Error Detection in Linacs with Machine Learning Techniques

Jonathan Edelen, Kathryn Wolfinger, Matt Kilpatrick, Joshua Einstein-Curtis (RadiaSoft),

Mike Wallbank, Dean Edstrom, Jonathan Jarvis (Fermilab)

March 14th 2024

FAST/IOTA Collaboration Meeting

radiasoft

Fermilab

Batavia IL, USA | radiasoft.net

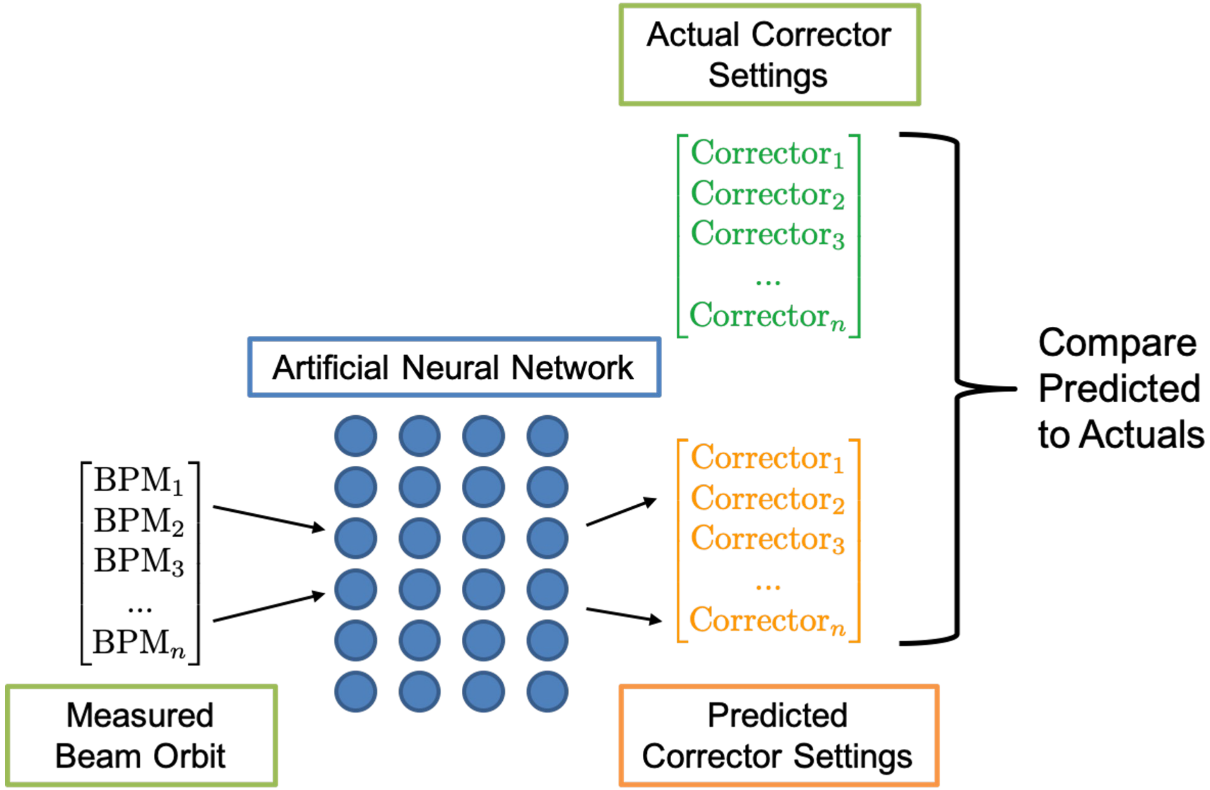
U.S. DEPARTMENT OF ENERGY | Office of Science

Outline

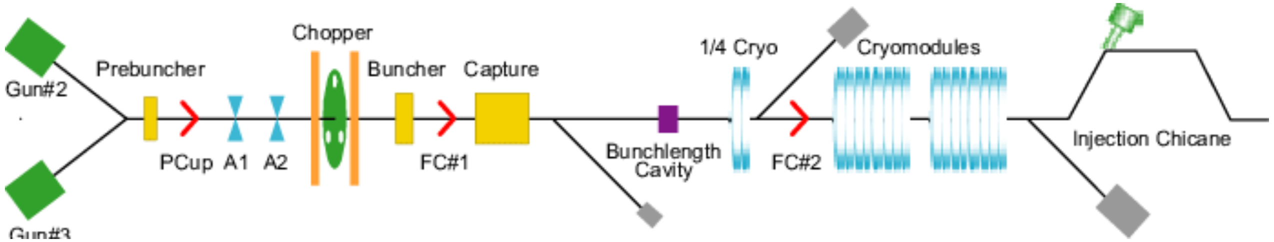
- Machine learning methods
- Initial demonstration at Jlab
- Transitioning to FAST
- Initial results
- Conclusions

Inverse models as a diagnostic tool

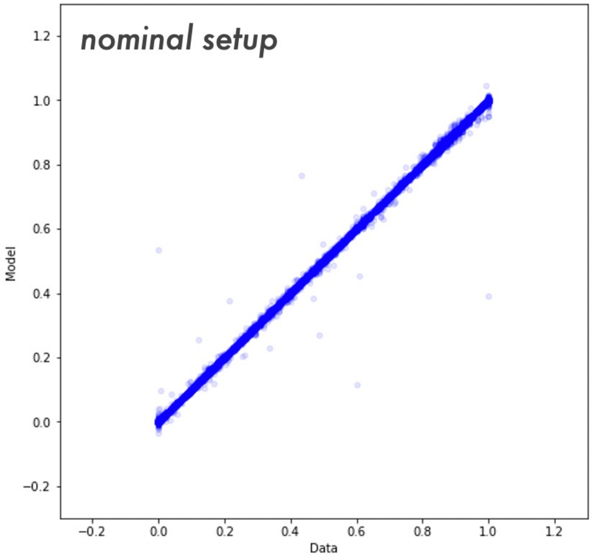
- Inverse models as a diagnostic in a supervised fashion
  - Direct comparison between predicted settings and actual settings informs operations of a potential anomaly
- Inverse models as a diagnostic in an unsupervised fashion
  - Assumptions
    - model errors are caused by other beamline elements
    - each beam-line element will have a unique error signature
- Inverse models for tuning
  - Minimize error between predicted settings and actual settings by varying quads
  - Right: model error as a function of quad strength error

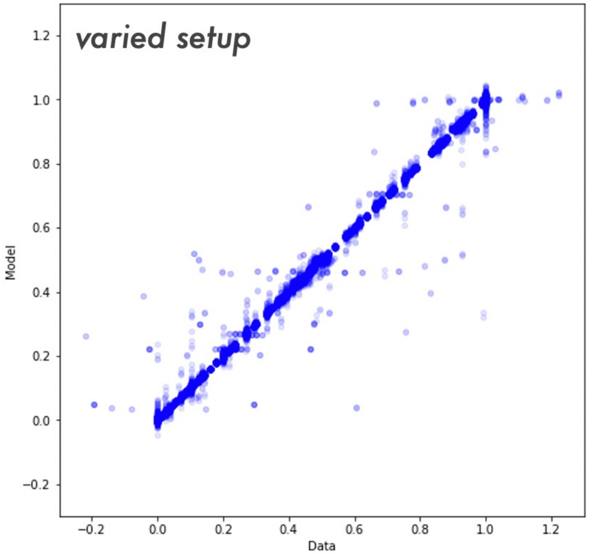


Inverse Model Anomaly Detection at CEBAF



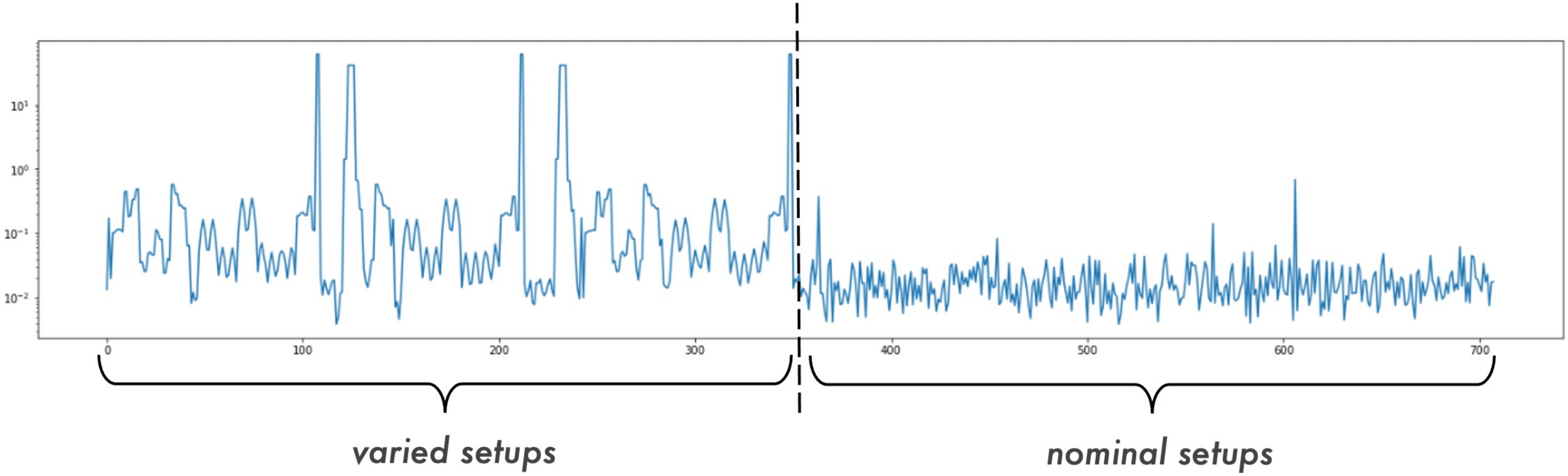
- Data collected during two different operational modes.
  - During normal operations
  - During a dedicated machine study
- Inverse model trained to predict settings from readings
  - Left: Model prediction vs the ground truth for the validation data from the nominal setup
  - Right: Model prediction vs the ground truth for the test data





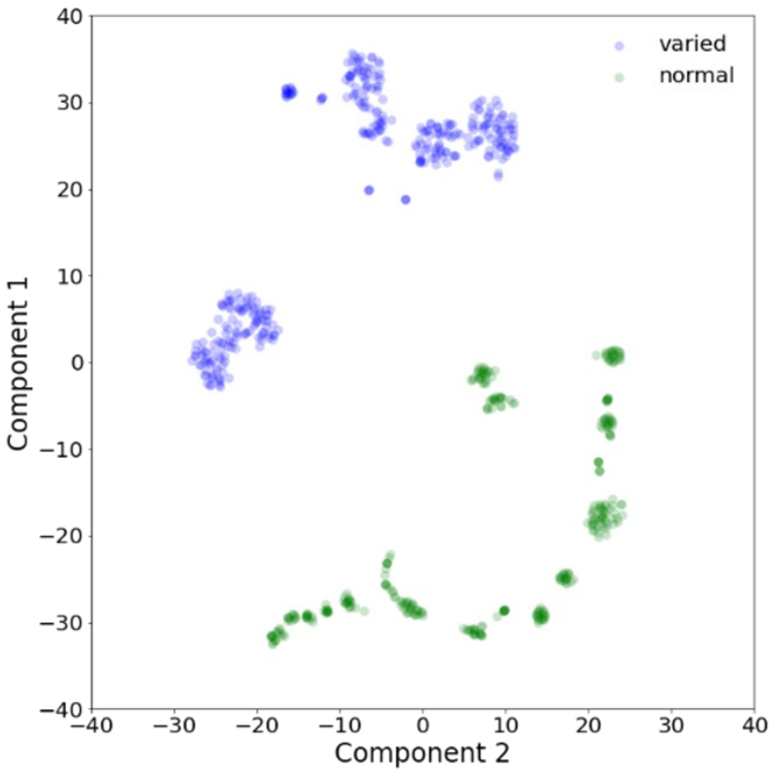
Establishing error thresholds

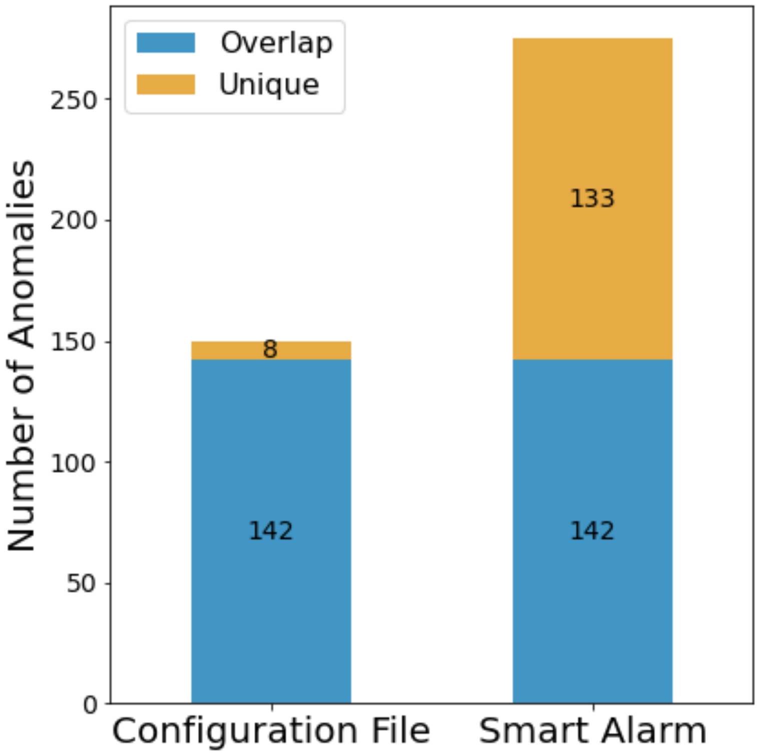
- RMS error of the predicted settings by parameter for the machine study (left) and the nominal setup (right).



A Smart Alarm System for the CEBAF Injector

- Left: T-SNE was used to reduce the dataset dimensionality
  - Operational data is shown in green and the study data in blue
  - The model correctly flagged the study data as anomalous
  - The T-SNE reduction of the data also provides a strong indication that these two datasets are distinct in nature
- Right: Comparison with conventional threshold-based alarming.
  - Threshold misses numerous configurations that would be undesirable by the user program





Transition to the FAST LINAC

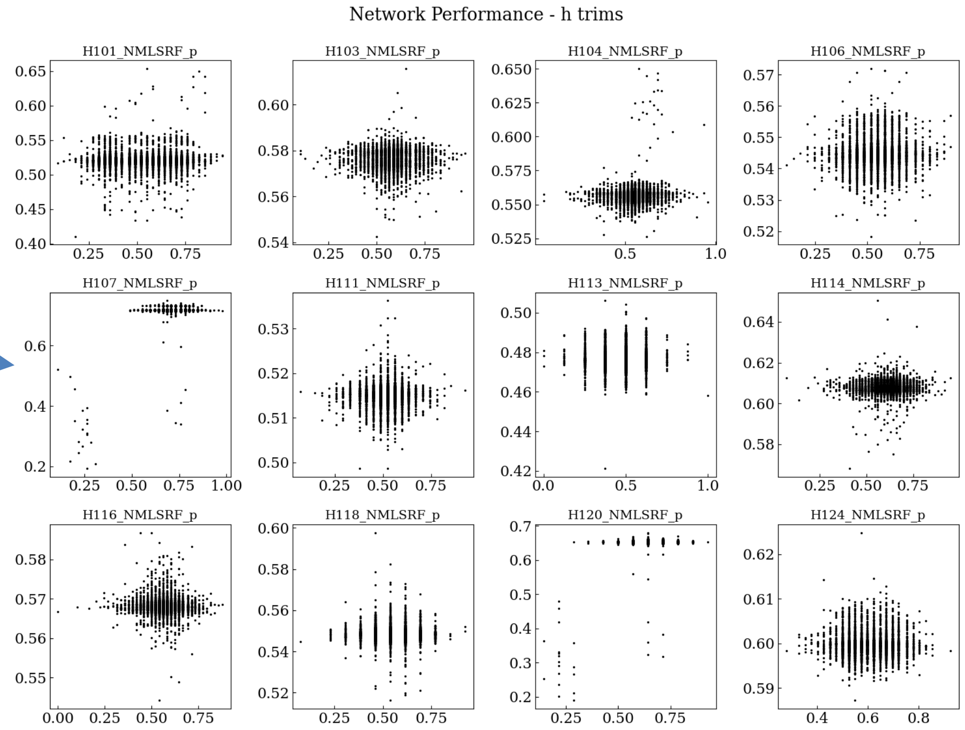
- Objective(s):
  - Near term: Develop model that can effectively detect anomalies in an explainable fashion → 60% complete
  - Longer term: Utilize anomaly detection tools to assist in automatic tuning of the machine

- Experimental plan:
  - Collect data during "normal" operational conditions
  - Train machine learning model on data archive
  - Test machine learning on study data where parameters are intentionally varied
  - Evaluate effectiveness at detecting anomalies
  - Develop uncertainty metrics for machine learning model
  - Deploy and test software during experimental run

Transition to the FAST LINAC

- Objective(s):
  - Near term: Develop model that can effectively detect anomalies in an explainable fashion → 60% complete
  - Longer term: Utilize anomaly detection tools to assist in automatic tuning of the machine

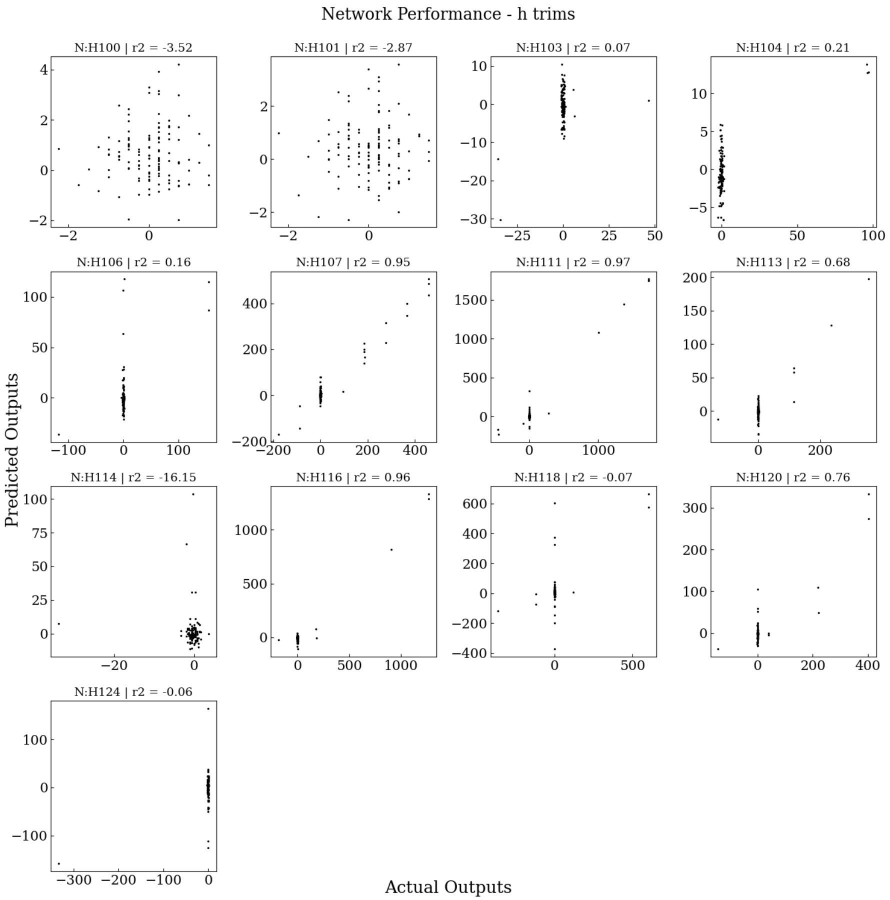
- What actually happened:
  - Collect data during "normal" operational conditions
  - Train machine learning model on data archive
    - Model learning not adequate
    - Troubles with data alignment
    - Identified potential logger issues



Transition to the FAST LINAC

- ## Objective(s):
  - Near term: Develop model that can effectively detect anomalies in an explainable fashion → 60% complete
  - Longer term: Utilize anomaly detection tools to assist in automatic tuning of the machine

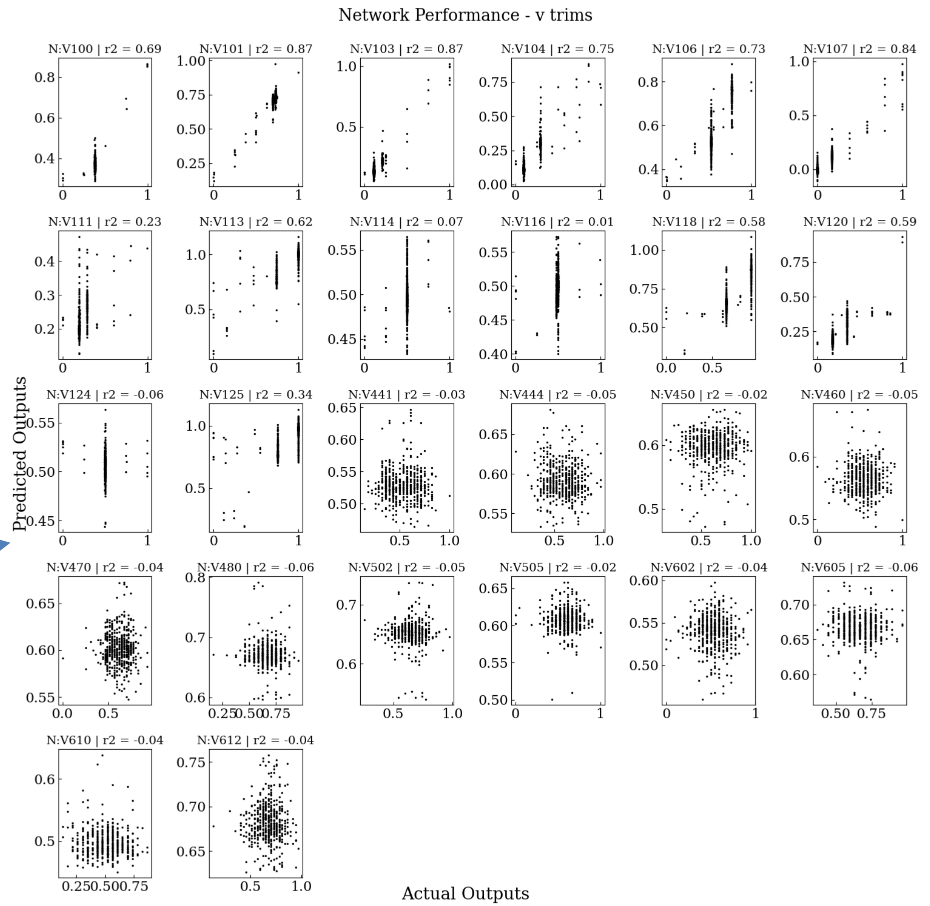
- ## What actually happened:
  - Collect data during "normal" operational conditions
  - Train machine learning model on data archive
    - Model learning not adequate
    - Troubles with data alignment
    - Identified potential logger issues
  - Collected study data for training
    - Model learning not adequate



Transition to the FAST LINAC

- Objective(s):
  - Near term: Develop model that can effectively detect anomalies in an explainable fashion → 60% complete
  - Longer term: Utilize anomaly detection tools to assist in automatic tuning of the machine

- What actually happened:
  - Collect data during "normal" operational conditions
  - Train machine learning model on data archive
    - Model learning not adequate
    - Troubles with data alignment
    - Identified potential logger issues
  - Collected study data for training
    - Model learning not adequate
  - Collected new study data for more training
    - Model learning not adequate



Transition to the FAST LINAC

- Objective(s):
  - Near term: Develop model that can effectively detect anomalies in an explainable fashion → 60% complete
  - Longer term: Utilize anomaly detection tools to assist in automatic tuning of the machine

- What actually happened:
  - Collect data during "normal" operational conditions
  - Train machine learning model on data archive
    - Model learning not adequate
    - Troubles with data alignment
    - Identified potential logger issues
  - Collected study data for training
    - Model learning not adequate
  - Collected new study data for more training
    - Model learning not adequate
  - Contemplate giving up
  - Try to see if anomaly detection is possible even with a ~~bad~~ sub optimal model

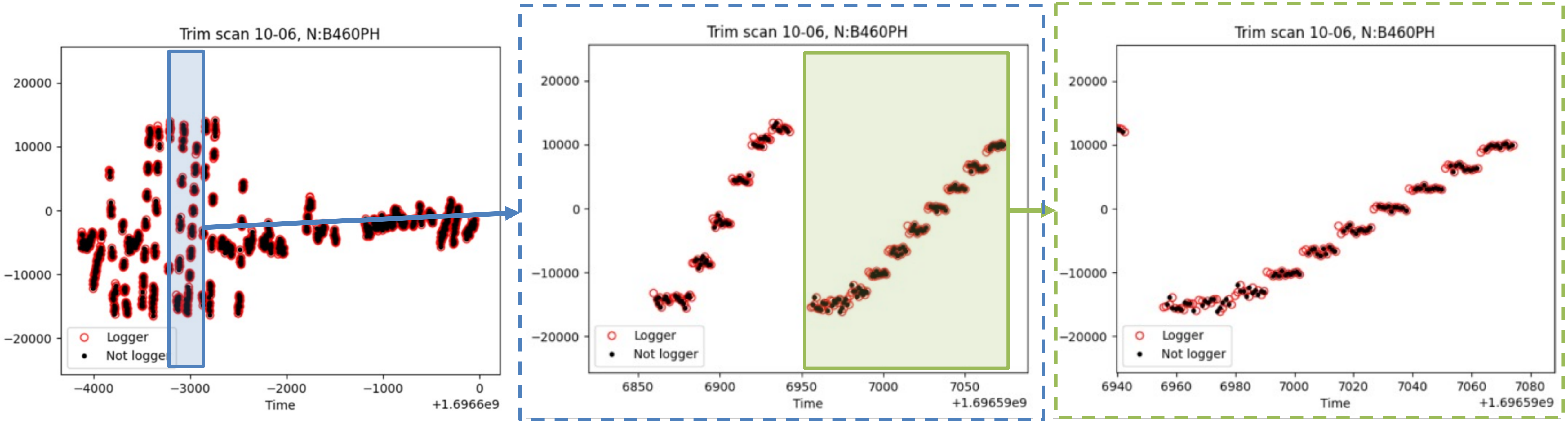
Transition to the FAST LINAC

- Objective(s):
  - Near term: Develop model that can effectively detect anomalies in an explainable fashion → 60% complete
  - Longer term: Utilize anomaly detection tools to assist in automatic tuning of the machine

- What actually happened:
  - Collect data during "normal" operational conditions
  - Train machine learning model on data archive
    - Model learning not adequate
    - Troubles with data alignment
    - Identified potential logger issues
  - Collected study data for training
    - Model learning not adequate
  - Collected new study data for more training
    - Model learning not adequate
  - Contemplate giving up
  - Try to see if anomaly detection is possible even with a ~~bad~~ sub-optimal model
  - Success! … sort of

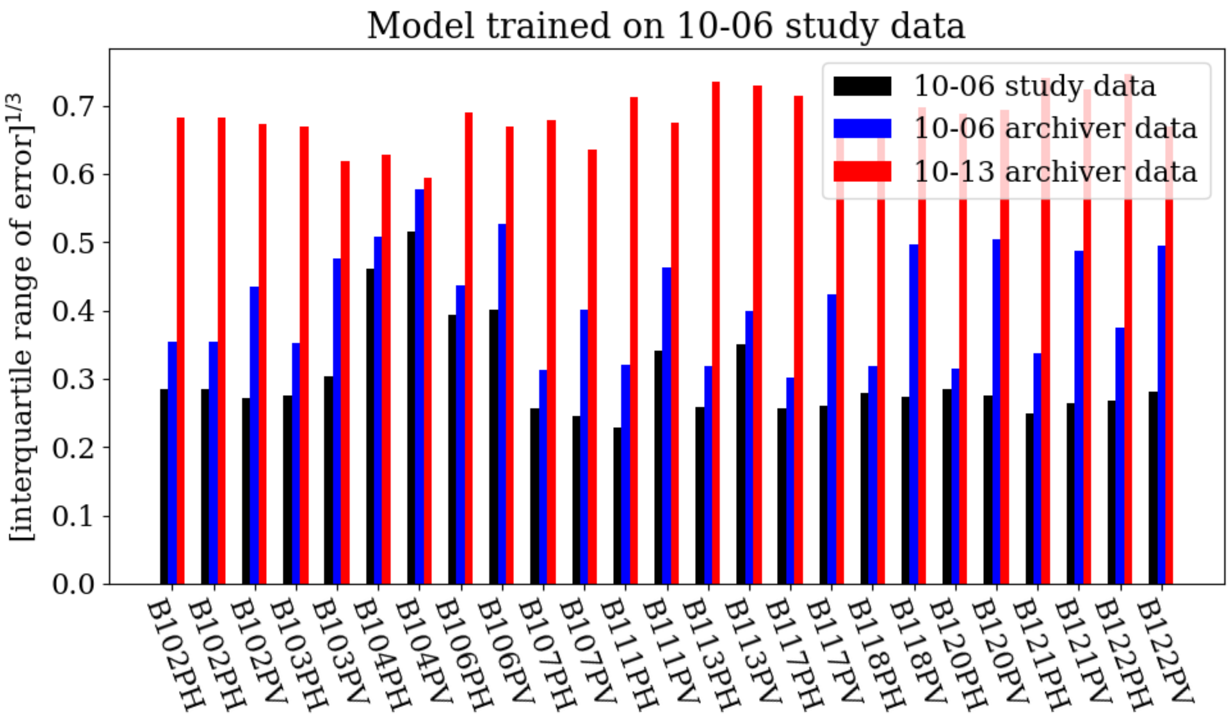
Identifying archiver issues

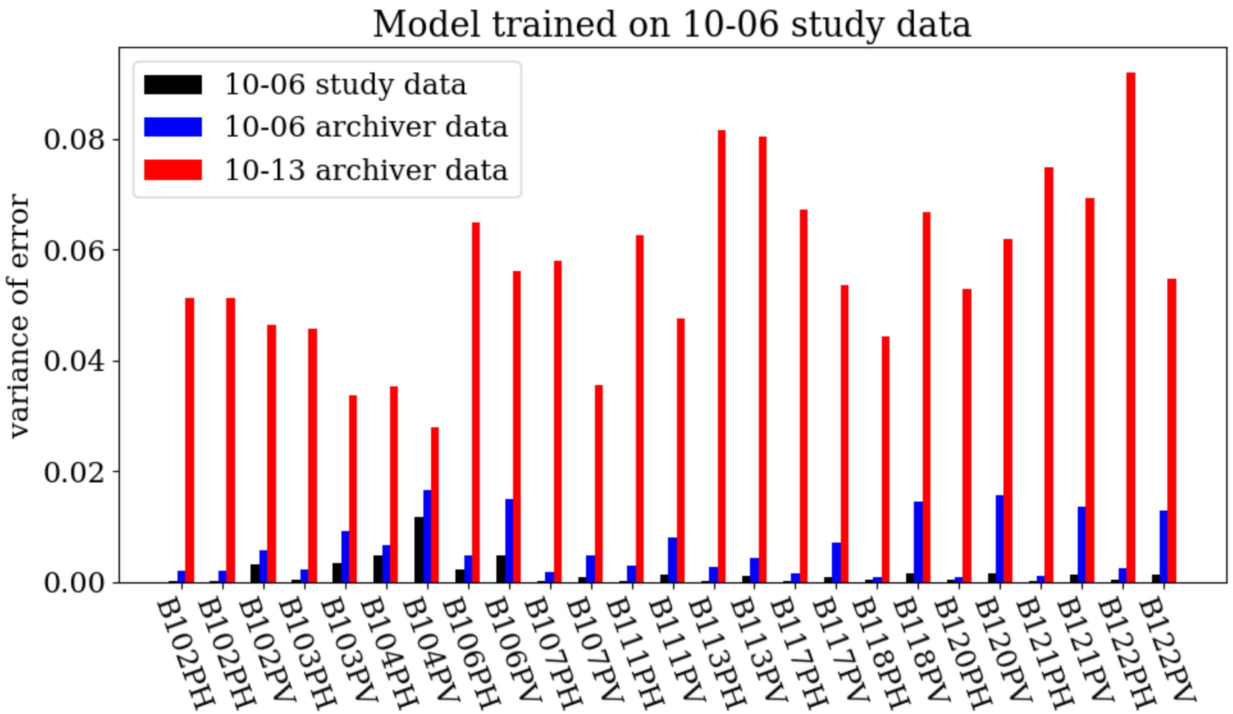
- Time alignment efforts with archive data revealed inconsistencies
  - Comparison of scripted data logging with archiver
  - Study on 10-6 scanned trims and recorded BPMs for model development and testing



Initial model training results

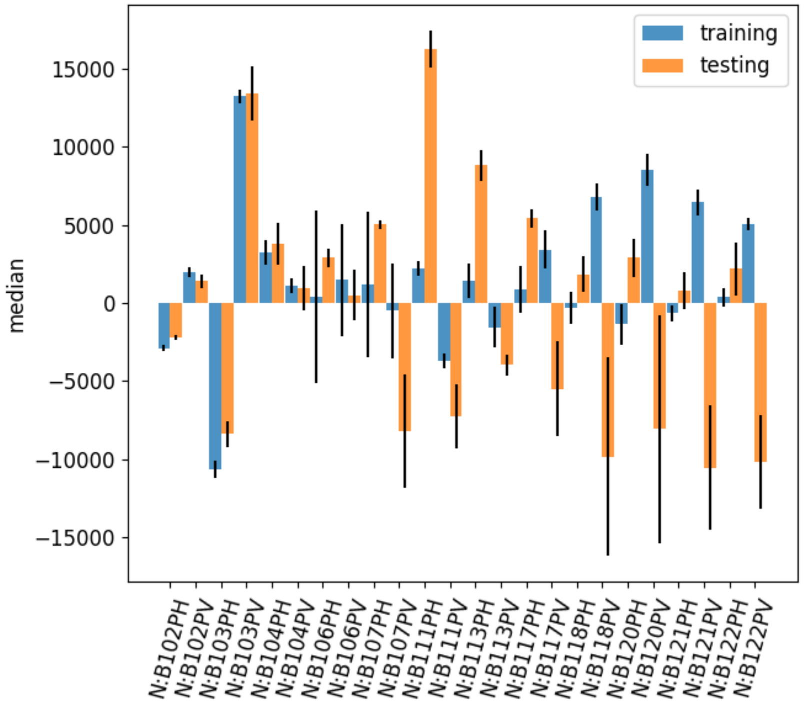
- We trained a model on only the 10-06 study data. Then, we applied that model to the archiver data and study data. The following plots show the error of the model predictions.
  - Note, we are considering an archiver timeframe on 10-06 when dedicated study data was collected and an archiver timeframe on 10-13 when no dedicated study was run.

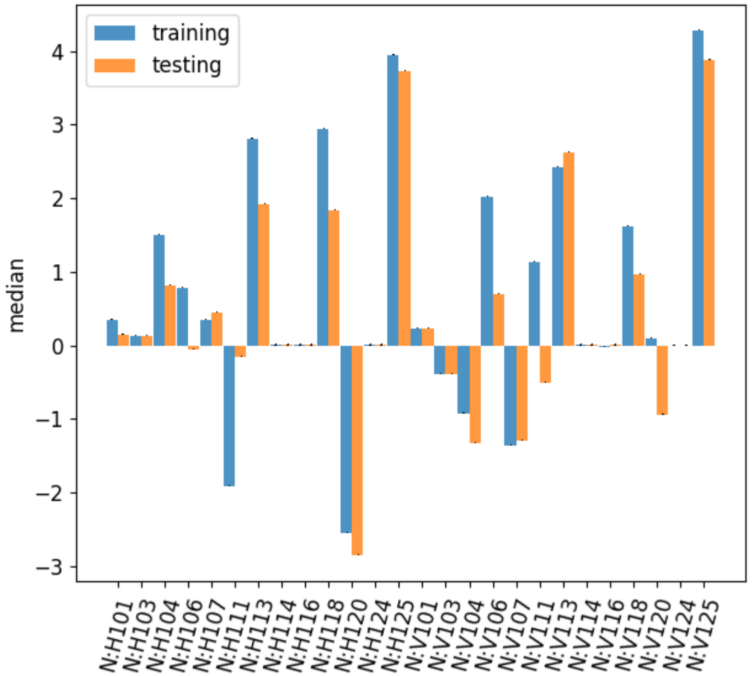


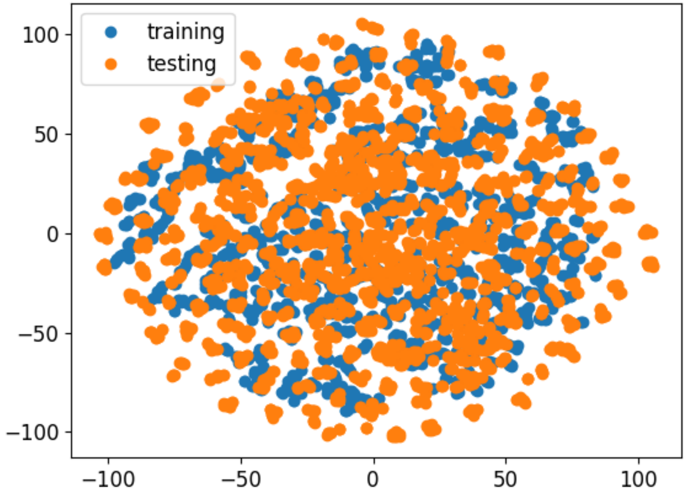


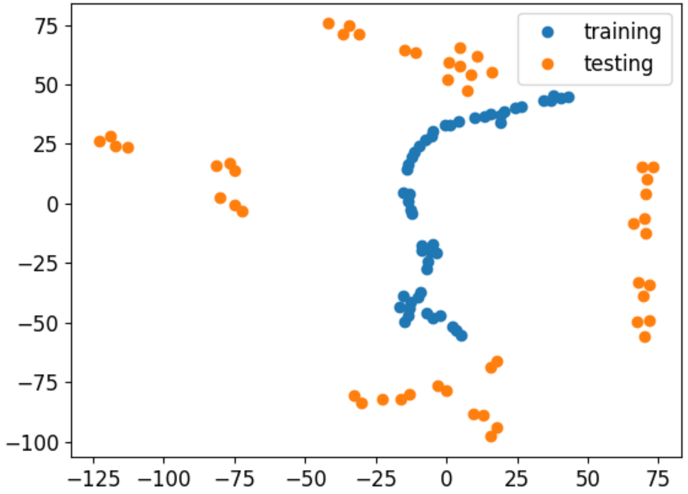
Understanding the training and testing data

- Training data collected from the data logger between studies ~ Oct 13
- Testing data collected from script during machine studies on Oct 6
- Bar plots showing the median and interquartile range
  - BPMs (bottom left): Correctors (bottom right)
- T-sne dimensionality reduction
  - Example-by-example (right top)
  - Model parameters (right bottom)



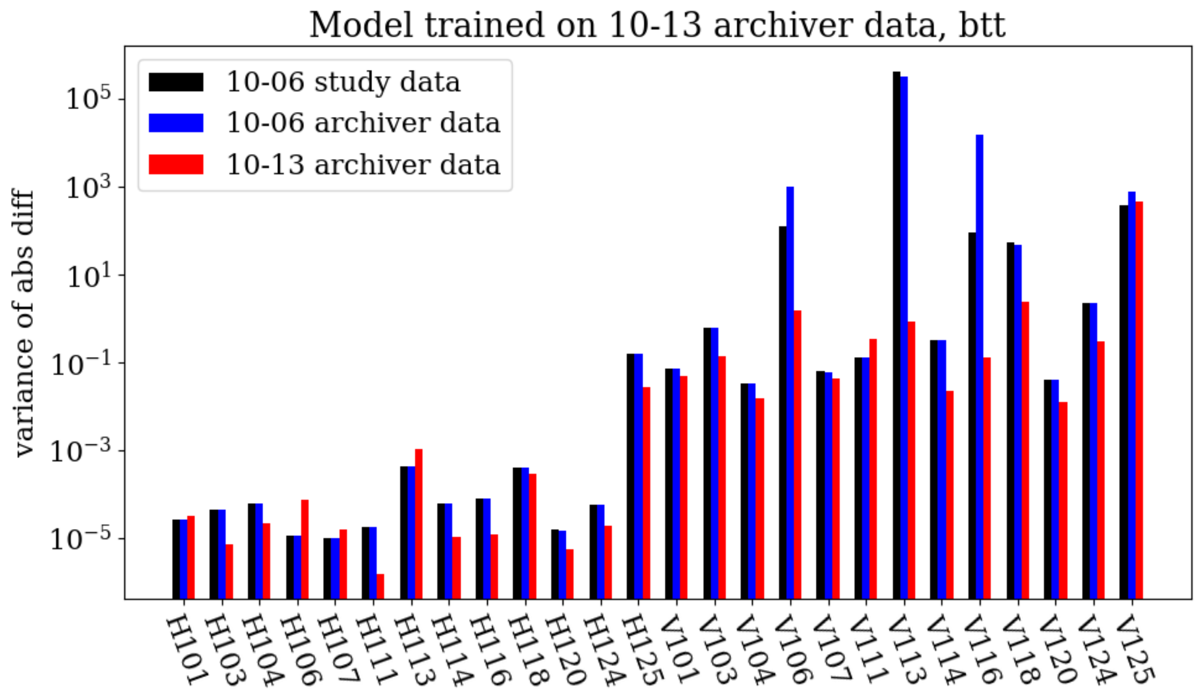


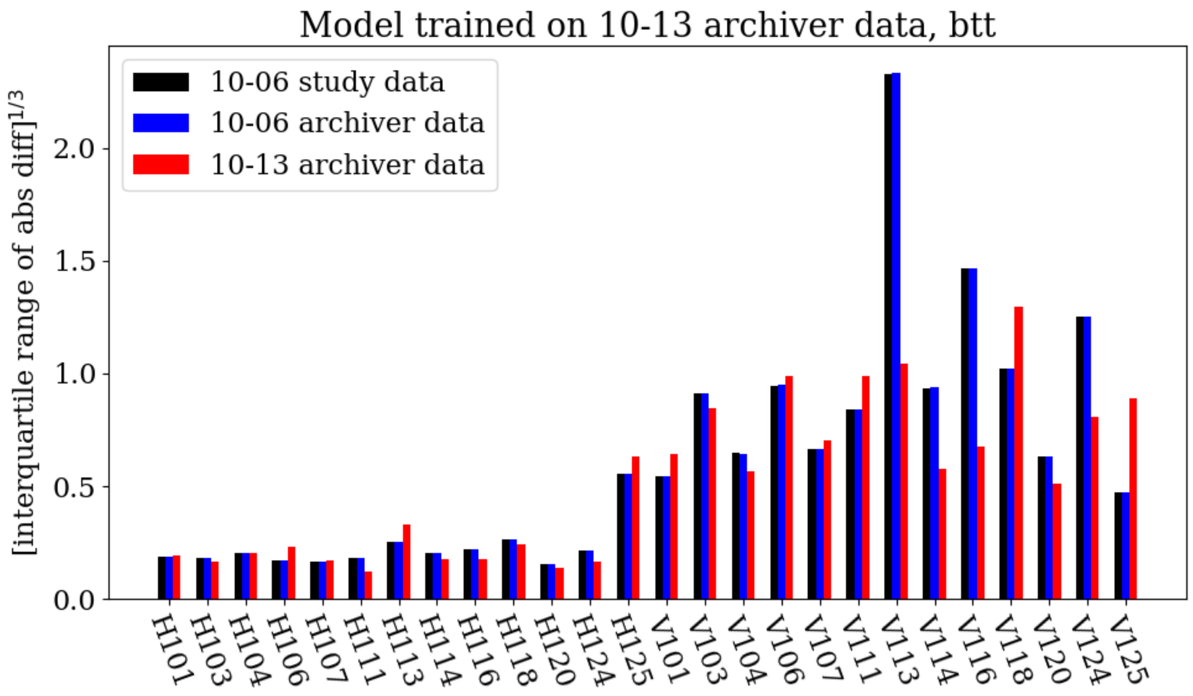




Initial machine learning studies

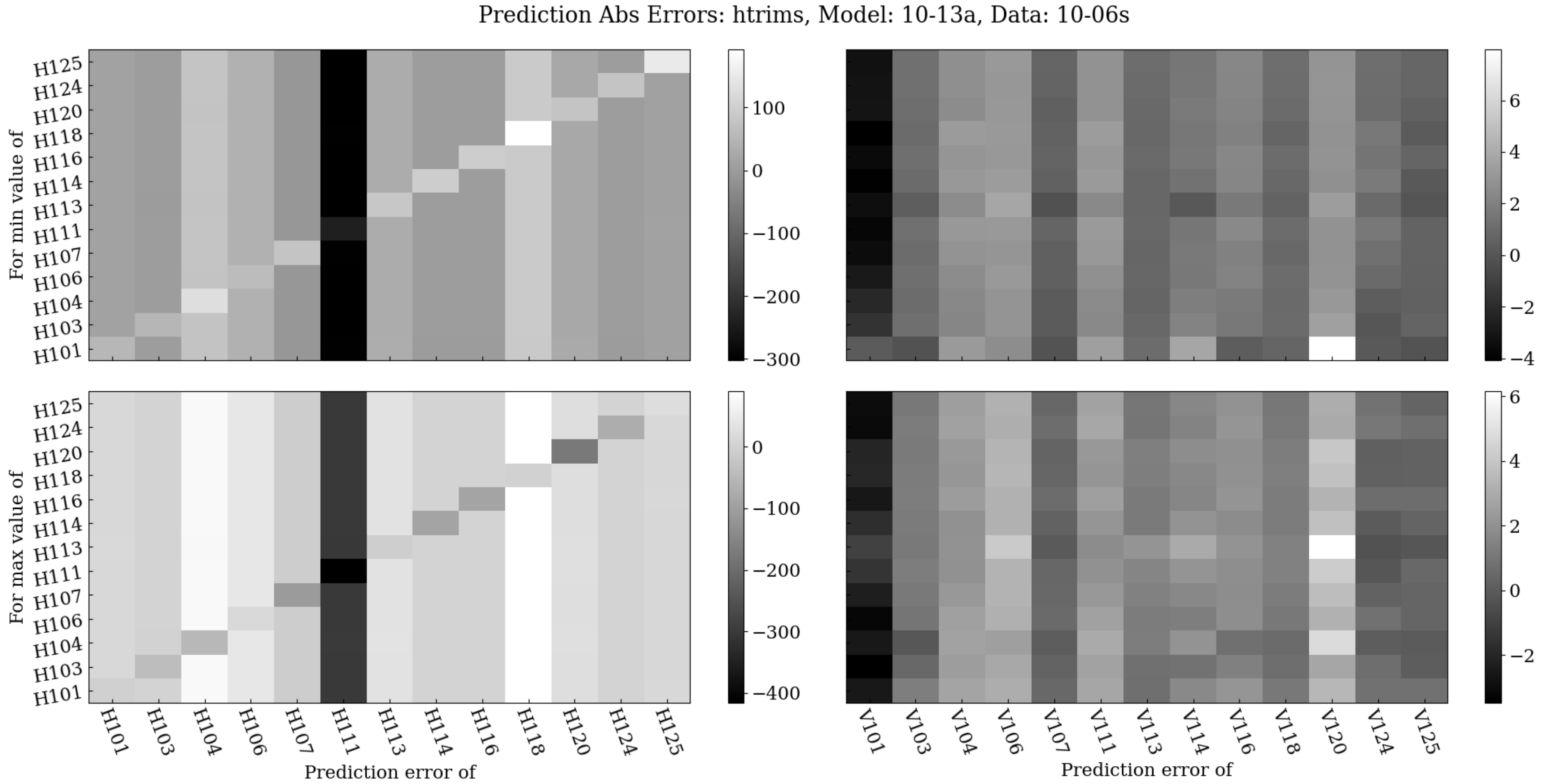
- Model is trained on operations data and tested on study data
  - Study was conducted on 6 October: Both scripted data collection and data logger collection





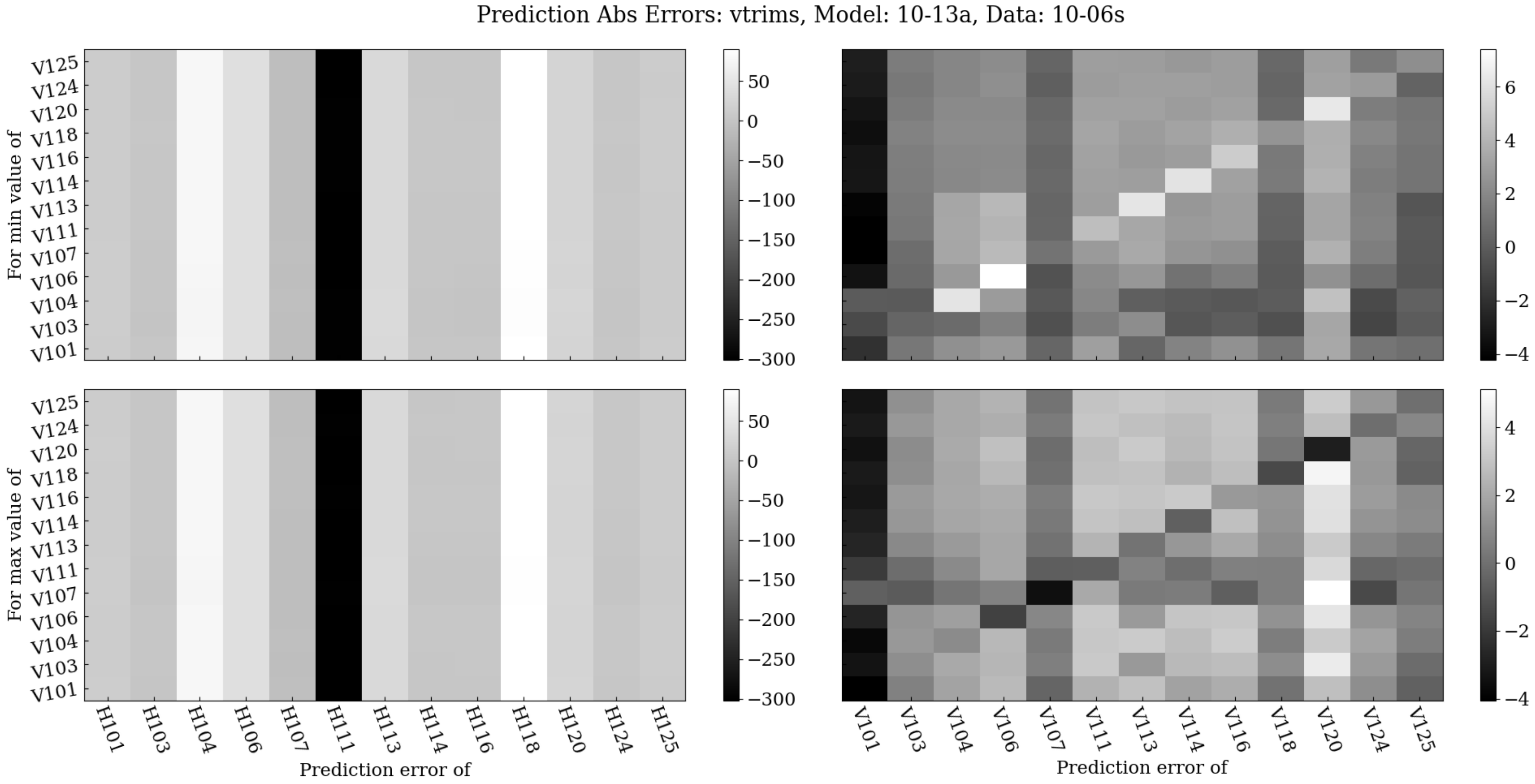
Predicting the trim settings from the BPMs

- Prediction errors for the horizontal trims
  - For the min (left) and max (right) trim setting during the study



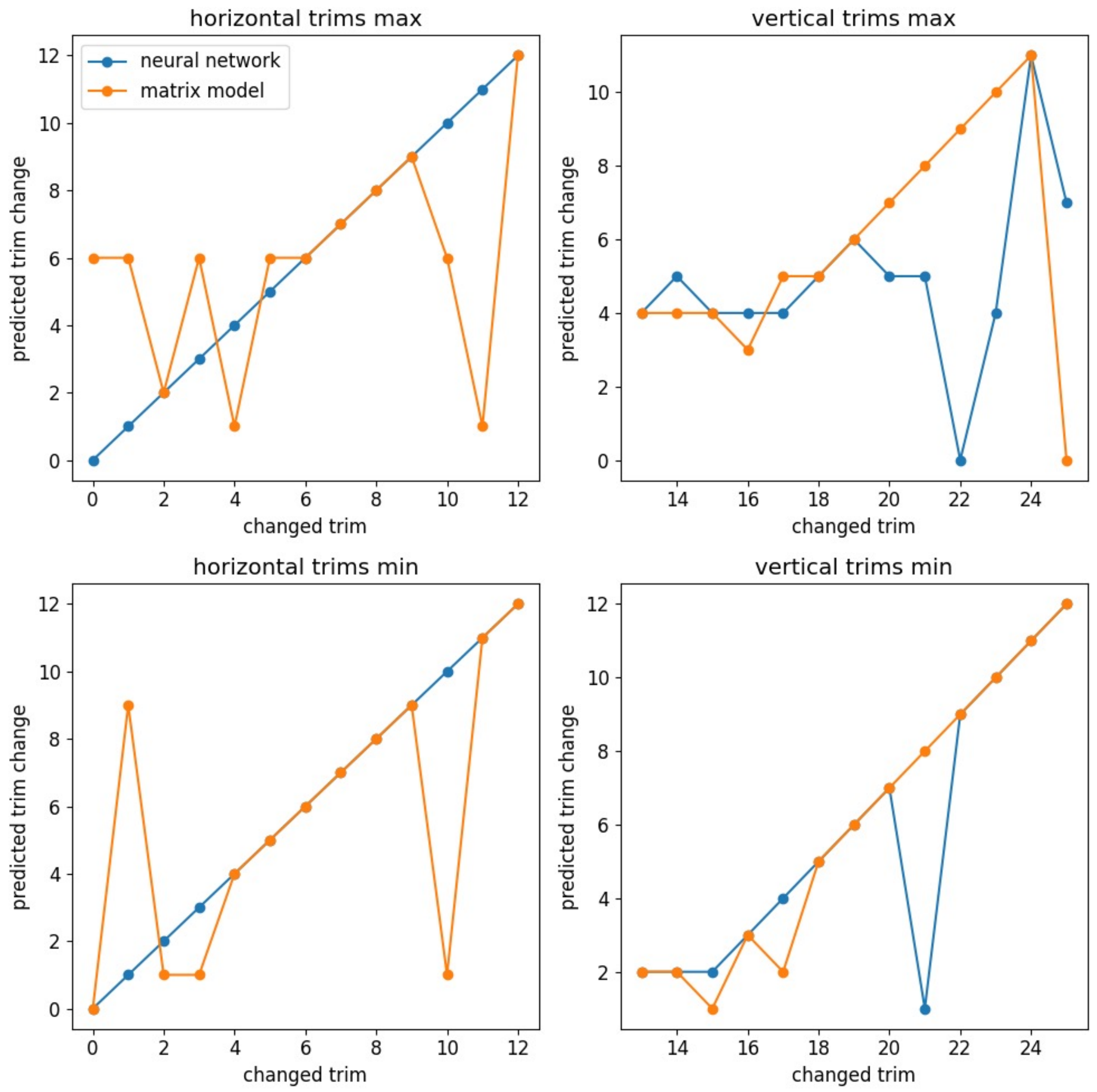
Predicting the trim settings from the BPMs

- Prediction errors for the vertical trims
  - For the min (left) and max (right) trim setting during the study



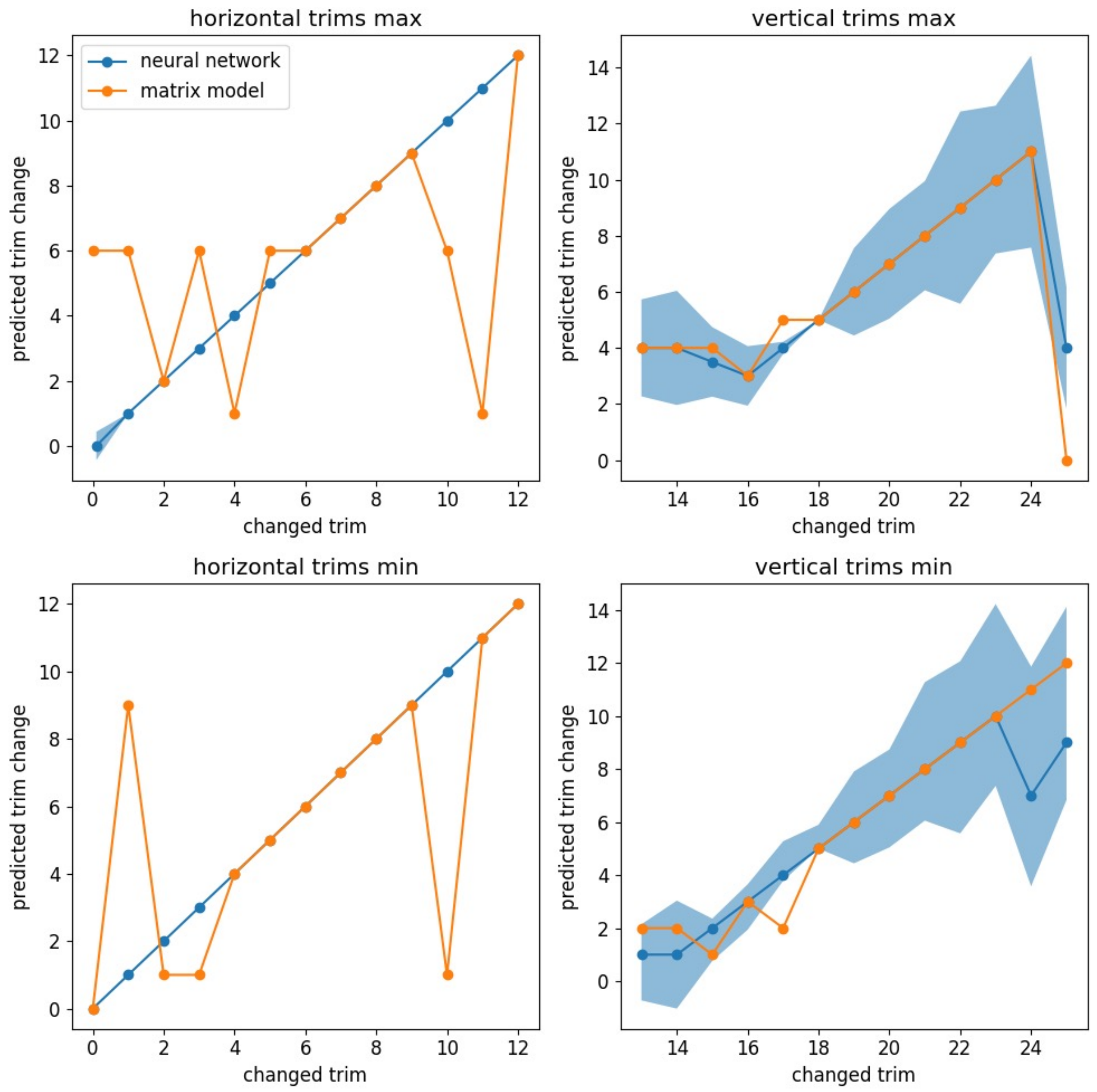
Identification of trim errors

- Use algorithm to identify trim errors
  - Model trained to predict trim settings from BPM readings
  - Compute error between predicted trim setting and actual trim settings
  - Large model offsets make threshold determination challenging
    - The changed trim has a clear signature over non changed trims
    - Subtract the mean error and then find the location with the max residual
  - Compare with linear matrix model



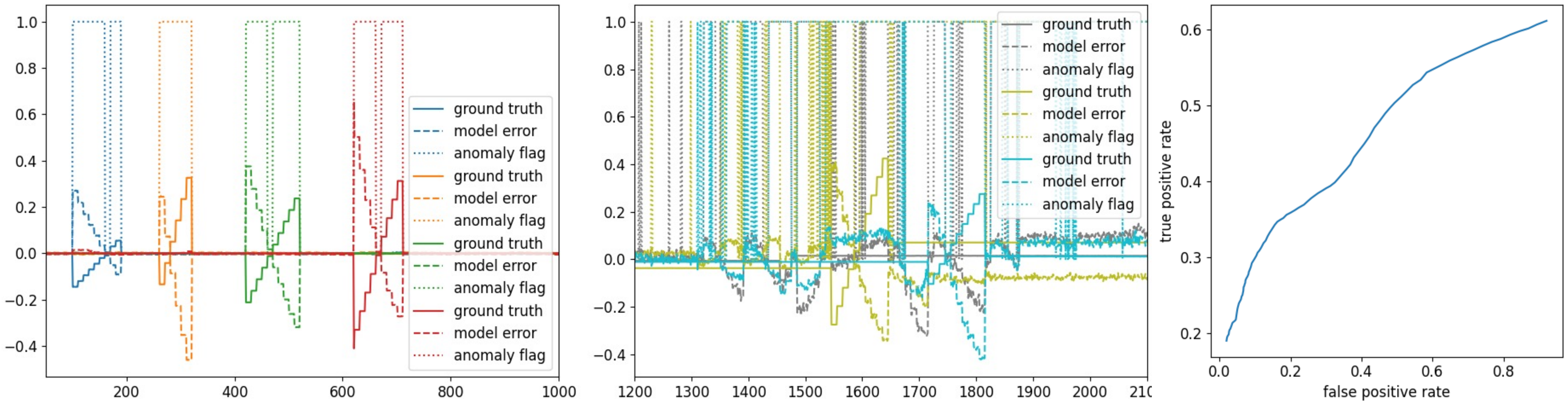
Identification of trim errors

- Use algorithm to identify trim errors
  - Model trained to predict trim settings from BPM readings
  - Compute error between predicted trim setting and actual trim settings
  - Large model offsets make threshold determination challenging
    - The changed trim has a clear signature over non changed trims
    - Subtract the mean error and then find the location with the max residual
  - Compare with linear matrix model
- Use model ensemble to improve prediction and provide error bars
  - Plots show median prediction with plus minus one standard deviation



Detailed error detection analysis / threshold evaluation

- Evaluate model performance on all test data
  - ROC curve (far right) shows true positive rate vs false positive rate for varying detection thresholds
- Error detection for horizontal trim examples (left) and vertical trim examples (middle)
  - Color coded by which trim is changed
  - Model error in dashed line
  - Anomaly flag in solid line showing that the BPM is correctly identified



Next steps

- Iterate on model learning
  - Try to improve the predictions for baseline model: planned
- Improve detection algorithm
  - Extend to remaining BPMs on the LINAC: planned
  - Expand ensemble to include linear and other models: planned
  - Try Siamese networks for stronger error discrimination: in progress
- Determine deployment path
  - Experiment on proton source?
  - Expand to include more diagnostics?

Conclusions

- Neural network model was able to detect changes in the machine in different operational modes
    - Improvement over matrix model
    - Training data collected during normal operations (no dedicated study required to train the model)

- Need to better understand the data logger issues

- Understand why the method works even with a poor (in my opinion) model

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.