
CERN

Machine Learning Study of CLIC RF Breakdowns

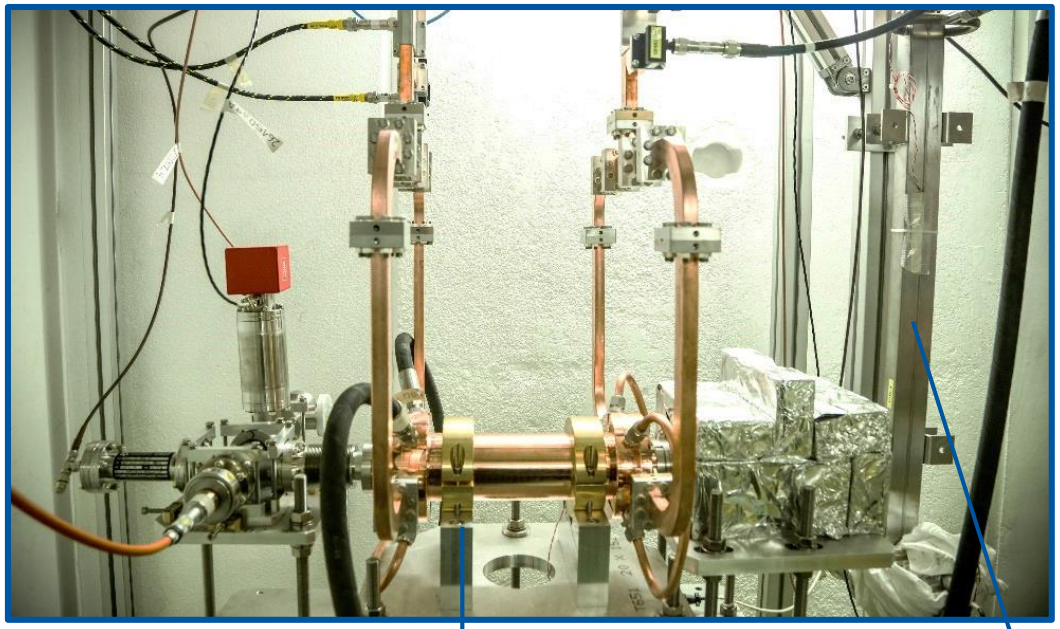
Christoph Obermair, TE-MPE-PE
Thomas Cartier-Michaud, TE-MPE-PE
Andrea Apollonio , TE-MPE-PE
Lee Millar, CLIC BE-RF
Franz Pernkopf, TU-Graz

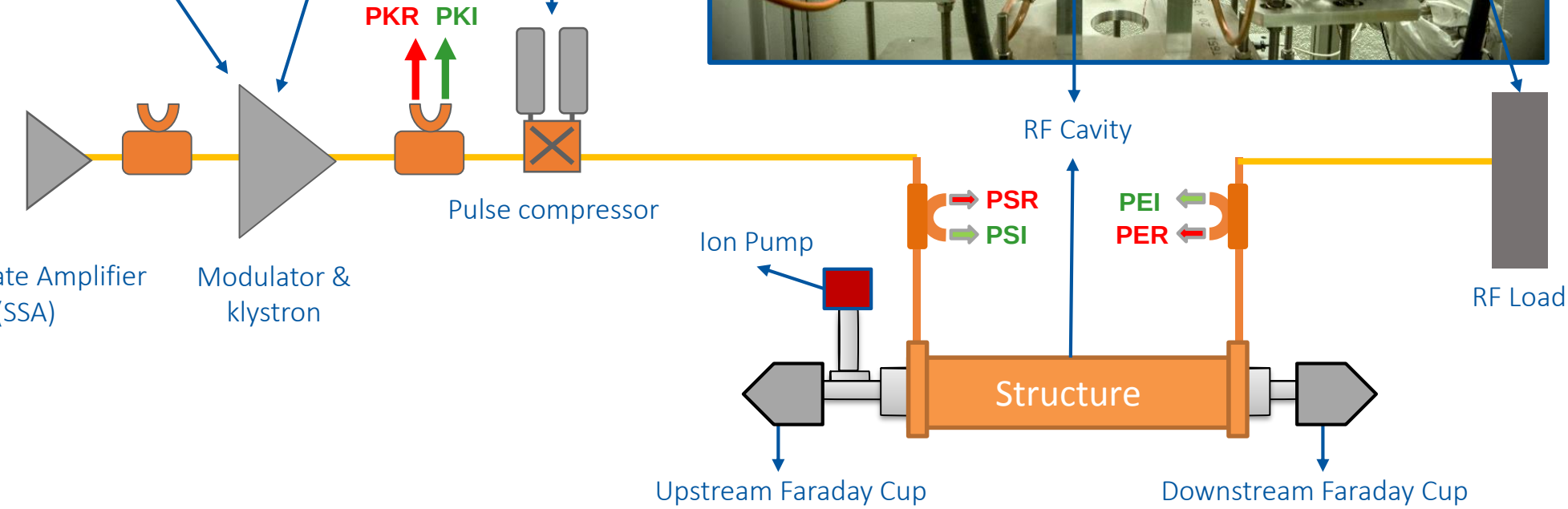
Presentation Outline

Presentation of preliminary analysis results:

1. Introduction
2. Transformation
3. Exploration
4. Modeling
5. Conclusion
6. Outlook

1. XBOX 2 Test Stand





Content provided by Lee Millar

1. Goals

Breakdowns:

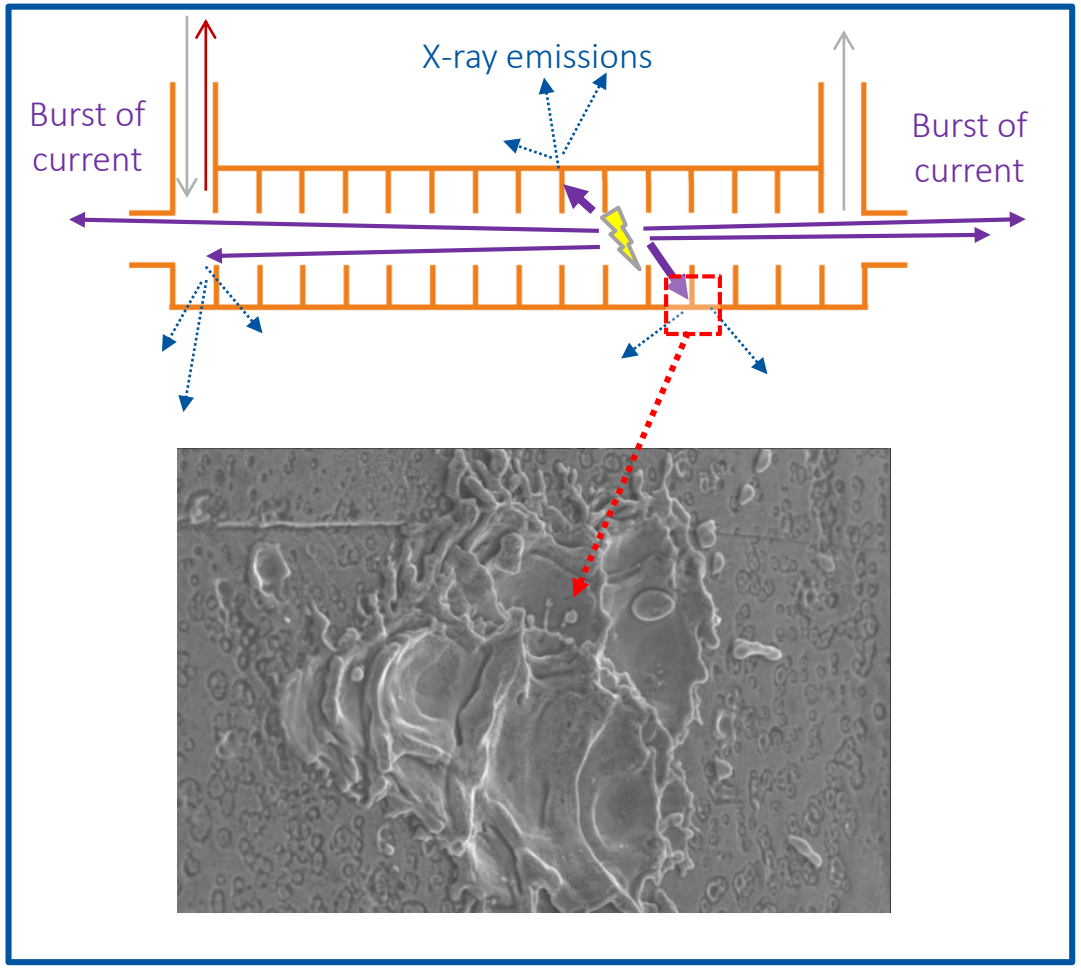
Small deformities cause field enhancement on surface which can lead to plasma formation (arc) and surface damage

- Labels “healthy” (=1) and “breakdown” (=0) are assigned by CLIC team with threshold on DC (Faraday Cup) and reflected signals

Goal (short-term): Investigate breakdown precursors in XBOX 2 test structure

Goal (ultimate): operational tool for BD reduction, in order to avoid their occurrence by suitable adjustment of the power in the cavity

RF Cavity breakdown



Content provided by Lee Millar

1. Data Overview

Data available:

90 GB of Data from 2018 (6months of data)

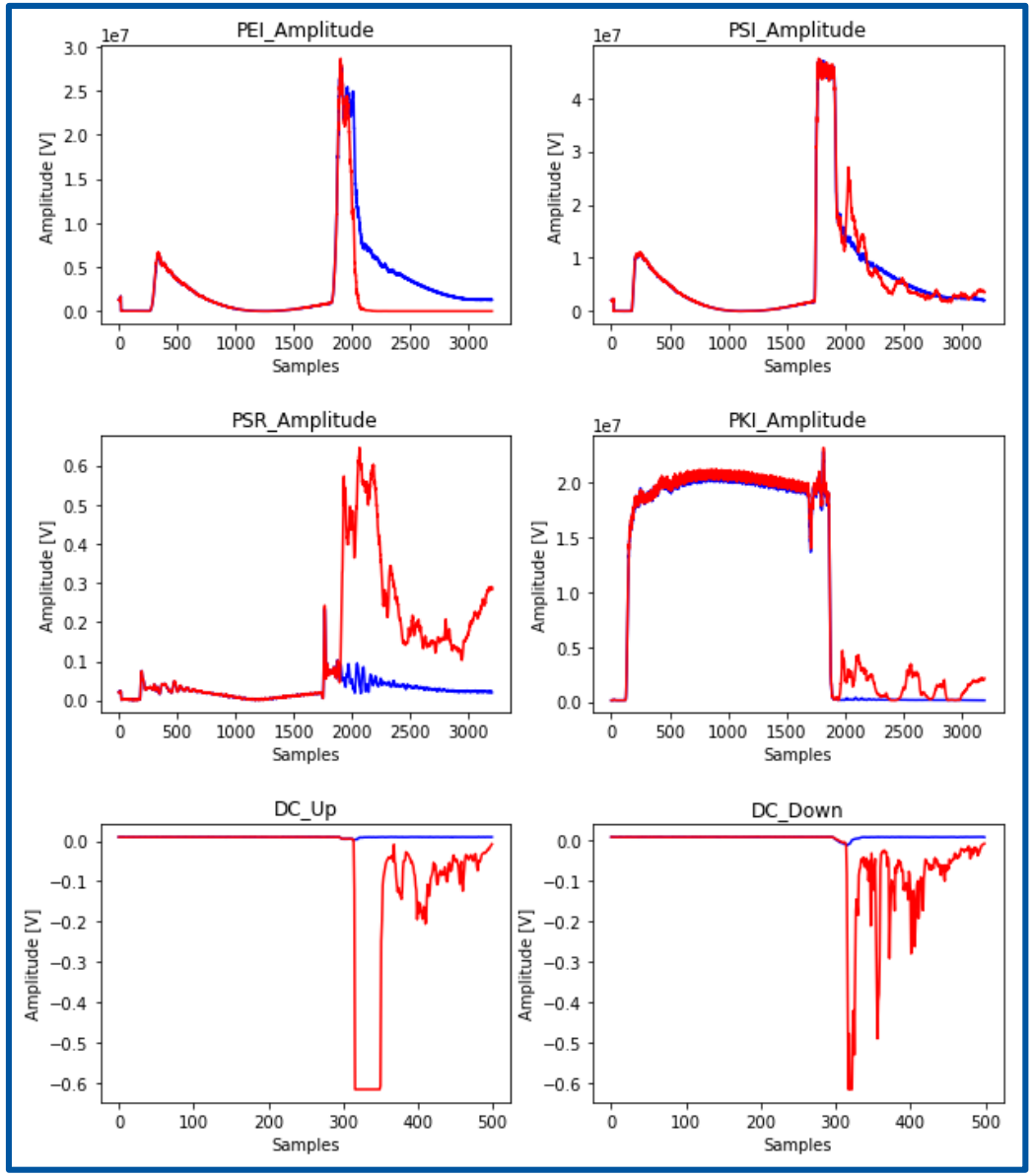
Event data:

- Data from test stand sensors: (healthy, breakdown)
  16 Channels with up to 3200 sample points (12 GHz sample rate)

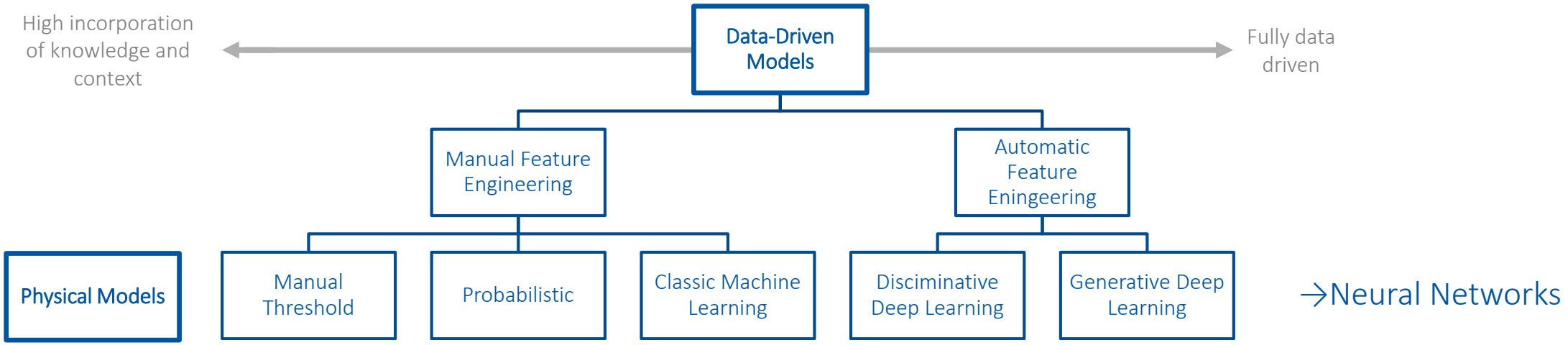
Trend data:

- Metadata about other machine parameters:
  i.e. Temperature, pressure (~1Hz sample rate)

	Bunker_WG_Temp	Klystron_Flange_Temp	Load_Temp	PC_Left_Cavity_Temp	PC_Right_Cavity_Temp
0	24.134285	24.116962	27.955711	30.285633	30.187632
1	24.082500	24.145723	27.876066	30.277182	30.208776
2	23.241980	24.400513	27.502401	30.253901	30.191576
3	18.783634	19.573864	27.900787	30.100540	30.056396
4	18.741947	19.662273	4.099991	30.123297	30.077822



1. Overview of Data-Driven Models



Approach 1:

Use physics knowledge to define model.

Requirement:

Deep understanding of given use case

Approach 2:

Use model to gain physics knowledge.

Requirement:

Data sets to measure performance e.g.: $D = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$

Content from M. Maciejewski and H. Fawaz et al., „Deep learning for time series classification“, 2019

1. Analysis Workflow

	Transformation →	Exploration	→	Modeling	→	Execution
Goal	Clean Data	Explore Data Properties		Train & validate Model		Implement Model into test facility?

Utilize **supervised** and **unsupervised** machine learning

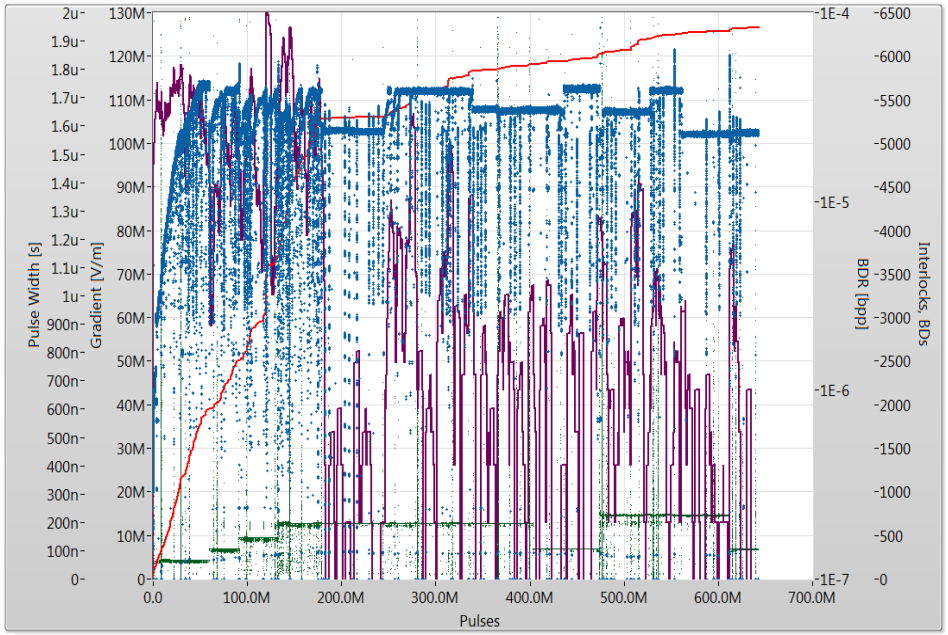
Content from M. Maciejewski

2. Transformation

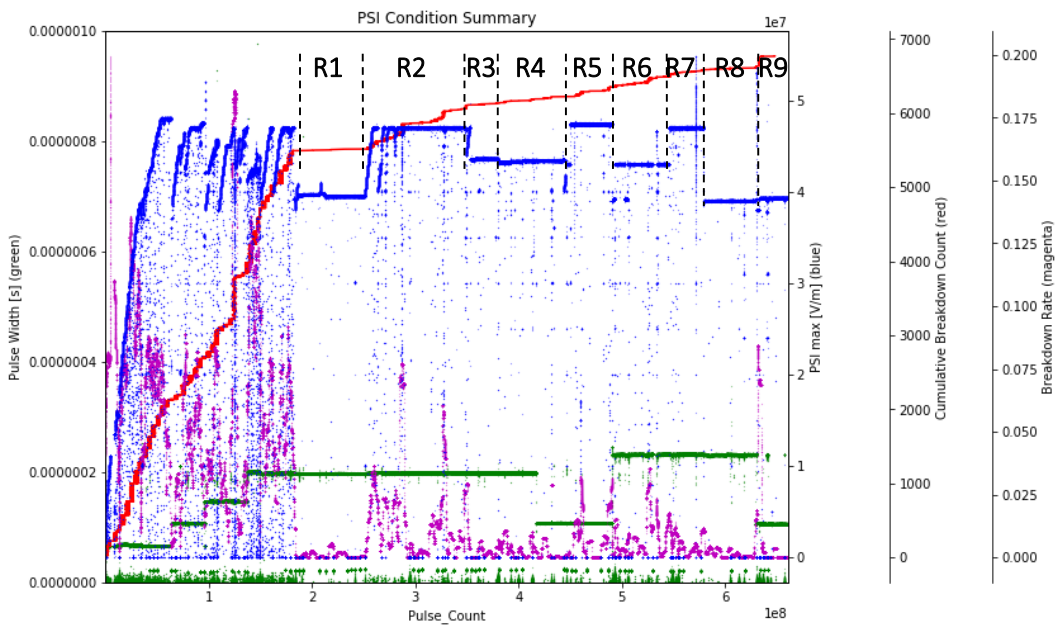
Goal: Clean data, fast queries, memory efficient storage, diverse usability

- Adding of extra features (e.g. run 1-9)
- Transformation of data into compressed pickle (24 GB, 1min to query data)

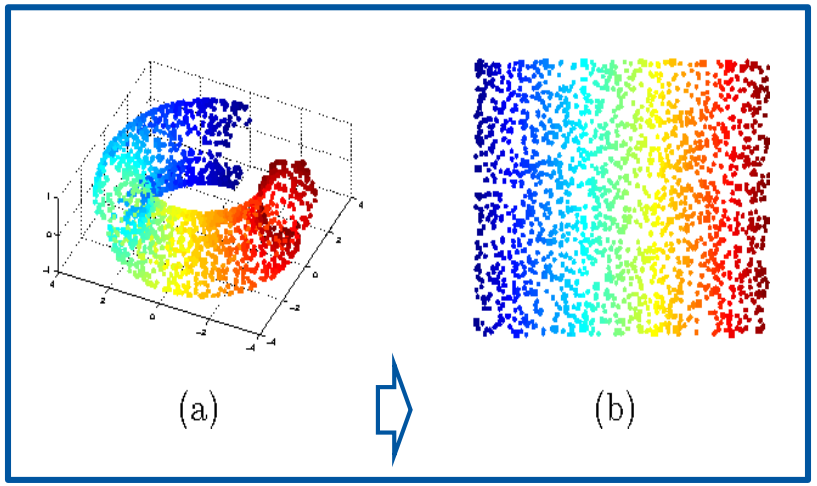
2018 Trend Data



Our replica



3. Exploration

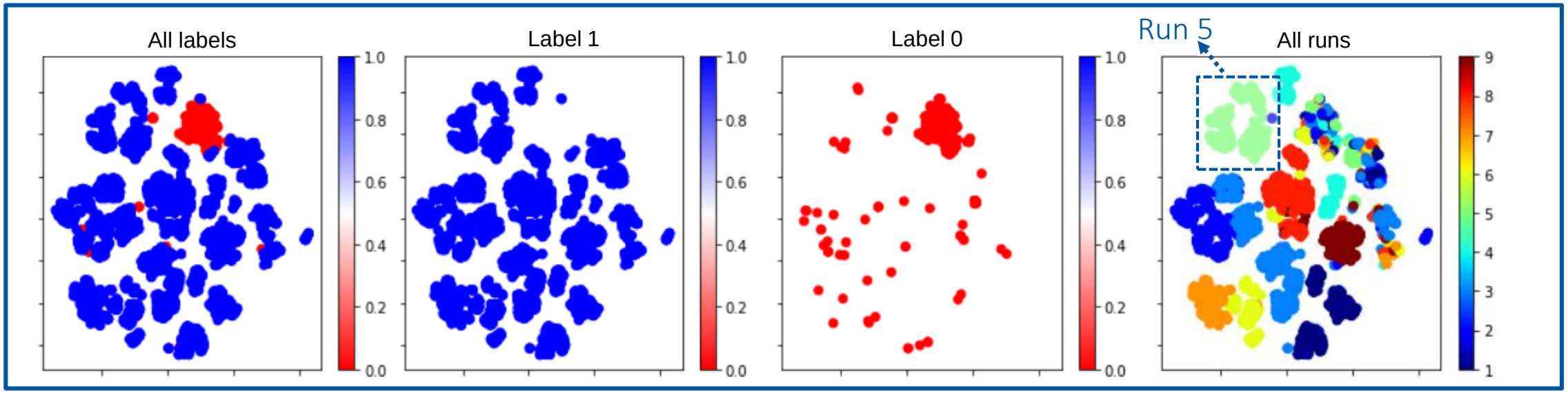


Goal: Easy and general method to explore multidimensional dataset.

- 2D t-SNE (t-Distributed Stochastic Neighbor Embedding)
- Find projection to 2D space by retaining most important information
- Axes are arbitrary

1 … healthy
0 … faulty

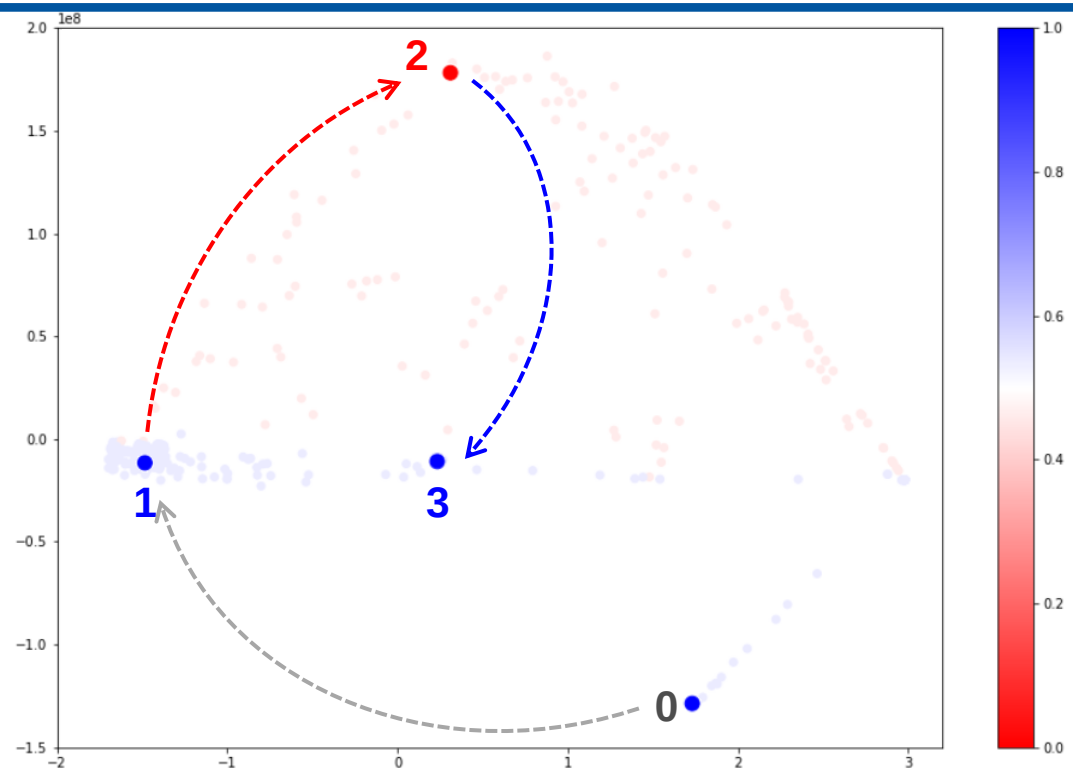
2D t-SNE of XBOX 2 Trend Data (during stable operation)



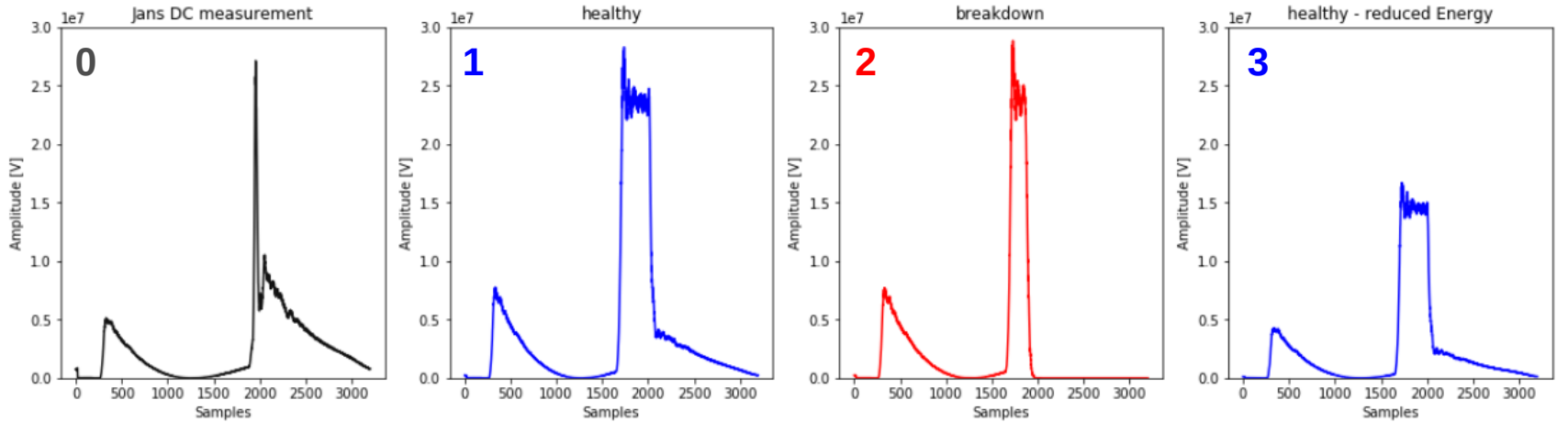
3. Dimension Reduction with 2D-tSNE

Data:

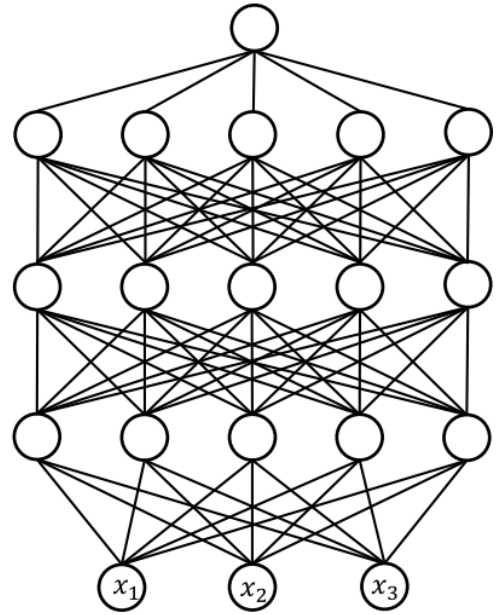
- Event data signal (timeseries):
  PEI Amplitude
- Date: 20-26.01.2018
- Axes represent arbitrary



	label_CLIC	timestamp
0	1	2018-01-26 14:52:30.443347
1	1	2018-01-26 14:59:50.549269
2	0	2018-01-26 15:26:23.099318
3	1	2018-01-26 15:26:28.481433

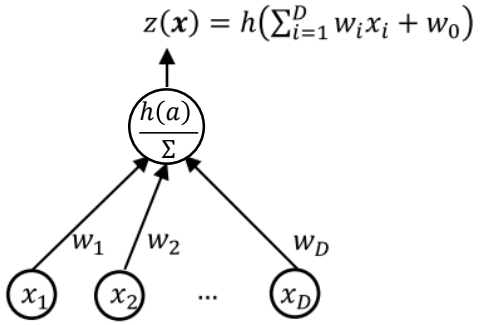


4. Introduction to Neural Networks

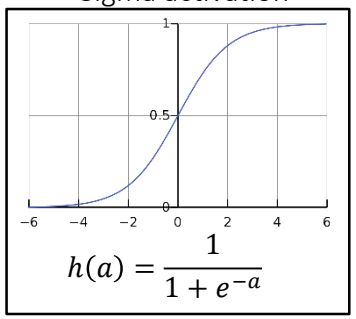


Perceptron:

Main component of neural network



Sigma activation

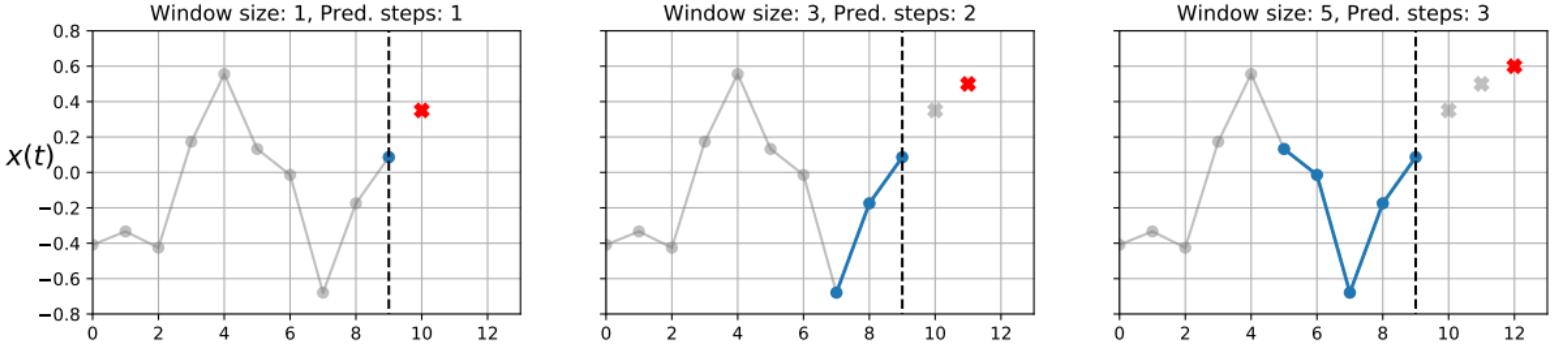


Neural networks:

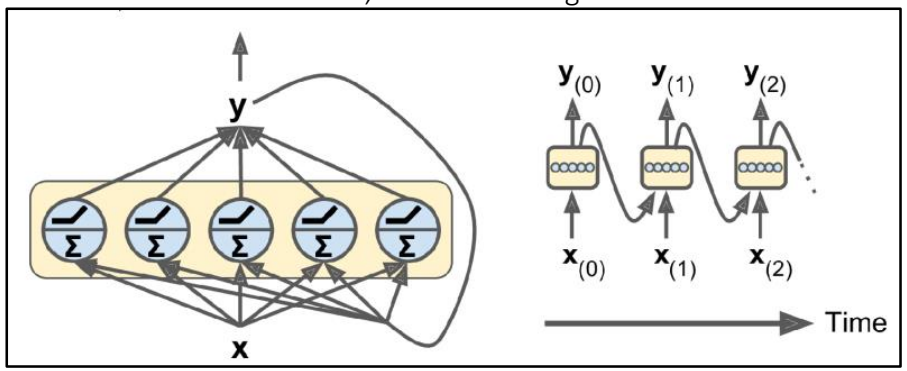
Many layers of perceptrons makes it possible to find nonlinear properties (deep learning)

Recurrent neural networks:

Neural Network, designed to handle sequential data



Recurrent Neurons, unrolled through time:

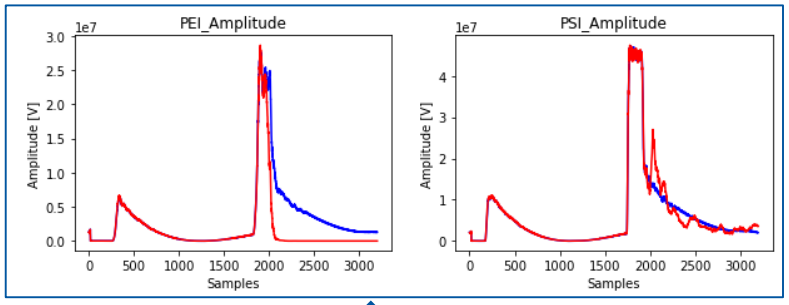


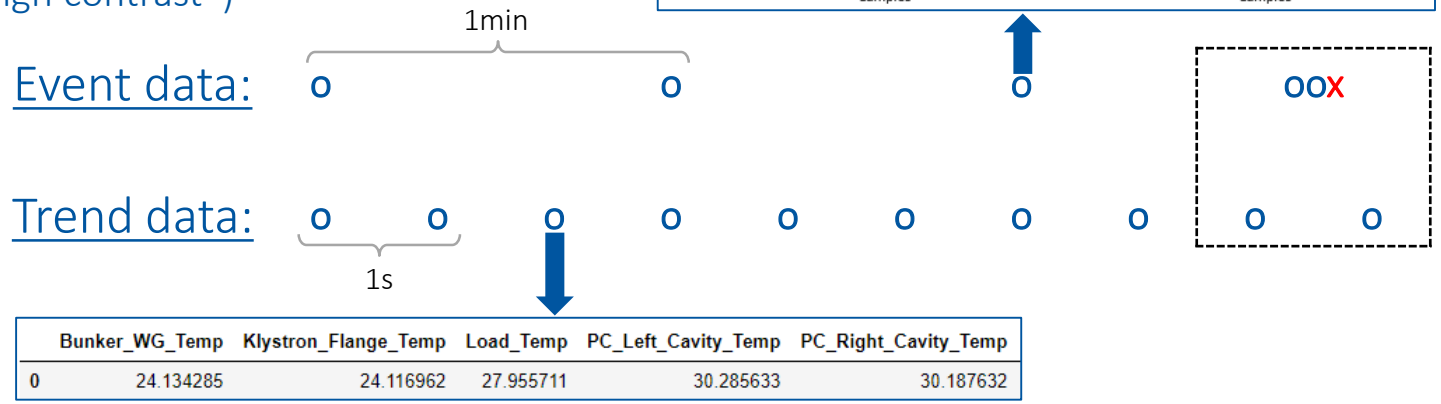
window size = 5

4. Merging of Event Data with Trend Data

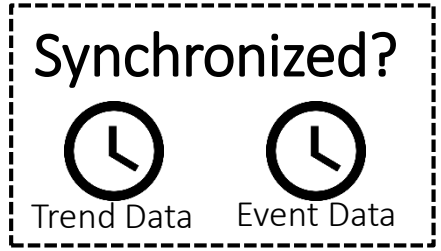
Input data:

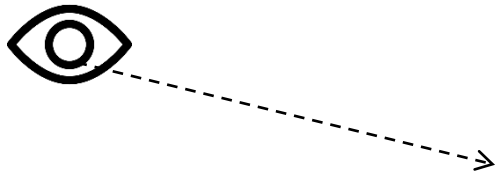
- Different sampling time of event data (~20ms/1min) and trend data (~1s)
- Clean the data such that we do not give the answer (“high contrast”)





	Bunker_WG_Temp	Klystron_Flange_Temp	Load_Temp	PC_Left_Cavity_Temp	PC_Right_Cavity_Temp
0	24.134285	24.116962	27.955711	30.285633	30.187632





4. Modeling Properties

Class imbalance (within runs):

"healthy": 124448, "faulty": 479 (250 are follow up bd)

Balanced accuracy:

Accuracy which takes class imbalance into consideration.

Labels:

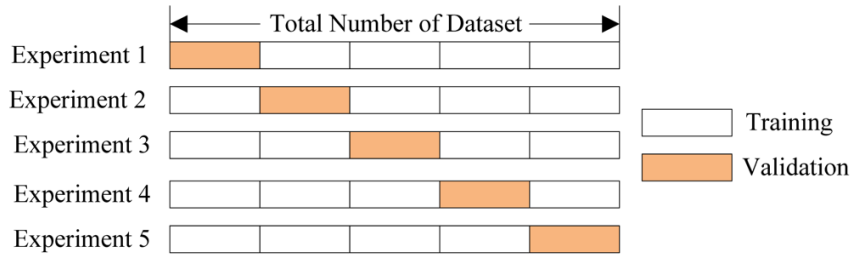
Own labels were created:

$$label = \begin{cases} 0 & if\ \min(dc\ up)\ or\ \min(dc\ down) < -0.05 \\ 1 & otherwise \end{cases}$$

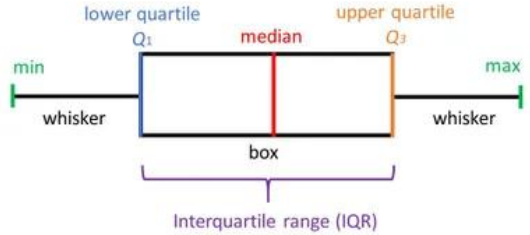
We ignored reflected signals, because DC signal is the most reliable filter for structure breakdown

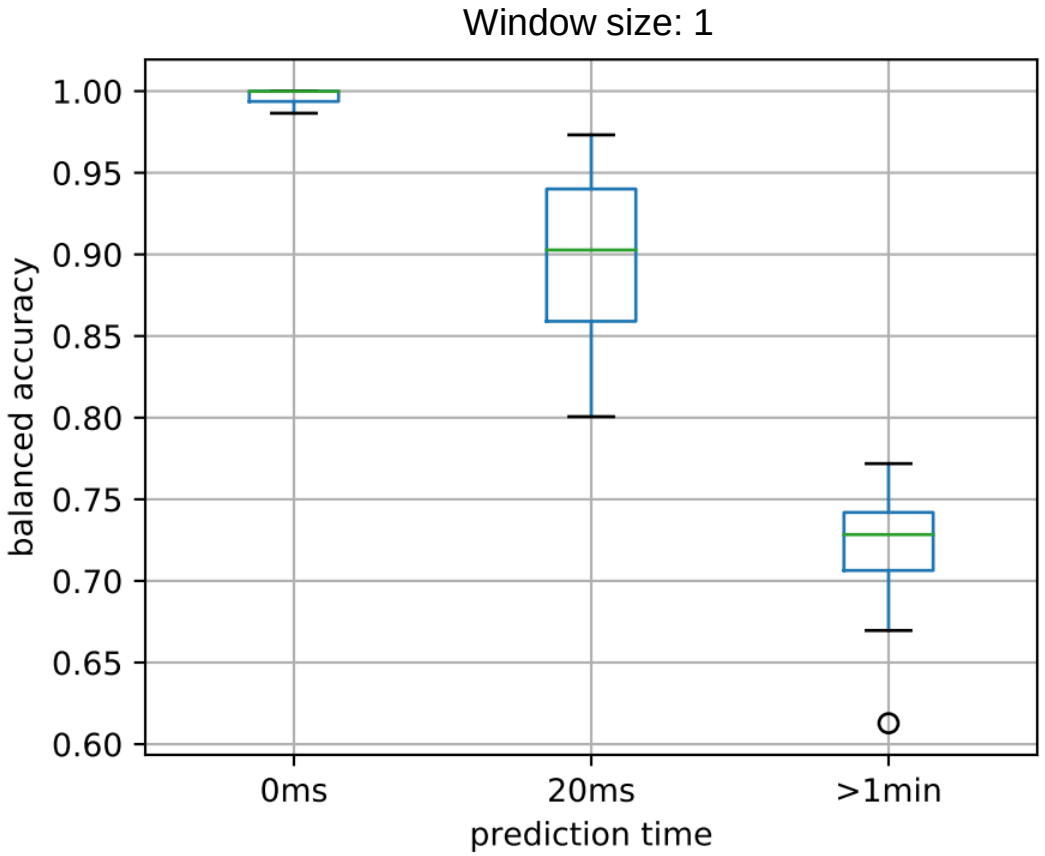
Leave one out crossvalidation:

Iterate through dataset using a different run for validation each time
e.g. train model on run 1-8, validate on run 9 - transfer information of run 1-8 to run 9
then train model on run 1-7,9, validate on run 8…



4. Results





Conclusion:

0ms: Almost 100% accuracy on classification

- 0ms: Network only had to restore thresholds on DC
- Important features: DC

20ms: Still good prediction accuracy

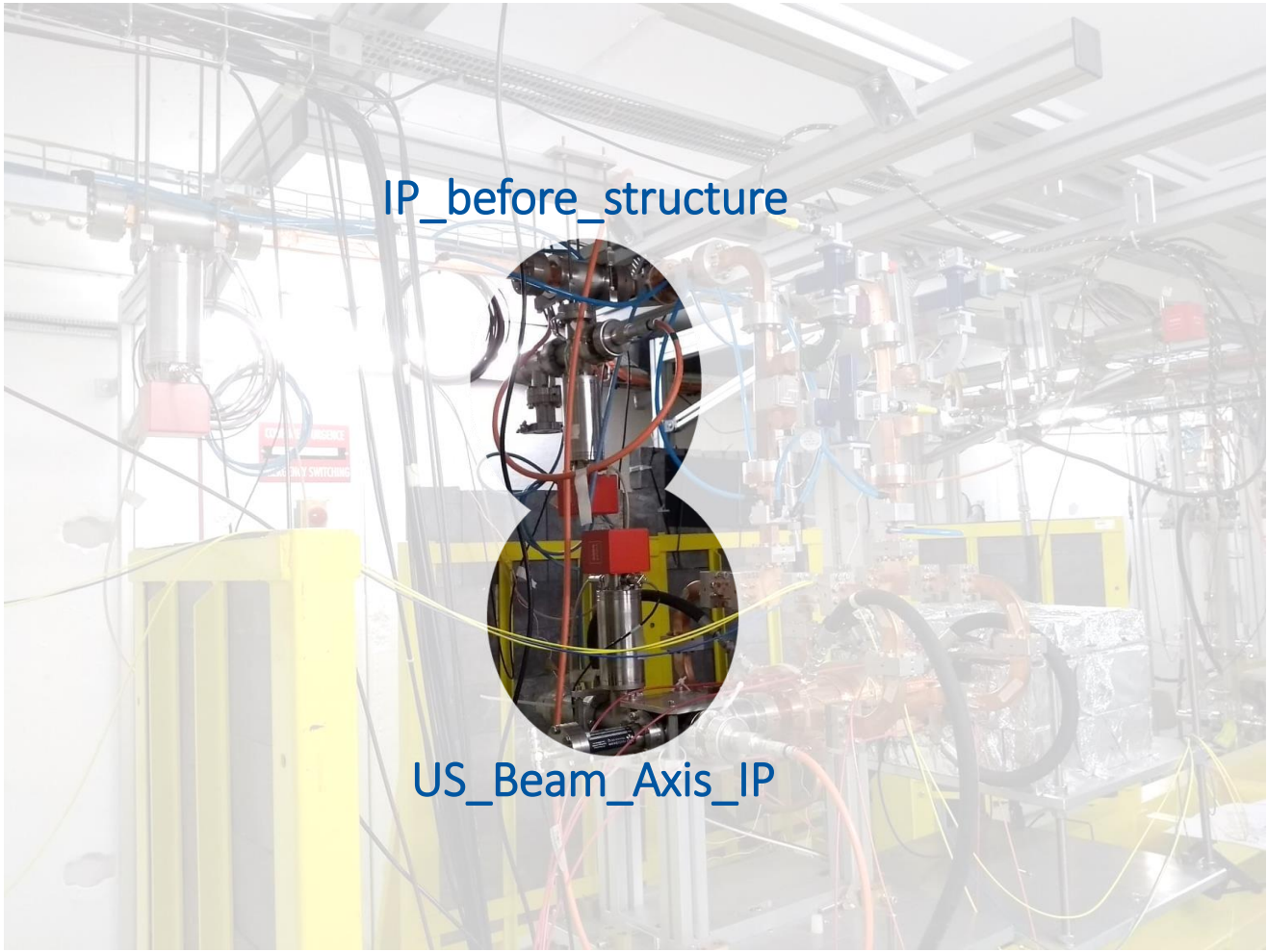
- Important features:
  - US_Beam_Axis_IP
  - IP_before_structure

>1min: Prediction of follow up breakdowns possible

Best results with window size 1:

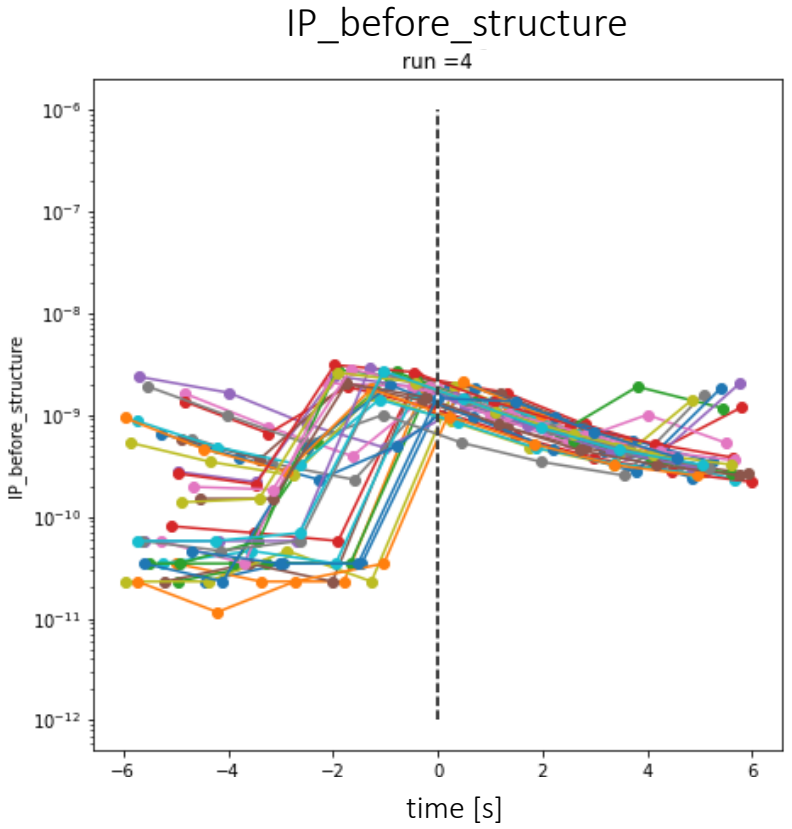
- Breakdowns depend strongest on last value

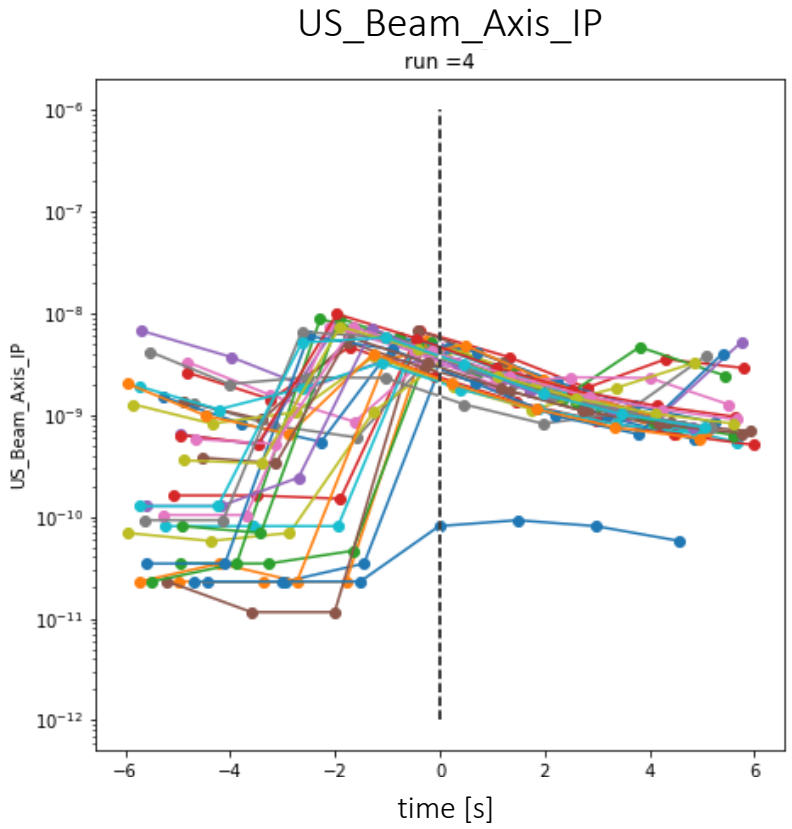
4. Important Features



4. Important Features

20ms: Does the good result come from data misalignment or is it a real precursor?





5. Conclusion

- Framework for Transformation, Exploration, Modeling available on gitlab
- Transformation:
  - Significant data cleaning performed
- Exploration:
  - Unsupervised: Clear clusters visible when using dimension reduction with 2D t-SNE
- Modeling:
  - Supervised: Good performance on prediction, (if sampling is correct).
- More refinement of the models needed with the ultimate goal of having an operational tool for precursor identification
- Support / comments from RF experts essential: Does it make sense that those features have such a high impact?

6. Outlook

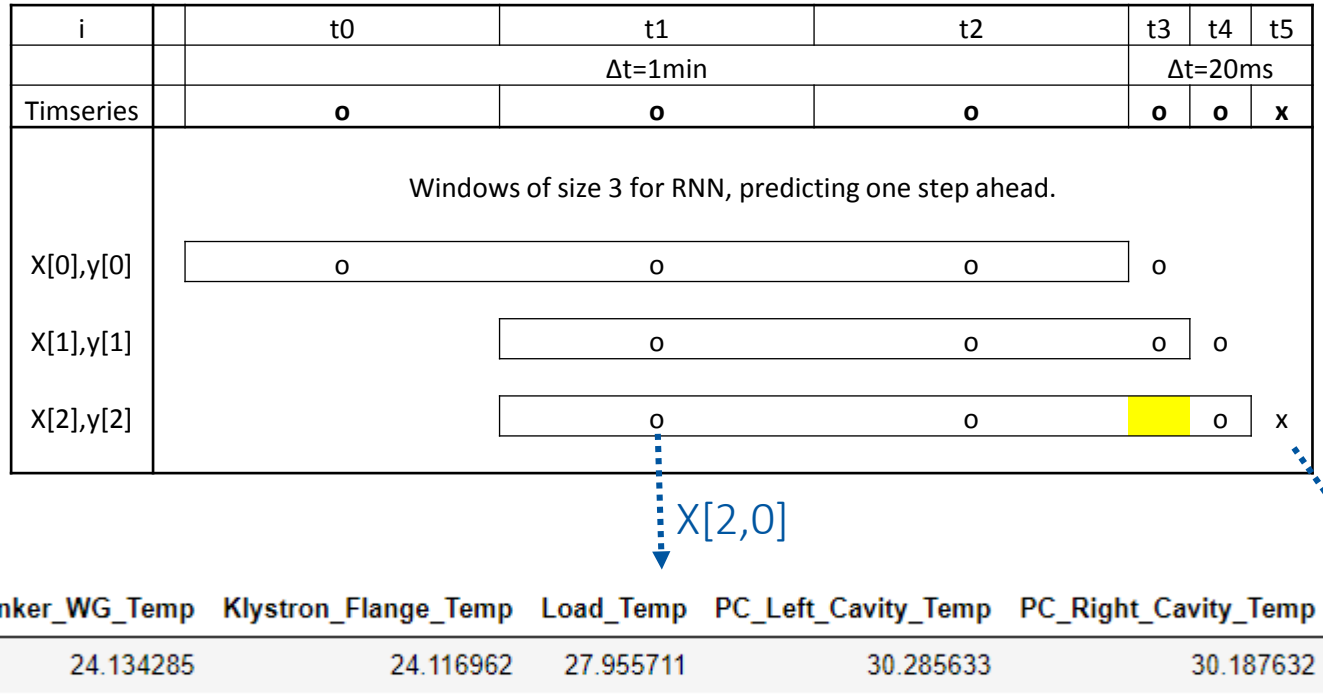
- Use raw time series for prediction, automatic feature calculation
  - Higher accuracy expected, but CPU / GPU expensive
  - Has there been any prior similar analysis?
  - Are there additional representative features which we could calculate out of the time series, to improve accuracy?
- Is it possible to log trend data with same frequency of BD event data for upcoming tests?
- Find physics meaning of good classification result
  - Support / comments from RF experts essential
- Use additional data from other test runs
- Long term goal: Implement model in test stand (if model proves to be robust enough)


CERN

High Contrast Function

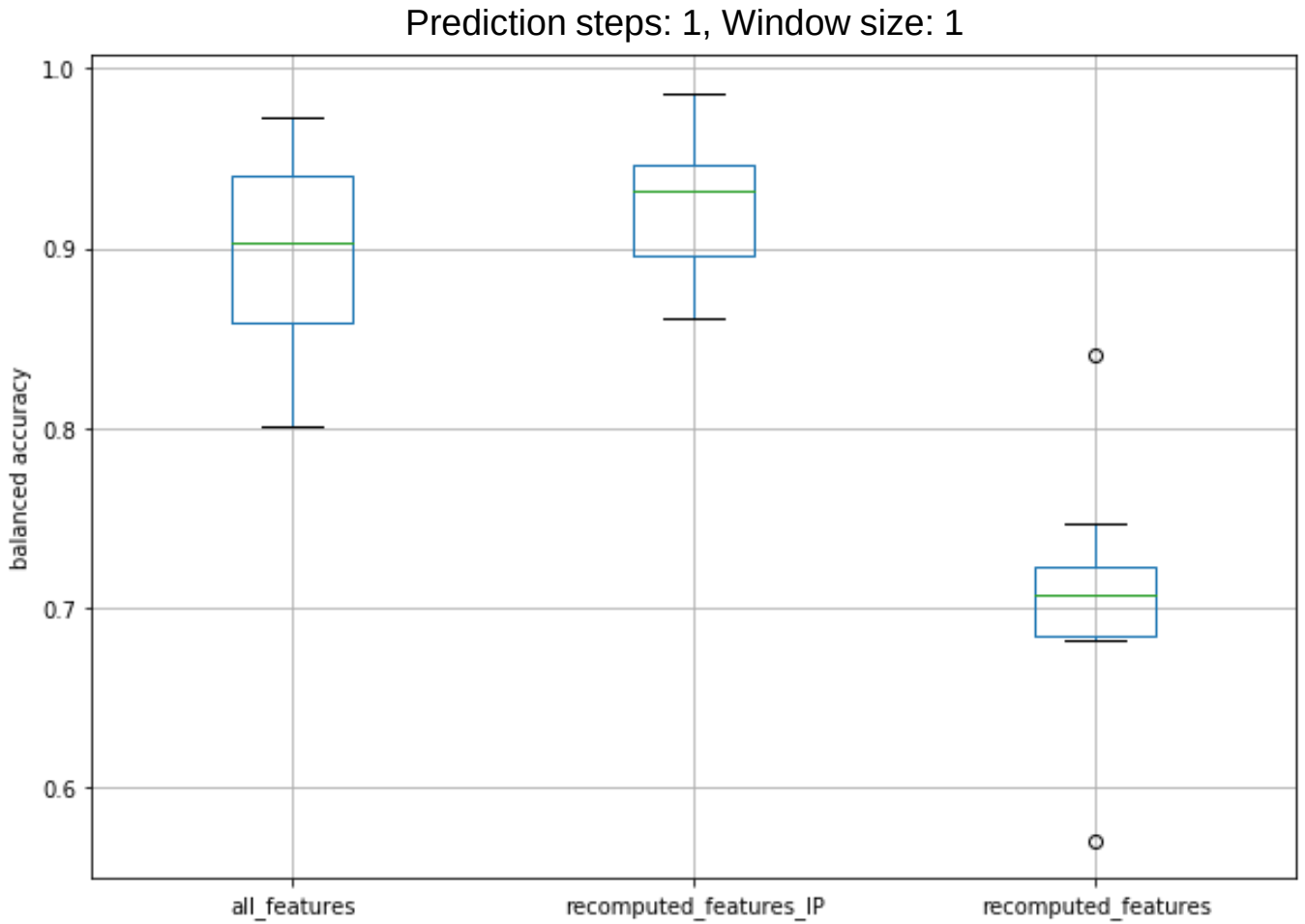
Clean the data such that we do not give the answer:

- Δt is not stored in Matrix, but RNN can detect the different Gradient
- “High Contrast” example for window size = 3, predicting 1 step into the future.

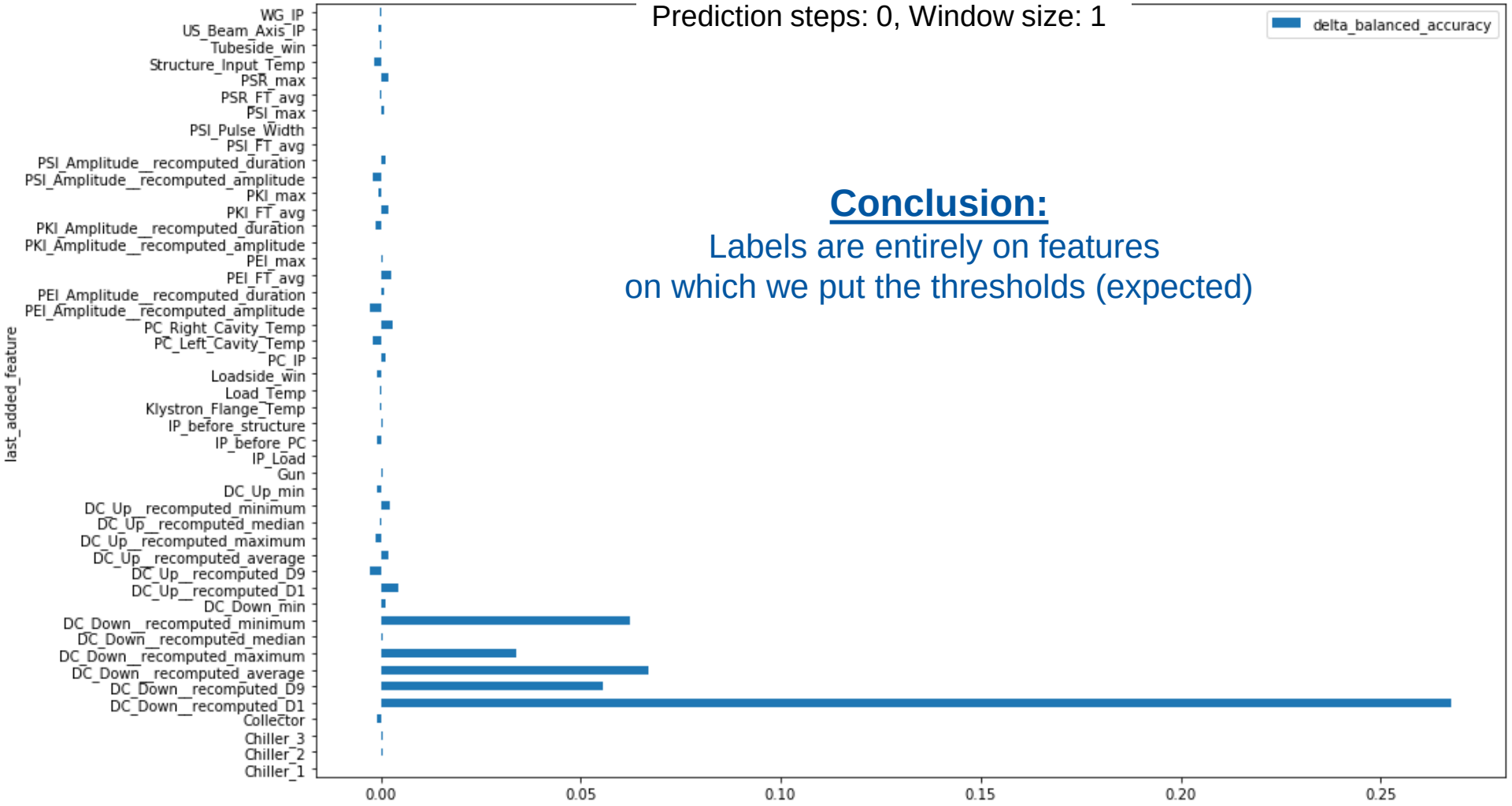


Bunker_WG_Temp	Klystron_Flange_Temp	Load_Temp	PC_Left_Cavity_Temp	PC_Right_Cavity_Temp
24.134285	24.116962	27.955711	30.285633	30.187632

Results


Prediction steps: 1, Window size: 1
balanced accuracy
1.0
0.9
0.8
0.7
0.6
all_features
recomputed_features_IP
recomputed_features

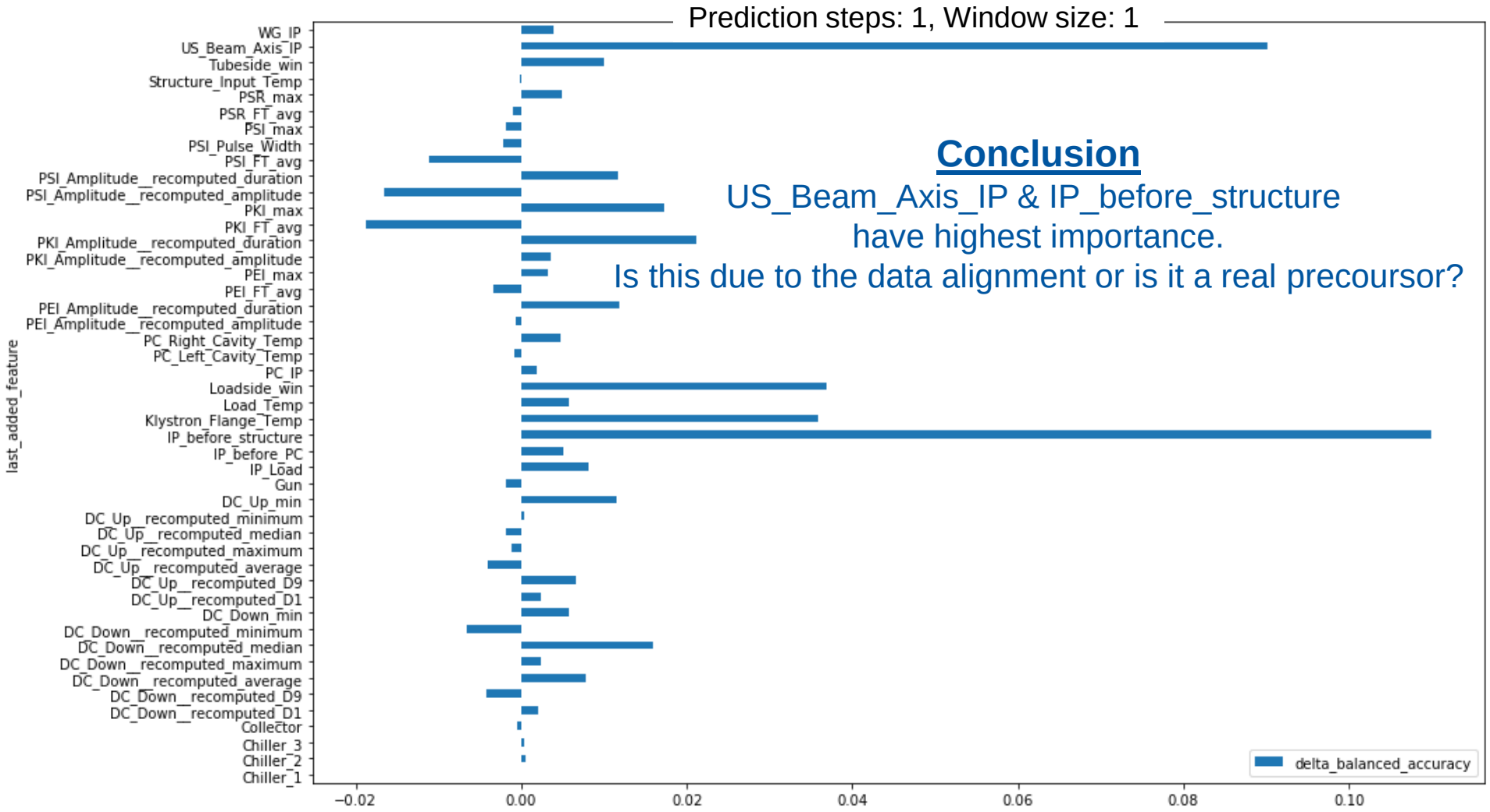
Feature Importance



Conclusion:

Labels are entirely on features
on which we put the thresholds (expected)

Feature Importance



Conclusion

US_Beam_Axis_IP & IP_before_structure
have highest importance.
Is this due to the data alignment or is it a real precoursor?