ISBN: 978-3-95450-261-5 ISSN: 2673-7000 doi: 10.18429/JACoW-NAPAC2025-MOP057

# MACHINE LEARNING AT THE SPALLATION NEUTRON SOURCE ACCELERATOR AND TARGET*

W. Blokland[†,1], A. Kasparian[2], K. Rajput[2], A. Raj[1], M. Schram[2], D. Winder[1], S. Zhukov[1]
[1]Oak Ridge National Laboratory, Oak Ridge, TN, USA
[2]Thomas Jefferson National Accelerator Facility, Newport News, VA, USA

*Abstract*

We describe the ongoing efforts to apply Machine Learning techniques to improve the performance of our accelerator and target. Specially, we are looking to minimize halo beam losses in the absence of a proper physics model, automatically detect and log anomalies in the target support systems such as cooling, and detect and prevent errant beam pulses in the linac. We also describe the infrastructure we use to acquire and stream data to the GPU cluster for training, our code development cycle, and edge computing for model inference. To minimize halo beam losses, we use a Reinforcement Learning technique tested on a virtual accelerator. The target anomaly detection is trained on archived data using unsupervised learning and is made part of the existing target reporting system. The errant beam prevention analyses beam current and beam phase waveforms as well as accelerator configuration data to predict errant pulses. We also develop continual learning to adapt to changes in the accelerator.

## INTRODUCTION

A review of the existing ML applications in particle accelerator applications can be found in Refs. [1, 2]. It shows that most ML solutions do not reach deployment in operational settings. We aim to not only research how Machine Learning (ML) can improve operations but to also integrate ML into Operations Spallation Neutron Source (SNS). To demonstrate our approach, we select problems that, when solved, have a chance to improve accelerator or target performance gains within a reasonable timescale. Previously, we found that ML can do prognostics, e.g. determining that a high-power supply is going to fail, but fixing the power supply is as time consuming as waiting for the equipment to fail. Thus, it is not a good use-case for improving performance. Based on previous experience, we applied the following criteria to select use-cases:

- Data must be available early in the project.
- The ML model, trained on data from the past, must function in the future.
- There must be an acceptable false positive rate.
- The result must be actionable.

*This work was supported by the DOE Office of Science, United States under Grant No. DE-SC0009915 (Office of Basic Energy Sciences, Scientific User Facilities Program). This manuscript has been authored by UT-Battelle, LLC, under Contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The Jefferson Science Associates (JSA) operates the Thomas Jefferson National Accelerator Facility for the U.S. Department of Energy under Contract No. DE-AC05-06OR23177. This research used resources at the SNS, a DOE Office of Science User Facility operated by the ORNL.
†blokland@ornl.gov

The selected use-cases based on these criteria are:

- Errant Beam Prognostics: ML aborts the accelerator before an errant beam pulse causes beam loss or damages the accelerator.
- Automatic and fast optimization of beam orbit to reduce beam losses: our goal is to halve the tune-up time which can last from an hour to a day.
- Detect and predict anomalies in the target support cooling and mercury loops: report anomalies in time for intervention.

We must add infrastructure for ML applications as it requires larger data streams and data storage than a typical accelerator control system. ML requires large data sets and significant computing power, especially GPUs, for training. The ML infrastructure needs to be within the accelerator network because even if the network needs to be air gapped, the ML applications must still be able to operate.

## USE-CASES

### *Errant Beam Prognostics*

The Differential Beam Current monitor supplies most of the data for the Errant Beam Prognostics (EPB). Previous studies [3] showed that precursors of errant beam were hidden in this data. This is important as we want to abort before the errant beam pulse is issued and not during to prevent the damage or radioactivation from happening. Our goal is to show that we can abort beam pulses before the errant beam pulse occurs.

We look at the True Positive Rate (TPR) at a very low False Positive Rate (FPR) < 0.5% to avoid unnecessarily abort beam. On a static dataset, we had implemented uncertainty aware Machine Learning model and achieved about a 60% TPR [1, 4]. We found that the TPR is roughly 30% for a non-static set (aka during Operations).

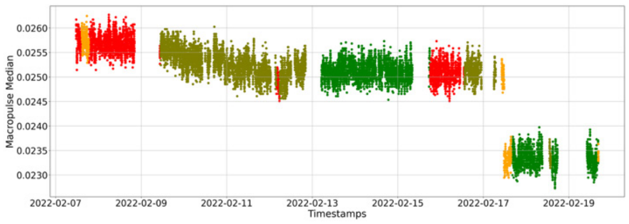

Figure 1: Changes in the beam current. Different colors represent data from different beam settings configuration.

The main challenge for operations is that we can only train on historical data and must infer during operations when the accelerator data varies both due to configuration changes and due to drift and the data changes (see Fig. 1). A configuration change is when the operators change a setting to the accelerator, e.g. a change in beam pulse length. A drift is any change not originating from changes in the observed settings.

While we could train over a large time span to capture most conditions, we might not be able to handle changes made to the accelerator and would have to start over and collect data over a long period. We opt instead to implement online continual and adaptive learning.

Our models use beam settings as a conditional input. We have evaluated both supervised conditional Siamese Model, and semi-supervised reconstruction-based conditional variational auto-encoder, methods. While supervised model has shown to outperform semi-supervised method in our application [2], we do not have sufficient anomaly data to train during the adaptation phase in online settings. As such, we are deploying a conditional auto-encoder that only requires normal samples for training, for online anomaly prediction and adaptation to drifting data.

To adapt the deployed model when data drifts, we implement memory based continual learning methods [5], with a constant size replay by selecting the most representative samples from previous data distribution and keeping them in memory for future model updates. Our online deployment and adaptation workflows are shown in Fig. 2.

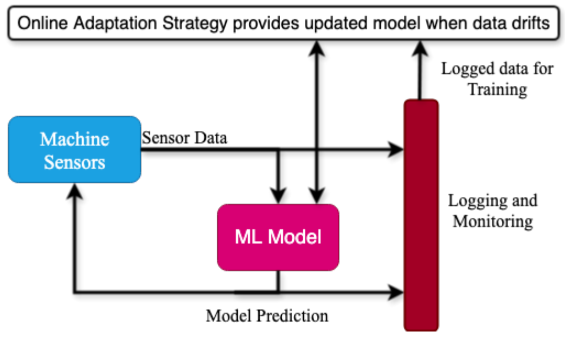

Figure 2: Online Inference and Adaptation Workflow. Logging and Monitoring include triggers for model update.

## *Beam Loss Optimization*

The beam loss optimization use-case applies ML techniques to tune up the accelerator to reduce halo losses. No physics models are available for halo losses, and it is typical for the operators to adjust magnets until halo losses are acceptable. This relies on the operator experience to do these fine adjustments. With a H- linac, such as at SNS, an additional complication is that intra-beam scattering increases if the beam is too tightly focused leading to additional restraints. Our goal is to apply ML techniques to halve the time it takes to tune the accelerator for losses and give this time back to the experimenters.

The Machine Protection System (MPS) aborts beam when settings changes and/or beam losses exceed thresholds. However, an abort by the MPS interrupts beam to the target. To avoid this interruption to ongoing experiments, we created a proxy server that ensures an application does not unintentionally trigger the MPS. These limits are based on operators' experience. The proxy server can be used by any algorithm, not just ML-based, and serves as an administrative method to allow the use of an experimental algorithm. In addition, we created a simulation of the accelerator, VIRAC, that interfaces through EPICS control system similar to the accelerator. This allows us to test the algorithm logic before applying to the real accelerator (see Fig. 3). While VIRAC can simulate many aspects of the accelerator using PyORBIT [6], it cannot simulate the halo beam losses as no physics model is available.

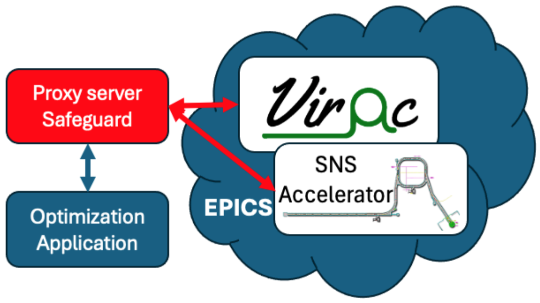

Figure 3: Proxy server and Virac tools.

The proxy, implemented in python, can run as a standalone process or within an optimizations script. The proxy saves all measurements and states in an HDF5 scan file. We tested it on real SNS accelerator with production beam running at 1Hz. The control variables, two RF cavity phases and two quadrupoles, were selected from operations experience. The optimization script was running in four-dimensional configuration space trying to minimize sum of two specific loss monitors out approximately sixty. Constraints included individual Beam Loss Monitors (BLM) thresholds enforced by MPS, sum of all losses was not allowed to grow more than 50% and individual loss was not allowed to increase over 75%. We used differential evolution [7] from SciPy to efficiently mitigate noise of BLMs. Within one hour of optimization the algorithm was able to decrease the sum of the two specific loss monitors by 8% percent while trading it for 30% increase of a different BLM and thus satisfying the objective.

In another test, we showed that Reinforcement Learning (RL) trained on the VIRAC for orbit correction was able to match the calculated optimal trajectory.

We've explored new ways to leverage historical loss data to create data driven surrogate models to aid in the development of control systems. We've introduced imitation learning algorithms that combine RL with supervised learning techniques [8], ultimately showing that agents initialized with imitation learning achieved better starting performance than randomly initialized agents on our simulated benchmark environments. These imitation learning agents are an extension of the DDPG/TD3/SAC off policy RL algorithms utilized for classical control. We aim to apply these algorithms to existing SNS beam loss data to jumpstart learning when running on the live system.

## *Target Anomaly Detection*

The reliability of target systems is a significant challenge for the SNS reliability. From FY20 through FY22, target systems were responsible for 35% of the overall hours of downtime. The target systems include more than thirty interconnected systems that interact in complicated ways with each other and with the accelerator beam. Target operational shift technicians (OSTs) monitor all systems during operations and report anomalies to the system engineers. All anomaly detection and reporting is done manually and thus labor intensive. The entries are made in an electronic logbook called the STAR System (System Tracking And Reliability). Our goal is to use ML to detect anomalies during operations and enter these automatically in the STAR system and alert operators.

We have implemented a system where operational data is regularly obtained from the archiver, and unsupervised anomaly detection methods are used to flag time periods of interest in the STAR System [9]. These potential anomalies are then graded by experts in the STAR system. This provides two benefits, first, the anomalies found can be used to prioritizing maintenance and upgrades. Second, the grading of the potential anomalies adds to our data that can be used in the future to aid supervised or semi-supervised anomaly detection models. We have seen good results in detecting anomalies, especially when using LSTM autoencoders, with a precision of over 90% for anomalies flagged by more than one LSTM autoencoders.

Figure 4 shows the general flow of data. Data obtained from the accelerator and target archiver is copied and preprocessed weekly. Anomaly detection is performed on a local DGX cluster, and the resulting anomalies are stored in an oracle database that is accessed through the STAR system. Those anomalies are evaluated by humans, who can then take any measures necessary to address the issues found. The results of their evaluations are stored in the oracle database for future improvement of the anomaly detection methods.

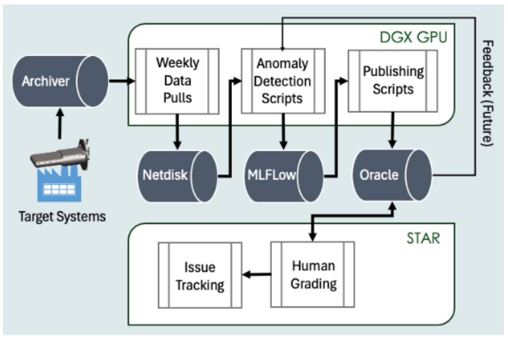

Figure 4: Target anomaly detection system diagram.

## INFRASTRUCTURE

We use Linux-based edge computers to collect data from the beam diagnostics instruments and run the inference on the data. The edge computer also displays the inference results to the operators and streams compressed data to the GPU and storage system for ML learning. New models and their performance will be presented to the Operators and, when approved, installed on the edge computer (see Fig. 5). Figure 6 shows how we use GitLab Runners to trigger training sessions from outside the accelerator network on the internal GPU server. This is GPU farm containing 8 H100 Nvidia GPU cards. The training is triggered by the user from the external Enterprise network. The results and models are logged onto MLFlow server hosted on GitLab to enable repeatability and model tracking and serving.

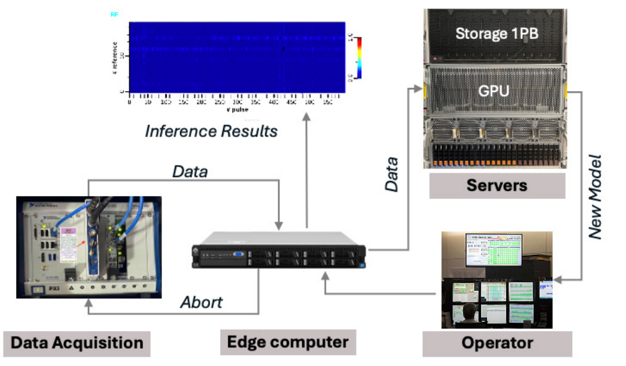

Figure 5: Edge computing, storage and GPU infrastructure inside Accelerator Network.

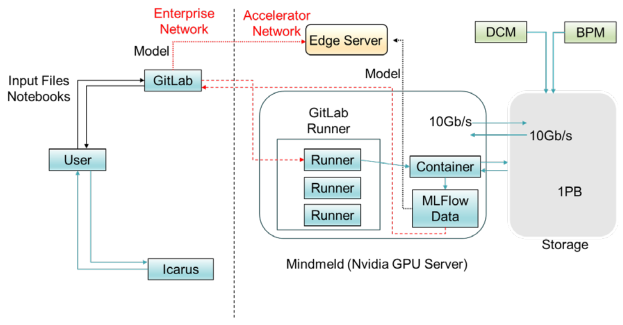

Figure 6: Workflow using GitLab Runners.

## FUTURE

We have successfully implemented a computing infrastructure for ML in Operations and have the integrated the Target Anomaly Detection into Operations. We are adding additional data from beam position monitors, specifically the phase data, as this is directly related to the acceleration process and can gain us better performance [10]. This should help us achieve a higher true positive rate. The Beam loss optimization will be applied to the accelerator to compare performance with different algorithms. The Target Anomaly Detection will be expanded to encompass additional support systems and integrate physics models. We also plan to provide the results much faster to the operators by running the scripts every hour or faster. We will add more operator tools to manage the approval of newly trained ML models for operational use.

## ACKNOWLEDGEMENTS

The authors wish to thank Allie Morris for her work on the target anomaly grading, Carrie Elliott for her work on the proxy server, and Brad Webb and Katia Danilova for their work with the STAR system.

## REFERENCES

[1] K. Rajput, M. Schram, and K. Somayaji, “Uncertainty aware deep learning for particle accelerators”, Jun. 2023,

2309.14502 [cs.LG].
doi:10.48550/arXiv.2309.14502

[2] K. Rajput *et al.*, "Robust errant beam prognostics with conditional modeling for particle accelerators", *Mach. Learn.: Sci. Technol.* v. 5 pp. 015044, 2024.
doi:10.1088/2632-2153/ad2e18

[3] M. Reščič *et al.*, "Improvements of pre-emptive identification of particle accelerator failures using binary classifiers and dimensionality reduction", *Nucl. Instrum. Methods Phys. Res., Sect. A*, v. 1025, pp. 166064, 2022.
doi:10.1016/j.nima.2021.166064

[4] W. Blokland *et al.,* "Uncertainty aware anomaly detection to predict errant beam pulses in the Oak Ridge Spallation Neutron Source accelerator", *Phys. Rev. Accel. Beams*, v. 25 pp. 122802, 2022.
doi: 10.1103/PhysRevAccelBeams.25.122802

[5] K. Rajput *et al.,* "Outlook towards deployable continual learning for particle accelerators", *Mach. Learn.: Sci. Technol.*, v. 6, pp. 031001, 2025.
doi:10.1088/2632-2153/adeb45

[6] A. Shishlo *et al.*, "The particle accelerator simulation code PyORBIT", *Procedia Comput. Sci.*, v. 51, no. 1, pp. 1272–1281, 2015. doi:10.1016/j.procs.2015.05.312

[7] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces", *J. Global Optim.,* v. 11, pp. 341 – 359, 1997.
doi:10.1023/a:1008202821328

[8] T. Hester *et al.*, "Deep Q-learning from demonstrations", Apr. 2017, arXiv:1704.03732 [cs.AI]
doi:10.48550/arXiv.1704.03732

[9] A. Raj *et al.*, "Automating anomaly detection for target systems at Spallation Neutron Source", in *Proc. NPIC&HMIT'25*, Chicago, IL, USA, Jun. 2025, pp. 286 – 295.

[10] Y. A. Yucesan *et al.*, "A machine learning approach for particle accelerator errant beam prediction using spatial phase deviation", *Nucl. Instrum. Methods Phys. Res., Sect. A*, v. 1063, pp. 169232, 2024.
doi:10.1016/j.nima.2024.169232