REAL-TIME EDGE AI FOR DISTRIBUTED SYSTEMS (READS): PROGRESS ON BEAM LOSS DE-BLENDING FOR THE FERMILAB MAIN INJECTOR AND RECYCLER

K.J. Hazelwood*, M.R. Austin, M.A. Ibrahim, V.P. Nagaslaev, D.J. Nicklaus, A.L. Saewert, B.A. Schupbach*, K. Seiya, R.M. Thurman-Keup, N.V. Tran, A. Narayanan†, Fermi National Accelerator Laboratory‡, Batavia, IL USA
H. Liu, S. Memik, R. Shi*, M. Thieme*, Northwestern University‡, Evanston, IL USA
1 also at Northern Illinois University, DeKalb, IL USA

Abstract
The Fermilab Main Injector enclosure houses two accelerators, the Main Injector and Recycler. During normal operation, high intensity proton beams exist simultaneously in both. The two accelerators share the same beam loss monitors (BLM) and monitoring system. Beam losses in the Main Injector enclosure are monitored for tuning the accelerators and machine protection. Losses are currently attributed to a specific machine based on timing. However, this method alone is insufficient and often inaccurate, resulting in more difficult machine tuning and unnecessary machine downtime. Machine experts can often distinguish the correct source of beam loss. This suggests a machine learning (ML) model may be producible to help de-blend losses between machines. Work is underway as part of the Fermilab Real-time Edge AI for Distributed Systems Project (READS) to develop a ML empowered system that collects streamed BLM data and additional machine readings to infer in real-time, which machine generated beam loss.

READS OVERVIEW
The Real-time Edge AI for Distributed Systems (READS) project is a collaboration between the Fermilab Accelerator Division and Northwestern University. The project has two objectives; first to implement Machine Learning (ML) into the future Delivery Ring slow spill regulation system [2] for the Mu2e experiment [3, 4], and second to create a real-time beam loss de-blending system for the Main Injector (MI) accelerator enclosure also utilizing ML [5].

Beam Loss De-blending
The Main Injector and Recycler Ring (RR) accelerators share a tunnel and one beam loss monitor (BLM) system. The 8 GeV permanent magnet Recycler was originally built as an anti-proton storage ring for the Tevatron collider [6]. Anti-proton losses in Recycler were insignificant compared to the 8 GeV to 120 GeV proton losses from Main Injector; there was less need to monitor ionization beam losses from Recycler. When the Tevatron was decommissioned, Recycler was re-purposed as a proton stacker for Main Injector 120 GeV NuMI beam operation [7] as well as for 8 GeV Muon g-2 experiment beam delivery [8]. Currently, normal operation of the accelerator complex has high intensity beams in both Main Injector and Recycler simultaneously. Beam losses from both machines are now a large concern. The origin of loss on any of the 259 operational BLMs can be difficult to attribute to any one machine. However, experts can often attribute losses to either Main Injector or Recycler based on time (Fig.1), machine state, and location (Fig.2).

Using streamed distributed readings and real-time ML inference hardware, this project aims to replicate and improve upon the machine expert’s ability to de-blend losses between machines.

PIRATE CARD DEVELOPMENT
In order to satisfy the data requirements for this project, a parasitic VME bus reader card, commonly referred to as a Pirate Card, is being developed and integrated into the existing MI BLM system. Each of the 7 BLM nodes distributed around the 2.2 mile Main Injector complex consists of a VME Crate Processor, Control Card, Timing Card, and an array of digitizers [9]. The sole responsibility of the Pirate Cards is to intercept the BLM values of each digitizer throughout the beam cycle without disturbing normal operations of the system. The digitized BLM values will be
MODELL ARCHITECTURE

The De-Blending Network (DBLN) is comprised of three parts: a BLM network, a State Network, and an Aggregation Network (Fig.3). At each point in time, the DBLN maps observations of the last n BLM loss signatures \( l_n \in \mathbb{R}^{n \times 259} \) and machine data \( e_n \in \mathbb{R}^{n \times 9} \) to class-conditional probabilities over individual accelerators \( p(BLM) \). Note here the overloading of the term "loss": when referring to the BLM losses we use \( l \), and when referring to the mathematical quantity related to the ML model performance we use \( \ell \).

The \( BLM \) Network is a convolutional neural network (CNN) with two max-pooled convolutional layers followed by two linear layers. The BLM network learns a mapping between the raw BLM loss data \( l_n \in \mathbb{R}^{n \times 259} \rightarrow B_n \in \mathbb{R}^9 \). This vector \( B_n \) is then ingested by the Aggregator where it is conditioned on a representation of the machine state generated by the State Network.

The \( State \) Network is a two layer fully-connected multi-layer perceptron (MLP) that learns a mapping between the last \( n \) state observations \( e_n \in \mathbb{R}^{n \times 9} \rightarrow S_n \in \mathbb{R}^9 \). The output \( S_n \) serves as a conditioning mechanism for the representation of the BLM loss signature \( B_n \).

The \( Aggregator \) Network is a three layer fully-connected MLP that learns to map \( B_n \oplus S_n \in \mathbb{R}^9 \), where \( \oplus \) is the elementwise sum, to class-conditional probabilities over \( p(a_i | l_n, e_n) \in \mathbb{R}^{259 \times 2} \). We choose the elementwise sum instead of concatenation to make the model more compact.

To train the model, we use Binary Cross-Entropy Loss and the Adam Optimizer with learning rate \( = 0.001 \). The final model has 1.3M trainable parameters.

![DBLN model architecture](image)

PRELIMINARY RESULTS

Initial results using the Sample Dataset show promising performance. Figure 4 details training accuracy and loss over the first 1000 batches using the past \( n = 2 \) observations and batch size \( = 32 \).

Figure 5 (A) shows the measured beam intensities in MI and Recycler over 24, 15 Hz ticks. Section (B) shows the normalized BLM measurements at each tick. Sections (C) and (D) show the output of our model scaled by the BLM loss intensity (significance). From these plots, we can see that our model is appropriately classifying the losses in each region. As beam is extracted from Recycler to MI, our model recognizes the change in the loss signature and switches the inferred label from RR to MI in turn.
Overall, high validation accuracies (95% and 96% for MI and RR respectively) is evidence that our model is learning meaningful mappings between BLM loss profiles and their machine of origin.

Model Confidence

Of particular interest is the model’s behavior on the BLM losses which cannot be attributed to a single machine, i.e. loss profiles acquired when MI and RR operate simultaneously. Each row in the output \( p(a_i|l_n, s_n) \in \mathbb{R}^{259 \times 2} \) corresponds to \( p(\text{MI}), p(\text{RR}) \). Probabilities \( \geq 0.5 \) are treated as positive identifications and \( < 0.5 \) as negative identifications. Performing inference on these data with unknown labels yields the confusion matrix in Table 1, where MI/RR (+) and MI/RR (-) represent positive and negative identifications, respectively.

### Table 1: Model Confusion Matrix

<table>
<thead>
<tr>
<th></th>
<th>MI (+)</th>
<th>MI (-)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RR (+)</td>
<td>1%</td>
<td>2%</td>
</tr>
<tr>
<td>RR (-)</td>
<td>77%</td>
<td>19%</td>
</tr>
</tbody>
</table>

Presently, the model is disproportionately recognizing these unknown losses as originating from MI and not RR. Experimentation is underway to better understand the model’s behavior on these data with unknown labels.

**MODEL IMPLEMENTATION**

The ML model will be implemented as an IP core on the Intel Arria-10 SOC. The board contains a FPGA side and a hard processor subsystem (HPS) side which have fast communication through HPS-FPGA bridges between them. The FPGA side can be used to implement the DBLN network for processing the data. The HPS side has Ethernet and can do complicated calculations. The HPS will also be useful for updating the implemented DBLN network by partial reconfiguration.

The hls4ml+Quartus tool flow will be used to generate an initial design for the ML IP. Based on the model development described in previous sections, the saved model will act as an input to hls4ml and will generate an implementation in C++ which uses HLS Compiler for hardware design. The Quartus backend of hls4ml will be used as a starting point with additional customization done later. Finally, The DBLN network will be implemented as an IP core and connected to HPS using the Platform designer.

During the design phase of the IP, there must be trade-offs between the expected latency and the limited resources. Various methods can be adopted to optimize the implementation by exploiting pipeline and parallelism. For example, unrolling the potential loops, modifying the initial interval and adding registers between each layer can all be used to help achieve the desired data processing pipeline. Multiple parallel kernels will be used, this will likely translate into parallel data paths. To fully utilize the limited resources on the FPGA, with respect to constraints imposed by other functionality that is co-located on it, we need to carefully select a proper precision of the data representation and consider reuse of the implemented kernel and BRAM buffers.

**SUMMARY**

The READS Main Injector accelerator enclosure beam loss de-blending project is progressing well. A Sample Dataset has been generated using MI/RR readings and a very promising preliminary ML model has been created from the data. To meet the project’s data needs, a custom BLM node VME bus reader card, commonly referred to as a Pirate Card, has been designed and is being manufactured. Delivery of the Pirate Cards is expected late spring 2021, just in time to collect high fidelity training data before the planned Fermilab accelerator complex summer maintenance shutdown. Hardware for the final FPGA ML model implementation has been acquired and is being developed on.
REFERENCES


