DISENTANGLING BEAM LOSSES IN THE FERMILAB MAIN INJECTOR ENCLOSURE USING REAL-TIME EDGE AI

K.J. Hazelwood, M.R. Austin, J.M. Arnold, J.R. Berlioz, P. Hanlet, M.A. Ibrahim, J. Mitrevski, V.P. Nagaslaev, D.J. Nicklaus, G. Pradhan, A.L. Saewert, B.A. Schupbach, K. Seiya, R.M. Thurman-Keup, N.V. Tran, A. Narayanan[1]
Fermi National Accelerator Laboratory*, Batavia, IL USA
[1]also at Northern Illinois University, DeKalb, IL USA
J.YC. Hu, C. Xu, J. Jiang, H. Liu, S. Memik, R. Shi, A.M. Shuping, M. Thieme
Northwestern University†, Evanston, IL USA

Abstract

The Fermilab Main Injector enclosure houses two accelerators, the Main Injector and Recycler Ring. During normal operation, high intensity proton beams exist simultaneously in both. The two accelerators share the same beam loss monitors (BLM) and monitoring system. Deciphering the origin of any of the 260 BLM readings is often difficult. The Accelerator Real-time Edge AI for Distributed Systems project, or READS, has developed an AI/ML model, and implemented it on fast FPGA hardware, that disentangles mixed beam losses and attributes probabilities to each BLM as to which machine(s) the loss originated from in real-time. The model inferences are then streamed to the Fermilab accelerator controls network (ACNET) where they are available for operators and experts alike to aid in tuning the machines.

PROJECT OVERVIEW

The Accelerator Real-time Edge AI for Distributed Systems (READS) project is a collaboration between the Fermilab Accelerator Directorate and Northwestern University. It aims to implement ML models on edge hardware for use on the Fermilab accelerator complex. The project consists of two sub-projects; improving Delivery Ring resonant extraction regulation [2–5] for the future Mu2e experiment [6] and aiding in the machine attribution of beam loss in the Main Injector enclosure [7].

Disentangling Beam Losses

The Fermilab Main Injector enclosure houses two accelerators; the Main Injector (MI) and the Recycler Ring (RR) (Fig. 1). The 8 GeV permanent magnet Recycler Ring acts as a proton stacker for the 120 GeV synchrotron Main Injector. To ensure the most protons are delivered to Fermilab's experiments, the Recycler Ring is loaded with Main Injectors next pulse of beam while the current MI pulse is accelerated and then extracted. During normal operations, there are high intensity proton beams in both Recycler Ring and Main Injector [8]. The two machines share the same beam loss monitors (BLM) and monitoring system. When beam losses occur, it can be difficult to attribute the origin of the loss to either machine resulting in delays tuning the machines and unnecessary downtime. However, machine experts are often able to decipher loss origin from the time in the cycle of the loss, the current machine states, local and global loss patterns and tunnel residual dose rate surveys (Fig. 3) [9, 10]. This suggests that given enough information, a ML model can be created to replicate, automate and perhaps improve upon the machine experts ability to attribute beam loss to the correct machine.

Figure 1: The Main Injector enclosure. Main Injector (bottom), Recycler Ring (top), P1 extraction beamline (middle).

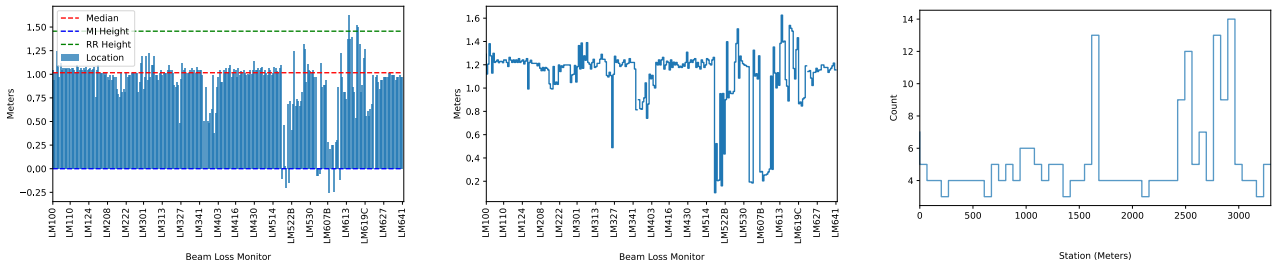
DATASETS

For this project, the ML models were trained using Supervised Learning. The training data consists of readings from all 260 BLMs around the MI enclosure, machine readings such as Main Injector and Recycler Ring beam intensities, machine state, Main Injector dipole bus ramp current, and clock events.

Beam Loss Monitor Location Recording

An assumption made at the beginning of the project was that any ML model created to attribute losses would be highly dependent on the placement and location of each BLM. While most BLMs are securely affixed to the machine, a good amount of BLM have been attached to moveable fixtures and experts from time to time have moved these

* Operated by Fermi Research Alliance, LLC under Contract No.De-AC02-07CH11359 with the United States Department of Energy. Additional funding provided by Grant Award No. LAB 20-2261 [1]
† Performed at Northwestern with support from the Departments of Computer Science and Electrical and Computer Engineering



(a) BLM floor heights. (b) BLM distance to the MI beampipe. (c) BLM density per 66.67 m.

Figure 2: Main Injector enclosure BLM recording.

BLM as they see fit to try and characterize problematic losses better. Before any data was collected, the location and orientation of each BLM in the Main Injector enclosure was recorded to allow for future administrative control and tracking. The records show that some BLM are much closer to either Main Injector or Recycler Ring and thus should be more sensitive to loss from that machine (Fig. 2a, 2b). Also, the BLMs are not uniformly ditributed about the enclosure, with 23 % of BLMs occupying a mere 10 % of the tunnel (Fig. 2c). The areas with the highest density of BLM tended to be areas of beam collimation, or injection and extraction regions.

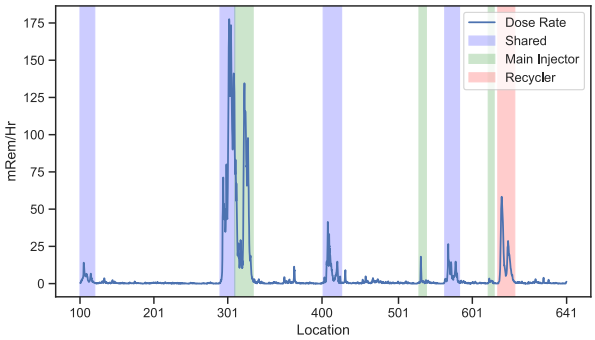


Figure 3: Location dependency of MI and RR beam loss as seen from tunnel residual dose rates.

Low Frequency (ACNET) Data

The first training data collected were relatively low frequency (LF) readings, either 15 Hz or 33 Hz, collected from the Fermilab accelerator controls network (ACNET) during machine operations [9]. The BLM nodes are capable of faster readouts, though for only a few BLM readings at a time due to their limited processing speed.

High Frequency (VME Reader Card) Data

In order to get the fastest possible readout of all the MI enclosure BLMs, custom cards were designed and built to listen to the BLM node VME crate backplane, intercept the crate digitizer readings and stream the high frequency (HF) readings, real-time over Ethernet (Fig. 8). The cards, known as VME Reader Cards or Pirate Cards, consist of an Intel Cyclone5 FPGA and an ARM Hard Processor System (HPS) [11]. The VME Reader Card FPGA decodes the backplane BLM signals, and also two front panel cabled signals for the Fermilab 10 MHz TeVatron Clock (TCLK) [12] and MDAT, a dedicated 720 Hz copper link that broadcasts select MI and RR machine readings as well as an epoch second timestamp (Fig. 4) [13]. The cards do not interfere with the operation of the BLM nodes that are still relied on daily for tuning and machine protection. Readings from all 7 of these cards began streaming reliably early 2023.

Labeling Data

Per machine, per BLM labels were generated using a multiprocessing script on each (15 Hz | 33 Hz | 320 Hz) sample. The result were [2, 260] labels where each row corresponded to a machine, and each column a BLM. Instances where the labeling logic determined only one machine was capable of generating loss (i.e. only that machine had beam in it at that time), that machines row of BLMs were given a value of 1.0, indicating that there was a 100 % probability that the beam loss originated from that machine. The machine(s) not capable of loss (and did not contain beam the previous n consecutive samples) are given a value of 0.0 for their row of BLMs. These samples are referred to as "known" samples and were used for training and validation of ML models. Samples where the labeling logic could not determine where the loss originated (i.e. both machines had beam in them), where given values of NaN for each column of BLM and each machine row. These samples are referred to as "unknown" samples and are the samples the operator and expert alike have trouble attributing to a machine and thus were used for testing and evaluation of ML models.

Beam Loss Studies

The Fermilab accelerator complex always strives to deliver the maximum beam intensity and power possible, with the least amount of beam loss and the highest transmission efficiency. This means that data collected from machine operations, baring failures or miss-tuning that the operators and experts work to prevent, tends to be very homogeneous. For days or weeks, the machines may run the same events, at the same beam intensity, with similar loss profiles. For this reason, studies were performed to sample losses that

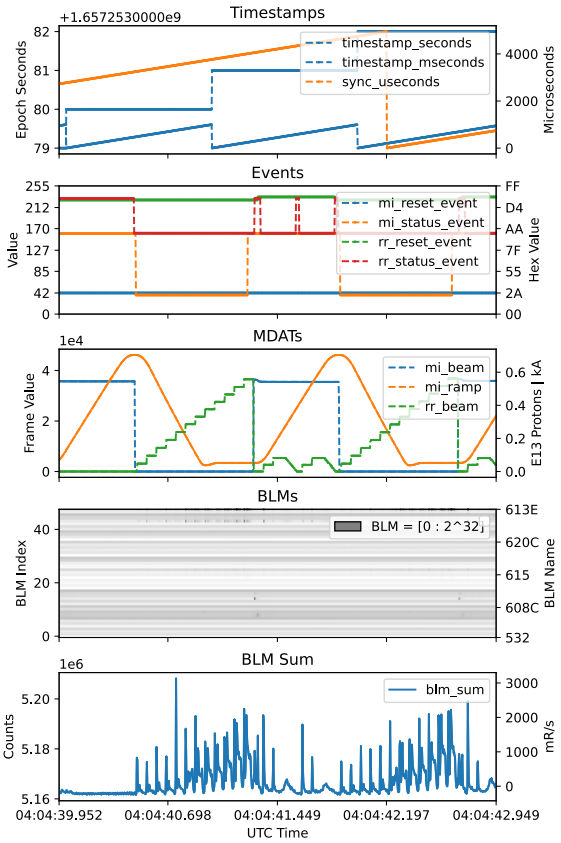


Figure 4: VME Reader Card data stream.

while entirely possible, are rare. During these studies, automated scripts purposefully created moderate localized and whole ring loss patterns through various miss-configurations and miss-tunings of the machine. Also, machine events and timelines were altered to remove occurrences of overlapping beam in Main Injector and Recycler, allowing for labeling of data from events that during normal operation could not be obtained. Two of these studies were performed, once in 2021 and again in 2022. Due to scheduling conflicts and less than desired machine availability in 2023, no beam loss study was performed and thus the high frequency dataset has none of these samples.

MODEL DEVELOPMENT

Over the course of the project, various ML model architectures were explored. Models are evaluated on prediction accuracy, how well they recognize state transitions (i.e. when beam exits or enters a machine), and during times when there is no truth, how close does the prediction resemble the machine experts best estimate.

Architecture Search

Models were initially trained using PyTorch [14] and the LF data collected from ACNET. Only known data samples were used for training and validation. The loss metric used for each models training was the Mean Squared Error (MSE), between the models predictions and the labels. Accuracy was defined to be the number of BLM predictions that were within 20 % of their label divided by the number of BLM in that sample. Accuracies for Main Injector and Recycler Ring were tracked separately so as to inform how the model performed for each machine.

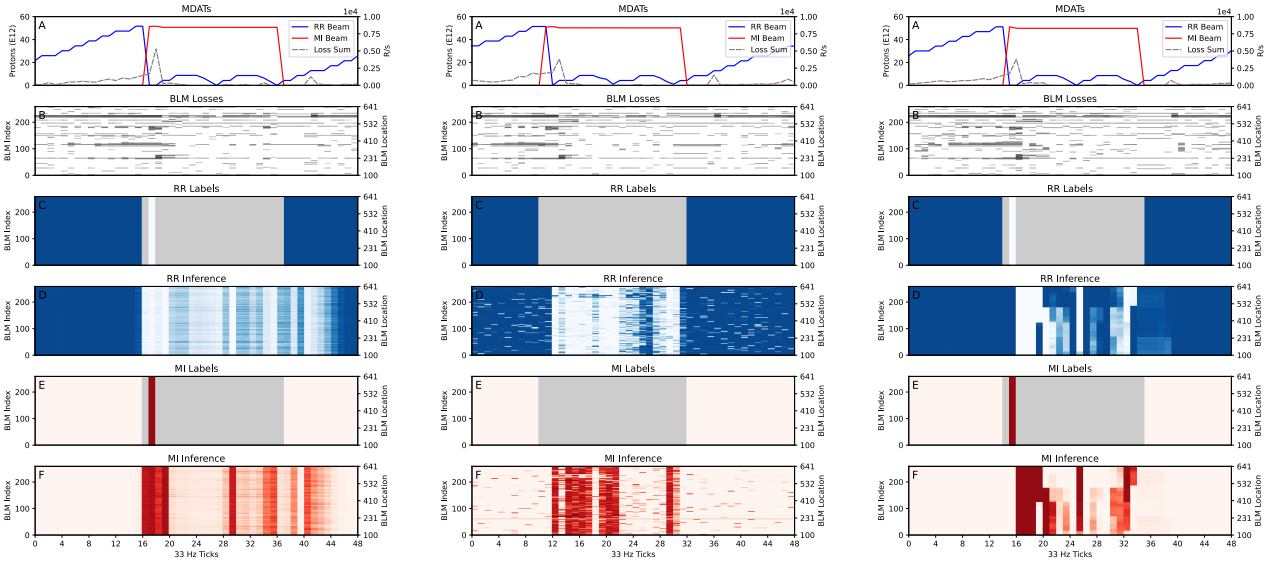
DBLN Model The first architecture investigated, the Deblending Model or DBLN, was a simple MLP model. It's input data consisted of all machine, clock and BLM readings. The model recognized some state transitions but showed limited loss pattern recognition during unknown samples (Fig. 5a).

Many Models Due to the geometry of the Main Injector tunnel, a beam loss at any one location is only recorded by a limited number of BLMs before the beam loss, projecting from the beamline at a tangent, is intercepted by the concrete enclosure wall. This fact inspired the creation of the Many Models architecture. Many Models consists of a separate MLP model for each BLM, ingesting readings for a limited window of its surrounding BLMs. The output of each model is aggregated and then AND with the output of a network dedicated to state of the machines. Many Models excelled at state transitions and showed much more local pattern recognition. However, the model was less accurate than desired on known samples and the large amount of individual networks made for a very large model, unlikely to fit within the limited resources of the FPGA it was to be deployed on (Fig. 5b).

UNet Model Though individual beam losses are not recorded by all BLMs, experts know that sometimes local loss patterns can often indicate a larger miss-tuning of the machines, and are actually part of a global or regional machine loss pattern. The UNet architecture, commonly used for medical imaging to label individual pixels or regions of pixels as abnormal tissue, was chosen as a next candidate, attempting to recognize both local and global beam loss patterns that the machines create [15]. After being trained on 9 M samples [15], the model had accuracies of 85 % and 94 % for Main Injector and Recycler Ring respectfully. The UNet architecture was the best performing of all the architectures explored. The model is fairly confident at state transitions and in regions of unknown loss attribution truth, the model attributes loss to likely locations of the machine that experts know to be were beam is purposefully collimated or has been identified to be a problem spot either from tuning or from residual dose rate surveys taken in the enclosure (Fig. 5c).

Omitting State Information During Training

During initial UNet training, both state and BLM data were used. However, with the possibility of the model picking up on global loss patterns, it was theorized that the model may not require state information. Further training was done omitting the state information; no appreciable degradation of



(a) DBLN (b) Many Models (c) UNet

Figure 5: Progression of ML model architectures.

the model's ability to recognize state transitions and boundaries was noticed. Omitting the state information not only allowed for a smaller model parameter size, but excluding this information may actually create a model more resilient to changes in mode of operation of the machines. Altering the clock events and machine states to accommodate a new beam request of the machines is a fairly common occurrence. One concern with building a model to disentangle losses was that it may be dependent on these states and thus need more frequent re-training. If one assumes that the loss patterns the model is recognizing is dominated by the geometry and imperfections in the build of the machine, these alterations to the machine are less frequent and thus may allow more time between retraining.

MODEL IMPLEMENTATION

In order to achieve true real-time inferences, the UNet ML model had to be implemented on an FPGA.

High Frequency Data UNet Model

The model architecture search utilized the LF ACNET data for training models. This was due to the long lead time to design, procure, populate, program and deploy the VME Reader Cards. Once the VME Reader Cards came online, the UNet architecture was used to train the model intended for deployment on the edge inference hardware. However, before training could begin, the UNet model architecture, originally defined in PyTorch, had to be translated to Keras [16]. The package intended to synthesize the trained model had much better support for Keras models and layers, namely convolutional layers, than for PyTorch. Translation of the model between packages was less trivial than originally thought, some layers did not have one-for-one counterparts. Once satisfied that the model had been translated, training was done using millions of random standardized BLM samples from approximately 6 months of machine operations.

Figure 7 shows the offline performance of the HF UNet model. Figure 7a shows the beam intensities in both Main Injector and Recycler Ring as well as the per sample (frequency tick) beam loss sums. Figure 7b gives the values of the individual BLMs around the machine at each sample. Figures 7c and 7e give the labels applied to those samples, with Red and Blue representing Recycler Ring and Main Injectors BLM probabilities of 1.0 respectfully. White regions represent BLM probabilities of 0.0 and Grey regions are where the labeling logic is unable to determine beam loss machine origin. Figures 7d and 7f show the models inferences as to the probability at each sample and each BLM where the loss originated.

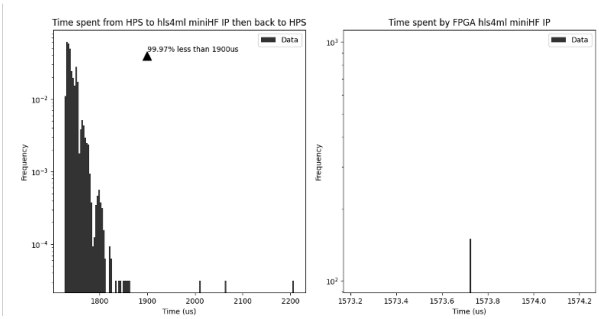


Figure 6: Aria10 FPGA inference latencies.

The model does does fairly well at recognizing state transitions confidently, especially when machine losses are elevated. For samples when the loss origin is unknown, the areas indicated by Grey, experts believe the model is doing well predicting the origin of loss as the time of the predictions agree with events in the cycle that would cause loss then and in the machines predicted. The model appears to be

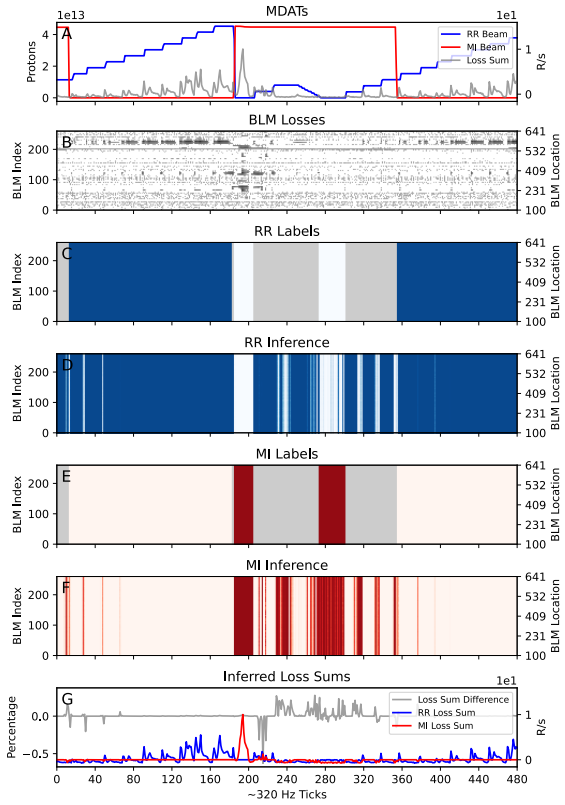


Figure 7: Offline HF UNet model inferences. The model shows great global beam loss pattern and state transition recognition but performs poorly on more localized losses.

preferring to apply probabilities uniformly to one machine or the other at any given sample, though not entirely. Experts know that at times both machines can be creating loss simultaneously so one would expect a more mixed prediction of loss attribution at limited times during the machine cycle. However, when the probabilities are multiplied by the BLM loss sums at each tick, the machine loss sum attributions look very believable and match the experts expectations, especially for the most difficult times to attribute loss where some fraction of the loss sum is applied to both machines (Fig. 7g). The disentangled loss sums consistently account for >= 99.5 % of the total loss sum, assuring that most beam loss is accounted for (Fig. 7g).

Work is still underway to improve the HF UNet models performance and attempt to regain the mixed localized probability predictions that experts believe should be more prevalent and were seen from the LF UNet model. Some avenues of approach are to complete a beam loss study and collect HF data to train on, curate the HF dataset better to include more unique samples during training, transferring the weights from the LF UNet model and weighing training loss higher for samples with rare beam loss profiles. With it's deficiencies known, it was felt that the model was accurate enough to implement on the edge hardware.

Synthesis

Synthesis of the trained HF UNet model was done using hls4ml, a package that converts ML models trained via various popular opensource ML platforms, into High Level Synthesis (HLS) code that can be integrated into a FPGAs firmware [17].

Layer Precision Tuning

To ensure the HF UNet model fit within the limited resources of the Central Node's Aria10 FPGA, post training, per layer bit precision tuning was performed. Each layers number of floating point bits were minimized, decreasing the models resource foot print and ensuring the models inference took as little time as possible to complete while also ensuring the firmware's output matched that of the original Keras model for the same input (Fig. 9). The resulting model fit comfortably on chip and had an average latency of approximately 1.7 mS, well within or specification of less than 3.125 mS (320 Hz) between inferences (Fig. 6).

In the future, quantization aware training of the HF UNet model will be done to hopefully avoid the need to do the per layer precision tuning which is very labor intensive.

Central Node Deployment

The synthesized HF UNet model was implemented on an Aria10 FPGA System On Module (SOM) that also includes dual ARM HPS cores [18]. The SOM lays upon a custom carrier board designed for both READS sub-projects, that provides power and IO (Fig. 8), though for the beam loss disentangling sub-project, much of the ADC available on the board is unused and Ethernet is the main data path.

The first ARM HPS is responsible for initiating data streams from the 7 VME Reader Cards. The HPS then ingests the streams, aligning all 7 streams to common samples using stream headers that contain an epoch timestamp second, a 1.3 mS resolution fractional second, and a microsecond counter each card increments and resets on a common TCLK event. Once a full sample is assembled, the HPS passes the sample to the HF UNet model implemented on the FPGA. The FPGA, upon inference completion passes the inference back to the HPS where it is packaged with the original sample data the inference was made on (Fig. 8). The packaged data is then streamed from the Central Node to a Redis server (Fig. 8).

Accelerator Controls System (ACNET) Integration

One of the primary goals of this project was to provide real-time predictions as to beam loss origin to the operators in the Main Control Room (MCR) and have those readings readable and plot-able from the same tools they use to tune and monitor the machines everyday (Fig. 10).

Redis Redis is a lightweight message brokering framework that enables streaming of data to multiple clients [19]. Redis provided a very flexible way to essentially create an Ethernet fieldbus serving Central Node readings (Fig. 8).

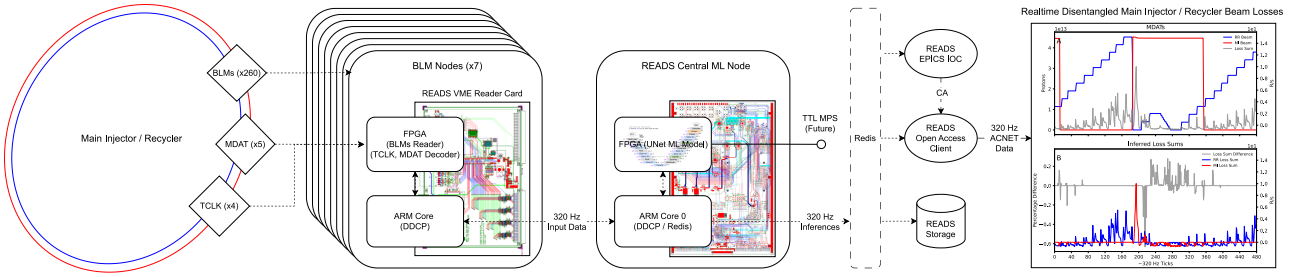


Figure 8: Complete beam loss disentangling network.

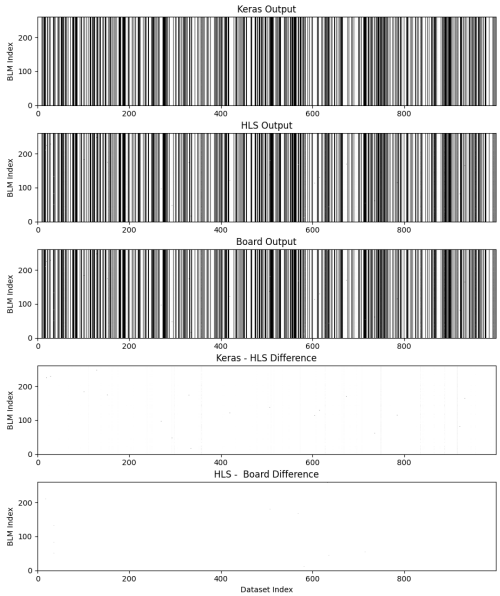


Figure 9: Differences between Keras, HLS, and FPGA output.

Open Access Client (OAC) Open Access Clients (OAC) are ACNET Java virtual Front Ends (FE) that are commonly used to translate Ethernet instrumentation to ACNET. For this project, there was need to translate and re-package the readings coming from the Central Node via Redis for use in ACNET. An entirely new OAC was written that consumed the Central Node Redis streams and translated those structs into 520+ individual ACNET devices, one for each BLM per machine, plus additional array, state, and machine reading devices. Fast Time Plot (FTP) data buffers, a legacy plot data protocol in ACNET where timestamps are 100 uS bins relative to a periodic TCLK event, were created for each device. (Fig. 8)

EPICS IOC The Fermilab accelerator complex has been controlled and monitored almost exclusively since the 1980's with ACNET. However, the PIP-II linear accelerator [20, 21] currently being built at Fermilab has decided that it will rely on EPICS for the bulk of it's controls [22]. With EPICS increasing in use at Fermilab, it was decided to attempt to create an IOC in parallel to our OAC as a proof of concept serving remote real-time inferences via an IOC at Fermilab. The project successfully created a basic Redis driver for EPICS and served up Channel Access (CA) variables for anyone to consume. As another proof of concept, the CA variables were consumed by the project's OAC and used to populate ACNET inference readings, demonstrating another viable path to getting inference readings to the MCR (Fig. 8).

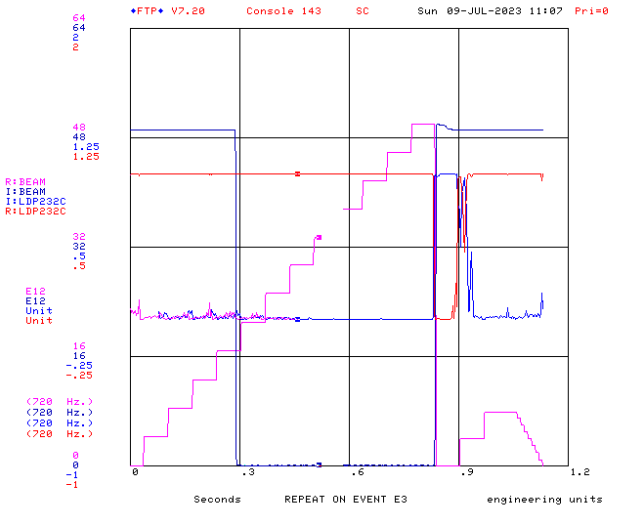


Figure 10: First live, 320 Hz beam loss disentangling inferences read through the ACNET controls system. Readings for the Main Injector and Recycler beam intensities are shown alongside the MI and RR loss attribution inferences for one BLM location at 232C.

PRELIMINARY TESTING

The Fermilab 2022-2023 accelerator run ended the second week of July 2023. The complete beam loss disentangling network was brought online just a few days before beam was turned off for that years summer maintenance period. In those last few days of beam, many experiments had already turned off, or in the case of the g-2 experiment, completed their experiment, testing out the beam loss disentangling network was difficult. However, a few small scale tests were performed to evaluate the online models performance. The inferences read out from ACNET as well as Redis streams saved to disk showed the expected output. More testing and evaluation of the system is planned for when beam returns.

As seen in Figure 10, per BLM inferences are available to the operators using their existing FTP plotting client. This allows the operators to plot any other accelerator instruments and correlate them with the disentangled beam loss inferences.

SUMMARY

The READS project has designed, trained, synthesized, implemented and deployed a real-time edge AI/ML model on the Fermilab accelerator controls network (ACNET); a first of its kind for the Fermilab accelerator complex.

Custom VME Reader Cards were built to enable streaming 320 Hz BLM readings from legacy beam loss monitoring systems without interfering with their functionality.

The ML model, a UNet type architecture trained to high accuracy, disentangles mixed beam losses during times of simultaneous high intensity beams in Main Injector and Recycler Ring and attributes probabilities per BLM as to which machine the loss originated from.

A Custom Central Node was created to implement the synthesized and layer precision tuned ML model on an FPGA and output real-time inferences. These inferences are streamed via Redis for any number of clients to consume.

An Open Access Client translates the Redis inference stream to 520+ individual ACNET devices which may be read or plotted within the Fermilab Main Control Room using the same tools operators use everyday to tune, monitor and diagnose failures on the accelerators.

REFERENCES

[1] Department of Energy, Office of Science. "Data, Artificial Intelligence, and Machine Learning at DOE Scientific User Facilities, DOE National Laboratory Program Announcement Number: LAB 20-2261." (2020), https://science.osti.gov/-/media/grants/pdf/lab-announcements/2020/LAB_20-2261.pdf

[2] A. Narayanan *et al.*, "Optimizing Mu2e Spill Regulation System Algorithms," in *Proc. IPAC'21*, Campinas, Brazil, May 2021, pp. 4281–4284. doi:10.18429/JACoW-IPAC2021-THPAB243

[3] A. Narayanan *et al.*, "Machine Learning for Slow Spill Regulation in the Fermilab Delivery Ring for Mu2e," presented at NAPAC'22, Albuquerque, New Mexico, USA, Aug. 2022, paper MOPA28, unpublished, 2022. https://napac2022.vrws.de/papers/mopa75.pdf

[4] V. Nagaslaev *et al.* "Third Integer Resonance Slow Extraction Using RFKO at High Space Charge." (2011), https://www.osti.gov/biblio/1031169

[5] M. Ibrahim *et al.*, "Preliminary Design of Mu2E Spill Regulation System (SRS)," in *Proc. IBIC'19*, Malmö, Sweden, 2019, pp. 177–180. doi:10.18429/JACoW-IBIC2019-MOPP033

[6] L. Bartoszek *et al.*, "Mu2e Technical Design Report," 2014. doi:10.2172/1172555

[7] K. Seiya *et al.*, *Accelerator real-time edge ai for distributed systems (reads) proposal*, 2021. doi:10.48550/ARXIV.2103.03928

[8] R. Ainsworth *et al.*, "High Intensity Proton Stacking at Fermilab: 700 kW Running," in *61st ICFA Advanced Beam Dynamics Workshop on High-Intensity and High-Brightness Hadron Beams*, 2018, TUA1WD04. doi:10.18429/JACoW-HB2018-TUA1WD04

[9] K. Hazelwood *et al.*, "Real-Time Edge AI for Distributed Systems (READS): Progress on Beam Loss De-Blending for the Fermilab Main Injector and Recycler," in *Proc. IPAC'21*, Campinas, SP, Brazil, 2021, paper MOPAB288, pp. 912–915. doi:10.18429/JACoW-IPAC2021-MOPAB288

[10] N. Chelidze *et al.*, "Residual Dose and Environmental Monitoring for the Fermilab Main Injector Tunnel Using the Data Acquisition Logging Engine (Dale)," presented at NAPAC'22, Albuquerque, New Mexico, USA, Aug. 2022, paper MOPA18, unpublished, 2022. https://napac2022.vrws.de/papers/mopa15.pdf

[11] J. Berlioz *et al.*, "Synchronous High-Frequency Distributed Readout for Edge Processing at the Fermilab Main Injector and Recycler," presented at NAPAC'22, Albuquerque, New Mexico, USA, Aug. 2022, paper MOPA15, unpublished, 2022. https://napac2022.vrws.de/papers/mopa15.pdf

[12] G. Vogel. "TCLK Event Definitions." (2021), https://www-bd.fnal.gov/controls/hardware_vogel/tclk.htm

[13] G. Vogel. "MDAT Frame Definitions." (2019), https://www-bd.fnal.gov/controls/hardware_vogel/mdat.htm

[14] "PyTorch." (2023), https://pytorch.org

[15] M. Thieme *et al.*, "Semantic Regression for Disentangling Beam Losses in the Fermilab Main Injector and Recycler," presented at NAPAC'22, Albuquerque, New Mexico, USA, Aug. 2022, paper MOPA28, unpublished, 2022. https://napac2022.vrws.de/papers/mopa28.pdf

[16] "Keras: Deep Learning for Humans." (2023), https://keras.io

[17] "Fast Machine Learning Lab: hls4ml Documentation." (2023), https://fastmachinelearning.org/hls4ml/

[18] M. Ibrahim *et al.*, "FPGA Architectures for Distributed ML Systems for Real-Time Beam Loss De-blending," presented at IBIC'23, Saskatoon, Canada, Sep. 2023, paper TU3C02, unpublished, 2023.

[19] "Redis: Remote Dictionary Server." (2023), https://redis.io

[20] "PIPII final design report," Tech. Rep., 2018. https://pip2-docdb.fnal.gov/cgi-bin/private/RetrieveFile?docid=5310

[21] M. Ball *et al.*, "The PIP-II conceptual design report," Tech. Rep. FERMILAB-DESIGN-2017-01; FERMILAB-TM-2649-AD-APC 1516858, 2017. doi:10.2172/1346823

[22] D. Nicklaus, "Controls at the Fermilab PIP-II Superconducting Linac," presented at this conference ICALEPCS'23, Cape Town, South Africa, Oct. 2023, paper FR1BCO02, unpublished, 2023.