
CERN

Anomaly detection for Beam Transfer installations

P. Van Trappen, M. J. Barnes (CERN TE/ABT)
T. Dewitte, E. Van Wolputte et al. (Leuven University)

2nd IFCA Workshop on Machine Learning Applications for Particle Accelerators, Feb. 26 – March 1, 2019, PSI Paul Scherrer Institut
https://indico.psi.ch/event/6698/

Content

1. Project motivation
2. Software tools (scikit-learn, Spark)
3. Anomaly Detection Engine Pipeline (ADEP)
   - Pre-processing, anomaly detection, post-processing, evaluation
4. Outlook

1. Project motivation

- Collaboration between CERN and Leuven University. We provide the data, problem case and machine expertise; they provide ML expertise and master student supervision.
- Problem statement: vast amount of sensor data (pressure, temperature, voltage, current, calculated metrics, beam parameters), resulting in:
  - Many measurements without thresholds for alarms generation
  - Time-consuming manual analysis after fault occurrence, to understand anomalous behaviour
  - Equipment interlocks (i.e. machine downtime) that could have been prevented!
- Use of ML as a solution by applying unsupervised learning for anomaly detection, based on historical data

2. Software tools

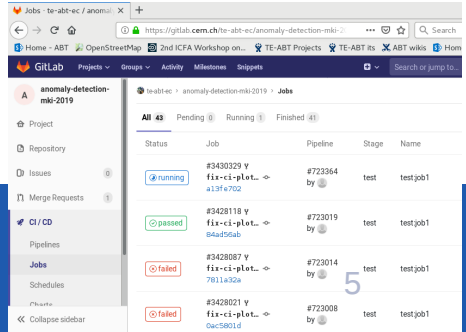
- Normal Software flow:

  CALS logging database (Oracle) → Python extraction script → MongoDB → Gitlab LFS →
  Pandas → Scikit-learn → scripts, Jupyter Notebooks and Plotly Dash

- In addition: porting the pipeline to use *Apache Spark Dataframes* and *MLlib*, because of CERN's Spark cluster with direct access to NXCALS logged data
  - Computational gain *vs.* porting effort and smaller user-community

Data, model and scripts:

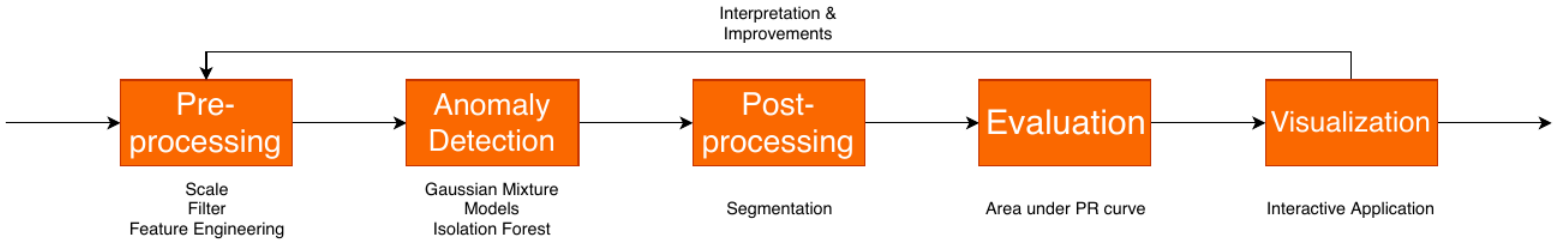
- All publicly at https://gitlab.cern.ch/te-abt-ec/anomaly-detection-mki-2019/
- Implemented Continuous Integration (CI) for reliable code



3. Anomaly Detection Engine Pipeline (ADEP)

Pipeline and grid-search

- Modular and object-oriented, to allow easy addition of e.g. models
- Grid search allows automated model hyper-parameter and evaluation tuning



3. Anomaly Detection Engine Pipeline (ADEP)

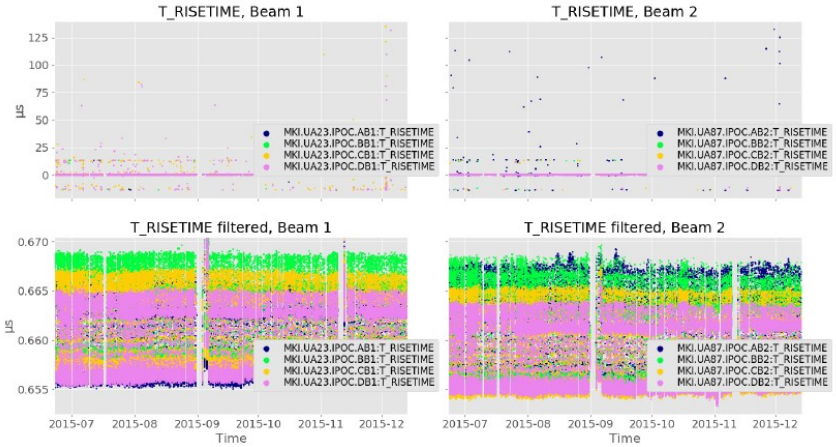
Pre-processing

- Dedicated to the x2 4 LHC injection kicker magnet pulse generators (MKI)
- Focussed on 18-months timespan (03/2016 – 09/2017), with some well-understood anomalies
- Wide variety of data, fixed-frequency and on-event sampled, some filtered at the level of the database → 120 variables total
  - Continuous data, IPOC data, controller data, beam data, e-logbook data
- Applied techniques: removal of bad measurements (hard-coded thresholds), data interpolation, resampling
- Feature selection and generation, including sliding window for temporal information and evaluation of TSFRESH* generated feature set

* python package ‘Time Series Feature extraction based on scalable hypothesis tests’

3. Anomaly Detection Engine Pipeline (ADEP)

Pre-processing



Effect of bad measurements filtering (magnet current > 1kA)

EVENTDATE	25/07/2016 00:19:48
PATH	LHC.MKI2
COMMENT	CCC calls for a faulty MKI2 generator during LHC fill. I checked and there was a flashover in magnet D, vacuum recov- ered rather slowly. IPOC also shows an increased current for IM-D. Flashover **not** detected by fast interlocks. . . .
TAG	anomaly

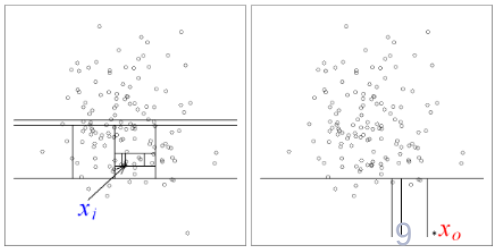
E-logbook example entry, manually tagged

Possible labels: anomaly, fault, info, intervention, research

3. Anomaly Detection Engine Pipeline (ADEP)

Anomaly detection

- Gaussian Mixture Models (GMM)
  - Fit all data to a mixture of finite Gaussian distributions with unknown parameters A datapoint with a low probability of belonging to these distributions, is anomalous
  - Scales well but interpretability is limited. Number of components hard to determine.
- Isolation Forests
  - Learn an ensemble of isolation trees, i.e. a random tree structure which aims to isolate individual points. Anomalous points will be found in leaf nodes with a shorter average path length to the root node.
  - Performs well in high-dimensional problems, some interpretability possible. Heavier computation-wise → very apparent during grid search



(a) Isolating x_i (b) Isolating x_o

3. Anomaly Detection Engine Pipeline (ADEP)

Post-processing

- Group the resampled datapoints into real-world ***segments***, which represents a period in time in which an anomaly could happen (i.e. LHC injection period)
- Needed for the evaluation cause a single anomaly will have several anomalous datapoints
- Initially by applying a complex segmentation algorithm, which introduced an additional *segmentation distance* parameter
- Now improved by using a sampled controller variable, yielding exact operational segments – assuming max. one anomaly/segment
- A ground truth is assigned to each segment, based on manually labelled CERN e-logbook entries (-12h timeframe)

3. Anomaly Detection Engine Pipeline (ADEP)

Evaluation

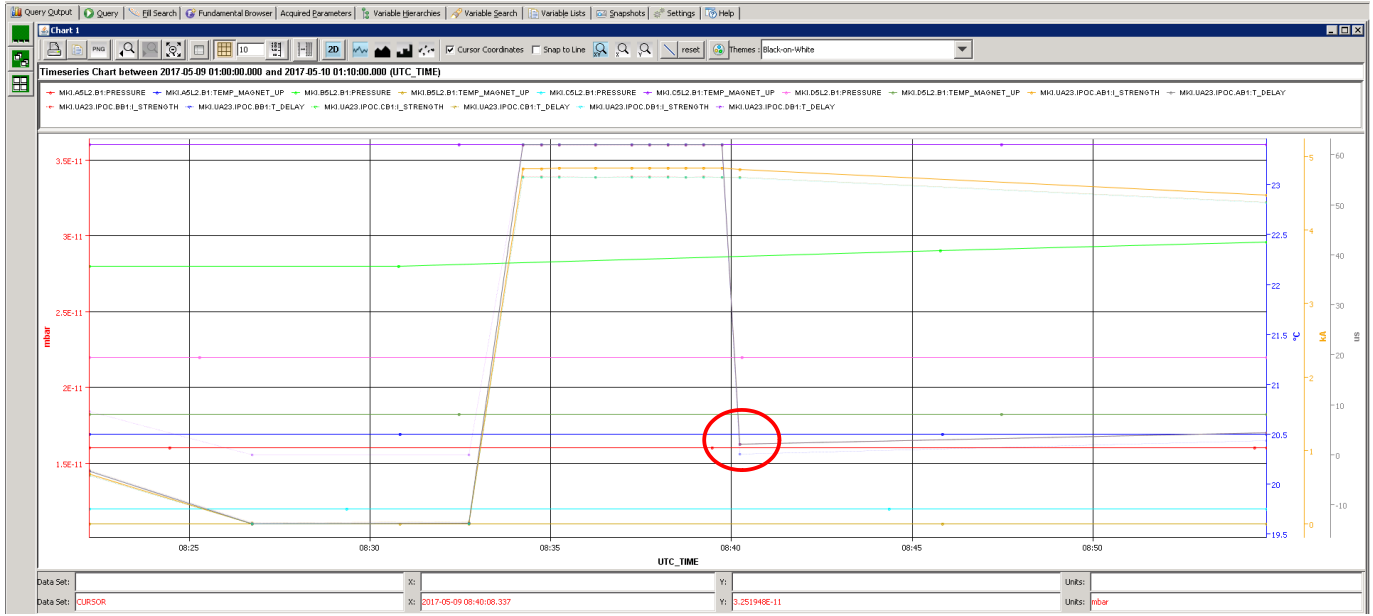
- Top-k score of individual points score the segment (k tuned by grid search)
- As a metric for the grid search, the area under the precision-recall curve is used
- Grid search has had a significant impact on improving the results, currently focused on a 3-month period (precision 0.58, recall 0.70):

	Anomaly	Normal
Detected	TP = 7	FP = 5
Undetected	FN = 3	TN = 585

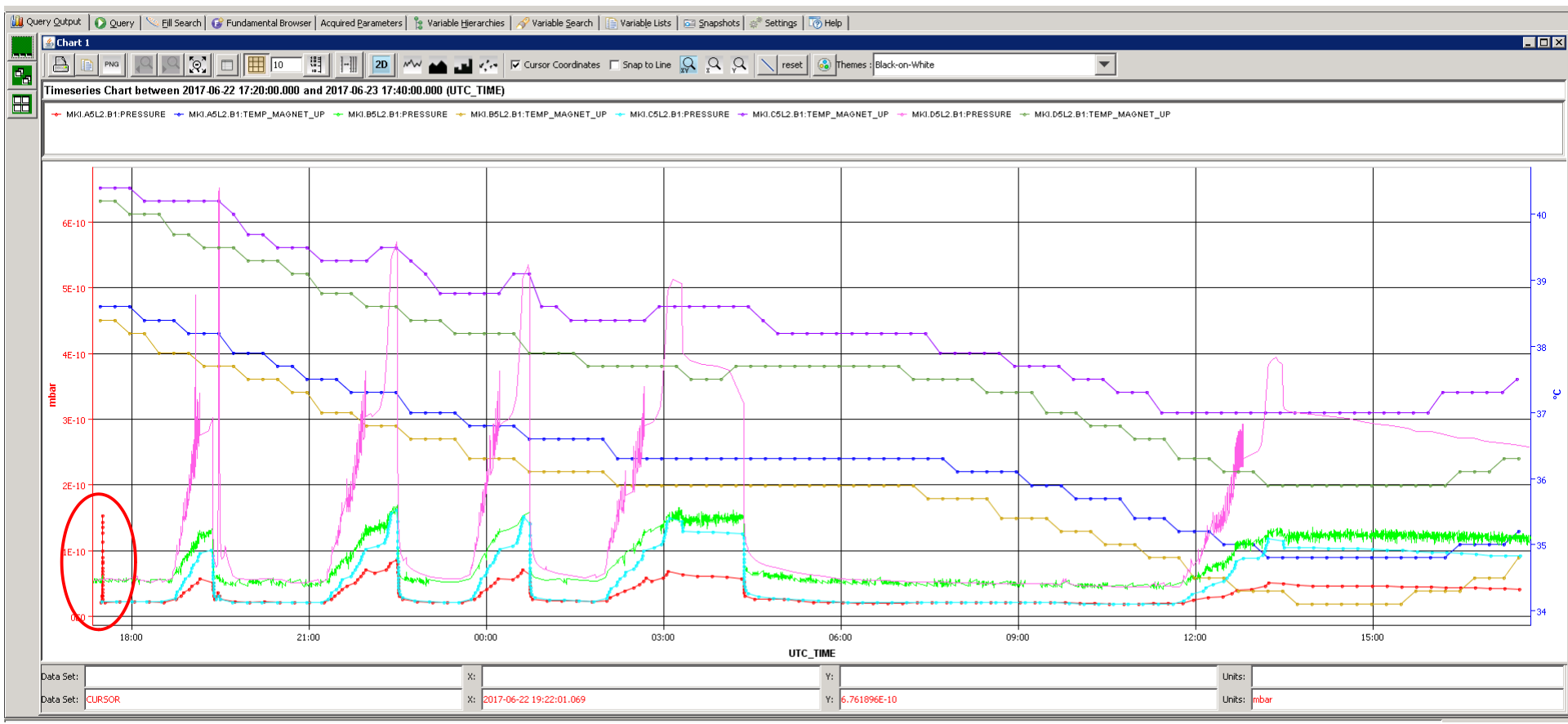
Not that false!

3. Anomaly Detection Engine Pipeline (ADEP)

Evaluation, using these false detections for bug hunting and model optimisation:



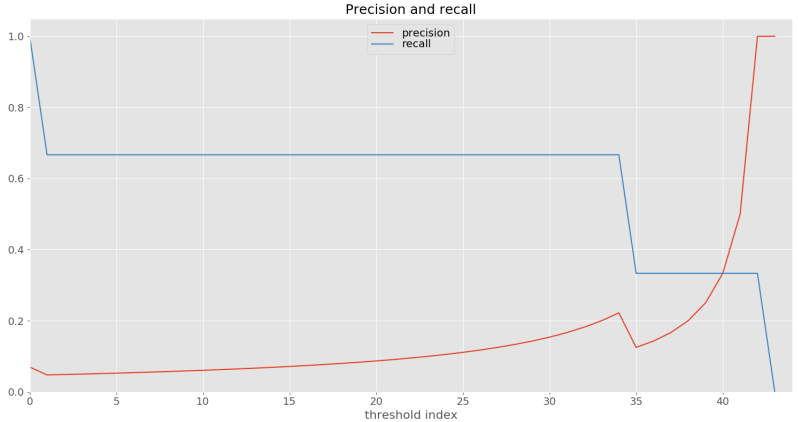
Erratic thyratron conduction (<< T_DELAY) at 8h40, FN due to wrong pre-processing filting



Pressure spike without operational influence but nevertheless interesting for magnet specialist (FP)

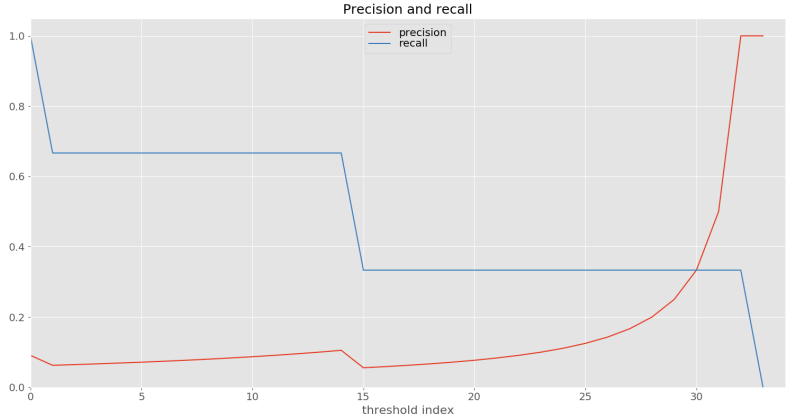
3. Anomaly Detection Engine Pipeline (ADEP)

Evaluation, another example - May 2017 **grid-search** results:



| GMM grid search execution time: 533.3 seconds

| AUC = 0.547 in 4.9s for feature_sel =all, scale_data = 1, seg_distance = STATE_MODE, a_score_method = max, types_of_labels = interventions+anomalies, params = {'covariance_type': 'full', 'init_params': 'kmeans', 'n_components': 2, 'n_init': 1, 'verbose': 1}



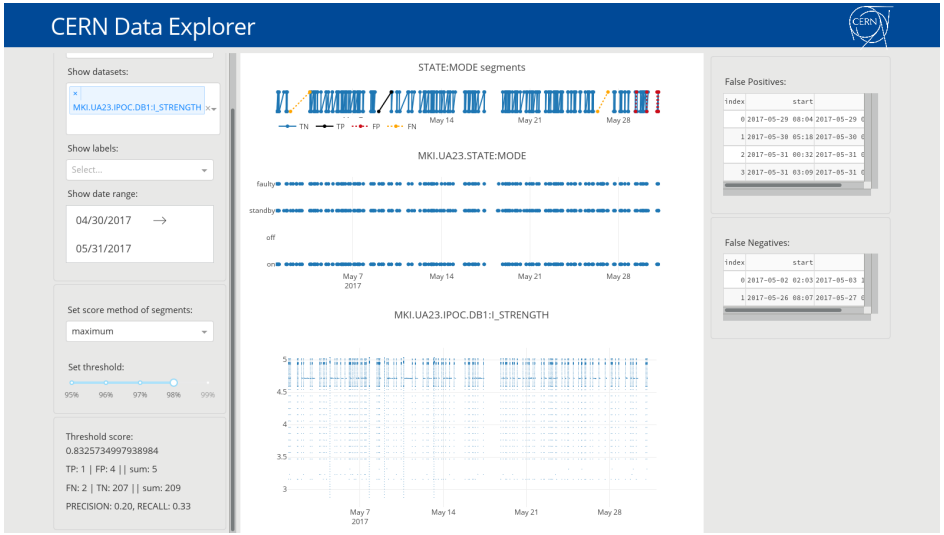
| Isolation Forest grid search execution time: 766.8 seconds

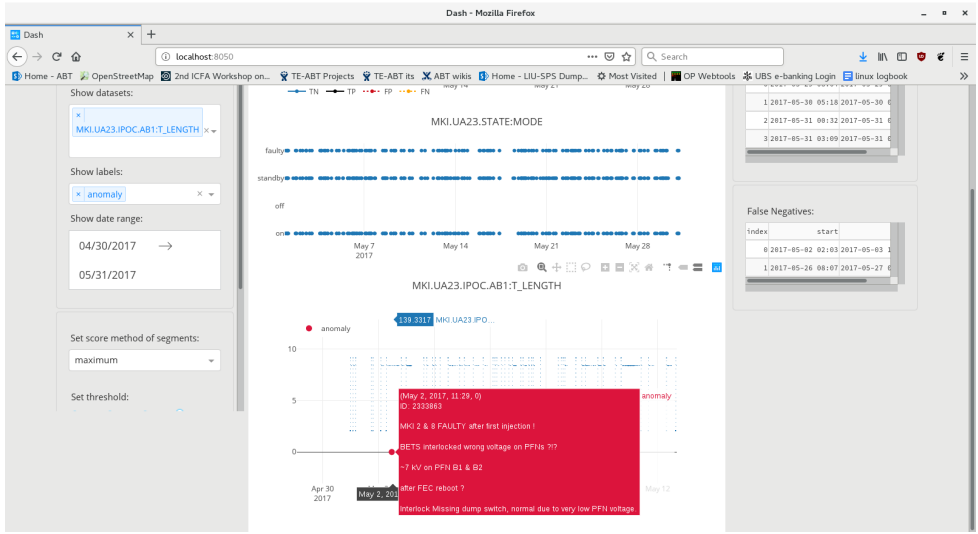
| AUC = 0.456 in 4.8s for feature_sel =all, scale_data = 0, seg_distance = STATE_MODE, a_score_method = max, types_of_labels = anomalies, params = {'max_features' : 1.0, 'max_samples': 25600, 'n_estimators': 100, 'n_jobs': 6, 'verbose': 1}

3. Anomaly Detection Engine Pipeline (ADEP)

Visualization

using Plotly Dash, interactive data browser with live validation metrics





4. Outlook

- Expand to bigger timeframes, get rid of last bugs
- Add feature selection to the grid search
- Implementation and use of the MERCS* algorithm for better interpretability
- Use of the COBRAS** algorithm to implement a 2nd clustering of the outputs of the anomaly detector, based on user-input
  - To incorporate the interesting FPs

* MERCS - https://eliavw.github.io/mercs-v5/
** COBRAS - https://dtai.cs.kuleuven.be/software/cobras/

Thanks for your attention! Questions?

www.cern.ch