Uncertainty aware anomaly detection to predict errant beam pulses in the SNS accelerator

Malachi Schram
On behalf of the Errant Beam Task Group
Data Science Department
Thomas Jefferson National Accelerator Facility

Machine Learning Grant from BES

Collaboration between SNS, ORNL, and Jefferson Lab.

1) Develop ML capabilities for time series analysis and prediction:

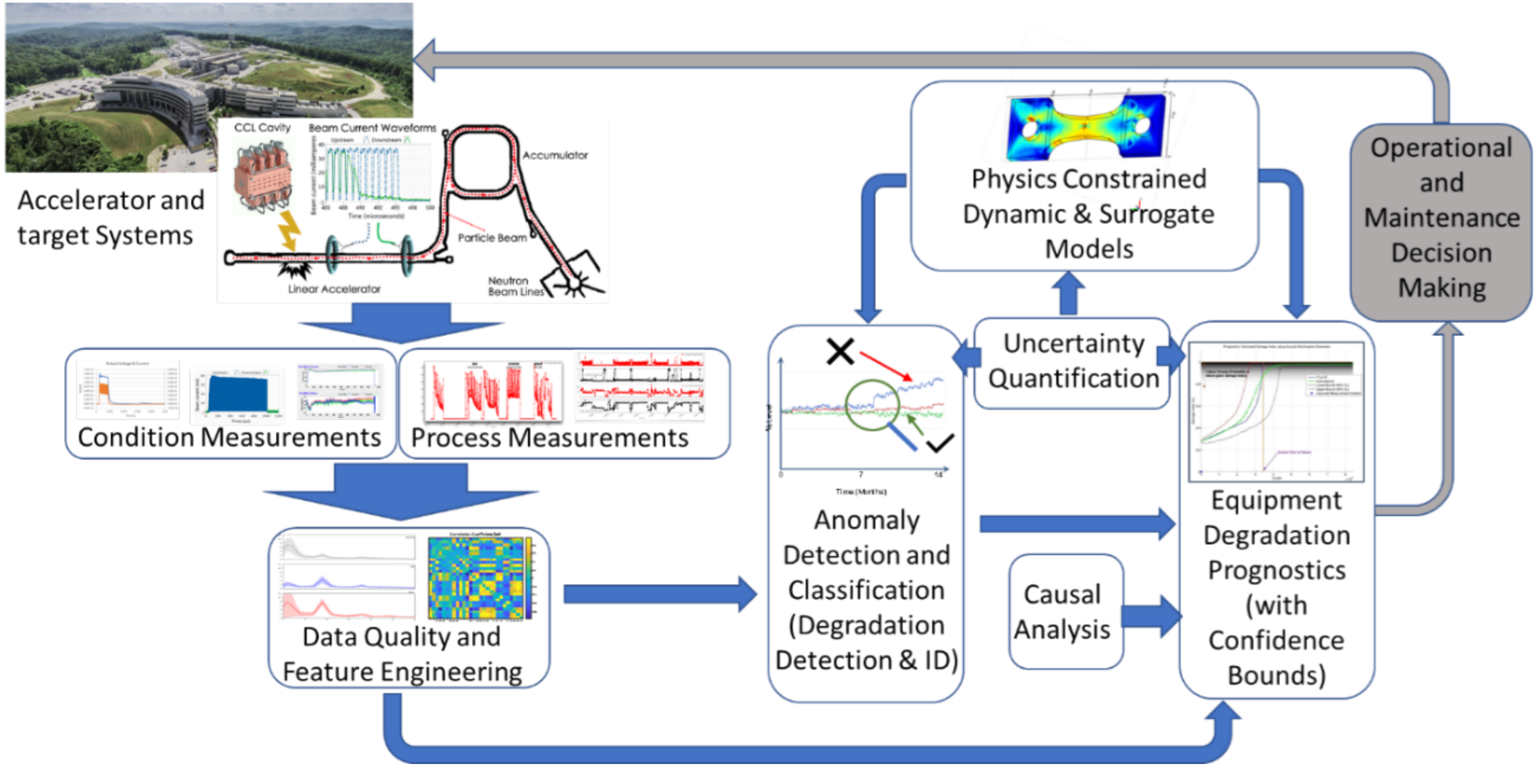
   a) Use unsupervised and semi-supervised learning with existing data to develop ML models with multiple timescales to predict failure scenarios.
   b) Develop ML-based anomaly detection capabilities for accelerator operations.
   c) Develop Uncertainty Quantification (UQ) capabilities for robust and reliable time-series analysis.
   d) Develop causal analysis capabilities relating failure predictions and anomalies to sensor measurements and system operating parameters.

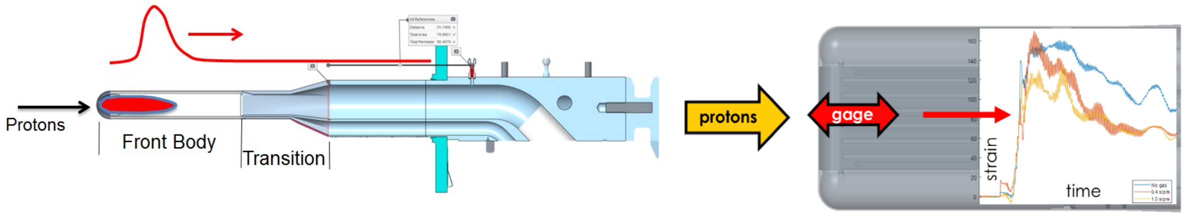
2) Demonstrate Objective (1) on accelerator and target systems to monitor condition, detect anomalies, and predict failure, using information from system instrumentation and beam-based signatures.

3) Demonstrate ML-based surrogate modeling to optimize parameters and inform design choices.

Four use-cases

1. HVCM
   Predict upcoming failure and parameter optimization
2. Cryogenic Moderator System
   Improve loop control
3. Target
   Surrogate modeling to improve simulation for design and failure prediction using strain data (HPC)
4. Beam-based
   Predict errant beam using beam Instrumentation: DCM/BPM/MPS





OAK RIDGE National Laboratory | SPALLATION NEUTRON SOURCE
Jefferson Lab

Data science pipeline for Errant Beam anomaly studies

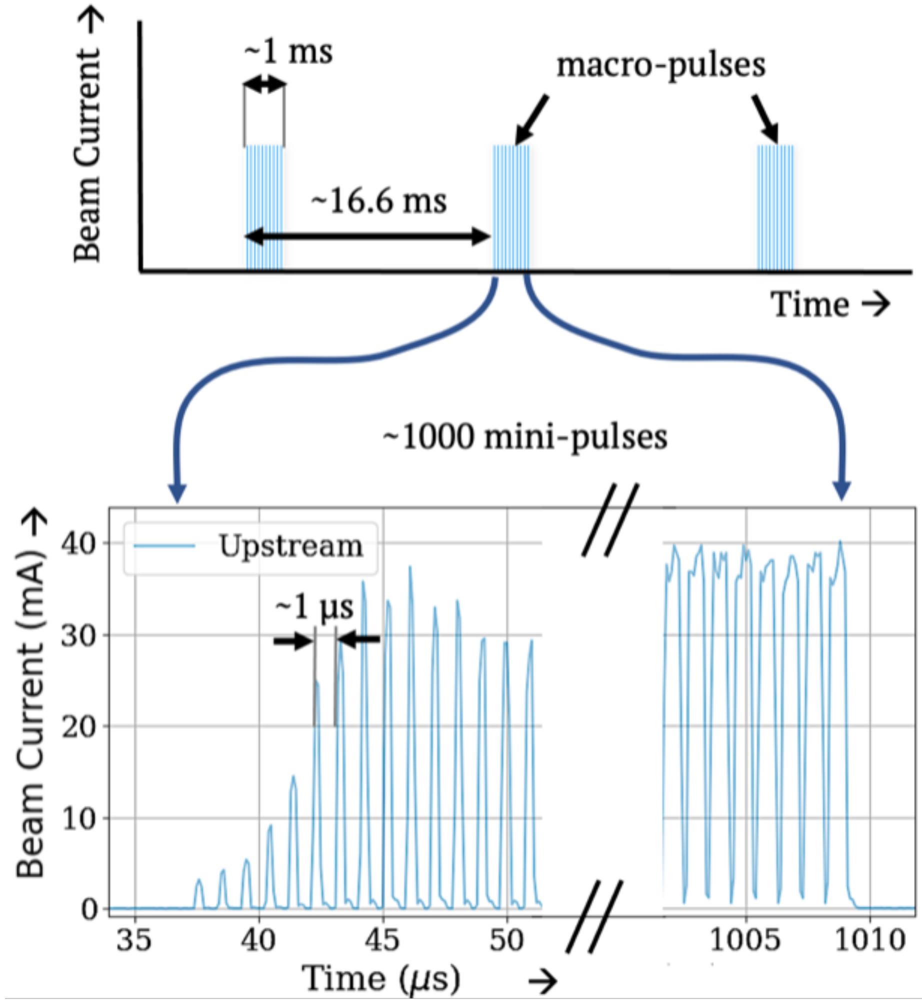
- **Goal**: Predict an upcoming machine trip at least one macro-pulse before it happens.
- **How:** We use pulses leading to a trip (tagged "Before") and to identify features that indicate an upcoming failure
- **Data science pipeline used**:

Data Source	Data Preparation	ML Applications	Training Tools	Results
• Real or synthetic • Quality • Dimensionality • Format • Density • Size	• Data cleaning • Data restructuring • Correlations • Dynamics • Visualization	• Classification • Regression • Clustering • Feature extraction	• Cross-validation • HPO	• Predictions • Confidence Level • Explainability

Stage 1/2 - Data Source and Preparation: Overview

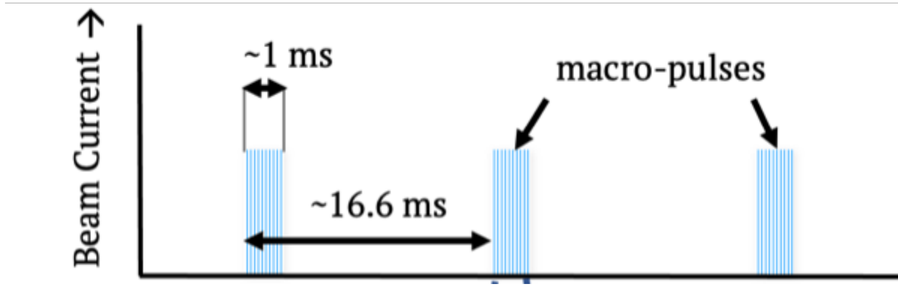
- **How was the data collected and labeled?**
  - Data acquired during March 2021
    - The accelerator creates a series of pulses ("macro-pulses") with each macro-pulse composed of ~1k mini-pulses
    - An errant-beam data file is composed of 25 "good" macro-pulses followed by the errant beam pulse
    - A "normal" data file has no errant beam pulse
  - We used the macro-pulse before the errant beam pulse and macro-pulses from the normal file for our studies
    - Our hypothesis is that there is a sign that something wrong is going to happen in the previous macro-pulse
    - We also need to forecast the fault in enough time to be actionable
  - We are using time samples between 3,500-13,500 (10k samples) for the primary analysis presented in these slides
  - Samples were divided into 3 orthogonal dataset:
    - Train/validation(80%)/test(20%)

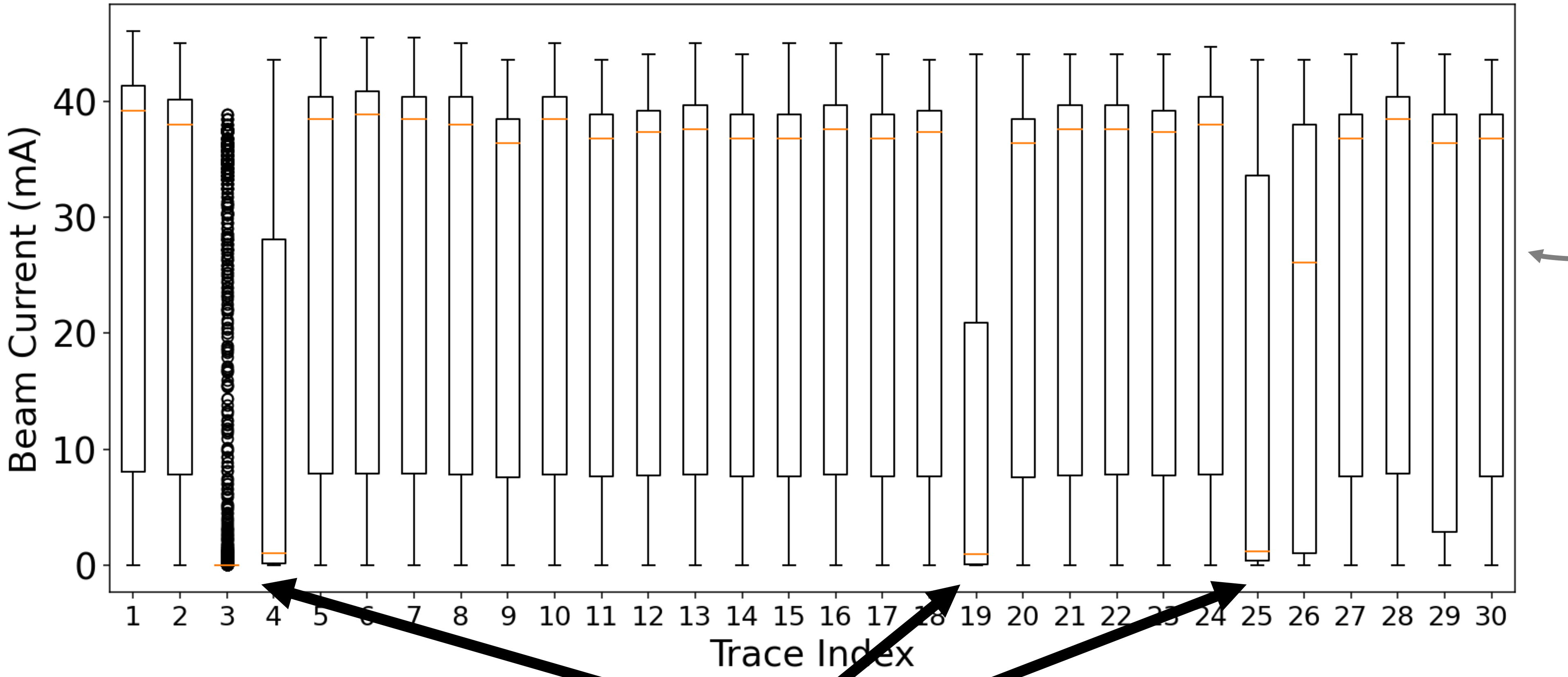
      – Train/validation (80%/20%)



Stage 1/2 - Data Source and Preparation: Macro-stats

- Explore the macro stats for all macro-pulses
- Example: random selection of 30 micro-pulse without any requirements on the number of peaks

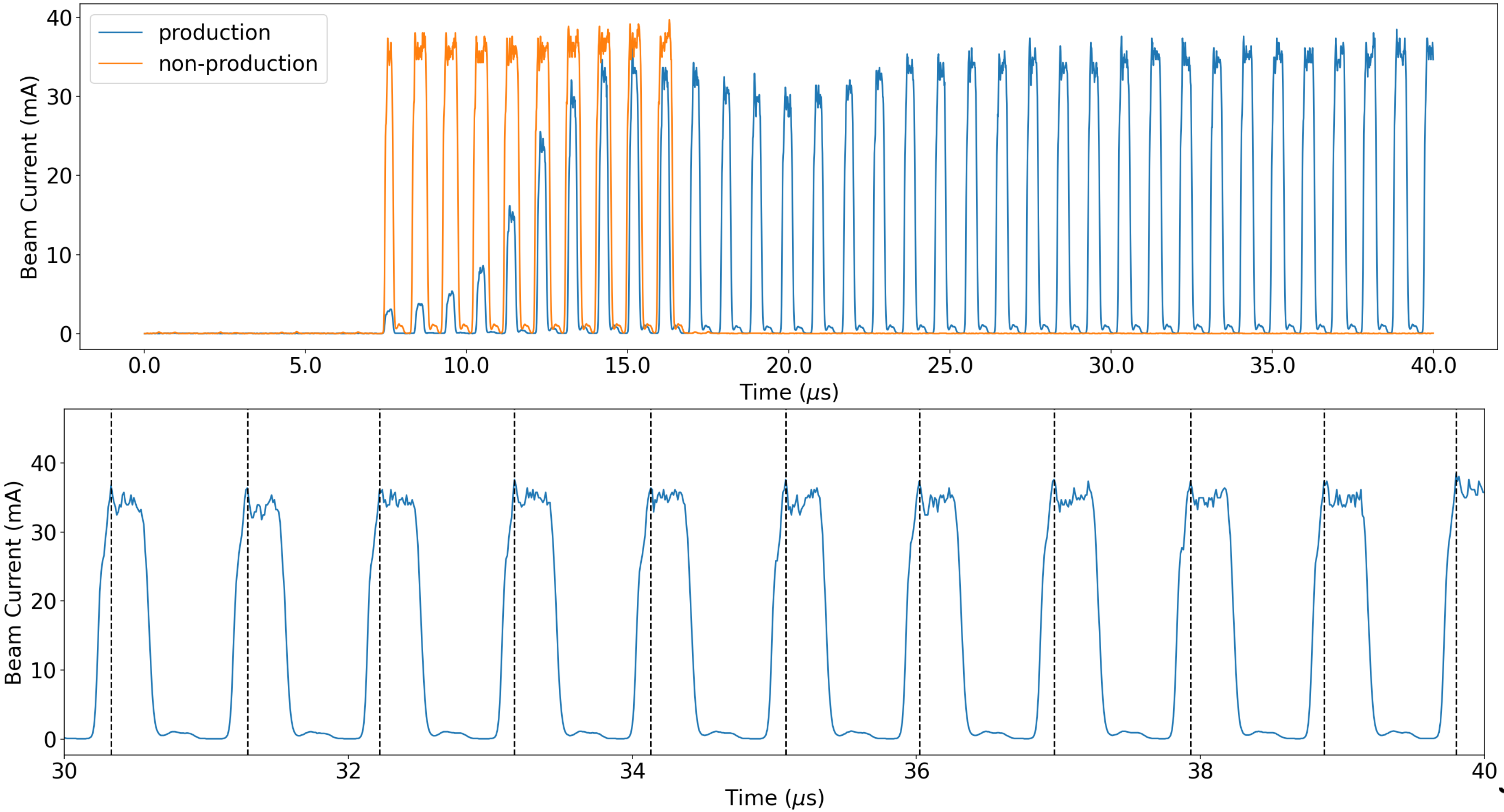




Found some statistical variations in the "Before" pulses

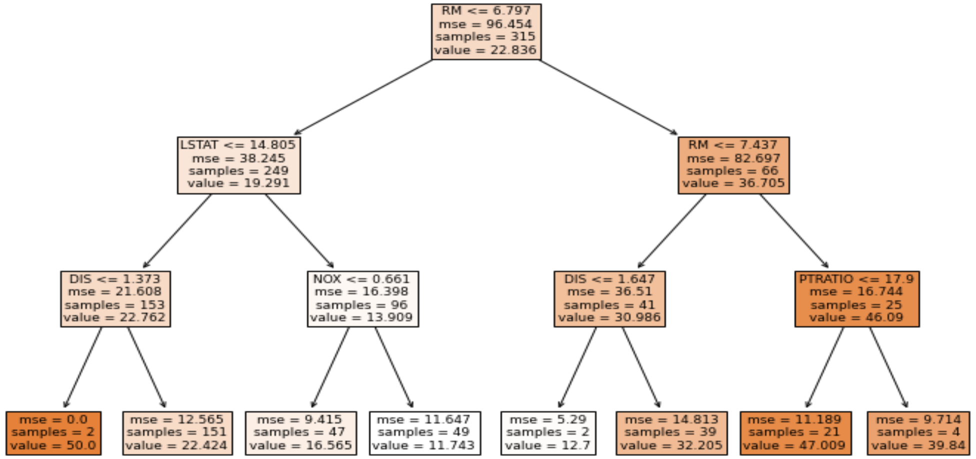
Stage 1/2 - Data Source and Preparation: Peak Finding

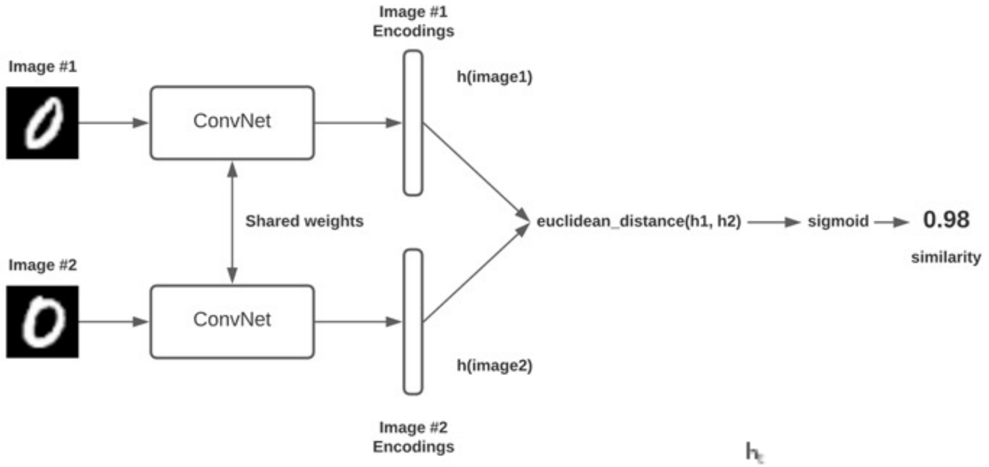
- Identified traces with statistical variations in our initial training sample (non-production beam)
- We applied a peak find algorithm to ensure the correct number of peaks
- Removed this data from our data-set

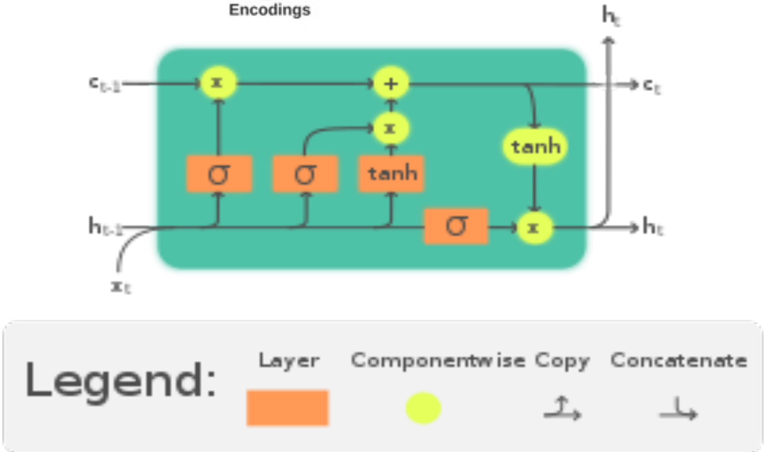


Stage 3 - Machine Learning

- Random Forest
  - Predict errant beam
  - Supervised
- Siamese twin
  - Predict errant beam
  - Monitor any drifts in the normal pulses
  - Maybe add failing equipment later
- RNN/LSTM Auto-encoder
  - Predict failing equipment
  - Use reconstruction error to identify anomalies
- Integration with ORNL FPGA Group
  - Implement RF model on FPGA







Stage 3 - Machine Learning Random Forest Work [1]

Work independent of grant

Method

- Random Forest classifier with 100 estimators
- Improvements: PCA, FFT, Voting, different dataset sizes

Metrics

- Train/test dataset separation, 5-fold exhaustive cross-validation during training
- True Positive / False positive = How many trips can we find without raising too many false alarms
- Classification speed (needs to fit within 16 ms)

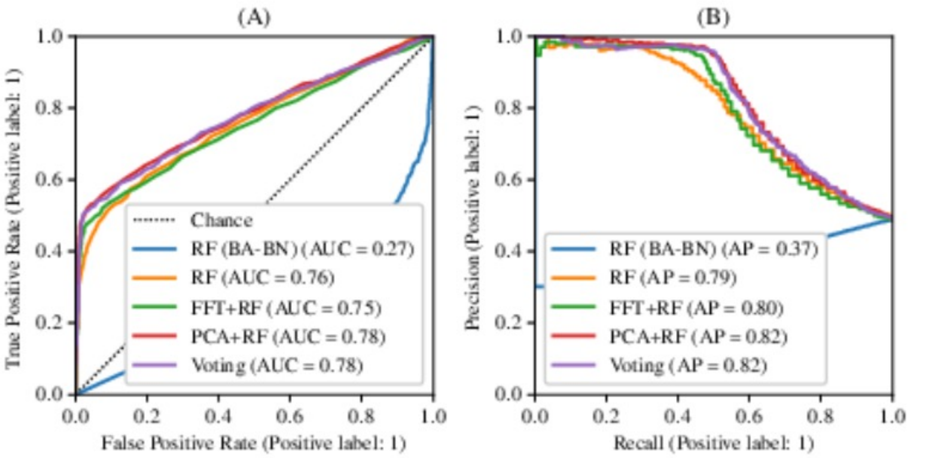


Table 6: List of label thresholds and the effect on T_p and F_p for different classifiers and dataset sizes

Classifier	Th_t	T_p	F_p	Th_e	T_p	F_p
RF	0.94	0.06	0.00165	0.73	0.31	0.0099
FFT + RF	0.93	0.11	0.00165	0.79	0.35	0.0099
PCA + RF	0.95	0.18	0.00165	0.73	0.42	0.0099
Voting	0.92	0.10	0.00165	0.70	0.40	0.0099
RF (B2N)	0.85	0.12	0.00126	0.59	0.32	0.0097
FFT + RF (B2N)	0.89	0.17	0.00126	0.49	0.45	0.0097

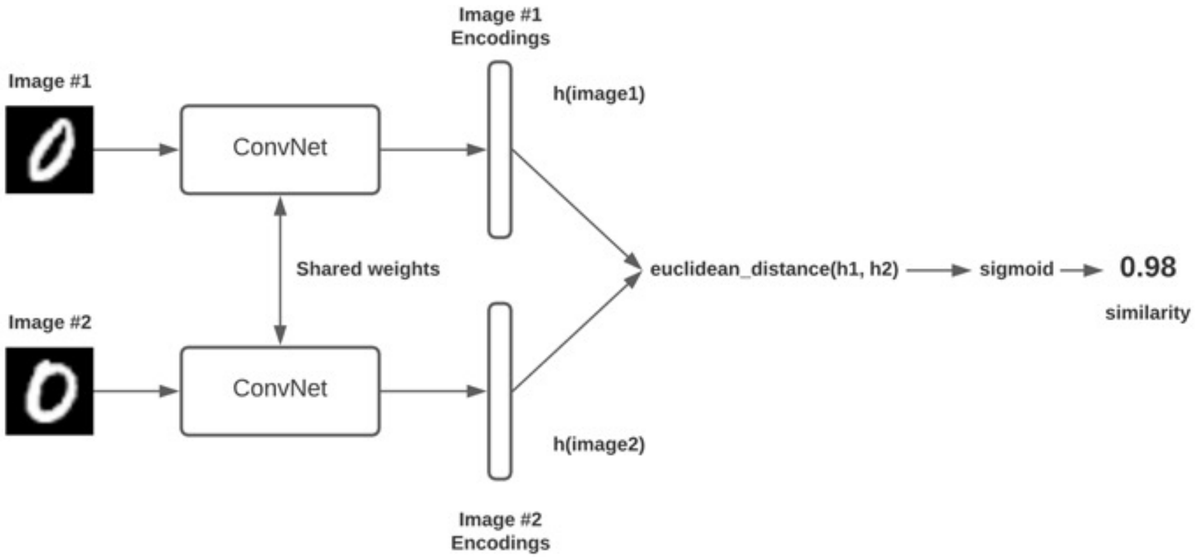
00110000 (no SCL beamloss): (FP/TP): 0.00126 / 0.17 = per day: 40/233 predicted trips, 6531 false alarms
NOT 00110000 (SCL beam loss): (FP/TP): 0.0008 / 0.74 = per day: 20/27 predicted trips, 4133 false alarms
(~5,184,000 pulses per day)

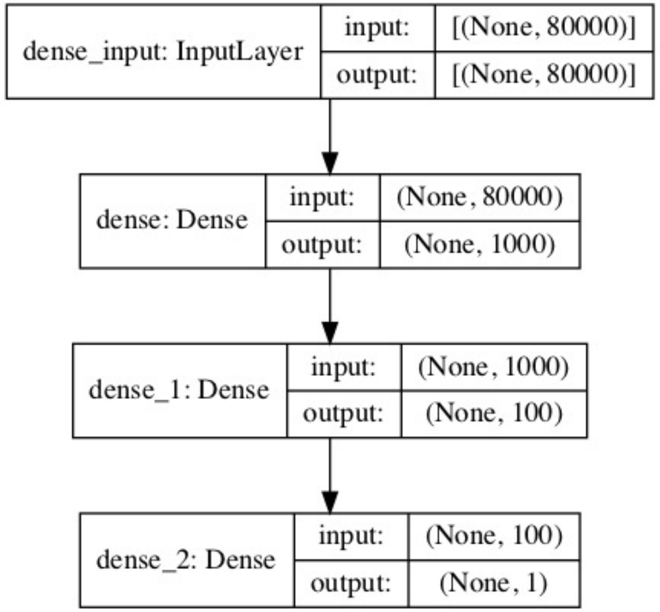
- Focusing on events with beam loss, we can already predict 75% of errant beam pulses with less than 0.1% of good beam aborted!
- Classifiers require 0.6 – 3 milliseconds on laptop.

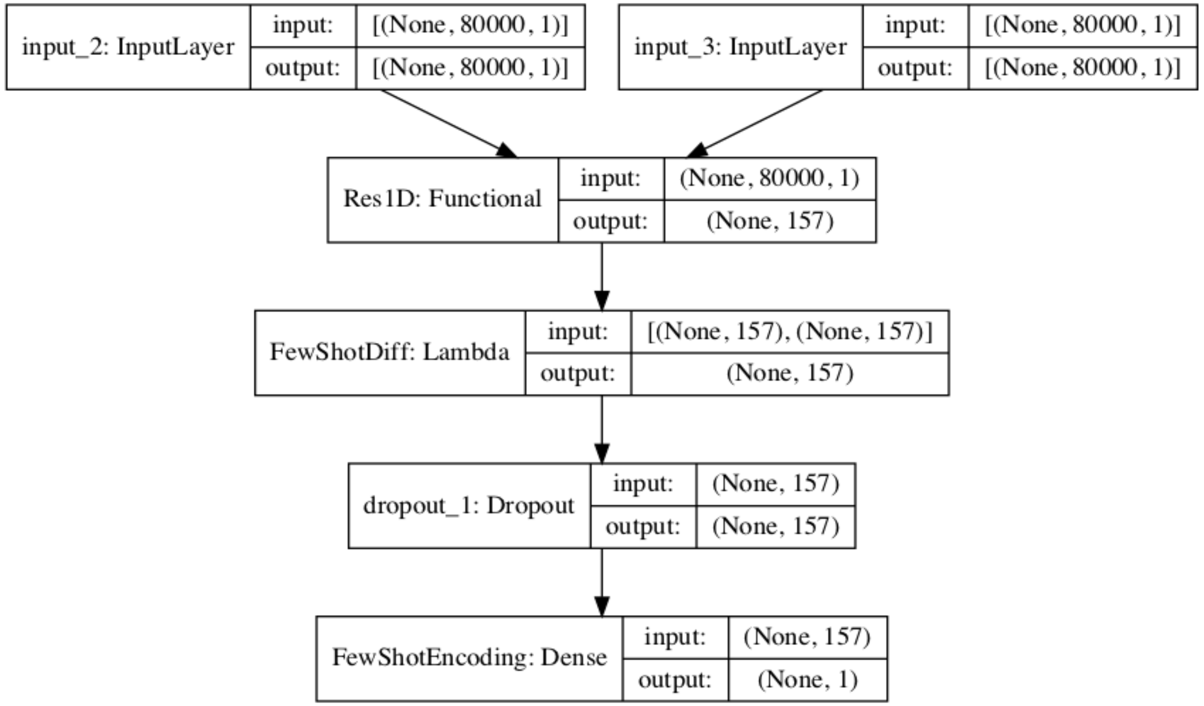
[1] Rescic, M., Seviour, R., & Blokland, W. (2020). Predicting particle accelerator failures using binary classifiers. *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers,*

Stage 3 - Machine Learning: Classification vs. Similarity

- We are using a Siamese model since we want to focus on the similarity between a reference pulse and the current pulse
- Siamese model does not explicitly model the classification but focuses on the similarities
- Embedding is done using a ResNetCov1D







Stage 3 - Machine Learning: Siamese Model

- Using a Siamese model to focus on the similarity between a reference (aka “normal”) pulse and the current pulse
- Siamese model does not explicitly model the classification but focuses on the similarities

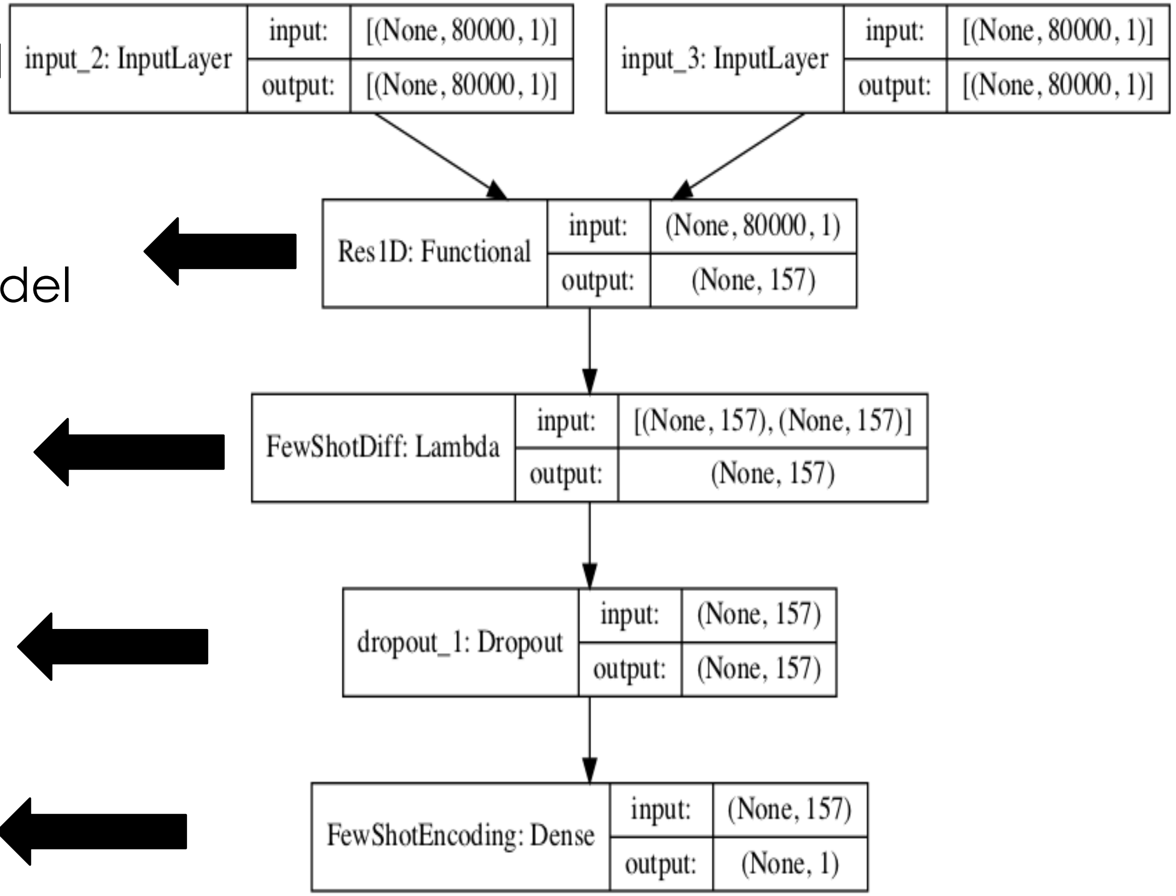
Input data
- Left input is mixed
- Right input is all “normal”

Reduced feature representation
- Using 1D convolutional resnet model

Similarity calculation:
- L1 and L2 methods are available

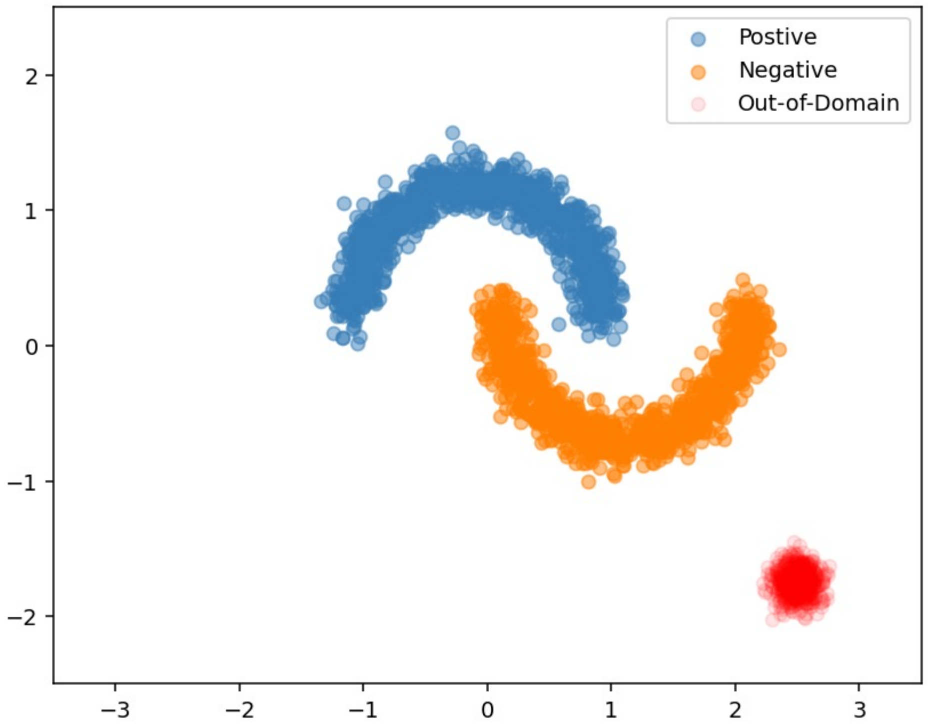
Used to avoid overfitting
- Using dropout layer

Final similarity score



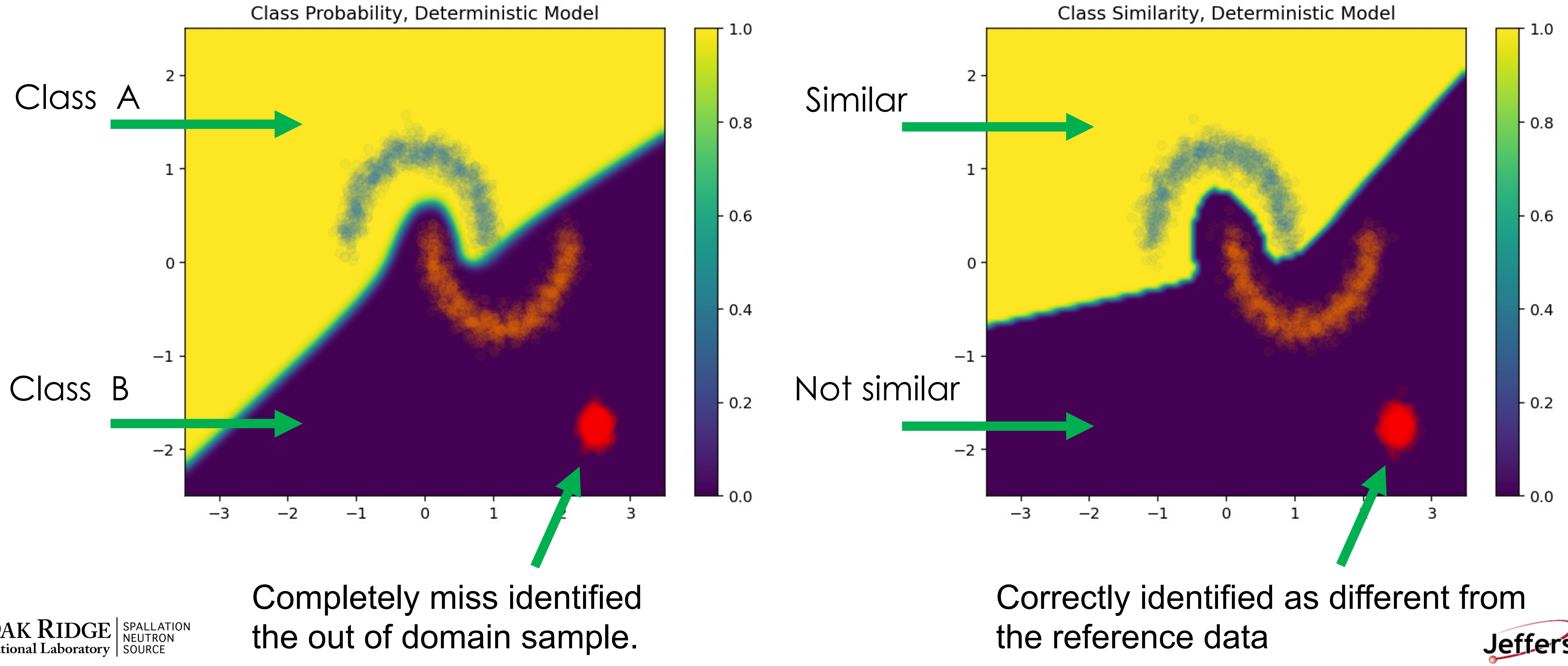
Stage 3 - Machine Learning: Two Moons Example

- In domain training data vs out-of-domain inference.
- The goal is to classify data points in a cartesian grid as “positive” or “negative”
- Traditional approach is to use a classification method
- We also include a 3rd data type that is not part of the training sample “out-of-domain” to illustrate a key problem



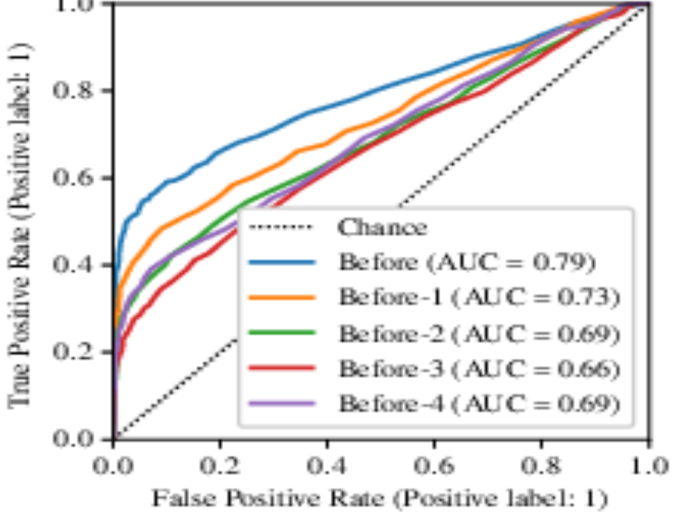
Stage 3 - Machine Learning: Classification vs. Similarity

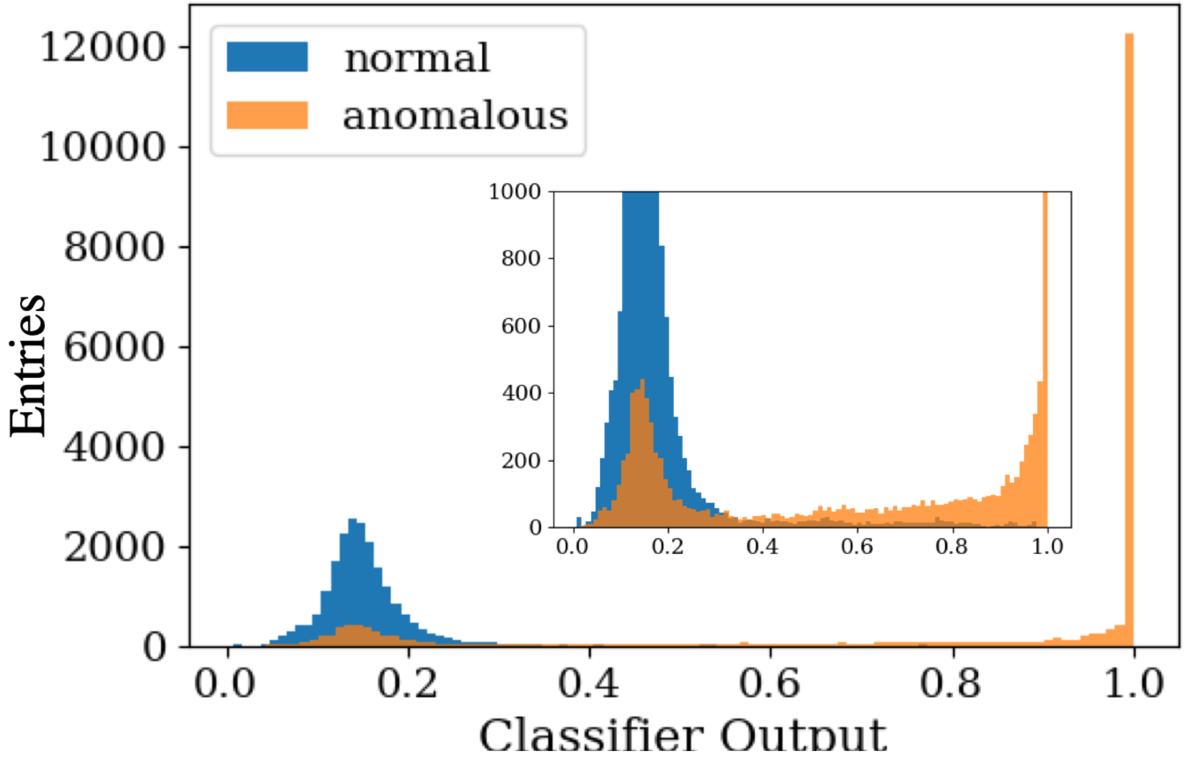
- For a classification algorithm, the goal is to identify each specific class.
- For Siamese algorithm, we are explicitly identifying what is different!

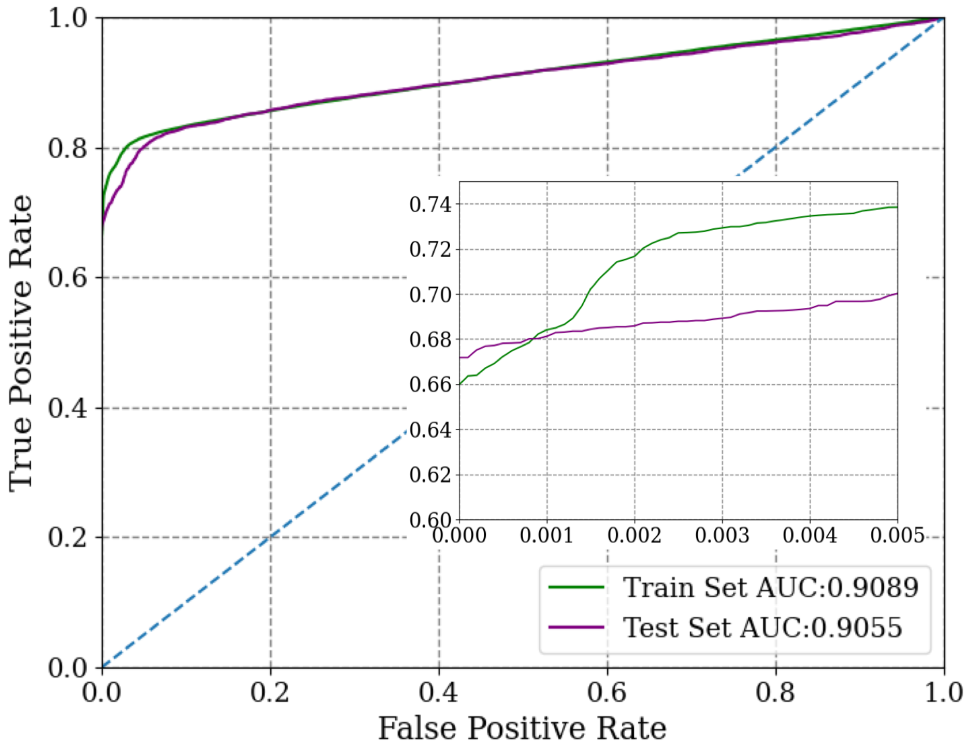


Results for deterministic Siamese Model

- The classification output is smooth and devoid of any weird artifacts (peaks/cliffs)
- The Siamese model as ~4x better performance than the previous RF results





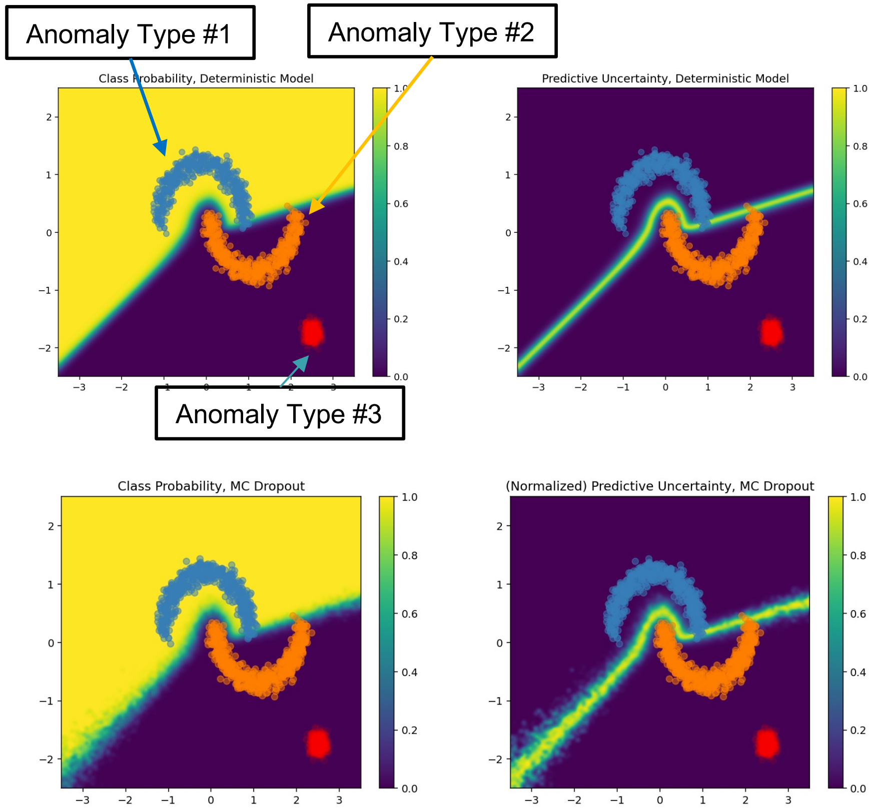


Understand what your model knows and doesn't know

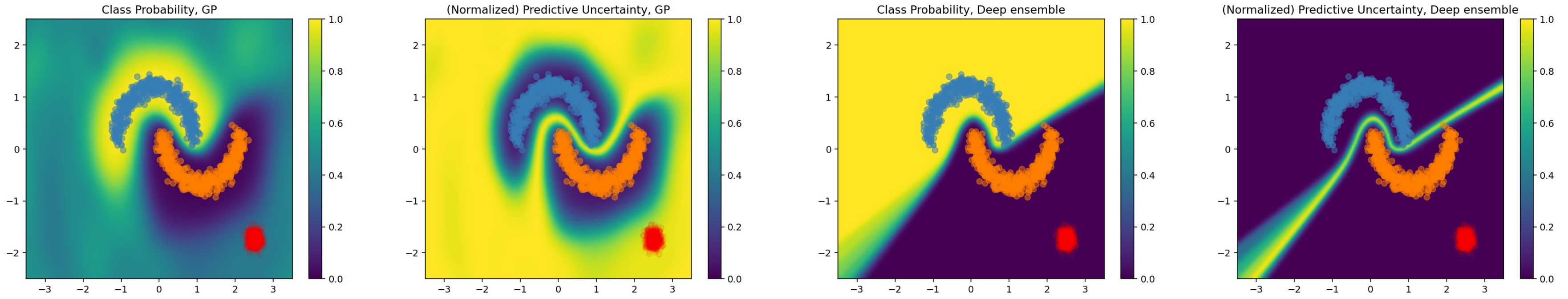
Providing methods to reliably quantify the predictive uncertainty for our models is critical for real-world applications.

Different method yield vastly different classification predictions, some examples:

- Deterministic
- MC Dropout
- Deep Ensemble
- Gaussian Processes
- Bayesian Neural Networks

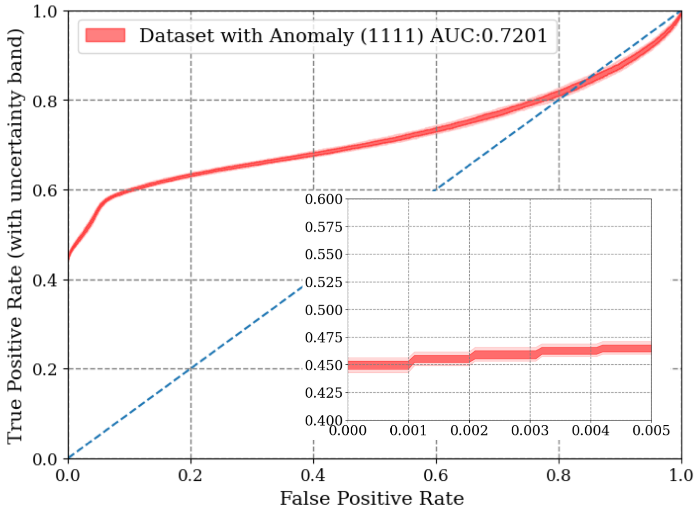


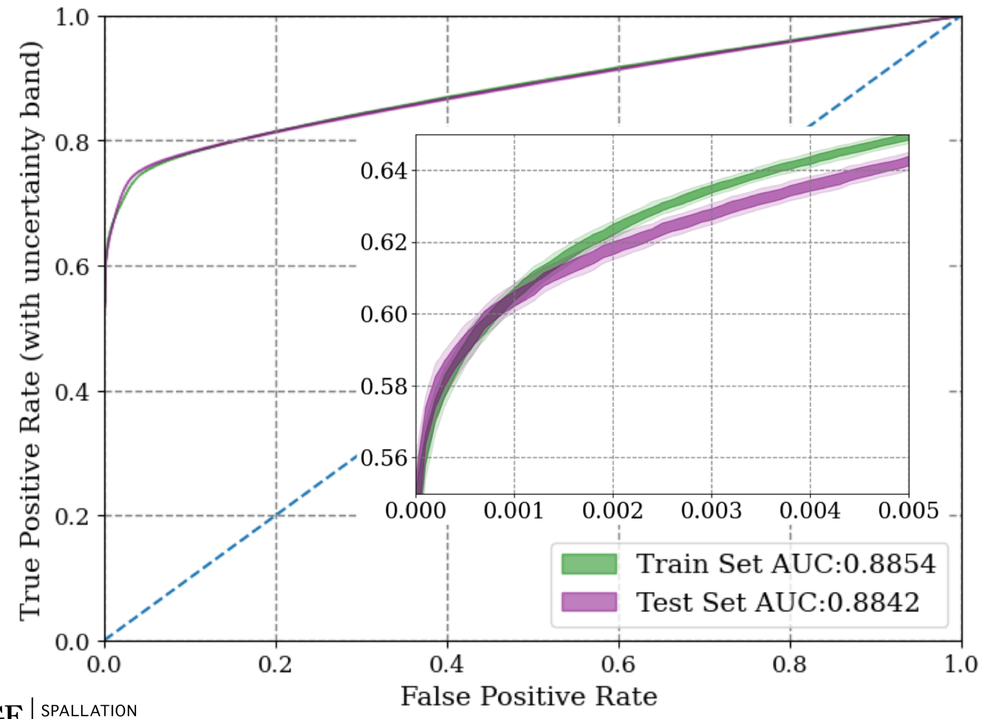
- Different models architectures can yield better results if you do not know all classifications

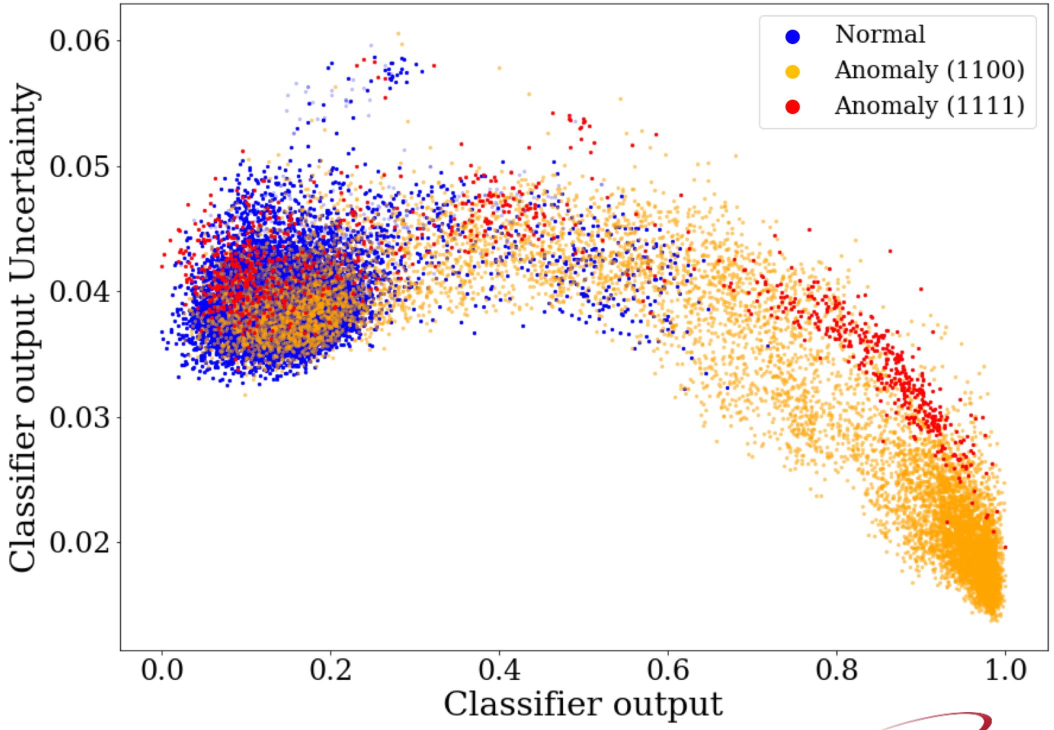


Results of the UQ-based Siamese Model

- We enhanced our Siamese model by adding GP layer providing an uncertainty estimate
- The ROC curves shows nearly the same level of performance (not optimized)
- We introduced an out-of-domain anomaly, labelled 1111 (red), the UQ-based model correctly identified the anomaly and indicated high uncertainty (as expected)







OAK RIDGE National Laboratory | SPALLATION NEUTRON SOURCE

Jefferson Lab

Thank you

M. Schram, K. Rajput, T. Jeske (JLAB)
W. Blokland, P. Ramuhali, C. Peters, Y. Yucesan, A. Zhukov (ORNL)

The authors acknowledge the help from David Brown in evaluating Operations requirements , Frank Liu, for his assistance on the Machine Learning techniques, and Sarah Cousineau for making this grant work possible.

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 Spallation Neutron Source, a DOE Office of Science User Facility operated by the Oak Ridge National Laboratory. The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).