Pre-emptive identification of particle accelerator failure using dimensionality reduction and label thresholding

Miha Rescic[a,*], Rebecca Seviour[a], Willem Blokland[b], Miha Rescic, Rebecca Seviour, University of Huddersfield, UK
Willem Blokland, Spallation Neutron Source, Oak Ridge, TN, US

[a] *University of Huddersfield, Queensgate, Huddersfield, HD1 3DH, UK*
[b] *Neutron Sciences Directorate, One Bethel Valley Rd, Oak Ridge, TN 37831, USA*

1. Introduction

Reliability is a crucial parameter in the design and operation of particle accelerators, and has been identified as a key factor limiting the viability of particle accelerators for many commercial applications such as energy production and waste-transmutation, which require a maximum downtime of a few seconds daily [1, 2]. This level of accelerator reliability is beyond the ability of existing machines [3–9]. Current research into particle accelerator reliability at the machine design stage have utilised conventional techniques such as Reliability Block Diagrams (RBD), Fault Tree Analysis (FTA), etc. [10–12] Although due to the bespoke nature of many of these machines the limited data for component reliability and failure modes limits the applicability and validity of conventional reliability analysis techniques [13].

In many existing accelerator facilities reliability focuses on identifying failures after they occur, where knowledge of failures is generally localized to specific failures in a subsystems (e.g. sparking in RF systems), the issue with this approach is it cannot be generalised. This is exacerbated by the large number of subsystems in an accelerator and the limited data on failures available for

*Corresponding author
Email addresses: miha.rescic@hud.ac.uk (Miha Rescic), R.Seviour2@hud.ac.uk (Rebecca Seviour), blokland@ornl.gov (Willem Blokland)

many subsystems. Several researchers have developed particle accelerator, machine specific, techniques to examine reliability such as; AvailSim 1.0 for the International Linear Collider [14], AvailSim 2.0 for International Fusion Materials Irradiation Facility (IFMIF) [15] and Monte Carlo based simulations for a linear collider [11]. These software packages have been to a certain extent validated against operating machine data, but are very reliant on the quality of data (e.g. MTBF, MTTR) fed into the simulations, and as mentioned earlier this data is not generally available (especially in case where the machine in question has not yet been built).

A better method to address accelerator reliability is a system-agnostic approach, as this reduces the dependency on needing to know all potential system failures and all potential combinations of failure modes. In this paper we present our advances to develop a system-agnostic approach to precognitive identify failure, i.e. to identify in real time a particle accelerator trip before the machine trip occurs. This development is driven in part by recent improvements in data acquisition technologies (e.g. fast Field Programmable Gate Arrays (FPGA)) and the need to record increasing amounts of data from existing particle accelerators, generating large amounts of data for individual particle bunches[16]. This bunch data contains information about both normal and non-normal accelerator operation. These large datasets, such as the Oak Ridge National Laboratory's Spallation Neutron Source (SNS) pulse data are ripe for data mining and analysis. During SNS accelerator operation a machine trip causes data acquisition of the beam current in three different time positions relative to the machine trip. The first, is data for a pulse that successfully passed through the accelerator in normal operation, just prior to the machine trip. The second, is data for the pulse in the machine at the point of trip. The third is data for the pulse that passes through the accelerator after the machine restart. This paper builds upon the holistic machine learning approach presented in [17] to identify machine failures before they occur, looking for emergent behaviour in the massive pulse datasets, identifying patterns and inference without explicit instructions.

Although machine learning approaches were first applied to particle accel-

erator in the 1980s [18, 19], with application to areas such as beamline control feedback [20, 21] it has only recently seen a significant increase in machine learning method applications, such as; LLRF feed forward control [16], RF pulses shape [22] and RF gun temperature control [23].

The authors of [17] used a machine learning approach based on binary classifiers to predict SNS machine failures via beam current measurements before the machine trips actually occurred with almost 80% accuracy. The authors of [17] demonstrated that tuning classifier parameters and pulse properties for refining datasets lead to almost 92% accuracy in classification of trip prediction. Although in practice for an accelerator, with an average number of pulses per day around 4.7×10^6, the approach of reference [17] results in 3.8×10^5 good pulses incorrectly labeled as pulses leading to a machine trip. Importantly [17] establishes there is information about impending machine failure encoded in the pulses prior to machine failure.

In this paper we build upon the approach of [17], looking at emergent behavior that could be indicative of imminent machine failure from beam pulses passing through the SNS accelerator normally. In the first part of the paper we measure the performance of previous methods on the newly acquired data and analyze any potential shortcoming. In the second part of the paper we propose and measure the performance of an updated classifier based on the newly discovered information. We finish by presenting metrics to support the feasibility of the new classifier to be used in a real-world installation to pre-emptively predict accelerator failures. We conclude the paper by presenting some next steps.

2. Motivation

With the availability of new SNS data some of the assumptions and results from previous work can be revisited and validated.

The results in [17] indicated the presence of trip information in the bad pulse preceding the trip but the 82% precision was not high enough for a real-world application as the number of False Positives would lead to reduced accelerator performance. In this paper we investigate if the applications and improvements

of the methods on the new data can yield better results more suitable for real-world applications.

Furthermore, due to the limitations of the data in [17] the assumption was made that the *After* pulses were representative of normal operation data. As new data is available (data acquired when machine is in normal operating mode) we revisit the assumption in this paper as it is crucial to the proposed classification method.

Previous work also avoided to address the question of classification speeds, a crucial requirement for any real-world applications.

Hence, this paper addresses the issues from previous work and tries to lay solid foundations for a real-world application of a failure prediction system.

3. Terminology

The research in this paper uses data from the SNS Differential Beam Current Monitor (BCM) acquired in March 2021. The BCM system acquired data on two occasions; the first is a failure of the machine leading to the Machine Protection System (MPS) to abort the beam, the second is a time-based interval data acquisition that happens every 20 minutes on a regular basis and is not failure related. We refer to the data from the first scenario as *trip data* and the later as *notrip data*.

The SNS machine runs at an operational rate of 60Hz meaning there is about 16 milliseconds between two consequent pulses during normal operation. We refer to these 16 milliseconds as the *time budget* available to make the next-pulse prediction. When measuring classification speeds we measure against this budget.

Each acquisition results in a file containing 29 pulses. If the acquisition has been triggered by the MPS system, the file contains 26 pulses leading up to the machine trip, the pulse that was aborted by the system and two pulses that successfully pass through the machine once the operational state has been restored. The 26 pulses leading to the trip are tagged with labels *Before-25* to *Before-1* and *Before* being the last pulse that successfully passed through

the machine before the pulse with the failure. The two pulses after the trip are labeled as *After* and *After+1* respectively. The tripping pulse has the tag *During*. The file might contain less than 26 pulses prior the trip if there was not enough pulses between the two failures.

The files acquired on a time interval basis always contain 29 pulses that are not trip related but contain same tags due to the nature of the system.

In our research we refer to all the pulses leading to the trip (tagged *Before-25* to *Before*) as **bad** pulses as they lead to a failure of the machine. Similarly, also for historical reasons we refer to the *After* and *After+1* pulses as **good** pulses as we assume they do not lead to a machine trip. We refer to the newly acquired, no-trip pulses simply as **notrip** pulses to make the distinction from good pulses. To be consistent with the terminology in the anomaly detection field, as the fault is the anomaly in the normal operation of the machine, we label the **bad** pulses with the positive label **1** and the good (or notrip, depending on the case) with the zero label **0**.

In this paper we refer to the classification of the bad and good pulses as the Before-After (BA) method and the classification of the bad and notrip pulses as the Before-Notrip (BN) method.

Using the terms from above, we can state that our research is focused on building a binary classifier capable of identifying and separating good pulses from bad pulses.

In the paper we use the term *True positive* (T_p) to label the correctly identified bad pulse (i.e. presence of anomaly, raised alarm), the term *False positive* (F_p) as a good pulse being falsely identified as bad (i.e. no presence of anomaly but raised alarm), the term *False negative* (F_n) to label a bad pulse mislabeled as good (i.e. presence of anomaly, no raised alarm) and *True negative* (T_n) as the good pulse correctly labeled as good (i.e. no anomaly, no alarm)

For the purpose of predicting failures, we focus on the following two: The T_p measure is important as it tells how many bad pulses a classifier is capable of detect therefore higher is better. F_p would determine how many false alarms there would be as good pulses would be mislabeled as bad. Given the nature

of the machine (much more good than bad pulses) it is more important to keep F_p low than T_p high, as there is already a system capable of preventing failures while they happen (the MPS system).

When measuring the performance of classifiers we use the term precision (P) as

$$P = \frac{T_p}{T_p + F_p}$$

Precision measures how many relevant results are returned by the classifier (i.e. how many of the actual failures are detected) Similarly, we define recall (R) as

$$R = \frac{T_p}{T_p + F_n}$$

which measures the ability of the classifier to find all the anomalies (i.e. raise all the appropriate alarms).

For example, a classifier with high recall but low precision will return many results, but most of its predictions will be incorrect when compared to the training set. Similarly, a classifier with high precision but low recall is the opposite, it will return very few results, but most of its predicted labels will be correct if compared to the training set. We aim for a classifier with both high precision and high recall as it will return many results, with all results predicted correctly.

For every iteration and use case of the classifiers in this paper we use both an exhaustive 5-fold cross-validation on the train set during training phase and a separate measurement against the independent test set once the model is trained. We mark the results throughout this paper from cross validation with label CV and independent results with *Test* respectively.

We omit listing standard deviation values from the results as they were always in the 1-2% range regardless of analysis and classifier.

Throughout the paper we provide precision and recall metrics where applicable as they give a good rough estimate of classifier performance. We present thorough analysis of T_p and F_p in the results section together with Receiver Operating Characteristics (ROC) and Precision-Recall plots.

4. Dataset, classifiers and validation

The source for our dataset is BCM acquired files in March 2021. We've picked 6.987 trip related files and 1.699 notrip files acquired during normal operation.

The BCM system records data at a speed of 100 MHz, storing each pulse as a waveform (i.e. array) of 120.000 datapoints. The duration of each acquisition is 1.2 milliseconds and the distance between two points in the array is 0.0001 milliseconds. A duration of a single pulse is about 1 millisecond in the acquired waveform.

Every pulse entry also contains measurements from both beam current monitors, referred to as *Trace-1* (upstream in the accelerator, closer to the source) and *Trace-2* (downstream in the accelerator, closer to the target). We use the measurements *Trace-1* in our research as we did not measure significant difference in performance between the two.

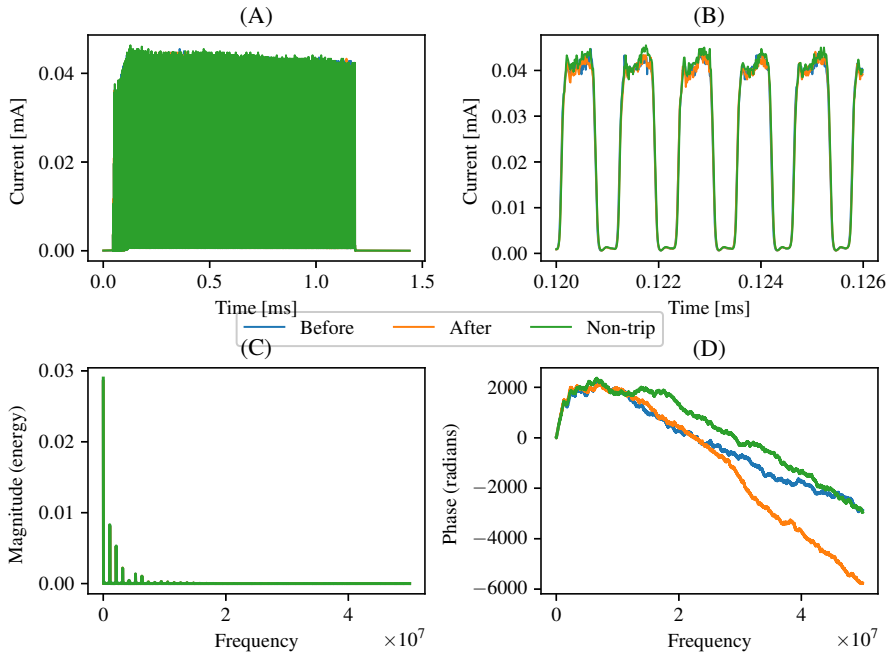
A pulse is in SNS terminology officially referred to as a *macropulse*. One such macropulse is composed of many smaller *minipulses*. The number of these minipulses is controlled by a control system parameter *bWidth*. This parameter is stored in the pulse metadata in the file alongside the pulse timestamp etc.

Figure 1 shows the overall structure of the three waveforms representative of the three pulse types: bad (Before), good (After) and no-trip related. We include the magnitude and phase spectrum plots generated using Fast-Fourier Transform (FFT) as the differences between the raw waveforms are hard to establish.

We compose each of the train and test datasets in the following fashion:

1. extract the required pulses from the files (e.g. *Before* and *After*) into separate sets,
2. perform random shuffle and split pulses into 80% train and 20% test sets,
3. label accordingly (e.g. *1* for *Before* and *0* for *After*)
4. join data to form X and y tuples (for train and test respectively)

Figure 1: Overall structure and differences of the three types of waveforms (Before, After and Non-trip): macro pulse structure (A), mini pulse structure (B), Magnitude spectrum (C) and Phase spectrum (D)



For analysis we only use pulses with the *bWidth* parameter greater than 900 as advised by SNS experts since lower values are probably not a good representative of normal operation i.e. could be odd erroneous beam or beams during beam studies. Using this filter removes some of the sample files rendering the total dataset size lower than the file count.

When the MPS system trips the machine it registers the alarm value raised by the MPS system. The value is a bitmask of the signals that triggered the trip. We distinguish two different types of trips based on what the signal bit mask is. The pulses with the alarm value of 48 (bitmask 00110000) indicate a trip where no beam loss (NBL) occurs, as the pulse was inhibited before it was injected into the accelerator. The MPS inhibits pulses if the difference in signal between two consequent pulses (*Before-1* and *Before*) is above a certain threshold. Alternatively, all other signal bitmasks represent a trip where beam loss (BL) occurred, as the MPS system aborted an ongoing beam when the difference between the upstream and downstream beam current measurements was above a certain threshold.

The NBL dataset is composed of 11.850 pulses (5.790 good and 6.060 bad) and the BL of 1.953 (1.576 good and 377 bad) pulses respectively.

In this paper we perform the validation of previous method on the NBL dataset and the performance analysis of the classifiers on two datasets, composed of NBL and BL pulses respectively, as the ability to predict the two types of trips has different implications.

As the classifier we use the Random Forest (RF) classifier with the default setting of 100 estimators, keeping the rest of the hyperparameters default too. We have evaluated the other 3 top performing classifiers listed in [17] during the writing of this paper. On a smaller subset of data the performance of k-Nearest Neighbours (kNN) and Gradient Boosting (GB) classifier was very similar to RF results. Performance issues emerged when tested against the whole dataset, as Linear Regression failed to converge, kNN classification time was too slow (around 30 milliseconds) and the training time for GB was too long (8 hours). Based on this results we decided to exclude them from analysis.

The analysis were run on a computer with Intel i7 processor with 8 1.8Ghz cores. Analysis code was written in python using the scikit-learn package. Unless stated otherwise all methods use the default hyperparameter settings as listed in the scikit-learn package documentation. For model persistence we use the dill [26, 27] package .

We've investigated potential contamination of Before pulses due to the nature of the data acquisition system (the resulting 29 pulses are merged from 2 separate buffers). We found that the before pulses were always exactly 0 pulses from the next trip pulse, whereas the notrip pulses were at least few 1.000 pulses away.

Training models on pulses of specific bWidth yielded results similar or worse than the baseline, so we did not proceed with further investigation in that direction.

Although recommended and considered best practice, data scaling using standard scaler yielded nearly identical results as the baseline so we did not include it in classification pipeline.

5. Validation of previous method

Previous work [17] established that the bad and good pulses can be distinguished by a classifier with an up to 82% precision. In this section we address this result with the new data and set a baseline. We refer to the method and the dataset as Before-After (BA).

For validation we've compiled a dataset of *Before* and *After* pulses, split 80/20 for train and test (9262 and 2316 pulses respectively). We measured precision, recall and average classification speed for both the train-set cross-validation and final evaluation with the test set.

With the availability of the new, notrip data we proceed and test the assumption that the BA model can be used to predict failures when applied to normal operating data. This was previously not possible as such data was unavailable.

For this purpose a Before-Notrip dataset was compiled composed of Before and Notrip pulses, split 80/20 train test as before, yielding 9479 and 2370 pulses

Table 1: Performance of the Random Forest (RF) classifier on the Before-After (BA) and Before-Notrip (BN) datasets

Train–Test dataset	Metric	Precision	Recall	Speed
BA–BA	CV	0.92	0.73	0.046 ms
BA–BA	Test	0.92	0.74	0.24 ms
BA–BN	Test	**0.42**	**0.74**	0.20 ms

respectively.

Table 1 shows the results for classifiers trained on the BA dataset and evaluated against both BA and BN test sets. The results listed confirm that a classifier can make a distinction between good and bad data. Unfortunately it also shows that the data previously assumed as good might not be a good representative of normal operating data as it performs worse when classifying actual notrip operations data.

Listed classification speed confirms that classification can be performed within the 16 millisecond time budget. We attribute the longer classification time on the test dataset to two reasons: during CV, the training datasets are smaller hence the decision trees are potentially smaller. Second, the RF classifiers trained during CV are mostly over-fitted hence the traversal of those decision takes less time compared to when classifying the test data.

We identified and investigated two potential causes of why the classifier performs differently when using notrip data as compared to good data.

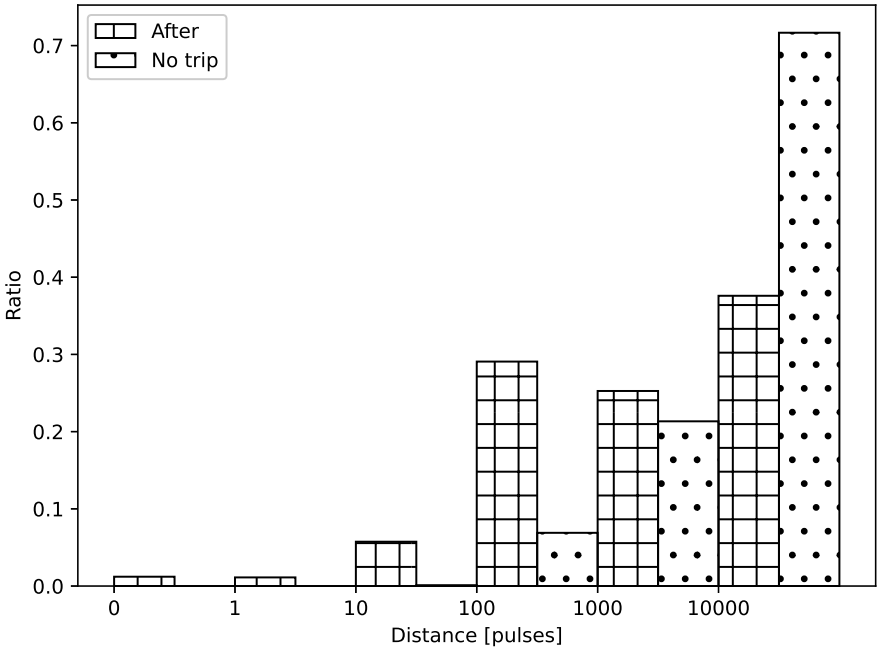
First, we assumed the *After* pulses could be contaminated by the just-occurred trip hence different from the normal operating pulses. They could also be affected by the state of the recovered machine as they were the first pulse to pass through successfully. We ran the same analysis using the second successful pulse tagged *After+1* pulse but the obtained results were unchanged.

Second, we observed that the good pulses tend to be much closer to the next trip as the notrip pulses. Figure 2 shows the distribution of measured distances. Using this we proceeded and removed pulses from the dataset based on the distance to the next trip and measured the differences in classifier performance. The results are listed in Table 2 where we can observe that classifier performance

does increase when pulses closest to the trip are incrementally removed.

Figure 2 also shows that some *After* pulses are also *Before* pulses, as the distance to next trip is 0. We could potentially classify this as dataset contamination and it could explain the better performance metrics when tested only on BA dataset.

Figure 2: Distribution of distances (in pulses) between a good pulse and the next trip pulse for the two different datasets: Before-After (BA) and Before-Notrip (BN).



6. New baseline

Using the new results from Section 5 we change the approach and start training classifiers on Before and Notrip (BN) data, as we know the latter to be a better representative of normal operations. The No-beam-loss(NBL) dataset is composed of 11.578 pulses, the Beam-loss (BL) of 1.953 (1.576 good and 377 bad) pulses respectively. The datasets are split 80/20 for training and testing.

Table 2: Performance of the RF classifier trained on BA dataset, incrementally removing pulses based on distance to next trip and tested to BN dataset.

Distance to next trip (in pulses)	Precision	Recall
0	0.43	0.69
1	0.44	0.73
10	0.43	0.69
100	0.44	0.72
1.000	0.48	0.89
10.000	0.49	0.90

We measured precision, recall and average classification speed for both the train-set cross-validation and final evaluation with the test set. Table 3 shows the performance of the RF classifier on the BN dataset confirming it can make the distinction between bad and notrip pulses.

Table 3: Performance of the RF classifier on the Before-Notrip (BN) dataset, for both No Beam Loss (NBL) and Beam Loss (BL) type of trips

Train–Test dataset	Metric	Precision	Recall	Speed
BN–BN NBL	CV	0.81	0.55	0.09 ms
BN–BN BL	CV	0.87	0.49	0.09 ms
BN–BN NBL	Test	**0.84**	**0.57**	0.48 ms
BN–BN BL	Test	**0.96**	**0.59**	0.45 ms

7. Improvements

In the following section we investigate different methods and adjustments as an attempt to improve the baseline performance listed in Table 3. All the results from this section are summarized in Table 4 and Table 5.

7.1. Frequency domain

As presented in Figure 1 there are no obvious differences between the raw waveforms of good, bad and notrip types. In this section we examine if any differences can be emphasized when transforming the waveforms into frequency domain using the Fast Fourier Transform (FFT).

The transformation is performed using numpy's fft functions generating the default half-sample-size (60.000) coefficients from the original waveform. We keep the default to avoid unnecessary information loss. We proceed by averaging the pulses over the dataset as to avoid per-pulse anomalies, Figure 3 shows that there is differences in the spectrum of the average pulse of each type. Even when averaged, the major frequencies stay the same (this does not change if a smaller subset of even a single triplet of pulses is considered).

We include *Before-25* pulse amplitude and phase in Figure 3 as to hint that it too differs from normal operating, notrip data.

For reference, the top 10 represented frequencies in all pulses are, in descending order: 833Hz, 1.6kHz, 2.5kHz, 5kHz, 1.05MHz, 1.050833MHz, 1.051666MHz, 2.102500MHz, 2.103333MHz, and 3.154166MHz. The frequencies around 1.05 MHz and its multiples are the mini-pulses harmonics (as the minipulse chopper runs at 1 microsecond). The rest of the frequecies we cannot correlate to any machine characteristics.

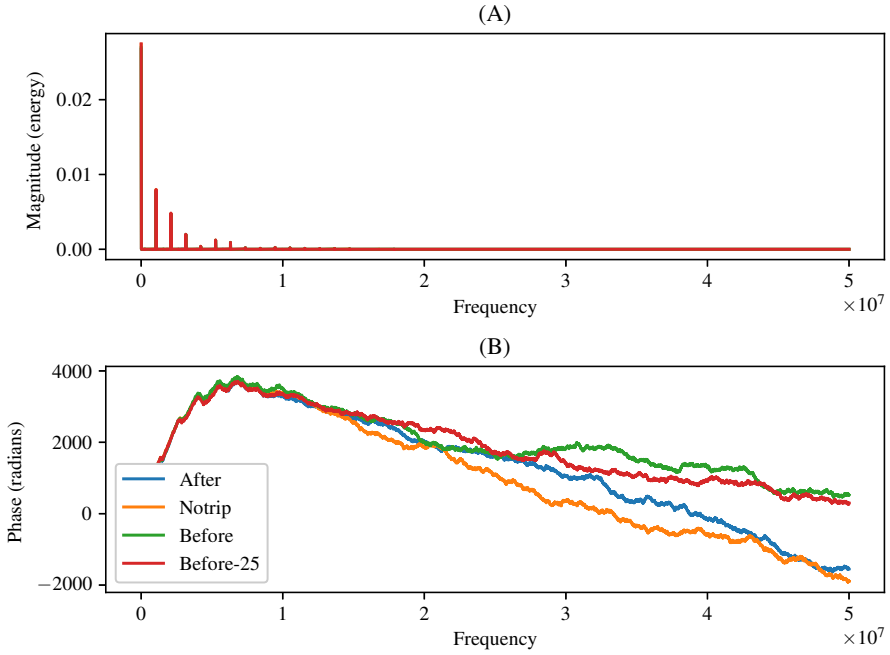
7.2. Principal Component Analysis

As our raw data is a waveform it is feasible to try different decomposition methods. There are two potential benefits to this approach: improved classifier performance as feature differences are made more clear and improved classification speed as the number of features is reduces.

PCA is a statistical procedure that essential converts a set of observations with an orthogonal transformation into a data set of linearly uncorrelated variables referred to as the *principal components*. Each waveform is then represented by a linear combination of k of these principal components where k is an arbitrary value between 1 and 120.000, all the features. We chose the value of k to be 100 as it yielded best results and those components combined explained more than 99% of the original sample variance.

Results in Table 4 show slower classification times when using PCA preprocessing compared to raw data. We attribute this to the time it takes to perform a SVD decomposition of a waveform prior to classification, an important step

Figure 3: Magnitude and phase spectrum of the average Before, After and Non-trip pulse



in the PCA process.

We omit plotting the components as we failed to find any relevance to pulse type or machine characteristics.

Table 4: Performance of the RF classifier on the BN dataset when using different improvements

Improvement	Dataset	Metric	Precision	Recall	Speed
FFT + RF	NBL	CV	0.88	0.52	0.55 ms
FFT + RF	NBL	Test	0.88	0.51	3.6 ms
PCA + RF	NBL	CV	0.86	0.55	0.47 ms
PCA + RF	NBL	Test	0.90	0.55	1.13 ms
FFT + RF	BL	CV	0.88	0.47	0.54 ms
FFT + RF	BL	Test	0.98	0.61	3.87 ms
PCA + RF	BL	CV	0.89	0.49	0.18 ms
PCA + RF	BL	Test	0.96	0.61	0.86 ms

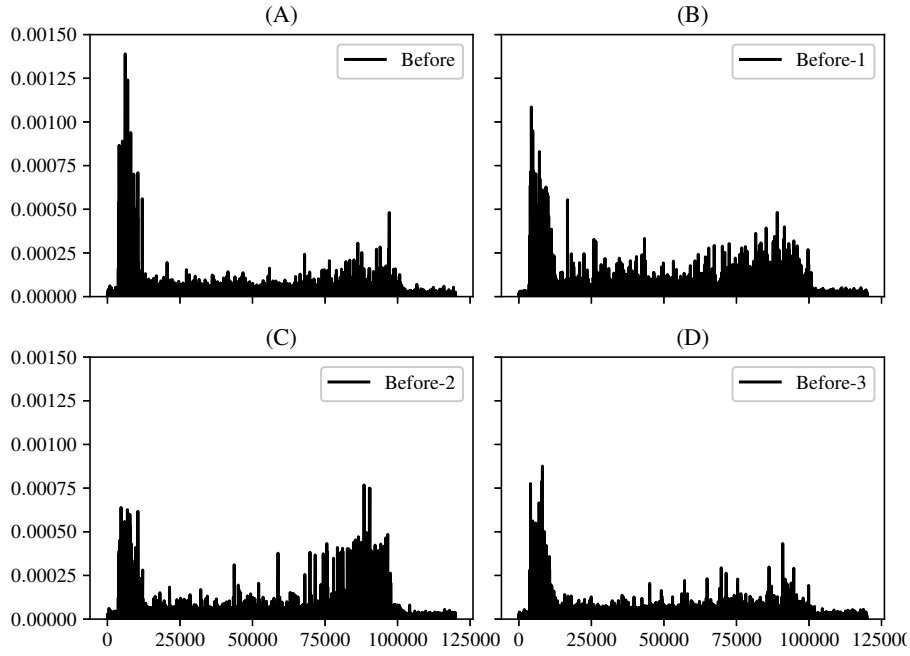
7.3. Increasing warning time

As the new datasets offers pulses up to 25 pulses before the trip we've investigated the performance of a classifier when trained with pulses tagged Before-x, x going from 1 to 25 (0 is the baseline). As this classifier would be able to predict the Before-x type of pulse this would allow the increase of the prediction window from 1 pulse to N pulse, N being between 2 and 25. We evaluate datasets for N from 1 to 5 and label these datasets with labels B-1N to B-5N respectively. This specific analysis is only performed on the NBL dataset as the sample size of the BL dataset is too small causing classifier performance to drop rapidly.

Table 5: Performance of the RF classifier on different variations of the BN dataset

Dataset	Metric	Precision	Recall	Speed
RF, B-1N	CV	0.89	0.59	0.05 ms
RF, B-2N	CV	0.81	0.55	0.05 ms
RF, B-3N	CV	0.75	0.41	0.05 ms
RF, B-4N	CV	0.76	0.39	0.05 ms
RF, B-5N	CV	0.74	0.37	0.05 ms

Figure 4: Feature importance of pulses Before (A), Before-1 (B), Before-2 (C) and Before-3 (D). The higher the importance the more important is a feature for classification. x-axis represents the position within the waveform.

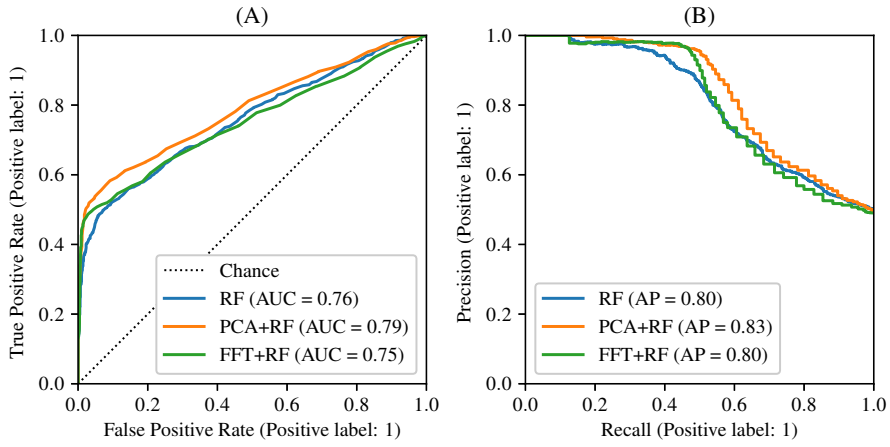


8. Feature importance

The RF classifier contains the information about which dataset features are important (and the related importance score) once trained on a dataset. This information can be used to observe what part of the waveform is the most crucial when the classification is made. Knowing what region of a waveform is more likely to contain information about the failure can be used to improve the detection of failures in the MPS system.

Figure 4 shows this information for pulses with tags ranging from Before to Before-3. We can observe that the most important features are at the beginning, at the ramp-up section of the waveform and that the importance drops as we move away from the tripping pulse. The latter is consistent with performance loss of classifiers trained on pulses Before-N.

Figure 5: Receiver Operating Characteristic (ROC) curve (A) and Precision-Recall curve (B) for the three classifiers on No beam loss (NBL) dataset.



9. Results

In this section we look in detail at classifier performance and how different improvements impact both T_p and F_p metrics. The Receiver Operating Characteristics (ROC) plots listed in this section show the relation between T_p and T_n as the classifier decision function threshold (probability of a label) is changed.

Figure 5 shows how the performance of the classifier on the NBL dataset is improved using different techniques (in ascending order of performance): FFT, PCA and baseline RF. Similarly, Figure 6 show that the classifier performance is already better on the BL dataset without any additional improvements, as PCA and FFT only slightly increase performance.

Figure 7 shows how the performance of the RF classifier changes as it's trained to classify Before-N pulses, as N moves from 1 to 5 on the NBL dataset.

Figure 6: Receiver Operating Characteristic (ROC) curve (A) and Precision-Recall curve (B) for the three evaluated classifiers on Beam-loss (BL) dataset.

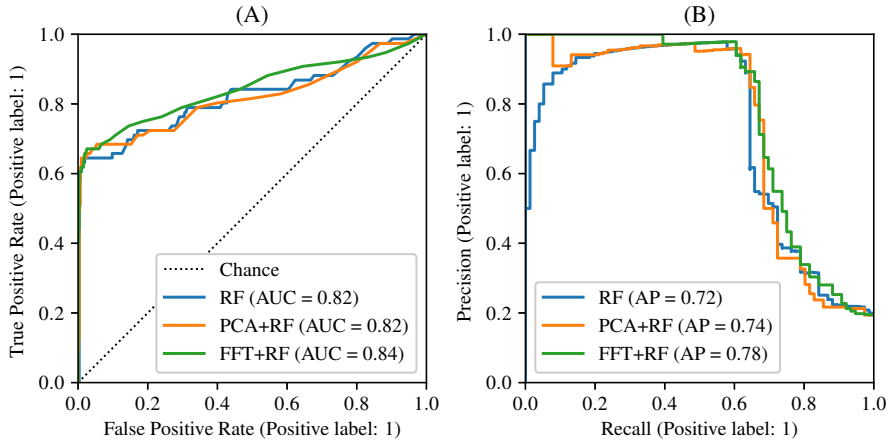
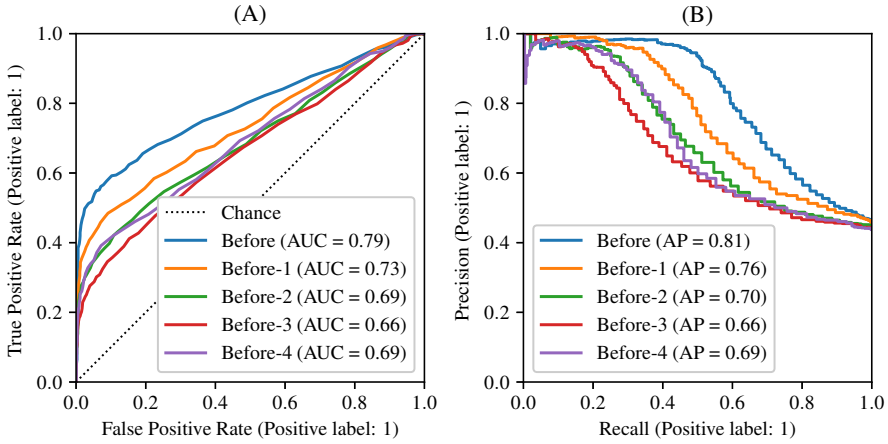


Figure 7: ROC curve (A) and Precision-Recall curve (B) for the five classifiers trained on Before-x pulses on NBL data



The classifier performance on the BL dataset degraded quickly making predictions beyond *Before-1* impossible. Regardless, the results tells us it is possible to distinguish Before-N pulses from normal operation in the NBL pulses, hence increase warning time, but the performance degrades as we move further away from the trip pulse.

The classifier threshold probability affecting T_p and F_p values is not visible in the charts so we list some information separately in Table 6. In that table we consider two different F_p values: first, we consider a value in the closest order of magnitude as the probability of a trip during normal operation (roughly 0.000045 [1]). We refer to this threshold as Th_t.The second value for F_p that is relevant is is 0.01, as it was communicated to us by SNS experts that such margin of error would be acceptable. We refer to this value as Th_e.

The measurements in Table 6 show that the classifier can be tuned to achieve the required performance for real-life applications.

Table 6: List of label thresholds and the effect on T_p and F_p for different classifiers

Classifier	Dataset	Th_t	T_p	F_p	Th_e	T_p	F_p
RF	NBL	0.88	0.13	0.00083	0.70	0.32	0.0099
FFT + RF	NBL	0.91	0.22	0.00083	0.64	0.44	0.0099
PCA + RF	NBL	0.91	0.15	0.00330	0.74	0.38	0.0090
RF	BL	0.69	0.58	0.00316	0.41	0.62	0.0094
FFT + RF	BL	0.80	0.49	0.00316	0.40	0.64	0.0094
PCA + RF	BL	0.96	0.36	0.00293	0.87	0.60	0.0088

10. Conclusions

In this paper we have aimed to validate previous methods on the newly acquired data and proceed with building and evaluating a classifier capable of successfully predicting failures. The information we were after was the quality

[1]The value is calculated by dividing the number of trips in month of March 2021 (cca 7.000) by the estimated number of all possible good pulses during the same period(cca 155 million)

of predictions and the speed of predictions as those two metrics determine real-world applications.

In Section 5 we've successfully confirmed findings from our previous paper [17] that a classifier can distinguish between *Before* and *After* pulses. We've also found that the *After* pulses, previously assumed a good representative of normal operation were quite the opposite.

We have then advanced that approach by establishing a new baseline in Section 6 that a Random Forest classifier can distinguish between bad and notrip pulses, which is by far a more realistic, real-world scenario.

In Section 7 we listed some intuitive improvements to the classifier while at the same time measured that all examined classifiers are fast enough to fit in the 16 milliseconds time budget available.

Section 7.3 shows that the warning time for a failure can be extended to more than one pulse using the Before-N pulses to train models and in Section 8 we extract, from the classifier, the regions of potential interest in the pulses for further optimizations of predictions.

Most importantly, we have successfully presented classifiers that are fast enough (1-3ms) to fit between the time gap between two pulses (16 ms) and capable of identifying up to 44% of bad NBL pulses and up to 64% bad BL pulses while maintaining the F_p rate under 1%. The SNS experts confirmed that such error rate is acceptable and applying mitigation using the classifier would be an improvement over the current setup. Currently SNS loses more pulses (beam time) due to downtime due to trips (estimated 55.000) compared to pulses lost due to false positives.

11. Next steps

As more data is collected, such data can be used to train better models and as sample sizes increase, more detailed analysis of pulse properties can be performed. We believe that joining pulse data with related metadats is the key to further improve classification performance. The MPS system collects the information on failing systems that can be correlated with the trip pulse

acquisition. This opens up the opportunity to both better refine the datasets or investigate potential unsupervised learning approaches to failure identification.

References

[1] Nuclear Energy Agency. Organisation for Economic Co-operation and Development. NEA Accelerator-driven Systems (ADS) and Fast Reactors (FR) in Advanced Nuclear Fuel Cycles: A Comparative Study, 2002.

[2] S. Henderson et al. Accelerator and Target Technology for Accelerator Driven Transmutation and Energy Production White Paper Working Group Report. 2010.

[3] Bauer, G. S., SINQ Status report, Proceedings, 15th Meeting of the International Collaboration on. Advanced Neutron Sources (ICANS), November 6-9,2000. Tsukuba, Japan, pages 103–107, 2000

[4] Groeschel, F. and Dementjev, S. and Heyck, H. and Leung, W. and Thomson, K. and Wagner, W. and Zanini, L., MEGAPIE Irradiation Experience of the First Megawatt Liquid Metal Spallation Target, Proceedings, 5th International Workshop on the Utilization and Reliability of High Power Proton Accelerators (HPPA5), May 6-9 2007, Mol, Belgium

[5] Galambos, John and Kim, SH and Campisi, I, SNS Experience with a High-Energy Superconducting Proton Linac, CARE-HHH-APD Event BEAM'07, Finalizing the Roadmap for the Upgrade of the LHC and GSI Accelerator Complex, 1-5 October 2007, CERN, Switzerland

[6] Galambos, John, SNS Operational Experience at the MW Level SNS Operational Experience, Proceedings, 46th ICFA Advanced Beam Dynamics Workshop on High-Intensity and High-Brightness Hadron Beams, (HB2010), September 27-October 1, 2010, Morschach, Switzerland

[7] Galambos, John, SNS Performance and the Next Generation of High Power Accelerators, Proceedings, North American Particle Accelerator Confer-

ence (NAPAC), September 29-October 4 2013, Pasadena, California, USA, pages 1443–1447

[8] Kim, S H, Afanador, R, Blokland, W, Champion, M, Coleman, A, Crofford, M, Degraff, B, Doleans, M, Douglas, D, Gorlov, T, Hannah, B, Howell, M, Kang, Y, Lee, S-w , Mcmahan, C, Neustadt, T, Ottaway, S, Peters, C, Saunders, J, Shishlo, A, Stewart, S, Strong, W H and Vandygriff, D The Status of the Superconducting Linac and SRF Activities at the SNS, Proceedings, 16th International Conference on RF Superconductivity, September 23-27 2013, Paris, France, pages 83–88

[9] Campisi, I E, Assadi, S, Casagrande, F, Crofford, M, Dodson, G, Galambos, J, Giannella, M, Howell, M, Kang, Y, Kasemir, K, Kim, S H, Kursun, Z, Ladd, P, Ma, H, Stout, D, Zhang, Y and Champion, M, Status and Performance of the Spallation Neutron Source Superconducting Linac, Proceedings, 22nd Particle Accelerator Conference (PAC), June 25-29, Albequerque, New Mexico, USA, 2007, pages 2502–2504

[10] Pierini, Paolo and Burgazzi, Luciano, ADS Accelerator Reliability Activities in Europe Utilisation and Reliability of High Power Proton Accelerators: Workshop Proceedings, Daejeon, Republic of Korea, May 2004

[11] N Phinney, T Himel, and MC Ross. Reliability Simulations for a Linear Collider. In 9th European Particle Accelerator Conference, 5 - 9 July, Lucerne, Switzerland, 2004, pages 857 -859, 2004.

[12] Pitigoi, A E and Ramos, P Fernández and Agrupados, Empresarios and Ea, Empresarios Agrupados, Modelling High-power Accelerators Reliability - Reliability Model of SNS Linac, Proceedings, 39. Annual Meeting of Spanish Nuclear Society; 39. Reunion Anual Sociedad Nuclear Espanola, 25-27 September, Reus, Tarragona, Spain, 2013

[13] Pierini, P, Reliability studies of the PDS-XADS accelerator, PDS-XADS deliverable 57: Potential for Reliability Improvement and Cost Optimization of Linac and Cyclotron Accelerators, 2003

[14] T Himel, J Nelson, and M Ross. Availability and reliability issues for ILC. pages 1966, 2007.

[15] Enric Bargallo, Pere Joan Sureda, Jose Manuel Arroyo, Javier Abal, Alfredo De Blas, Javier Dies, Carlos Tapia, Joaqun Molla, and Angel Ibarra. Availability simulation software adaptation to the IFMIF accelerator facility RAMI analyses. In Fusion Engineering and Design, volume 89, pages 2425-2429. Elsevier B.V., 2014.

[16] M Laverty and K Fong. An iterative learning feedforward controller for the triumf e-linac. Proceedings of LINAC2016, East Lansing, MI, USA, pages 485-487, September 2016.

[17] M.Rescic, R. Seviour and W. Blokland, Predicting particle accelerator failures using binary classifiers Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 955, 2020

[18] T Higot, H Shoaee, and Stanford Linear Accelerator. Some Applications of AI to the Problems of Accelerator Physics. pages 701-703, 1987.

[19] A Corneliusen P Terdali T Knight and J Spencer. Computation and Control with Neural Nets. 1989.

[20] M Lee, R Sass, and H Shoaee. Accelerator and feedback control using neural networks. 1991:1-3, 1991.

[21] A L Edelen, S G Biedron, B E Chase, D Edstrom Jr, S V Milton, and P Stabile. Neural Networks for Modeling and Control of Particle Accelerators. 63(2):878-897, 2016.

[22] Amin Rezaeizadeh, Paul Scherrer Institut, Thomas Schilcher, Paul Scherrer Institut, and Roy Smith. Rf pulse attening in the swissfel test facility based on model-free iterative learning control. Proceedings of FEL2014, Basel, Switzerland, August 2014, pages 824-827.

[23] A L Edelen, S G Biedron, S V Milton, Fort Collins, B E Chase, D J Crawford, N Eddy, D Edstrom Jr, E R Harms, J Ruan, J K Santucci, and P Stabile. Initial experimental results of a machine learning-based temperature control system for an rf gun. Proceedings of IPAC 2015, May 3-8 2015, Richmond, VA, pages 1217-1219.

[24] Kesselman, M. and Witkover, R. and Doolittle, L. and Power, Jerome, SNS project-wide beam current monitors AIP Conference Proceedings 546, 464 (2000)

[25] W. Blokland, C.C. Peters, T.B. Southern. Enhancements to the SNS differential current monitor to minimize errant beam, IBIC2019, 2019, Malmö, Sweden.

[26] M.M. McKerns, L. Strand, T. Sullivan, A. Fang, M.A.G. Aivazis, "Building a framework for predictive science", Proceedings of the 10th Python in Science Conference, 2011; http://arxiv.org/pdf/1202.1056

[27] Michael McKerns and Michael Aivazis, "pathos: a framework for heterogeneous computing", 2010- ; https://uqfoundation.github.io/project/pathos