Machine Learning for Improving Accelerator and Target Performance

Machine Learning to improve SNS Operations

W. Blokland, M. Radaideh, Y. Yucesan

for full team:

Malachi Schram, Kishan Rajput, Thomas Britton, Torri Jeske, Lasitha Vidyaratne, Sarah Cousineau (PI), Aaron Young, Austin Bullman, Bryan Maldonado, Cary Long, Charles Peters, Chris Pappas, Dan Lu, David Anderson, David Brown, David Womble, Drew Winder, Frank Liu, Hao Jiang, Hoang Tran, Jared Walden, Lianshan Lin, Majdi Radaideh, Mark Wezensky, Matt Howell, Miha Rescic, Mike Dayton, Narasinga Miniskar, Nolan Goth, Pradeep Ramuhalli, Rich Crompton, Sarma Gorti, Sasha Zhukov, Vivek Rathod, Willem Blokland, Yigit Yucesan,

ORNL is managed by UT-Battelle, LLC for the US Department of Energy
JLab is managed by Jefferson Sc. Assoc., LLC for the US Department of Energy

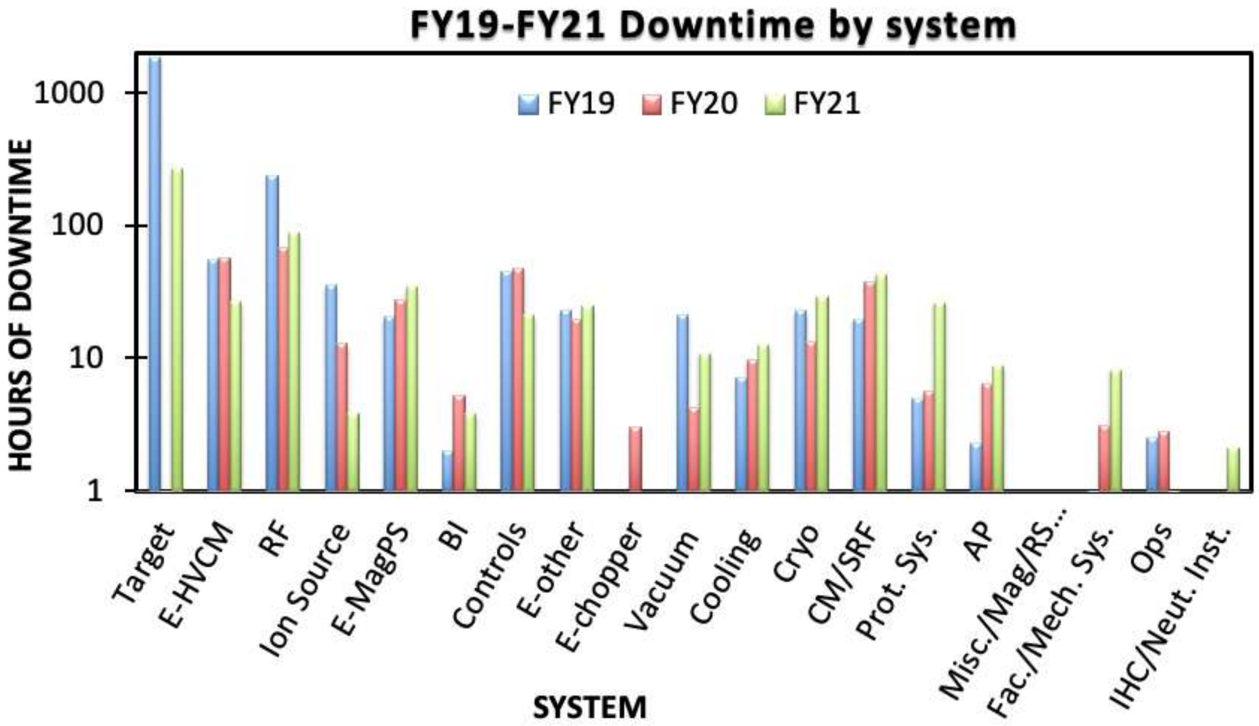
Outline

- Machine Learning to improve operations
  - Minimize downtimes of target and accelerator
- What is Machine Learning
- Background:
  - First system: Differential Beam Current Monitor (University of Huddersfield)
- Use-cases in collaboration with JLAB
  - Beam-based
  - High Voltage Converter modulator
  - Target
  - Cryo Moderator System

Why Machine Learning at SNS

- Not all problems are (can be) well defined or understood
  - System not well understood (cryo loop), models incomplete (target, HVCM)
  - Large data sets that are hard or not suitable to process with classical methods

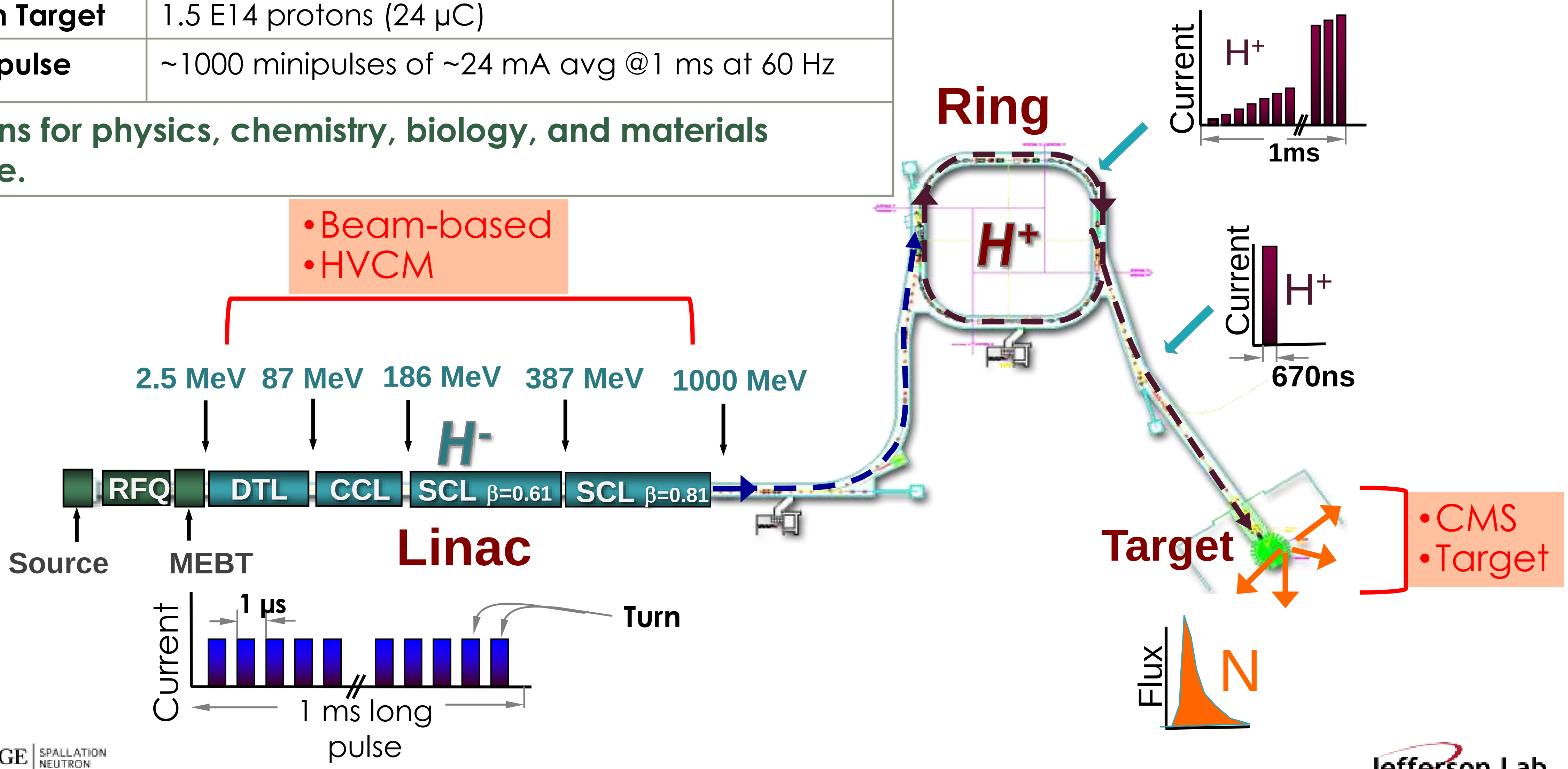
- Many improvements have been made over the years, but we still have downtimes → can ML decrease downtimes even further?
  - Proton Power Upgrade
  - Second Target Station



SNS downtime statistics

Spallation Neutron Source Complex

Power on Target	1.4 MW at 1.0 GeV
Pulse on Target	1.5 E14 protons (24 μC)
Macro-pulse	~1000 minipulses of ~24 mA avg @1 ms at 60 Hz
Neutrons for physics, chemistry, biology, and materials science.	



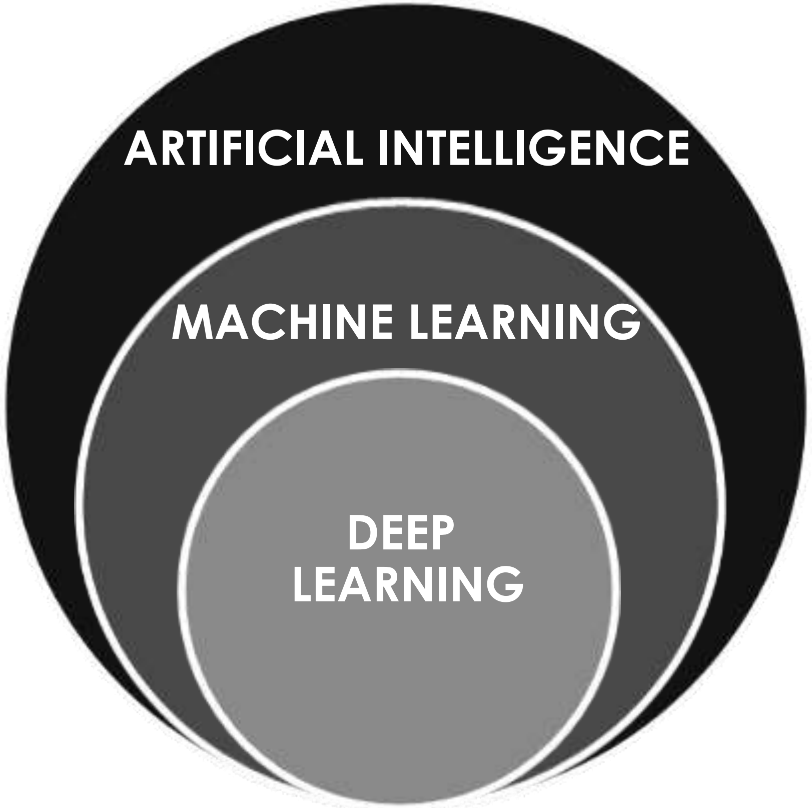
Machine Learning in a Nutshell

*“**AI** leverages computers and machines to mimic the problem-solving and decision-making capabilities of the human mind.”*

*“**ML** is a branch of **AI** which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.”*

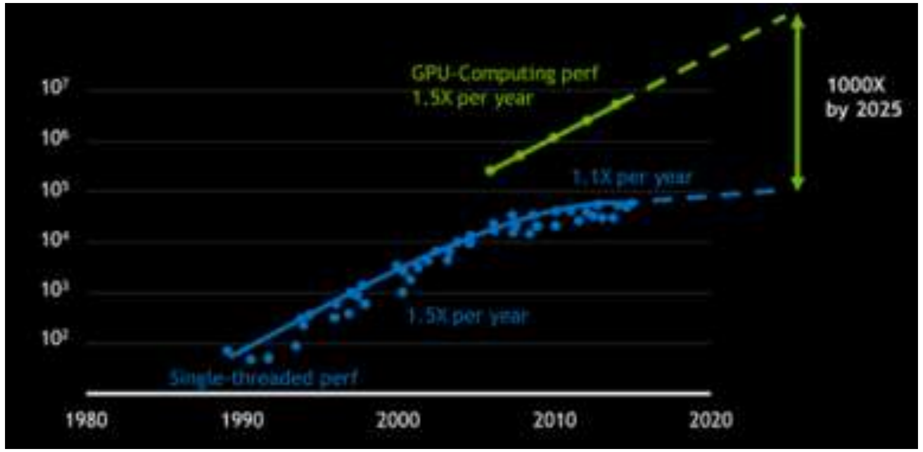
*“**DL** as a subset of **ML** use neural networks with hidden layers to learn from vast amount of data.”*

https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks

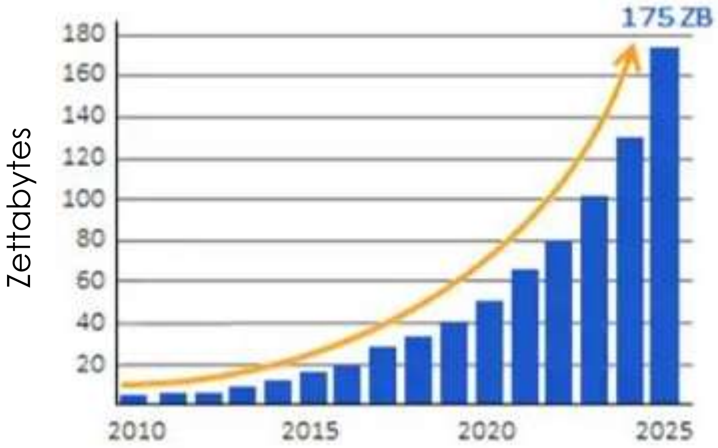


Why is Machine Learning becoming more popular?

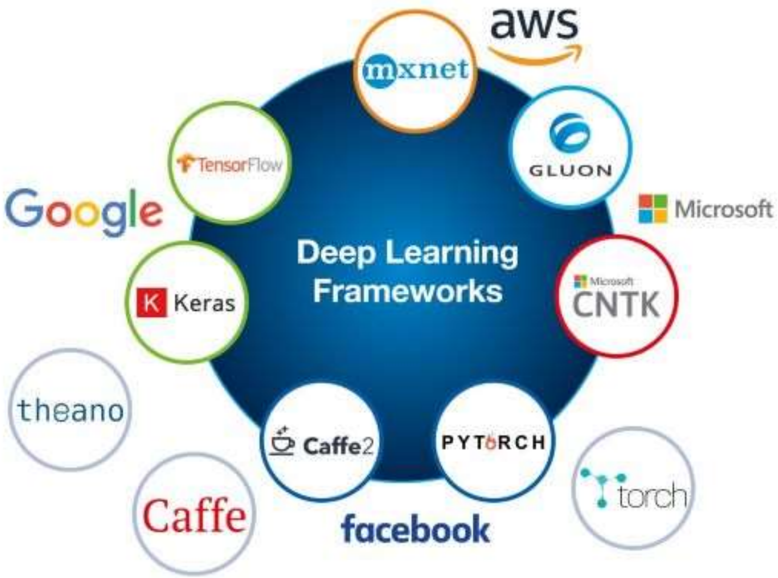
1. Rise of GPU Computing



2. Annual Size of the Global Datasphere



3. Open-Source Machine Learning Community



[1]https://blogs.nvidia.com/blog/2017/05/24/ai-revolution-eating-software/
[2]https://medium.com/analytics-vidhya/the-5-vs-of-big-data-2758bfcc51d
[3]https://devopedia.org/deep-learning-frameworks

Machine Learning Types

Supervised Learning	Unsupervised Learning	Reinforcement Learning
Labeled inputs and outputs	Unlabeled data	Rewards system

Supervised Learning

- **Classification**
  - ✓ Diagnostics
  - ✓ Fraud detection
- **Regression**
  - ✓ Prognostics
  - ✓ Weather forecasting

Unsupervised Learning

- **Clustering**
  - ✓ Customer segmentation
- **Dimensionality Reduction**
  - ✓ Structure discovery

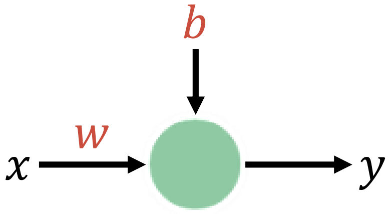
Reinforcement Learning

- **Dynamic Programming**
  - ✓ Robot navigation
- **Environment-based**
  - ✓ Game AI

- Slides from tutorial at SNS complete with demo code

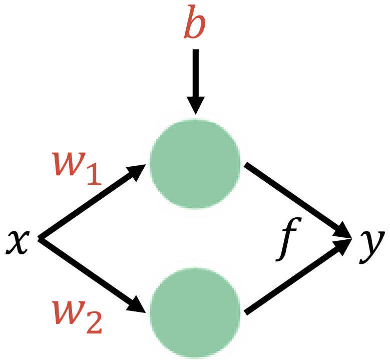
Backbone of ML: Artificial Neural Networks

Single neuron: linear regression

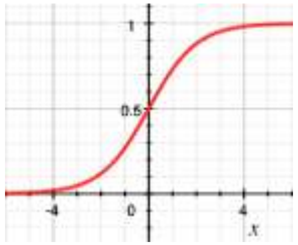


$$wx + b = y$$

Activation function: introduce nonlinearity to learn complex correlations

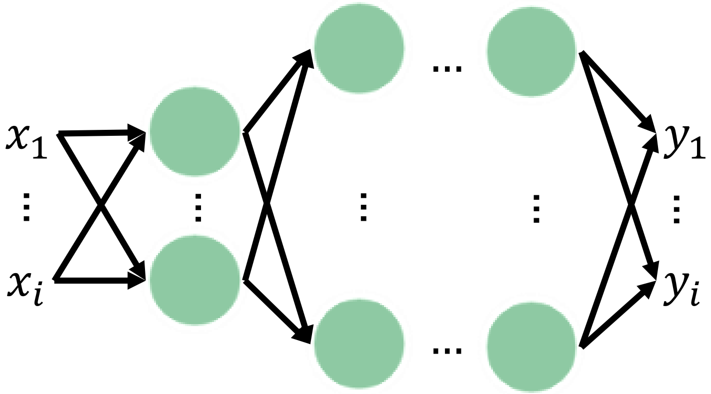


$$f(w_1 x + w_2 x + b) = y$$



$$y = \frac{1}{(1+e^{-x})}$$

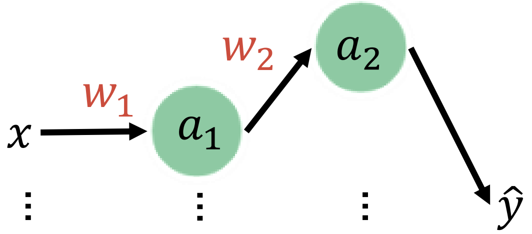
Fully connected multi-layered neural networks



Neural Networks Training: Forward and Backward Pass

Forward

- Outputs and ground truth data used to calculate the ***loss function***
- Selection of the loss function depends on the problem:
  - *Mean Squared Error* — Deterministic
  - *Mean Absolute Error* — Deterministic
  - *KL – Divergence* — Probabilistic
  - *Maximum Likelihood* — Probabilistic



$$a_1 = w_1 x$$

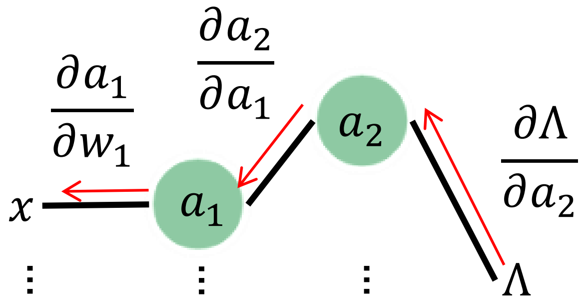
$$\hat{y} = a_2 = w_2 a_1$$

$$\Lambda = (y - \hat{y})^2$$

Lambda = loss function

Backward

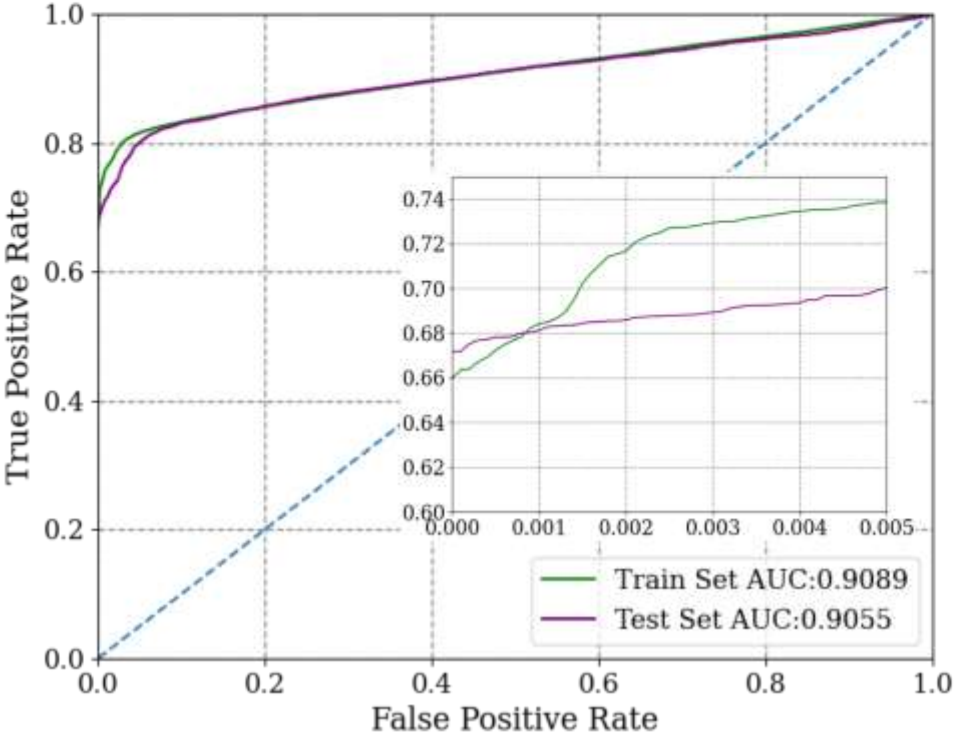
- Gradients calculated using chain rule
- Loss and activation functions must be differentiable (or have the gradients provided)



$$\frac{\partial \Lambda}{\partial w_1} = \frac{\partial \Lambda}{\partial a_2} \frac{\partial a_2}{\partial a_1} \frac{\partial a_1}{\partial w_1}$$

Machine Learning Performance Metrics

- Concepts:
  - ROC curve: Receiver Operating Characteristic curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.
  - True Positive Rate (TPR) = TP/P = TP/(TP+FN)
    - TP= True Positives, P = Positives, FN = False Negative
  - False Positive Rate (FPR) = FP/N = FN/FN+TP
    - FP=False Positives, N=Negatives
    - For SNS: FPR = FP/N ≈ FP(N+P) as N>>P



ROC curve showing the performance of the ML method

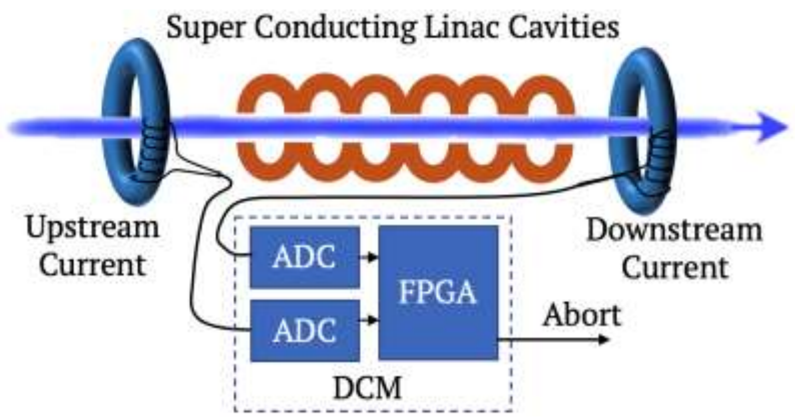
We want low FP or FPR and high TP or TPR

ML Learning Projects at SNS Accelerator and Target

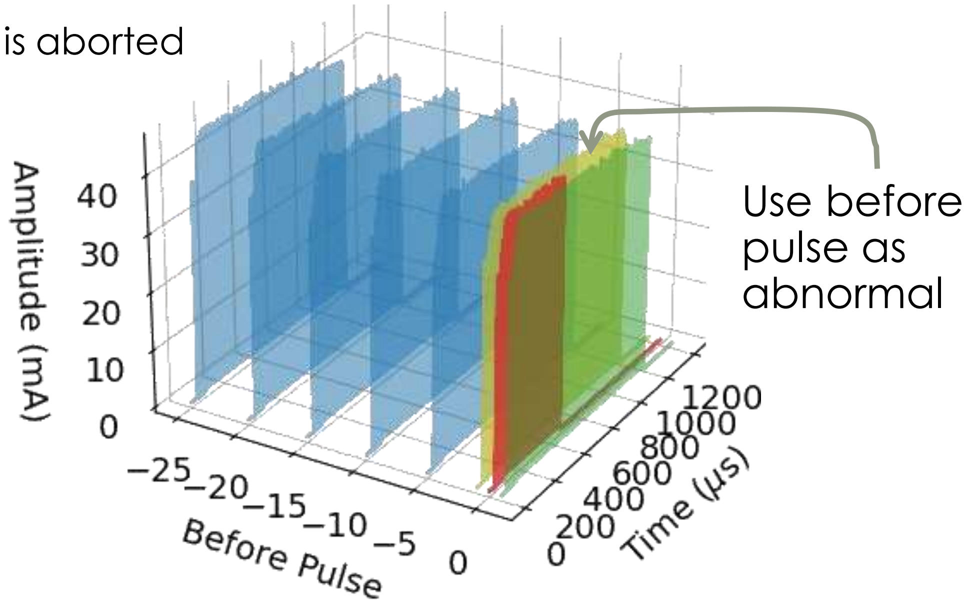
- PhD Student Miha Rescic (Huddersfield University, Rebecca Seviour)
  1. Errant beam prediction using beam current data (2015)
- BES Grant, PI: Sarah Cousineau
  1. Beam-based: Predict errant beam, classify equipment faults
  2. Target: Improve target modeling to increase lifetime
  3. HVCM: Predict failure and prognostics to determine component lifetime remaining
  4. CMS: Better controller algorithm to reduce downtime

Beam-based: Using Differential Beam Current Monitor

- Goal
  - Prevent cavity damage and avoid equipment down times
- Approach
  - Expensive to install diagnostics per equipment. But equipment affects beam→ leaves fingerprint
  - Use existing diagnostics → Differential Beam Current Monitor
    - Archives at full rep rate (LabVIEW FPGA and RT) when beam is aborted



Differential Current Monitor to protect SCL from beam loss damage (2013)*



DCM archives not only errant beam pulses but also up to 25 pulses before and two after → the before pulse becomes the "abnormal" class pulse

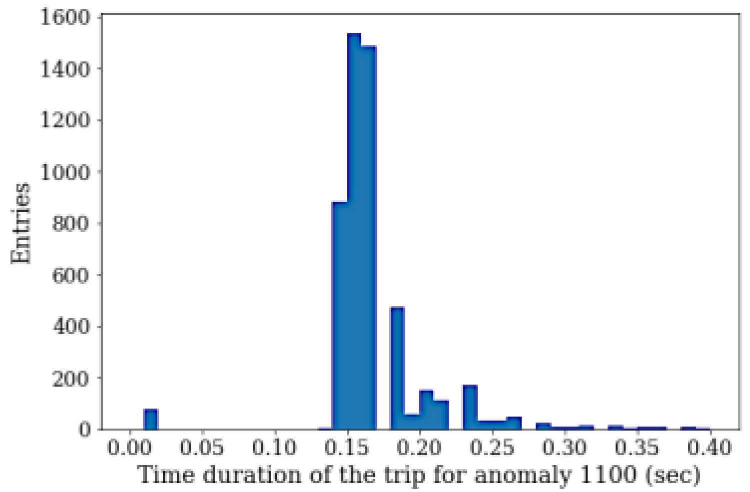
*Blokland, Willem, and Peters, Charles C. *A NEW DIFFERENTIAL AND ERRANT BEAM CURRENT MONITOR FOR THE SNS* ACCELERATOR*. IBIC 2013 conference proceedings, pp921 to 924, Oxford, United Kingdom, Sep 16, 2013 - Sep 19, 2013

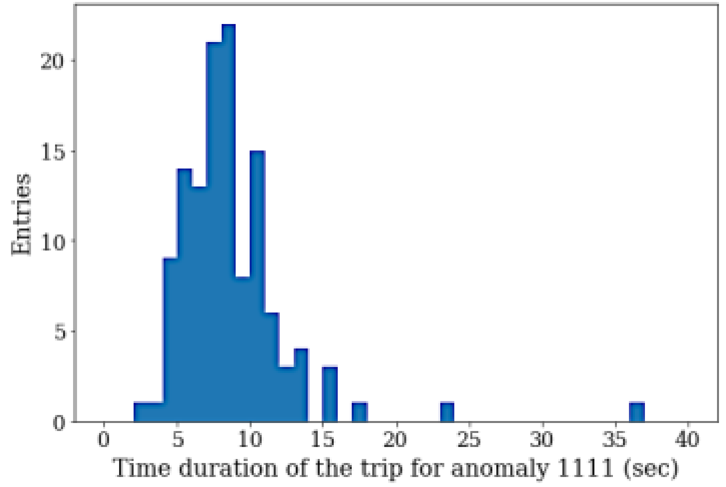
Errant Beam metrics

- The DCM archives data:
  1. When (Downstream – Upstream) > threshold
     - Beam loss in the SCL: **1111 events**
  2. When the pulse is truncated
     - Beam loss upstream or aborted by another device: **1100 events**

- Metrics: How well should ML perform
  - March 2021, production was 26.4 days, 1.5% beam lost
    - 0.22% beam lost due to SCL beam loss
    - 1.30% beam lost due to truncated beam
  - We need to predict a fraction of the errant pulses: TPR ≈ 50%
  - We shouldn't add much down-time due to false positives
    - An insignificant amount would be 0.2% of beam pulses
    - but penalty is 4 pulses per abort

→ we want to achieve a FPR ≈ 0.05%





Trip statistics derived from DCM data

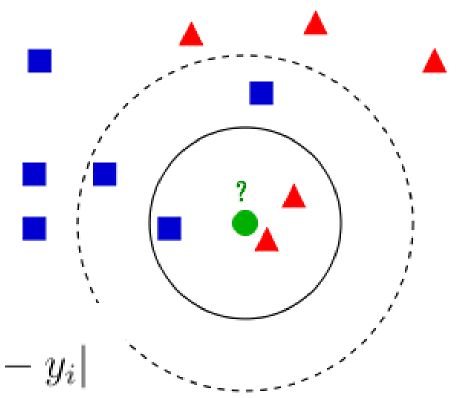
Beam-based: Early Work by Miha: K-NN Method

Method

- K-Nearest Neighbor Method using different distance functions L1, L2, and CC

Results

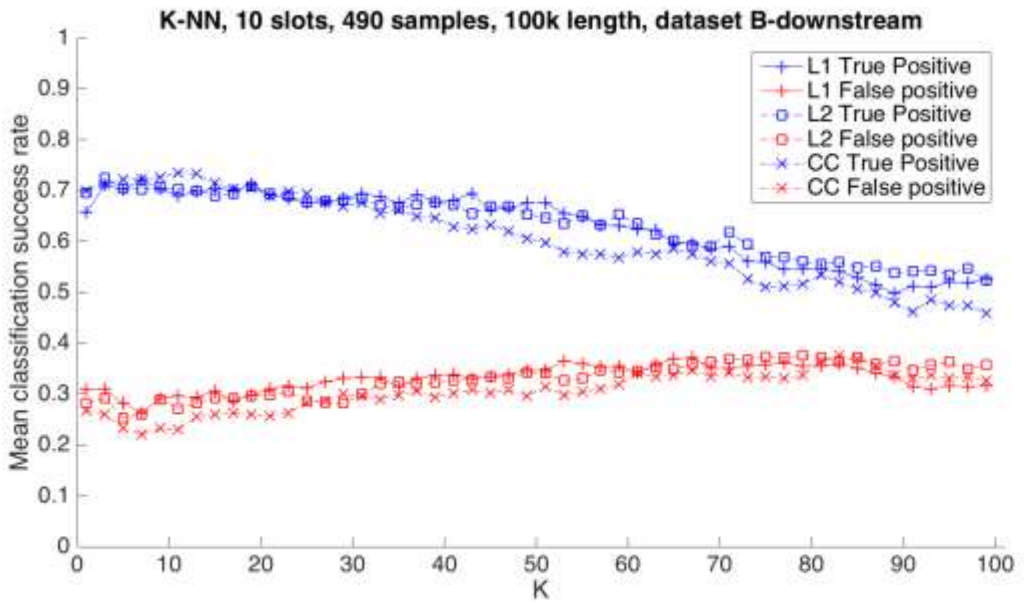
- Up to 75% success rate but very high FPR



$$L_1(x, y) = \sum_{i=1}^{N} |x_i - y_i|$$

Credit: Wikipedia

K-NN: Assign new data point class based on distance to training set data points



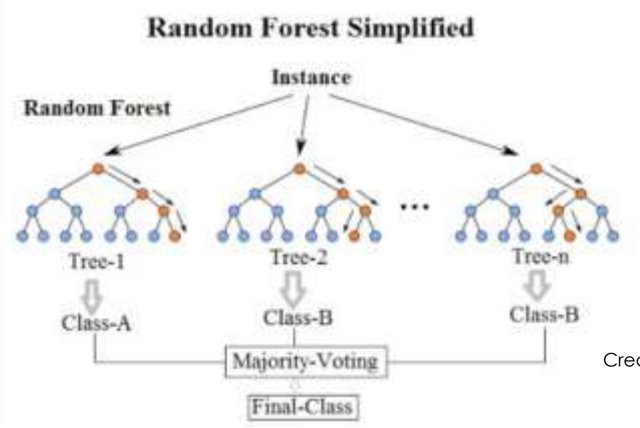
K-NN Plot: Very typical of K-NN is to get better success when increasing K at first but eventually for large K it will mimic the ratio of good and bad pulses

→ While there is indication that we find precursors, we abort too much beam

Beam-based: Errant Beam Work by Miha*

Method

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

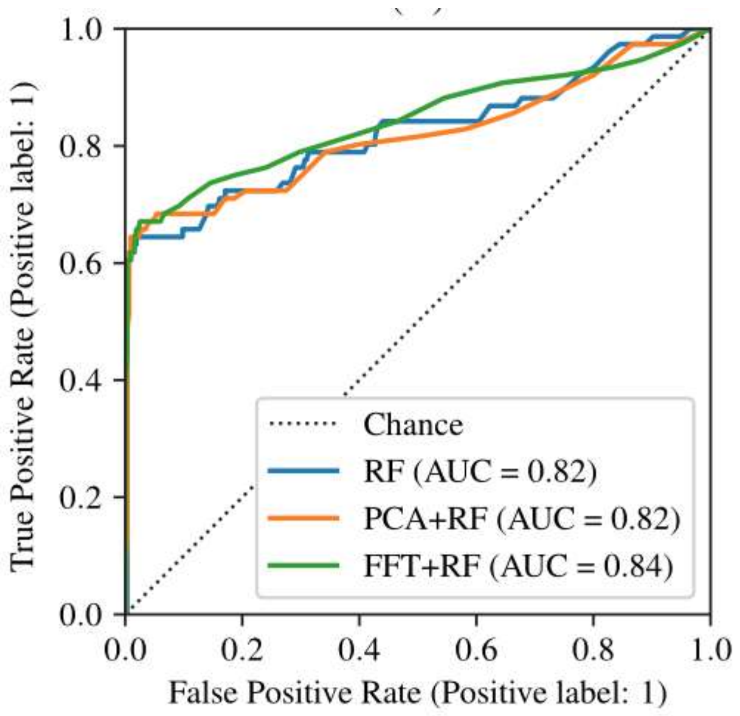


Credit: Wikipedia

Results

- No SCL beam loss: 40/233 predicted trips, 6531 false alarms
- SCL beam loss: 20/27 predicted trips, 4133 false alarm
- (~5,184,000 pulses per day)

We predict 75% of SCL beam loss pulses with ~0.2% *4 of good beam aborted.

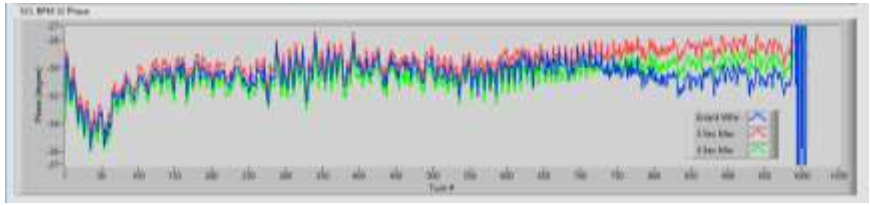


ROC curve with preprocessing

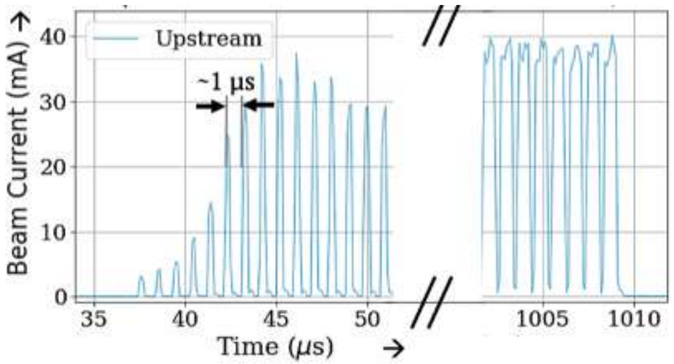
*M. Reščič, R. Seviour, W. Blokland, Improvements of pre-emptive identification of particle accelerator failures using binary classifiers and dimensionality reduction,, NIM-A,Volume 1025, 2022, 166064,ISSN 0168-9002, https://doi.org/10.1016/j.nima.2021.166064.

Beam-based: Next Phase

- Approaches
  - Beam Position Monitor phase data:
    - Map upstream to downstream to detect abnormal pulses. If mapped version differs from measured, then we have an abnormal condition
  - Differential Current Monitor data:
    - Identify the faulty equipment using labeled Machine Protection System (MPS) data
    - Siamese twin model to detect abnormal beam pulses
      - This model looks at similarities of two inputs and provides you with a similarity value



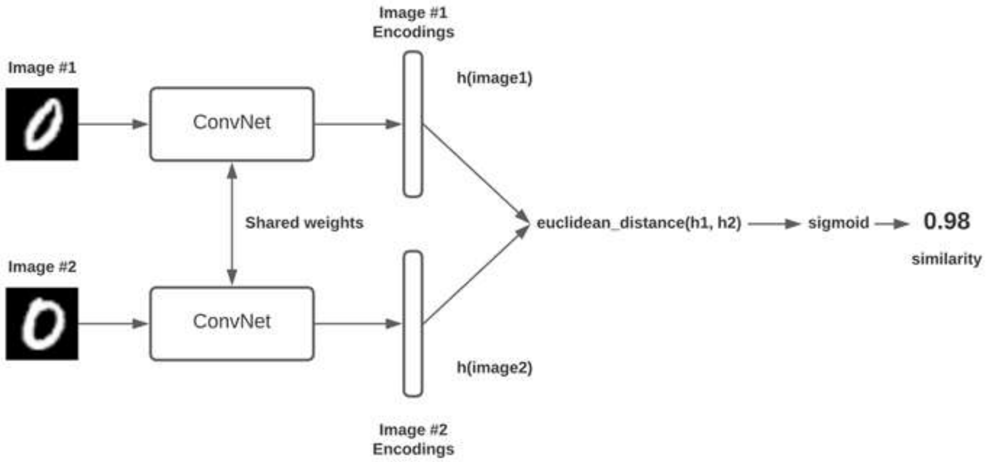
BPM phase turn-by-turn data



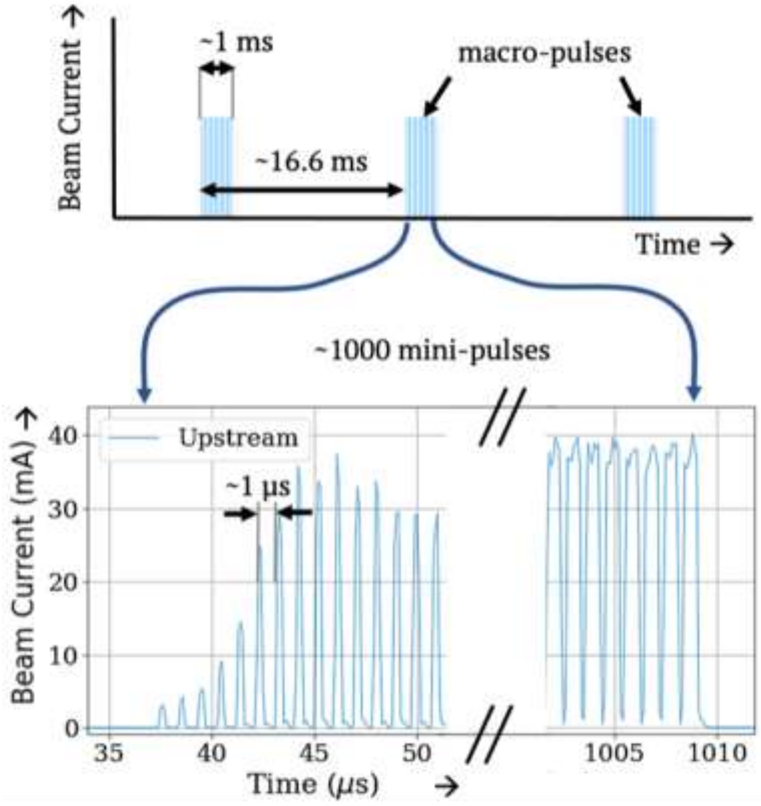
Beam current waveform

Beam-based: Uncertainty aware anomaly detection

Siamese Model:



- By using a reference pulse from the training set, we can compare a normal pulse to a normal reference pulse to see if they are still similar (if not, retrain)
- We can run multiple inferences of same pulse versus multiple references to majority vote
- Similarity allows to classify pulses not seen before



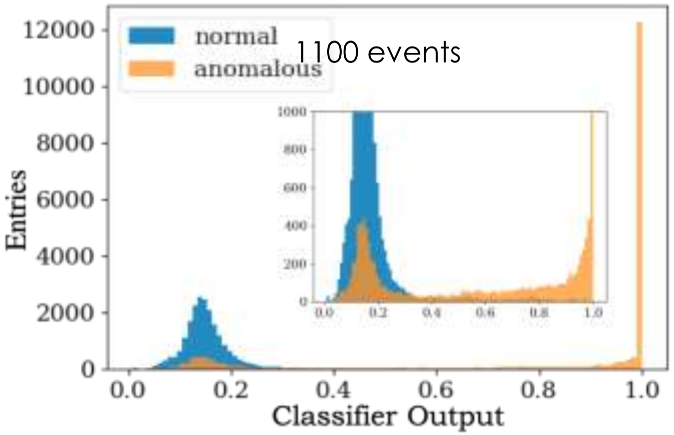
DCM data: 60 Hz pulses sampled at 100 MS/s

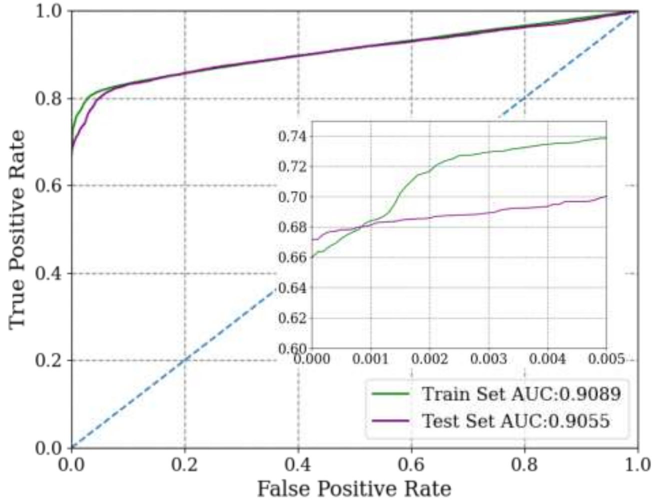
Beam-based: Uncertainty aware anomaly detection

- Probabilistic model (Gaussian Approximation) adds uncertainty to similarity predictions
- Anomaly, red 1111, that has not been part of training set, is identified correctly but has higher uncertainty

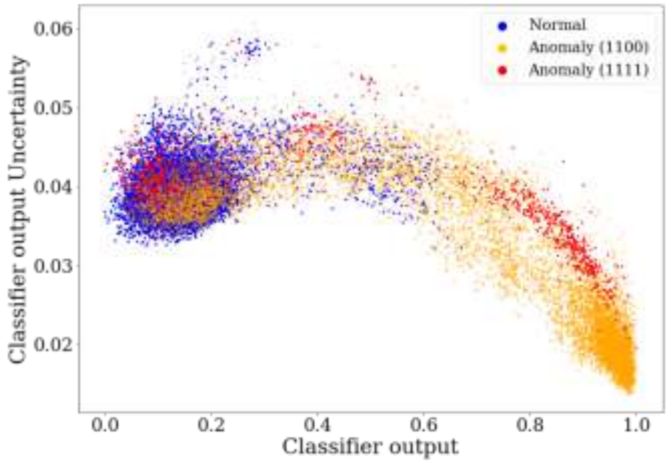
→ We can have very low FPR, e.g. 0.05% aka 0.2% of beam wrongly aborted, with ~50% of abnormal beam predicted*

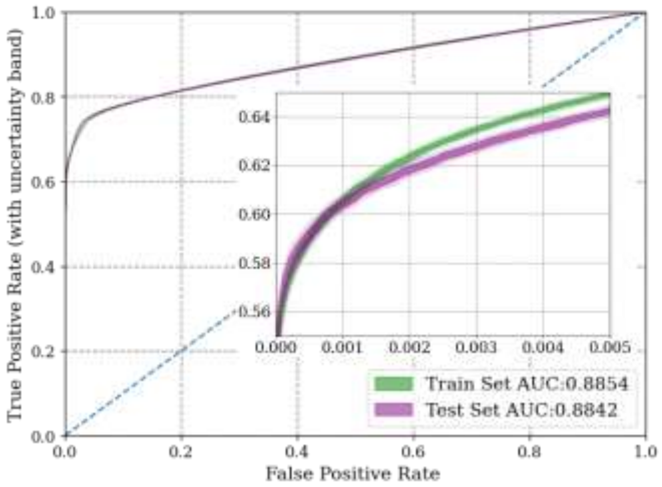
Deterministic Model





Probabilistic Model



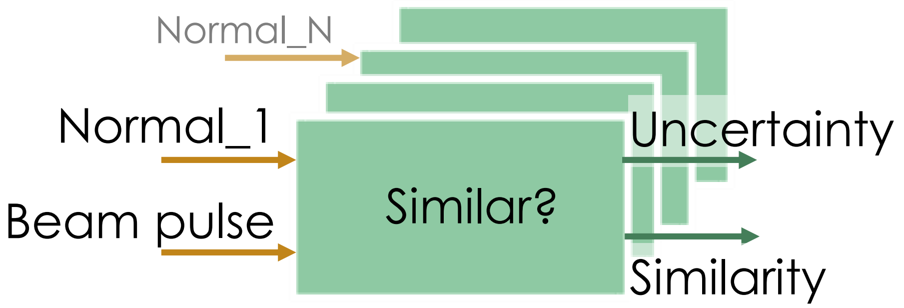


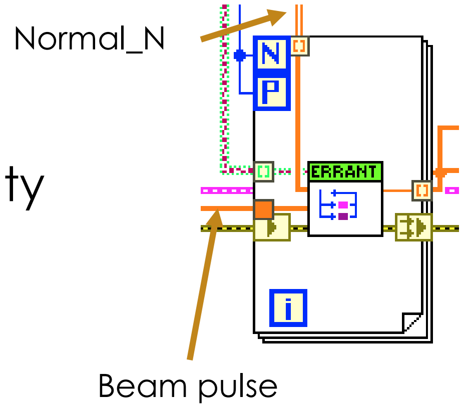
*Blokland, W., Ramuhalli, P., Peters, C., Yucesan, Y., Zhukov, A., Schram, M., ... & Jeske, T. (2021). Uncertainty aware anomaly detection to predict errant beam pulses in the SNS accelerator. arXiv preprint arXiv:2110.12006.

Beam-based: Field Implementation

Implementation in the field

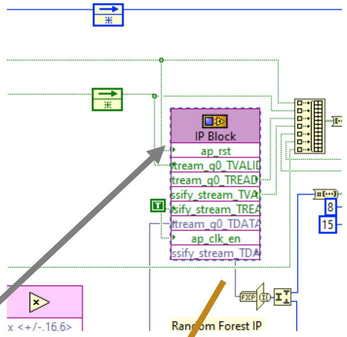
- Installed second DCM (DCML) and fed it duplicate analog signals from beam current sensors
- Implement Siamese model on DCML RT
- Implement RF on FPGA (upcoming paper on development environment)
- Analyze all incoming beam current waveforms

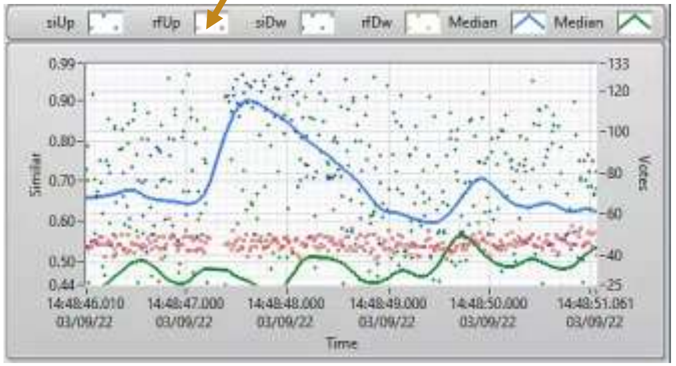




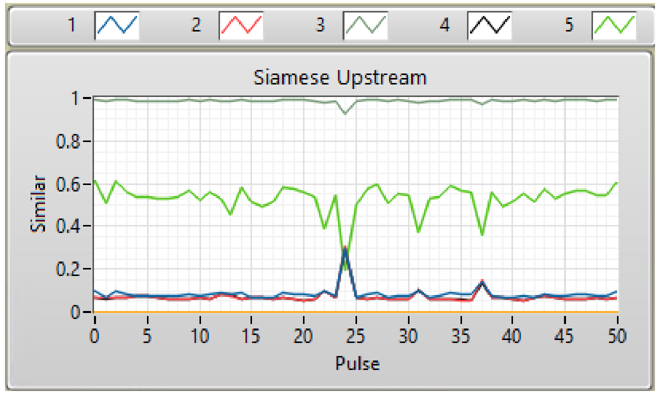
Siamese model applied to multiple Normal references

→ We see certain events but not yet operational in terms of TP and FP





Pulse-by-pulse analysis on up/down stream both Siamese and RF

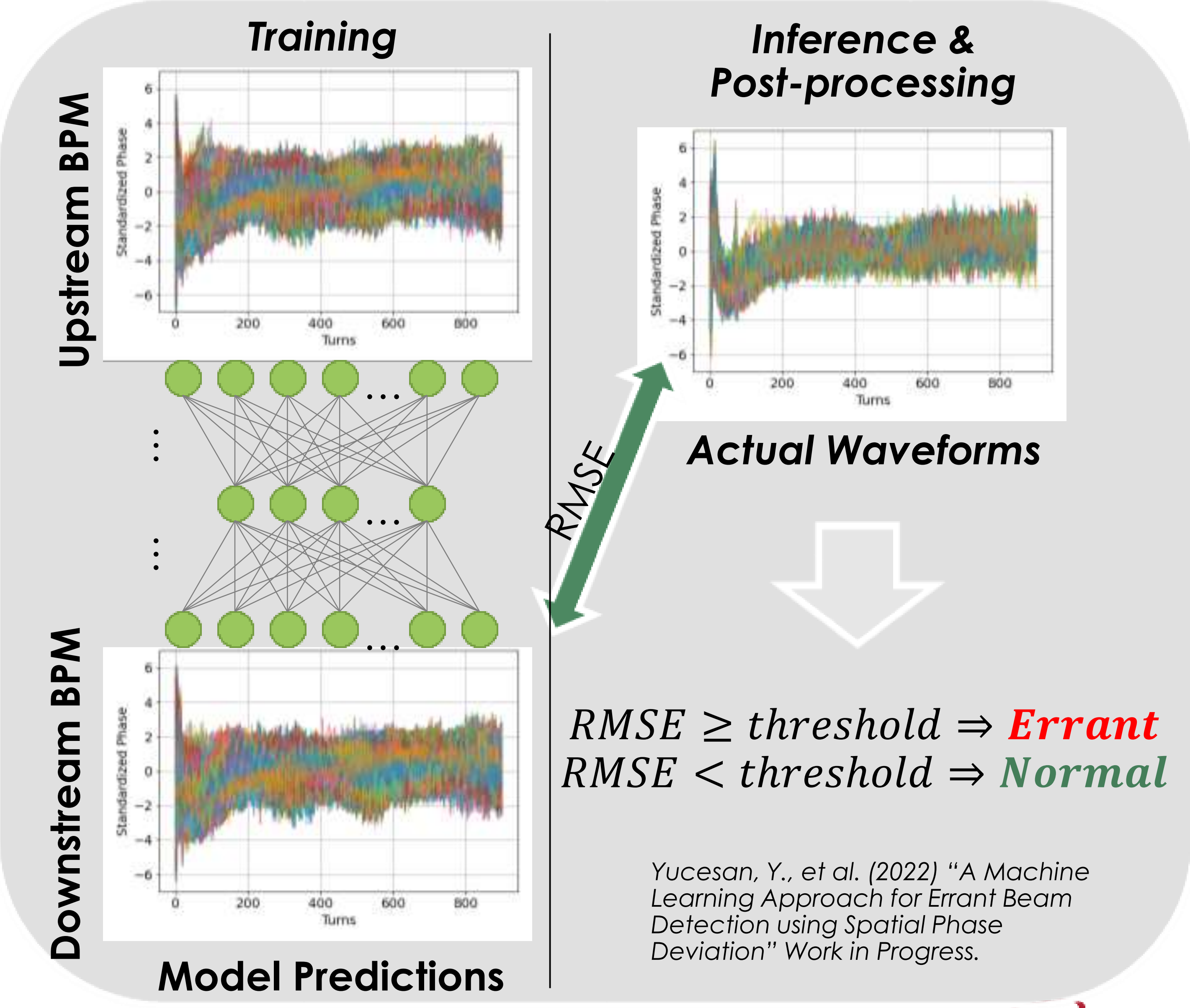


5 Inferences with different references per beam pulse

Beam-based: BPM Phase Data

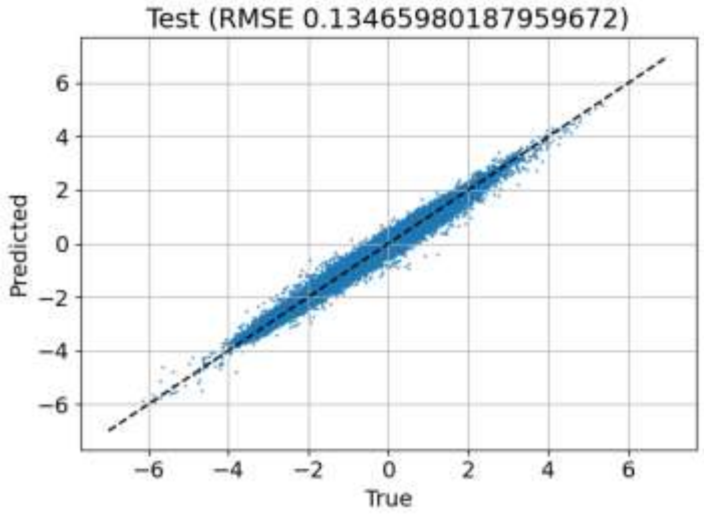
- Use **normal** traces to model mapping from an upstream and downstream BPM
- Pass **faulty** traces from the trained model
- Compare error between true phase and predicted phase

- ***Model:*** *Multi-layered Perceptron*
- ***Input:*** *HEBT-BPM01 Phase*
- ***Output:*** *HEBT-BPM32 Phase*
- ***Training:*** *Normal Waveforms*



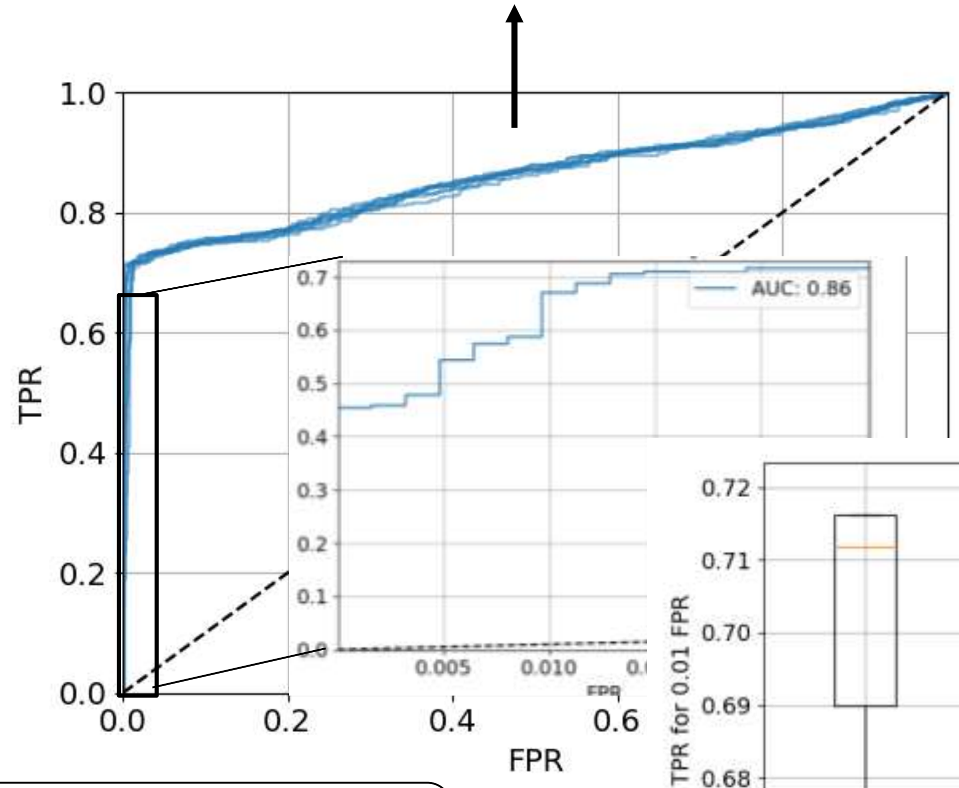
Beam-based: BPM Phase Data

Training and test performance on normal-to-normal mapping

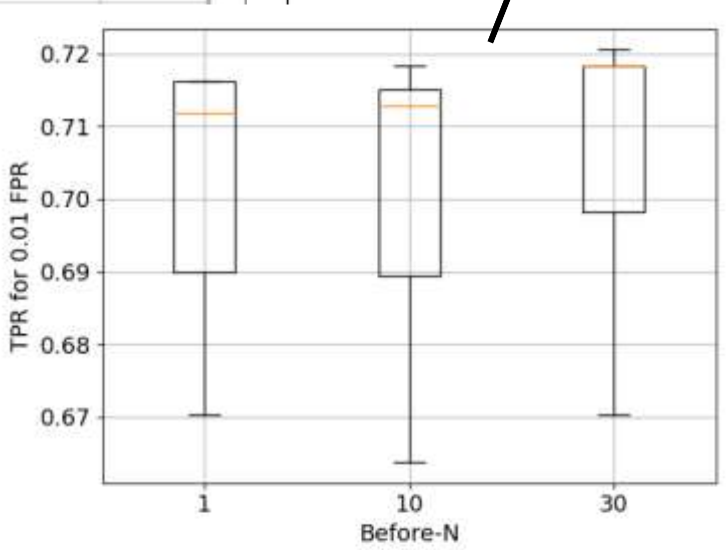


Phase Predicted vs Measured

Cross-validated model performance on target FPR



Precursors on fault 30 pulses before beam trip!



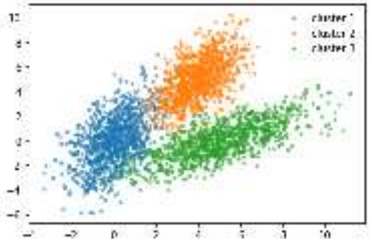
Results:
- FPR < 0.25 % while maintaining TPR ~45%

OAK RIDGE National Laboratory | SPALLATION NEUTRON SOURCE
Jefferson Lab

Beam-based; Equipment Fault Classification

Goal: identify the equipment causing errant beam

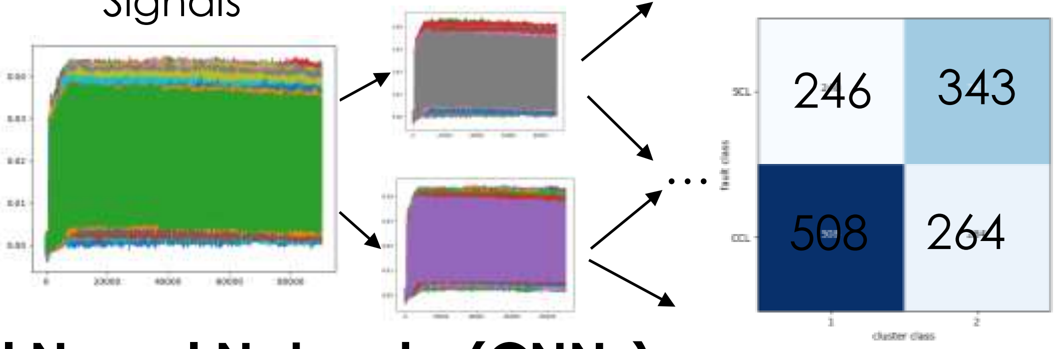
- **Unsupervised Clustering:**



Example of clusters

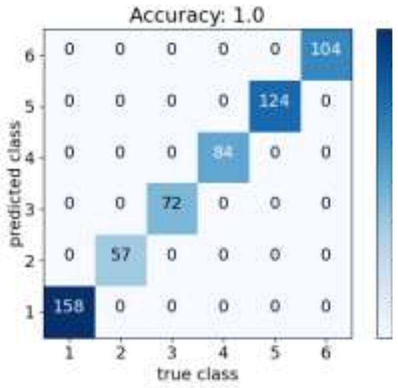
Signals

Confusion Matrix

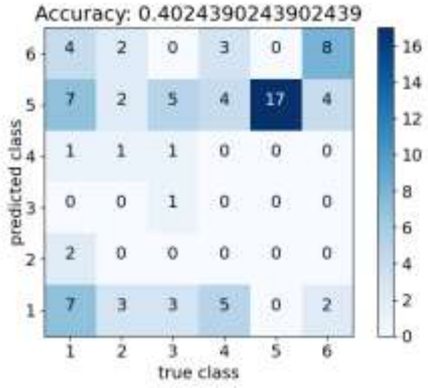


- **Convolutional Neural Networks (CNNs):**

Training



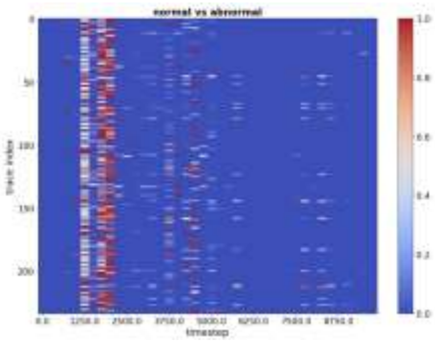
Test



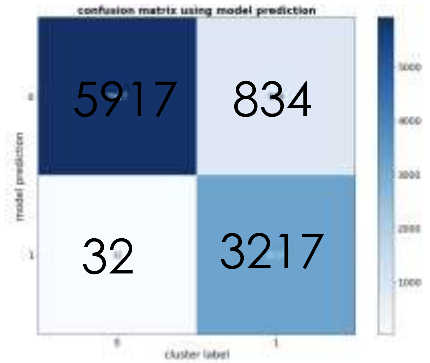
In progress:
- Weak classification performance
- Overfitting

- **Siamese model:**

Use gradCAM* to generate heatmaps and see if heatmaps are different for different equipment



*Gradient-weighted Class Activation Mapping



In progress:
- Ok for finding anomalies → might be ok for equipment classes

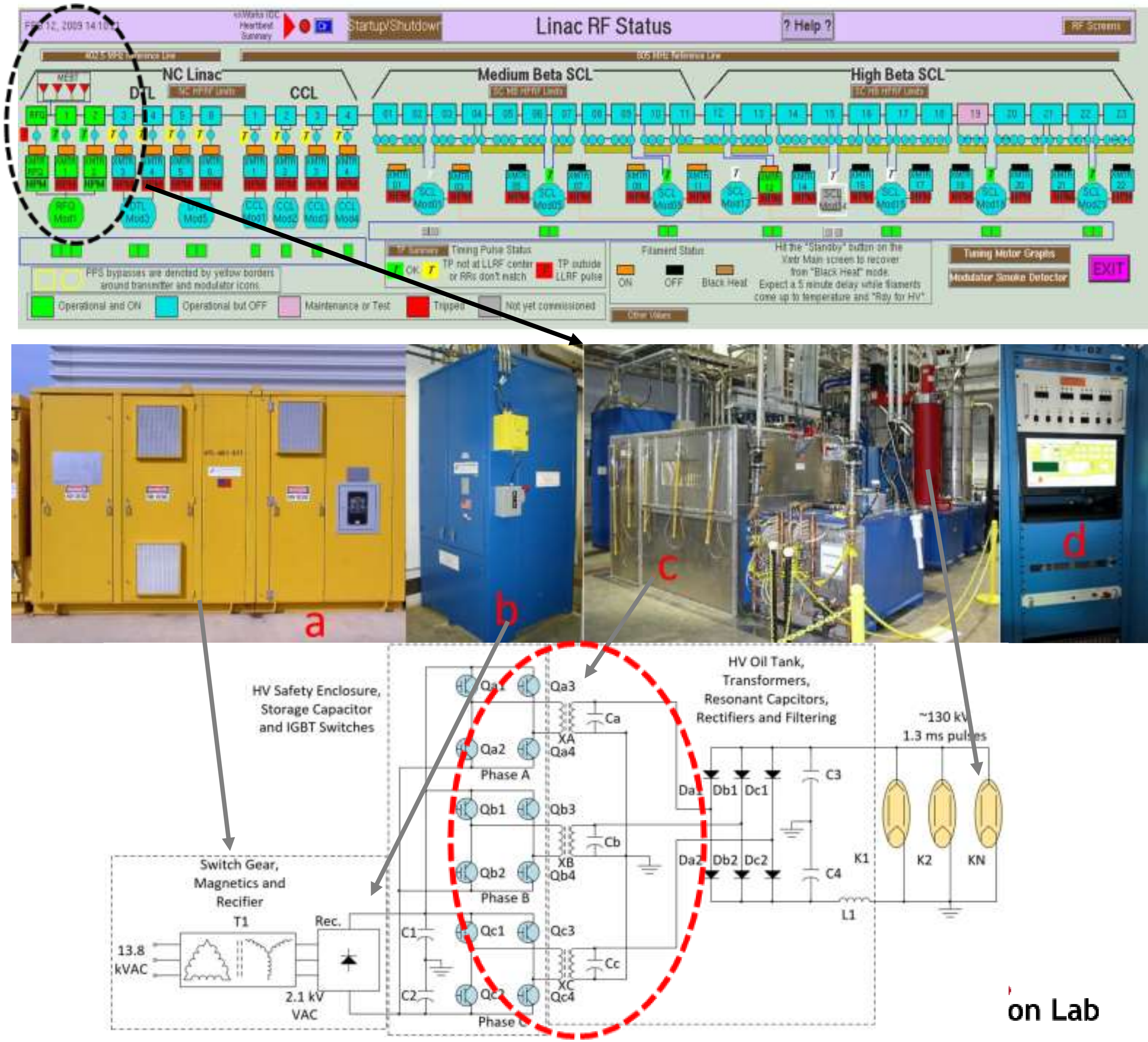
OAK RIDGE National Laboratory | SPALLATION NEUTRON SOURCE

Jefferson Lab

High Voltage Converter Modulators

HVCM Issue:

- Capacitor degradation during the pulse time causes anomalies in the signals, that could potentially lead to catastrophic failure.



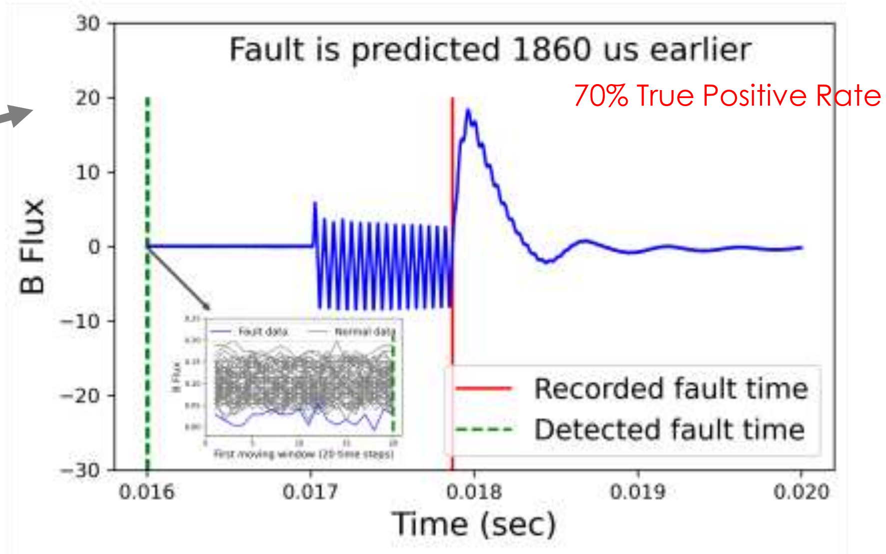
OAK RIDGE National Laboratory | SPALLATION NEUTRON SOURCE

on Lab

HVCM

- Research: How to minimize downtime due to the modulator
- Approach:
  - Abort beam before failure
  - Prognostics: predict component health. Capacitors slowly drop in capacitance over a periods of years, then fail suddenly
- Status:
  - Initial ML NN predicted HVCM failure
    - But we had a high FPR >10% → promising there is info in the waveforms
  - SPICE model of HVCM to research effect of capacitor values on measured waveforms
  - Second approach with LSTM and Conv1D

Transistor failure due to transformer saturation

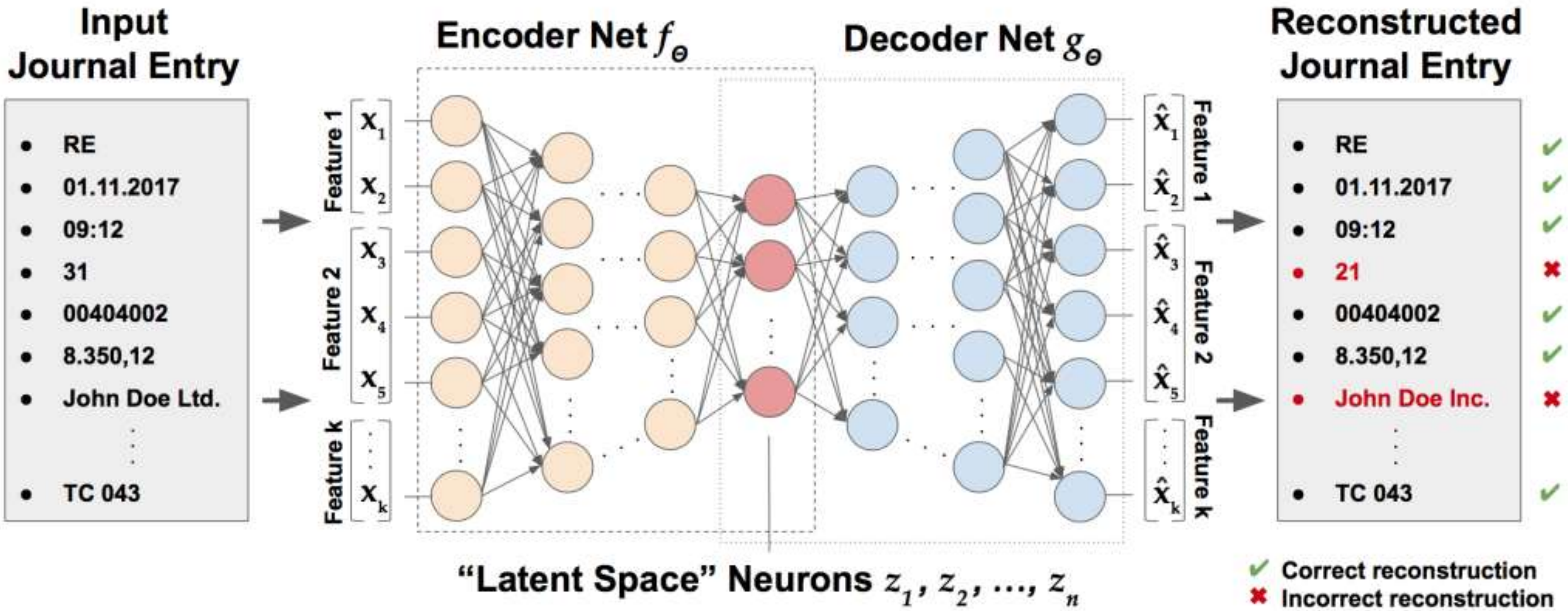


Failure prediction

HVCM

ML Technique: Self-constructors:

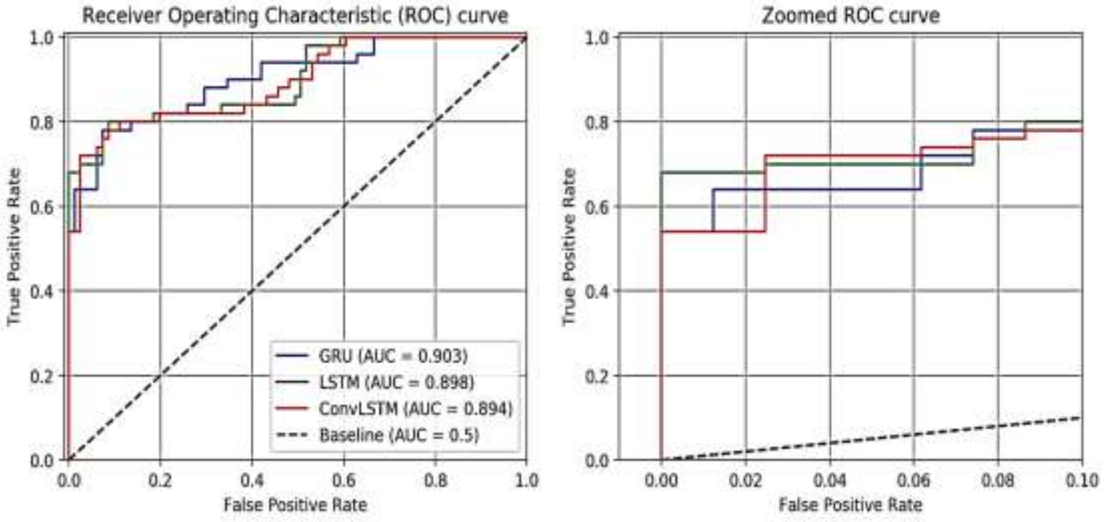
- Used mainly for dimensionality reduction, image noise removal, and anomaly detection (or binary classification). Latent space represents most important features. One type is the auto-encoder.

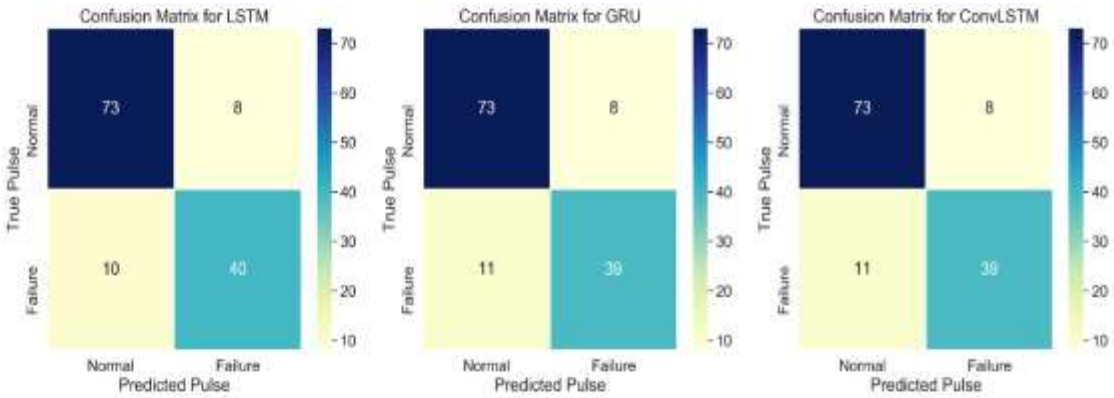


HVCM

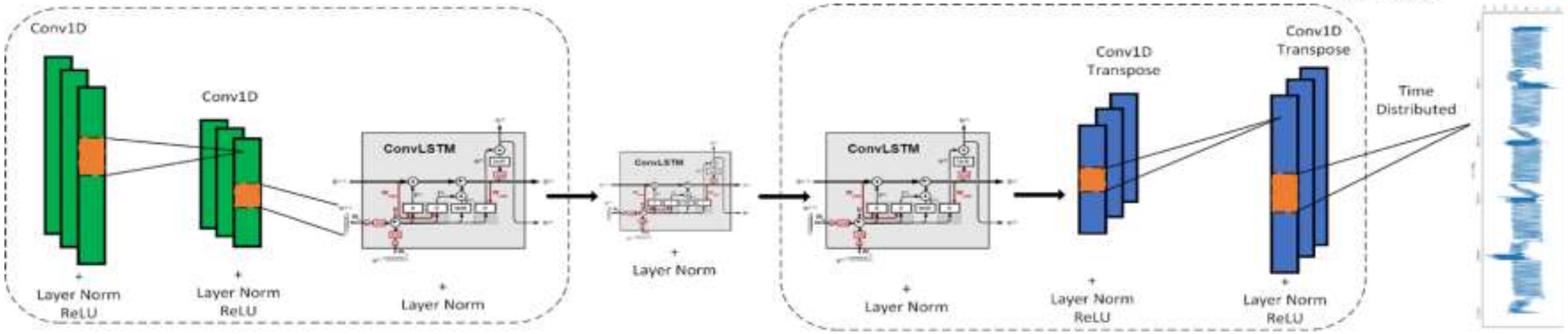
- Recurrent neural networks perform well in time sequences.
- Train on normal data to make it reproduce normal data. If the output waveform is not close to the input waveform, then we have an anomalous waveform.
- Conv1D will help to improve the latent space features.
- LSTM (Long Short Term Memory) will properly capture the time-series dynamics.

→ Improved FPR but need more statistics





Improved auto-encoder layout

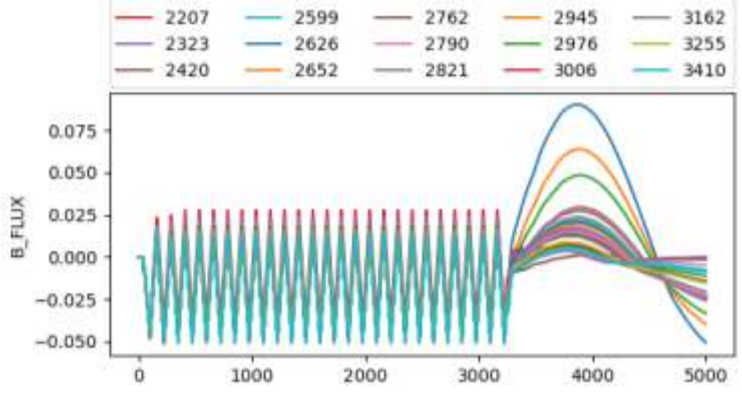


Radaideh, M. I., et al. "Time Series Anomaly Detection in Power Electronics Signals with Recurrent and ConvLSTM Autoencoders." *Digital Signal Processing* (2022): Under Review.https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4069225

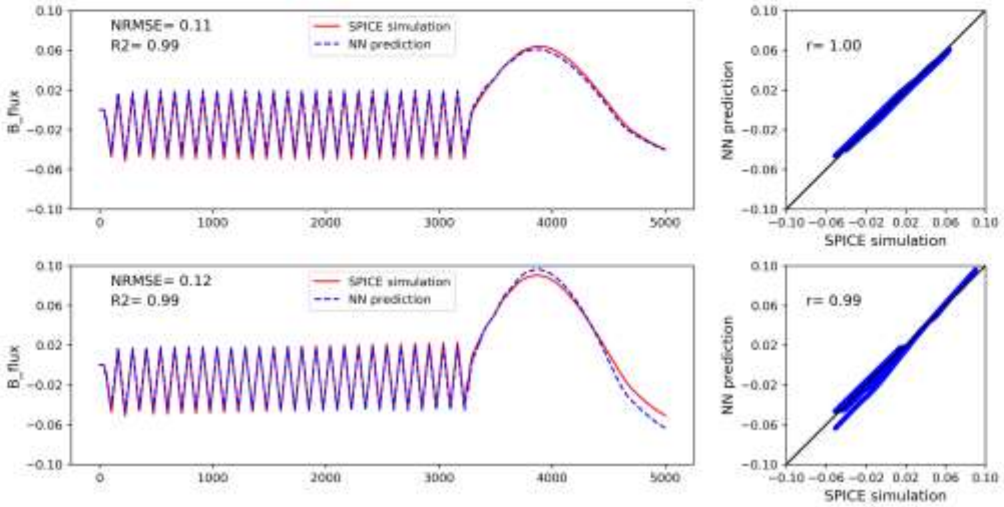
Jefferson Lab

HVCM: Prognostics

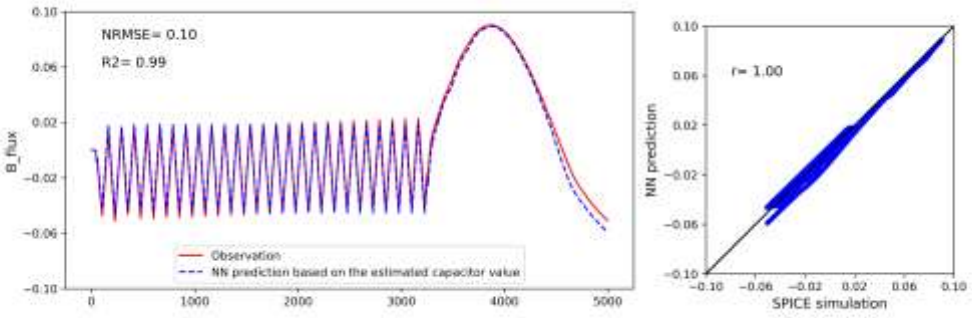
Neural Net modeling of waveform:



Step 1: Generate SPICE simulation data



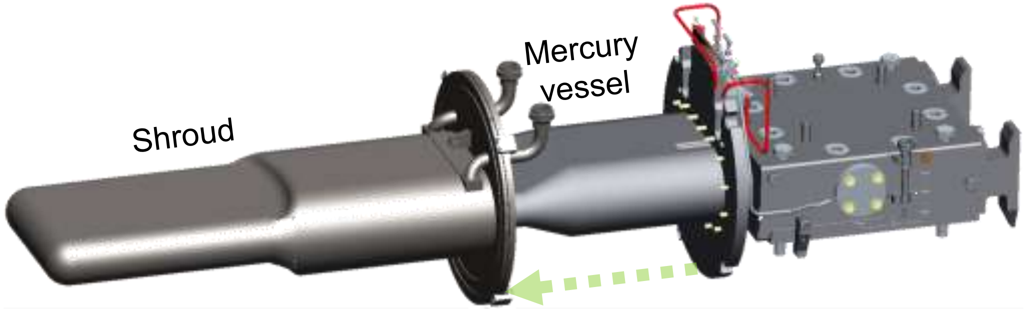
Step 2: Training: Neural Network (NN) learns the relationship between capacitor values and waveforms



Step 3: Testing: determine component value. E.g. simulated waveform capacitance estimate of 1609 pF versus 1550 pF.

- Plan:
  - Determine effect of other circuit parameters: charge voltage, switching frequency and the transformer leakage inductances. This is where we expect ML to show its strengths.

Target Machine Learning

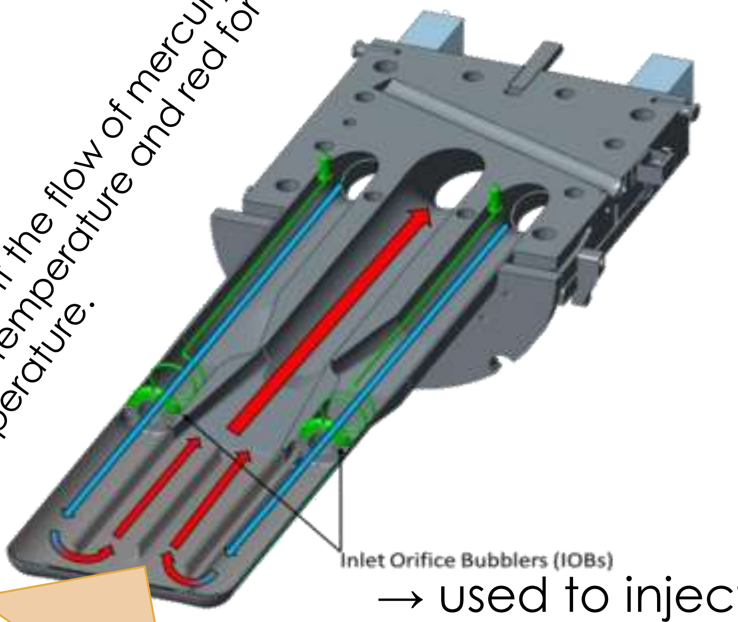


- Research: How to increase target lifetime
- Approach:
  - Use surrogate model to get faster simulation
    - Develop multi-phase physics model for mercury with gas bubbles
    - Match strain measurements to verify the simulation based on model(Sierra with VUMAT)
    - Train ML surrogate using polynomial approximations
- Status:
  - Using HPC resources to execution model-based simulation and train surrogate
    - Multiple different surrogate models are tested to identify the best metric and best model for the problems and design parameters

Target: Mercury Vessel

Front sensors die after few days of operation due to radiation damage reducing our diagnostics data

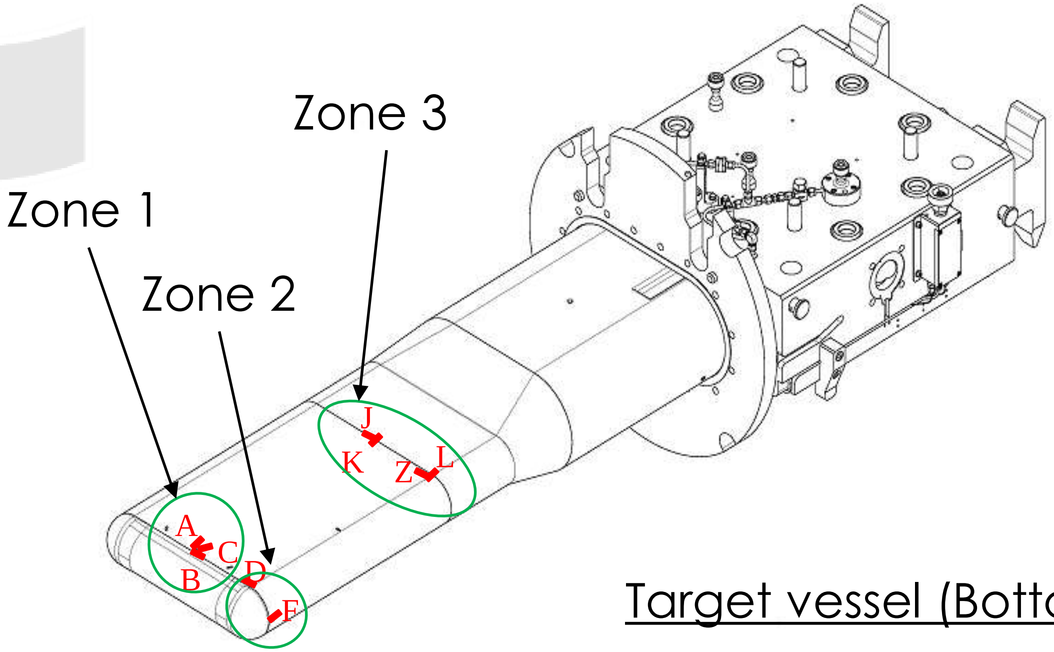
Arrows represent the flow of mercury, blue for low temperature and red for high temperature.



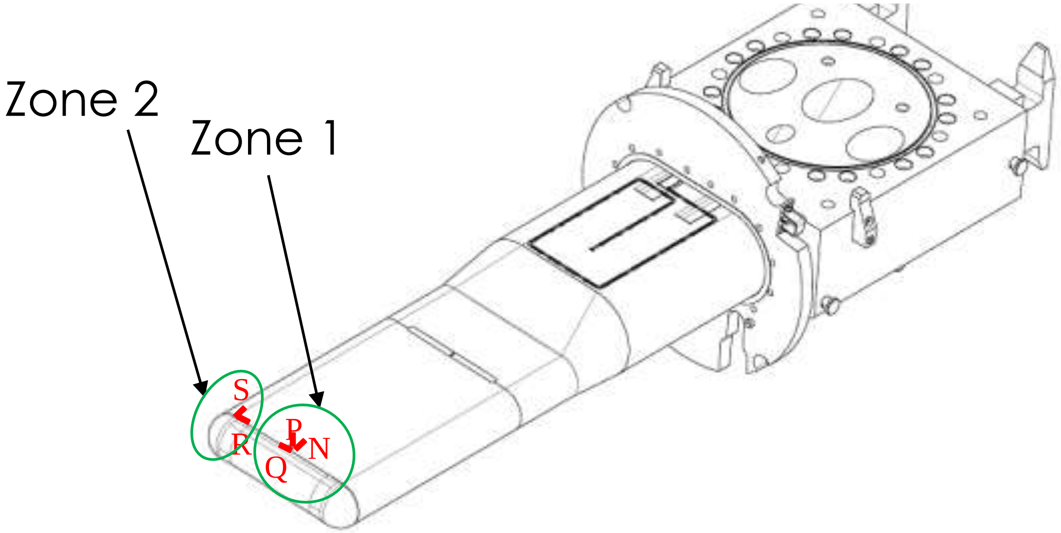
Proton pulse

→ used to inject helium gas to reduce pressure and cavitation damage

Target vessel (Top)



Target vessel (Bottom)

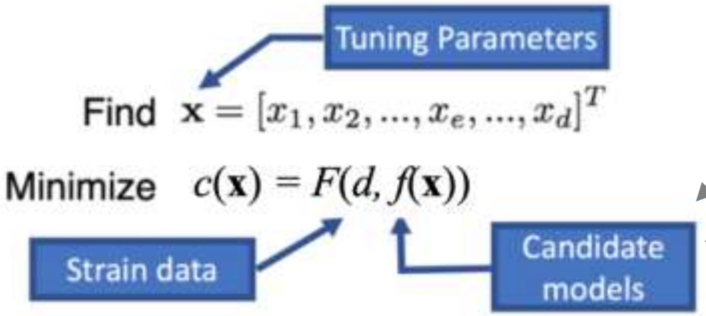


Target: Inverse Problem

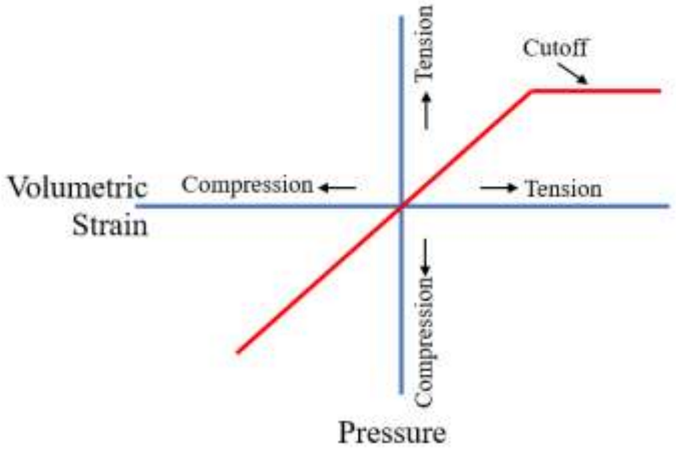
*We can use an accurate calibrated simulation to carry fatigue analysis and estimate target life and maintenance times**

*Mach, Justin, et al. "Fatigue analysis of the Spallation Neutron Source 2 MW target design." *Nuclear Instruments and Methods Section A* 1010 (2021): 165481.

Inverse Problem: find the model parameters (x) to minimize the difference between the measurements and the model



- ❑ **Now:** Equation of State Model for cavitation in mercury (3 unknown parameters)
- ❑ **Future:** Rayleigh-Plesset Model for general bubble dynamics (**8** parameters)



Initial focus on the 3-parameter model[&]

x_1: Tensile cutoff threshold (Pa)
x_2: Mercury Density (kg/m3)
x_3: Mercury Speed of Sound (m/s)

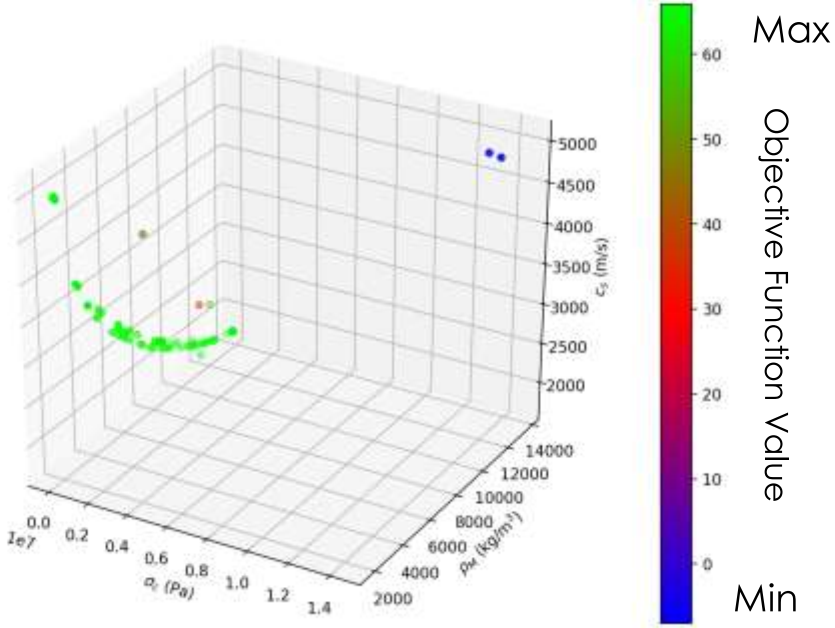
[&]Radaideh, M. I., et al. "Bayesian Inverse Uncertainty Quantification of the Physical Model Parameters for the Spallation Neutron Source First Target Station". https://arxiv.org/abs/2202.03959, *Accepted in Results in Physics*

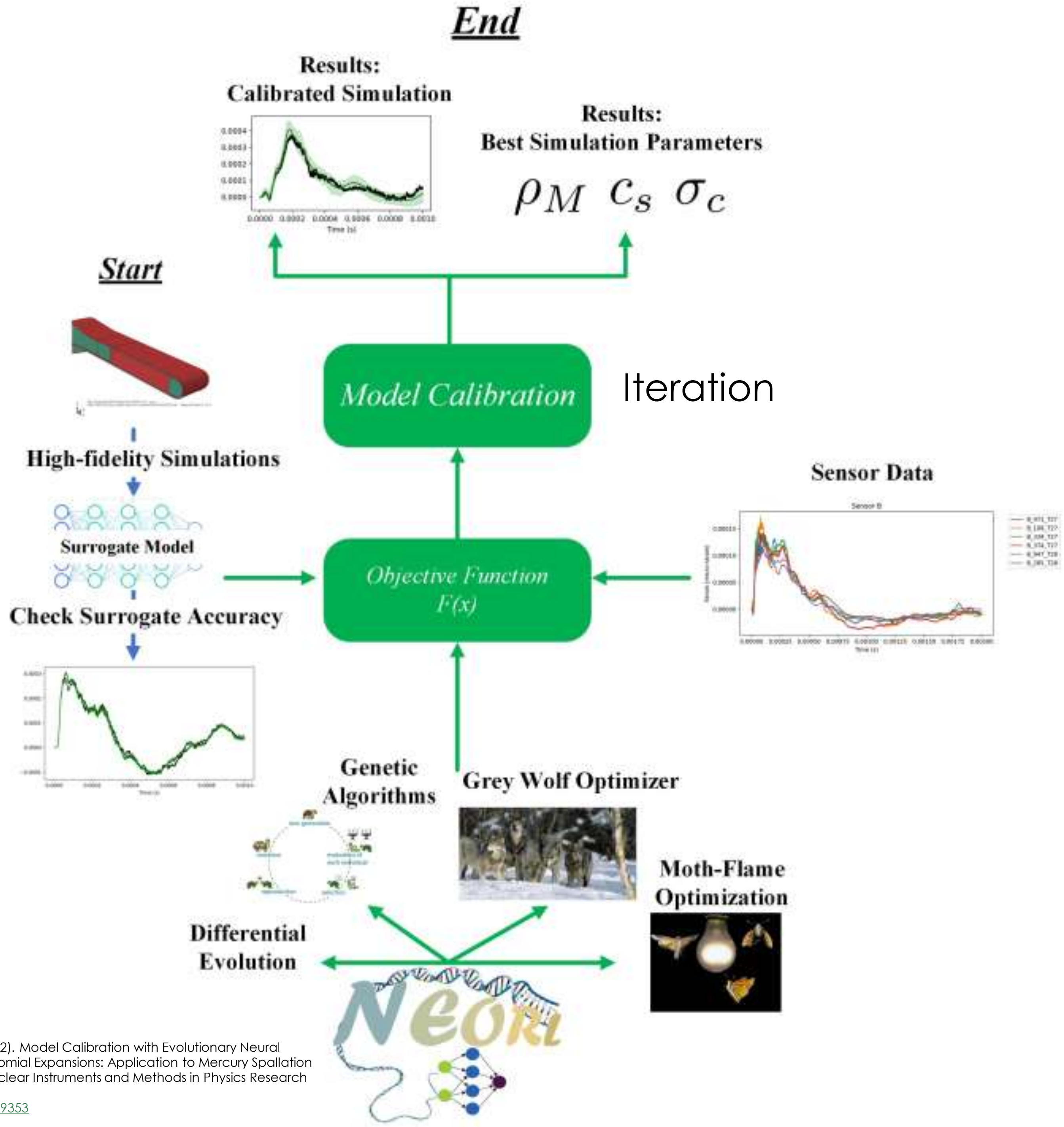
Target: Surrogate Model*

The method has four major parts:

1. Neural networks act as surrogate model to replace the expensive Sierra code.
2. Sensor data collected from the target.
3. External optimization algorithm (e.g. genetic algorithms).
4. Objective function brings 1-3 together.

   The objective function applies to the simulation parameters that make simulation and data close.



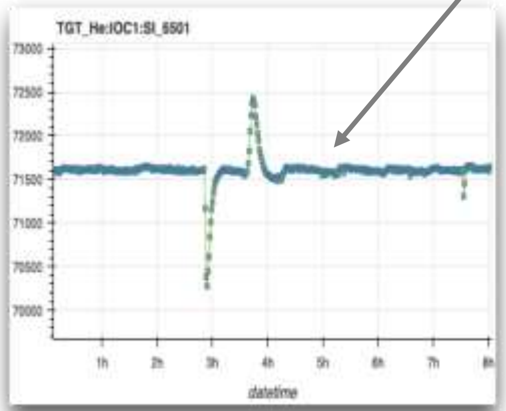


*Radaideh, M. I., et al., (2022). Model Calibration with Evolutionary Neural Networks and Sparse Polynomial Expansions: Application to Mercury Spallation Target Solid Mechanics, Nuclear Instruments and Methods in Physics Research Section B, Under Review. https://arxiv.org/abs/2202.09353

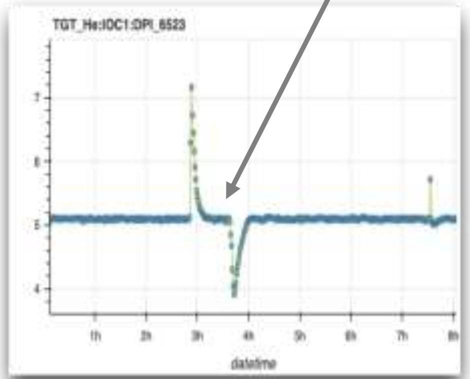
Cryogenic Moderator System

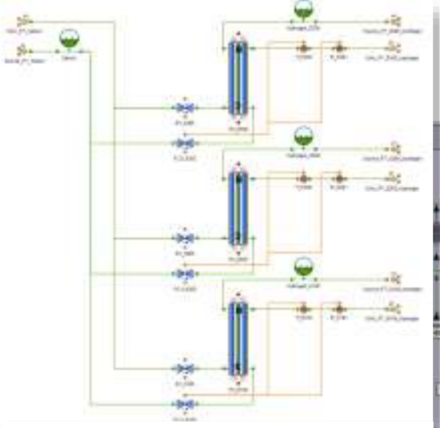
- Research: how to avoid long CMS trips
- Approach:
  - Not a lot of data to apply ML for predictions
    - Use simulation to generate data
  - Improve whole system modeling by combination of model and data-driven ML techniques
  - ML-based controller
- Status:
  - Building of the CMS model

Excessive turbine fluctuation



Long system recovery time

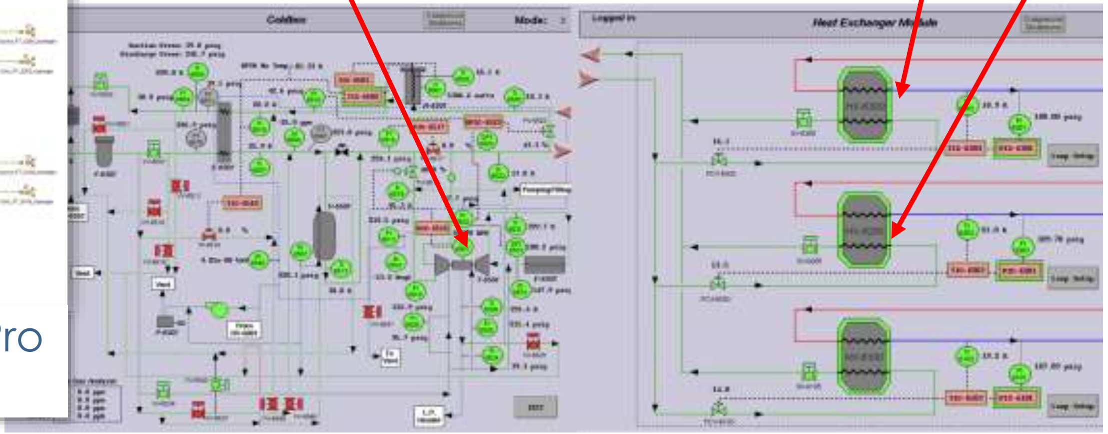




H2 loop EcoSimPro model

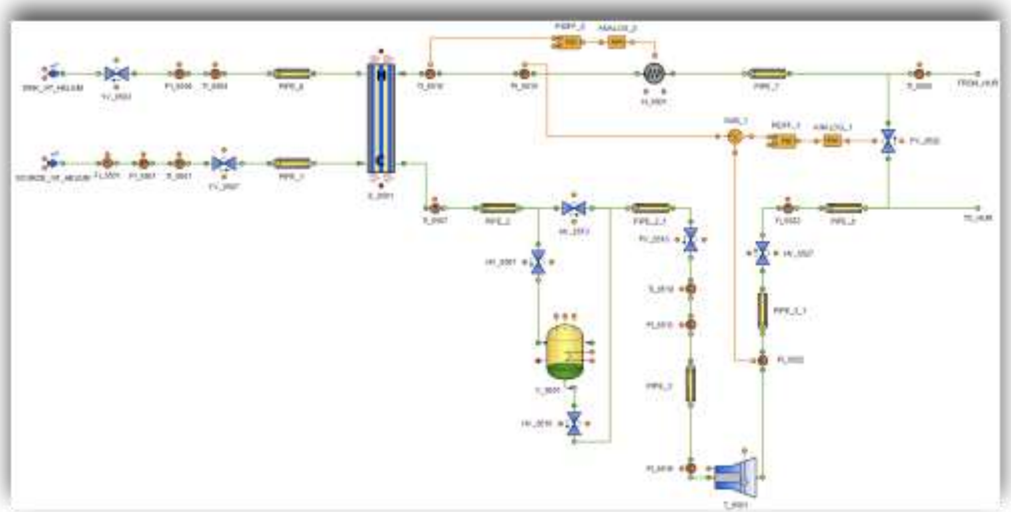
turbine in the He loop

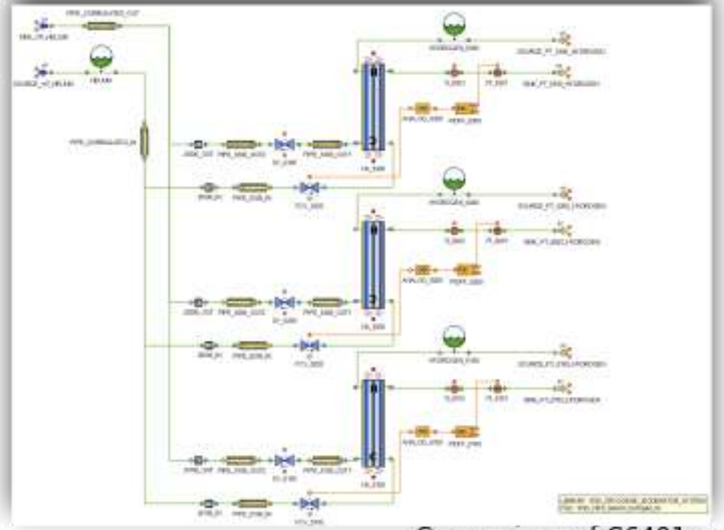
H2 loops where trips register



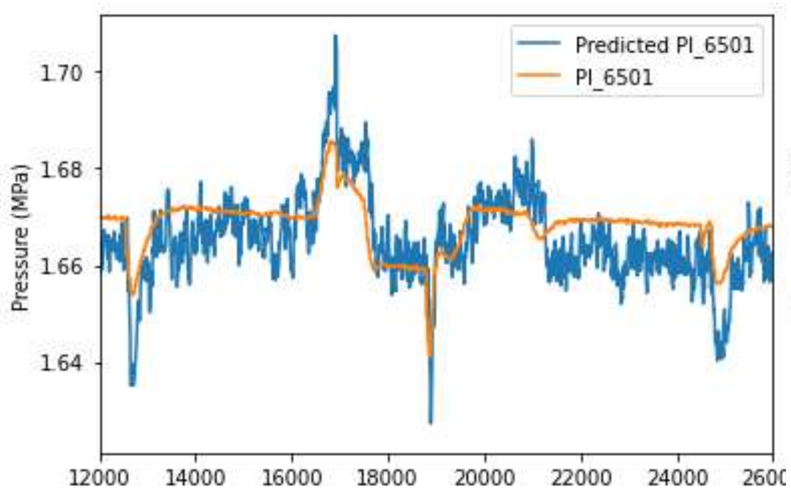
System layout

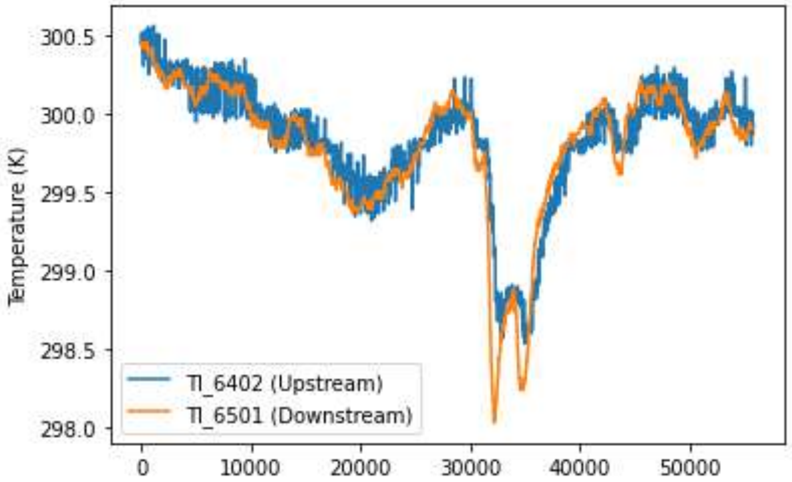
CMS: thermal-hydraulic and data-driven models

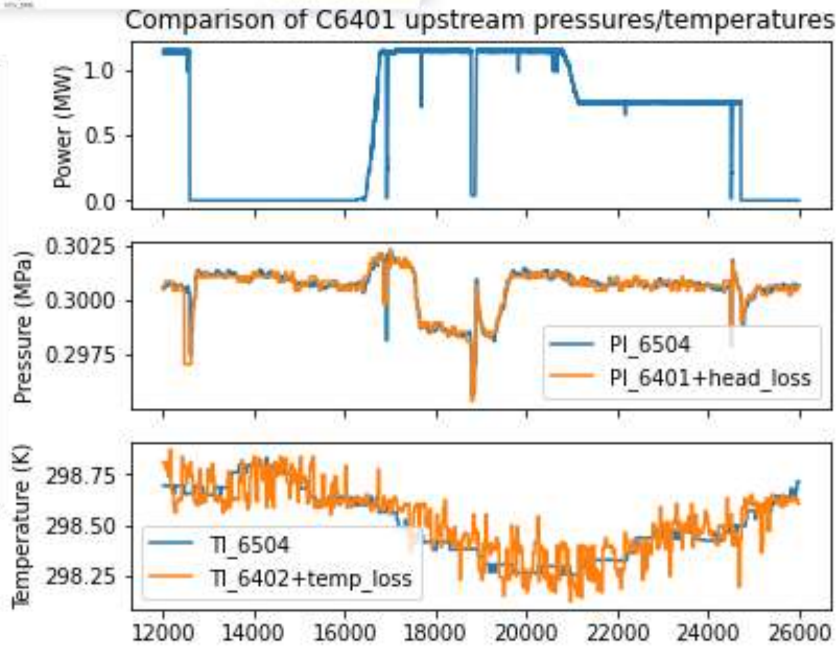




EcoSim Simulation







Summary

We presented four ML use-cases:

- Classification, Prevention, Surrogate, Prognostics, Controller
  - All using existing data and systems
- Three use-cases significantly invested in physics modeling
  - Target is very difficult because of the complex simulation requiring HPC
  - Cryo system has many unknowns and not as much data as we like
- One use-case is entering operation in the field!
- We have setup collaboration tools and a data infrastructure to support data streaming and collection

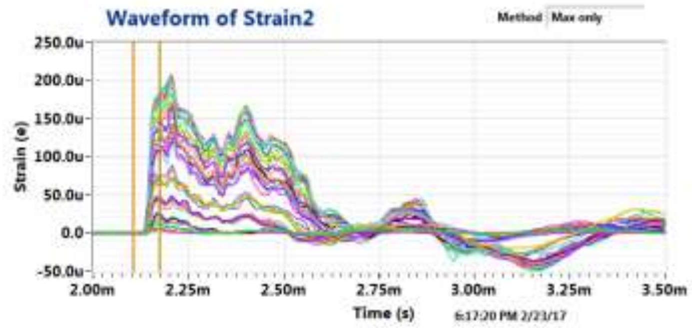
Thank you for your attention!

Questions?

Extra

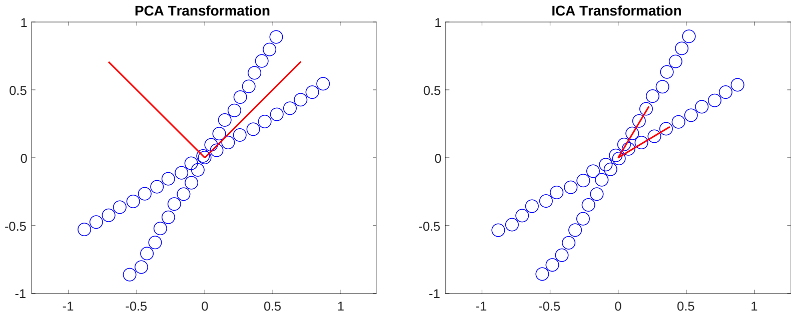
Scrum was held at SNS for the week Dec. 17 – 21, 2018

- Goal was to find physical insight in target strain sensor data using Data Analytics
  - Down select data, convert to ascii
  - Analyze each sensor vs beam power
  - Spatially identify on a target map sensor locations with nonlinear features.
  - Propose possible linkages associated with physics.



Mostly Linear strain vs beam power

One week in same room = all walls written on

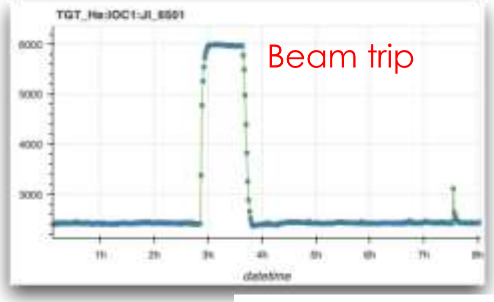


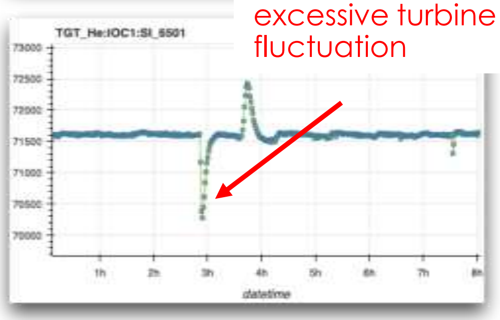
PCA and ICA transformations

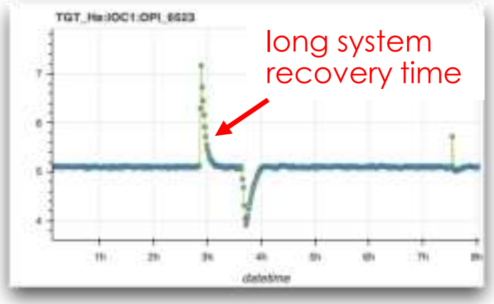
Better Control of Cryogenic Moderate System

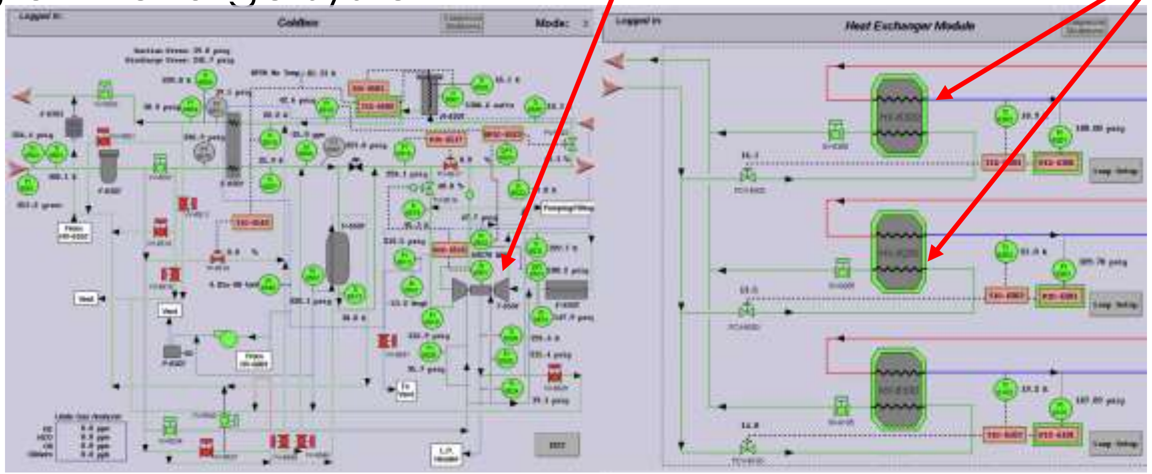
- Problem to solve: beam trips have long system recovery time and cause excessive turbine speed fluctuation

- Solution:
  - Improve overall system responses to the beam trips by better controller parameters and using meta-control methods
    - Better whole system modeling by combination of mechanistic model and data-driven ML techniques
    - ML-based control and/or meta-control methods
    - Address inherent uncertainties by UQ techniques
    - Off-line testing of the large system









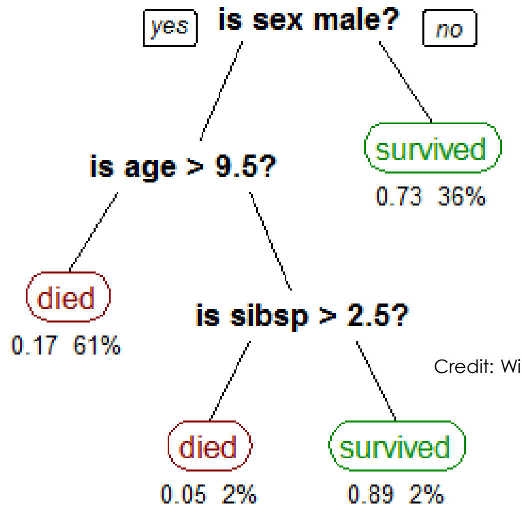
Status

- Curating existing PV data from operation:
  - Finished the python library to extract key PVs related to the specific beam trip windows
  - Interactively interrogate and compare the system dynamics
- Building of the mechanistic CMS model
  - Secured software license of CRYOLIB in EcoSimPro
  - Model construction currently underway
- System identification and ML-based modeling to better understand system dynamics
  - To better understand the dynamic behavior of the systems from the PVs
- CMS-specific diagnostic run during October study time
  - Requested study time to run CMS-specific diagnostics by placing CMS under different load conditions and excite "step responses"

Decision Tree

True Positives Results

Dataset	K-NN True Pos (var)	Decision Tree True Pos (var)
A upstream	0.60 (0.04)	0.58 (0.02)
A downstream	0.52 (0.03)	0.55 (0.04)
B upstream	0.74 (0.04)	0.63 (0.06)
B downstream	0.69 (0.05)	0.53 (0.05)
C upstream	0.70 (0.03)	0.72 (0.03)
C downstream	0.54 (0.02)	0.65 (0.02)
D upstream	0.76 (0.03)	0.73 (0.01)
D downstream	0.55 (0.02)	0.60 (0.01)



Credit: Wikipedia

Decision trees (DT) are a non-parametric supervised learning method used for classification.

Classification is performed by traversing a tree-like structure created in the learning process

- Initial results of decision tree are not better

Errant Beam Analysis

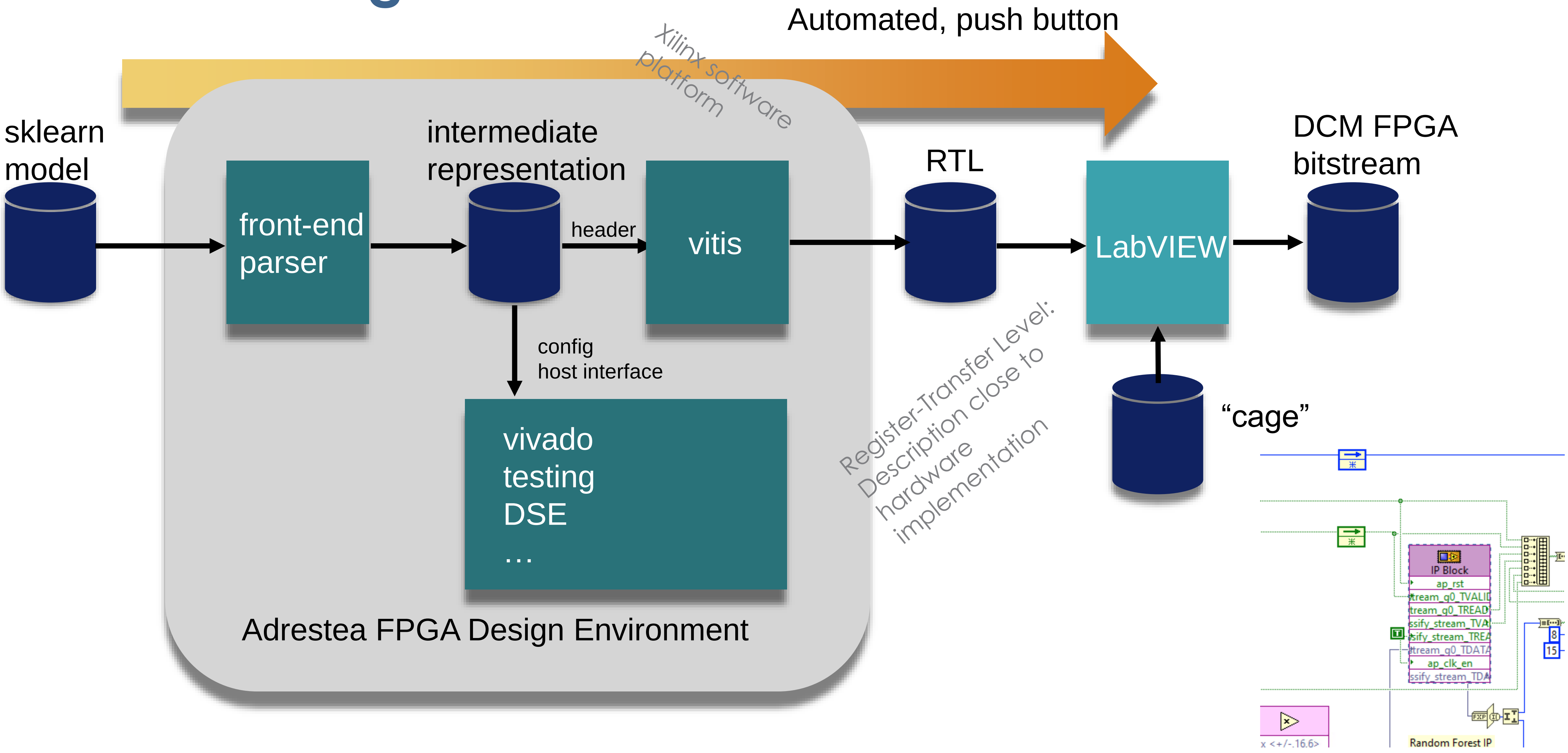
University of Huddersfield
International Institute for Accelerator Applications

Collaboration with Miha Reščič (PhD student with Rebecca Seviour)

- Subset of available data 2015 selected
- Initial non-ML techniques applied
  - Hidden Markov Models, Electronics Network Frequency Criterion
    - No promising results
- Initial ML Methods (supervised)
  - K-Nearest Neighbors algorithm
  - Decision Trees
  - Naïve Bayes (no promising results)
  - Preprocessing:
    - Principle Component Analysis
    - Feature extraction to compose datasets

Data set	Samples	# pulses	Dates
A	25000	1540	2015-02-28, 2015-03-01, 2015-03-18, 2015-03-28, 2015-03-29, 2015-03-30
B	100000	490	2015-05-05, 2015-05-28
C	25000	1400	2015-08-25
D	25000	2640	2015-10-25, 2015-10-26, 2015-10-27 2015-10-28, 2015-10-29

RF FPGA Design Flow



Extreme Low-Latency FPGA Design

- Unique design features
  - Deep optimization based on the tree structures
  - Pre-compiled vote counter logic
  - Streaming input data
- Benefits
  - Area efficiency: <u>10X reduction</u>
  - Extremely low latency: <u>3X reduction of worst-case latency, early decision possible</u>