Machine Learning for Anomaly Detection and Classification in Particle Accelerators

Ihar Lobach (APS, ANL)
NAPAC 2022 TUYE4
Albuquerque, NM
Tuesday, August 9th, 2022

Other contributors:
Michael Borland, Katherine Harkay, Yine Sun, Robert Soliday, Nikita Kuklev, Alexandre Sannibale (APS)
Jonathan Edelen, Abdou Diaw (RadiaSoft)

The work is supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357.

Outline

A set of studies to explore
possible ML applications at APS:

Supervised ML for anomaly classification in the APS injector:

- Perturbed Process Variables (PVs) are included in model input
- Perturbed PVs are not included in model input

Unsupervised ML for anomaly detection and clustering:

- Autoencoders for groups of related PVs
  - PAR BPMs and Correctors
  - Detection of power supply trip precursors in the APS storage ring
- Autoencoders and Variational Autoencoders for anomaly clustering

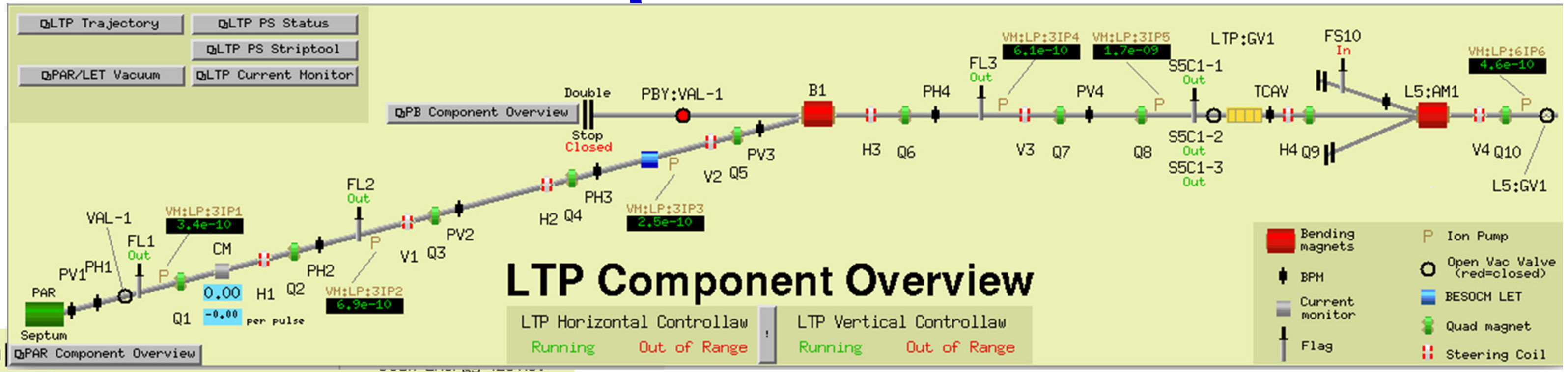
Part 1: Supervised ML

APS Injector Complex: $\underbrace{\text{Linac} \rightarrow \text{PAR}}_{\text{Location of our studies}} \rightarrow$ Booster $\rightarrow$ Storage Ring

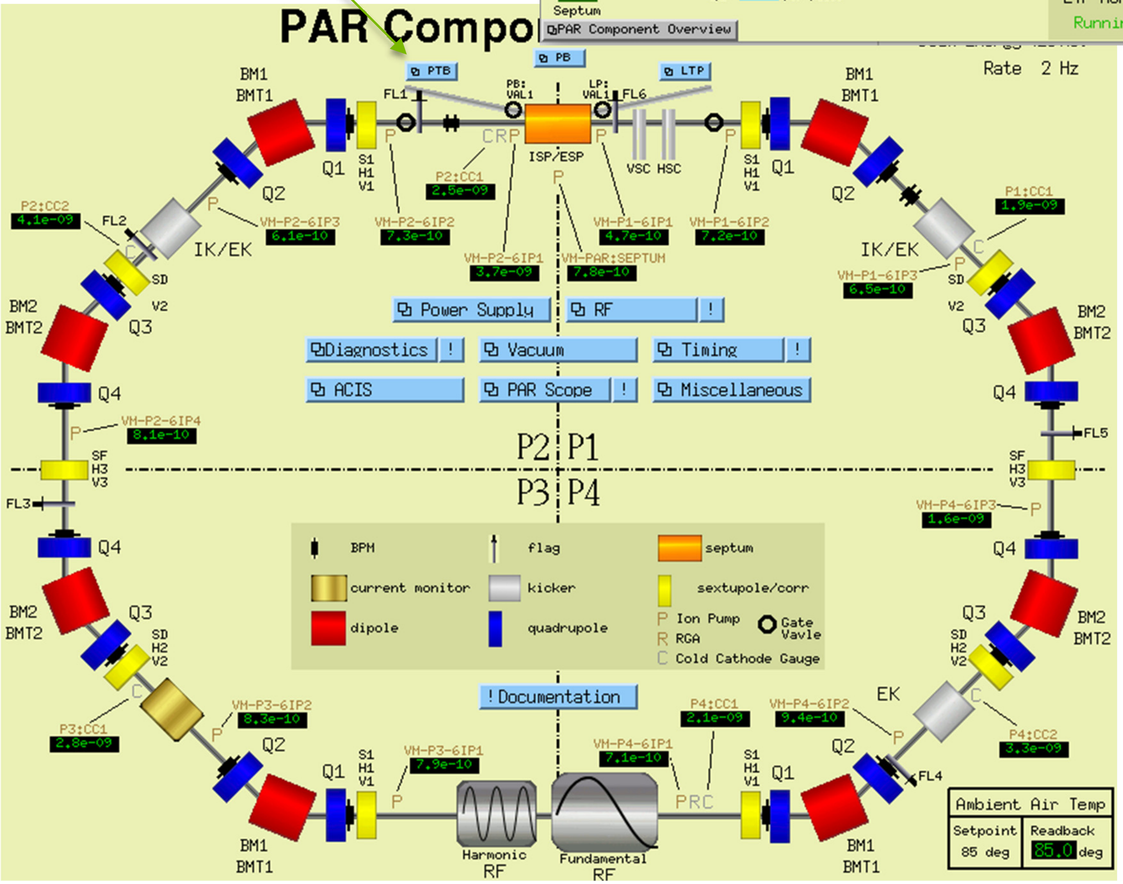
Problem Statement:

- Occasionally, poor transmisssion efficiency is observed **in the Particle Accumulator Ring (PAR) and in the Linac-To-PAR transport line (LTP)**
- The machine expert is needed to restore good efficiency, but they may not be readily available.
- Can we train an ML model to pinpoint the source of poor performance?
- Advantage of carrying out this study in PAR and LTP is that a considerable dedicated study time is available. Therefore, we can:
  - Create intentional perturbations in PAR and LTP, resulting in poor injection/extraction efficiency
  - Log all available PVs of the machine
  - Use these data as labeled training data for supervised ML

LTP and PAR Component Overview



Extraction to Booster
PTB = PAR-To-Booster



Considered perturbations:

LTP perturbations:
- B1 dipole *'LTP:B1:CurrentAO',*
- H3 corrector *'LTP:H3:CurrentAO'*

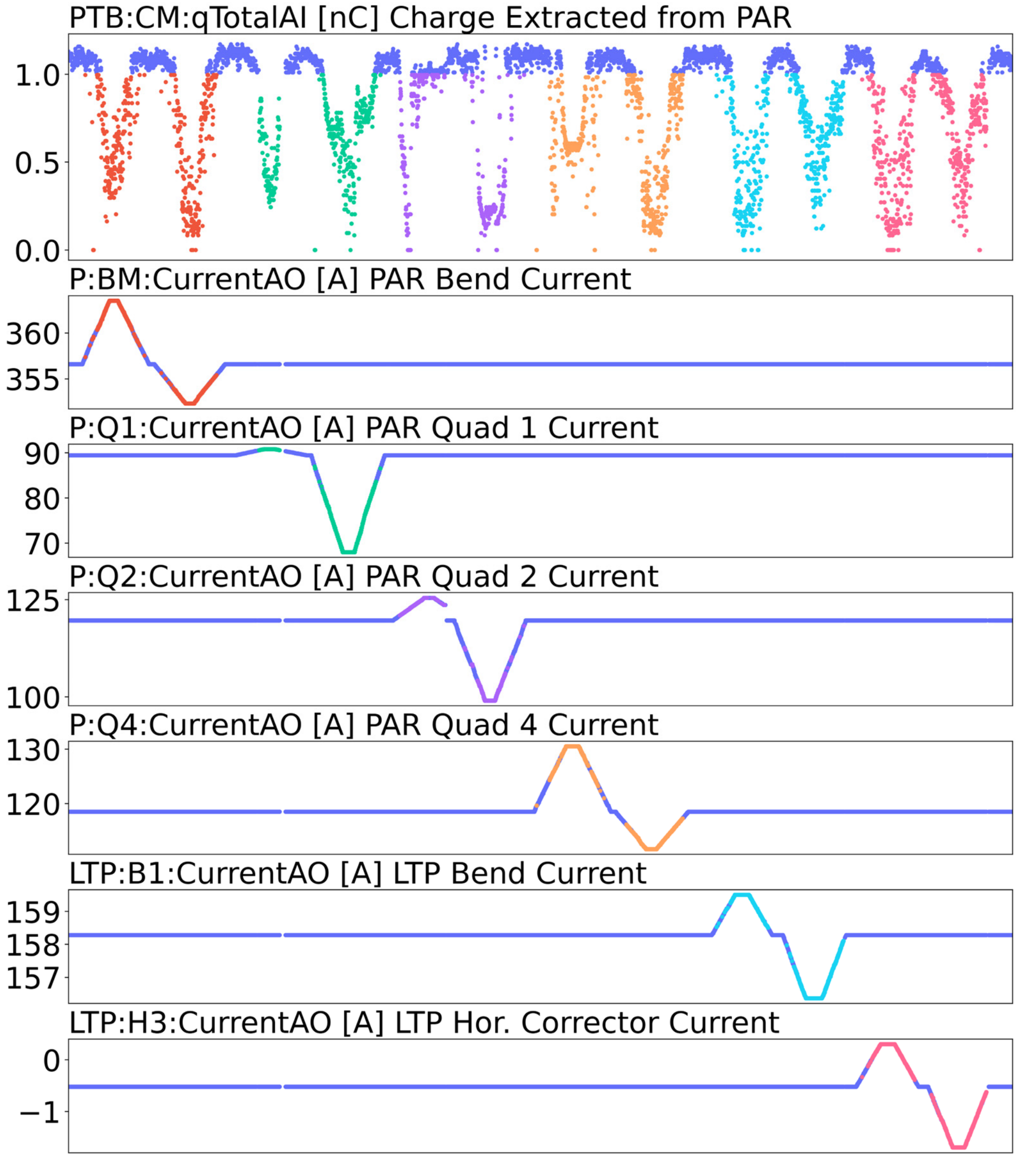
PAR perturbations:
- Dipole current *'P:BM:CurrentAO',*
- Quadrupoles *'P:Q1:CurrentAO', 'P:Q2:CurrentAO', 'P:Q4:CurrentAO',*

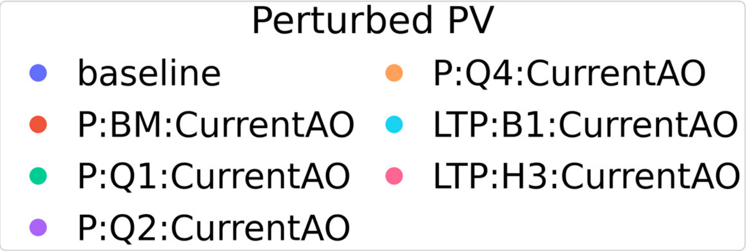
Timings:
- Linac Trigger Timing *'It:LinacTrig2ParIpAO',*
- PAR Kicker Timings *'It:P1IKtrig2ParIpAO', 'It:P2IKtrig2ParIpAO'*

* We also considered beam-to-rf phases in the linac

Data Collection Routine

- 5 studies between Nov, 2021 and July, 2022
- At all times, we log ~9000 PVs at 2Hz, which is the injection cycle rate
- First, we search for the perturbation limits
- Then, we automatically ramp perturbation PVs one by one within the found ranges
- The limit search can be automated: we developed a program that moves each PV in small steps until the extracted charge drops below a certain threshold





Model Input and Output

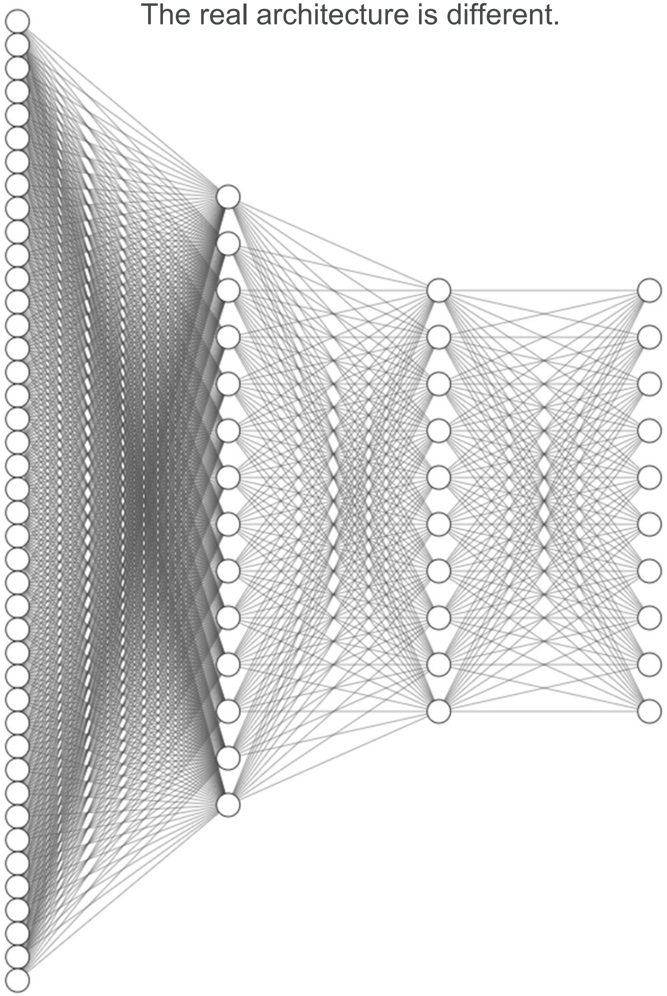
Model input:

- PAR BPMs
- PAR Correctors
- LTP BPMs
- LTP Correctors
- PAR Charge (10 timesteps within injection cycle)
- LTP and PTB charges
- Injection and Extraction efficiencies
- Linac trigger timing
- PAR kicker timings
- PAR Quadrupoles and Dipoles
- LTP Quadrupoles and Dipole

Overall, ~200 PVs

The perturbed PVs can be included in the model input, or not.

This neural network is for illustration purposes only.
The real architecture is different.



The model returns probabilities for the possible causes of poor performance, .e.g.,

Model output:

'baseline'	*0.50*
'P:BM:CurrentAO'	*0.04*
'P:Q1:CurrentAO'	*0.01*
'P:Q2:CurrentAO'	*0.10*
'P:Q4:CurrentAO'	*0.01*
'LTP:B1:CurrentAO'	*0.09*
'LTP:H3:CurrentAO'	*0.19*
'It:LinacTrig2ParIpAO'	*0.03*
'It:P1IKtrig2ParIpAO'	*0.01*
'It:P2IKtrig2ParIpAO'	*0.02*

The possible signatures include the reduced charges in different parts of the machine, changed BPM signals, changed charge vs. time within one injection cycle, changed corrector strengths (due to the orbit control law in PAR and due to the trajectory control law in LTP).

NN diagram:
https://alexlenail.me/NN-SVG

Possible Implementation in Daily Operations

model	prediction number 1	prob1	prediction number 2	prob2	prediction number 3	prob3
Neural Network	LTP:H3:CurrentA0	1	P:BM:CurrentA0	0	P:Q4:CurrentA0	0
LightGBM	LTP:H3:CurrentA0	0.995	L2:PP:phaseAdjA0	0.004	baseline	0
AdaBoost	LTP:H3:CurrentA0	0.771	L2:PP:phaseAdjA0	0.097	baseline	0.045
Random Forest	LTP:H3:CurrentA0	0.213	LTP:B1:CurrentA0	0.135	P:Q4:CurrentA0	0.124

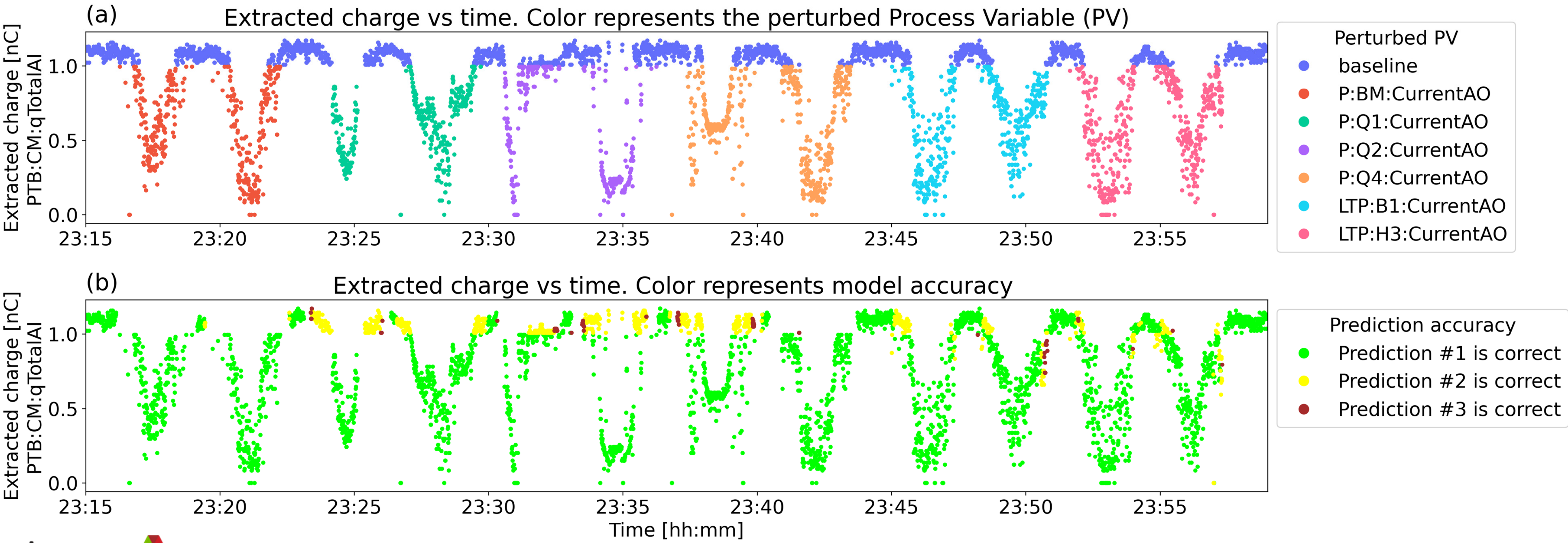
This video is using synthetic data for illustration purposes.
This is why predictions change so often.

- A simple prediction table in the terminal (updated every 2 seconds)
- The predictions and the machine state are continuously logged
- When new training data are available, all 4 ML models can be retrained by running one script in the terminal

Neural Network Classifier Performance

Perturbed PVs **are** included in model input

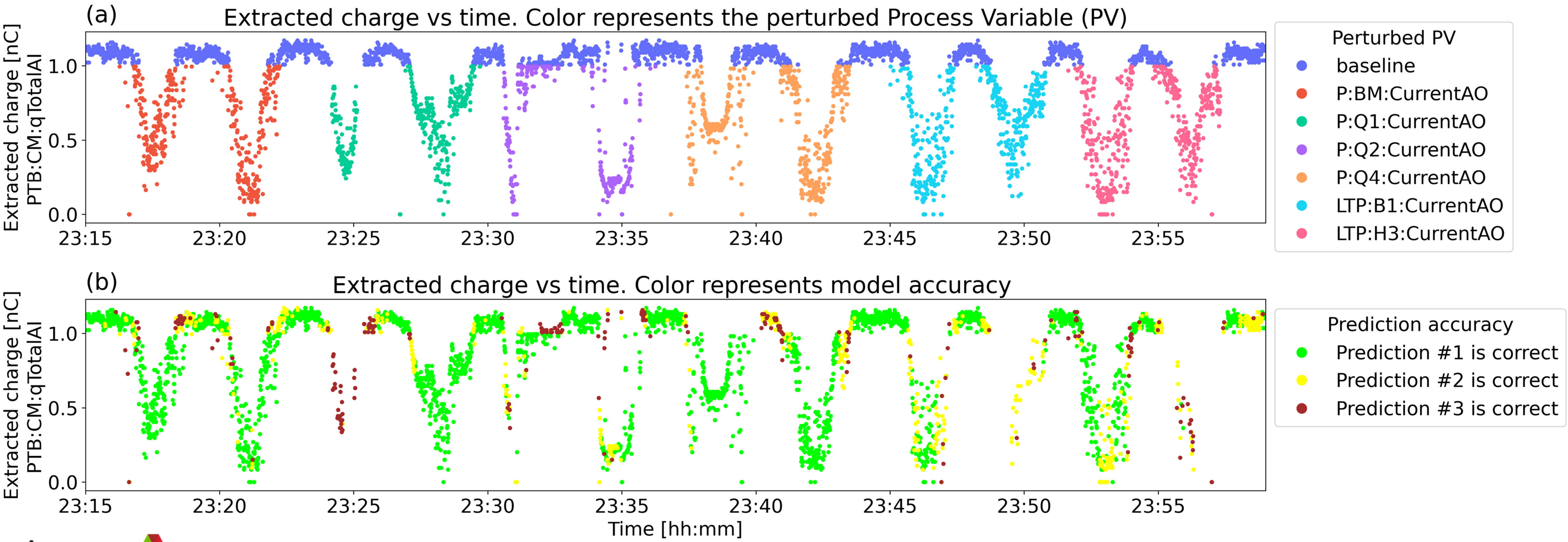
- The training data were collected in 2021 on November 4, 6, and 14
- Neural net architecture: 167→80→Dropout→80→Dropout→40→7→SoftMax
- Then, the model is tested (see below) on the data collected on January 30, 2022



Neural Network Classifier Performance

Perturbed PVs **are not** included in model input

- The training data were collected in 2021 on November 4, 6, and 14
- Neural net architecture: 155→80→Dropout→80→Dropout→40→7→SoftMax
- Then, the model is tested (see below) on the data collected on January 30, 2022



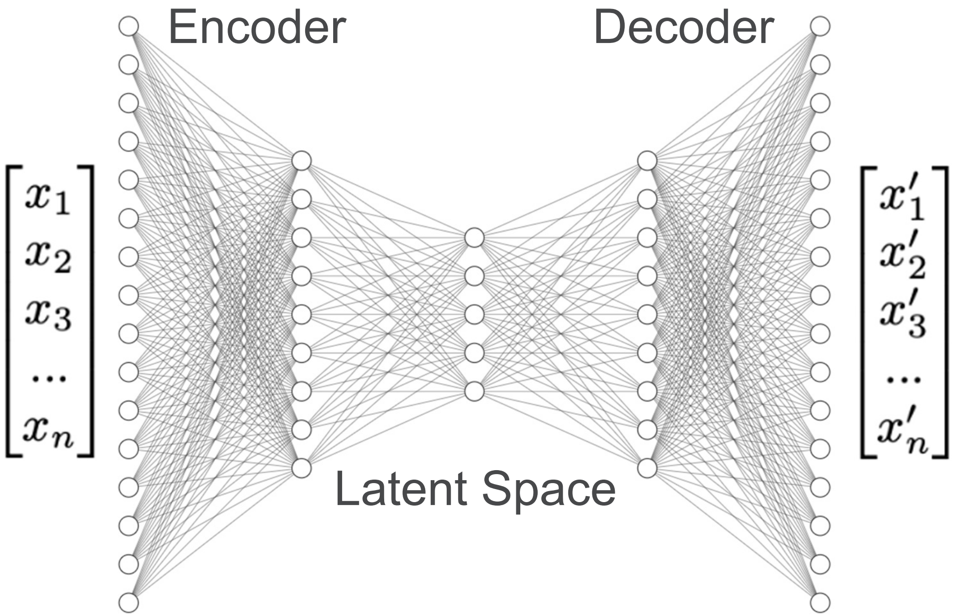
Lessons Learned

- Outliers must be removed. An effective one-fits-all approach is Isolation Forest.
- Training convergence is sensitive to data scaling. Good choices from sklearn: RobustScaler, QuantileTransformer, StandardScaler
- Loss function must be weighted according to the sizes of anomaly classes.
- To improve accuracy for the ”baseline” class, one can regularly augment the training data by
  - Fresh baseline data
  - Fresh routine SCR saves (machine snapshots, made by operators or studiers)
- Using many input PVs may cause overfitting and poor generalization. Dropout layers are necessary to prevent overfitting.
- Dimensionality reduction techniques may further improve performance.

https://scikit-learn.org/
https://pytorch.org/
https://www.tensorflow.org/

Part 2: Unsupervised ML

- Neural Network Autoencoder:



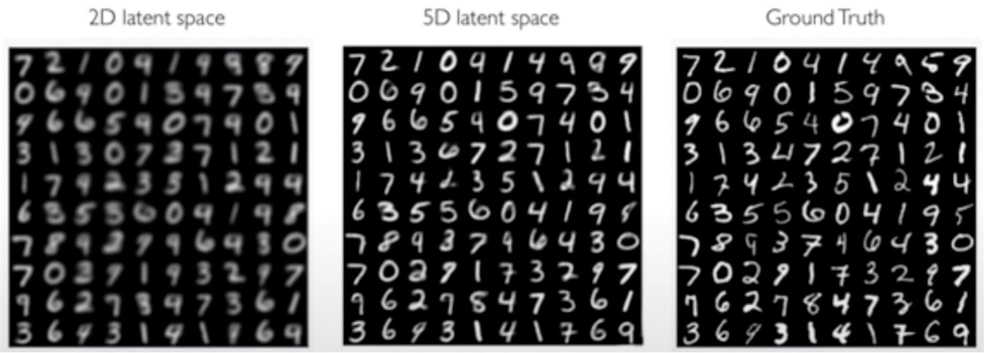
Goal: reconstruct input values

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{n}\left(x_i^{(j)} - x_i'^{(j)}\right)^2$$

NN schematic:
https://alexlenail.me/NN-SVG

- If we train an autoencoder on "normal" data, then the reconstruction error will be low for normal data, and (likely) high for anomalous data

Autoencoder performance example with the MNIST data set:



Source:
MIT 6.S191 (2020): Introduction to Deep Learning
Deep Generative Modeling
Lecturer: Ava Soleimany
January 2020

Argonne NATIONAL LABORATORY

Possible Applications in APS Injector

Use for groups of PVs to better localize the source of poor performance:

Examples of such groups:

- Beam-to-RF phases at different klystrons in the linac
- Linac-To-PAR trigger and PAR kicker timings
- Linac-To-PAR transport line BPM signals and corrector strengths
- PAR BPM signals and corrector strengths

NN schematic:
https://alexlenail.me/NN-SVG

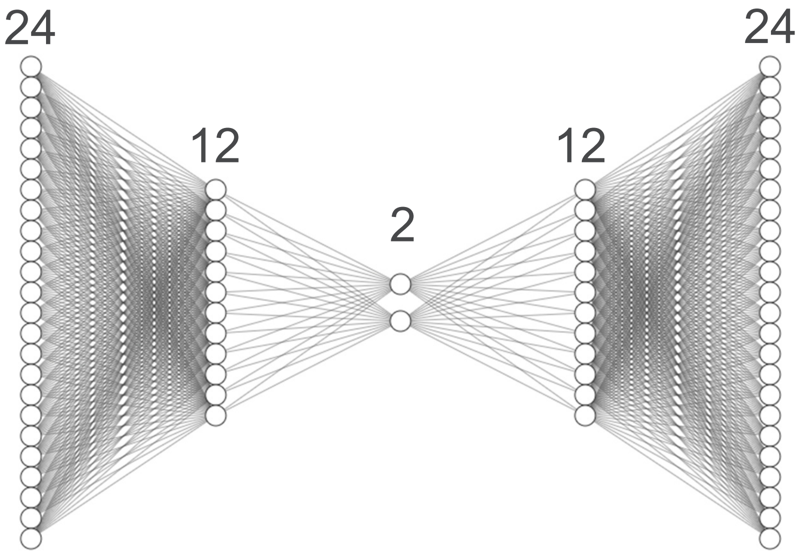
Possible Applications in APS Injector

Use for groups of PVs to better localize the source of poor performance:

Examples of such groups:

- Beam-to-RF phases at different klystrons in the linac
- Linac-To-PAR trigger and PAR kicker timings
- Linac-To-PAR transport line BPM signals and corrector strengths
- PAR BPM signals and corrector strengths

24 inputs:
Horizontal BPM signals (16)
Horizontal corrector currents (8)
PAR dipole and quadrupole currents were **not** included

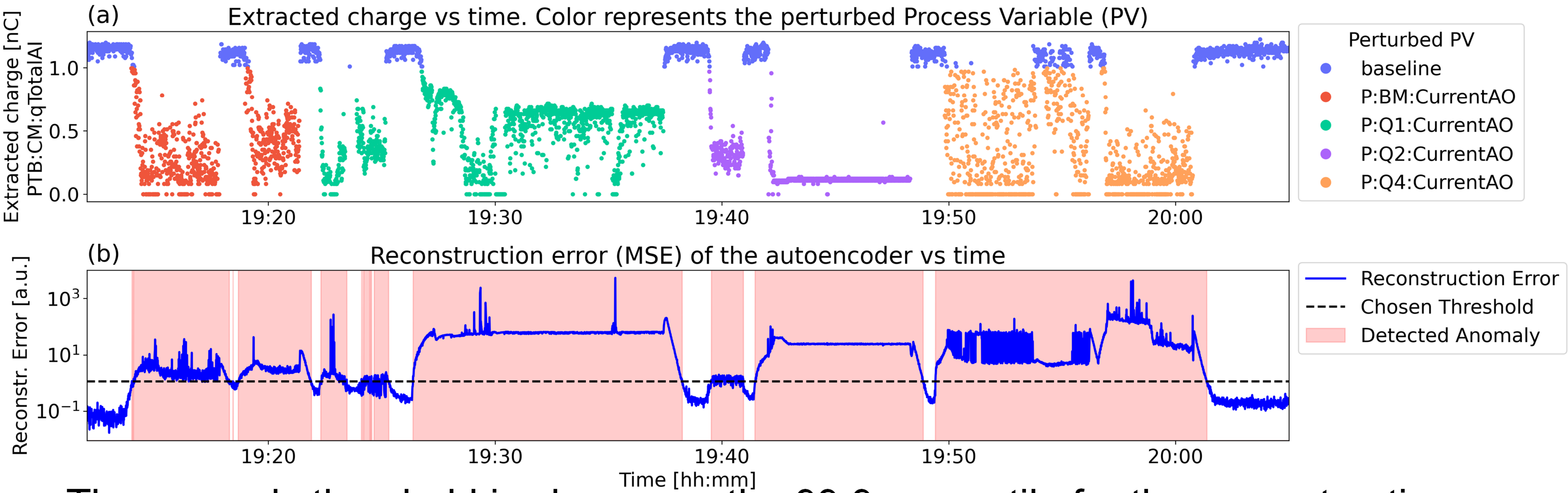


Loss function:
Mean Squared Error (MSE)

NN schematic:
https://alexlenail.me/NN-SVG

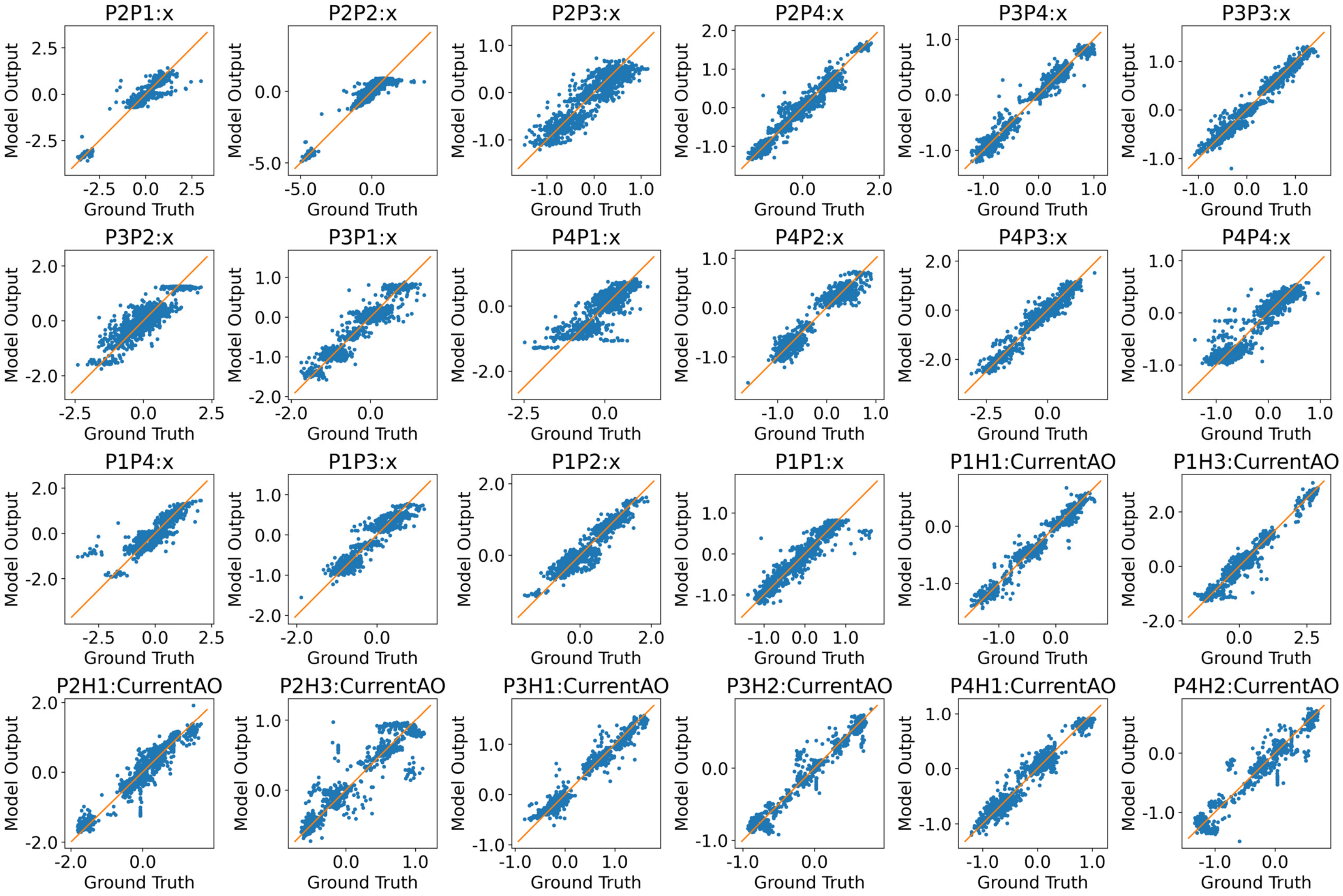
Autoencoder Anomaly Detection Performance in PAR

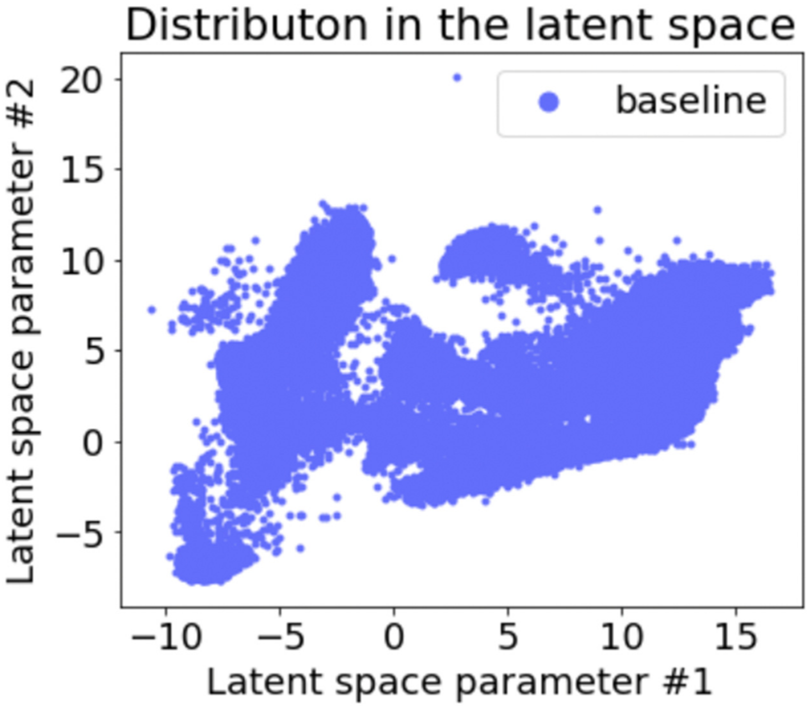
- Autoencoder was trained on the ”baseline” data from Jan 24 to Jan 30, 6 pm
- It is then tested on the poor-performance data with intentional perturbations during the study on Jan 30, after 6 pm (see below)



- The anomaly threshold is chosen as the 99.9 percentile for the reconstruction error in the training data

Autoencoder Reconstruction Performance and Latent Space (for the baseline data that were used for training)

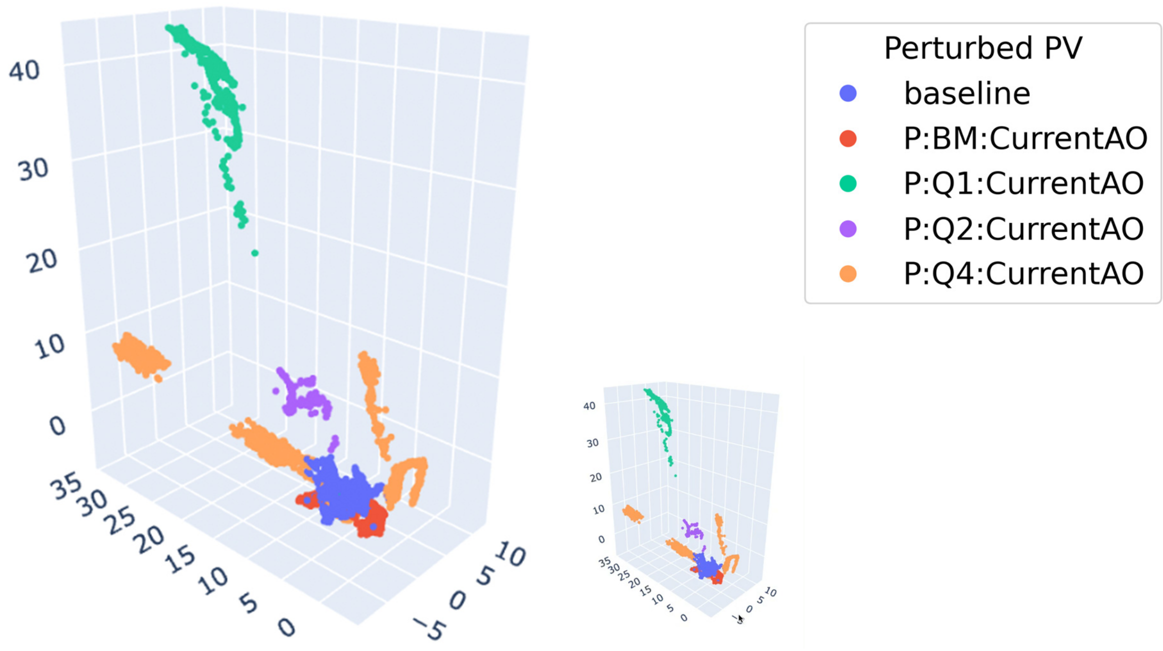




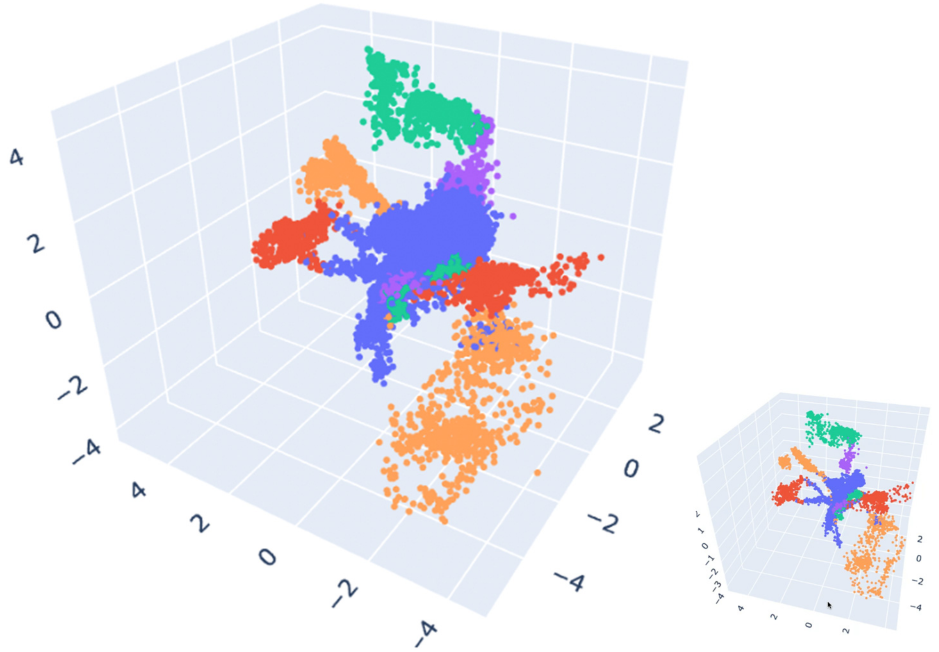
Autoencoders for Anomaly Clustering

Data from Jan 30 shift, 3D latent space

We couldn't achieve disentanglement (partly, because we varied PVs one by one). But, at least, the latent space is more regularized with β-VAE



Regular Autoencoder



β-Variational Autoencoder

- Nothing can be assumed about the latent space distribution
- Regular autoencoders may place two “similar” points far apart in the latent space, if it minimizes the reconstruction error. Prone to overfitting
- The distribution in the latent space is often sparse and hard to interpret

- Points are more symmetrically distributed around (0,0,0) and the size of the distribution is of the order of 1, because of the Gaussian prior.
- The clusters are more continuous because of the probabilistic nature of VAE
- Parameter $\beta > 1$ encourages a disentangled representation

Argonne NATIONAL LABORATORY

VAE and β-VAE

$$\mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta\, D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}))$$

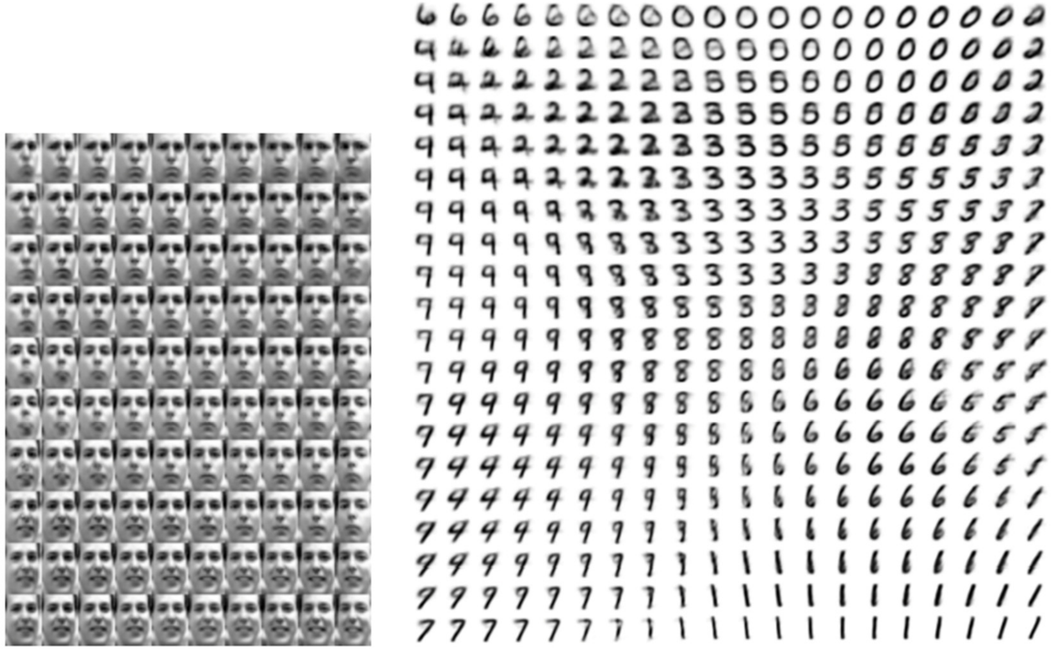
Approximately MSE

Kullback-Leibler divergence between the encoder and the prior distribution of latent space $p_\theta(z)$ [usually $N(z; 0, I)$]

[Submitted on 20 Dec 2013 (v1), last revised 1 May 2014 (this version, v10)]

Auto-Encoding Variational Bayes

Diederik P Kingma, Max Welling

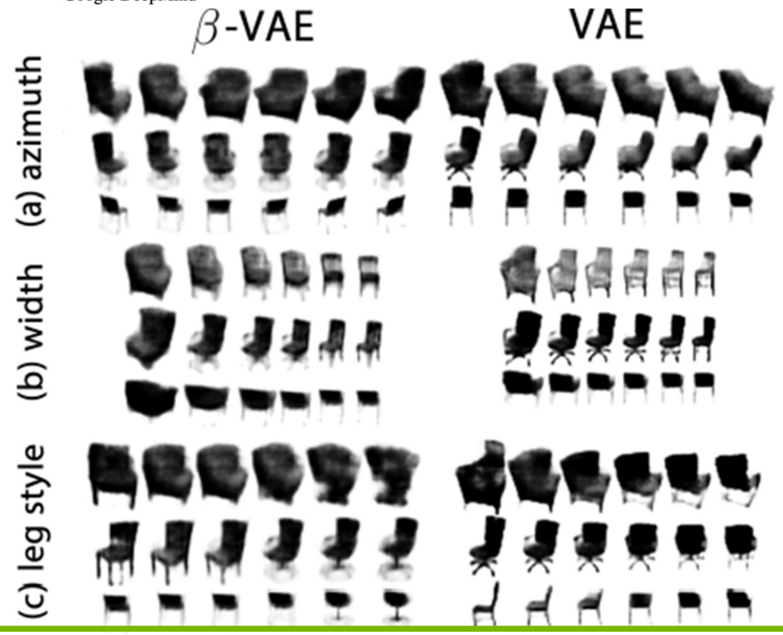


(a) Learned Frey Face manifold (b) Learned MNIST manifold

Published as a conference paper at ICLR 2017

β-VAE: LEARNING BASIC VISUAL CONCEPTS WITH A CONSTRAINED VARIATIONAL FRAMEWORK

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, Alexander Lerchner
Google DeepMind



VAE learns a constrained and continuous representation

β-VAE encourages a disentangled representation

Other Applications of Autoencoders at APS

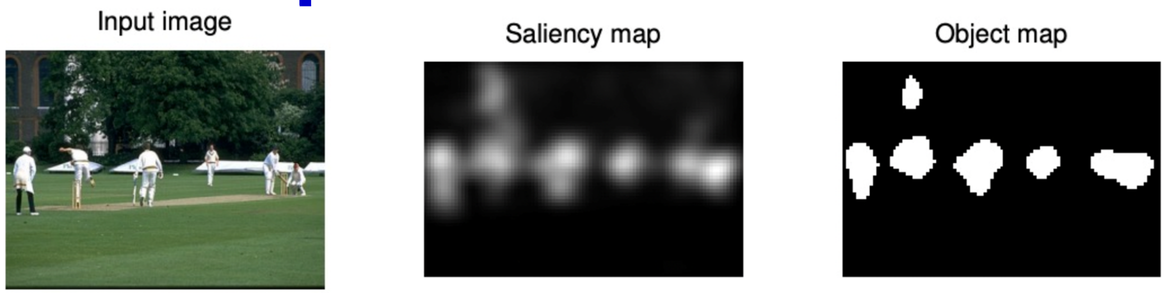
Detecting precursors for magnet power supply trips in the APS **Storage Ring**:

- Based on temperature data
- Based on current noise monitor data

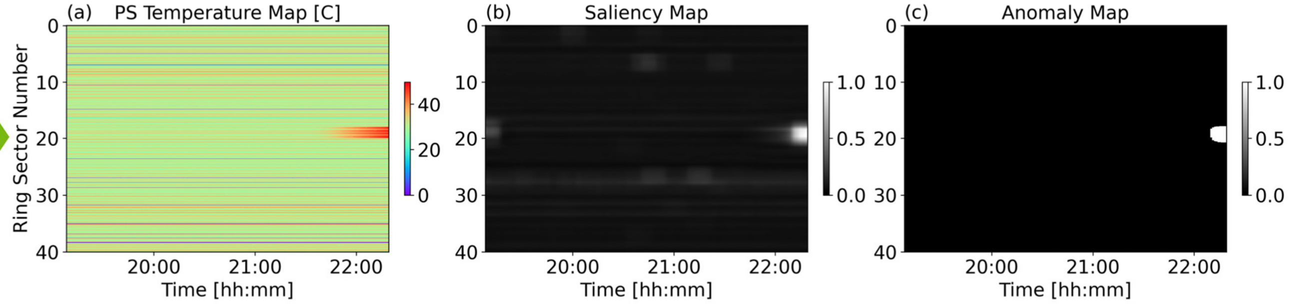
For more details, please see our poster **TUPA29**

Anomalies in Power Supply Temperatures in the Storage Ring

X. Hou and L. Zhang, "Saliency Detection: A Spectral Residual Approach," *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1-8, doi: 10.1109/CVPR.2007.383267.

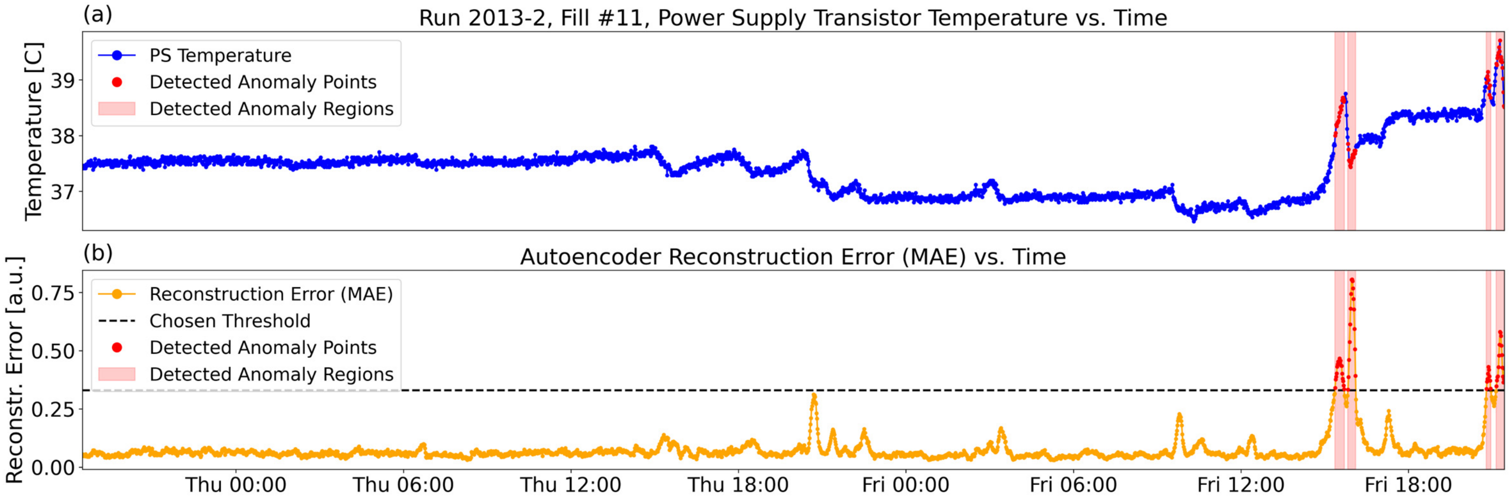


It may be useful in our case



- Detection of anomalies in power supply temperature maps is similar to object detection in images. Therefore, we have used object detection methods and are considering Convolutional Neural Networks for PS temperature maps.

- One can also use several contiguous temperature values of a single PS as input for an autoencoder. Such autoencoders can be sensitive to unusual temperature behavior in time.

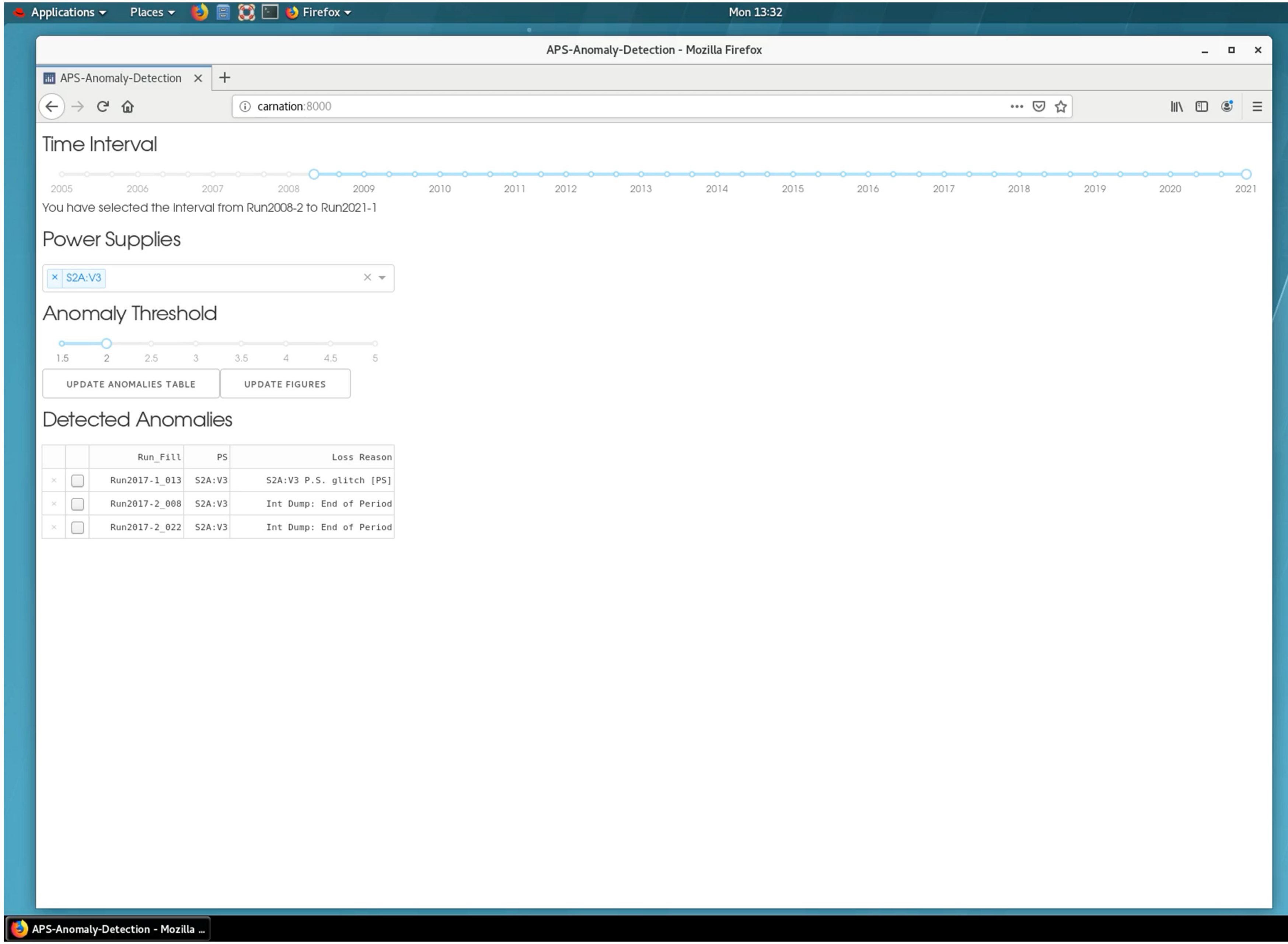


For more details, see our poster later today: **TUPA29**

Anomalies in Power Supply Currents in the Storage Ring

- Trained neural network autoencoders for 1320 power supplies for quadrupoles, sextupoles, horizontal and vertical correctors.
- Using historical data from the current noise monitors
- The application allows to quickly examine every anomaly and decide whether any maintenance is needed

For more details, see our poster later today: **TUPA29**



https://plotly.com/dash/

Conclusions

Supervised ML:

- The neural network classifier for PAR and LTP is rather accurate and generalizes well (with perturbed PVs included or not included in model input)
- One disadvantage is that supervised ML requires labeled data
- One can automate the data collection process and re-run it after every shutdown

Unsupervised ML:

- Autoencoders, trained on normal-operation data, can be effective at detecting anomalies in groups of related process variables (PAR orbit, LTP trajectory, timings, beam-to-rf phases)
- β-Variational Autoencoders can be useful for anomaly clustering and dimensionality reduction

Thank you for your attention!