workflow offers a promising solution to incorporate expert decisions, the advantages of the data-driven model can be even further exploited. As a first step, several different actions could be introduced, based on the classification confidence. For example selected experts could already be notified, if there is a "healthy" label which is close to the decision boundary. Moreover, the classification with the SVM can be improved by adjusting the used kernel to the exact probability density function of the features. Furthermore, it is important to mention that the SVM was only chosen as an example for data-driven models. However, an in-depth analysis of further data-driven classification models, e.g. Random Forest, Neural Networks would be interesting. By doing so it, could be helpful to define more class labels to distinguish between critical and non-critical QH faults and precursors.

While no precursors of fault, which could be used to predict a condition change of a QH, could be found, further prediction opportunities can be investigated in the future. As there is a lack of data, those methods will, however, likely require specific techniques to implement meta information (e.g. magnet replacements) and to correlate the behaviours of different QHs with each other.

One of the limitations within the presented approach is still the extraction of the exponential decay from the QH signals during preprocessing. As this step impacts the following signal comparison drastically, an improvement of this method could have a strong positive effect on the performance of any classifier.

While all classification approaches within the LHC signal monitoring project were only tested with offline data in SWAN, the next step is to prepare the approach for online monitoring.

Bibliography

[1] "LHC Season 2 facts & figures," CERN, Ed., 2017.

[2] Ashish, *What Does a Particle Accelerator Actually Do?* en-US, Oct. 2016. [Online]. Available: https://www.scienceabc.com/pure-sciences/what-does-a-particle-accelerator-actually-do.html (visited on 10/24/2019).

[3] M. Maciejewski, "Co-Simulation of Transient Effects in Superconducting Accelerator Magnets," PhD thesis, Łódź University of Technology, Oct. 2018.

[4] A. Siemko, *Safeguarding the superconducting magnets – CERN Courier*, Sep. 2013. [Online]. Available: https://cerncourier.com/safeguarding-the-superconducting-magnets/ (visited on 09/12/2018).

[5] AC-Team, *Diagram of an LHC dipole magnet - CERN Document Server*, Jun. 1999. [Online]. Available: http://cds.cern.ch/record/40524 (visited on 12/04/2019).

[6] A. Siemko, "Magnet Quench Process," *CERN, Geneva, Switzerland*, 2001.

[7] Education-Communications-and-Outreach-Group, "LHC FAQ the guide," 2017.

[8] F. Rodriguez-Mateos, R. Schmidt, A. Siemko, and F. Sonnemann, "Quench Process and Protection of LHC Dipole Magnets," *LHC Project Note 184*, Jul. 1999.

[9] E. Ravaioli, K. Dahlerup-Petersen, F. Formenti, J. Steckert, H. Thiesen, and A. Verweij, "Modeling of the Voltage Waves in the LHC Main Dipole Circuits," en, *IEEE Transactions on Applied Superconductivity*, vol. 22, no. 3, pp. 9 002 704–9 002 704, Jun. 2012, ISSN: 1051-8223, 1558-2515. DOI: 10.1109/TASC.2011.2176306. [Online]. Available: http://ieeexplore.ieee.org/document/6082398/ (visited on 11/04/2019).

[10] A. Liakopoulou, "Analysis of Short-Circuit Transients in the LHC Main Dipole Circuit and Development of an Automated Analysis Algorithm," en, p. 73, 2019.

[11] CERN, *CERN releases analysis of LHC incident | Media and Press Relations*, 2018. [Online]. Available: https://press.cern/press-releases/2008/10/cern-releases-analysis-lhc-incident (visited on 09/19/2018).

[12] Celladoor, *Daten, Fakten und Funktionen des neuen Teilchenbeschleunigers von CERN bei Genf*, de-CH, LHC; Physik, 2019. [Online]. Available: http://www.lhc-facts.ch/index.php?page=beamdump (visited on 10/24/2019).

[13] M. Zerlauth, O. O. Andreassen, V. Baggiolini, A. Castaneda, R. Gorbonosov, D. Khasbulatov, H. Reymond, A. Rijllart, I. R. Ramirez, and N. Trofimov, "The LHC Post Mortem Analysis Framework," en, p. 3, 2009.

[14] C. Roderick, L. Burdzanowski, and G. Kruk, "The CERN Accelerator Logging Service-10 Years in Operation: A Look at the Past, Present and Future," *14 th International Conference on Accelerator & Large Experimental Physics Control Systems*, 2013.

[15] K. Sigerud, *CALS - CERN Accelerator Logging Service*, Oct. 2016. [Online]. Available: https://be-dep-co.web.cern.ch/content/cals (visited on 09/12/2018).

[16] InfluxData, *InfluxData Documentation*, en, 2019. [Online]. Available: https://docs.influxdata.com/ (visited on 11/05/2019).

[17] M. Maciejewski, K. Andersen, P. Hagen, C. Obermair, A. Verweij, M. Mentink, T. Buffet, J.-C. Garnier, T. M. Ribeiro, M. Zerlauth, and Z. Charifoulline, "LHC Signal Monitoring Project," en, p. 17, Jul. 2019.

[18] K. Andersen, "Development of a lean API for time-series data," en, p. 27, Jul. 2019. [Online]. Available: https://jakevdp.github.io/PythonDataScienceHandbook/index.html.

[19] W. McKinney and PyData-Development-Team, *Pandas: Powerful Python data analysis toolkit*, Documentation, Release 0.23.4, Aug. 2018. [Online]. Available: https://pandas.pydata.org/pandas-docs/version/0.23.4/generated/pandas.DataFrame.html (visited on 02/20/2020).

[20] D. Piparo, E. Tejedor, P. Mato, L. Mascetti, J. Moscicki, and M. Lamanna, "SWAN: A service for interactive analysis in the cloud," en, *Future Generation Computer Systems*, vol. 78, pp. 1071–1078, Jan. 2018, ISSN: 0167739X. DOI: 10.1016/j.future.2016.11.035. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167739X16307105 (visited on 02/20/2020).

[21] P. J. Community, *Project Jupyter*, Jan. 2020. [Online]. Available: https://www.jupyter.org (visited on 02/20/2020).

[22] CERN-IT-Department, *Git Service*, 2019. [Online]. Available: http://information-technology.web.cern.ch/services/git-service (visited on 11/08/2019).

[23] SonarSource, *SonarQube*, 2019. [Online]. Available: https://www.sonarsource.com/products/sonarqube/ (visited on 11/08/2019).

[24] C. Obermair, "Signal monitoring for the LHC," en, p. 10, 2018. [Online]. Available: http://cds.cern.ch/record/2639871?ln=en.

[25] Z. Charifoulline, M. J. Bednarek, R. Denz, S. Le Naour, C. Scheuerlein, A. Siemko, J. Steckert, J.-P. Tock, A. Verweij, and M. Zerlauth, "Resistance of Splices in the LHC Main Superconducting Magnet Circuits at 1.9 K," en, *IEEE Transactions on Applied Superconductivity*, vol. 28, no. 3, pp. 1–5, Apr. 2018, ISSN: 1051-8223, 1558-2515. DOI: 10.1109/TASC.2017.2784355. [Online]. Available: http://ieeexplore.ieee.org/document/8219360/ (visited on 08/17/2018).

[26] P. P. Granieri, M. Breschi, M. Casali, and L. Bottura, "Stability analysis of the interconnection of the LHC main superconducting bus bars," CERN, Presentation, 2011.

[27] Z. Charifoulline, Dahlerup-Petersen, R. Denz, A. Siemko, and J. Steckert, "Splice Resistance Measurements in the LHC Main Superconduction Magnet Cuircuits by the New Quench Protection System," en, *Proceedings of IPAC2012, New Orleans, Louisiana, USA*, vol. THPPD026, 2012, OCLC: 868315783. [Online]. Available: http://accelconf.web.cern.ch/accelconf/IPAC2012/ (visited on 08/17/2018).

[28] C. Obermair, "Development of an application for analysing the main quadrupole busbar resistance," CERN, Presentation, Sep. 2018. [Online]. Available: https://indico.cern.ch/event/758122/.

[29] Z. Charifoulline, L. Bortot, R. Denz, F. R. Mateos, A. Siemko, J. Steckert, A. Verweij, and G. Willering, "Overview of the Performance of Quench Heaters for High-Current LHC Superconducting Magnets," en, *IEEE Transactions on Applied Superconductivity*, vol. 27, no. 4, pp. 1–5, Jun. 2017, ISSN: 1051-8223, 1558-2515. DOI: 10.1109/TASC.2016.2642991. [Online]. Available: http://ieeexplore.ieee.org/document/7792684/ (visited on 09/13/2018).

[30] G. Chlachidze, N. Andreev, G. Apollinari, B. Auchmann, E. Barzi, R. Bossert, M. Karppinen, F. Nobrega, I. Novitski, L. Rossi, D. Smekens, M. Tartaglia, R. Yamada, and A. V. Zlobin, "Quench Protection Study of a Single-Aperture 11 T Nb3sn Demonstrator Dipole for LHC Upgrades," en, p. 5,

[31] M. Bajko, "Quench heater failure in dipole magnets," en, p. 15, 2009.

[32] N. I. Corporation, *Was ist LabVIEW?* 2019. [Online]. Available: https://www.ni.com/de-at/shop/labview.html (visited on 12/11/2019).

[33] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing*, en, 3rd ed. Upper Saddle River: Pearson, 2010, ISBN: 978-0-13-198842-2.

[34] J. VanderPlas, *Python Data Science Handbook*. O'Reilly Media, Nov. 2016. [Online]. Available: https://jakevdp.github.io/PythonDataScienceHandbook/index.html.

[35] G. Bianchi and R. Sorrentino, "Electronic filter : Simulation design," en, p. 626, 2007.

[36] D. J. Acunzo, G. MacKenzie, and M. C. van Rossum, "Systematic biases in early ERP and ERF components as a result of high-pass filtering," en, *Journal of Neuroscience Methods*, vol. 209, no. 1, pp. 212–218, Jul. 2012, ISSN: 01650270. DOI: 10.1016/j.jneumeth.2012.06.011. [Online]. Available: https://mne.tools/stable/auto_tutorials/discussions/plot_background_filtering.html (visited on 12/21/2019).

[37] P. Rajan, "Two-dimensional digital signal processing II: Transforms and median filters," en, *Proceedings of the IEEE*, vol. 70, no. 7, pp. 780–781, 1982, ISSN: 0018-9219. DOI: 10.1109/PROC.1982.12396. [Online]. Available: http://ieeexplore.ieee.org/document/1456664/ (visited on 12/21/2019).

[38] C. Zaiontz, *Basic Concepts of Correlation | Real Statistics Using Excel*, 2012. [Online]. Available: http://www.real-statistics.com/correlation/basic-concepts-correlation/ (visited on 01/22/2020).

[39] C. M. Bishop, *Pattern recognition and machine learning*, en, ser. Information science and statistics. New York: Springer, 2006, ISBN: 978-0-387-31073-2.

[40] G. King and L. Zeng, "Logistic Regression in Rare Events Data," en, p. 28, 2001.

[41] T. Fawcett, "An introduction to ROC analysis," en, *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, Jun. 2006, ISSN: 01678655. DOI: 10.1016/j.patrec.2005.10.010. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S016786550500303X (visited on 02/07/2020).

List of Figures

List of Tables

Acronyms

API Application Programming Interface. 18–20, 24, 29, 38, 41

CALS CERN Accelerator Logging Service. LXXV, 18–20, 23, 29

CERN Conseil Européen pour la Recherche Nucléaire. LXXV, 14, 18, 20, 25

CL Current Leads. 16, 24

CT Characteristic Time. LXXV, 49

DF DataFrame. 23

EE Energy Extraction. 16, 18, 24

GUI Graphical User Interface. LXXV, 19, 21, 31, 36

HF High Field. 33, 45

HWC Hardware Commissioning. VII, 24, 33, 34, 38, 39

LDB Logging Database. 19

LF Low Field. 33, 37, 45

LHC Large Hardron Colider. VII, IX, LXXV, LXXVII, 13–21, 23–25, 27–29, 33, 38, 41, 45, 51, 52, 57, 66, 69, 70

LS Long Shutdown. 24

MDB Measurement Database. 19

MPS Machine Protection System. 18, 20

Nb-Ti Niobium-Titanium. 14

NXCALS Next Generation CERN Accelerator Logging Service. 20, 29

PC Power Converters. 16, 24

PM Post Mortem. 18, 23, 35, 37–40, 51

PMA Post Mortem Analysis. 18, 36, 39

QH Quench Heater. LXXV, LXXVII, 33–40, 42–46, 49–57, 59, 60, 65, 69, 70

QHDA Quench Heater Discharge Analysis. LXXV, LXXVII, 36–38, 51, 57–59, 65–67, 69

QPS Quench Protection System. 16, 29

RAC Real Application Clusters. 19

ROC Region Of Convergence. LXXV, 67

SCADA Supervisory Control And Data Acquisition. 19

SVM Support Vector Machine. LXXV, 61–67, 69, 70

SVN Apache Subversion. 25

SW Switches. 16

SWAN Service for Web-based ANalysis. 25, 31, 70

TB threshold-based. 57, 58, 64–67