DEPARTMENT OF MATHEMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Mathematics

Deep learning for anomaly detection in high-energy beam dump data from the Large Hadron Collider

Fredrik Hülphers

DEPARTMENT OF MATHEMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Mathematics

Deep learning for anomaly detection in high-energy beam dump data from the Large Hadron Collider

Deep Learning für die Erkennung von Anomalien in hochenergetischen Strahlabbrüchen im Large Hadron Collider

Author:	Fredrik Hülphers
Supervisor:	Prof. Dr. Daniel Cremers
Advisor at TUM:	Dr. Vladimir Golkov
Advisor at CERN:	Dr. Christoph Wiesner
Submission Date:	16 May 2022

I confirm that this master's thesis in mathematics is my own work and I have documented all sources and material used.

Katwijk aan Zee, 16 May 2022 Fredrik Hülphers

Acknowledgments

I would like to express my sincere gratitude to Dr. Christoph Wiesner in the Machine Protection and Electrical Integrity Group at CERN for his unlimited support and guidance throughout this master's thesis work. His motivational words kept me going and helped me overcome obstacles. I truly appreciate all the time spent on supervising me, and for believing in me.

I would also like to express my sincere appreciation to Dr. Vladimir Golkov in the Computer Vision Group at TUM. His supervision, guidance and expert knowledge in deep learning were crucial for this project. I am thankful for his help with developing ideas, and for providing many useful tips on good research practices.

I am also thankful to Dr. Daniel Wollmann in the Machine Protection and Electrical Integrity Group at CERN. His feedback, support and encouraging words helped me a lot, and made my time at CERN more enjoyable.

Abstract

The Large Hadron Collider (LHC) is the world's largest and most powerful particle accelerator, in which beams of particles are accelerated to near the speed of light. Partial losses of high-energy beams can damage crucial LHC equipment, requiring tedious and costly reparations. Beam loss monitoring is performed in order to ensure the safe continuous operation of the LHC required for studying rare phenomena in high-energy physics. In complement to automatic system checks, a manual inspection of beam losses by a machine protection expert is performed after every high-energy beam dump in the LHC, to allow the early identification of issues.

In this master's thesis, we explore two different data-driven approaches to automatizing the analysis of beam losses during high-energy beam dumps in the LHC, with the aim to facilitate the analysis of future beam dumps and contribute to the understanding of beam loss mechanisms. In the two approaches, methods for anomaly detection are applied to historical beam dump data from several years of successful operation. For the first approach, we develop and test a physics-model–driven method to classifying beam dumps, where beam losses are modeled using regression analysis and expert knowledge about physical phenomena in the LHC. For the second approach, we develop and test a classification method using deep learning, for the purpose of efficiently extracting relevant information from large beam dump datasets. Building on this method, we also propose a method using variational autoencoders, for which several different anomaly measures can be defined. Classification results from the physics-model–driven method were validated by machine protection experts, and show that the method works very well for detecting anomalous beam losses. We propose several possible improvements to this promising method, that could lead to its full implementation into the existing beam loss diagnostics systems. Results from the classification method using deep learning show that it can detect known types of beam loss anomalies, and proves the capability of the neural network architecture. This provides a foundation for future studies on the subject, for which we identify several possibilities.

Contents

1. Introduction

1.1. The Large Hadron Collider

The Large Hadron Collider (LHC) is a roughly circular particle accelerator with a circumference of 26.7 km, located at the particle physics laboratory CERN in Geneva, Switzerland [Brü+04]. Beams of charged particles (protons or heavy ions) are accelerated in opposite directions using superconducting radiofrequency cavities. The beams are kept in place using superconducting magnets, that bend the beams (dipole magnets) and focus the beams (quadrupole magnets). The two beams travel in a high-quality vacuum in two parallel beam pipes, and are made to collide inside particle detectors. The main particle detectors are ALICE, ATLAS, CMS and LHCb, which collect experimental data from the particle collisions.

The LHC is divided into eight *octants* and eight *sectors*. An octant consists of two half-sectors, and a sector consists of two half-octants. Each octant consists of a straight section and two *arcs* (one at each end). Figure 1.1 shows a schematic layout of the LHC. In addition to the four experiments, there are four *insertion regions* (IRs) that contain the experiments and machine equipment. IR3 and IR7 contain the collimation system of the LHC, which is responsible for cleaning the beams from too large longitudinal and transversal oscillations, respectively. *Collimators* are absorbers located inside the beam pipes, intended to absorb particles that deviate too far from the intended beam path.

A particle beam in the LHC consists of a number of *bunches* (up to 2808), where each bunch contains many particles (in the order of 1×10^{11}). An operation cycle of the LHC starts with filling the beam pipes with particles, that have been pre-accelerated in smaller accelerators. Individual particles have a kinetic energy of 450 GeV at injection into the LHC. The particle beams, now consisting of bunches, are accelerated to a top energy of 6.5 TeV in Run 2 of the LHC (2015-18). After reaching top energy, the beams are made to collide inside the experiments. The rate of interaction (collision) between the two particle beams is referred to as *luminosity*, and is a key measure for the performance of the LHC. A higher luminosity yields more experimental data, which is required for studying rare interaction phenomena between the particles.

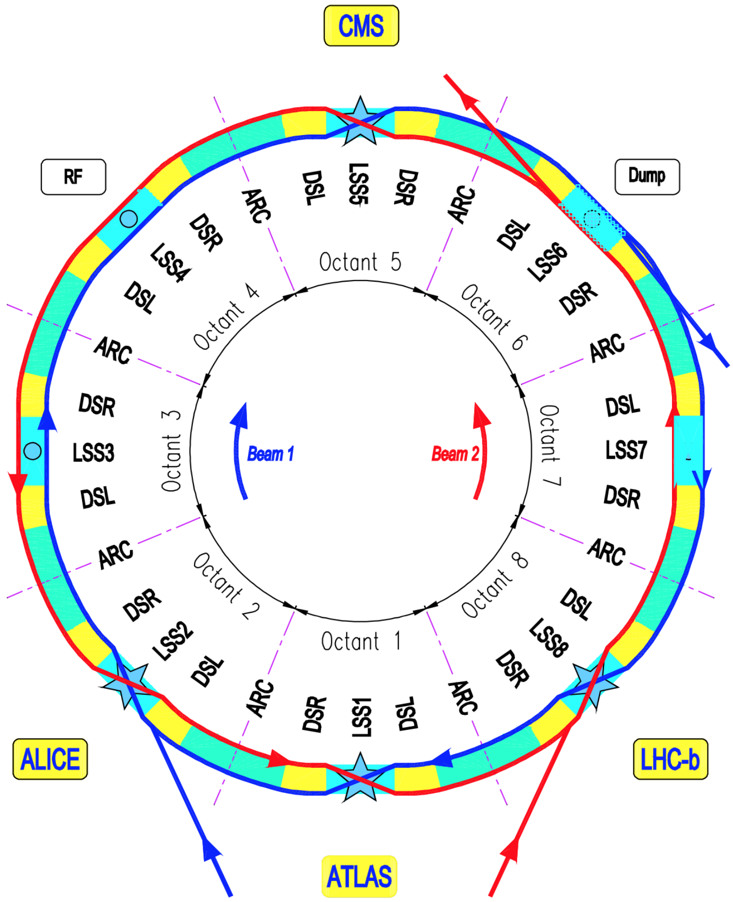


Figure 1.1.: Schematic layout of the LHC (modified [Brü+04]).

After a successful cycle of the LHC, or if an issue is detected, the particle beams are dumped. They are then diverted into two dedicated beam dump lines, where the beams are absorbed by carbon beam dump blocks. The beams are extracted from the main beam pipes into the beam dump lines using kicker magnets. This process is referred to as *beam dump*. In each beam, there is a gap that contains no bunches, with a width corresponding to the time it takes for the kicker magnets to reach their intended magnetic field strength. This gap is referred to as the *abort gap*, and ensures that the particle beams do not hit any equipment inside the beam pipes as they are being extracted. Nonetheless, absorbers (called TCDS and TCDQ) are fitted in the beam extraction regions in IR6, intended to protect sensitive machine equipment.

1.2. Beam losses in the Large Hadron Collider

The loss of particles from the beams inside the LHC is referred to as *beam loss*. Particles are lost when they hit the inner walls of the beam pipes or other equipment. Beam losses can occur for several reasons. Examples include imperfect vacuum, improper focusing or bending of the beams, and issues during the beam extraction process. Even though beam losses only result in a fraction of the beams being lost, they still present an issue for the LHC. The high-energy particles can damage equipment, potentially requiring repairment or replacement.

Beam losses in connection to beam dumps are of special interest. Historically, many beam dumps have been performed to protect machine equipment due to the detection of high beam losses in the LHC. Additionally, the beam dump process itself is prone to beam losses. For example, a high *abort gap population intensity* (a large number of particles in the abort gap) can result in beam losses when the magnetic field of the extraction kicker magnets is building up during the beam dump process.

1.2.1. Beam loss monitors

In order to monitor beam losses in the LHC, *beam loss monitors* (BLMs) are placed in connection to the beam pipes. BLMs are ionization chambers that detect beam losses through the radiation created by beam particles interacting with solid matter inside the LHC. BLMs are typically located just outside the beam pipes. Approximately 4000 BLMs are placed along the LHC ring. Sections of the LHC that are known to be prone to beam losses are fitted with more BLMs than other regions. For example, the concentration of BLMs is higher in the beam extraction regions compared to the arcs of the LHC. Figure 1.2 shows a picture taken inside the LHC tunnel, with BLMs visible on the outside of the LHC.

The two main types of BLMs are Ionization Chambers (ICs) and Little Ionization Chambers (LICs). The main difference between the two is that LICs are designed to observe very high beam losses without saturating. Therefore, LICs are generally placed in high-beam-loss areas. Usually, around 3700 ICs are installed in the LHC, while around 100 LICs are installed.

During operation of the LHC, BLMs continuously log the recorded beam losses. They are stored with a 40 μs integration time window, measured in bits per time window. The bit values are directly proportional to the radiation at the location of the BLM. From this basic integration time window, 12 *running sums* are constructed with different integration intervals. The running sum with the shortest integration time (and hence the

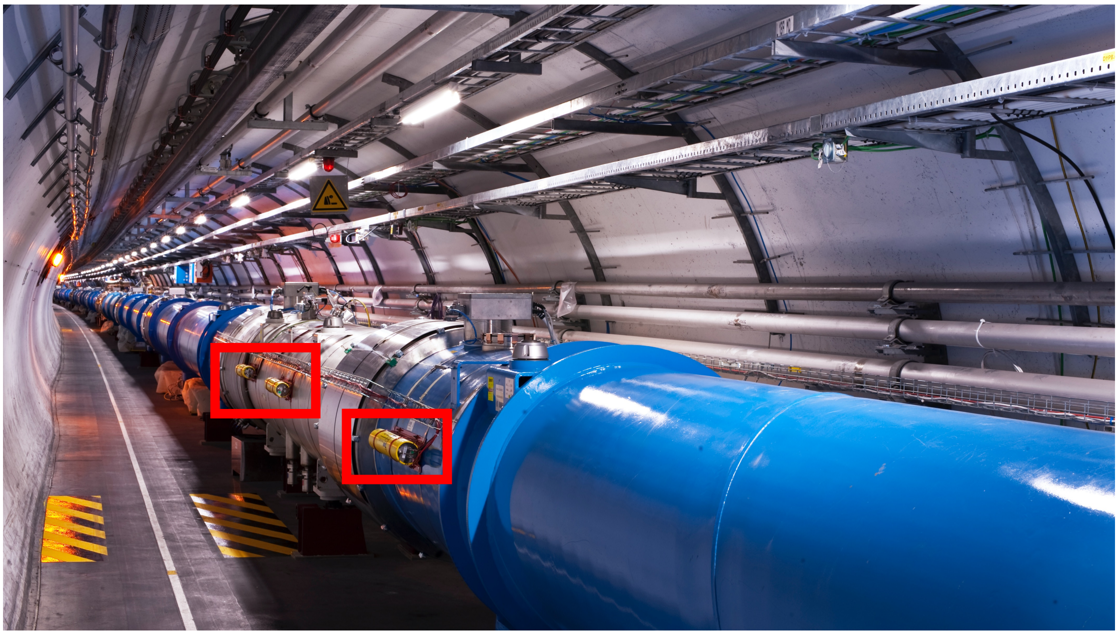


Figure 1.2.: Beam loss monitors (inside red rectangles) installed in the LHC (modified [Bri09]).

highest time resolution) is Running Sum 01 (RS01). When a beam dump is performed in the LHC, a time series with 25 600 RS01 values is recorded, for a total of 1.024 s of data around the moment of the beam dump. RS01 provides high time resolution that allows for detailed analysis of beam losses around the moment of a beam dump.

In addition to RS01, Running Sum 09 (RS09) is stored when a beam dump is performed. The integration time window for RS09 is 1.310 72 s, it holds 393 values, for a total of 515.112 96 s of data around the moment of a beam dump. This provides information about the more long-term development of beam losses before the beam dump.

When a beam dump is performed in the LHC, the RS01 and RS09 time series are saved to Post Mortem, a system with the purpose of collection and analysis of LHC data in connection to beam dumps [And+10]. The beam loss data is stored in 28 different *BLM crates*, with 256 slots each. One slot can store the data from one BLM. The crates roughly correspond to the spatial distribution of the BLMs inside the LHC, where groups of BLMs are connected by cables to physical crates that continuously log the beam loss data. This allows for a total of 7168 BLMs. However, since there are only around 4000 BLMs installed in the LHC at any given moment, around 3000 of the BLM slots in Post Mortem are unused.

In addition to beam losses recorded by the BLMs, a large number of other variables are recorded around the moment of a beam dump. These provide information about machine configuration settings, as well as measured properties such as beam intensity and beam width. All such recorded variables that contain information about the LHC at the moment of the beam dump are referred to as *context variables*, or *global features* in the setting of deep learning.

NXCALS is a logging database for LHC data [WR19]. Beam loss values are stored continuously with a 1 Hz frequency. Additionally, it stores context variables. APEX is yet another database, containing context variable data from historical beam dumps in the LHC.

In this work, an *anomaly* (an anomalous beam dump) is defined as a sample where the beam losses recorded by one or more BLMs have low conditional probability given the context variables (global features) at the time. For example, this could be cases where the overall sum of beam losses was unexpected, or cases where the distribution of beam losses between different BLMs was unexpected. Any sample that is not an anomaly according to this definition, is defined as "normal" or "not anomalous". It should be noted that for certain combinations of context variable values, very high beam losses are expected and should therefore not be considered anomalous.

1.3. Motivation for automatized beam loss analysis

For all high-energy beam dumps in the LHC, a post-mortem analysis is performed to allow early identification of potential issues. This includes manual evaluation of beam losses by a machine protection expert, which complements existing automatic system checks. The manual evaluation of beam losses can be tedious, and it is possible that issues are overlooked simply because they do not appear clearly upon inspection.

Automatizing the analysis of beam losses after high-energy beam dumps has several benefits. It makes the analysis more time efficient, it can potentially identify issues that are not directly apparent to machine protection experts, and it can contribute to the understanding of beam loss phenomena in the LHC. The classification of beam losses specifically in connection to high-energy beam dumps is of special importance due to the damage that the high-energy particles can cause.

This master's thesis explores two different approaches to automated detection and classification of beam loss anomalies in connection to beam dumps. Both approaches aim to incorporate the knowledge that LHC context variables have a direct influence on beam losses. Historical beam dump data from Run 2 of the LHC, spanning from 2015 to 2018, is used for the creation and validation of the methods of the two approaches. Only beam dumps using protons at top energy (6.5 TeV) are considered.

The first approach builds on BLM-wise modeling of beam losses as a function of some context variables, using understanding of physics phenomena in the LHC. This can be viewed as an extension of the existing BLM-wise beam loss thresholds used in automatic system checks, which do not take context variables into account. The BLM-wise modeling of beam losses and the overall classification of beam dumps make use of existing LHC expert knowledge.

The second approach uses deep learning to learn and extract relevant features from beam loss data and context variable data. The extracted features are used to classify beam dumps through the detection and classification of beam loss anomalies. Deep learning presents a versatile approach to the problem. It is especially suited for gaining insights from very large and complex datasets. It can identify patterns that would be difficult to discover using classical statistics techniques, potentially leading to discoveries of previously unknown beam loss phenomena in the LHC.

The main purpose of this work is to start the development of an automatized beam loss analysis tool that can be fully integrated into existing beam diagnostics systems. This requires intensive testing and validation of the methods.

1.4. Related work on data-driven analysis of beam losses and anomaly detection

Beam losses in the LHC have been studied since before the completion of the LHC. Some studies have focused on modeling and prediction of beam losses, based on LHC settings. In [ASV19], the relationship between collimator positions and beam losses observed by downstream BLMs is studied using a mathematical model of the spatial distribution of beam losses, showing good prediction results compared to experimental data. In [Neb+11], beam losses caused by foreign objects in the beam pipe are studied by investigating how the peak beam loss and beam loss duration depend on the beam energy and beam intensity. Beam loss patterns in space and time are studied in [Mar12] using singular value decomposition to reduce the dimensionality of beam loss data. The decompositions are compared to well-understood beam loss scenarios. It is concluded that it is often sufficient to only consider a small subset of BLMs to identify anomalous beam losses.

Other beam loss studies have focused on anomaly detection. In [Val+17], an unsupervised method based on principal component analysis and local outlier factor is successfully applied to spatial beam loss patterns in the LHC, conditioned on collimator positions. Beam losses in connection to special LHC magnet failure events are classified in [Obe+21] using a hybrid model combining existing beam loss thresholds and a support vector machine. The model is trained on historical data containing many different signals from LHC monitoring systems. The method is able to learn from expert decisions, and achieves a lower false positive rate compared to existing beam loss thresholds only.

The last decades have seen many papers on the general subject of anomaly detection, and more broadly, clustering. Novel techniques have successfully been applied in many different settings. Autoencoders (AEs) have been a popular choice for clustering algorithms. One example is Deep k-Means [FTG18], where the AE reconstruction losses and k-means clustering losses are optimized jointly. This method provides a versatile approach that performs well on both image data and text document data. Variational autoencoders (VAEs) have been popular for generative models. In [Dil+16] and [Jia+16], VAEs are combined with Gaussian mixture models to achieve good clustering results on different types of datasets. In addition, the models are capable of generating highly realistic samples. Different novelty measures are defined for VAEs in [Vas+18]. The proposed novelty measures are shown to perform well in varied settings. Certain novelty measures perform better on some datasets than others, requiring experimentation to find the most suitable one.

Conditional anomaly detection is studied in [He+20], where a set of known covariates is considered in addition to the main features. In order to maximize the dependence between the data and the clustering assignment, a novel method is proposed. The method introduces an extension of the Hilbert-Schmidt independence criterion and treats the variables in reproducing kernel Hilbert space. The projection of data and the clustering assignment are optimized alternately. The method is parameter-light, and the kernel measure makes no assumption about the data distribution.

1.5. Contributions

In this master's thesis we show that automatized analysis of beam loss data from high-energy beam dumps in the LHC can be performed using data-driven methods. In particular, we study the possibility of detecting beam loss anomalies, for the purpose of creating a tool that can classify and find beam loss anomalies in future beam dumps. The main contributions of this thesis are as follows:

- We develop a new method for classifying beam losses in high-energy beam dumps using ordinary least squares modeling for beam losses observed by individual BLMs. The classification results are validated by beam loss experts. The promising results demonstrate the capability of this method.

- We propose several extensions and improvements to the aforementioned method, based on the insights gained through the method development process.
- We propose and test a classification method using a deep neural network architecture for efficiently encoding and extracting features from beam dump data. We use this method to detect known types of beam loss anomalies in historical beam dump data, with promising results, showing the potential of the proposed neural network architecture for future studies.
- We propose an unsupervised deep learning method for the identification and classification of beam loss anomalies, based on the aforementioned neural network architecture. This method focuses on the detection and classification of novel beam loss anomaly types.

2. Methods

2.1. Data preparation

Relevant beam dump data was collected for beam dumps in Run 2 of the LHC. The data included beam loss data from BLMs and context variables. The collected data was inspected and validated, and data cleaning was performed to handle issues that were found. The cleaned dataset was then pre-processed for the purpose of making it directly compatible with deep learning. The following sections describe these steps in detail.

2.1.1. Data collection

The following sections describe how different data is collected from the different databases. Finally, the iterative data collection process, using the described data collection techniques, is outlined.

2.1.1.1. BLM data from Post Mortem

When querying beam loss data from a specific beam dump in Post Mortem, the full RS01 and RS09 time series for each BLM are initially retrieved. For the entire Run 2 of the LHC, this would result in a dataset size in the order of terabytes. To facilitate data storage and analysis, it was, thus, decided early on that the beam loss time series would be condensed to only a few selected features.

Using expert knowledge about beam loss time series in connection to beam dumps, the following basic features were selected for RS01: Maximum, sum, rise time and fall time. The rise time was defined as the time between first reaching 10 % of the maximum value to first reaching 90 % of the maximum value, with a corresponding definition for the fall time. The features were selected based on the most important characteristics of the RS01 time series when manually inspected by an LHC expert. In the following, the maximum value of a RS01 time series is referred to as `RS01_max`.

Further, it was decided to split the RS01 time series in two parts: Before the moment of beam dump, and after the moment of beam dump. This is because the RS01 time series typically have different characteristics before and after the beam is dumped.

The four basic features were calculated for the entire RS01 time series, as well as individually for the subseries before and after the beam dump, respectively, for a total of 12 RS01 features. For RS09, only the maximum value and the sum was calculated. Thus, in total, the 25 993 beam loss values from RS01 and RS09 were condensed to only 14 *BLM-wise features*. These features were calculated on the retrieved beam loss data, while the full RS01 and RS09 time series were discarded.

2.1.1.2. Context variables from Post Mortem, NXCALS and APEX

For the purpose of this analysis, 133 context variables were selected based on how relevant they were deemed for influencing beam losses in the LHC. These context variables were queried for all LHC beam dumps in Run 2. A comprehensive list of the selected context variables can be found in the Appendix. Out of the 133 selected context variables, 12 are stored in Post Mortem, 101 are stored in NXCALS, and the remaining 20 are stored in APEX.

It was verified that the data was available for the selected context variables and, if not, the context variables were replaced accordingly.

The values of some context variables from NXCALS are not available right at, or just before, the moment of the beam dump. Therefore, one has to query the last available value before the beam dump to get the most representative value. For context variables corresponding to measured quantities with a relatively large margin of error, the last available value before the moment of the beam dump is not very reliable. Instead, a more representative value of the context variable can be obtained by taking the average of a number of the last available values before the beam dump moment. The exact method of obtaining a representative value depends on the uncertainty and logging frequency of the measured quantity. Typically, the average of the 5 last available values were used for context variables with a logging frequency of 1 Hz, when collecting data for this analysis.

2.1.1.3. Additional variables from external tables

The beam dumps from Run 2 of the LHC have been screened and manually labeled for different phenomena and special events. Some of these were planned events, performed for research purposes, while others were unplanned phenomena that can sometimes cause beam losses and other issues. Below are descriptions of four types of such events or phenomena, that are of importance to this analysis.

- **Unidentified falling object (UFO)**: An object (dust particles or frozen gas) crosses the beam path, causing beam losses through scattering. The resulting beam losses can be very high, causing a protection dump of the beam.
- **Missing beam-beam kick**: During normal operation of the LHC, the two countercirculating beams exert a repelling force on each other, which is compensated for by corrector magnets. If one of the beams is dumped while the other keeps circulating, the remaining beam orbit will be overcorrected, causing a beam excursion [Lin+20].
- **10 Hz beam dumps**: On ten separate occasions during Run 2 of the LHC, oscillating and increasing beam losses were observed in the LHC, causing the beam to get dumped based on high beam losses observed by BLMs [WL]. This phenomenon has only been observed in Beam 1, and its cause is not fully understood.
- **Asynchronous beam dump tests**: An asynchronous beam dump is a failure scenario where the beam extraction kicker magnets fail to fire with proper timing [Wie+18]. This could lead to beam losses in the LHC extraction region and around the beam aperture. In order to validate the setup of the existing protection elements for such a failure scenario, artificial tests are regularly performed, where an asynchronous beam dump is performed on purpose but with very low beam intensity [BBF].

Information about the occurrence of these events or phenomena in the LHC during Run 2 has been compiled by experts in different tables. Information from these tables has been manually included in the high-energy beam dump dataset in the form of Boolean variables, indicating the presence of the corresponding phenomenon in a beam dump.

UFOs, asynchronous beam dump tests and 10 Hz beam dumps are well-known phenomena that can be considered beam loss anomalies. In the following, these are referred to as the 3 *well-known anomaly types*.

2.1.1.4. Iteratively collecting the data

A Python script was developed for collecting beam dump data from the sources mentioned above. The basis of the data collection procedure is the list of beam dump timestamps retrieved from APEX. Iteratively, each timestamps is used as parameter for queries sent to Post Mortem and NXCALS. Post Mortem is queried using an extension of the LHC Signal Monitoring API, which is a Python interface for querying LHC databases and processing data [And+]. Data from NXCALS is retrieved by connecting to an Apache Spark cluster. Once the data has been collected for a beam dump, it is appended to the already collected data in a CSV file. Finally, the information from the additional variables described above is added to the dataset.

2.1.2. Data inspection and validation

2.1.2.1. Comparing beam loss data from Post Mortem and NXCALS

In order to verify that the stored beam loss data is identical in Post Mortem and NXCALS, beam loss data was queried from both systems for a number of randomly selected beam dumps from the year 2018. The beam loss values were then compared BLM by BLM. Since NXCALS stores RS01 beam losses in units of Gy/s, while Post Mortem uses bit/(40 μs), the beam loss data from the two sources need to be converted before comparison. The conversion factors for ICs and LICs are listed in Table 2.1.

Table 2.1.: Conversion factors between [bit/40 μs] and [Gy/s] (values taken from NXCALS).

IC	3.62×10^{-9} Gy/bit	(years of 2015-2018)
LIC	2.17×10^{-7} Gy/bit	(year of 2015)
LIC	5.07×10^{-8} Gy/bit	(years of 2016-2018)

For this comparative analysis, only `RS01_max` values of 100 bit/(40 μs) were considered. The results showed that a majority of ICs had identical `RS01_max` values in NXCALS and Post Mortem, within the uncertainty of the conversion factor.

Detailed analysis showed that a different conversion factor, 2.17×10^{-7} Gy/bit, had to be used for LICs for beam dumps from the year 2015, in order to obtain the same results expressed in Gy/s as in NXCALS. The conversion factor 5.07×10^{-8} Gy/bit was used for LICs for beam dumps from the years 2016 to 2018.

A discrepancy between NXCALS and Post Mortem was found in beam loss data from LICs from the year 2016. After a meeting with BLM experts, it was concluded that this discrepancy was due to an incorrect conversion factor used solely in NXCALS for beam dumps between March and November of 2016. This did not affect the data from Post Mortem, and it could be used with the same conversion factor.

The ICs and LICs whose beam losses did not agree between Post Mortem and NXCALS, despite taking into account the changing conversion factor for LICs, were further analyzed. It was found that no BLMs had systematic differences over several beam dumps. Additionally, the discrepancies affected different BLMs from beam dump to beam dump. A probable cause for the discrepancies is misaligned time integration windows.

It was decided that beam losses in NXCALS and Post Mortem agreed sufficiently to use data from only one of them for this analysis. Post Mortem is believed to be the most credible source, due to its higher logging frequency compared to NXCALS. Hence, only beam loss data from Post Mortem was used for the following analysis.

2.1.2.2. BLM data consistency check

As a consistency check, the `RS01_max` values of all BLMs in the retrieved dataset were plotted, BLM by BLM, as a function of time over the year of 2018 to see if any major changes had occurred during that time period. A distinct jump in the `RS01_max` value of a BLM in a plot could indicate that the BLM had been moved to another location at that point in time, or that another machine element had been installed in the vicinity of the BLM, causing different beam loss characteristics in that region. No such jumps were found in any of the plots.

The plotting of `RS01_max` over time for each BLM also served as a way to get familiar with the beam loss data. Figures 2.1—2.3 show examples of BLMs that observe moderate or high beam losses. An explanation of the BLM naming convention can be found in Appendix A. Figure 2.4 and Figure 2.5 also confirm that ICs and LICs saturate at the same beam loss level, when expressed in bits.

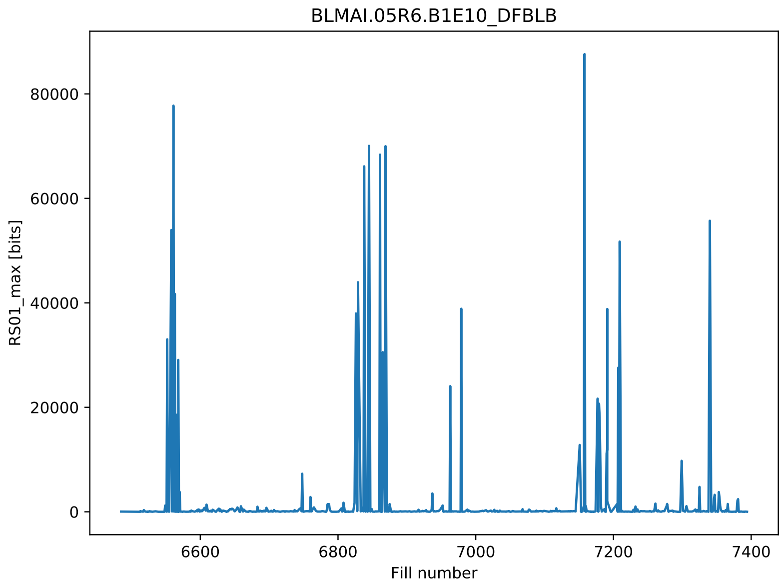


Figure 2.1.: Example of beam losses (`RS01_max`) observed by an IC BLM with generally low beam losses, but with large spikes throughout the year of 2018. This type of pattern over time is typical for many BLMs.

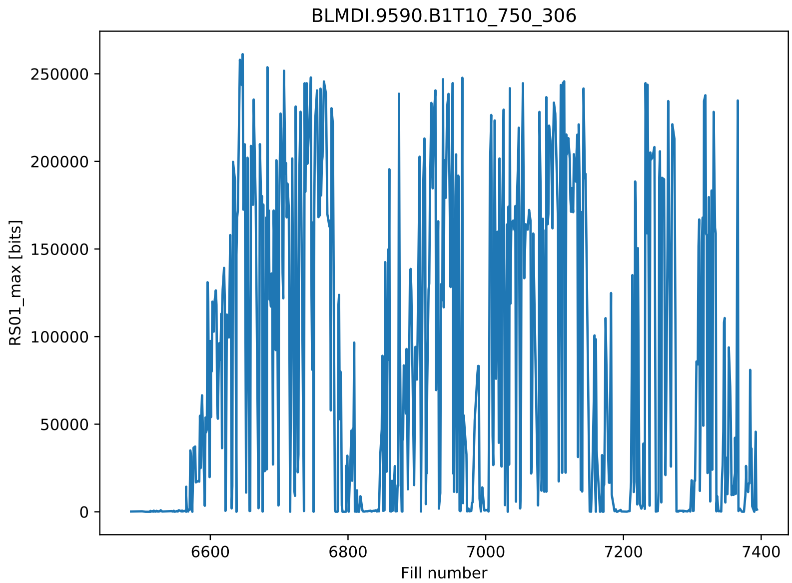


Figure 2.2.: Example of beam losses (`RS01_max`) observed by an IC BLM, located in beam dump line 1, with high beam losses in the year of 2018. This type of pattern is typical for BLMs located in the beam dump lines.

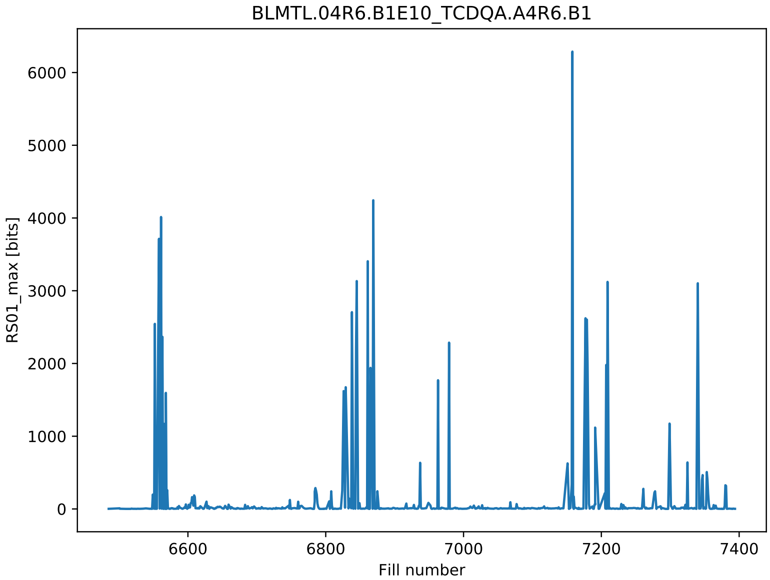


Figure 2.3.: Example of beam losses (`RS01_max`) observed by a LIC BLM with generally low beam losses, but large spikes throughout the year of 2018. Some BLMs only rarely observe considerable beam losses, with this pattern being a typical example.

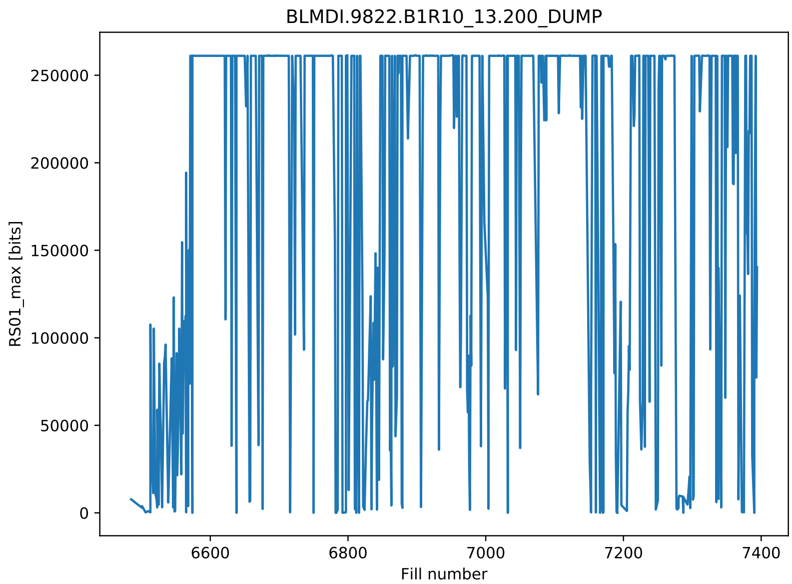


Figure 2.4.: Example of beam losses (`RS01_max`) observed by an IC BLM, located in beam dump line 1, that frequently saturated in 2018. There are often high beam losses in the beam dump lines, causing saturation of some of the BLMs located there. The effect of saturation can clearly be seen in this pattern over time as clipped values for very high beam losses.

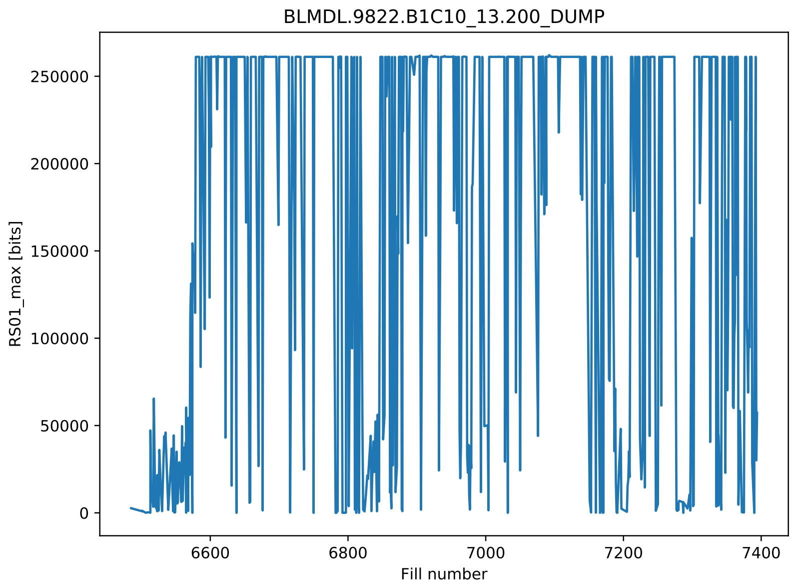


Figure 2.5.: Example of beam losses (`RS01_max`) observed by a LIC BLM, located in beam dump line 1, that frequently saturated in 2018. Despite LIC BLMs being designed to observe very high beam losses without saturating, some of them still saturate frequently. The effect of saturation can clearly be seen in this pattern over time as clipped values for very high beam losses.

2.1.2.3. Defining the moment of beam dump

For every beam dump in the LHC, the RS01 loss series are aligned such that the moment of beam dump happens at approximately 1.018 s into the RS01 time series. However, there is some uncertainty in this number, stemming from the timing of electronics in the LHC, and other factors. Therefore, it is not a priori possible to know exactly at what time index in the RS01 time series the beam dump happened, for a given beam dump.

In general, most BLM observe the highest RS01 values exactly at, or just after, the moment of the beam dump. Using this knowledge, one can define the moment of the beam dump as the most common time index for `RS01_max` for all BLMs and all beam dumps in Run 2. Figures 2.6—2.7 show the resulting histograms. As expected, the highest peak is located close to the end of the RS01 time series interval. Unexpectedly, there was a second peak visible at about 0.6 s. Investigating it showed that it was the result of faulty (unphysical) beam loss values from mobile BLMs (BLMMs). Figure 2.8 shows an example of such a faulty RS01 time series, with a jump at approximately 0.6 s, just like the smaller peak in Figure 2.6.

Discarding unphysical beam loss values that exceeded the saturation level of BLMs, the corrected histograms are shown in Figures 2.9—2.10. The peak is located at 1.018 84 s. It can be seen that there were few instances where a BLM had `RS01_max` index before 1.017 76 s. Approximately 17 % of instances occur before or at this index, while approximately 83 % of instances occur after. It is believed that most of the instances before index 1.017 76 s were not directly due to a beam dump.

For the purpose of separating the RS01 time series into two subseries, the index 1.017 76 s was chosen as artificial definition of the moment of beam dump, as the real moment of beam dump happened directly after that in most instances.

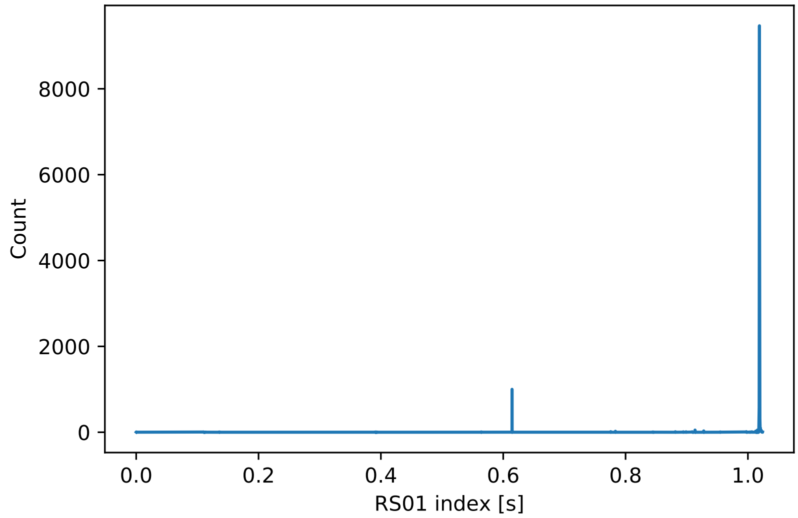


Figure 2.6.: Histogram of the time index of `RS01_max` from all RS01 time series in the beam dump data from the year of 2018, including data from mobile BLMs. The largest peak at about 1.0 s is located near the moment of the beam dump, while the smaller peak at about 0.6 s is unexpected.

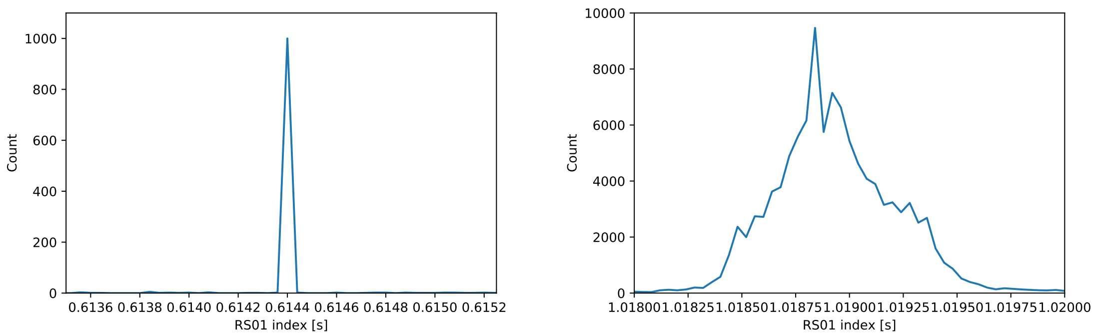


Figure 2.7.: Zoom of the two peaks visible in Figure 2.6. The left panel shows that the values around the left peak are concentrated to a single time index. The right panel shows a larger spread around the most common time index for the right peak.

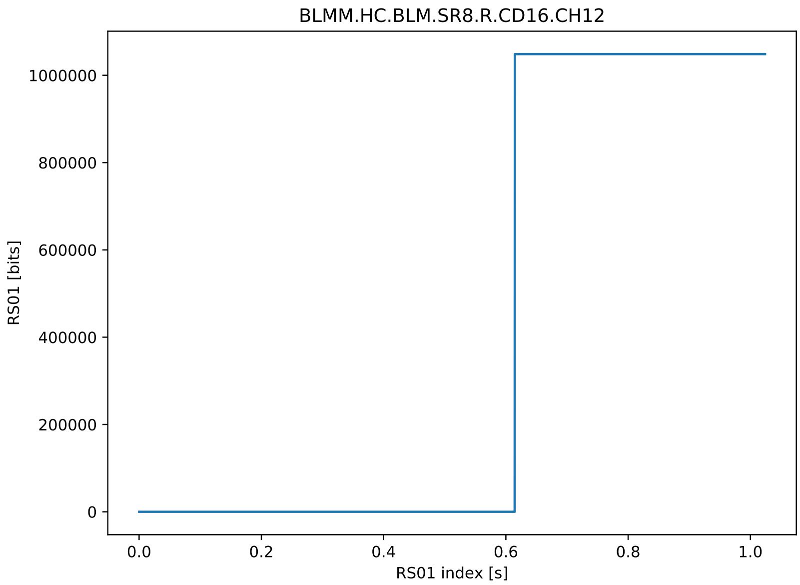


Figure 2.8.: Example of the full RS01 time series of a mobile BLM that contains unphysical beam loss values starting at approximately 0.6 s. This indicates the presence of faulty values that need to be excluded from the analysis.

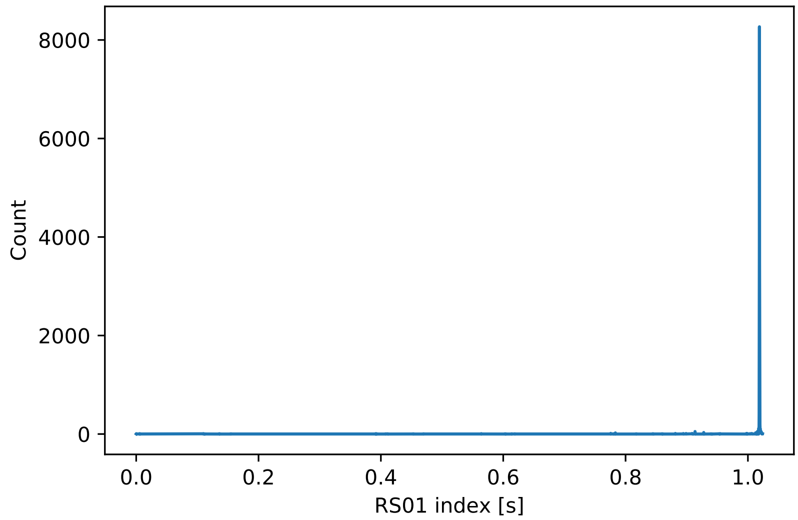


Figure 2.9.: Histogram of the time index of `RS01_max` for all RS01 time series in the 2018 beam dump data, excluding unphysical beam loss values. Compared to Figure 2.6, only the peak at about 1.0 s remains, indicating that the smaller peak in Figure 2.6 was caused by unphysical beam loss data from mobile BLMs.

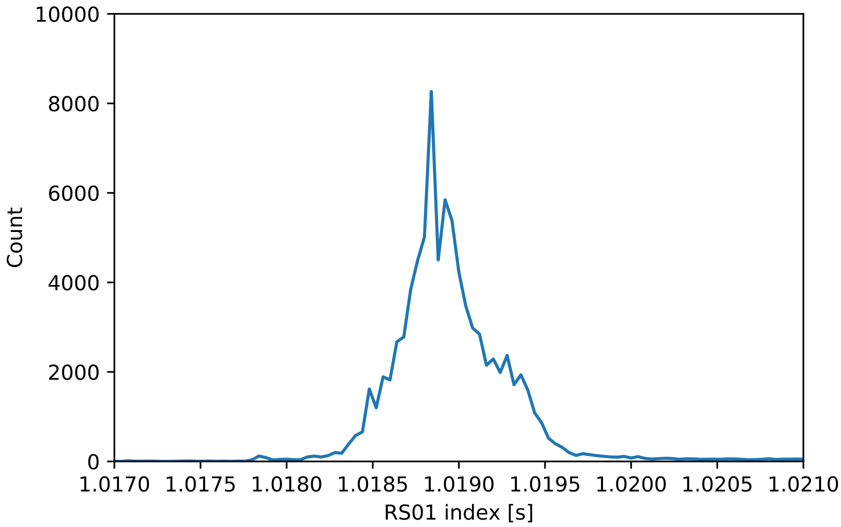


Figure 2.10.: Zoom of the peak in Figure 2.9. Most values are located to the left of time index 1.017 76 s, indicating that this could serve as a lower bound for the actual moment of the beam dump. This time index can then be used to artificially define a time index for the moment of the beam dump.

The same analysis was repeated, but instead of grouping all BLMs together, it was done separately for each BLM. Figure 2.11 shows typical examples. Comparing these four histograms, one can see that the most common time indexes for `RS01_max` differ between them. In addition, there are different degrees of variation around the most common value. This shows that the real moment of beam dump tends to be closer to the artificially defined moment beam dump more frequently for some BLMs than for others.

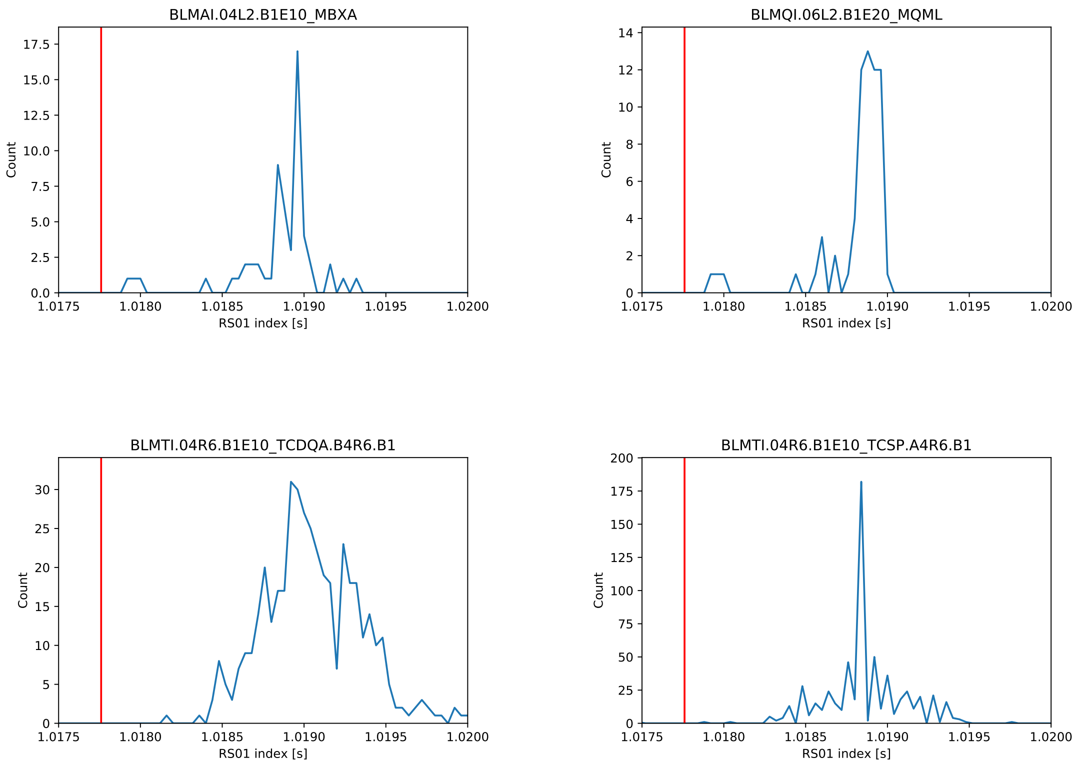


Figure 2.11.: Examples of distributions of most common time indexes for `RS01_max` for four separate BLMs. Both the range and the spread of the `RS01_max` indexes differ from BLM to BLM, showing that the artificially defined moment of beam dump is more suitable for some BLMs than for others. The red vertical line marks the defined moment of beam dump.

2.1.2.4. Beam loss time series visualizations

Examples of RS01 beam loss curves, for different BLMs from selected beam dumps, are shown in Figures 2.12—2.16. The left figure panels show the entire RS01 time series, while the right figure panels show a narrow window around the beam dump moment. The artificially constructed moment of beam dump is marked by the vertical red line in the right figure panels.

Figure 2.12 shows no beam losses before the moment of beam dump, followed by a rapid rise in beam losses in connection to, and after, the beam dump.

BLMs located in high beam loss regions of the LHC sometimes observe beam losses that exceed their saturation level, resulting in a capped RS01 curve, as seen in Figure 2.13.

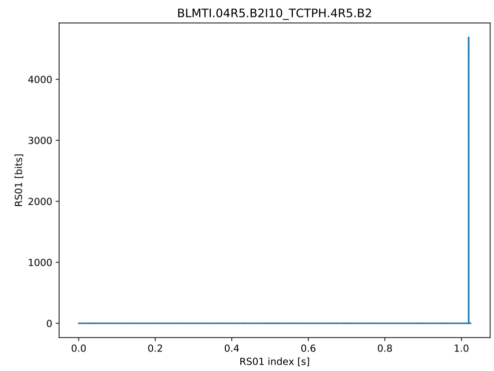
In order to mitigate the effects of saturating BLMs, some BLMs located in high beam loss regions are equipped with electronic filters that integrate the recorded beam losses over a longer time period. This affects the appearance of the RS01 curve, with a resulting long fall time, as seen in Figure 2.14.

Figure 2.15 shows beam losses that are increasing and oscillating with a main frequency around 10 Hz, similar to the phenomenon that caused 10 protection beam dumps during Run 2 of the LHC.

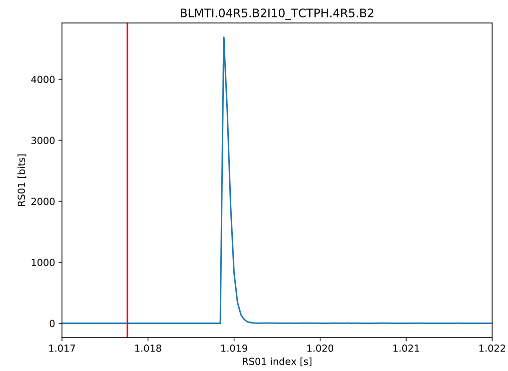
Even with no beam losses present, BLMs always record a few bits of noise, usually below 10 bits. In fact, the majority of BLMs in the LHC rarely observe any considerable beam losses, so their RS01 beam loss curves typically look the one depicted in Figure 2.16.

2.1.2.5. Very low beam losses

Approximately 90 % of BLMs essentially always record `RS01_max` values in the order of 0—100 bit/(40 μs). Beam losses on this scale are prone to statistical fluctuations and noise, stemming from (electrical noise and other (external sources)). It is entirely possible that a recorded `RS01_max` value of 80 bit/(40 μs) is entirely caused by such noise, without any real measurable beam losses in that location. Therefore, `RS01_max` values below approximately 100 bit/(40 μs) should either be excluded from the analysis, or be given only a small influence. Further, the lowest beam loss value considered is referred to as the *noise level*.

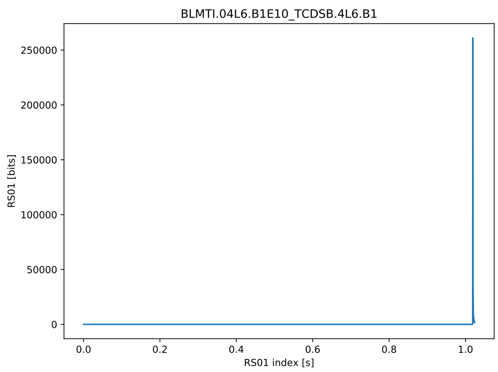


(a) Full RS01 time series. No beam losses (i.e., no peaks) before the beam dump, which occurs at around 1 second.

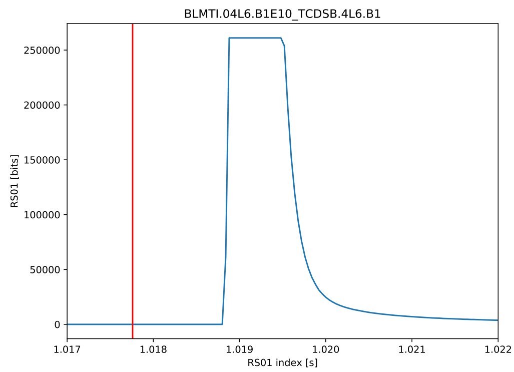


(b) Zoom around the moment of the beam dump. A sharp beam loss spike is visible. The red vertical line marks the defined moment of beam dump.

Figure 2.12.: An example of a typical RS01 curve with no considerable beam losses before the beam dump, and a sharp peak around the moment of beam dump.

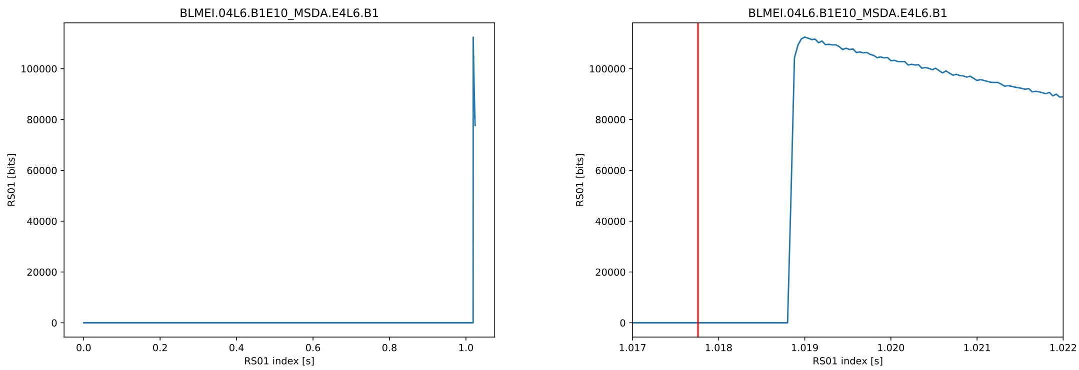


(a) Full RS01 time series. No considerable beam losses before the beam dump.



(b) Zoom around the moment of the beam dump. The beam losses saturate at approximately 261 100 bits.

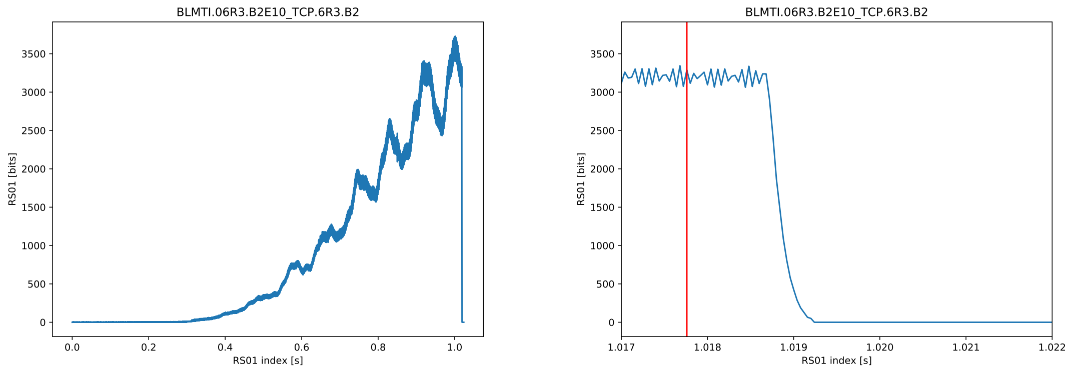
Figure 2.13.: Example of an RS01 curve for a saturating BLM, showing capped beam loss values at the BLM saturation level. This specific BLM is located close to a TCDS absorber, where high beam losses are common.



(a) Full RS01 time series. No considerable beam losses before the beam dump.

(b) Zoom around the moment of the beam dump. After the beam is dumped, the RS01 value drops slowly due to an integrating electronic filter installed in this specific BLM.

Figure 2.14.: Example of an RS01 curve for a BLM with an electronic filter, showing sustained high bit counts (recorded beam losses) after the beam dump. Some BLMs have an electronic filter installed that integrates beam losses over a longer time to avoid saturation of the BLM. These BLMs have different beam loss curves than BLMs without electronic filters.



(a) Full RS01 time series with rising beam losses. In this instance, oscillations in the order of both 10 Hz and 10 000 Hz are visible.

(b) Zoom around the moment of the beam dump. The beam losses drop rapidly as the beam is dumped.

Figure 2.15.: Example of an RS01 curve showing beam losses that slowly build up before the beam is dumped, with a distinct pattern visible. Oscillating, rising beam losses like this have triggered beam dumps in the past. The cause of the rising beam losses of this type is not well-understood.

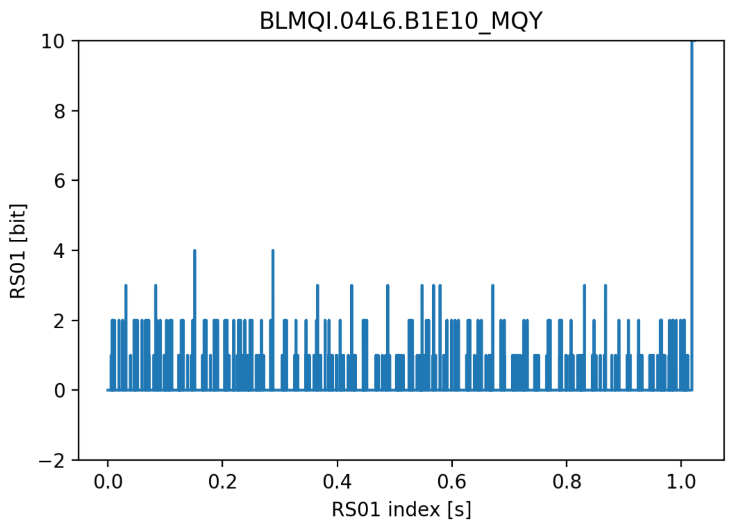


Figure 2.16.: An example of a noisy RS01 curve with no considerable beam losses. RS01 time series like this are typical for a majority of BLMs.

2.1.3. Data cleaning (based on insights from inspection and validation)

Data cleaning was performed to handle the data issues discovered during inspection and validation. The following sections describe the main steps of the data cleaning.

2.1.3.1. Ensuring correct data types

The first step of cleaning the data consisted of ensuring that the collected features were of the expected data types, e.g., integers, floating point numbers, and strings. It was found that in some instances, unexpected data types had been used to mark the absence of data, such as "-" or "UNDEFINED". All such values were removed from the dataset.

2.1.3.2. Undefined beam dumps

In 32 of the retrieved beam dumps, the field `Event Category` was not consistent with the expected values of "PROTECTION DUMP" or "PROGRAMMED DUMP". These beam dumps were removed from the dataset, along with any beam dumps that had undefined `Beam Mode`. The removed beam dumps had in common that they all had happened during periods of machine development.

2.1.3.3. Undefined fill number

Already during data collection, it was noticed that the querying of context variables from Post Mortem was failing for 12 timestamps from the year 2018. Upon further inspecting, it turned out that all of these have undefined `Fill Number`, and were labeled as "timing events". Additionally, their timestamps were within two minutes of the preceding beam dumps, and most context variable values were the same as the preceding beam dumps. A beam dump expert confirmed that such "timing events" occur in the database when some beam dump information is added after the initial logging, such as when the beam operator decides to change the description of the beam dump. It was, thus, concluded that these event were not real beam dumps, and hence they were removed from the dataset.

2.1.3.4. Beam dumps in 2015

As the first operational year of Run 2 of the LHC, 2015 had an unusually large amount of testing and commissioning. This means that a considerable fraction of the recorded beam dumps of 2015 contain machine settings and machine behavior that are not representative of the following operational years of Run 2. While, in principle, it is possible to manually identify and remove such non-representative beam dumps from the dataset, it would be a tedious process that is prone to errors. Additionally, the chosen selection method could introduce biases in the resulting data set. Therefore, it was decided to completely exclude the beam dump data from 2015 from this analysis.

2.1.3.5. Corrupted BLM cards

In the initial data inspection and validation, it was noticed that an unexpectedly large fraction of `RS01_max` values were either exact powers of two, or close to powers of two. Upon closer inspection, it became clear that not all BLMs exhibited such behavior. Instead, only a fraction of all BLMs were responsible for the high frequency of powers of two in the `RS01_max` data. For each such BLM, the corresponding `RS01_max` values were plotted as a function of beam dump number in the year of 2018. Some typical plots are shown in Figures 2.17—2.22. The powers of two are visible as spikes or plateaus in the plots, and it is clear that these values are not a result of normal beam loss behavior in the LHC. This is especially evident in cases where the `RS01_max` value exceeds the saturation level of the BLM, which is unphysical. Additionally, it was noted that these suspicious values were not contained to only a few beam dumps. Instead, they seemed to occur in different beam dumps for different BLMs.

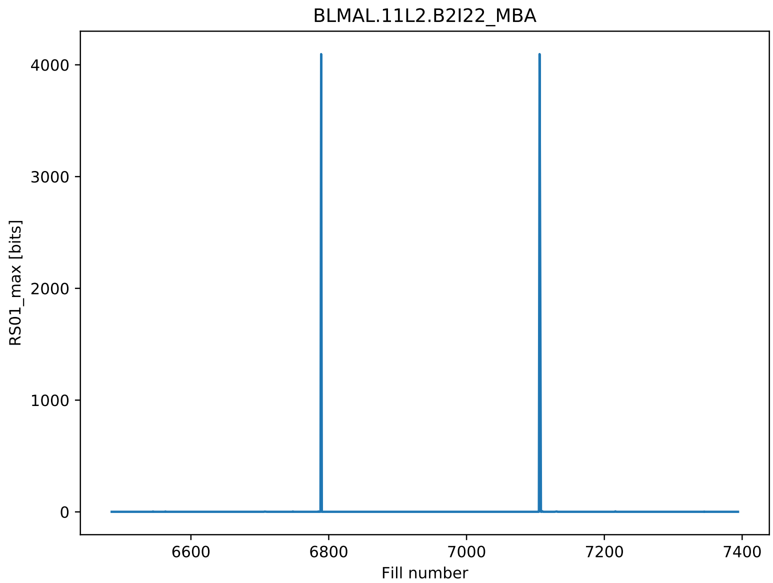


Figure 2.17.: Beam losses (RS01_max) observed by a BLM with peaks at 4096 bits at two different points in the year of 2018. The two outlier values are powers of two, and could indicate faulty values.

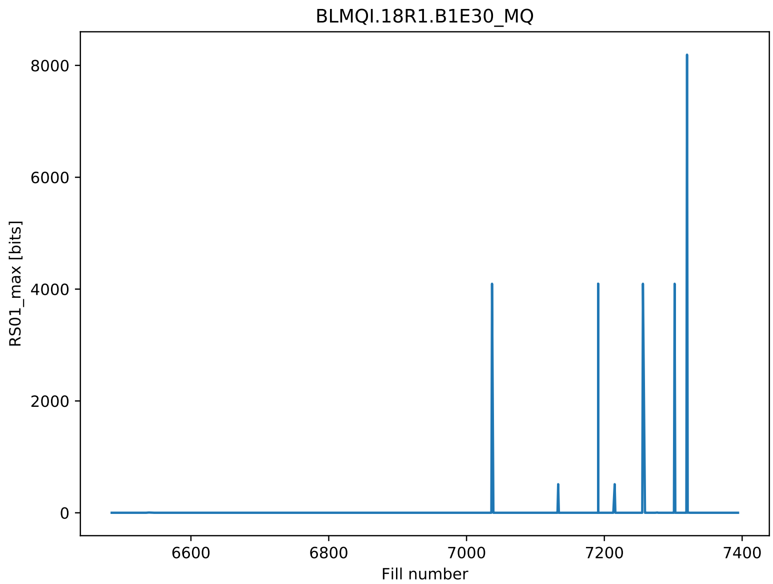


Figure 2.18.: Beam losses (`RS01_max`) observed by a BLM with one peak at 8192 bits, four peaks at 4096 bits, and two peaks at 512 bits. The peak values are powers of two, and clearly stand out from the rest of the data, indicating that these values might be faulty.

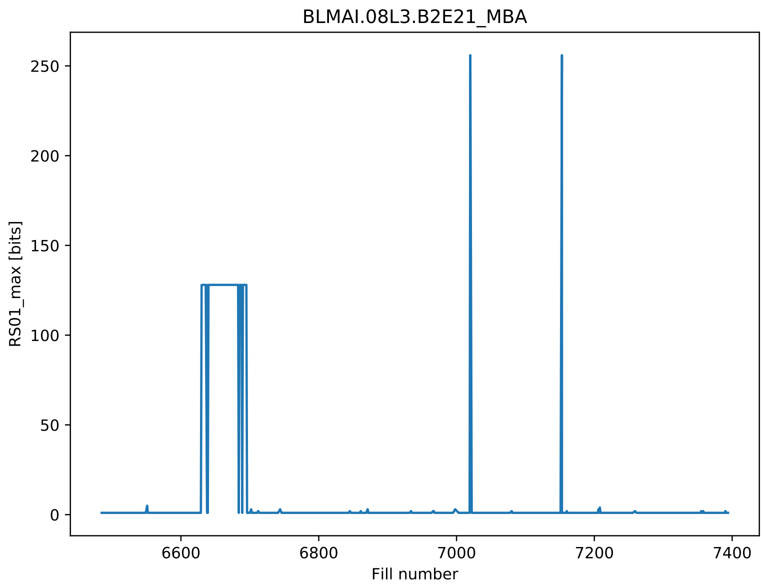


Figure 2.19.: Beam losses (`RS01_max`) observed by a BLM with several consecutive values of 128 bits, and two separate instances at 256 bits. It is very unlikely that a BLM would observe the same `RS01_max` value above 100 bits several beam dumps in a row. This is a strong indication of faulty beam loss values in this BLM.

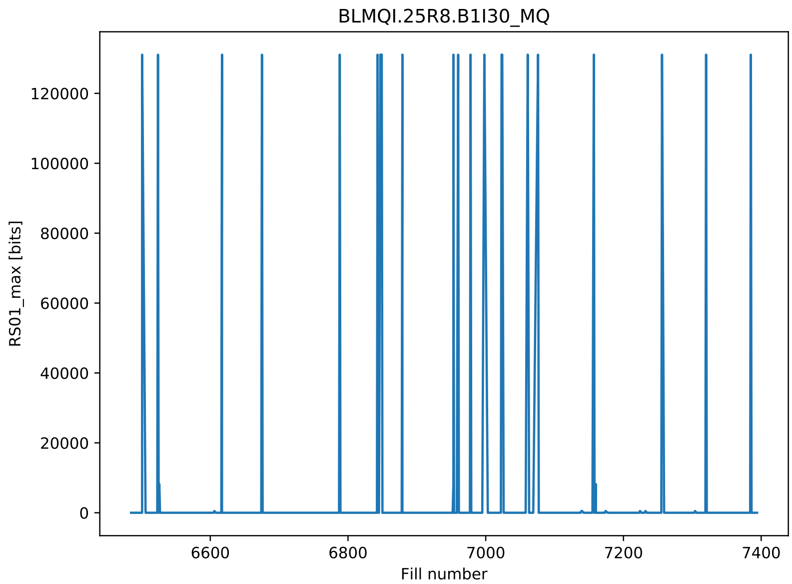


Figure 2.20.: Beam losses (`RS01_max`) observed by a BLM with several distinct spikes at 131 072 bits. The peaks are clear outliers, and they are all the same power of two. Beam loss data like this is completely improbable to be an accurate representation of the physical beam losses in vicinity to this BLM.

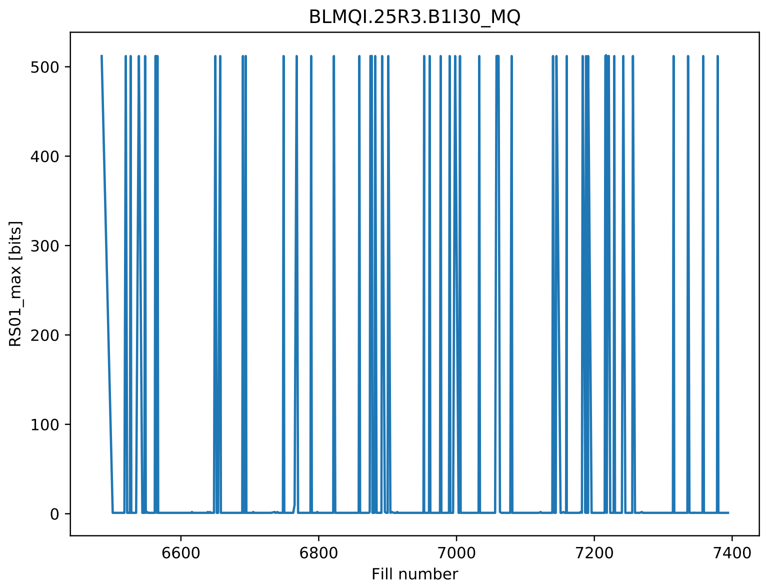


Figure 2.21.: Beam losses (`RS01_max`) observed by a BLM with several distinct spikes at 512 bits, and a single spike at 513 bits. Outlier beam loss values that are close to powers of two, like 513, are common in the original dataset. In this case, it is not fully clear whether the spike at 513 bits is a faulty value or not.

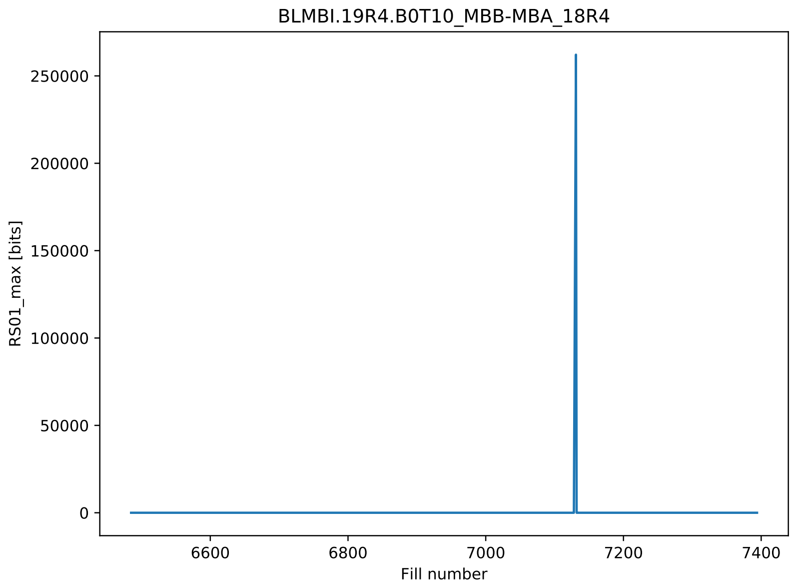


Figure 2.22.: Beam losses (`RS01_max`) observed by a BLM with a single peak at 262 144 bits, which is deceivingly close to, but above, the BLM saturation level of approximately 262 100 bits. The outlier value is unphysical, in addition to being a power of two, indicating that it is a faulty value.

For some of the instances where BLMs recorded RS01_max as exact powers of two, or neighboring values, the whole RS01 time series was plotted, as in Figure 2.23. What all these instances had in common was that the peak RS01 value had been reached at an unexpected time index, not in connection to the moment of beam dump itself.

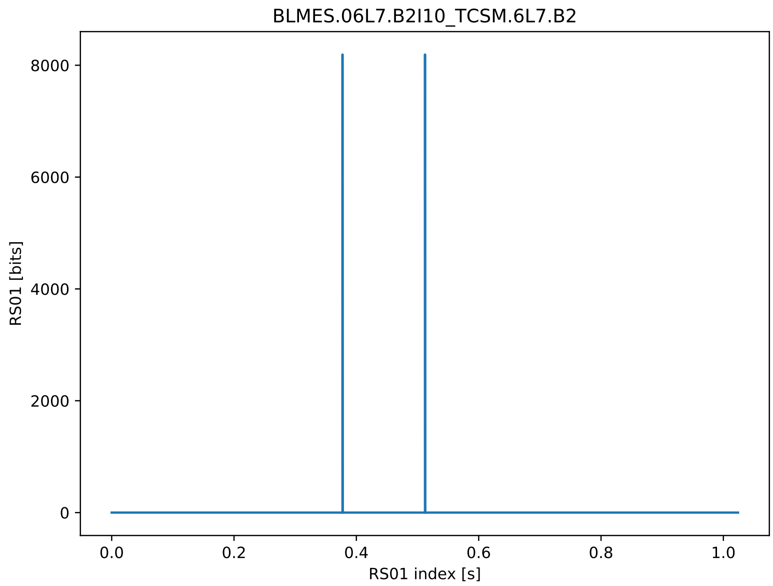


Figure 2.23.: Example of an instance where a BLM recorded an RS01_max value of 8192 bits, which occurred at two different points in the time series. These values are outliers in both absolute value and time index, suggesting that these could be a faulty values.

After an extensive investigation, including debugging the scripts for data querying and re-collecting the data set, the suspicious values still remained. After a meeting with BLM experts where these findings were discussed, it was hypothesized that the suspicious values were indeed faulty, and that they were the result of data corruption in the data collection cards in the BLM crates.

Together with the BLM experts, a strategy for removing faulty beam loss values resulting from corrupted BLM data collection cards was devised. After every beam dump in the LHC, all BLM cards that contain corrupted data are autoamtically marked with an unphysically large beam loss value for one of the BLMs in that card. An example of such a marker value is $1\,048\,575 = 2^{20} - 1$. Hence, by looking for such unphysically large marker values in the BLM data from a beam dump, and discarding the data belonging to all BLMs in the affected data collection card for that beam dump, one should get rid of all corrupted data.

A script that discards the faulty values according to the procedure described above was run on the dataset, removing corrupted BLM data. The results of applying the script to the dataset containing only beam dumps from 2018 are shown in Table 2.2, where the overrepresentation of powers of two is clearly visible, in particular for 2048 and 2049 bits, when compared to neighboring values. It can be seen that the filtering is far from solving the problem of having too many powers of two.

While the procedure managed to get rid of 299 of the approximately 1500 suspicious values in the 2018 dataset, the majority still remained. Evidently, data corruption in the BLM cards was not the only explanation behind the overrepresentation of powers of two in the BLM data. After another meeting with BLM experts, it was concluded that the remaining suspicious values were probably caused by digital binning. When beam losses are continuously measured by a BLM during operation, the beam losses are measured as a voltage, directly proportional to the radiation at the location of the BLM. This voltage reading is converted to a measurement in bits, which is recorded. The conversion process from voltage to bits contains rounding errors, such that voltages that would correspond to bit values close to powers of two are recorded as exact powers of two. Hence, despite containing rounding errors such that bit counts close to powers of two are likely to be rounded to exact powers of two, these values correspond to real beam losses in the LHC and should be kept in the dataset.

2.1.3.6. Unphysical values

Throughout the analysis, additional unphysical values were discovered in the context variables of the dataset. One example is the maximum abort gap population intensity. Typical values for this context variable are in the order of 10^7 to 10^{11} protons, never exceeding 2.2×10^{12} protons. In a few cases, this context variable took on values in the order of 10^{20}, which is clearly erroneous. The cause of these values was not investigated further, and the unphysical values were filtered out by simple checks in the subsequent analysis.

2.1.3.7. Saturated values

All BLMs in this analysis saturate at radiation levels corresponding to measured beam losses of approximately 261 100 bit/(40 μs). Beam losses higher than that would yield

Table 2.2.: RS01_max count for selected powers of two and their neighboring values in the 2018 dataset, before and after filtering out data from corrupted BLM cards. Decreased counts are shown in bold. A clear overrepresentation of exact powers of two is visible, even after filtering.

RS01_max [bit]	Count before filtering	Count after filtering
255	114	114
256	263	**261**
257	90	90
511	29	29
512	273	**269**
513	36	36
1023	18	18
1024	52	52
1025	12	12
2047	6	6
2048	125	**30**
2049	177	**8**
4095	1	1
4096	249	**248**
4097	7	7
8191	0	0
8192	287	**264**
8193	1	1
16 383	0	0
16 384	17	17
16 385	1	1
32 767	1	1
32 768	18	18
32 769	0	0
65 535	0	0
65 536	33	33
65 637	0	0
131 071	0	0
131 072	244	**239**
131 073	3	3

the same value, and there is no direct way of knowing how high the beam losses actually were. Including saturated values in the data set can be problematic, since the real beam losses could have been considerably higher than what the data indicates. The saturated data points could skew correlation analyses and influence other statistical figures, which could lead to inaccurate conclusions. However, completely discarding saturated values would mean a loss of valuable information, since knowing that a BLM was saturated is valuable in itself, without one having access to the exact beam loss value.

In the subsequent analysis, saturated beam loss values are either discarded completely, or kept in, depending on use case. In the cases where the values are kept, an additional variable is added for each BLM, that indicates whether the BLM was saturated or not. In order to know if a BLM was saturated during a specific beam dump, it is sufficient to check whether `RS01_max` is at the saturation value, instead of analyzing all 14 beam loss features.

2.1.3.8. BLM name changes

Early into the initial data analysis after querying the beam dumps data for Run 2 of the LHC, it was noted that data from a number of BLMs only existed for a subset of the beam dumps, often only in the beginning or in the end of Run 2. BLM experts confirmed that this is because several BLMs were either removed, installed or had their names changed during Run 2. In the case of name changes, a single BLM that changed name will manifest itself as two different BLMs in Post Mortem, when looking at the Run 2 data set.

In most cases, the BLM name changes took place only because of a newly introduced naming convention, and the BLM itself was unchanged. Hence, the data sets for those two BLM names can safely be merged into one. However, in a few cases, BLMs were either moved to another position, or had other machine hardware installed in their proximity. In those cases, the new position or the newly installed hardware could potentially result in the BLM registering higher or lower beam losses than before. Hence, it is not clear whether the BLM before the change and after the change should be treated as two separate BLMs.

A new naming convention had caused name changes in a total of 1084 BLMs. These BLMs had their beam loss data re-mapped accordingly, such that the data belonging to the old name was merged with the data belonging to the new name, and saved under the new name. In order to ensure that no data was mapped incorrectly, `RS01_max` was plotted against fill number for each of the merged BLM data series. An incorrect mapping would manifest itself as a distinct jump in the plot at the time (fill number) when the name change happened. No such jumps were visible in the plots, and therefore it was concluded that all BLM loss data had been correctly mapped to their respective BLM.

2.1.3.9. Merging BLM XPOC check variables

For redundancy, the context variables corresponding to BLM XPOC checks were retrieved from both APEX and Post Mortem. When inspecting the data, it was found that some beam dumps had missing BLM XPOC check data in one of the data sources, but not the other. Apart from the missing values, none of the BLM XPOC check data was contradicting. The data from the two data sources was merged as the union of the two.

2.1.4. Overview of the cleaned dataset

The cleaned dataset consists of data from 1079 beam dumps from the period of 2016-2018 (inclusive). For each beam dump, 14 BLM-wise features are available for each of the 3805 BLMs. Additionally, 219 context variables and other labels are available for each beam dump.

2.1.5. Pre-processing the dataset for deep learning

The cleaned dataset is not directly suitable for deep learning tasks. For instance, missing values, categorical features and imbalanced data distributions present issues that need to be addressed before initiating deep learning tasks. The steps of pre-processing the dataset before deep learning tasks are outlined below.

2.1.5.1. Excluding specific features

Some of the context variable features included in the cleaned dataset are difficult to assign numerical values to. Examples of such features are the features `Mps Expert Comment`, `Operator Comment` and `Pm Analysis Result Description` that contain comment strings. These features can provide useful information for validating classification results, but can not be used together with numerical model-fitting methods. Therefore, they were excluded from the dataset for the purpose of deep learning.

Context variable features corresponding to timestamps could potentially confuse deep learning models. The timestamp features correspond to timestamps in close connection to the moment of the beam dump, such as the timestamp of the abort gap. Hence, the timestamp features essentially only contain information about when the beam dump happened. It is known that there are clear patterns in the data over time, since the LHC settings of two consecutive beam dumps often are similar. Such patterns in time are decidedly not relevant for this analysis, and any context variable features that indicate timestamps, including `Fill Number`, were excluded.

In the end, 207 context variables remained, to be used in the deep learning experiments.

2.1.5.2. Encoding categorical features and BLM identifiers

Categorical context variable features such as `Accelerator Mode` and `Beam Mode` need encoding before being used in numerical models. Even though one can easily assign a numerical value to each categorical value, the ordering would not have any connection to reality, potentially confusing models during fitting. Instead, the encoding of such features was done using one-hot encoding.

In order to distinguish between BLMs, each BLM was assigned a unique number. This was then one-hot encoded to ensure that each BLM gets a unique identifier that contains no information about the location of the BLM or the "order" of the BLMs in the dataset.

2.1.5.3. Adding indicator features for very low or saturated beam loss values

For each BLM, indicator features were added for cases when the BLM observed beam losses below the noise level and cases where the BLM observed beam losses above the saturation level, respectively. These indicator features can help deep learning models assign appropriate importance to beam loss features from such cases.

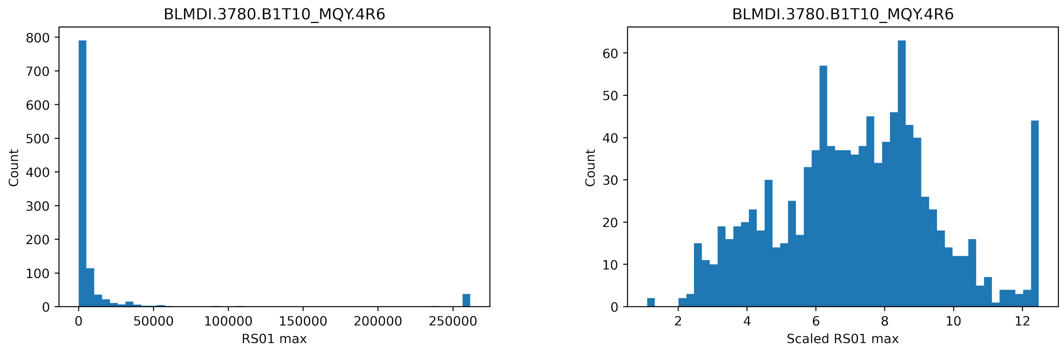
2.1.5.4. Transforming beam loss features

Beam loss data corresponding to `RS01_max` below approximately 100 bits is sensitive to noise, and should not be given much influence when classifying beam dumps. In order to ensure this for the purpose of deep learning, it was chosen to clip all `RS01_max` values below 100 bits, such that they were all set to 100 bits.

The distribution of beam loss features corresponding to the maximum or sum of running sums are known to be skewed toward lower values. From a machine protection perspective, it is important to be able to distinguish between `RS01_max` values of, e.g., 150 bits and 350 bits. The difference between these two values only make up a very small percentage of the possible `RS01_max` range, which goes up to approximately

261 100 bits. In comparison, the difference between, e.g., 260 150 bits and 260 350 bits is unimportant. Therefore, nonlinear feature scaling is desirable for beam loss features corresponding to the maximum or sum of running sums. Such a scaling function should be monotonous, steep in regions where small value differences are important and flat where they are unimportant.

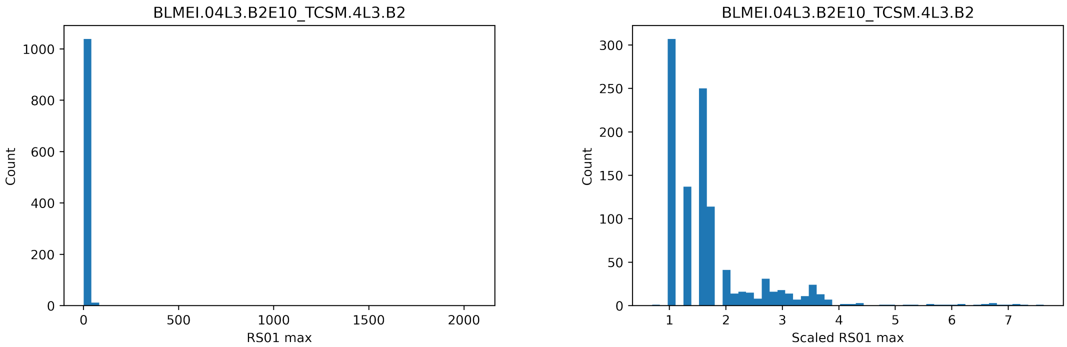
A logarithmic scaling function was chosen for this purpose, since it fulfills these criteria. In addition, it preserves relative differences for all orders of magnitude. Figures 2.24—2.26 show examples of the effect of transforming beam losses (`RS01_max`) using $\ln(x+1)$, where x is the `RS01_max` value. It can be seen that the transformation results in a more even distribution of beam losses over the available range.



(a) Distribution of beam losses before logarithmic scaling.

(b) Distribution of beam losses after logarithmic scaling.

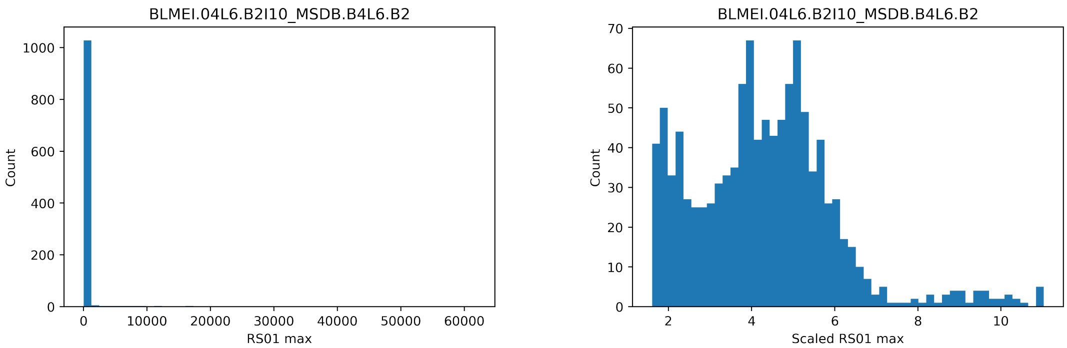
Figure 2.24.: Effect of logarithmic scaling of beam losses (`RS01_max`) for a BLM in beam dump line 1 that frequently observed high beam losses during Run 2 of the LHC. The transformation results in a more even distribution of beam losses over the range. Peaks can be seen at the far right in both panels, corresponding to saturated beam loss values.



(a) Distribution of beam losses before logarithmic scaling.

(b) Distribution of beam losses after logarithmic scaling.

Figure 2.25.: Effect of logarithmic scaling of beam losses (`RS01_max`) for a BLM in octant 3 of the LHC. The distribution of beam losses after transformation is more even than before, although it is still skewed toward lower values. This is because this specific BLM only rarely observed considerable beam losses.



(a) Distribution of beam losses before logarithmic scaling.

(b) Distribution of beam losses after logarithmic scaling.

Figure 2.26.: Effect of logarithmic scaling of beam losses (`RS01_max`) for a BLM in octant 6 of the LHC. The transformation results in a more even distribution of beam losses over the range.

2.1.5.5. Imputing and normalizing features

Some features have a large fraction of missing values, such as the rise time and fall time features of low-beam-loss BLMs. Deep learning methods are typically not able to directly handle missing values. Therefore, for all existing features in the dataset, a new feature was added indicating when a value was missing in the corresponding original feature. Missing values in the dataset were then imputed such that they were replaced by the mean of the corresponding feature taken over the dataset. Exceptions to this are the Boolean features corresponding to asynchronous beam dump tests, 10 Hz beam dumps, and UFOs, which were converted to numerical values and imputed with zeros.

Finally, the dataset was normalized such that each feature is scaled to the range $[0, 1]$. This was done to avoid features with greater ranges being given more influence than features with smaller ranges. The maximum and minimum value of each feature before normalization were saved, to be used for reconstruction of the original feature ranges. When appropriate, the normalization was applied to individual data folds instead of the full dataset.

2.2. Physics-model–driven approach to classifying beam dumps

The aim of the physics-model–driven approach was to create a tool that can classify beam losses in a given beam dump as either "ok" or "not ok". Historical beam loss data from Run 2 of the LHC was used to define BLM-wise thresholds. If, for a given test beam dump, more than a pre-defined number of BLMs record beam losses larger than their set threshold, the beam dump is considered "not OK". The process of defining beam loss thresholds and classifying beam dumps is explained below.

2.2.1. Correlations between beam losses and context variables

It is well-known by LHC machine protection experts that beam losses in certain parts of the LHC are strongly correlated with certain context variables. For example, some BLMs frequently observe beam losses that are correlated with the beam intensity. Using such knowledge, it is possible to predict beam losses observed by certain BLMs, only through the values of a small number of context variables. In addition to beam intensity, other context variables that are known to be correlated with beam losses are abort gap population intensity, instantaneous luminosity at the experiments and, trivially, beam energy. By taking into account such correlations between beam losses and context variables, it is possible to improve the beam loss classification when compared to using only static thresholds.

For correlation analysis presented here, only linear correlations between beam losses and context variables were examined, BLM by BLM. The only BLM-wise beam loss feature used was `RS01_max`. The Pearson correlation coefficient, or r-value, was used to quantify the correlations. Equation 2.1 shows the formula for the Pearson correlation coefficient for a sample of n pairs $\{(x_i, y_i)\}_{i=1}^{n}$ with sample means $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. The r-value is limited to the range $[-1, 1]$, with the interpretation that $r = -1$ is a perfect negative correlation, $r = 1$ is a perfect positive correlation, and $r = 0$ is no correlation at all. For $0 < |r| < 1$ there is some correlation, with larger $|r|$ indicating a stronger correlation.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.1}$$

The Pearson correlation coefficient was chosen as correlation measure because of its simplicity of implementation and interpretation. While the measure is not robust against outliers, it was decided that it would be quicker to manually filter out cases where outliers skewed results considerably, rather than constructing another measure of correlation.

Due to missing values in the dataset, especially among context variables, some correlation calculations had very few data points compared to the theoretical maximum, which is the total number of beam dumps in the dataset. It was decided that a minimum number of data points, e.g. 10, was required to calculate a meaningful correlation coefficient. In addition, only `RS01_max` values above the defined noise level were used.

Correlation coefficients were calculated for every possible pair of `RS01_max` and numerical context variable. The context variables that are specific to either Beam 1 or Beam 2 were only paired with BLMs that monitor beam losses in the corresponding beam pipe.

Any correlation with $r > 0.6$ was examined further. Scatter plots of these relationships were inspected manually. Some relationships were linear in nature, as shown in Figure 2.27, while others, such as the ones shown in Figure 2.28, showed more complicated relationships. The manual inspection helped identify cases where a high correlation coefficient was the result of one or more outliers. In such cases, the outlier values were discarded and the correlations were re-calculated without them.

While context variables corresponding to beam position and collimator positions are continuous and numerical, they are typically clustered around only a few values. Manual inspection revealed that this clustering skews correlation calculations to the extent where it was decided to completely omit the affected context variables from the analysis.

While many BLMs have beam losses that are strongly correlated with beam energy, no such relationships were examined, since this analysis only considers beam dumps at top energy.

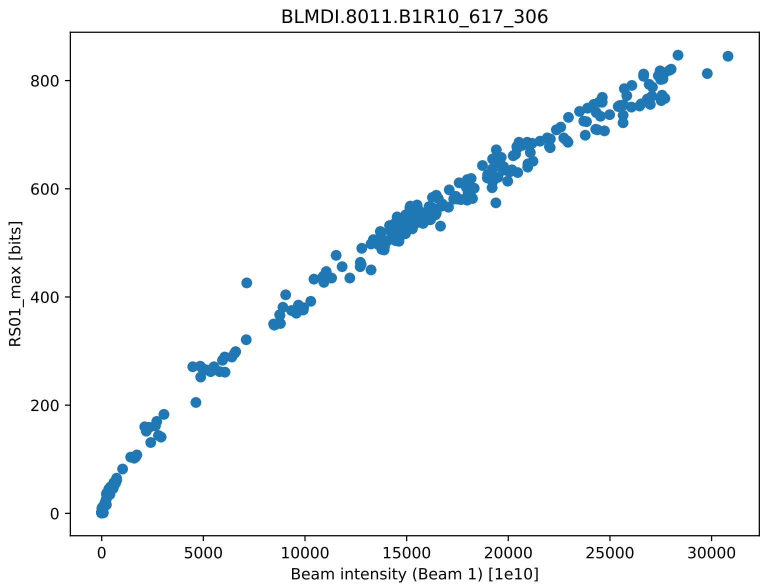
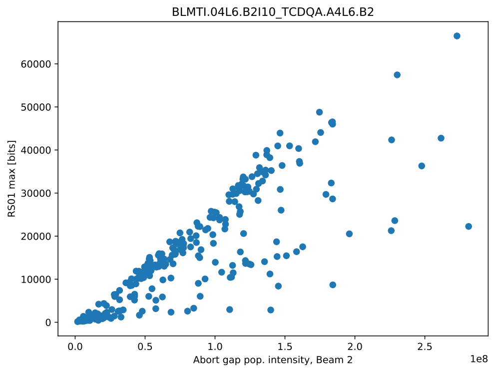
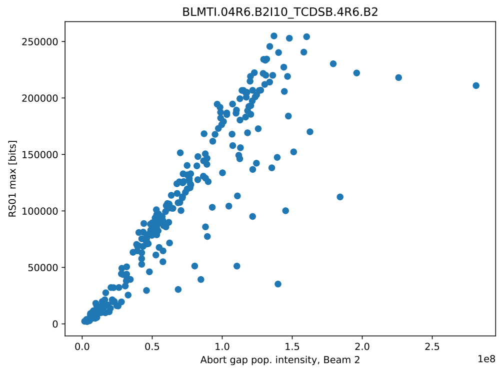


Figure 2.27.: An example of a strong linear relationship ($r = 0.98$) between beam losses and beam intensity for a BLM located in beam dump line 1. The relationship is nonlinear for lower beam intensities. This example shows that simple linear models can be used to explain beam losses from context variables for certain BLMs.



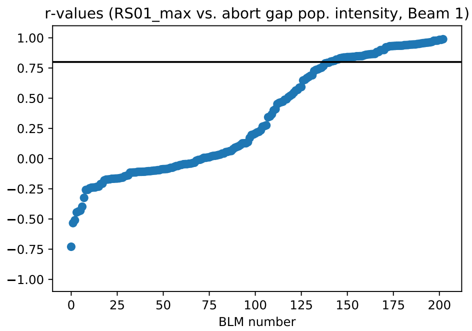
(a) A TCDQ BLM, $r = 0.82$.



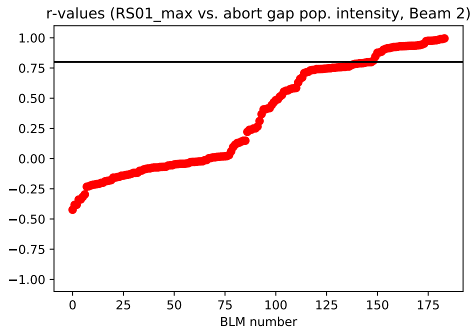
(b) A TCDS BLM, $r = 0.81$.

Figure 2.28.: Examples of BLMs with strong positive correlations ($r > 0.8$) between beam losses and abort gap population intensity. In both examples, there is one main linear trend along with at least one subtle linear trend. The cause of the multiple linear trends is not known. This example shows that for some BLMs, the relationship between beam losses and context variables are better explained by models that are not linear, despite having a high linear correlation factor. Similar patterns were observed in several other BLMs located near the TCDQ or TCDS absorbers, indicating a possible connection to physics phenomena in these regions.

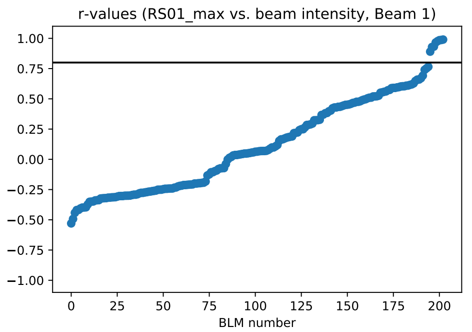
To get an overview of the distribution of r-values for the correlations between a certain context variable and the BLMs, the r-values were plotted in ascending order, as shown in Figure 2.29. This shows that essentially no BLMs have strong negative correlations with abort gap population intensity or beam intensity. About a quarter of ICs have a strong positive correlation ($r > 0.8$) with abort gap population intensity, and about 5 % of ICs have a strong positive correlation ($r > 0.8$) with beam intensity.



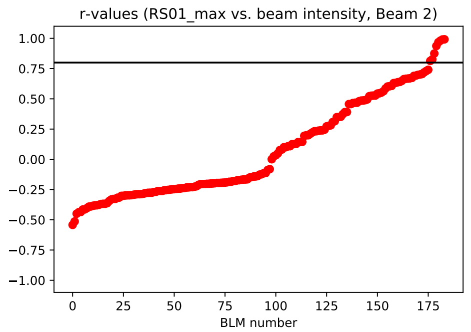
(a) Abort gap population intensity, Beam 1.



(b) Abort gap population intensity, Beam 2.



(c) Beam intensity, Beam 1.



(d) Beam intensity, Beam 2.

Figure 2.29.: Sorted r-values for correlations between beam losses (`RS01_max`) observed IC BLMs and abort gap population intensity (upper panels) and beam intensity (lower panels). This shows that most BLMs only have weak correlations between beam losses and these context variables. The only strong correlations ($|r| > 0.8$) are positive. The correlations were calculated using top energy beam dumps only, with at least 150 data points above the noise level required per BLM. The horizontal lines are at $r = 0.8$.

BLMs intended to monitor beam losses in one beam pipe can sometimes observe beam losses originating from the other beam pipe. This is due to particle showers caused by the impact of high-energy protons traveling between the two beam pipes. Therefore, it is of interest to investigate correlations between beam losses in one beam pipe and context variables corresponding to the other beam pipe. However, there is a strong correlation between the context variables in the two beam pipes. For example, abort gap population intensity and beam intensity are typically about the same in both beams. This complicates the correlation analysis between beam-specific BLMs and beam-specific context variables. A quick analysis showed that no such cross-correlations between the two beam pipes were stronger than the corresponding correlations within each beam pipe. In order to facilitate the remaining analysis and increase the interpretability of the tool, it was decided to not consider correlations between the two beams.

In order to find any clear trends in time that could skew the overall correlation analysis, each context variable was plotted against fill number in Run 2. Such trends in time could be, for example, a beam position context variable slowly drifting throughout Run 2, which would mean that the time component needs to be accounted for in the following analysis. No clear trends were found and, hence, no further actions needed to be taken with respect to the time component.

After omitting context variables corresponding to beam position and collimator positions, only 3 types of context variables remained among those that were strongly correlated ($r \geq 0.9$) with the correlated BLMs. These context variables were the ones corresponding to beam intensity (Beam 1 and Beam 2), abort gap population intensity (Beam 1 and Beam 2), and instantaneous luminosity (at the ALICE, ATLAS and CMS experiments). Before the start of the analysis, these context variables had been expected to be important for explaining beam losses in certain BLMs, due to their clear connection to physical phenomena related to beam losses in the LHC. Because of their obvious relevance, and in order to simplify the analysis, it was decided that only correlations including these 3 types of context variables would be considered in the following.

2.2.2. Defining BLM classes

From the correlation analysis, it is clear that the BLMs can be divided into 3 classes, based on the correlation between their beam losses and certain context variables, according to the following:

- **Correlated BLMs**: Have beam losses that are strongly correlated with one or more context variables (abort gap population intensity, beam intensity, or instantaneous luminosity). They frequently observe beam losses distinctly above the noise level.

- **Low-beam-loss BLMs**: Rarely observe beam losses above the noise level. No meaningful correlations with context variables can be calculated for these BLMs.
- **Remaining BLMs**: Frequently observe beam losses distinctly above the noise level, but are not strongly correlated with any relevant context variables. These are excluded from this analysis.

This grouping of BLMs into classes aligns well with previous knowledge and experience with machine protection in the LHC. The number of BLMs that fall into each class depends on the parameters used, as explained in Section 2.2.4 below. Further, two subclasses for the low-beam-loss BLMs were defined:

- **Always-low-beam-loss BLMs**: Did not record beam losses above the noise level in any of the beam dumps in the dataset.
- **Frequently-low-beam-loss BLMs**: Recorded beam losses above the noise level only in a few beam dumps in the dataset, typically defined as 1 to 10 beam dumps.

2.2.3. Modeling beam losses and defining thresholds from nominal beam dump data

From the correlation analysis and the definition of the BLM classes, it is clear that beam losses should be modeled differently for the different BLM classes. The thresholds are constructed using data from a subset of beam dumps from the dataset, as explained below.

2.2.3.1. Defining training set and test set

Among the context variables, there are a number of variables that give direct information about a given beam dump that can be used to pre-classify it as "nominal" or "non-nominal". These context variables are listed together with their corresponding "nominal" values in Table 2.3, defining a "nominal" beam dump. For the purpose of this analysis, any beam dump that does not fulfill all these criteria is pre-classified as "non-nominal".

Using these definitions of "nominal" and "non-nominal" beam dumps, there are 312 "nominal" and 767 "non-nominal" beam dumps in the high-energy beam dump dataset for the years 2016 to 2018. Based on the pre-classification, it is assumed that the 312 "nominal" beam dumps contain nominal beam loss behavior. Therefore, the "nominal" beam dumps are used as training data to define nominal beam loss behavior in the LHC, by defining thresholds based on this "nominal" data.

Table 2.3.: Values of context variables that define "nominal" beam dumps.

Context variable	Nominal value
`XPOC_check_ok_B1`	1
`XPOC_check_ok_B2`	1
`Event Category`	"PROGRAMMED_DUMP"
`Accelerator Mode`	"PROTON PHYSICS"
`Pm Machine Protection Result`	"OK"
`Overall Result`	"OK"
`Orbit Changes`	"No considerable Orbit Changes"
`Missing BBK B1 effect`	0 or N/A
`Missing BBK B2 effect`	0 or N/A
`10 Hz B1`	`False` or N/A
`10 Hz B2`	`False` or N/A
`Asynchronous dump test`	`False` or N/A

However, not all "non-nominal" beam dumps are believed to contain anomalous LHC behavior, especially beam dumps that fulfill most of the criteria in Table 2.3 apart from having missing data. The idea is to tweak and test the tool by classifying the "non-nominal" beam dumps using thresholds defined from the "nominal" beam dumps. A "non-nominal" beam dump can be classified as "ok" if the beam losses are close enough to the nominal behavior, and classified as "not ok" otherwise. If a "non-nominal" beam dump is classified as "not ok", it could indicate that something was anomalous in that beam dump, and should be investigated further. The details of defining beam loss thresholds and classifying "non-nominal" beam dumps is explained below.

2.2.3.2. Modeling beam losses in correlated BLMs

For the correlated BLMs, beam losses can be modeled using a linear model fitted to data that is known to represent nominal behaviour in the LHC. The constant term is set to zero since the BLM is expected to observe no beam losses when the corresponding context variable (e.g., beam intensity) is zero, with the assumption that the beam losses observed by each correlated BLM only depend on one single context variable. The model can be written as

$$y = cx + \epsilon \tag{2.2}$$

where $y \in \mathbb{R}^+$ is the beam loss expressed as `RS01_max`, $c \in \mathbb{R}$ is a proportionality factor, $x \in \mathbb{R}^+$ is the context variable, and $\epsilon \in \mathbb{R}$ is the error.

For this analysis, it was assumed that the error term ϵ could be modeled as a Gaussian variable. This model suggests the thresholds to be set as a certain deviation from the expected beam loss value, expressed as a number of standard deviations σ, according to

$$y_{\text{lower}} = cx - n_\sigma \sigma \tag{2.3}$$

$$y_{\text{upper}} = cx + n_\sigma \sigma \tag{2.4}$$

where n_σ is the number of standard deviations. If a correlated BLM observes beam losses outside of these thresholds during a beam dump, it could indicate anomalous beam loss behavior in the LHC.

Figures 2.30—2.33 show examples of ordinary least squares model fits for four different correlated BLMs, showing bounds at 2 standard deviations from the mean.

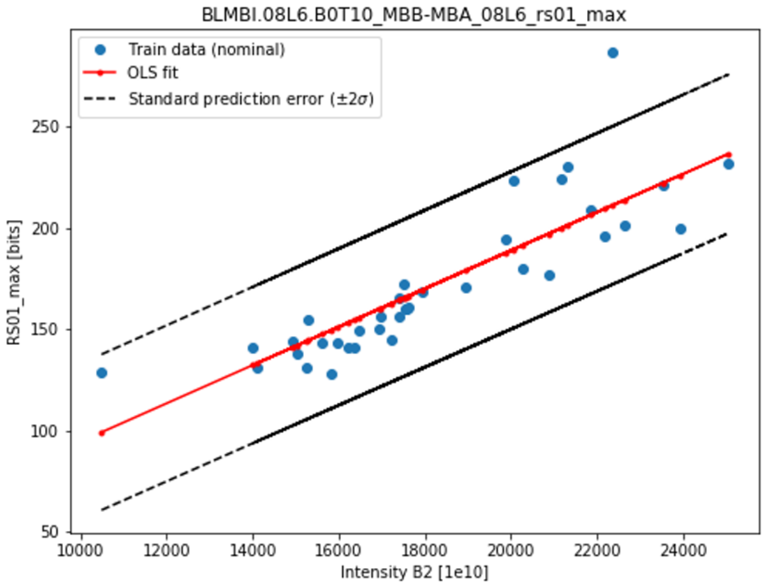


Figure 2.30.: Example of an ordinary least squares model fit for the joint distribution of beam losses (`RS01_max`) measured by one BLM and the beam intensity. This BLM only observed low beam losses in the training data. The trend is linear with moderate spread. It can be seen that this linear model can describe the relationship between the context variable and the beam losses observed by this BLM well. The model can be used to define beam loss thresholds for this BLM, from the uncertainty of the model fit, which can then be used to classify beam dumps.

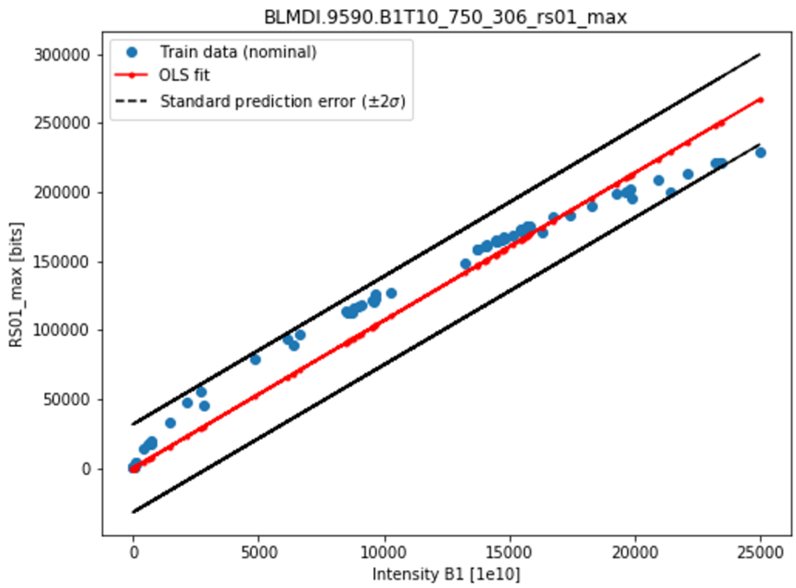


Figure 2.31.: Example of an ordinary least squares model fit for the joint distribution of beam losses (`RS01_max`) measured by a dump line BLM and the beam intensity. This BLM frequently observed high beam losses in the training data. A nonlinear trend with low spread can be seen, typical for beam dump line BLMs. Despite the nonlinear behavior, this linear model can describe the relationship between the context variable and the beam losses observed by this BLM well for this range. The model can be used to define beam loss thresholds for this BLM, from the uncertainty of the model fit, which can then be used to classify beam dumps.

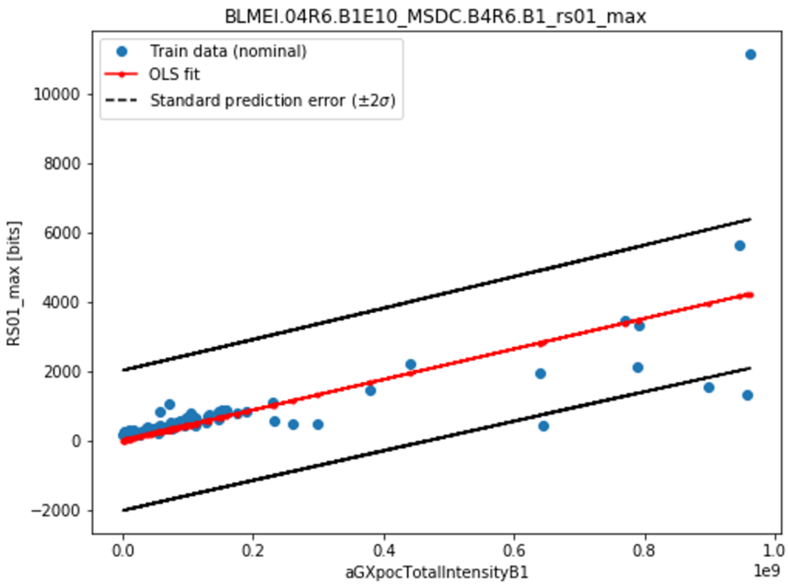


Figure 2.32.: Example of an ordinary least squares model fit for the joint distribution of beam losses (`RS01_max`) measured by one BLM and the abort gap population intensity, showing a clear linear trend for abort gap population intensity below 2×10^8 protons, with a higher spread for larger values. The higher spread in higher values complicates the modeling of beam losses as a function of abort gap population intensity. Nonetheless, some BLMs, including this specific one, observed beam losses that are only strongly correlated with abort gap population intensity, making it the only option for linear modeling. The model can be used to define beam loss thresholds for this BLM, from the uncertainty of the model fit, which can then be used to classify beam dumps.

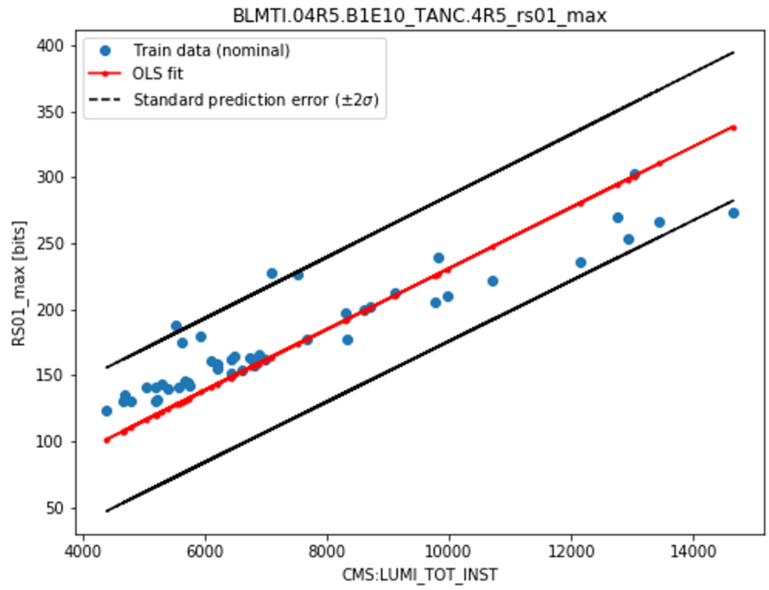


Figure 2.33.: Example of an ordinary least squares model fit for the joint distribution of beam losses (`RS01_max`) measured by one BLM and the instantaneous luminosity at the CMS experiment. The relationship is linear with moderate spread. It can be seen that this linear model can describe the relationship between the context variable and the beam losses observed by this BLM well. The relationship resembles that in Figure 2.30, showing that beam loss models as functions of beam intensity or instantaneous luminosity at the experiments are similar. The model can be used to define beam loss thresholds for this BLM, from the uncertainty of the model fit, which can then be used to classify beam dumps.

Data points where `RS01_max` was either at the saturation level or below the noise level are not included in the calculation of the thresholds.

The correlated BLMs that are strongly correlated with abort gap population intensity represent a special case for the modeling. Beam losses that are lower than expected from the abort gap population intensity should still be considered as "ok" since lower beam losses represent a safer situation. Therefore, the lower threshold is set to 0 for all correlated BLMs that are modeled using abort gap population intensity, according to

$$y_{\text{lower}} = 0. \tag{2.5}$$

Additionally, abort gap population intensity should never exceed $X_{\text{AG}} = 4 \cdot 10^{10}$ protons, as this in itself would be considered anomalous. Thus, the corresponding upper threshold for abort gap population intensity-correlated BLMs is modeled as

$$y_{\text{upper}} = \begin{cases} cx + n_\sigma \sigma & \text{if } x < X_{\text{AG}} \\ cX_{\text{AG}} + n_\sigma \sigma & \text{else.} \end{cases} \tag{2.6}$$

2.2.3.3. Modeling beam losses in low-beam-loss BLMs

The low-beam-loss BLMs are expected to observe low beam losses, typically below the noise level, and are assumed to be independent of any context variables. Therefore, the low-beam-loss BLMs can be modeled using a constant term plus an uncertainty, according to

$$y = k + \epsilon \tag{2.7}$$

where $y \in \mathbb{R}^+$ is the beam loss expressed as `RS01_max`, $k \in \mathbb{R}$ is a constant, and $\epsilon \in \mathbb{R}$ is the error.

A reasonable choice for the constant term k is the mean `RS01_max` for the BLM in question, taken over the training data. As in the case with the correlated BLMs, the uncertainty is assumed to be Gaussian. Hence, a reasonable choice for modeling the uncertainty is to use a certain number of standard deviations σ.

Additionally, to avoid the thresholds being set below the noise level, they are adjusted according to

$$y_{\text{lower}} = 0 \tag{2.8}$$

$$y_{\text{upper}} = \max(\mu + n_\sigma \sigma, y_{\text{noise}}) \tag{2.9}$$

where μ is the mean `RS01_max` taken over the training data for the specific BLM. This modeling applies to both subclasses of the low-beam-loss BLMs.

Initial tests showed that many low-beam-loss BLMs, according to this modeling, got their upper threshold set to just above the noise level (e.g. 120 bits). From a machine protection perspective, an `RS01_max` value of 120 bits is not a cause for concern, even for BLMs that generally observe very low beam losses, and should not be directly considered as an indication of a potential anomaly. Therefore, an additional factor f was introduced, to be multiplied with the upper limit in Equation 2.9, resulting in an upper threshold of

$$y_{\text{upper}} = f \cdot [\max(\mu + n_\sigma \sigma, y_{\text{noise}})] \, . \tag{2.10}$$

2.2.3.4. Beam dump classification

When classifying a "non-nominal" beam dump, the BLM classes are treated separately, including the two subclasses of low-beam-loss BLMs, for a total of 3 BLM classes. If more than a predefined number of BLMs observed beam losses outside of the defined thresholds, for any or all of the BLM classes, the beam dump in question will be marked as "not ok". The number of BLMs (per BLM class) that are allowed to exceed the thresholds per beam dump is an additional parameter of the model, and will be denoted as N_{ex}. For each beam dump classified as "not ok", the names of the BLMs that exceeded the modeled thresholds are saved together with the ratio by which they violated the thresholds.

2.2.4. Parameter tuning

The modeling of beam losses in correlated and low-beam-loss BLMs had introduced several parameters, shown in Table 2.4. They all directly effect the output of the tool, and needed tuning for optimal classification results. The parameter tuning was done using a grid search approach, where the number of beam dumps marked as "not ok" was examined for each combination of parameters, except for the parameter n_{EN} that could be tuned independently of the others. In addition, the output of the classifications were examined manually to ensure that the tool made reasonable classifications. For example, all asynchronous beam dump tests should be classified as "not ok". Similar known cases were used to verify the soundness of the resulting classifications from a given configuration of parameters.

Table 2.4.: Parameters of modeling beam losses and classifying beam dumps.

r_{min}	Minimum r-value required for correlated BLMs	Correlated BLMs
n_{min}	Minimum data points required to calculate a meaningful r-value	Correlated BLMs
y_{noise}	Minimum `RS01_max` value required for data points to be considered	Both BLM classes
n_σ	Number of standard deviations used for modeling the error of BLM class models	Both BLM classes
f	Additional factor used to relax the upper beam loss threshold	Low-beam-loss BLMs
N_{ex}	Number of BLMs allowed to exceed the beam loss thresholds per beam dump	Both BLM classes
n_{EN}	Maximum number of instances with `RS01_max` exceeding y_{noise} allowed for a low-beam-loss BLM	Low-beam-loss BLMs

Before initiating the parameter sweep, suitable ranges for each parameter had been devised. The testing, verification and modeling up to this point had already yielded approximate ranges for several of the parameters where optimal parameter values were likely to be found, as outlined below.

- The low-beam-loss BLMs are defined by two parameters, the noise level y_{noise}

and the maximum number of data points (beam dumps) allowed where the BLM in question observed beam losses above the noise level, n_{EN}. The most strict definition of low-beam-loss BLMs, corresponding to the subclass always low-loss BLMs, would allow no instances with `RS01_max` exceeding the noise level. A more relaxed definition used for the frequently-low-beam-loss BLMs, allowing for up to, e.g., 10 instances with beam losses above the noise level, would include more BLMs while still ensuring that all of them almost exclusively observe very low beam losses. Figure 2.34 shows how the number of low-beam-loss BLMs depends on the parameter n_{EN}. Starting from low n_{EN}, the number of low-beam-loss BLMs increases rapidly until about $n_{\mathrm{EN}} = 10$, indicating that this is a good value for this parameter.

- The correlation analysis had shown that an appropriate range for r_{min} was between approximately 0.6 and 0.9. For simplicity, 0.6 was chosen as the default value. But, in order to make the parameter tuning more complete, several values in range [0.6-0.9] were tested as well in the grid search.
- In order to calculate a meaningful r-value, a minimum number of data points is required. A correlation based on only 2 or 3 data points is clearly not meaningful in most cases, while, e.g., 50 data points give more credibility to the correlation. During the correlation analysis, it was found that reasonable values for this parameters could be, for example, 10, 50 or 100.
- Machine protection experts had recommended to use a noise level y_{noise} somewhere between 20 bits and 120 bits. The inspection of BLM data, together with the correlation analysis, confirmed that this was a suitable range for the noise level.
- The number of standard deviations that beam losses are allowed to deviate from the expected mean (excluding the additional factor for low-beam-loss BLMs), n_{σ}, was initially thought to be in the range of approximately 2 to 4.
- The additional factor for the low-beam-loss BLM thresholds, f, has only a weak connection to LHC knowledge and intuition. From initial test modeling of the low-beam-loss BLMs, it was clear that the factor needed to be greater than 1, and probably set to at least 2. For an upper bound for the parameter, there was no obvious value. However, it was clear that, e.g., 5 would likely be too generous of a factor. Hence, the range was limited to between 2 and 5.
- The number of BLMs allowed, per BLM (sub-)class, to have beam losses outside of the thresholds in a given beam dump without the beam dump being considered as "not ok", N_{ex}, is a parameter that can be used to adapt how strict the beam

loss classification should be. Setting the parameter too low would lead to too many beam dumps being classified as "not ok", while setting it too high would risk missing beam dumps with anomalous beam loss behavior.

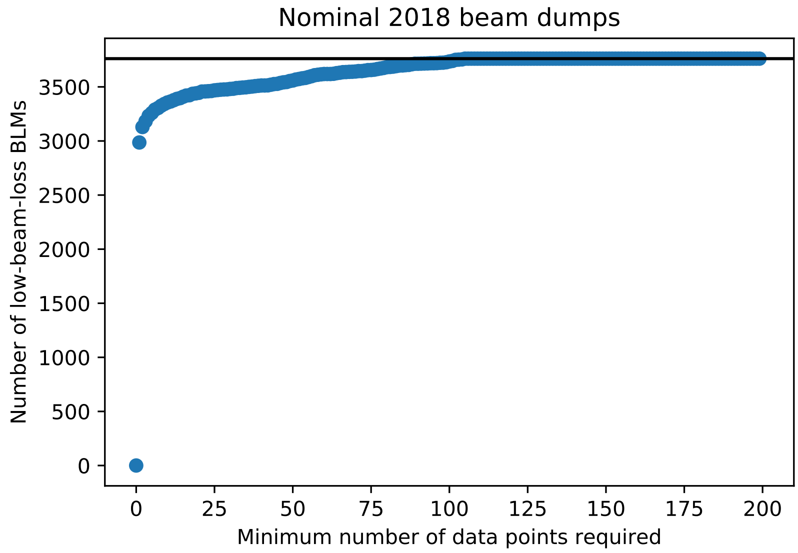


Figure 2.34.: The number of BLMs classified as low-beam-loss BLMs as a function of the parameter n_{EN} (the maximum number of instances with `RS01_max` exceeding the noise level y_{noise} allowed for a low-beam-loss BLM). The number of low-beam-loss BLMs is sensitive to changes for small n_{EN}, especially below 10. The horizontal line marks the maximum number of BLMs, which is equal to the total number of BLMs.

Visualizations of examples of partial parameter sweeps for the parameters f and N_{ex} are shown in Figure 2.35, where the other parameters were fixed. Suitable parameter values are found where the curves start to flatten out and converge. It can be seen that, for these examples, $f = 3$ and $N_{\mathrm{ex}} = 3$ are good choices.

Table 2.5 contains the ranges used for parameter tuning, together with the set of parameters that yielded the best result. Using these values, the BLM count for the different (sub-)classes were as shown in Table 2.6.

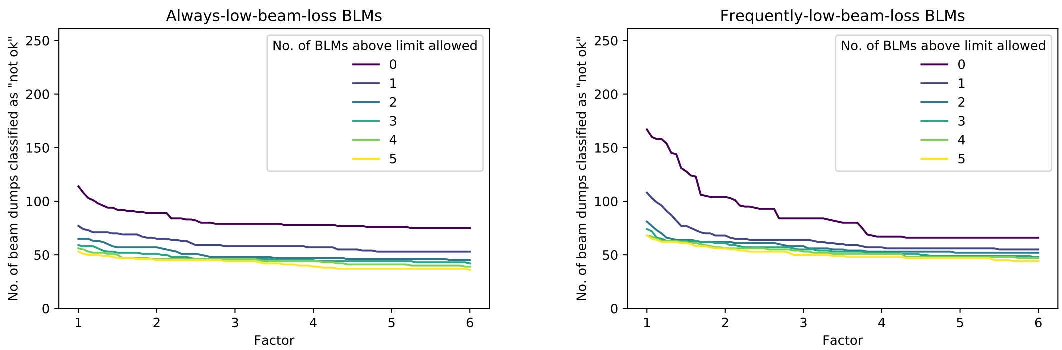


Figure 2.35.: The number of "non-nominal" beam dumps classified as "not ok" as a function of the factor f used in classifications for the BLM subclasses "always-low-beam-loss BLMs" (left) and "frequently-low-beam-loss BLMs" (right), with y_{noise} set to 120 bits. Different curves are shown, corresponding to different values for the parameter N_{ex}. High sensitivity for $f < 3$ and $N_{\text{ex}} < 2$ can be seen, suggesting that these parameters should be chosen above those values to ensure stability in the classification model.

Table 2.5.: Parameter values used for grid search and the final parameter values yielding the best results.

Parameter	Range used for grid search	Final parameter value
r_{min}	[0.6, 0.9]	0.6
n_{min}	[10, 100]	10
y_{noise}	[20, 120]	120
n_{σ}	[2, 4]	4 (correlated BLMs) 5 (low-beam-loss BLMs)
f	[2, 5]	3
N_{ex}	[0, 5]	1 (correlated BLMs) 2 (always-low-beam-loss BLMs) 2 (frequently-low-beam-loss BLMs)
n_{EN}	[0, 10]	0 (always-low-beam-loss BLMs) [1, 10] (frequently-low-beam-loss BLMs)

Table 2.6.: BLM counts for the classes used for beam dump classification.

BLM class	Count
Correlated BLMs	49
Always-low-beam-loss BLMs	3287
Frequently-low-beam-loss BLMs	165

2.3. Deep learning approach to classifying beam dumps

The aim of the deep learning approach was to develop a method that can be used to identify and classify beam loss anomalies in beam dump data. Compared to the physics-model–driven approach, deep learning approaches have the advantage of being able to extract relevant information from the data that would be difficult or impossible for a human to identify. A neural network encoder architecture was designed, implemented and tested in a binary classification setting, using existing knowledge about the 3 well-known types of beam loss anomalies. The sections below explain this process, together with a proposed unsupervised method designed to find novel types of subtle anomalies.

2.3.1. Encoder design

The aim of the encoder is to reduce the dimensionality of the input data while capturing important features. Knowledge about the LHC and the layout of BLMs can, and should, be used to design the encoder such that the network can learn relevant features more easily. An example encoder design is shown in Figure 2.36. The sections below outline the main steps through which LHC and BLM knowledge is used to encode beam dump data.

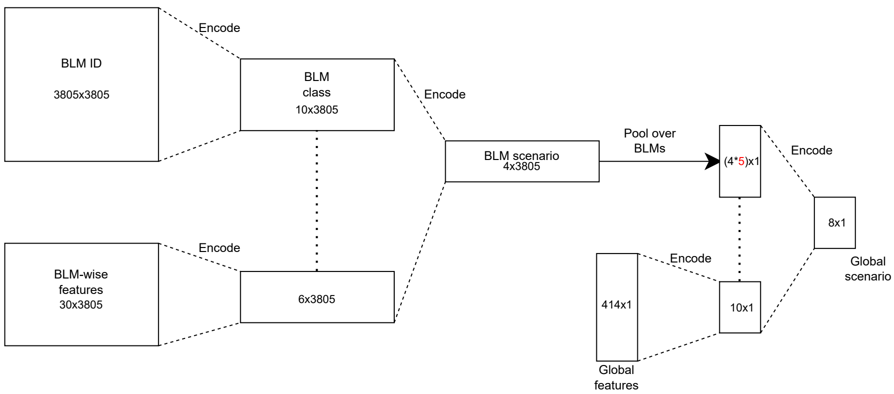


Figure 2.36.: Schematic overview of the encoder architecture. Example array sizes are indicated at each step. The architecture is designed to make use of LHC knowledge to encode features in meaningful and interpretable ways. The output of the encoder can be directly interpreted as classification of a sample (beam dump). The red "5" indicates 5 different pooling types in parallel.

2.3.1.1. Encoding unique BLM identifiers

Since different BLMs have different beam loss characteristics, it is important to be able to distinguish between BLMs. The dataset does not contain information about exact BLM positions, and the BLM names indicate only their approximate positions in the LHC (see Appendix A). Thus, there is no simple way of directly using BLM positional data as unique BLM identifiers.

A simple way to distinguish between BLMs is to assign a one-hot encoded identifier (ID) to each BLM, as explained in Section 2.1.5.2. However, directly distinguishing between every single BLM is not necessary, and could cause issues such as overfitting.

One can divide the BLMs into a relatively small number of BLM classes, where all BLMs in a class have similar beam loss characteristics. For example, all BLMs in the arc sections of the LHC are expected to observe essentially no beam losses and could, thus, form a single BLM class. On the other hand, the 8 straight sections might have unique beam loss characteristics, meaning that BLMs in these sections could be divided into 8 classes, each corresponding to one of the sections. Finally, the BLMs in the beam dump lines are known to observe distinct beam losses compared to the BLMs in the LHC ring, suggesting the BLMs in the beam dump lines to be grouped into a single BLM class. Thus, a reasonable number of BLM classes is 10.

In the first step of encoding BLM data for a given BLM in a given beam dump, the one-hot encoded BLM ID should be encoded into a vector corresponding to BLM class, as shown in Figure 2.36.

Distinguishing between the BLMs through their IDs enables permutation equivariance for the BLMs, i.e., the order of the BLMs does not affect the output. This enables weight sharing in the neural network between BLMs, facilitating the process of learning relevant feature extractors.

2.3.1.2. Encoding BLM class and encoded BLM-wise features

For a given BLM, the concatenation of BLM class and encoded BLM-wise features can be encoded into a *BLM scenario*. This can be thought of as a representation of how "severe" the beam losses were, given that they were observed by a BLM from a specific BLM class. The number of possible BLM scenarios is huge, but should be limited for the sake of efficiently encoding the information. For example, one could choose to define 4 BLM scenarios and think of them as "no beam losses", "beam losses of low severity", "beam losses of moderate severity" and "beam losses of high severity".

2.3.1.3. Pooling over BLMs

In order to achieve permutation invariance for BLMs, global pooling should be applied after having computed the BLM scenarios. The pooling extracts features independently of the order of the BLMs, only keeping information about the BLM scenarios.

In order to extract a sufficient amount of information from the BLM scenarios, different pooling techniques can be used in parallel, with the results concatenated. Examples of pooling techniques that could be relevant are maximum pooling, minimum pooling, average pooling and quantile pooling.

2.3.1.4. Encoding pooled BLM scenarios and encoded global features

Analogous to the concept of BLM scenario, one can define a *global scenario*. This is the encoded concatenation of pooled BLM scenarios and encoded global features. The global scenario contains information that describes the beam dump as a whole. As for the BLM scenarios, the number of possible global scenarios is huge, but should be limited to a small number, taking LHC knowledge into account. For example, if one expect there to be 4 types of anomalies in the data, and 4 types of non-anomalous scenarios, then it is reasonable to say that the number of global scenarios is 8.

2.3.1.5. General encoding considerations

Convolutional layers are used for encoding BLM-wise features and BLM identifiers. The filter size is 1 along the BLM dimension, and maximal along the other dimension. This allows the network to share feature extractors between BLMs. An added benefit is that fewer network parameters are needed.

The activation function used for the intermediary steps of "BLM class", "BLM scenario" and "global scenario" is the softmax function, ensuring that the output of the layer can be interpreted as a probability vector. For all other steps, leaky ReLU is used as activation function.

Batch normalization is used after convolutional layers, before nonlinear activation functions, in order to accelerate and regularize the training procedure. Additionally, dropout is used in the final fully connected layer of the encoder during training.

2.3.2. Decoder design

The proposed decoder design largely mirrors the encoder design. Instead of convolutional layers, the decoder uses transpose convolutional layers. In the autoencoder setting, where the encoder is followed by the decoder, skip-connections are used to feed non-pooled features from the encoder to the decoder, bypassing the pooling layers and the bottleneck. This is to ensure that the autoencoder receives sufficient information for input reconstruction. Without skip-connections, the pooling layers would discard all information about the order of the BLMs.

2.3.3. Experiments

Experiments of two types were performed, intended to gauge the need for sophisticated deep learning methods and to test the proposed encoder architecture in an easy-to-control setting. The neural networks were implemented using PyTorch [Pas+19] and trained on NVIDIA V100S GPUs. The outline of the two types of experiments are outlined below.

2.3.3.1. Initial tests: Training basic autoencoders

The need for more advanced deep learning methods was investigated by training autoencoders with very simple architectures. These autoencoders did not make use of any of the encoding or decoding steps explained above, but instead consisted of a fully-connected architecture with leaky ReLU activations. Hence, they did not feature equivariance or invariance under permutations of BLMs.

The autoencoders were trained on random samples (beam dumps), without any prior pre-classification. The assumption was that the fraction of anomalous training samples would be low enough that the autoencoders would only learn to reconstruct nominal samples well, such that the reconstruction error could serve as an anomaly measure. The loss function used was the reconstruction error measured by the L^2 distance. Training was stopped early when the loss measured on a validation set started to increase. Hyperparameters such as the number of neurons in each layer, the number of layers, etc., were tuned using grid search.

After training, none of the autoencoders were able to distinguish between samples that are known to be anomalous (e.g., asynchronous beam dump tests) and nominal samples. This indicated that more sophisticated approaches were required, for example by including the encoder architecture described above.

2.3.3.2. Training binary classifiers

Supervised experiments were performed in order to test the encoder architecture and to gain intuition for well-performing hyperparameter settings, in preparation for future unsupervised experiments. The 3 well-known anomaly types presented in Section 2.1.1.3 were considered as one anomaly label, where the label 0 denotes "normal sample" and the label 1 denotes "anomalous sample". Binary classifier were trained to distinguish between "anomalous samples" (beam dumps with one of the 3 well-known anomaly types) and "normal samples" (all other beam dumps).

The binary classifiers used the encoder design explained above, with the addition of a final sigmoid layer transforming the multidimensional global scenario into a one-dimensional label. Binary cross-entropy was used to quantify the neural network loss.

The dataset was randomly split into 5 folds. 5 different classifiers were trained, each using 4 of the folds as training data and the 1 remaining fold as validation data. For each classifier, the training set was normalized, and the resulting scaling factors were applied to the validation set.

For each classifier, Kaiming initialization [He+15] was used for the network's weights, which where then updated using the backpropagation algorithm. Early stopping was employed by measuring the network's performance on the validation set. Hyperparameter tuning was performed in order to achieve the lowest loss possible on the validation set. A grid search was used to find optimal values for hyperparameters such as the number of neurons in each layer, optimizer, learning rate, dropout rate, and pooling functions. The Adam optimizer was used with standard momentum rates ($\beta_1 = 0.9$, $\beta_2 = 0.999$).

The classifiers were evaluated by studying their final classification results on the validation sets.

2.3.4. Anomaly detection using variational autoencoders

For an unsupervised anomaly detection method, a variational autoencoder (VAE) can be used. An architecture using the encoder and decoder designs described above can be used to achieve appropriate properties, such as invariance and equivarance under permutations of BLMs. With this architecture, skip-connections should be used to feed non-pooled features from the encoder to the decoder, bypassing the bottleneck. Overall, the architecture resembles the architecture of U-Net [RFB15].

It is important that proper regularization is performed, such that the decoder gets just enough information to be able to reconstruct the input well. In addition to the regularization techniques already described for the encoder, regularization should be applied to the non-pooled features fed via skip-connections to the decoder, as these features likely contain a lot more information than what passes through the bottleneck of the VAE.

A fundamental difference compared to regular autoencoders is the bottleneck of the variational autoencoder. In a regular autoencoder, an input sample $\mathbf{x}$ is encoded into a latent representation $\mathbf{z}$, which is then decoded to a reconstruction $\tilde{\mathbf{x}}$. In a variational autoencoder, an input sample $\mathbf{x}$ is encoded into distribution parameters, commonly for a Gaussian distribution: $\boldsymbol{\mu}_{\mathbf{x}}$ and $\boldsymbol{\sigma}_{\mathbf{x}}$. A sample $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\sigma}_{\mathbf{x}})$ is generated from the distribution, which is finally decoded into a reconstruction $\tilde{\mathbf{x}}$.

The latent space of a VAE consists of distributions, as opposed to points in regular autoencoders. The distributions enable continuity in latent space, meaning that proximity in latent space implies proximity in original feature space. Latent space can be regularized through the objective function.

It is possible that there are unknown anomaly types that are more subtle than the 3 well-known anomaly types described in Section 2.1.1.3. In order to efficiently detect new and possible more subtle anomaly types, it is therefore advisable to leave out samples containing the well-known anomaly type, since including them could hinder the algorithm from learning more subtle types.

The main idea of this approach is to train the VAE only on samples that are known to be normal (i.e., do not contain any issues). Normality can in this case be defined by four context variables corresponding to automatic system checks in the LHC, shown in 2.7 along with corresponding statistics in the dataset. A sample for which all checks are "ok", is highly unlikely to contain any considerable beam loss anomalies. By only training the VAE on such samples, it learns to reconstruct normal samples well. When tested on a non-normal sample, the reconstruction is likely to be considerably worse. Hence, the VAE reconstruction error can directly be used as a novelty measure, and can be interpreted as a measure of "severity" of the anomaly, or as a probability that the sample is an anomaly.

In the dataset, 72.4 % of samples have all system checks "ok", 26.3 % have one or more checks "not ok", and 1.3 % have one or more checks missing. Hence, the normal samples constitute 72.4 % of all samples, while the non-normal samples constitute 27.6 % of all samples. It is believed that beam loss anomalies exist among the non-normal samples, albeit rare. Finding such subtle anomalies and understanding them is if high interest.

In addition to the VAE reconstruction errors, several different novelty measures can be

Table 2.7.: Descriptions and statistics for context variables corresponding to automatic system checks in the LHC.

`XPOC_check_ok_B1`	Checks beam instrumentation measurements and ensures correct functioning of electromagnets for Beam 1	89.3 % ok	9.4 % not ok	1.3 % missing
`XPOC_check_ok_B2`	Checks beam instrumentation measurements and ensures correct functioning of electromagnets for Beam 2	87.7 % ok	11.1 % not ok	1.1 % missing
`PM Machine Protection Result`	Checks whether any BLMs recorded beam losses significantly higher than the predefined BLM-wise thresholds	94.7 % ok	5.3 % not ok	0 % missing
`Overall Result`	A subjective check by the beam operators, taking into account several system	84.0 % ok	16.0 % not ok	1.1 % missing

defined in latent space and original feature space, as in [Vas+18]. Additionally, clusters in latent space can be interpreted as different global scenarios, as described above. Hence, the variational autoencoder approach is capable of both novelty detection and clustering.

3. Results and discussion

3.1. Physics-model–driven approach to classifying beam dumps

Using the final values of models parameters as shown in Table 2.5, the 767 "non-nominal" beam dumps were classified as either "ok" or "not ok". Out of these beam dumps, 223 were classified as "ok" and 544 were classified as "not ok". Table 3.1 shows the overall classification result statistics together with individual result statistics for each BLM (sub-)class.

Table 3.1.: Classification results by BLM (sub-)class.

BLM (sub-)class	Number of beam dumps classified as "ok"	Number of beam dumps classified as "not ok"
Always-low-beam-loss BLMs	600 (78.2 %)	167 (21.8 %)
Frequently-low-beam-loss BLMs	588 (76.7 %)	179 (23.3 %)
Correlated BLMs	599 (78.1 %)	168 (21.9 %)
All	638 (83.2 %)	129 (16.8 %)
Any	**544 (70.9 %)**	**223 (29.1 %)**

A total of 378 different BLMs contributed to one or more beam dumps being classified as "not ok". Out of these, 244 were always-low-beam-loss BLMs, 102 were frequently-low-beam-loss BLMs, and 32 were correlated BLMs. Tables 3.2—3.4 show the BLMs from each (sub-)class that contributed to the most beam dumps being classified as "not ok". It can be seen that BLMs in octant 7 of the LHC are overrepresented, indicating that they observe beam losses outside of the modeled thresholds more frequently than BLMs in other regions. This can be explained by the collimation system in IR7, where beam losses are expected from collimation of the beams.

All "non-nominal" beam dumps that were either asynchronous beam dump tests or

Table 3.2.: The top 5 always-low-beam-loss BLMs that contributed to the most "not ok" classifications.

Always-low-beam-loss BLM	Number of "not ok" classifications contributed to
BLMQI.04L7.B2I10_MQWA.E4L7	105
BLMQI.05L7.B2I10_MQWA.E5L7	104
BLMQI.04L7.B1E10_MQWA.D4L7	103
BLMQI.05L7.B1E10_MQWB.5L7	102
BLMQI.05L7.B1E10_MQWA.A5L7	102

Table 3.3.: The top 5 frequently-low-beam-loss BLMs that contributed to the most "not ok" classifications.

Frequently-low-beam-loss BLM	Number of "not ok" classifications contributed to
BLMAI.06L7.B1E10_MBW.A6L7	108
BLMQI.05L7.B1E10_MQWA.D5L7	101
BLMQI.05L7.B2I10_MQWA.C5L7	97
BLMQI.05L7.B2I10_MQWA.B5L7	96
BLMQI.04R7.B2I30_MQWA.A4R7	87

Table 3.4.: The top 5 correlated BLMs that contributed to the most "not ok" classifications.

Correlated BLM	Number of "not ok" classifications contributed to
BLMTI.04L7.B2I10_TCSG.A4L7.B2	131
BLMTI.04R7.B2I10_TCSG.D4R7.B2	101
BLMTI.06L7.B1E10_TCP.D6L7.B1	87
BLMEI.06R7.B2I10_TCP.A6R7.B2	65
BLMTI.04L6.B2I10_TCDQA.B4L6.B2	64

10 Hz beam dumps were classified as "not ok". Out of the 48 "non-nominal" beam dumps that had known UFO events, 41 were marked as "not ok".

Manual validation of the classification results were performed by comparing context variables such as `Mps Expert Comment`, `Operator Comment` and `Pm Analysis Result Description` to the classification results. It was confirmed that nearly all beam dumps with any clear mention of anomalous events or anomalous beam loss behavior, that should be considered "not ok" from a machine protection perspective, had been classified as "not ok". Additionally, it was validated that very few beam dumps without any sign of anomalies had been classified as "not ok". The classification results were further validated by machine protection experts, who confirmed that the results were good and mostly agreed with their own judgment.

The 7 UFO beam dumps that had been classified as "ok" were examined further. The context variables `Mps Machine Protection Result`, `Overall Result` and `XPOC_check_ok_B2` all had the values "OK" or "1" for all 7 of these beam dumps. However, `XPOC_check_ok_B1` was "0" for one of them. `Mps Expert Comment` made it clear that all 7 of them had been clean beam dumps. Thus, none of these beam dumps seemed to have had any serious issues, and it was concluded that the classification of these 7 beam dumps was acceptable. If desired, further tweaking of model parameters could likely lead to all UFO beam dumps being classified as "not ok".

3.2. Deep learning approach to classifying beam dumps

As the experiments with binary classifiers were intended as tests in preparation for unsupervised experiments using the VAE method described in Section 2.3.4, no extensive hyperparameter optimization was performed. Instead, a grid search using only a few different values for each key hyperparameter was run.

The output of the classifier is a number in the range $[0, 1]$, where the extreme ends correspond to the two possible labels (0: "normal sample", 1: "anomalous sample"). Hence, the output can be seen as a probability that a given test sample contains an anomaly. By considering output values above 0.5 as predicted anomalies, and output values below 0.5 as predicted normal samples, the results from one of the experiments are presented below. These results are representative of several tests, using sets of slightly different hyperparameters that performed approximately equally well.

The classifier correctly classified 52 out of 63 anomalous samples, and correctly classified 1010 out of 1019 normal samples. The misclassified samples were investigated by reading the operator and machine protection expert comments for those beam dumps.

All of the 11 anomalous samples that had been misclassified as normal by the classifier had UFOs in them, according to the comments. There were no mentions of severe beam losses. Even though UFOs typically cause characteristic beam loss patterns, it is possible that the beam losses in these 11 instances were low enough that it could be considered correct that they were classified as normal.

All of the 9 normal samples that had been misclassified as anomalous by the classifier had comments about considerable beam losses. 4 of them had mentions about beam instability or beam orbit changes, and 1 had a mention about a magnet failure. It is possible that the classifier was able to identify these mentioned issues in the beam dump data, and therefore classified them as anomalous.

Even though no extensive hyperparameter optimization was performed, the results show that this binary classification method works well for detecting the known anomaly types. It is likely that results could be improved considerably with proper hyperparameter tuning, potentially leading to correct classification of all samples.

4. Conclusions and outlook

4.1. Physics-model–driven approach to classifying beam dumps

Using the best-performing set of model parameters found, the classification tool yielded good results that agreed with the opinions of machine protection experts. The results proved the effectiveness and potential of this method for beam loss classification. This work laid the foundation for future development of this classification tool.

4.1.1. Future work

A number of possible improvements and extensions of the physics-model–driven classification tool have been identified, and are outlined below. They were all considered at different stages of this project, but had to be left out due to the complexity and time frame of this master thesis. Future extensions of this tool could build on one or more of the following ideas.

4.1.1.1. Fine-tune model parameters

While a set of model parameters that yielded good classification results was found, it is possible that there are other combinations of model parameter values that would perform even better. The grid search approach used a limited number of values for each parameter, due to the tedious process of validating classification results. Hence, a grid search using a finer step size for each parameter, possibly with extended ranges, could yield improved classification performance. A more in-depth validation process of results could also benefit the performance of the tool.

4.1.1.2. Treat beam losses before and after the moment of beam dump separately

The current classification tool considers only the maximum value of the full RS01 time series. It is known that beam loss characteristics can be considerably different when comparing the moments before the beam dump to the moments after the beam dump. Therefore, a more intricate analysis could be achieved by considering these two time intervals separately. For instance, beam losses could be modeled using two different models for each BLM; one for beam losses before the beam dump, and one for beam losses after the beam dump.

4.1.1.3. Improve definition of moment of beam dump

The exact moment of the beam dump in a given RS01 time series differs from beam dump to beam dump. An artificially constructed moment of beam dump is therefore defined, and is used for all BLMs in all beam dumps. The current definition is based on statistics from historical data, erring on the side of caution such that the defined moment of beam dump often is slightly before the real moment of the beam dump. It is possible to improve the definition, such that the moment of beam dump is defined individually for each beam dump. Such a definition would lead to more accurate beam loss modeling on subintervals of the RS01 time series. For instance, an improved definition could be achieved by considering rising beam losses in the BLMs located near the TCDQ or TCDS absorbers. These BLMs are typically among the first to observe rising beam losses in connection to a beam dump, and could be used as a timing reference for all other BLMs.

4.1.1.4. Consider more beam loss features

The current beam loss models only consider the feature `RS01_max`. While it is believed that this is the most relevant beam loss feature for classifying beam dumps, the inclusion of more BLM-wise features in the analysis would enable more sophisticated models, potentially improving classification performance. Additional BLM-wise feature could also be defined. Alternatively, automated feature extraction could be employed on the full RS01 time series.

4.1.1.5. Consider the left-out BLMs

The 304 BLMs that do not fall into any of the currently defined BLM classes were not considered in this analysis. They are characterized by having beam losses above the noise level more frequently than low-beam-loss BLMs, while not being strongly linearly correlated with any of the context variables. It is still possible that these remaining BLMs could be used for beam dump classification, perhaps by using a different kind of beam loss model, or by considering different BLM-wise features.

4.1.1.6. Use a different measure of correlation

The Pearson correlation coefficient used in this analysis is simple to interpret, but is sensitive to outliers. Early in the analysis of BLM data, it became evident that outliers influenced the correlation analysis for many BLMs, sometimes to the point where the value of the correlation coefficient was completely misleading when compared to a visual inspection of variable relationships. A correlation measure that is more robust against outliers could perhaps be used to qualitatively improve the quantification of linear correlations. This could, in turn, lead to better classification performance, as well as an improved understanding of the relationship between beam losses and context variables in certain BLMs.

4.1.1.7. Use nonlinear beam loss models

The beam losses of correlated BLMs were modeled using a linear model of some context variable. This approach proved to work well for many BLMs. However, it was clear that some of the observed relationships between beam losses and context variables could be more accurately explained by nonlinear models. For example, BLMs located in the beam dump line had a nonlinear relationship between beam losses and beam intensity. Using nonlinear models would result in better model fits, potentially with improved classification performance.

4.1.1.8. Use multivariate beam loss models

The beam losses of correlated BLMs were modeled as a function of a single context variable, more specifically, the context variable with the highest correlation coefficient with the beam losses. It is possible that models could be improved by considering two or more context variables for each BLM, leading to multivariate models. Tendencies of multivariate behavior were observed in some relationships between beam losses and context variables. For instance, BLMs that are strongly correlated with abort gap population intensity often had several linear trends visible in the data. The different trends could potentially be explained by a second context variable.

4.1.1.9. Consider the proximity of different BLMs

The current classification tool does not take into account the location or proximity of different BLMs. However, if two or more BLMs in close proximity to each other observe beam losses outside of the thresholds, it is almost certain that these readings correspond to a considerable beam loss event. This idea can be used to improve the classification of beam dumps.

4.2. Deep learning approach to classifying beam dumps

Even without extensive hyperparameter optimization, the binary classifiers achieved good results for identifying known types of beam loss anomalies, being able to correctly identify most of the anomalies. In addition, the incorrect classifications suggest that the classifiers learn the concept of considerable beam losses (i.e., beam losses that warrant further investigation after a beam dump). The promising results indicate that the encoder design has good potential for deep learning experiments on the dataset. The tests with binary classifiers prepare for future experiments using the proposed unsupervised anomaly detection method.

4.2.1. Future work

4.2.1.1. Randomly drop features to combat issues from missing data

In this work, missing values in the dataset have been addressed using imputation along with indicator features. In order to make the deep models more robust against missing values in future beam dump data, one can implement random dropping of features during training. Unlike the regular dropout technique commonly used in deep learning, features would be dropped with a probability that is representative to the rate of missing data for that specific feature. This would force the network to learn to handle missing data, and would likely increase performance on future data. In addition, this would serve as a regularization technique during training, preventing overfitting and increasing generalization.

4.2.1.2. Anomaly detection using variational autoencoders

With proper hyperparameter tuning, the method described in Section 2.3.4 is likely able to identify novel anomaly types. In addition to hyperparameter tuning, some changes to the neural network architecture could be appropriate. For example, the effect of using combinations of different global pooling techniques in parallel (used to pool over BLMs) should be studied.

The proposed VAE approach allows for many different novelty measures defined on original feature space and latent space. Several clustering methods can be defined in latent space, potentially leading to discoveries of currently unknown anomaly types. Much experimentation is likely required to find optimal ways to discover anomalies.

**

A. Explanation of the BLM naming convention

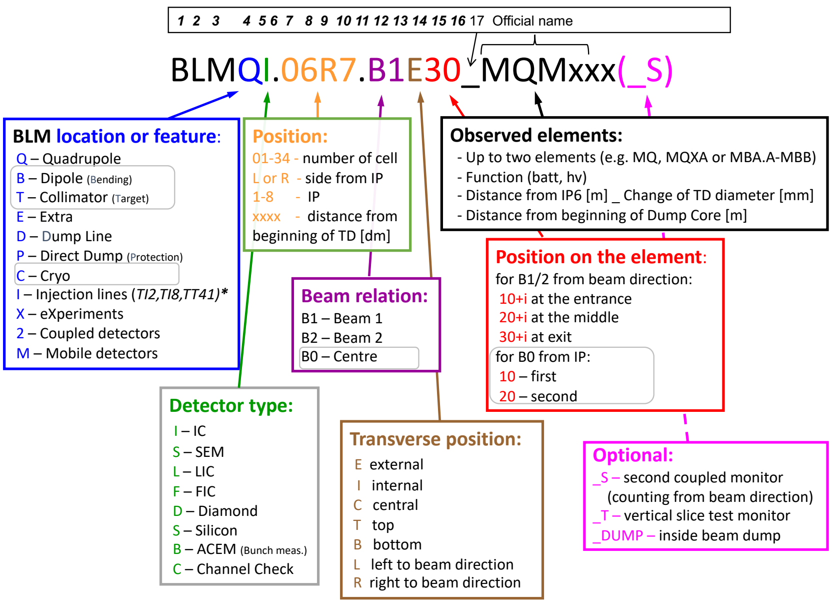


Figure A.1.: Explanation of the BLM naming convention [ZG14].

B. Lists of LHC context variables used in the analysis

Table B.1.: LHC beam position context variables used in the analysis.

Context variable name	Description
LHC.BPMSI.B4R6.B1:ORBIT_H	Average horizontal orbit position
LHC.BPMSI.B4R6.B1:ORBIT_V	Average vertical orbit position
LHC.BPTDH.A4R6.B1:CALIBCORRECTEDPOS	Calibrated corrected position
LHC.BPMSI.B4L6.B2:ORBIT_H	Average horizontal orbit position
LHC.BPMSI.B4L6.B2:ORBIT_V	Average vertical orbit position
LHC.BPTDH.A4L6.B2:CALIBCORRECTEDPOS	Calibrated corrected position
LHC.BPTUH.A4L1.B1:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUH.A4L2.B1:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUH.A4L5.B1:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUH.A4L5.B2:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUH.A4L6.B2:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUH.A4L8.B1:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUH.A4R1.B2:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUH.A4R2.B2:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUH.A4R5.B2:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUH.A4R6.B1:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUH.A4R8.B2:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUH.C6L7.B1:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUH.D4R7.B2:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUV.A4L1.B1:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUV.A4L2.B1:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUV.A4L5.B1:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUV.A4L8.B1:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUV.A4R1.B2:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUV.A4R2.B2:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUV.A4R5.B2:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUV.A4R8.B2:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUV.A5L1.B2:CALIBABSOLUTEPOS	Calibrated absolute beam position
LHC.BPTUV.D4R7.B2:CALIBABSOLUTEPOS	Calibrated absolute beam position

Table B.2.: Miscellaneous LHC context variables used in the analysis.

Context variable name	Description
Event Timestamp	Beam dump timestamp
Event Category	Beam dump type
BEAM_MODE	Beam mode
OVERALL_ENERGY	Beam energy
OVERALL_INTENSITY_1	Beam intensity, Beam 1
OVERALL_INTENSITY_2	Beam intensity, Beam 2
Fill Number	LHC fill number
Mps Expert Comment	Machine protection expert comment
Injection Scheme	Injection scheme
Beam Loss Type	Beam loss type
Beam Losses	Specification of detected beam losses
Bstar1	β^* in IR1
Bstar2	β^* in IR2
Bstar5	β^* in IR5
Bstar8	β^* in IR8
Dump Classification	Beam dump classification
Event Classification	Event classification
Mps Dump Cause	Beam dump cause
Mps First Detection	First detection by machine protection system
Operator Comment	Operator comment
Orbit Changes	Orbit changes
Overall Result	Overall result of beam dump
Pm Analysis Result Description	Post Mortem analysis result description
Pm Machine Protection Result	Post Mortem machine protection result
First Bic Input	First beam interlock system input
timestamp_blm	Beam dump timestamp recorded by BLM system
aGXpocTotalIntensityB1	Total abort gap population intensity Beam 1
aGXpocTotalMaxIntensityB1	Maximum abort gap population intensity Beam 1
timestamp_abort_gap_b1	Timestamp of abort gap of Beam 1
aGXpocTotalIntensityB2	Total abort gap population intensity Beam 2
aGXpocTotalMaxIntensityB2	Maximum abort gap population intensity Beam 2
timestamp_abort_gap_b2	Timestamp of abort gap of Beam 2
UFO	Unidentified falling object
ULO	Unidentified lying object
16L2	UFO in section 16L2
ALICE:LUMI_TOT_INST	Total instantaneous luminosity at ALICE
ATLAS:LUMI_TOT_INST	Total instantaneous luminosity at ATLAS
CMS:LUMI_TOT_INST	Total instantaneous luminosity at CMS
LHCB:LUMI_TOT_INST	Total instantaneous luminosity at LHCb
HX:OPTID	LHC optics identifier
XPOC.B1:ST_CHECK	XPOC system check
BLM.XPOC.B1:ST_CHECK	XPOC BLM system check
LHC.RUNCONFIG:IP1-XING-V-MURAD	Crossing angle in IR1
LHC.RUNCONFIG:IP5-XING-H-MURAD	Crossing angle in IR5
XPOC.B2:ST_CHECK	XPOC system check
BLM.XPOC.B2:ST_CHECK	XPOC BLM system check
Asynchronous dump	Asynchronous dump test
10 Hz B1	10 Hz phenomenon beam dump
10 Hz B2	10 Hz phenomenon beam dump
Missing BBK B1 effect	Missing beam-beam kick effect
Missing BBK B2 effect	Missing beam-beam kick effect

Table B.3.: LHC collimator position context variables used in the analysis.

Context variable name	Description
XRPH.B6L5.B2:MEAS_LVDT_LU	Collimator position
TDI.4L2:MEAS_LVDT_GU	Collimator position
TDI.4R8:MEAS_LVDT_GU	Collimator position
TCDQA.A4R6.B1:MEAS_LVDT_LD	Collimator position
TCTPH.4L1.B1:MEAS_LVDT_GU	Collimator position
TCTPH.4L2.B1:MEAS_LVDT_GU	Collimator position
TCTPH.4L5.B1:MEAS_LVDT_GU	Collimator position
TCTPH.4L8.B1:MEAS_LVDT_GU	Collimator position
TCTPV.4L1.B1:MEAS_LVDT_GU	Collimator position
TCTPV.4L2.B1:MEAS_LVDT_GU	Collimator position
TCTPV.4L5.B1:MEAS_LVDT_GU	Collimator position
TCTPV.4L8.B1:MEAS_LVDT_GU	Collimator position
XRPH.B6R5.B1:MEAS_LVDT_LU	Collimator position
TCDQA.A4L6.B2:MEAS_LVDT_LD	Collimator position
TCTPH.4R1.B2:MEAS_LVDT_GU	Collimator position
TCTPH.4R2.B2:MEAS_LVDT_GU	Collimator position
TCTPH.4R5.B2:MEAS_LVDT_GU	Collimator position
TCTPH.4R8.B2:MEAS_LVDT_GU	Collimator position
TCTPV.4R1.B2:MEAS_LVDT_GU	Collimator position
TCTPV.4R2.B2:MEAS_LVDT_GU	Collimator position
TCTPV.4R5.B2:MEAS_LVDT_GU	Collimator position
TCTPV.4R8.B2:MEAS_LVDT_GU	Collimator position
TCP.B6L7.B1:MEAS_LVDT_GU	Collimator position
TCP.C6L7.B1:MEAS_LVDT_GU	Collimator position
TCP.D6L7.B1:MEAS_LVDT_GU	Collimator position
TCP.B6R7.B2:MEAS_LVDT_GU	Collimator position
TCP.C6R7.B2:MEAS_LVDT_GU	Collimator position
TCP.D6R7.B2:MEAS_LVDT_GU	Collimator position
TCP.6L3.B1:MEAS_LVDT_GU	Collimator position
TCP.6R3.B2:MEAS_LVDT_GU	Collimator position
TCSG.6L7.B2:MEAS_LVDT_GU	Collimator position
TCSG.A4L7.B1:MEAS_LVDT_GU	Collimator position
TCSG.A4L7.B2:MEAS_LVDT_GU	Collimator position
TCSG.A5L7.B1:MEAS_LVDT_GU	Collimator position
TCSG.A6L7.B1:MEAS_LVDT_GU	Collimator position
TCSG.B4L7.B1:MEAS_LVDT_GU	Collimator position
TCSG.B5L7.B1:MEAS_LVDT_GU	Collimator position
TCSG.B5L7.B2:MEAS_LVDT_GU	Collimator position
TCSG.D4L7.B1:MEAS_LVDT_GU	Collimator position
TCSG.D5L7.B2:MEAS_LVDT_GU	Collimator position
TCSG.E5L7.B2:MEAS_LVDT_GU	Collimator position
TCSG.6R7.B1:MEAS_LVDT_GU	Collimator position
TCSG.A4R7.B1:MEAS_LVDT_GU	Collimator position
TCSG.A4R7.B2:MEAS_LVDT_GU	Collimator position
TCSG.A5R7.B2:MEAS_LVDT_GU	Collimator position
TCSG.A6R7.B2:MEAS_LVDT_GU	Collimator position
TCSG.B4R7.B2:MEAS_LVDT_GU	Collimator position
TCSG.B5R7.B1:MEAS_LVDT_GU	Collimator position
TCSG.B5R7.B2:MEAS_LVDT_GU	Collimator position
TCSG.D4R7.B2:MEAS_LVDT_GU	Collimator position
TCSG.D5R7.B1:MEAS_LVDT_GU	Collimator position
TCSG.E5R7.B1:MEAS_LVDT_GU	Collimator position
TCSPM.D4R7.B2:MEAS_LVDT_GU	Collimator position
TCSG.4L3.B2:MEAS_LVDT_GU	Collimator position
TCSG.5L3.B1:MEAS_LVDT_GU	Collimator position
TCSG.A5L3.B2:MEAS_LVDT_GU	Collimator position
TCSG.B5L3.B2:MEAS_LVDT_GU	Collimator position
TCSG.4R3.B1:MEAS_LVDT_GU	Collimator position
TCSG.5R3.B2:MEAS_LVDT_GU	Collimator position
TCSG.A5R3.B1:MEAS_LVDT_GU	Collimator position
TCSG.B5R3.B1:MEAS_LVDT_GU	Collimator position

List of Figures

List of Tables

Bibliography

[And+] K. Andersen, Z. Charifoulline, P. Hagen, M. Maciejewski, C. Obermair, A. Verweij, and S. Sagmo. *LHC Signal Monitoring Project: Development of an Embedded Domain Specific Language for Signal Query and Analysis*.

[And+10] O. O. Andreassen, V. Baggiolini, A. Castaneda, R. Gorbonosov, D. Khasbulatov, H. Reymond, A. Rijllart, I. Romera Ramirez, N. Trofimov, and M. Zerlauth. *The LHC Post Mortem Analysis Framework*. Tech. rep. Geneva: CERN, Jan. 2010.

[ASV19] G. Azzopardi, B. Salvachua, and G. Valentino. "Data-driven cross-talk modeling of beam losses in LHC collimators." In: *Phys. Rev. Accel. Beams* 22 (2019), 083002. 11 p.

[BBF] W. Bartmann, C. Bracco, and M. Fraser. *Asynchronous beam dump validation test*. CERN, Geneva, Switzerland, EDMS Nr. 1698830, Jun. 2017.

[Bri09] M. Brice. "Views of the LHC tunnel in sector 3-4." Oct. 2009.

[Brü+04] O. S. Brüning, P. Collier, P. Lebrun, S. Myers, R. Ostojic, J. Poole, and P. Proudlock. *LHC Design Report*. CERN Yellow Reports: Monographs. Geneva: CERN, 2004.

[Dil+16] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan. *Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders*. 2016.

[FTG18] M. M. Fard, T. Thonet, and E. Gaussier. *Deep k-Means: Jointly clustering with k-Means and learning representations*. 2018.

[He+15] K. He, X. Zhang, S. Ren, and J. Sun. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. 2015.

[He+20] X. He, T. Gumbsch, D. Roqueiro, and K. M. Borgwardt. "Kernel conditional clustering and kernel conditional semi-supervised learning." In: *Knowl. Inf. Syst.* 62.3 (2020), pp. 899–925.

[Jia+16] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou. *Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering*. 2016.

[Lin+20] B. Lindström, P. Bélanger, L. Bortot, R. Denz, M. Mentink, E. Ravaioli, F. R. Mateos, R. Schmidt, J. Uythoven, M. Valette, A. Verweij, C. Wiesner, D. Wollmann, and M. Zerlauth. "Fast failures in the LHC and the future high luminosity LHC." In: *Physical Review Accelerators and Beams* 23.8 (2020).

[Mar12] A. Marsili. "Identification of LHC beam loss mechanism : a deterministic treatment of loss patterns." presented 21 Nov 2012. 2012.

[Neb+11] E. Nebot, B. Velghe, E. Holzer, B. Dehning, A. Nordt, M. Sapinski, J. Emery, C. Zamantzas, E. Effinger, A. Marsili, J. Wenninger, T. Baer, R. Schmidt, Z. Yang, F. Zimmerman, and N. Fuster. "Analysis of fast losses in the LHC with the BLM system." In: (Sept. 2011), 3 p.

[Obe+21] C. Obermair, A. Apollonio, Z. Charifoulline, M. Maciejewski, F. Pernkopf, and A. Verweij. "Machine Learning with a Hybrid Model for Monitoring of the Protection Systems of the LHC." In: *JACoW IPAC* 2021 (2021), 1072–1075. 4 p. DOI: `10.18429/JACoW-IPAC2021-MOPAB345`.

[Pas+19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 8024–8035.

[RFB15] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." In: *CoRR* abs/1505.04597 (2015). arXiv: `1505.04597`.

[Val+17] G. Valentino, R. Bruce, S. Redaelli, R. Rossi, P. Theodoropoulos, and S. Jaster-Merz. "Anomaly Detection for Beam Loss Maps in the Large Hadron Collider." In: *J. Phys.: Conf. Ser.* 874 (2017), 012002. 6 p. DOI: `10.1088/1742-6596/874/1/012002`.

[Vas+18] A. Vasilev, V. Golkov, M. Meissner, I. Lipp, E. Sgarlata, V. Tomassini, D. K. Jones, and D. Cremers. *q-Space Novelty Detection with Variational Autoencoders*. 2018.

[Wie+18] C. Wiesner, W. Bartmann, C. Bracco, E. Carlier, L. Ducimetière, M. Frankl, M. Fraser, B. Goddard, C. Heßler, T. Kramer, A. Lechner, N. Magnin, V. Senaj, and D. Wollmann. "Asynchronous Beam Dump Tests at LHC." In: *Proc. 9th Int. Particle Accelerator Conf. (IPAC'18)*. Vancouver, BC, Canada. May 2018, pp. 265–268.

[WL] J. Wenninger and A. Lechner. *Update of 10 Hz-like dump list*. LBOC meeting 111, 29.10.2019.

[WR19] J. Woźniak and C. Roderick. "NXCALS - Architecture and challenges of the next CERN accelarator logging service." In: 2019.

[ZG14] C. Zamantzas and S. Grishin. *LHC BLM system: Expert name convention additions*. May 2014.