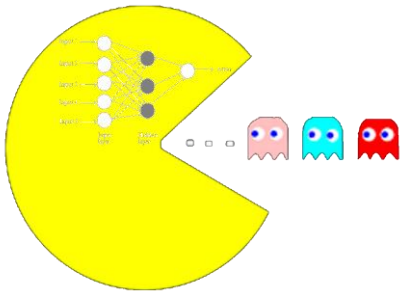
Experience with Machine Learning for Particle Accelerators at PSI

Jaime Coello de Portugal

PAUL SCHERRER INSTITUT

PACMAN (Particle Accelerators & Machine Learning) project.



Swiss Data Science Center (SDSC), the École Polytechnique Fédérale de Lausanne (EPFL) and PSI

- Particle accelerators are complex facilities that produce and handle large amounts of data.
- The project target is to explore the possibilities to use machine learning to improve the performance of particle accelerators.
- Started in March of 2019 and is about to finish.

Outline

1. Interlock forecasting at HIPA
2. Beam losses optimisation
3. Anomaly detection using ensemble models on streaming data
4. Virtual diagnostics for SINQ temperature sensors
5. Data collection and exploitation of the models

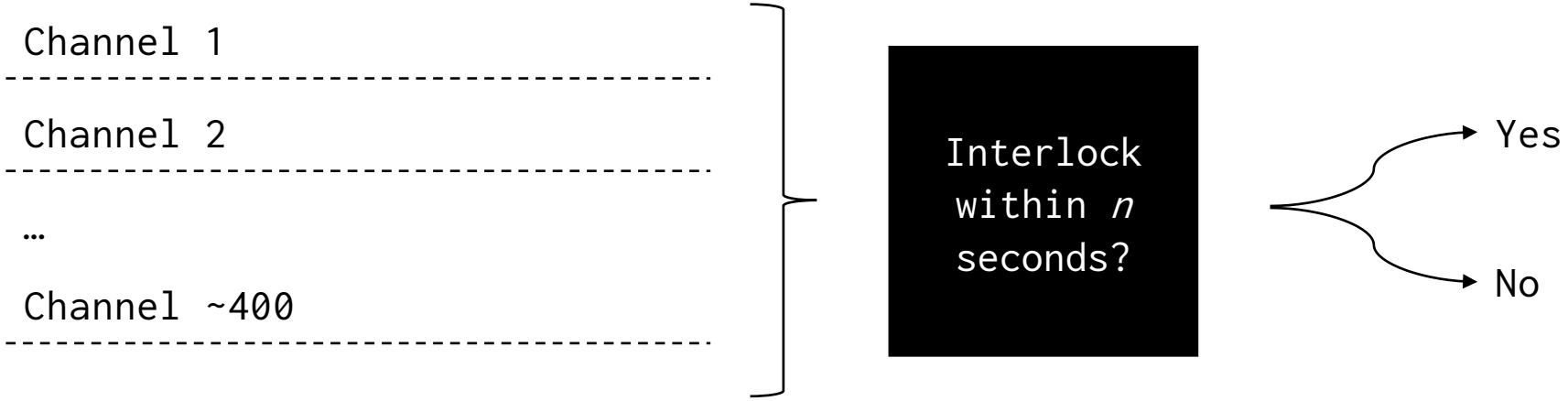
Interlock forecasting: introduction

- Beam interruptions lead to abrupt operational changes. About 20% of all the lost beam time is due to short beam interruptions.

- Can we forecast beam interruptions at HIPA?
  -> If forecasted enough time in advance, measures could be taken to prevent them.

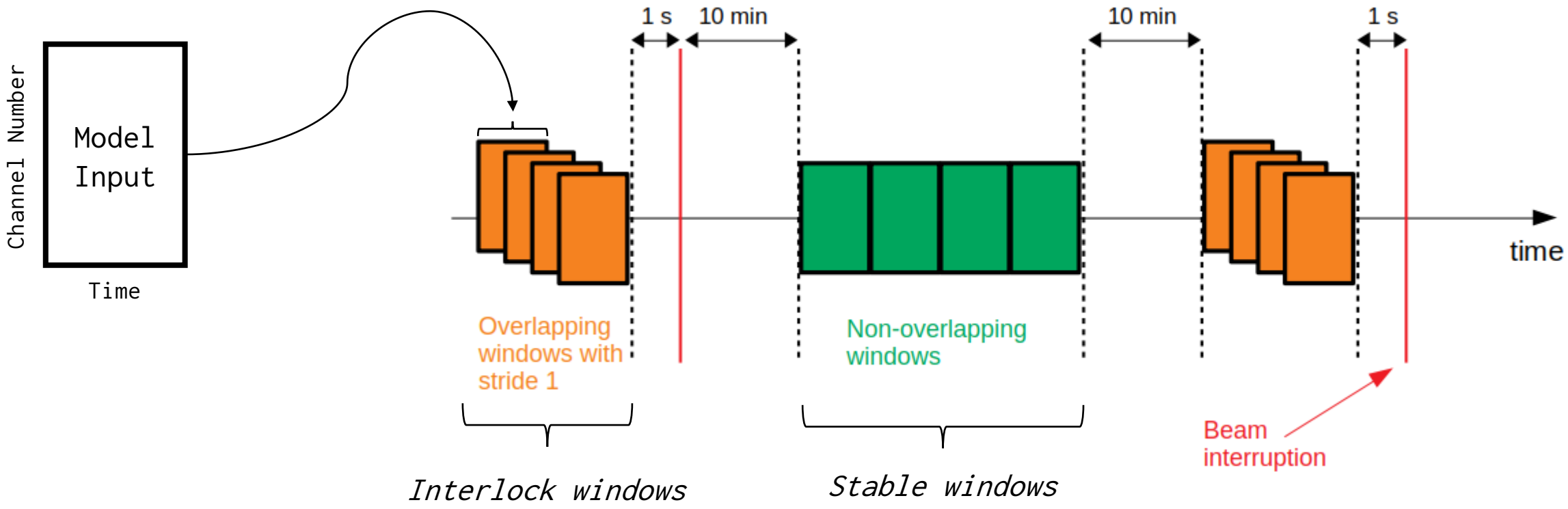
Take data from about 400 channels all over HIPA:

Train a model to answer the question:



- This work was presented here before: https://indico.psi.ch/event/10674/

- The input channels are cut in windows:



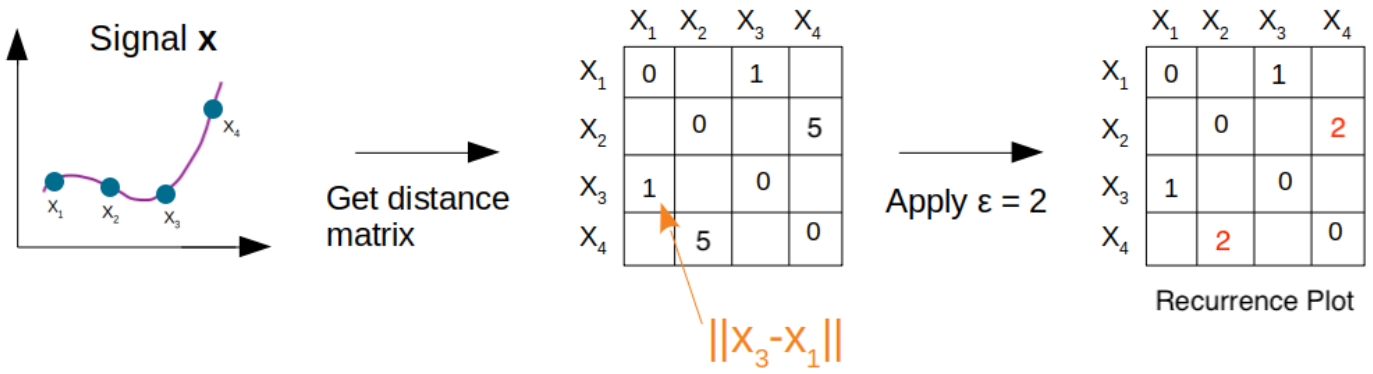
- Close to the interlock we take N sliding windows as *positive examples.*
- As stable windows (*negative examples*) are abundant and similar, non-overlapping windows are taken.

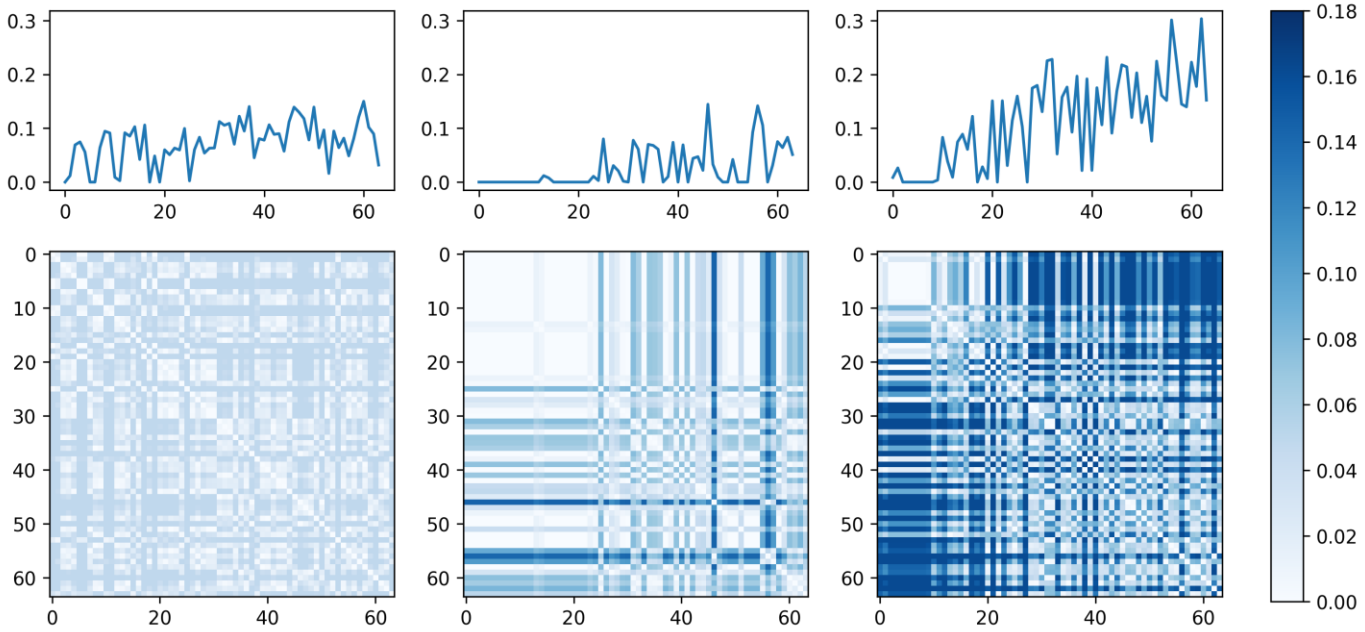
- *Recurrence Plots* are made from each input signal.

$$D_{i,j} = \begin{cases} \|\vec{x_i} - \vec{x_j}\|, & \|\vec{x_i} - \vec{x_j}\| \leq \epsilon \\ 0, & \|\vec{x_i} - \vec{x_j}\| > \epsilon \end{cases}$$

- Made to detect hidden dynamical patterns and non-linearities.

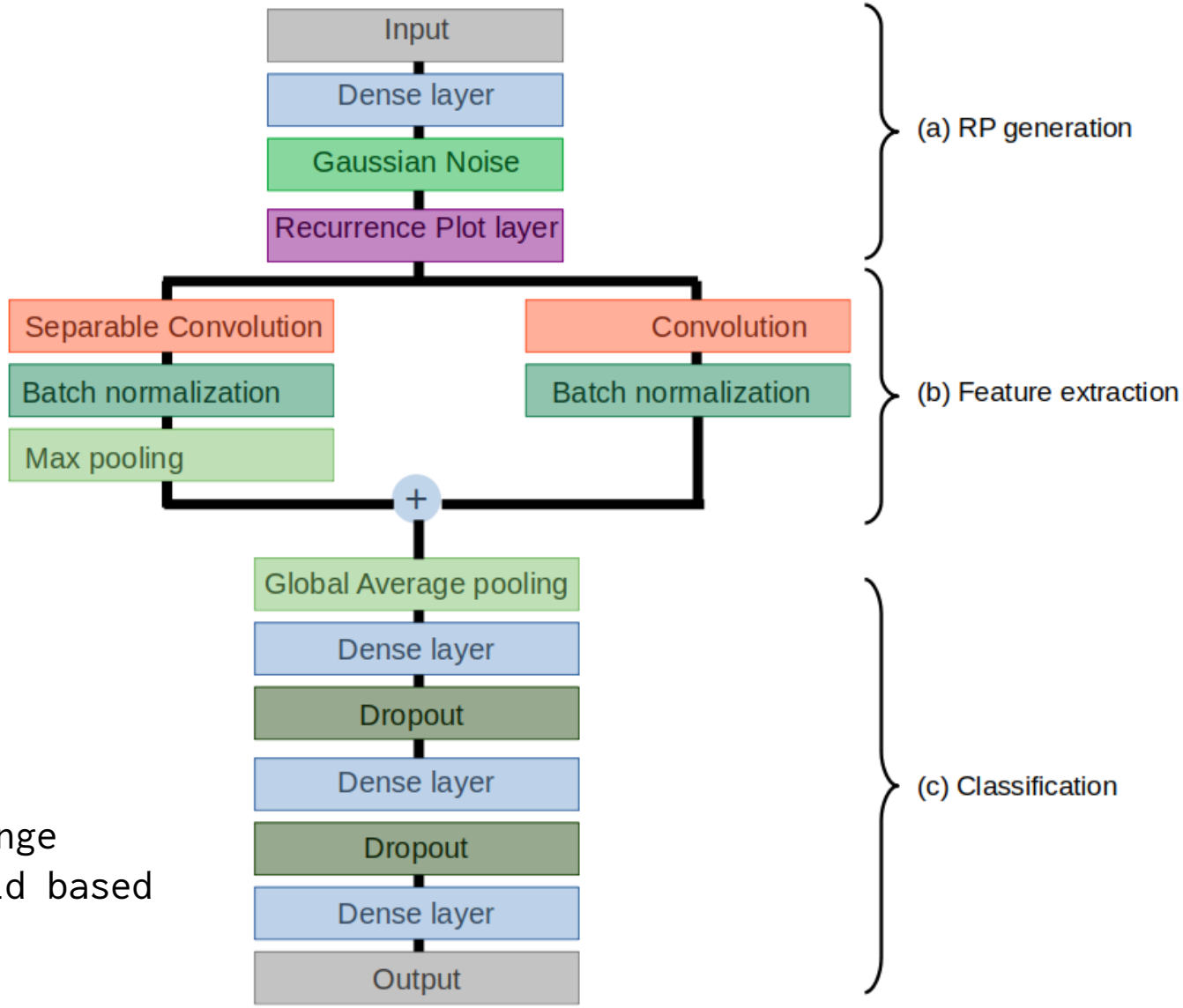
The resulting matrices can be exploited using well established image processing models like Convolutional Neural Networks.





Interlock forecasting: model

- Randomly choose 97 out of 376 channels.
- Reduce dimensionality to 20 signals.
- Add gaussian noise to the signals to reduce overfitting.
- Generate recurrence plots: Produces images of 64x64 pixels and 20 channels (“colours”).
- The model is well known in literature and used for image classification, for example in self-driving cars.

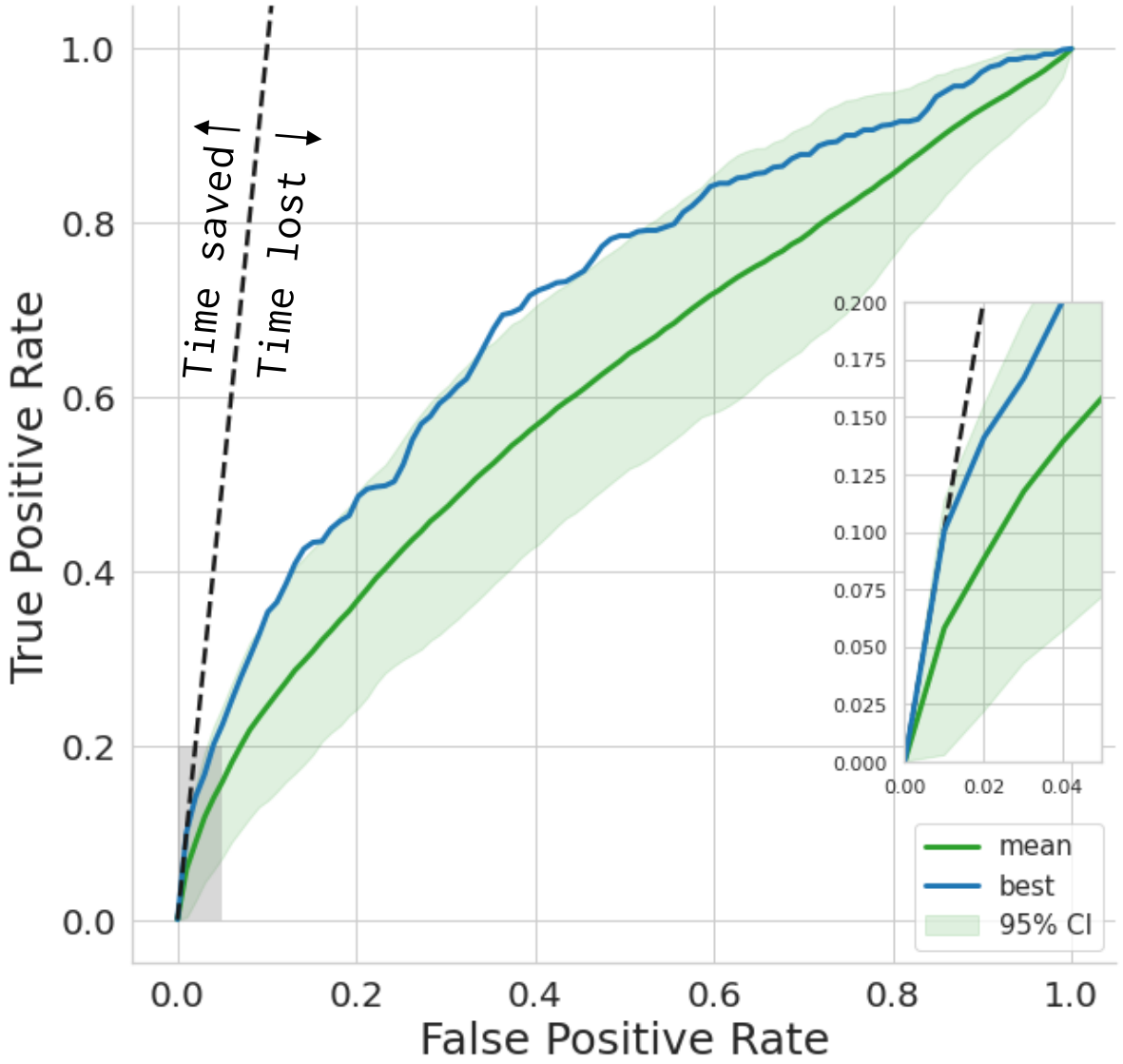


The model outputs a number in the [0,1] range transformed to a binary output using a threshold based on performance.

Interlock forecasting: results

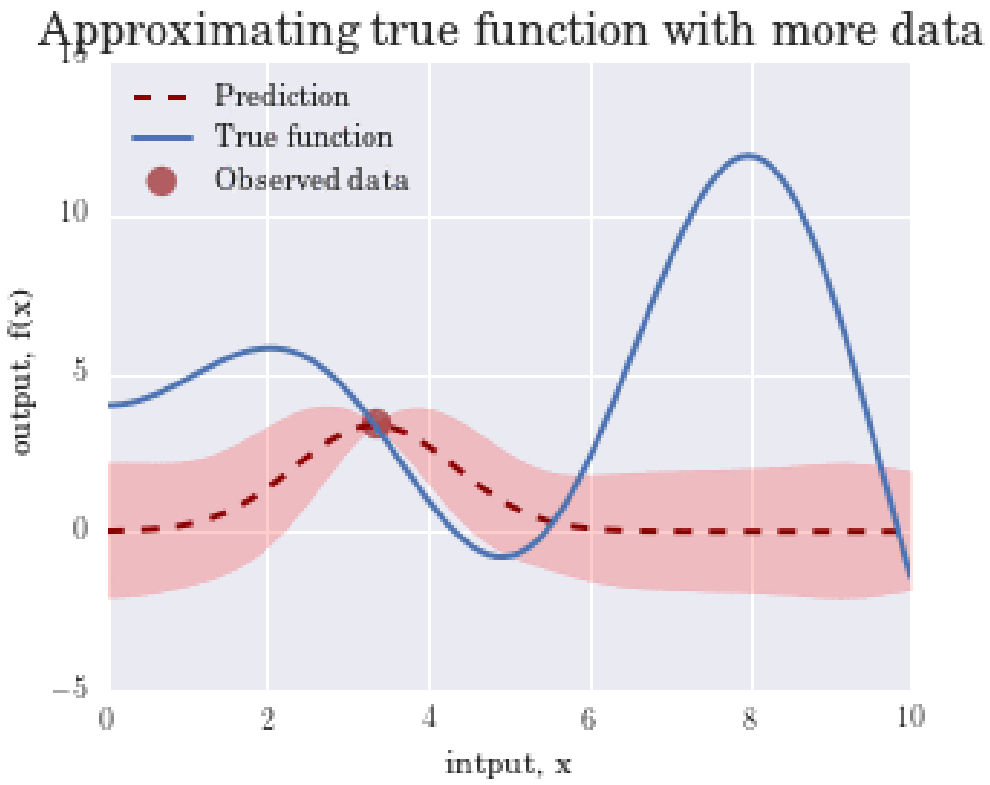
- The dataset contains 894 interlock samples.
- The best RPCNN model was able to correctly detect 40 of these samples.
- However, it also falsely classified 75 stable samples as interlocks.
- This best model could have "saved" 7.45 minutes of beam delivery. (assume that we reduce the current)

These preliminary results have been published: https://www.mdpi.com/2078-2489/12/3/121



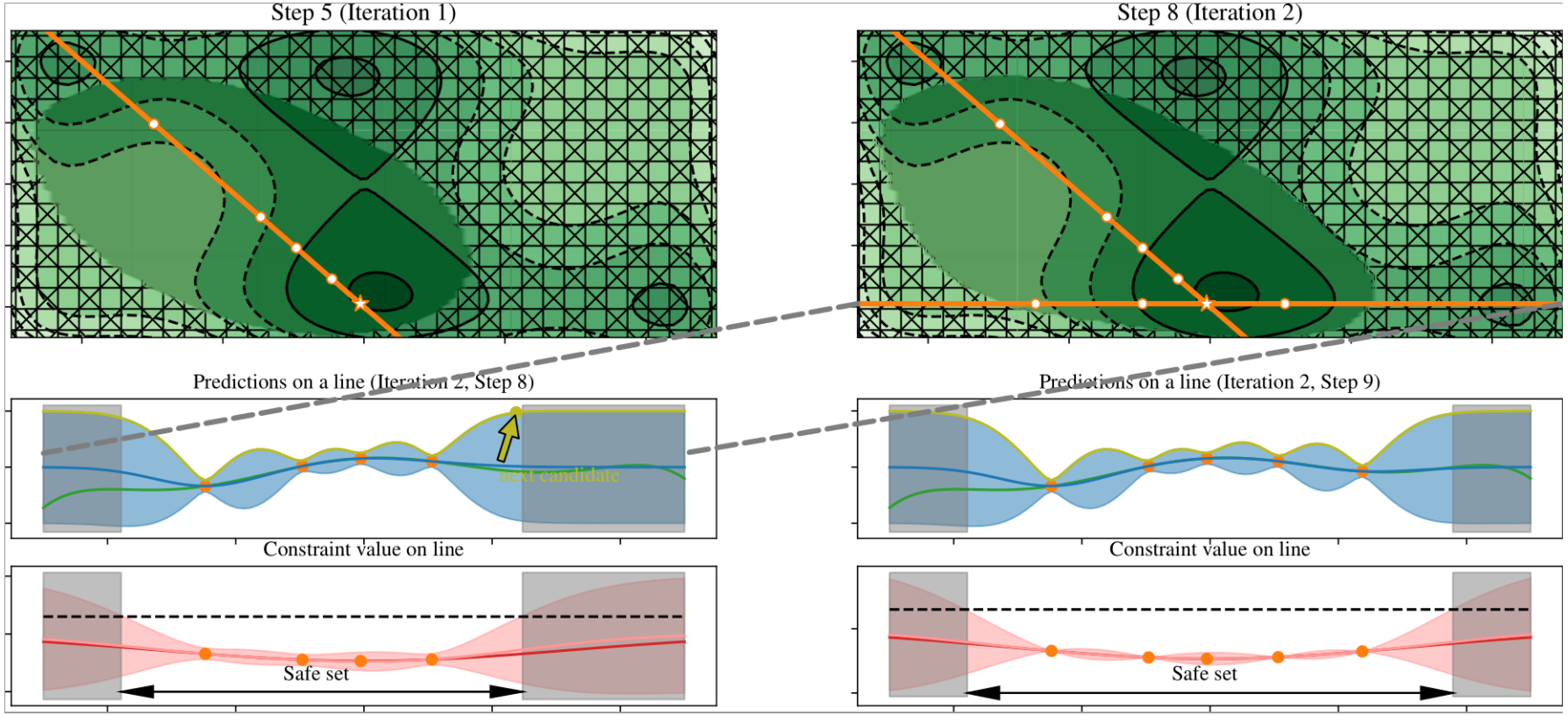
- Tuning particle accelerators is a time-consuming task for operators.
- We can use automatic methods if they can handle noisy readings and safety conditions.

- Bayesian Optimization: flexible, data-driven approach for global optimization with noisy feedback.
- Fit a statistical regression model of the data, typically a gaussian process.
- On each iteration the model looks for evaluation points that reduce the uncertainty of the unknown target function.



- Safe version of the Bayesian Optimization: LineBO.
- A constraint function is also evaluated and provides a *safe* region.
- The next evaluation is always done in the safe region which expands with more information.

Example of 2 iterations of LineBO on a 2 dimensional target function:



Beam loss optimization: Outline and methods

…beam losses …interlock limits

- Both the **target** function and a **constraint** function are measured on each step:
  - Tries to maximize the target function.
  - Keeping the constraint function (and its confidence boundaries) under certain threshold.

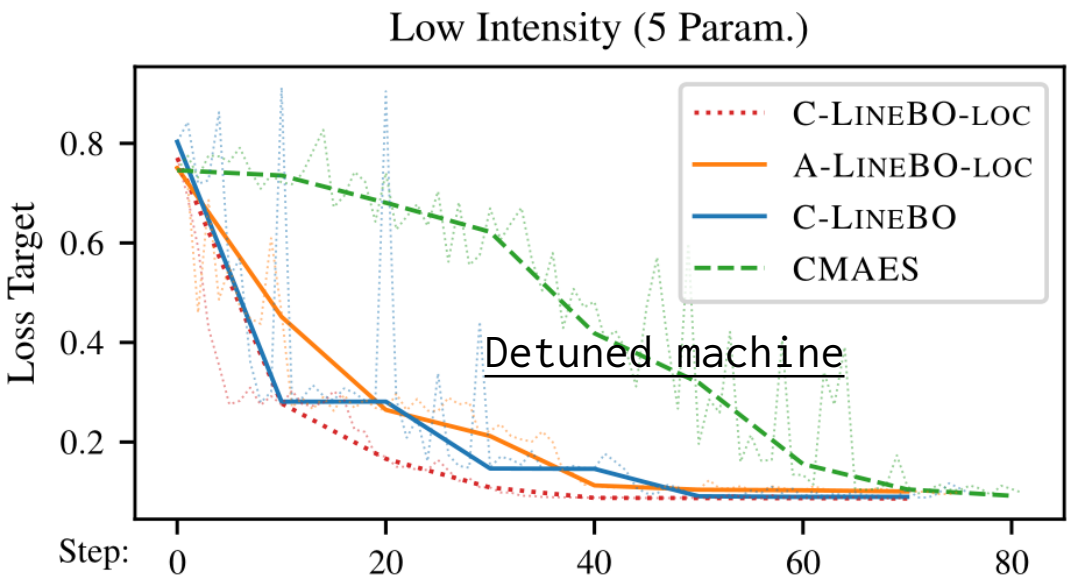
Variants:

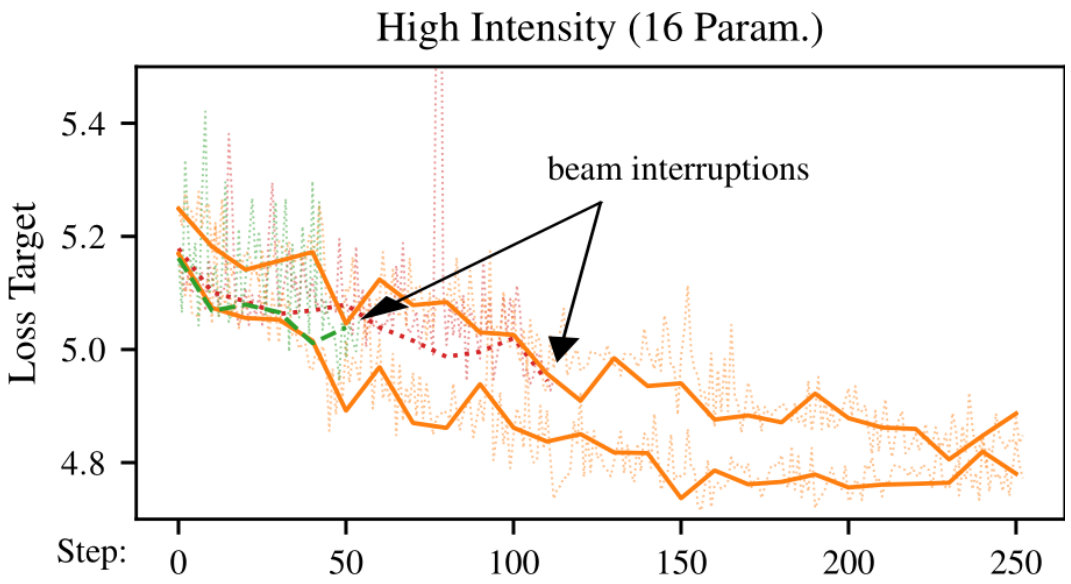
- **C-LineBO**: Find each parameter target optimum individually.
- **A-LineBO**: Move each parameter to find the “best-direction” and then search for the best point in that direction.
- ***-Loc:** “Localized” Maximum step size is 10% of the available range.
- **CMAES**: Similar to swarm optimization. Does not have safety constraints. Standard well developed global optimization *evolutionary* algorithm. Used here as baseline to compare the rest of the methods.

Beam loss optimization: Experiments

- Tested on HIPA in several beam development slots.
- Target: Weighted average of about 60 loss monitors all around the machine.
- Constraints: Closest distance of about 200 signals to the beam interruption *warning* levels.
- Parameters: Field strength of quadrupoles in the 870 keV line.

- All methods reach the same approximate solution.
- However, as CMAES is unconstrained it is likely to hit unsafe configurations.
- A-LineBO-Loc reaches the best solution most consistently.

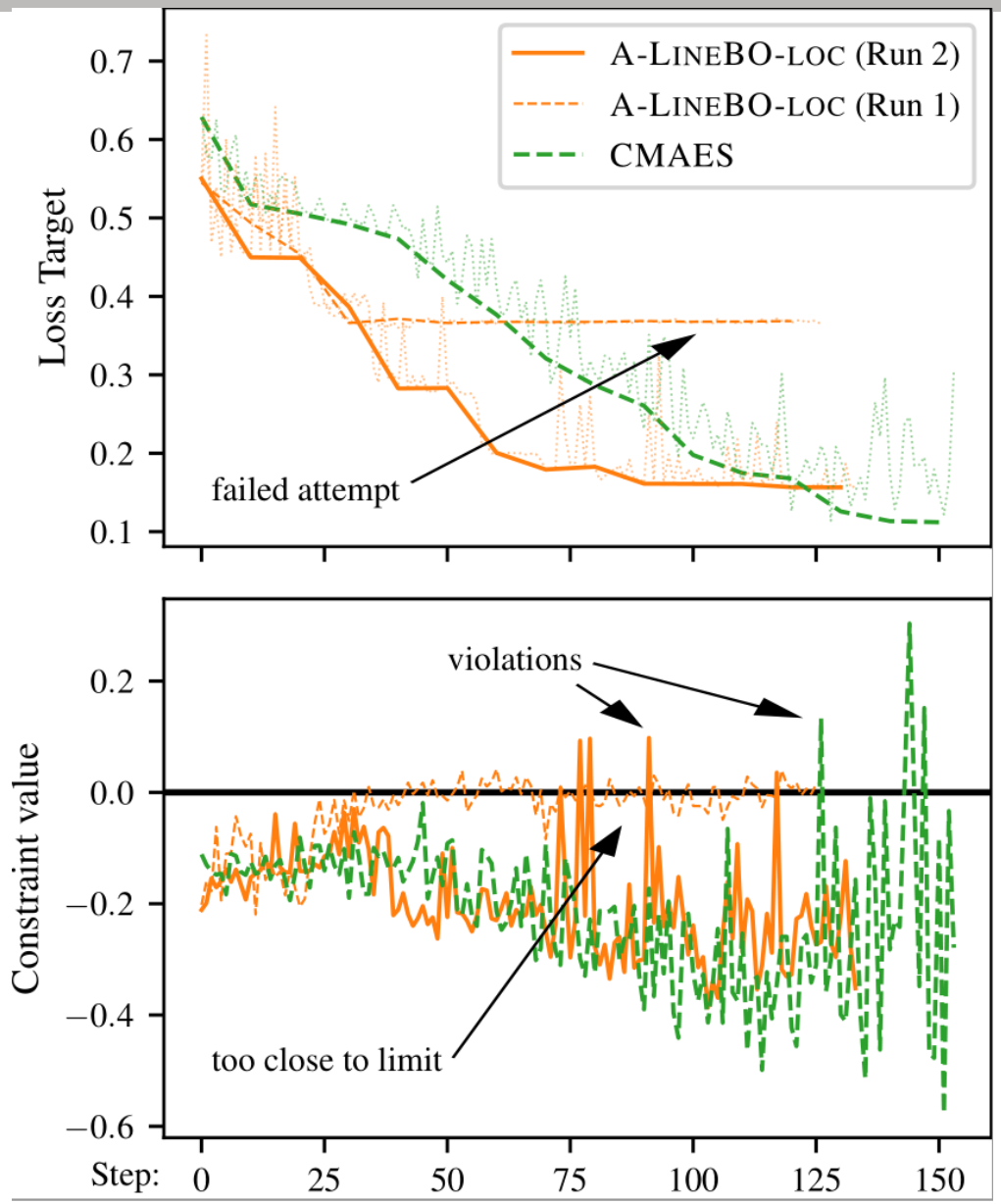




Beam loss optimization: Constraints

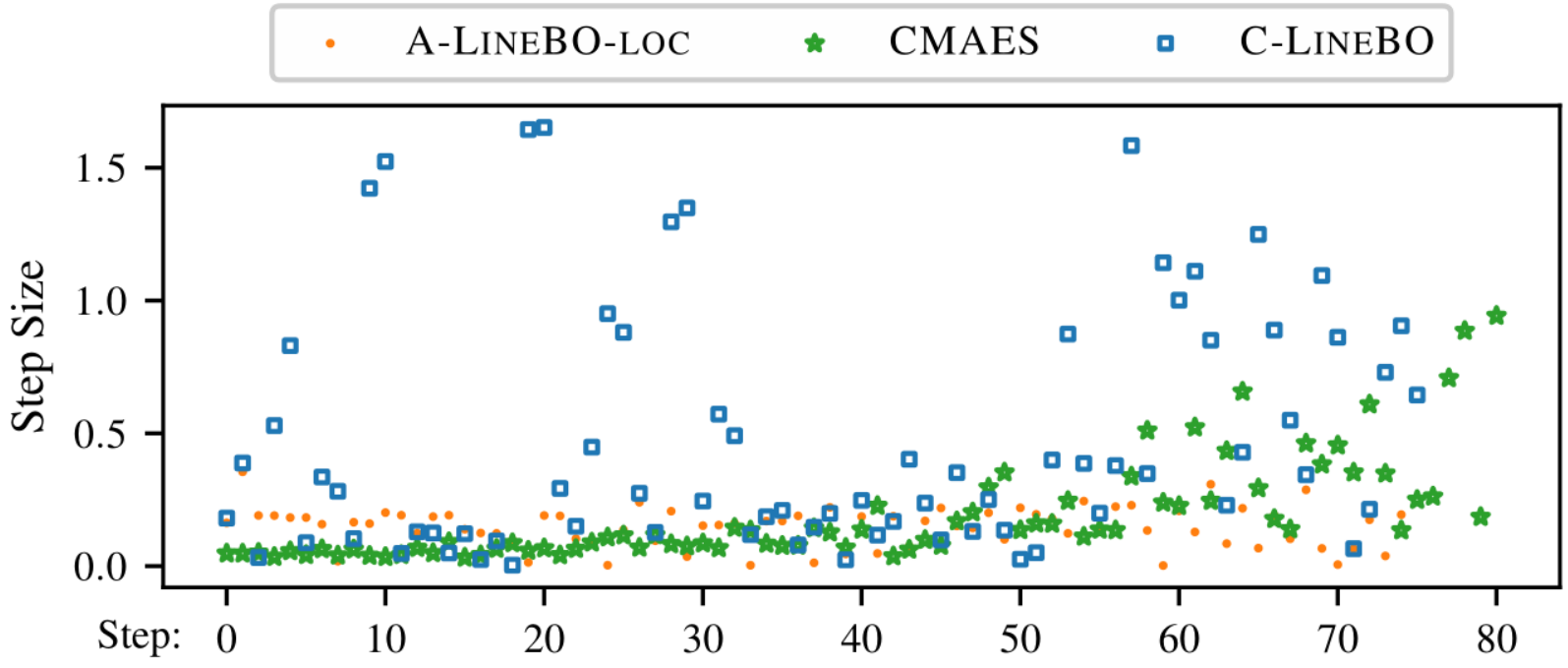
- Tested on HIPA in several beam development slots.
- Target: Weighted average of about 60 loss monitors all around the machine.
- Constraints: Closest distance of about 200 signals to the beam interruption *warning* levels.
- Parameters: Field strength of quadrupoles in the 870 keV line.

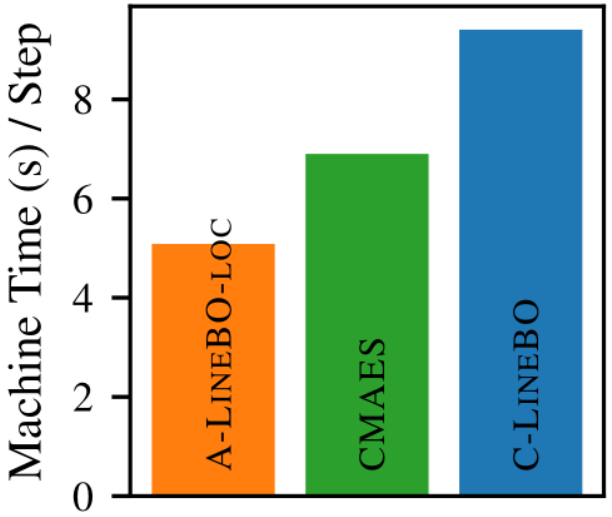
- A-LineBO-Loc manages to keep the constraint function under the limit except few cases caused by noise.
- CMAES also reaches the solution but produces large constraint violations at the end.
- A-LineBO-Loc can get stuck without safe space to explore.



Beam loss optimization: Step sizes

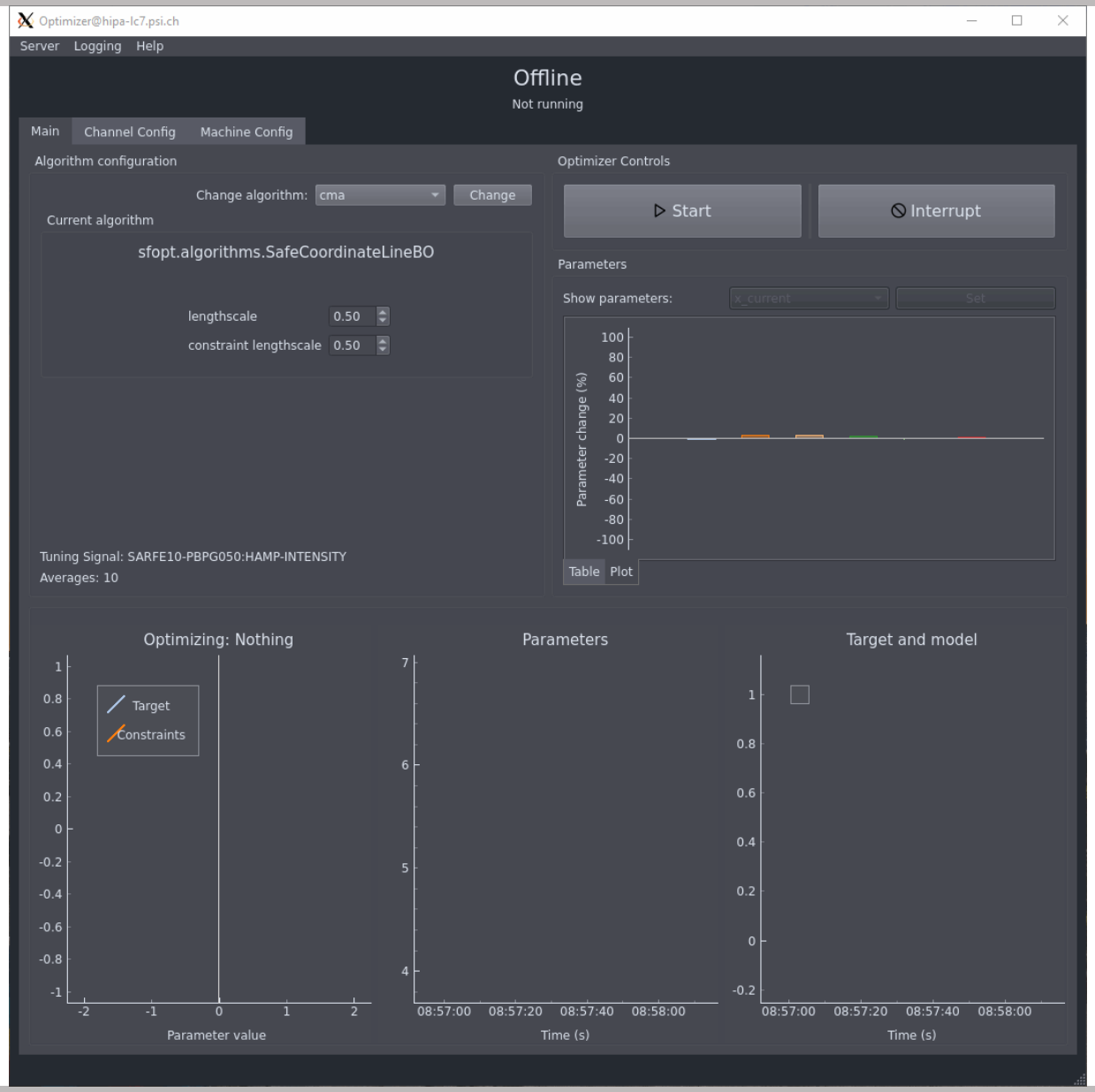
- The non-localized methods take much larger steps.
- A-LineBO-Loc manages to reach similar solutions while taking smaller steps.
- The smaller step sizes make the optimization faster: Larger steps increase waiting times for feedbacks (beam centering, current regulation, etc.) or make them unable to follow the changes.





Beam loss optimization: Experiments

- GUI developed to make the optimization algorithm easy to run and set up:
  - Algorithm selection and configuration.
  - Parameter selection.
  - Constraints definition.
  - Waits for feedbacks configuration.
- Provides step-by-step feedback to the user in case any of the algorithms misbehave.



Beam loss optimization: Conclusions

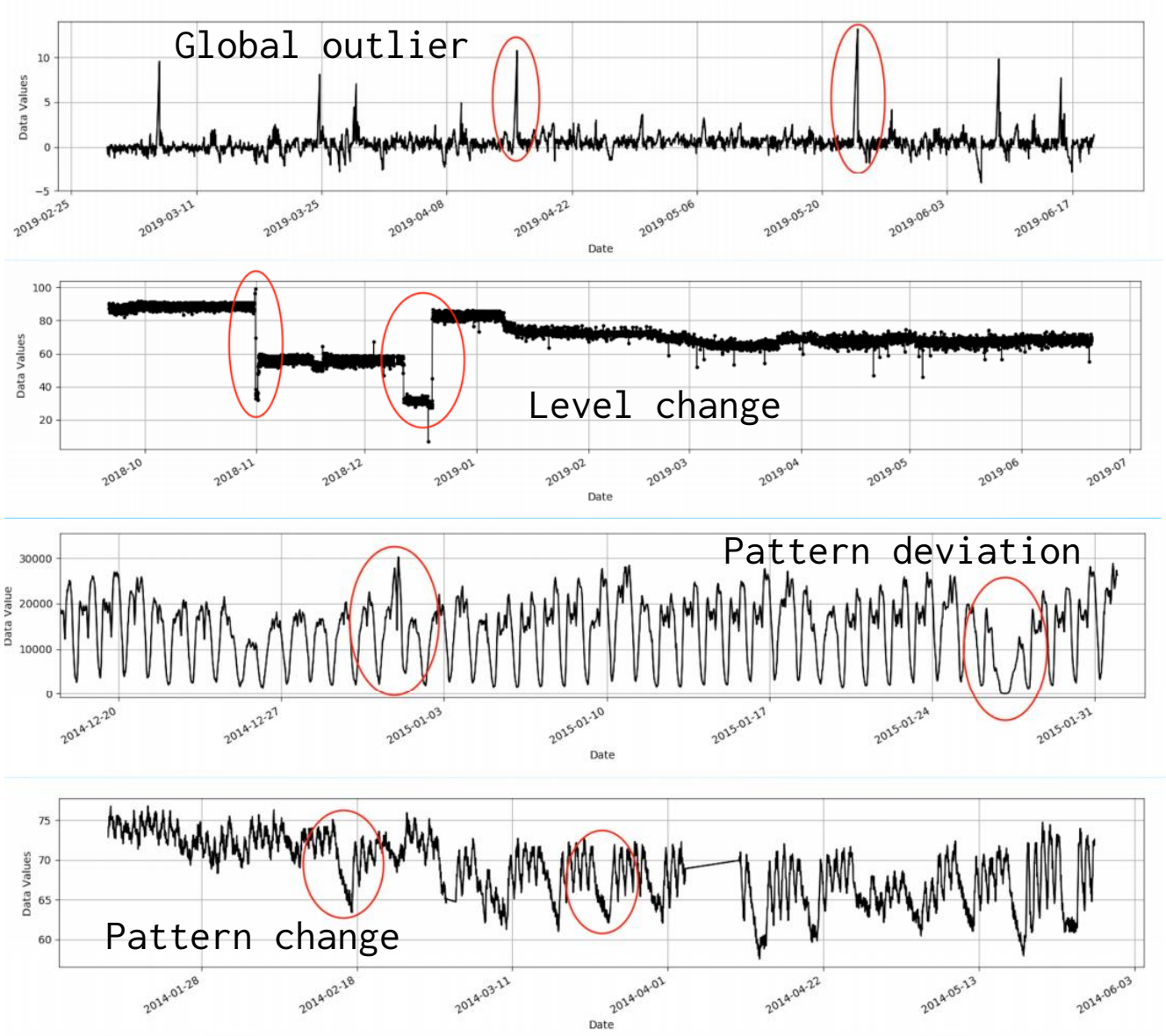
- The loss optimization algorithms and software framework are well tested and ready to use.
- The concept is machine independent and has already been used at HIPA and SwissFEL.
- The algorithms reach similar levels of losses to human operators, in a relatively short time and producing few interlocks.
- A paper with these results have been submitted for publication.

Anomaly Detection: Motivation

"An observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism."

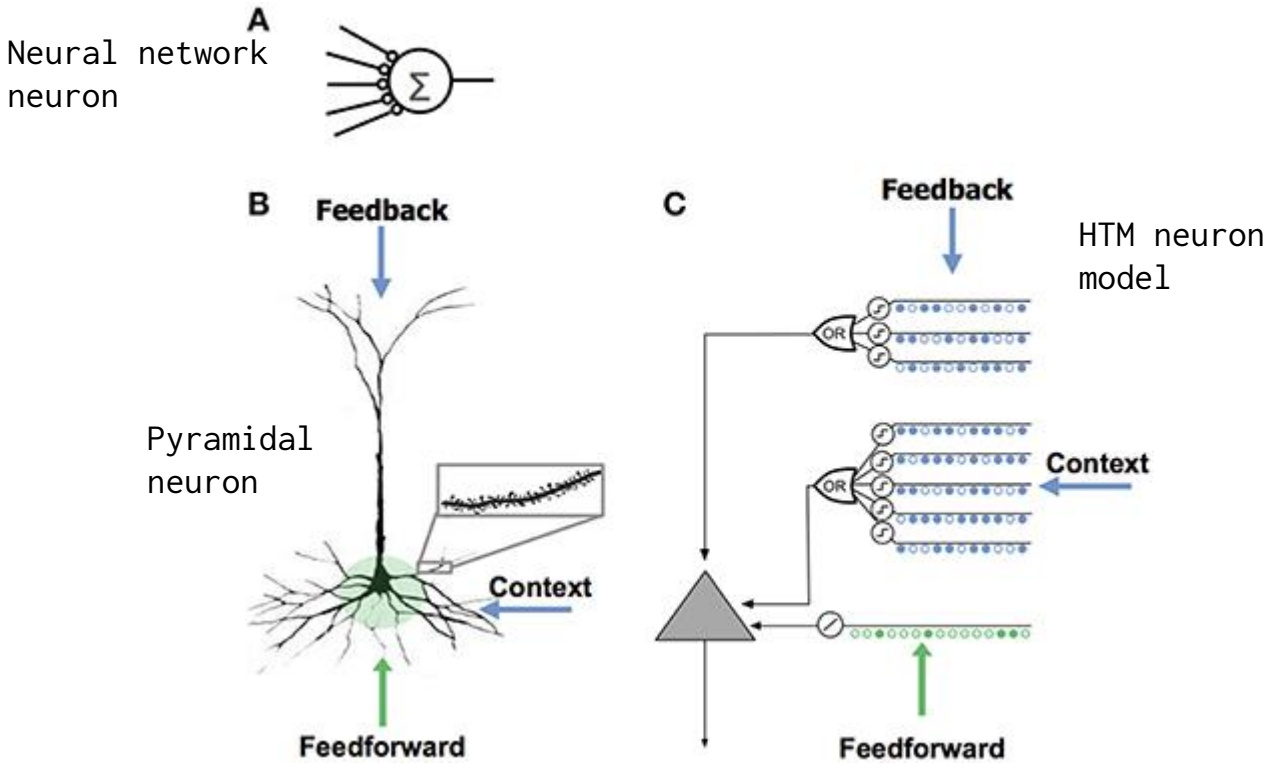
— Hawkins, 1980

- Try to find observation or sequences that deviate from the "normal behaviour".
- Experts would recognize these anomalous patterns easily but cannot be monitoring the huge amount of data some systems produce.
- E.g: Credit card fraud detection, intrusion detection in cybersecurity, or **fault diagnosis in industry**.
- Specific e.g: At HIPA the MHB7R:ILOG:2 "Blende" (aperture) loss monitor broke down without anyone noticing.

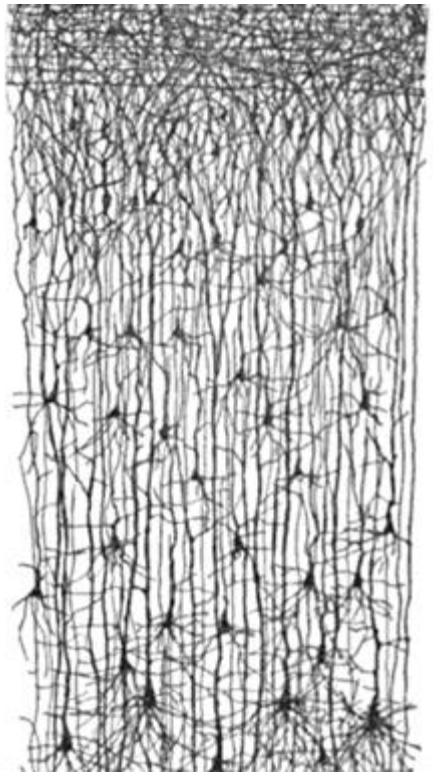


*Plots from: Ted D'Ottavio et al, **Experience Using NuPIC to Detect Anomalies in Controls Data**, ICALEPCS 2019*

- Model of the organization of “pyramidal neurons” in the neocortex of mammals.
- Tries to explain how neuronal structures remember sequences.



Pyramidal neurons connect forming “columns” that share input and output.



J. Hawkins and S. Ahmad, ***Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex***
https://doi.org/10.3389/fncir.2016.00023

Anomaly detection: HTM and LSTM

- The model we developed is the combination of 2 anomaly detection models:
  - The HTM and a LSTM autoregressor.

HTM Algorithm

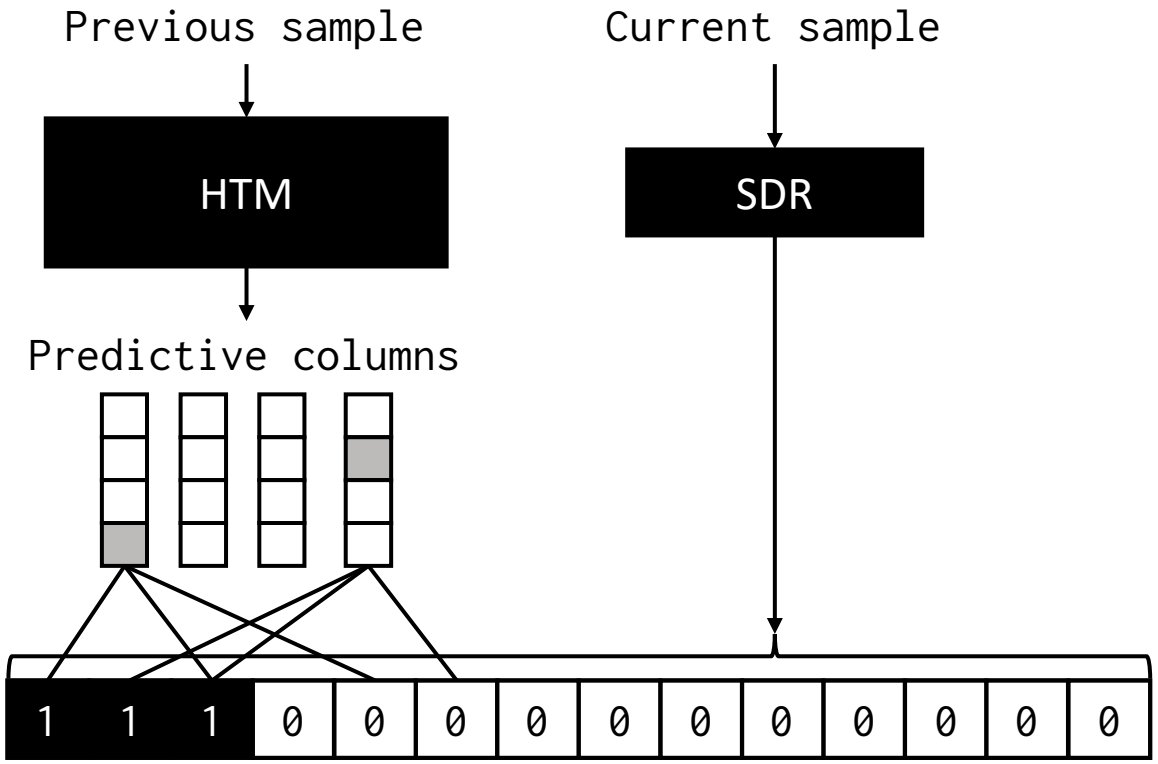
- Model of the neocortex in mammals.
- Consists of three phases:
  - Sparse representation of input, encode the input numeric values in sparse binary vectors.
  - Spatial Pooler: Recognizes and encodes similar spatial patterns in the input.
  - Temporal Memory: Recognizes and predicts temporal patterns from the spatial patterns.
- Very noise resistant and high capacity sequence memory.

LSTM Autoregressor

- A type of recurrent neural network.
- Used in speech recognition and translation.
- Configured as an autoregressor, tries to predict what the next value will be based on previous values.
- The output is discretized in *bins.* This way the model can predict many possible outcomes at the same time.

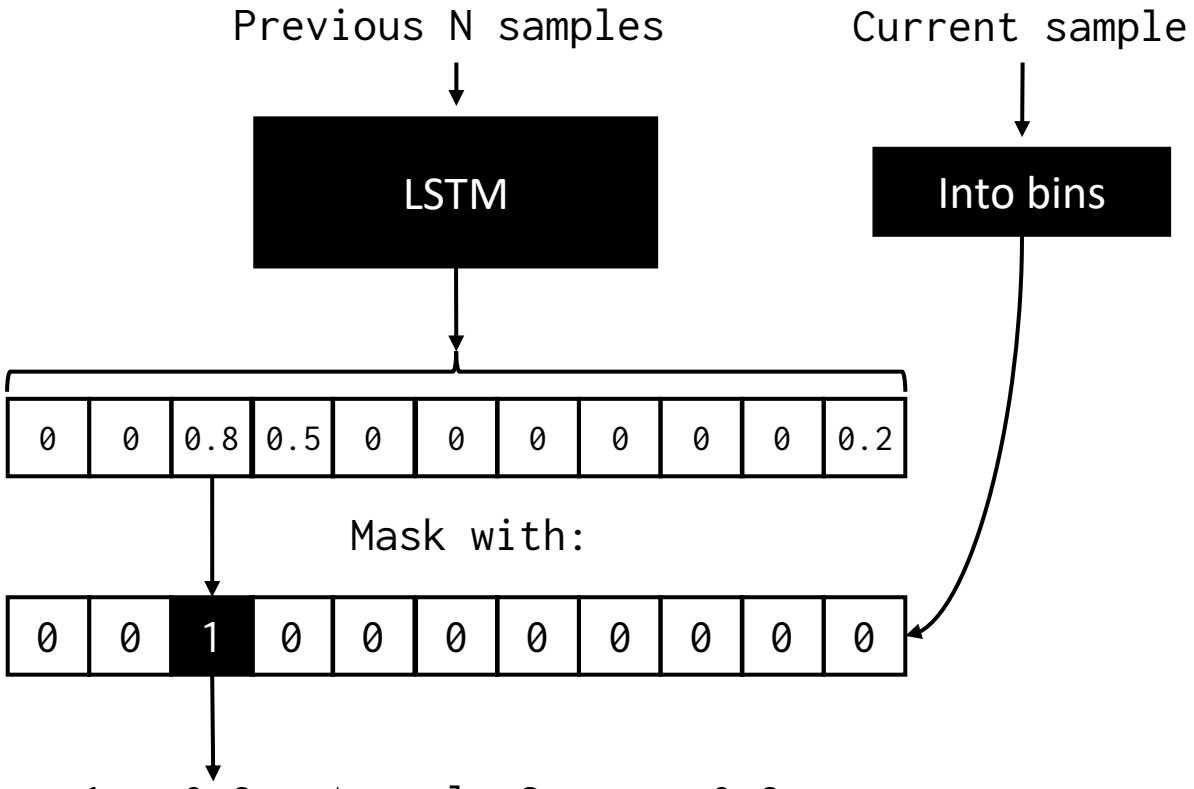
Anomaly detection: HTM and LSTM ensemble model

HTM Anomaly Score:



$$\text{Anomaly Score} = \frac{\text{Correctly predicted inputs}}{\text{Number of connections}}$$

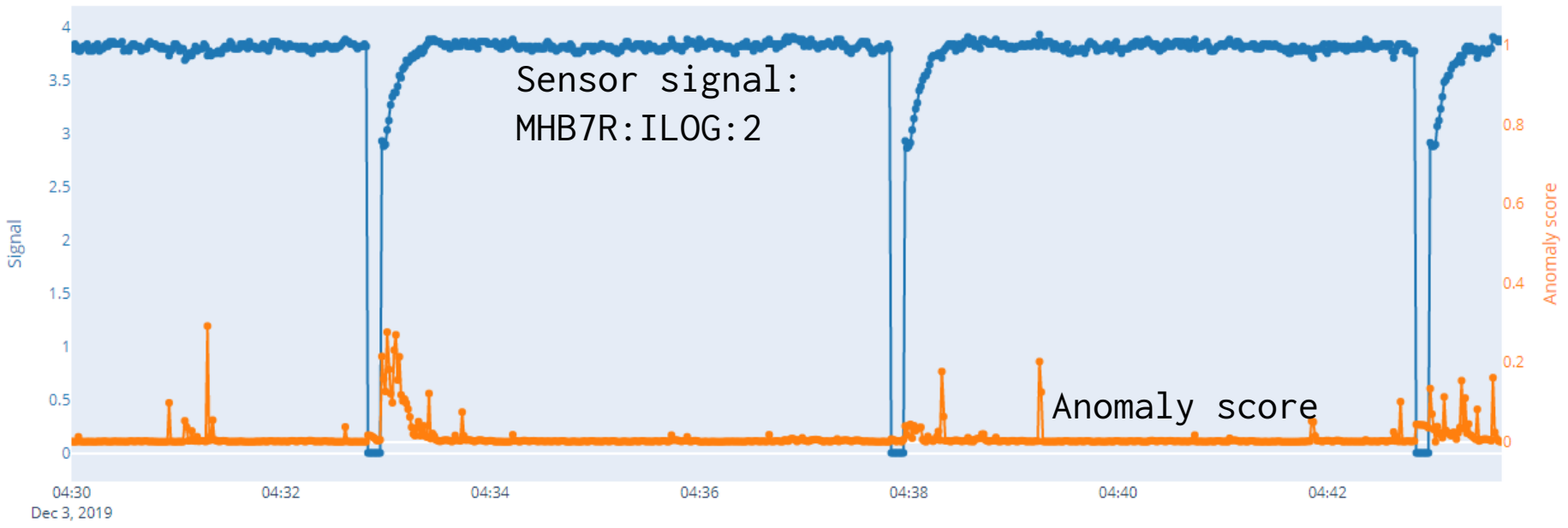
LSTM Anomaly Score:



1 − 0.8 = Anomaly Score = 0.2

$$\text{Training penalty} = \text{Anomaly Score} + \gamma \sum |prediction|$$

Anomaly detection in HIPA



History

Start 2019-12-03 04:30:00

End 2019-12-03 06:00:00

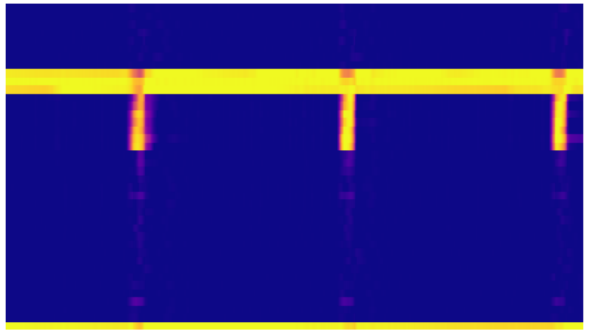
Replay from here?

Refresh

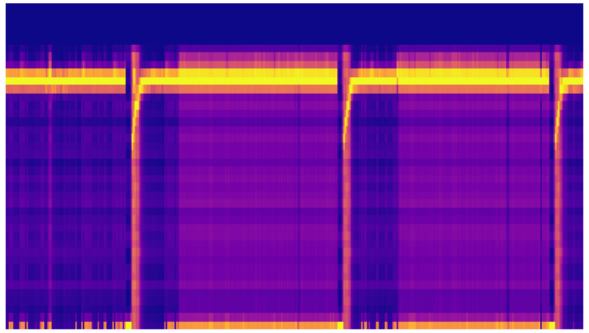
No alarms

Normal

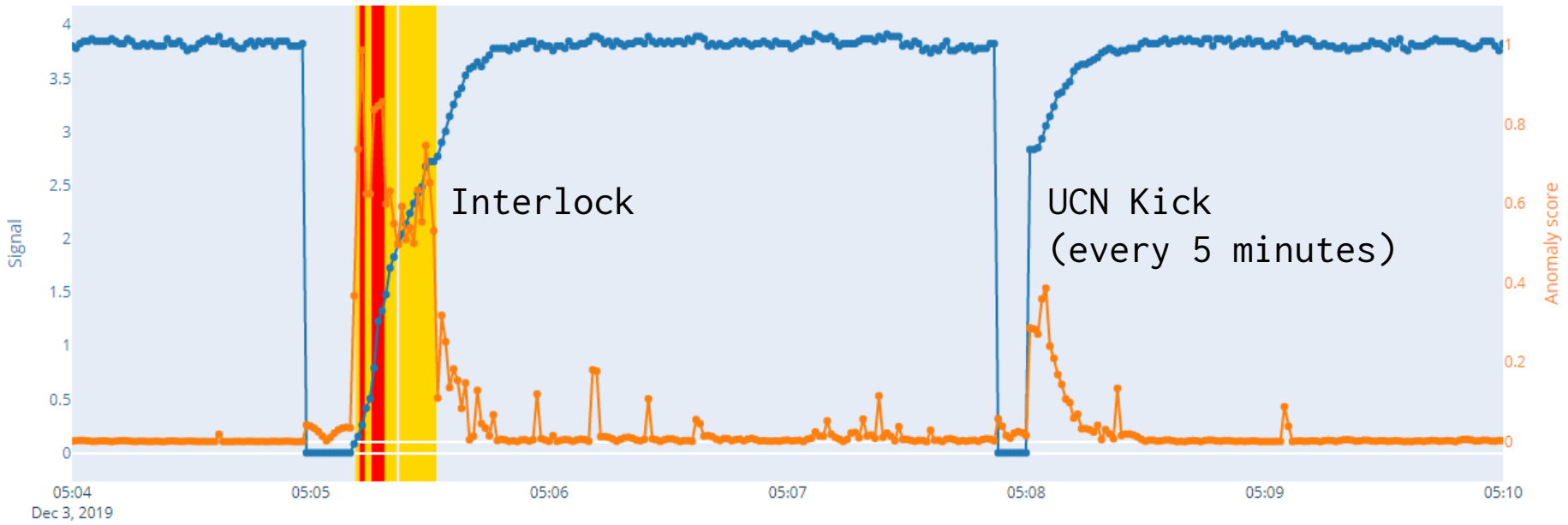
LSTM probabilities



NUPIC probabilities

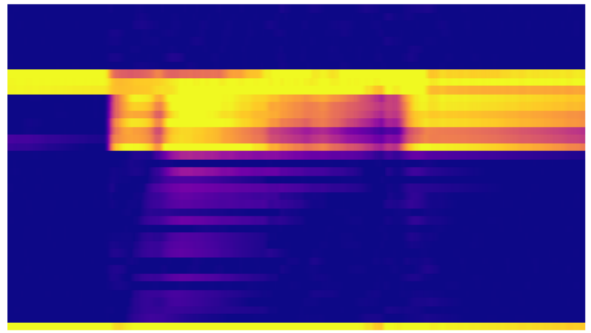


Anomaly detection in HIPA

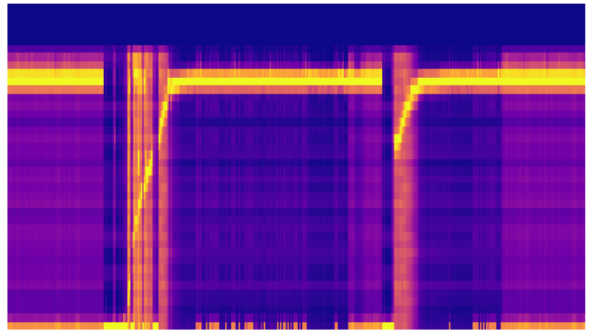


Normal

LSTM probabilities



NUPIC probabilities

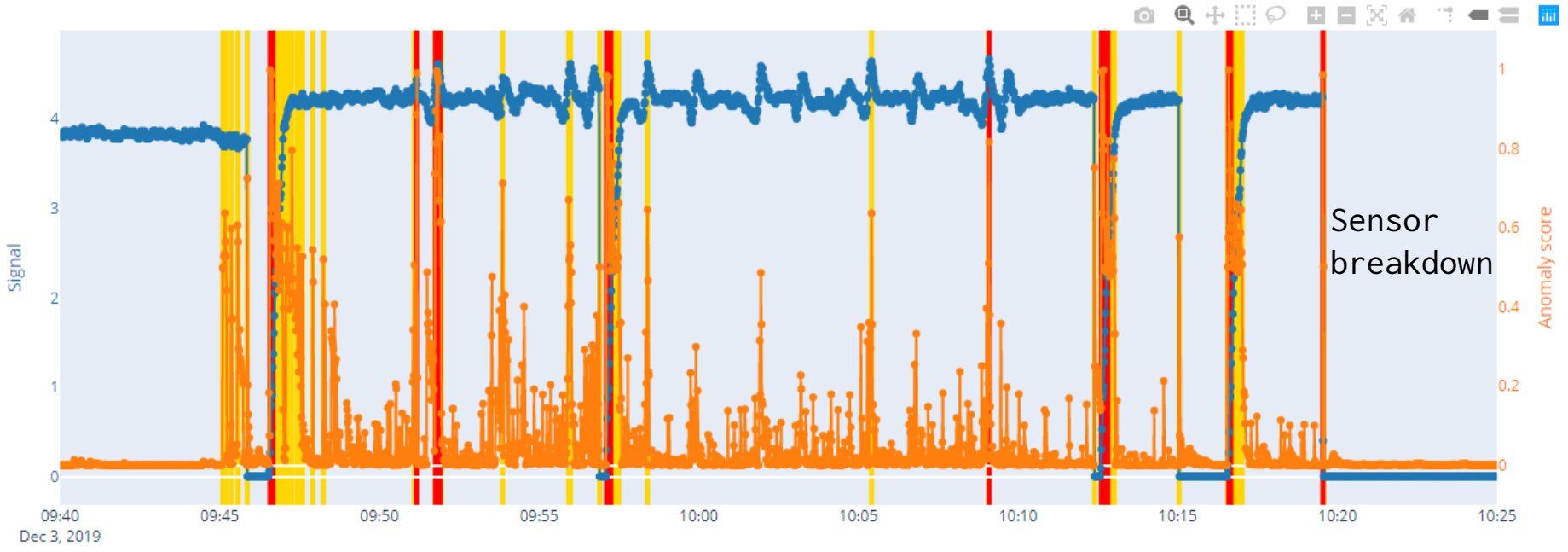


History

Start	2019-12-03	05:04:00	Replay from here?
End	2019-12-03	05:10:00	Refresh

No alarms

Anomaly detection in HIPA



History

Start 2019-12-03 09:40:00

End 2019-12-03 10:25:00

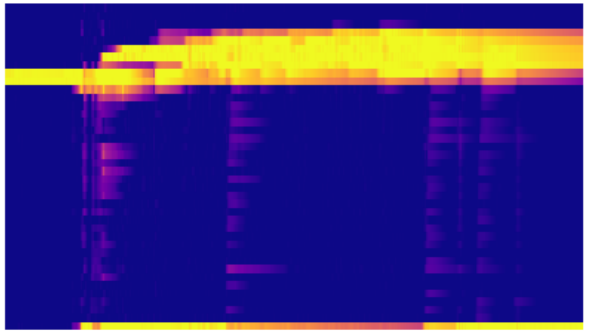
Replay from here?

Refresh

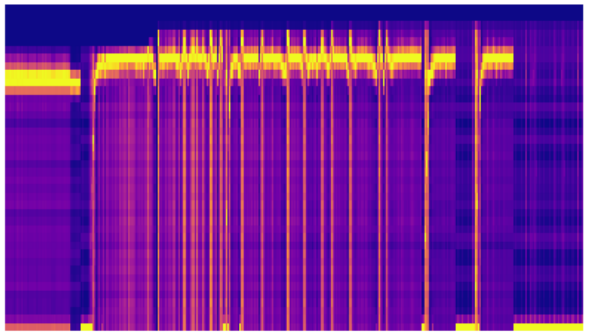
No alarms

Normal

LSTM probabilities



NUPIC probabilities



Live anomaly detection in PROSCAN

- Anomalies detected live in the vacuum level in the cyclotron:

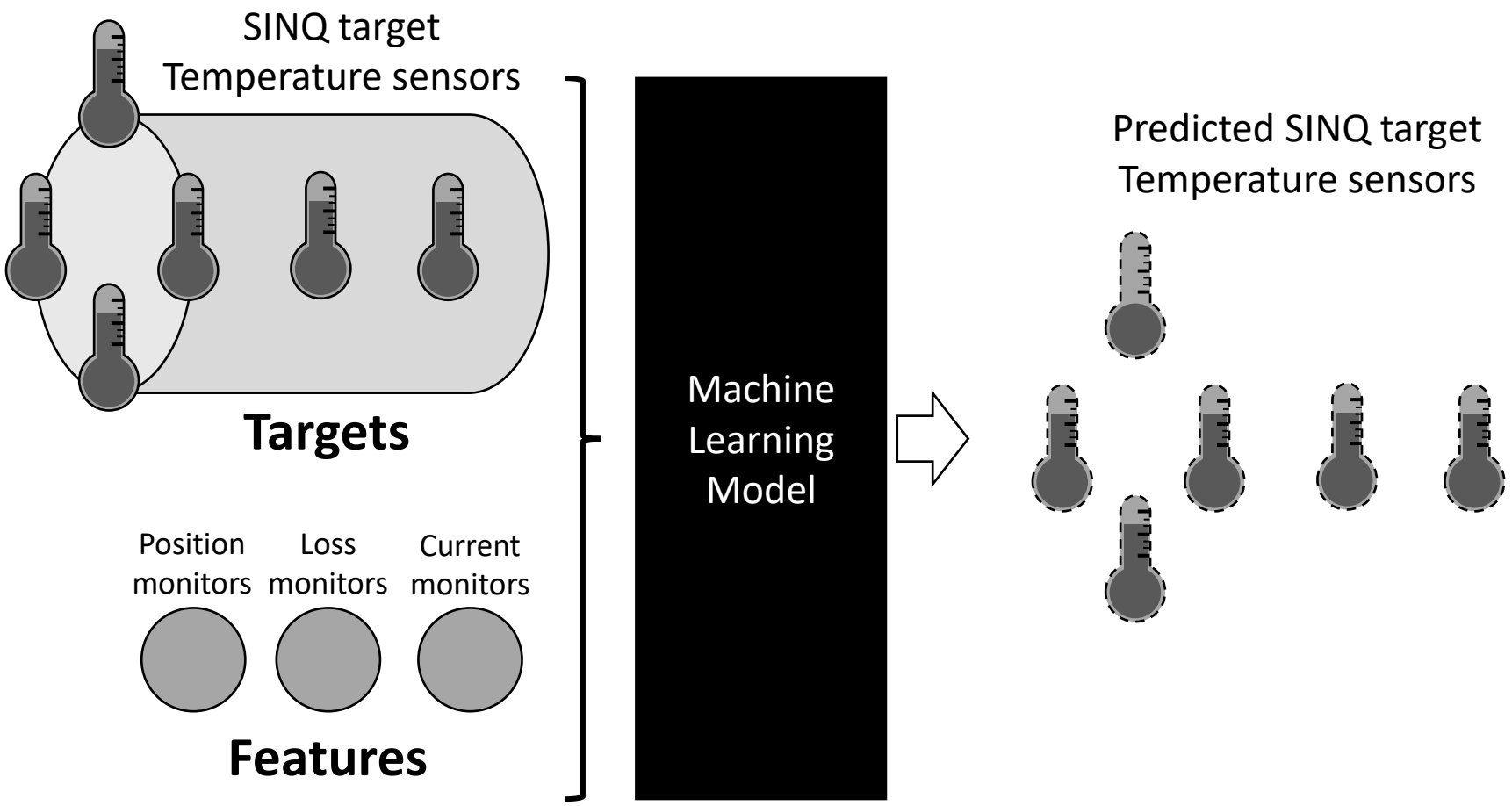
Anomaly detection conclusions

- The combination of the HTM network and the LSTM seems to provide a good real time anomaly detection system.
- The system works well on archived and live data, adapts to both stable and periodic behaviour.
- It would have alerted the experts of the failure of the MHB7R sensor tenths of minutes in advance.
- Tested in real time on both HIPA and PROSCAN and ready to use in production.
- For the MHB7R sensor, ignoring the first two hours and operational changes, on the day of the failure one single false positive was reported by the model.
- These results have been collected and a paper is ready to be submitted for publication.

- Machine learning can learn the behaviour of diagnostic devices and replace them (Virtual Diagnostics):
  - Measurement devices that interfere with the beam.
  - Degrading devices that are expecting to fail.

Proof of concept:

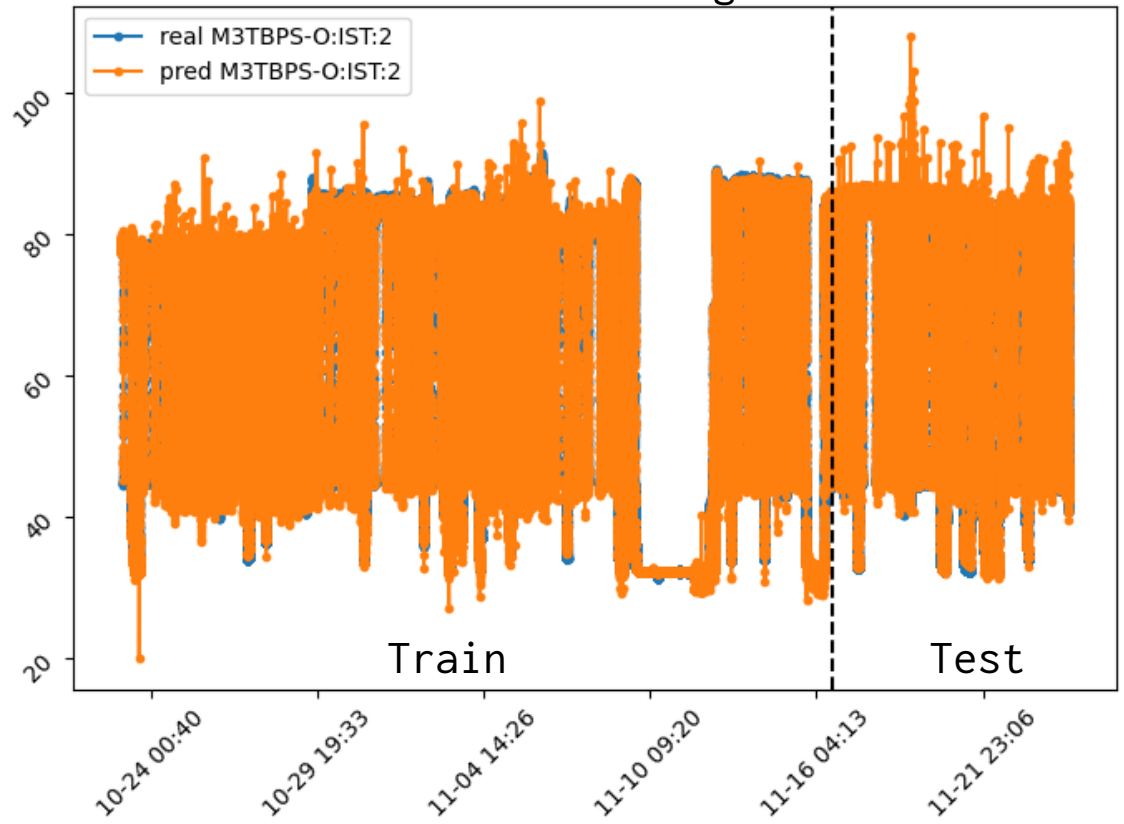
SINQ target
temperature sensors



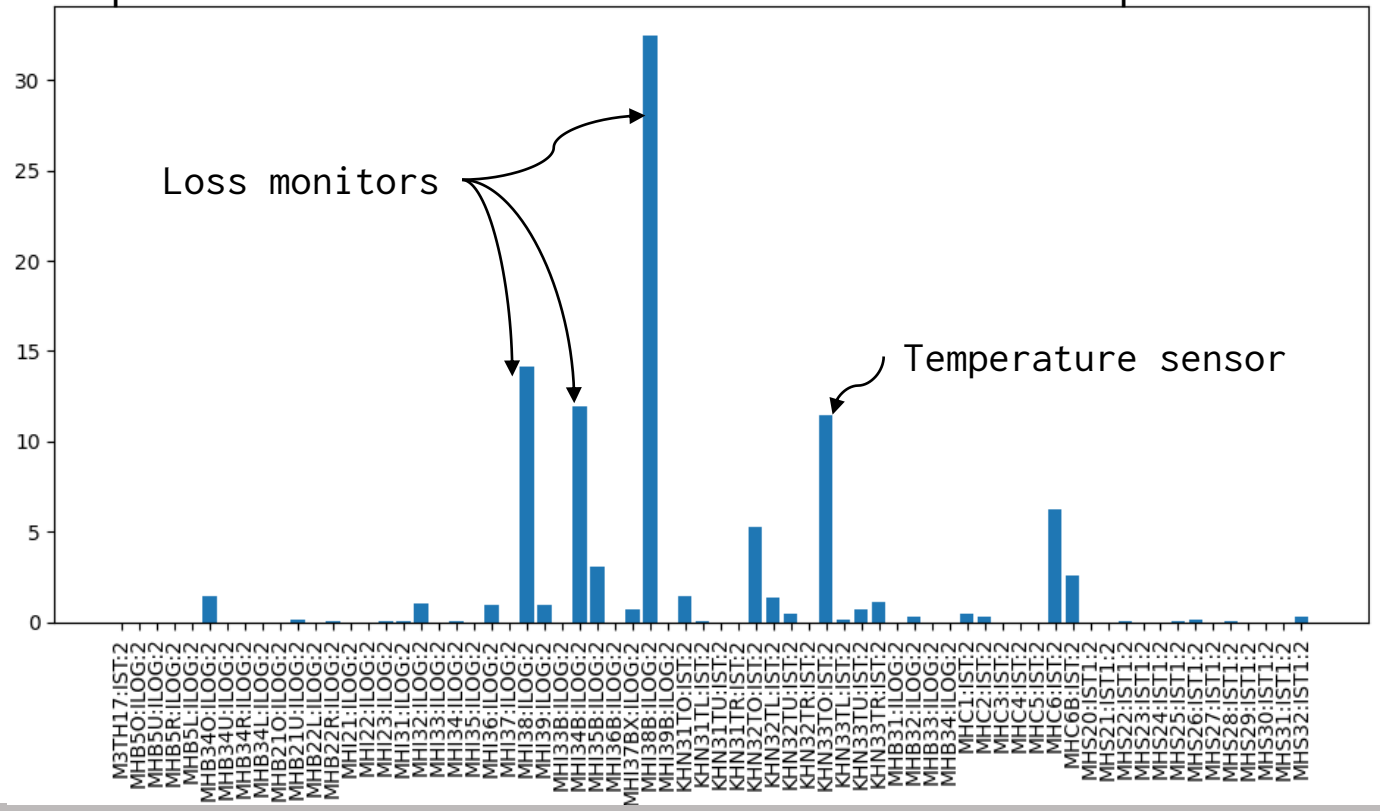
SINQ temperature sensors: Virtual diagnostics

- Model for M3TBPS-O:IST:2 (one of the four external sensors).
- Uses 68 monitors all over the machine, many in the SINQ line.
- The model is a Gradient Boosting model (tree ensemble) made with CatBoost.

Prediction vs. actual reading of M3TBPS-O:IST:2

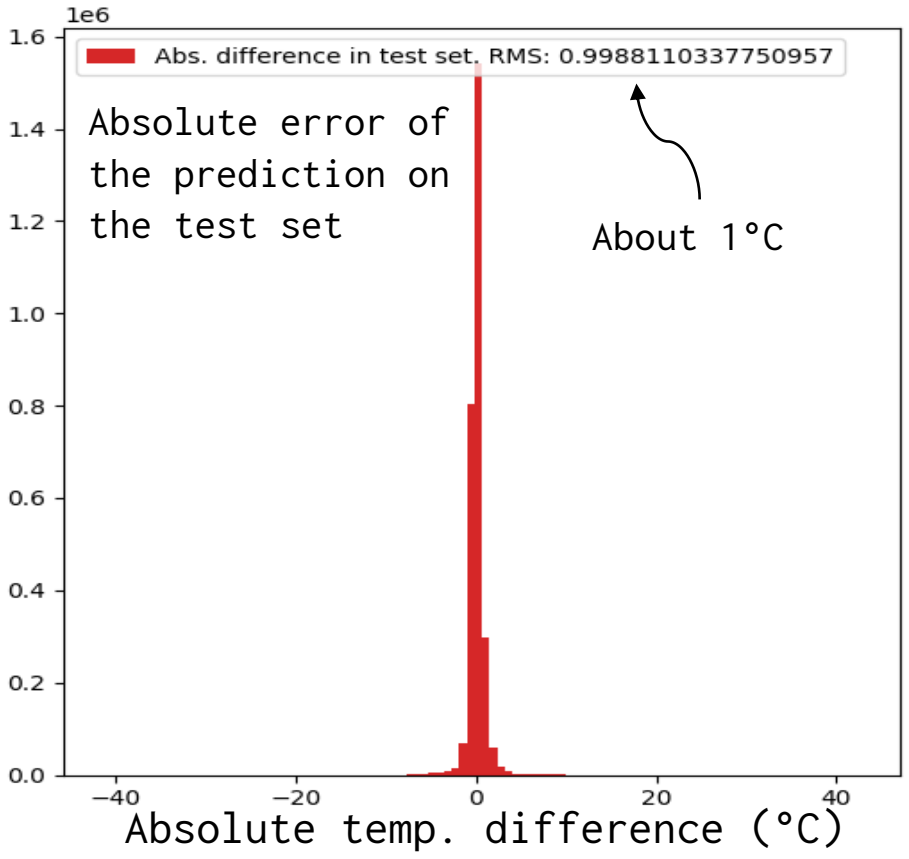


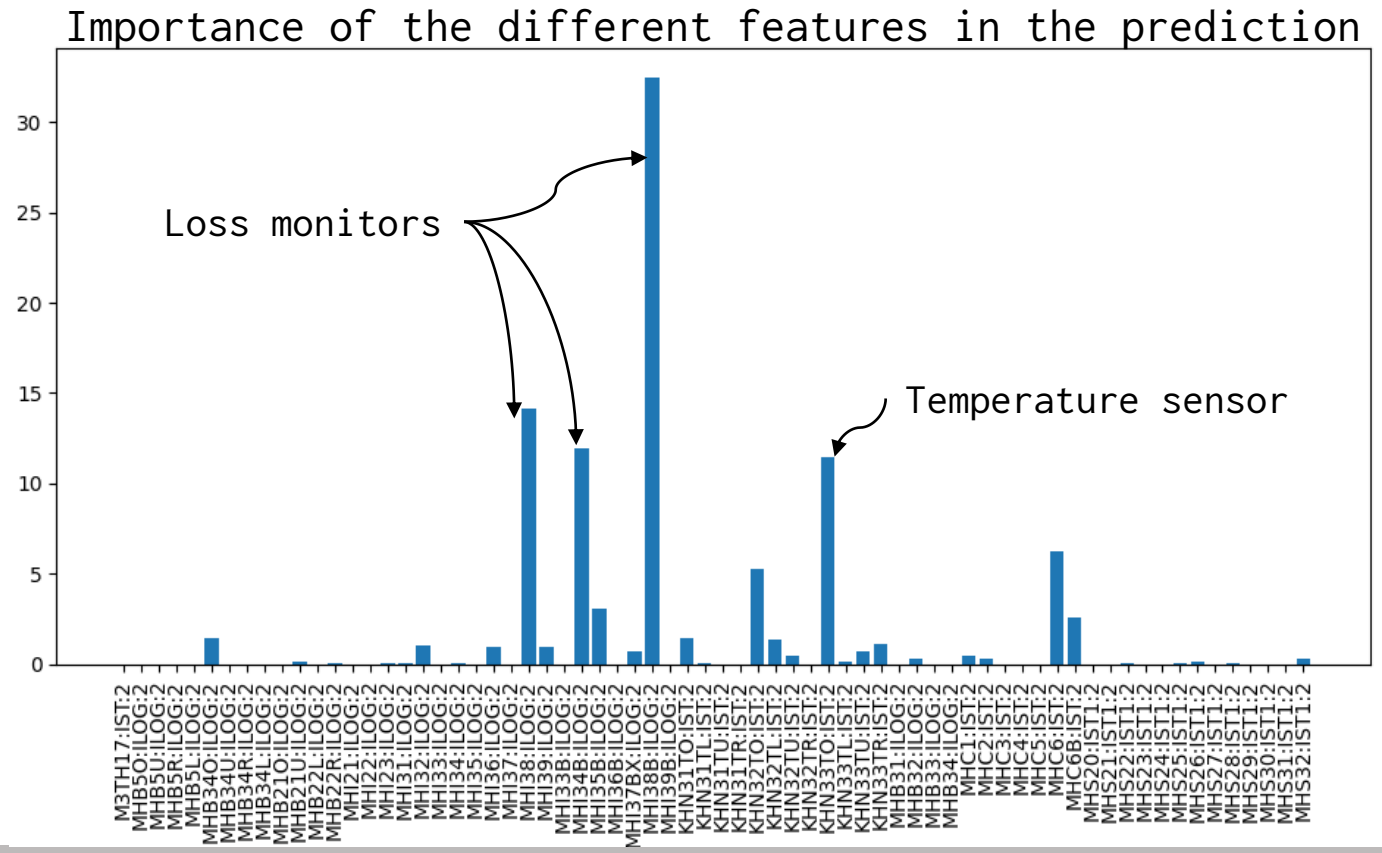
Importance of the different features in the prediction



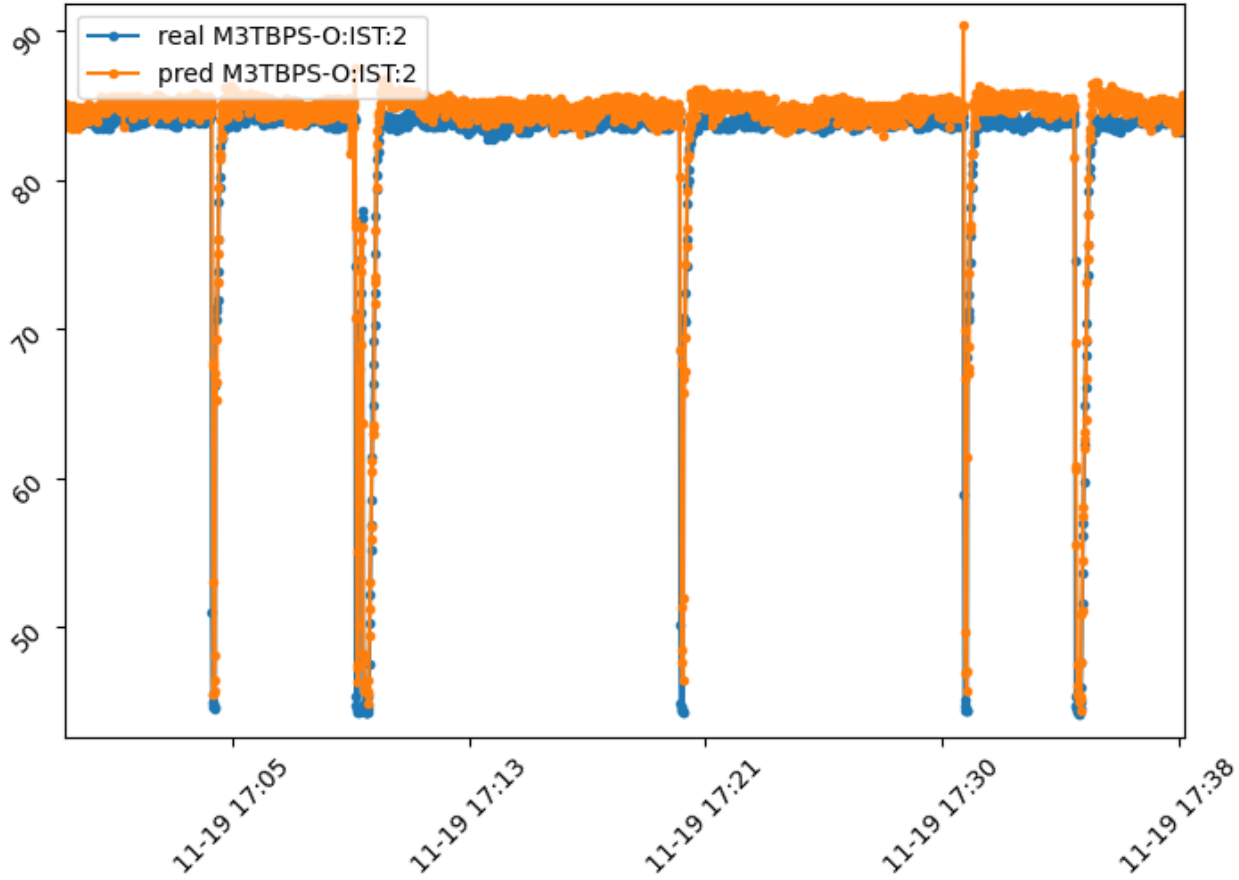
SINQ temperature sensors: Virtual diagnostics

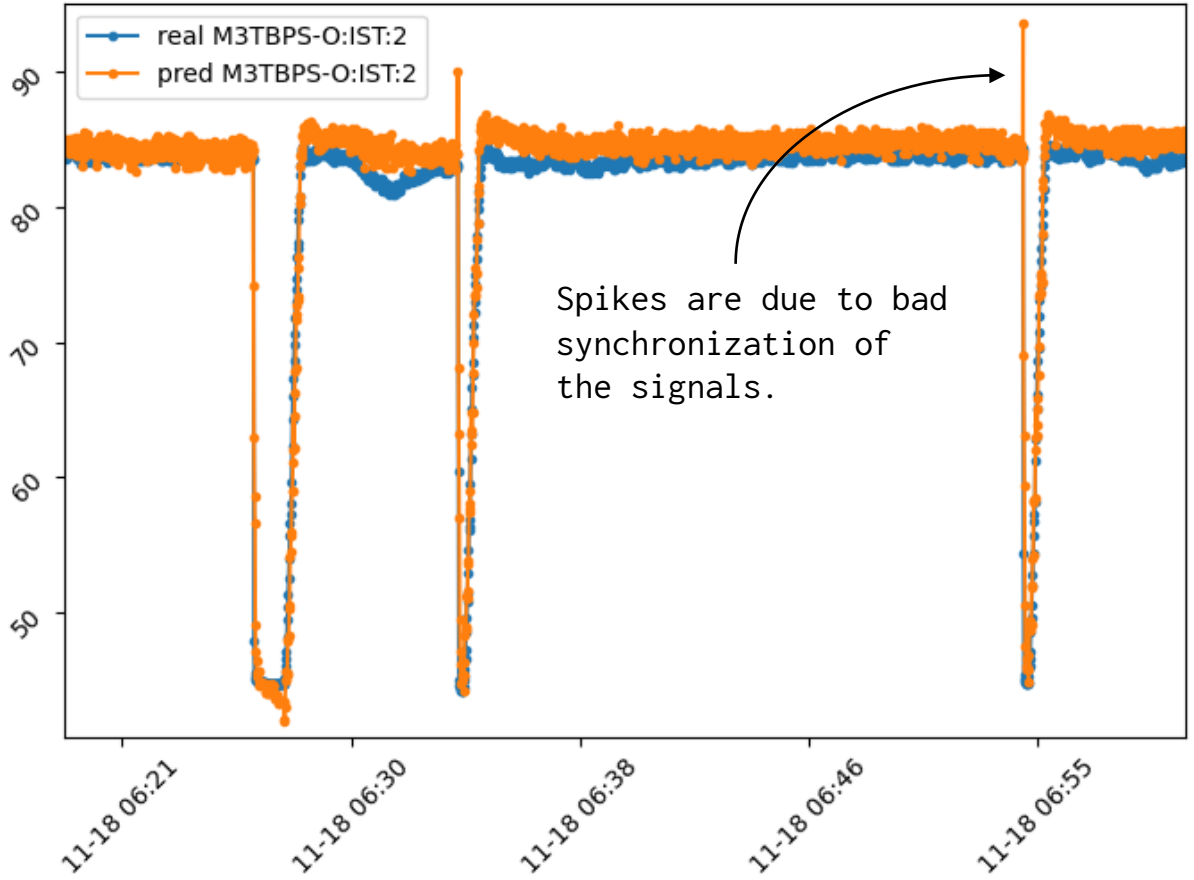
- Model for M3TBPS-O:IST:2 (one of the four external sensors).
- Uses 68 monitors all over the machine, many in the SINQ line.
- The model is a Gradient Boosting model (tree ensemble) made with CatBoost.





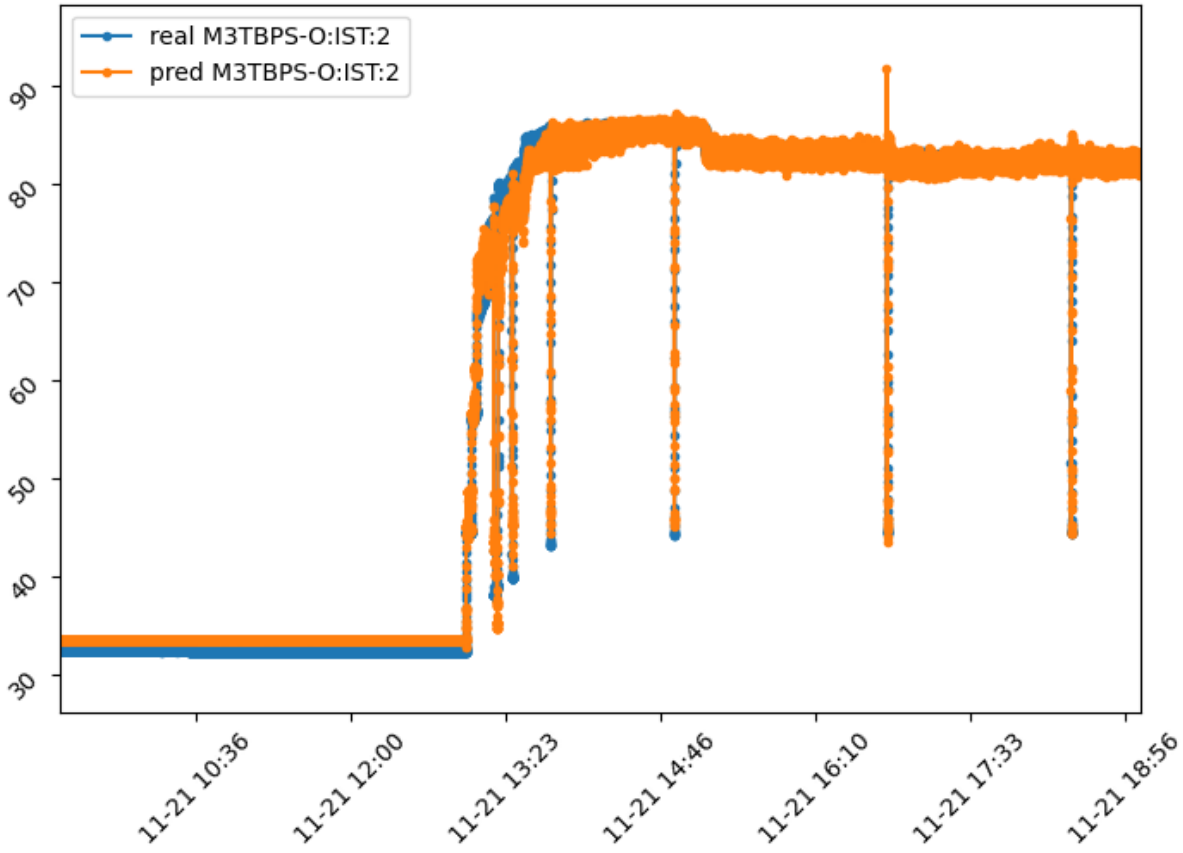
Zooming in some examples in the test set:

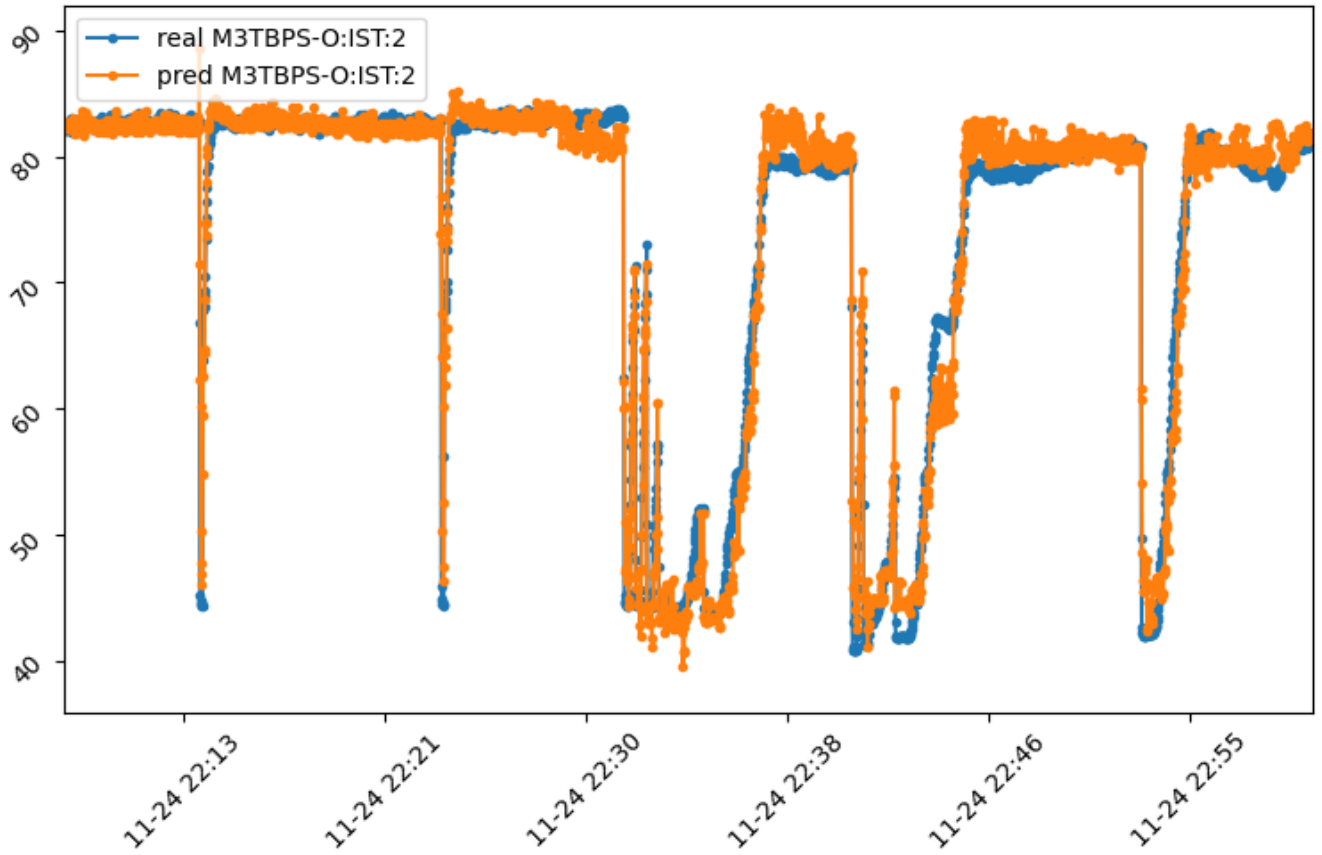




Spikes are due to bad synchronization of the signals.

Zooming in some examples in the test set:





SINQ temperature sensors: Virtual diagnostics

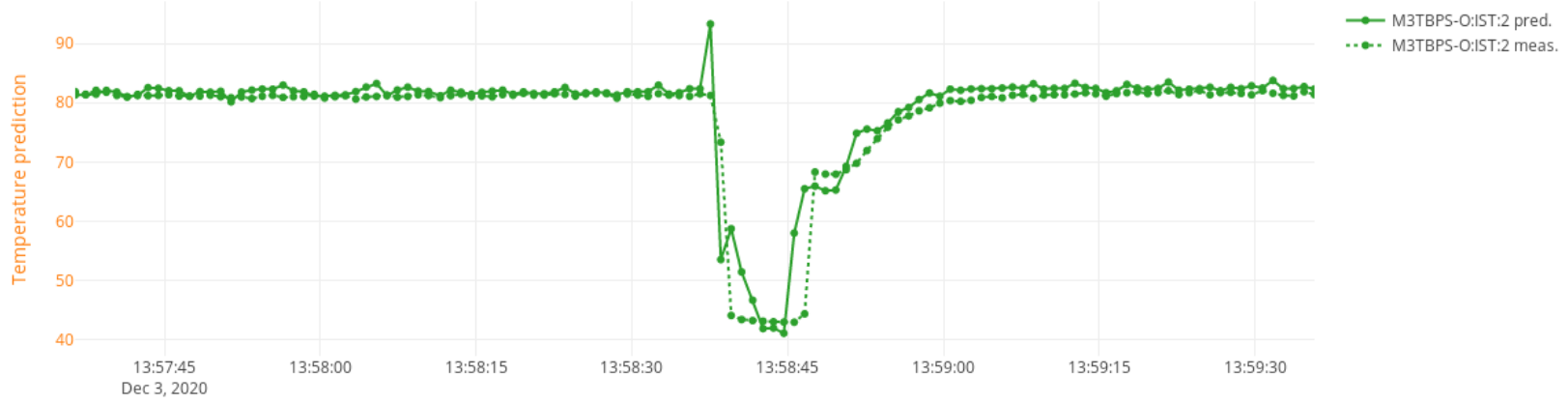
SINQ temperature predictions

History Live

Show measured values

M3TBPS-L:IST:2 M3TBPS-R:IST:2 M3TBPS-O:IST:2 M3TBPS-U:IST:2 M3TBPS-Z12:IST:2 M3TBPS-Z14:IST:2

Model results running live at HIPA many days after the training set:



No alarms

M3TBPS-Z14:IST:2
Warn
30.0
c

M3TBPS-Z12:IST:2
Warn
30.0
c

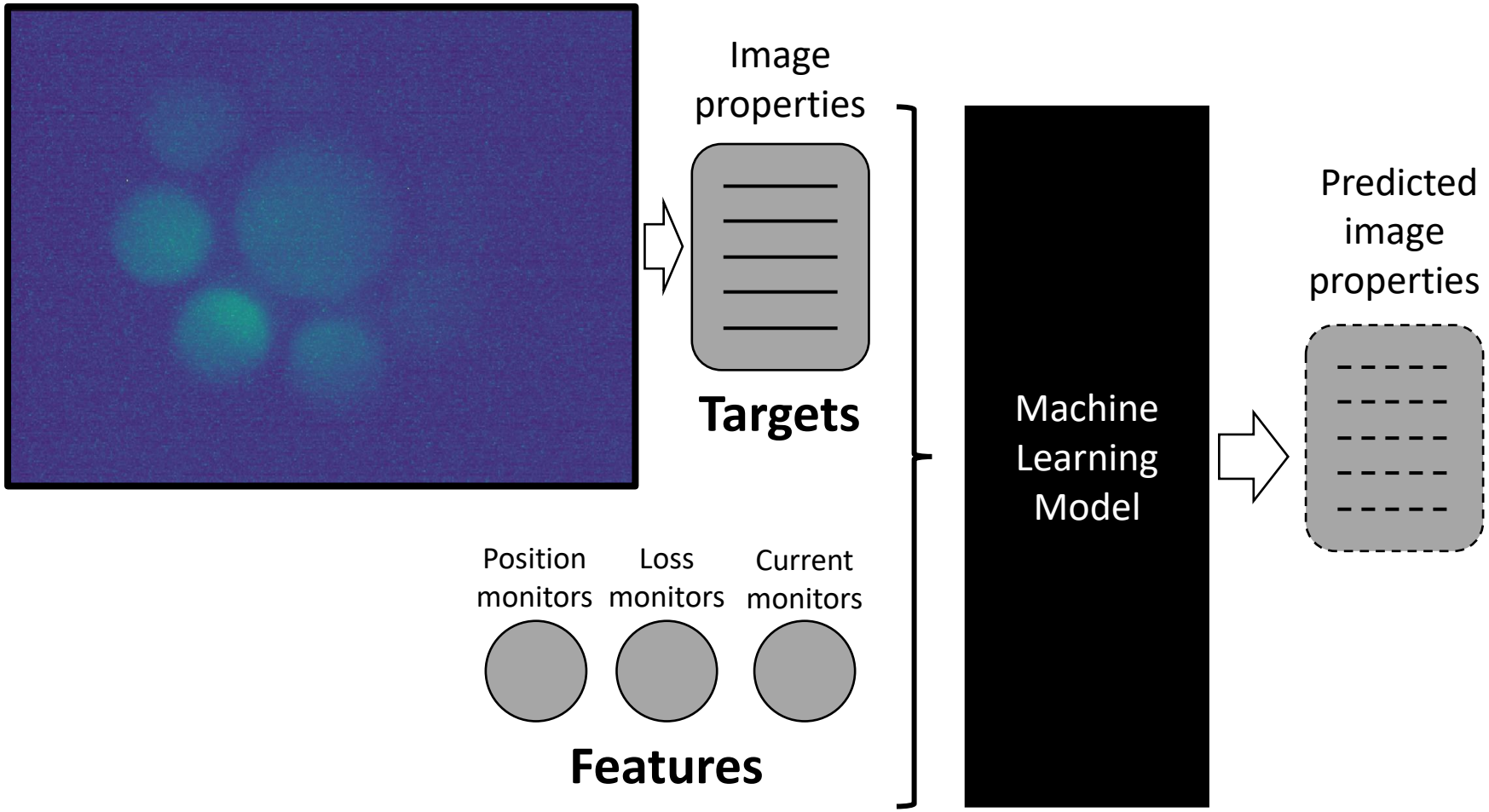
M3TBPS-O:IST:2
Warn
82.3
c

M3TBPS-L:IST:2
Warn
30.0
c

M3TBPS-R:IST:2
Warn
30.0
c

M3TBPS-U:IST:2
Warn

- Another proof of concept: the VIMOS system, part of the SINQ target protection.
- The beam shines on a grid and a camera captures the beam position and shape.



- Good models are available, but the camera image processing is being improved.

Virtual diagnostics conclusions

- Virtual diagnostic models have been developed and tested at PSI.
- The have the potential to approximate degrading, effectively replacing them.
- These models are relatively easy to build and train.
- As a proof a concept:
  - A model for a temperature sensor at the SINQ target has been developed and predicts the signal of the sensor within 1 degree.
  - The model uses just passive beam diagnostic sensors.
  - The predictions stays correct on data observed weeks after the training set, no long term drifts.

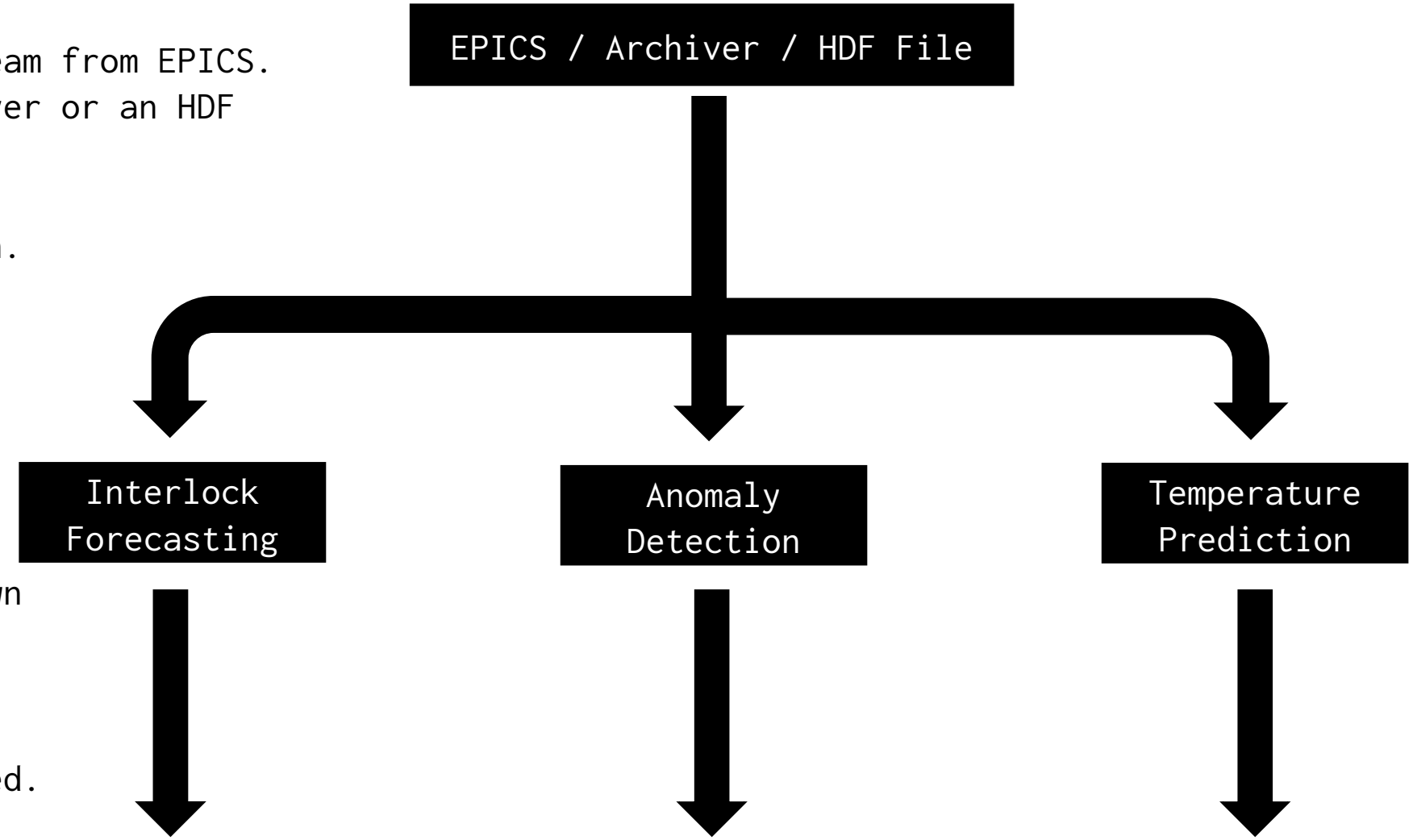
Models deployment

Data Collection:

- Alignment of the data stream from EPICS. Or loading from the Archiver or an HDF File.
- Missing data interpolation.
- Windowing as requested.

Model output computation:

- Each model produces its own results.
- Additional diagnostics information can be provided.



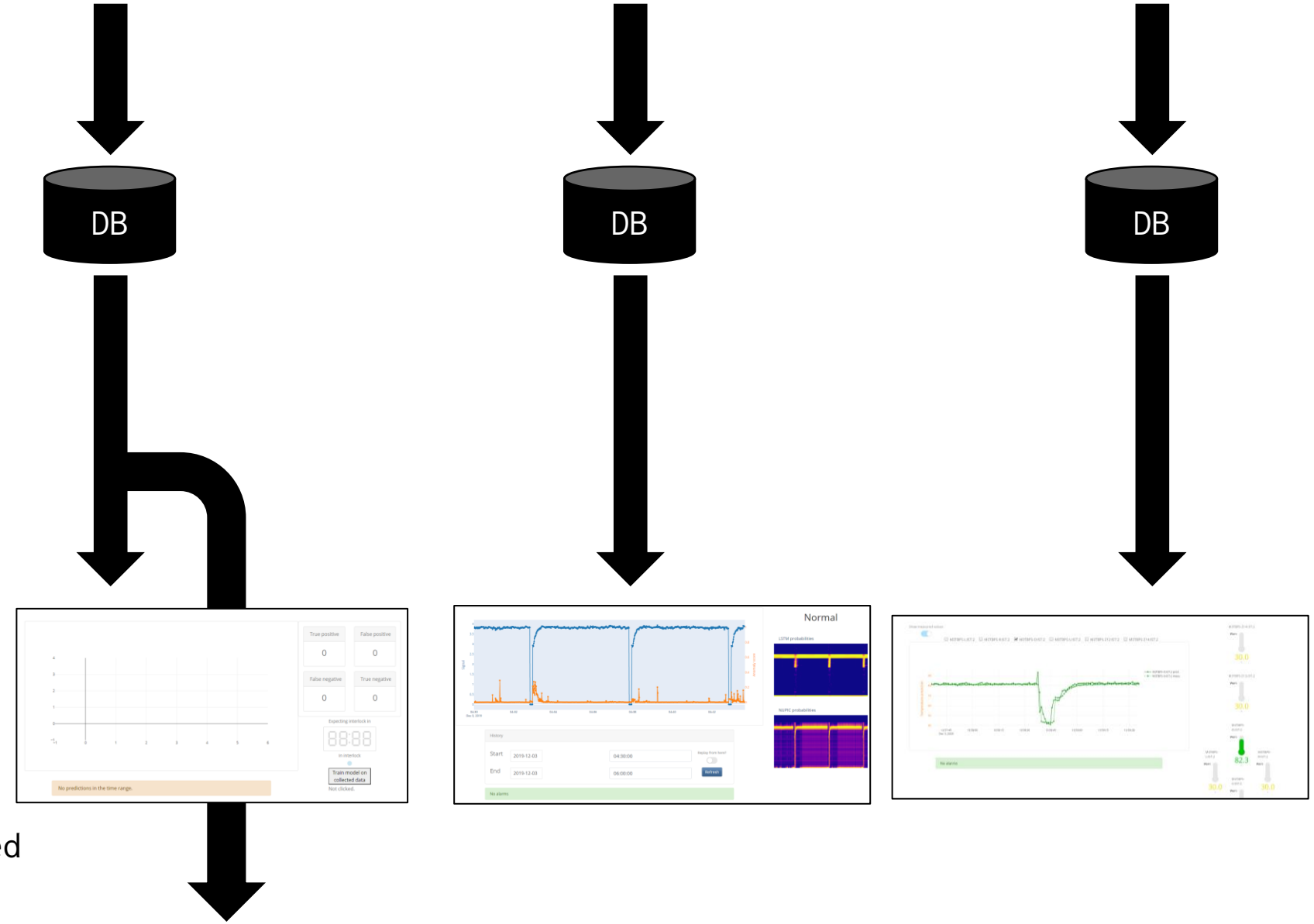
Models deployment

Data persistence:

- Input, prediction and diagnostic information are stored in a database.
- Each entry is keyed with the acquisition timestamp.
- All data is available for live display or later analysis.

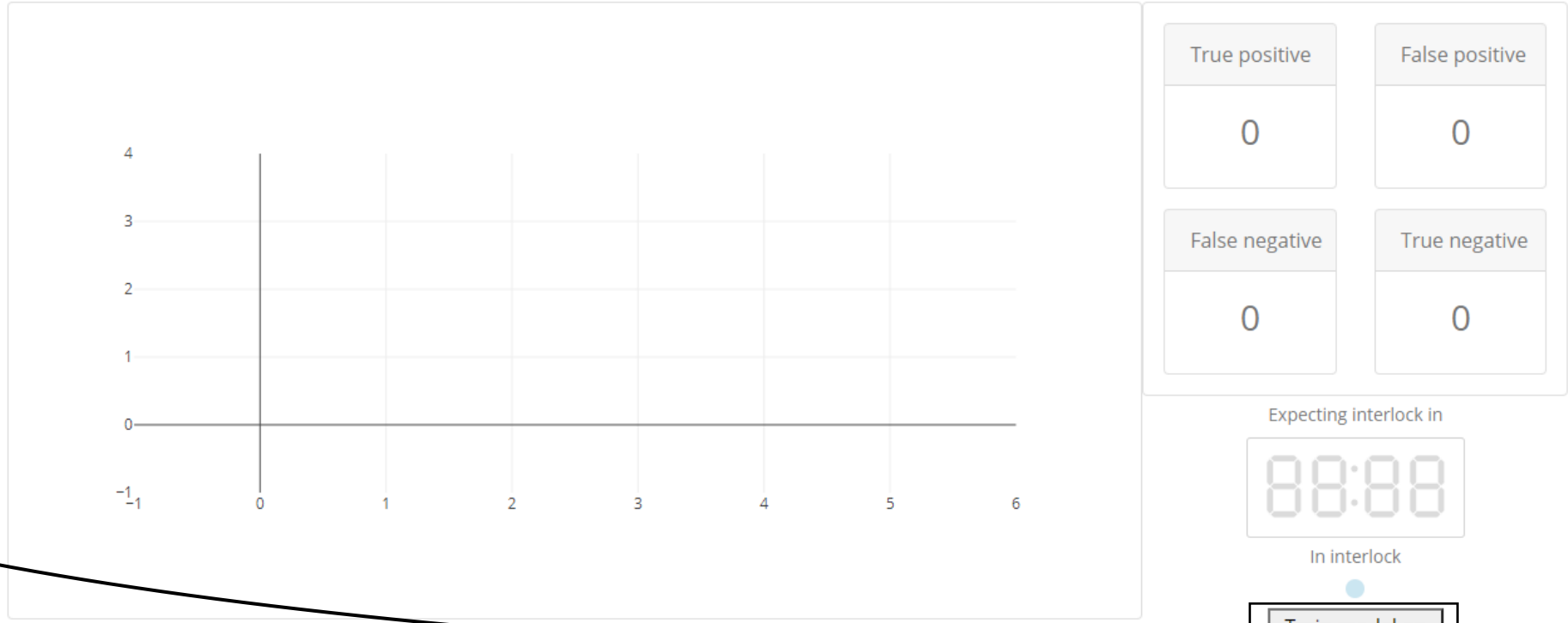
Display:

- Live or historic display for each of the models.
- Each display can be customized for the specific project.



Interlocks forecasting model retraining

- As the input data is recorded in the database, it can be used to train the model.
- The interactive training GUI is accessible from the display.



Train model on collected data

Not clicked.

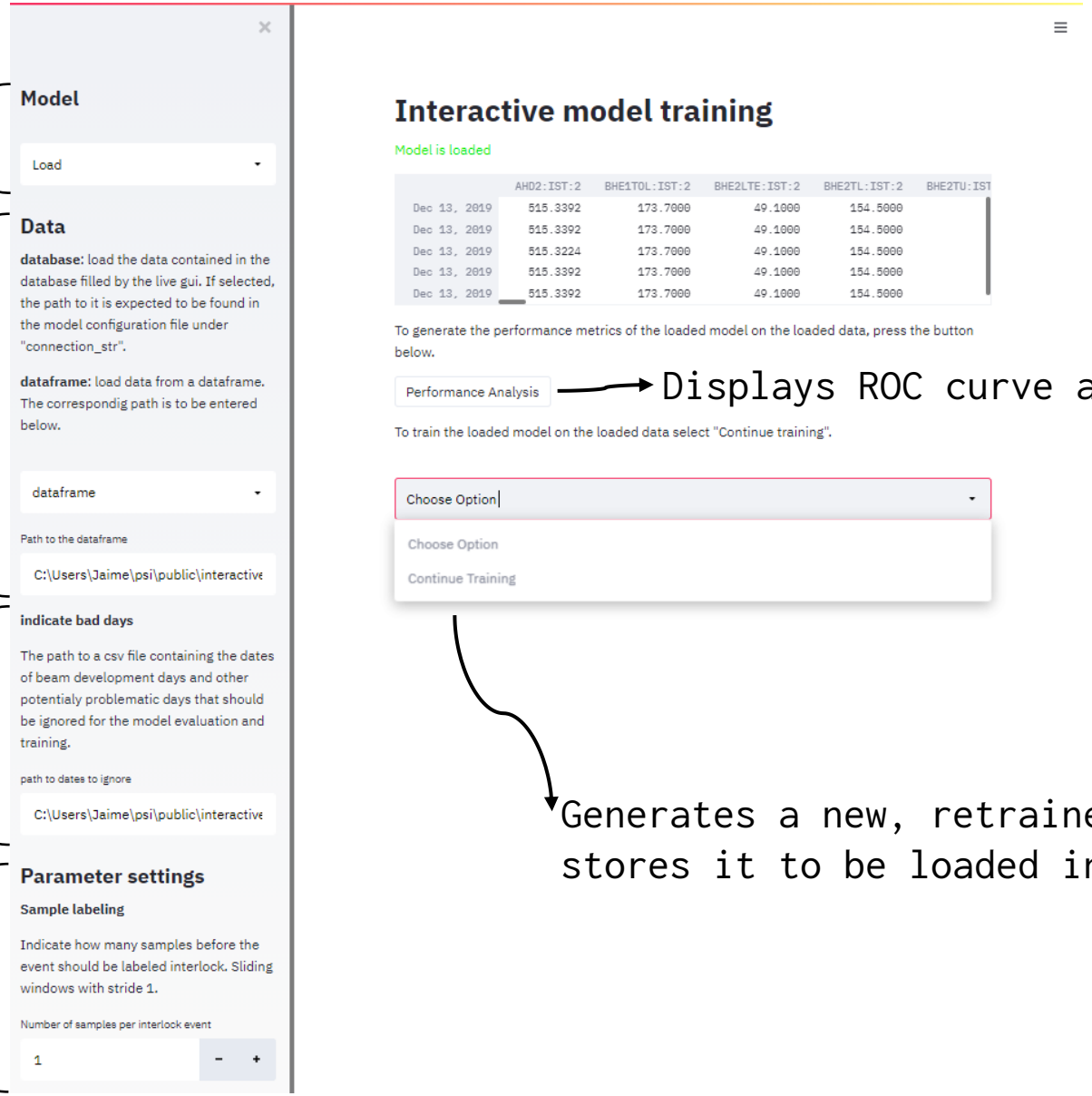
Interlocks forecasting model retraining

Takes the same configuration file as the live display.

Data can be loaded from the live display DB or from a HDF file.

A CSV file with days to ignore can be given (like beam development days)

Configure sample labelling

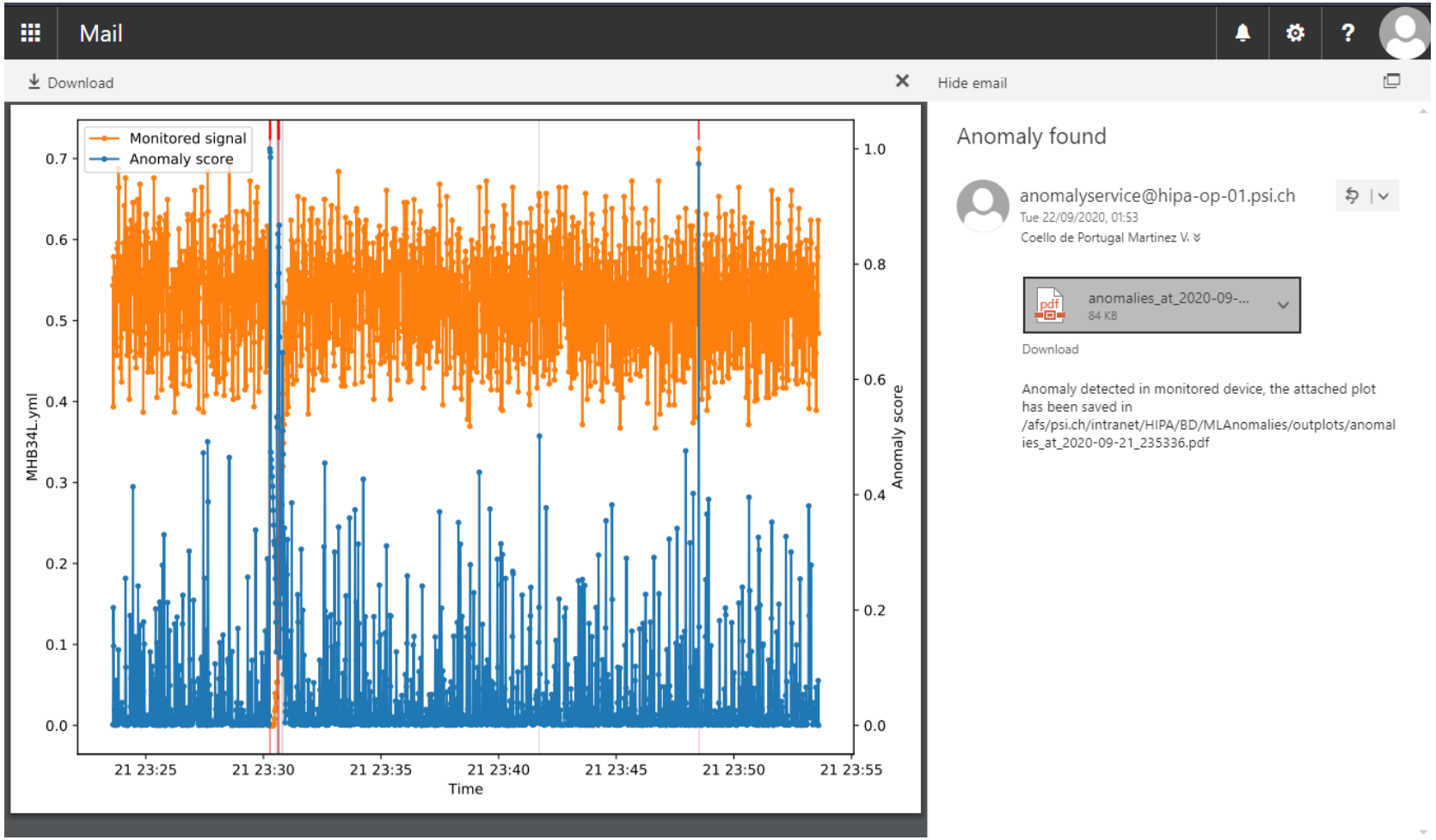


Displays ROC curve and confusion matrix

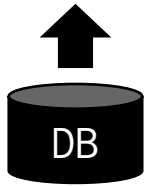
Generates a new, retrained model and stores it to be loaded in the live GUI

Email service for anomaly detection

- As data is stored and updated live in a database, other services can monitor it.
- This email service checks the database every few seconds and reports by email if an anomaly is detected.
- Sends a plot of a time window around the anomaly of every monitored device that detected the anomaly.



Anomaly detection



Models deployment conclusions

- All the machine learning models we developed are deployed using an unified system.
- The system can be relatively easily customized to deploy any live ML model on a machine controlled by EPICS.
- All input information, model results and diagnostic information are recorded in a database.
- The information from the database can be shown live or as historical data.
- In the case of the interlock forecasting model, the database information can be used to interactively retrain the model.
- The interactive training GUI allows to check the performance and redeploy retrained models on live recorded data.

General conclusions

During the progress of the PACMAN project, we:

- Developed a preliminary model able to detect incoming interlocks in advance, that could have saved about 7 minutes of beam time.
- Tested at HIPA and SwissFEL an automatic beam loss optimization algorithm, which is able to optimize the machine to human levels without producing many interlocks.
- Developed and tested a streaming anomaly detection framework, which could have alerted human experts about the failure of a diagnostic device many minutes in advance.
- Explored and developed proof-of-concept virtual diagnostic at HIPA. The temperature sensor model is able to replace the original device with an error of 1°C.
- Developed a general machine learning model exploitation framework, which eases the interaction between EPICS and the models and the presentation of the results to the users.

For more talks from other labs around the world: OWLE seminars https://sites.google.com/view/owle/home

Acknowledgements

Many thanks for the work presented here and the support during these studies to:

Jochem Snuverink
Melissa Zacharias
Johannes Kirschner
Andreas Adelmann
Fernando Perez Cruz
Nicole Hiller
Sichen Li
Anastasia Pentina
The SDSC