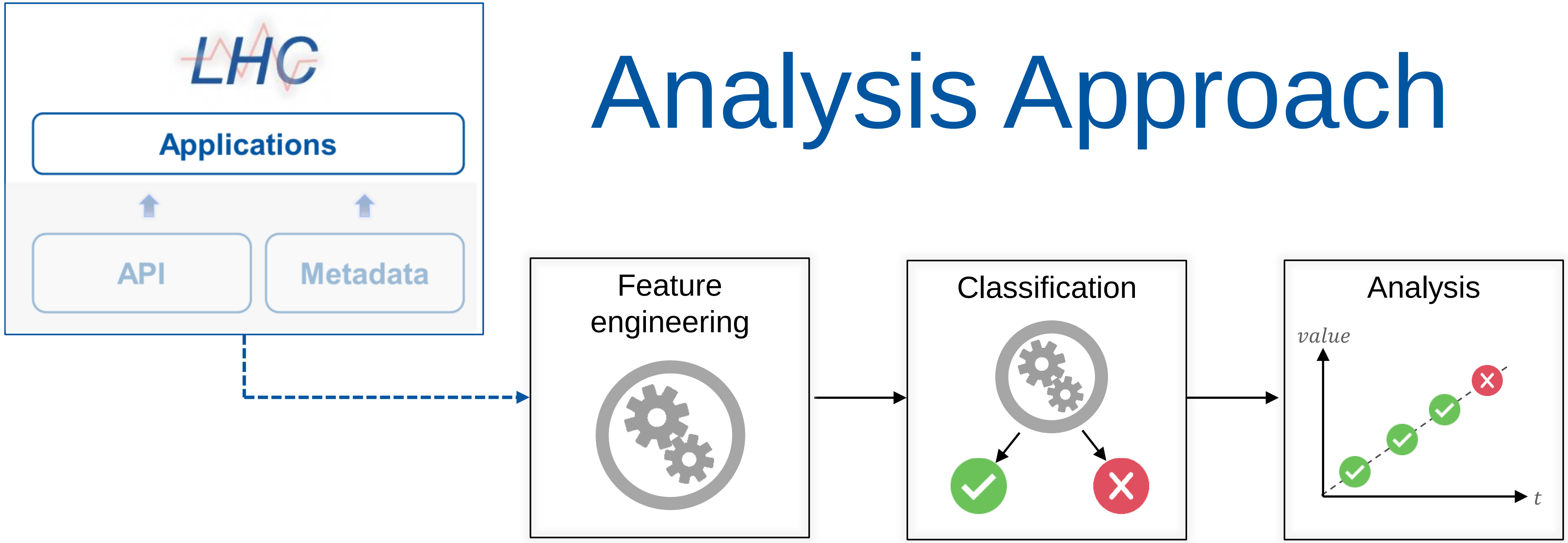
CERN

Python Implementation and Extension of the Quench Heater Monitoring Framework

Christoph Obermair

Analysis Approach



1. Feature engineering → summarize historical data

    e.g. resistance, characteristic time…

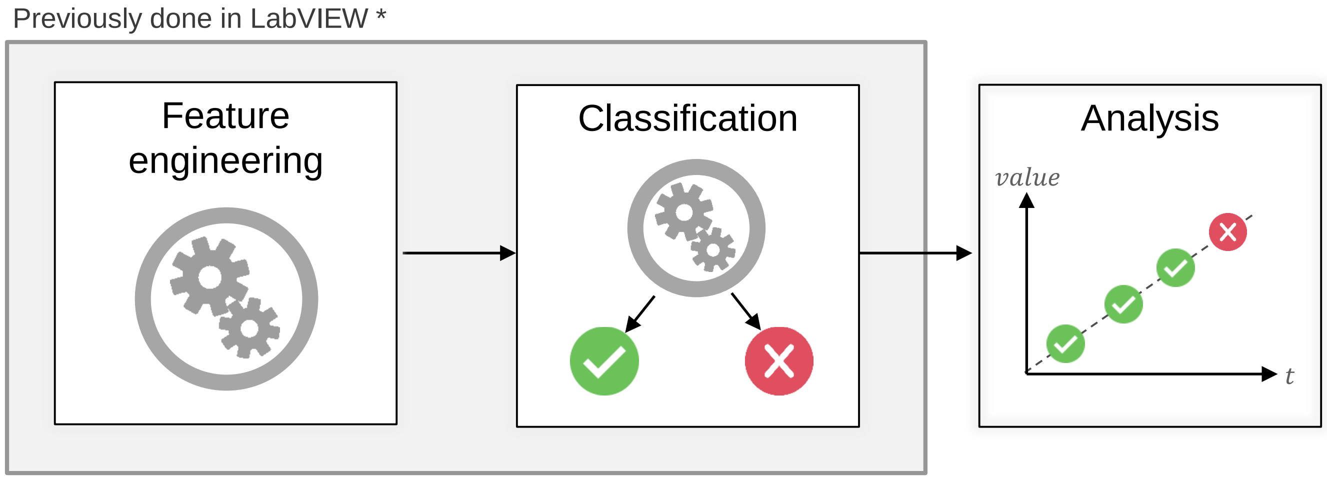
2. Classification → label features

    e.g. ✓, ✗ , no statement possible…

3. Analysis → comparison of features across time and circuits

    e.g. clustering, trend analysis…

Quench Heater Analysis (MB)



Presentation Outline:

1. Feature engineering
   a. Compression: Implementation and extension of the existing QH monitoring features
2. Classification
   a. Classification: Threshold based features classification to ✅ and ❌
   b. Comparison: Compare to LabVIEW classification and check differences with experts
   c. Extention: Extend existing classification methods
3. Trend analysis
   a. Analysis: Comparison of QH features across time and circuits

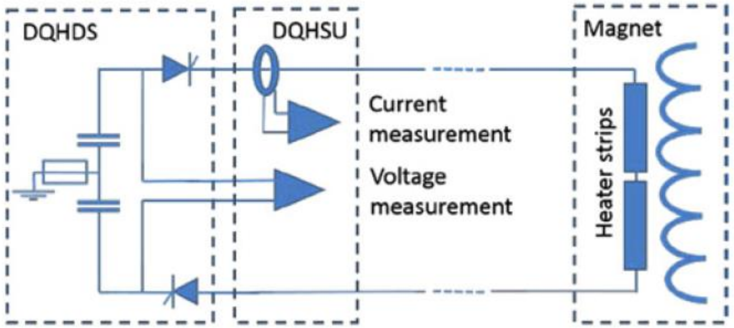
* Previously done in LabVIEW:
Z. Charifoulline et al., "Overview of the Performance of Quench Heaters for High-Current LHC Superconducting Magnets", IEEE TAS 27(4), 06/2017

1. Feature engineering

a. Compression: implementation and extension of the existing QH monitoring features

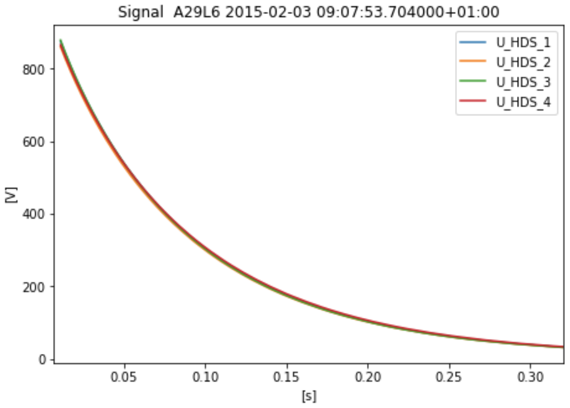
1.a Compression: Which features are extracted?

GOAL: The features should summarize the properties of the exponential decay of the quench heater signals.

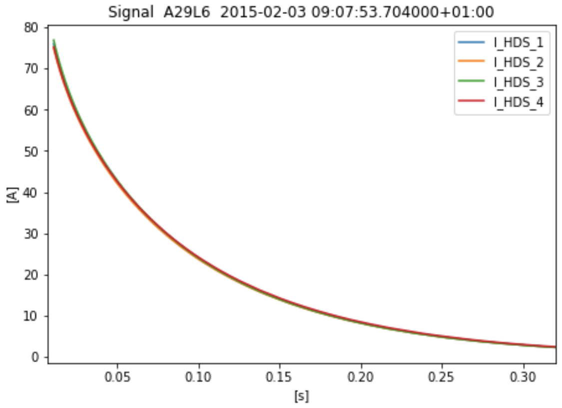


*contributed by Zinur Charifoulline

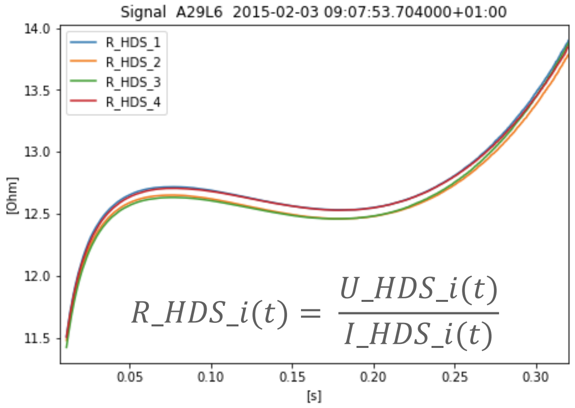
Voltage



Current



Resistance



$$R\_HDS\_i(t) = \frac{U\_HDS\_i(t)}{I\_HDS\_i(t)}$$

1.a Compression: Which features are extracted?

A. Initial values:

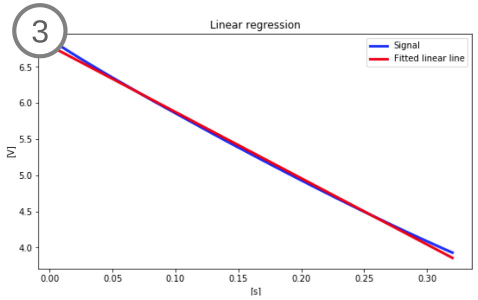
1. $\text{first}=mean(medianf(data[0{:}19],w=3))$ $\rightarrow$ *save value for U, I and R*
2. $\text{last}=mean(medianf(data[-20{:}-1],w=3))$ $\rightarrow$ *save value for U, I and R*

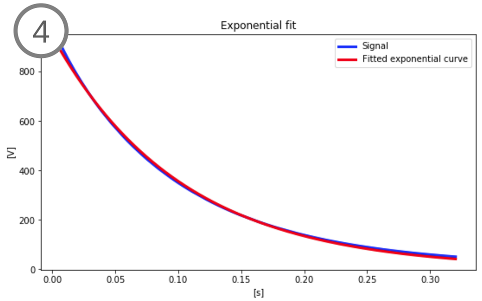
B. Characteristic time of exponential decay:

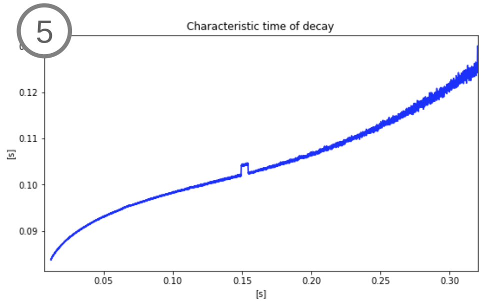
1. Charge approach: $\int_{\theta_0}^{\theta_1} f(t)d\theta = \int_{\theta_0}^{\theta_1} f_0\,(t)e^{-\frac{t}{\tilde{\tau}}}d\theta$ $\rightarrow$ *save scalar $\tilde{\tau}$ for U and I*
2. Energy approach: $\int_{\theta_0}^{\theta_1} f^2(t)d\theta = \int_{\theta_0}^{\theta_1} f_0^2(t)e^{-\frac{t}{\tilde{\tau}}}d\theta$ $\rightarrow$ *save scalar $\tilde{\tau}$ for U and I*
3. Linear regression: $\min_{p}(f(t) - (p_0 + p_1 x))$ $\rightarrow$ *save scalar p_1 for U and I*
4. Exponential fit: $\min_{p}(f(t) - p_0 e^{-p_2(t-p_1)})$ $\rightarrow$ *save scalar p_2 for U and I*

---

5. Change in characterisic time: $\frac{f(t)}{f(t)`} = -\tilde{\tau}\frac{f_0\,(t)e^{-\frac{t}{\tau}}}{f_0\,(t)`e^{-\frac{t}{\tau}}}$ $\rightarrow$ *save mean & std of* ***vector*** *$\tilde{\tau}$ for U and I*







1.a Compression: Which features are extracted?

C. Signal Similarity:

1. Normalize signals:

$$f^* = \frac{f - \min(f)}{\max(f) - \min(f)}$$

2. Euclidean distance within the signals:

$$\left\| f_i^* - f_j^* \right\|_2 = \sqrt{\sum_{t=0}^{T} (f_i^*(t) - f_j^*(t))^2} \quad \forall\, i, j = 1,2,3,4$$

$$\begin{pmatrix} 0 & \|f_1^* - f_2^*\|_2 & \|f_1^* - f_3^*\|_2 & \|f_1^* - f_4^*\|_2 \\ \ddots & 0 & \|f_2^* - f_3^*\|_2 & \|f_2^* - f_4^*\|_2 \\ \ddots & \ddots & 0 & \|f_3^* - f_4^*\|_2 \\ \ddots & \ddots & \ddots & 0 \end{pmatrix} \rightarrow save\ 6\ values\ for\ U, I\ and\ R$$

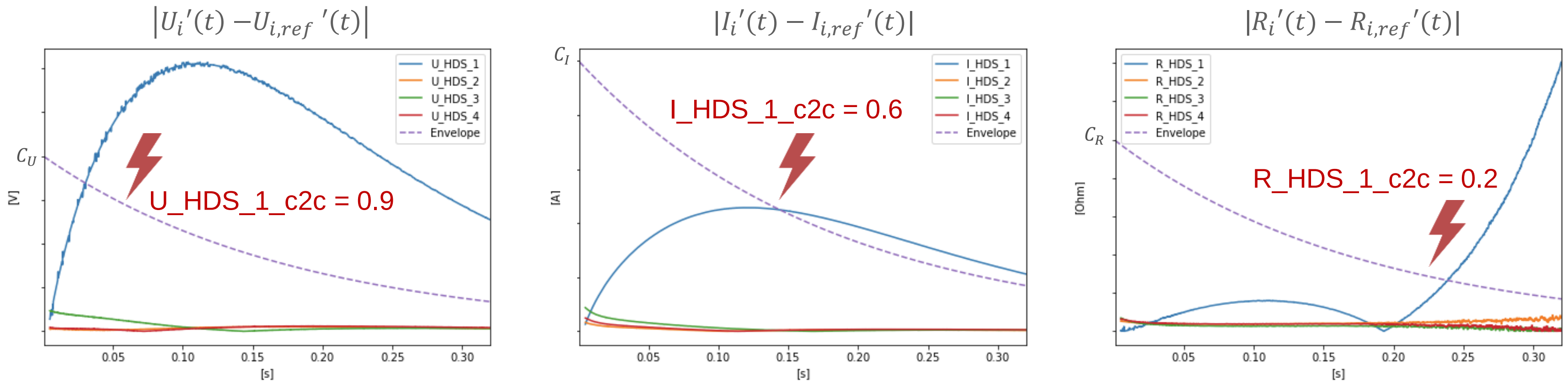
1.a Compression: Which features are extracted?

D. Subtract normalized signals with normalized reference signal and look if they are out of a certain envelope

$$|U_i'(t) - U_{i,ref}'(t)| < C_U\, e^{-\frac{t}{2\tau}}; \quad i = 1,2,3,4 \quad \rightarrow save\ \%\ for\ which\ this\ is\ true$$

$$|I_i'(t) - I_{i,ref}'(t)| < C_I\, e^{-\frac{t}{2\tau}}; \quad i = 1,2,3,4 \quad \rightarrow save\ \%\ for\ which\ this\ is\ true$$

$$|R_i'(t) - R_{i,ref}'(t)| < C_R\, e^{-\frac{t}{2\tau}}; \quad i = 1,2,3,4 \quad \rightarrow save\ \%\ for\ which\ this\ is\ true$$

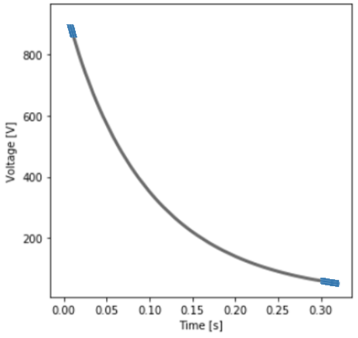


2. Classification

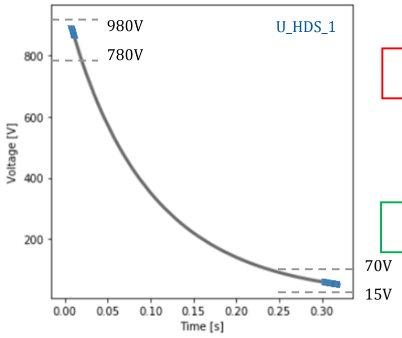
a. Classification: Threshold based features classification to ✅ and ❌
b. Comparison: Compare to LabVIEW classification and check differences with experts
c. Extention: Extend existing classification methods

2.a Classification: Threshold based signal classification

1. Difference across single component



> 20V



Discharge within range?

$780V < first < 980V$

$15V < last < 70V$

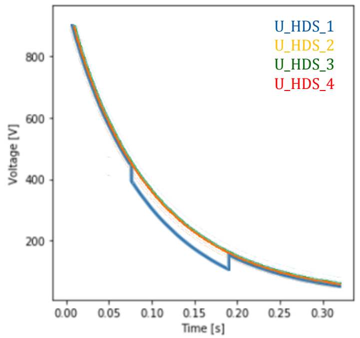
$first = mean(medianf(data[0:19], w = 3))$

$last = mean(medianf(data[-20:-1], w = 3))$

Decay if:

$last - first > 20V$

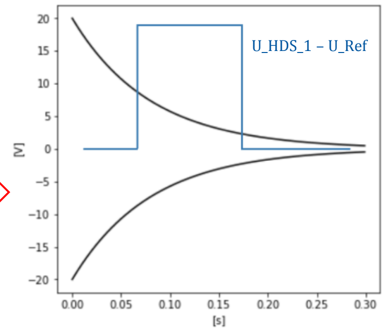
NO_Decay



Similarity:

$$\|f_i' - f_j'\|_2$$

2. Difference to other component (reference)

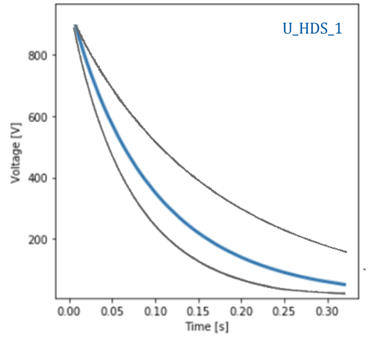


Curve2Curve comparison:

$|U_i'(t) - U_{i,ref}'(t)| < C_U\, e^{-\frac{t}{2\tau}}$;

$|I_i'(t) - I_{i,ref}'(t)| < C_I\, e^{-\frac{t}{2\tau}}$

$|R_i'(t) - R_{i,ref}'(t)| < C_R\, e^{-\frac{t}{2\tau}}$;



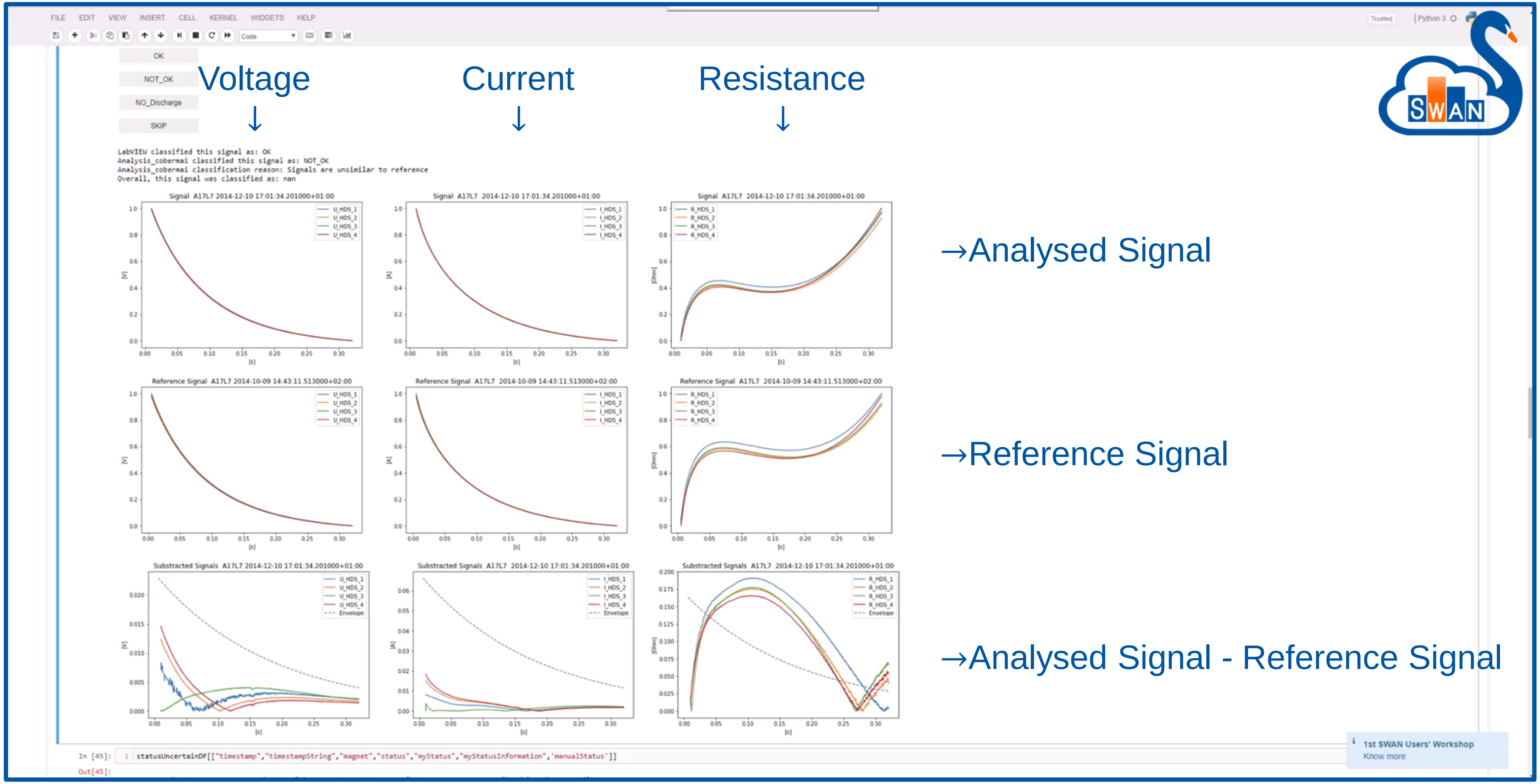
τ within range?

$\tau_i - \tau_{ref,i} < \pm 3ms\,; i = 1,2,3,4$

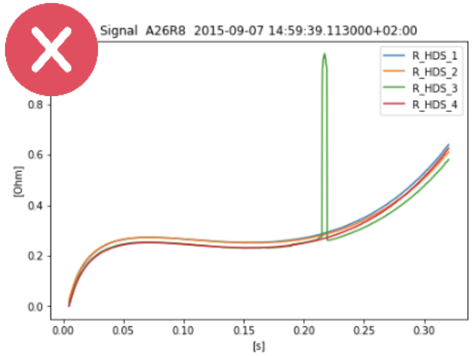
… OK

… NOT_OK

2.b Comparison: Classify difference manually



2.b Comparison: Overview



For all 7140 PM entries from 2014 to 2018:

- Classification deviates ~0.6% from LabVIEW classification:
  - Different similarity measures
  - **Critical NOT_OKs have all been detected by both methods** →
- Classification difference checked manually:
  - Manually labelled database contains 3130 OK, 116 NOT_OK and 3894 NO_Discharge
  - LabVIEW: 99.75% intersection with manually labelled database
    - Wrong classified signals: **1 OK, 17 NOT_OK***
  - My classification: 99.54% intersection with manually labelled database
    - Wrong classified signals: **24 OK, 9 NOT_OK**

→ Next: comparison of features across time and circuits

*LabVIEW feature calculation got currently updated with signal normalization before comparison as well, and should therefore perform even better in the future.

2.c Extension: Comparison of CT calculation

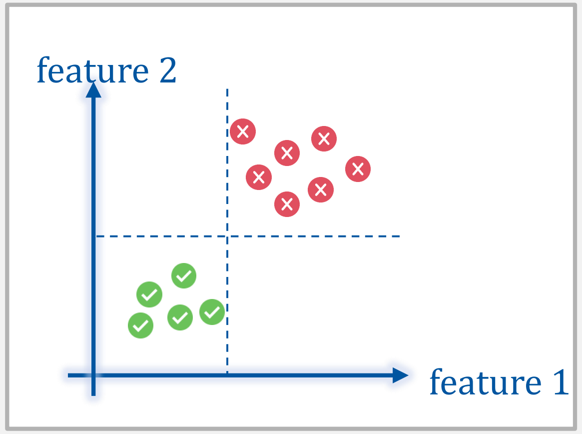
Approach	Explanation	Calculation Time / signal	Distribution [ms]*	Number of Outliers**
Charge approach	$\int_{\theta_0}^{\theta_1} f(t)d\theta = \int_{\theta_0}^{\theta_1} f_0\,(t)e^{-\frac{t}{\tilde{\tau}}}d\theta$	0.01s	80.7 ± 46.3	40
Energy approach	$\int_{\theta_0}^{\theta_1} f^2(t)d\theta = \int_{\theta_0}^{\theta_1} f_0^2(t)e^{-\frac{t}{\tilde{\tau}}}d\theta$	0.01s	79.2 ± 12.6	40
Linear regression	$\min_p(f(t) - (p_0 + p_1 x))$	0.05s	96.7 ± 6.9	2
Exponential fit	$\min_p(f(t) - p_0 e^{-p_2(t-p_1)})$	0.33s	89.5 ± 7.0	2
Change in characterisic time	$\frac{f(t)}{f(t)`} = -\tilde{\tau}\frac{f_0\,(t)e^{-\frac{t}{\tau}}}{f_0\,(t)`e^{-\frac{t}{\tau}}}$	0.01s	93.9 ± 11.3	2

*Mean value of U&I Signals < 1s
**Considering 3246 PM entries with discharges > 1s

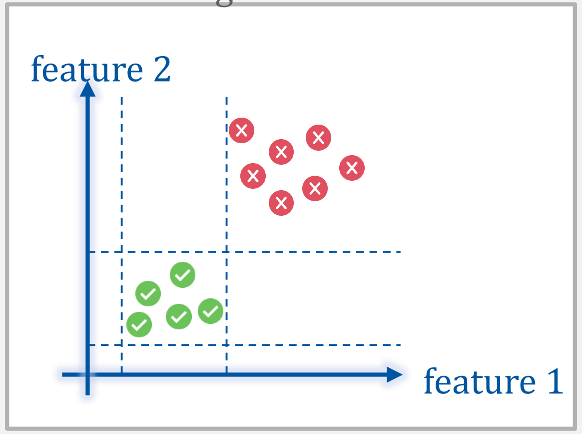
2.c Extension: Different ways for classification

Currently used thresholds:

Threshold



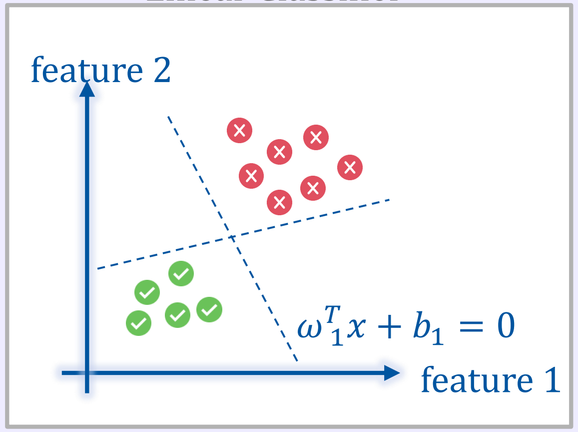
Range threshold



Threshold is chosen, by experts.

Possible extensions:

Linear Classifier



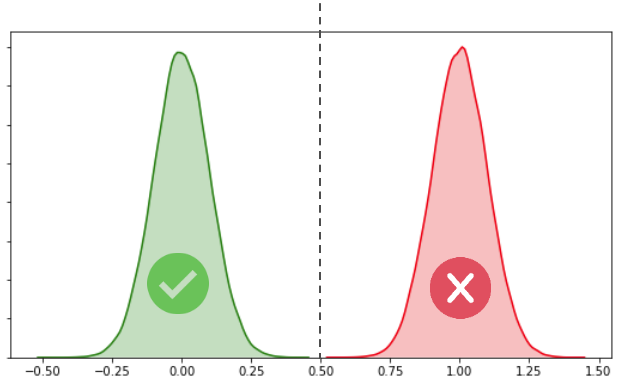
Now we have access to historical data , one could also set the threshold, such that the distance to sample points is minimized.

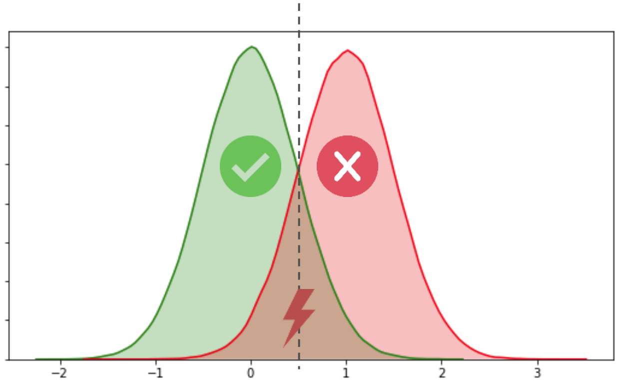
$$e.g. \min_{\omega_1} \|\omega_1\|$$

Scalable to higher dimensions and higher order decision boundaries.

2.c Extension: Hard classification vs. soft classification

An overlap in the feature distribution can lead to false signal classification



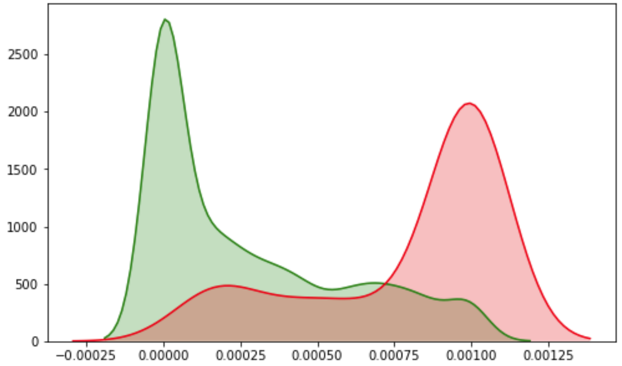


feature $= 0.25 \rightarrow$ P(✓) $= 100\%$
$\rightarrow$ P(✗) $= 0\%$

feature $= 0.25 \rightarrow$ P(✓) $= ?$
$\rightarrow$ P(✗) $= ?$

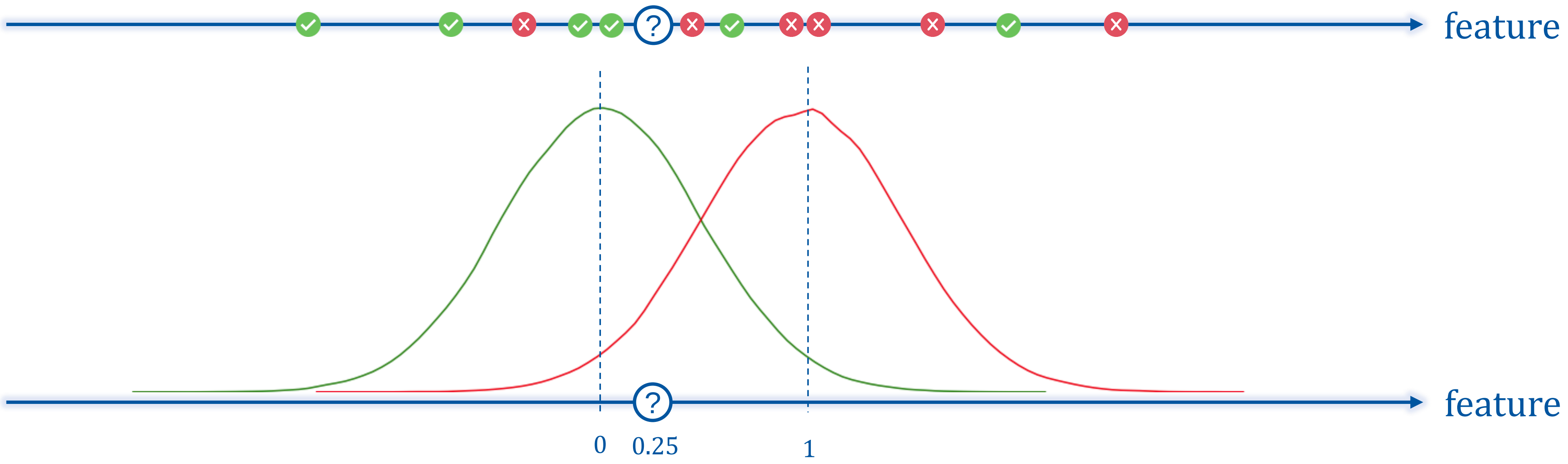
In some of our features there is an overlap in feature distribution

$mean(\tau_i - \tau_{i,ref}); i = 1,2,3,4$



2.c Extension: classify data using a Gaussian distribution

Instead of assigning a explicit state to the feature, one could also assign a probability to it.



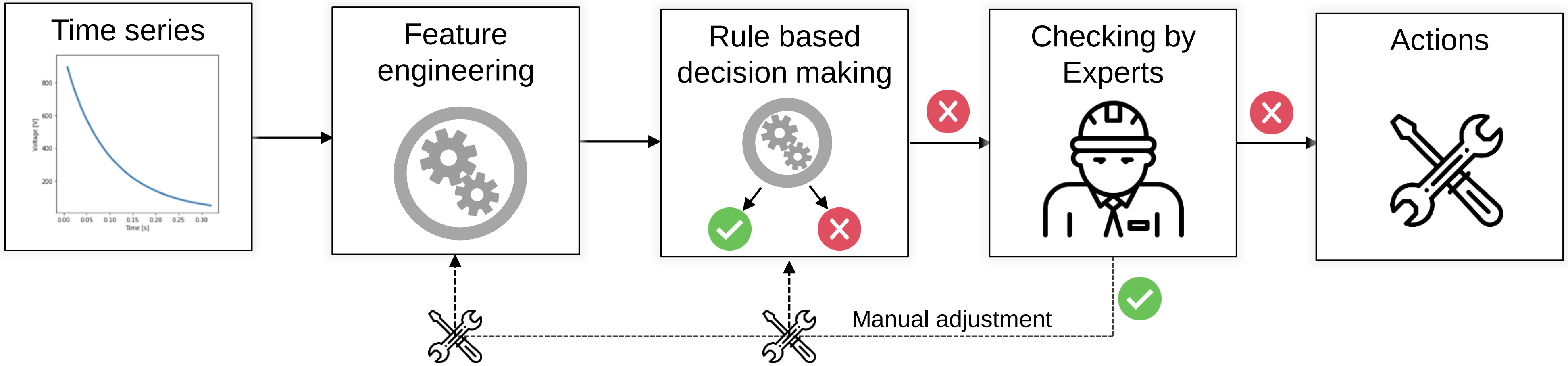
$$K(x,\mu) = e^{-\frac{\|x-\mu\|^2}{2*\sigma^2}} = e^{-\frac{(0.25-0)^2}{2*0.5^2}} = 0.882 \qquad K(x,\mu) = e^{-\frac{\|x-\mu\|^2}{2*\sigma^2}} = e^{-\frac{(0.25-1)^2}{2*0.5^2}} = 0.325$$

$$\mathrm{P}(\checkmark) = \frac{0.882}{0.882 \;\; +0.325} * 100\% = 73\% \qquad \mathrm{P}(\times) = \frac{0.325}{0.882 \;\; +0.325} * 100\% = 27\%$$

→scalable to more features

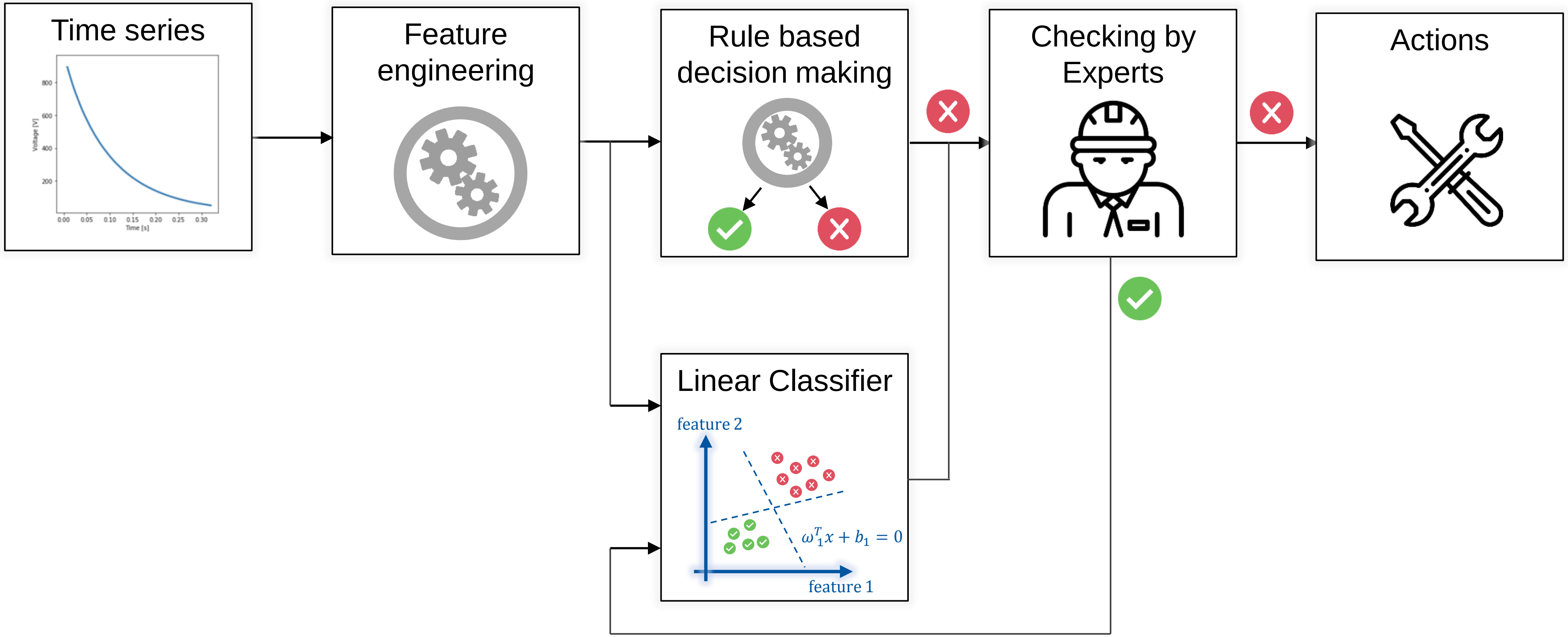
2.c Extension: Current workflow

In case of a wrong decisions, experts have to manually adjust the thresholds or the feature calculation.



2.c Extension: Workflow with feedback loop

Linear classification makes it possible incorporate expert knowledge into the classification progress



<u>Proof of concept linear classifier:</u>

- Only PM entries with discharges (3246) for classification
- A linear classifier needs data to learn from: Dividing dataset into training and validation set (50%/50%)

2.c Extension: Performance on validation set

- Performance (intersection with manually labelled dataset) on validation set:
  - 99.32% with the LabVIEW classification: 0 OK and **11 NOT_OK** wrong classified
  - 98.82% with my classification: 11 OK and **8 NOT_OK** wrong classified
  - 98.40% with linear classifier trained on my features* : 10 OK and **16 NOT_OK** wrong classified

- Linking my classification and linear classifier with logic &:
  - **4 NOT_OK** wrong classified

My Cl.	Lin. Cl.	Output
OK	OK	OK
OK	NOT_OK	NOT_OK
NOT_OK	OK	NOT_OK
NOT_OK	NOT_OK	NOT_OK

- With the used linear classifier* also the decision confidence is available
  - Can be used to adjust actions (E.g. send out warning at already 30% confidence)

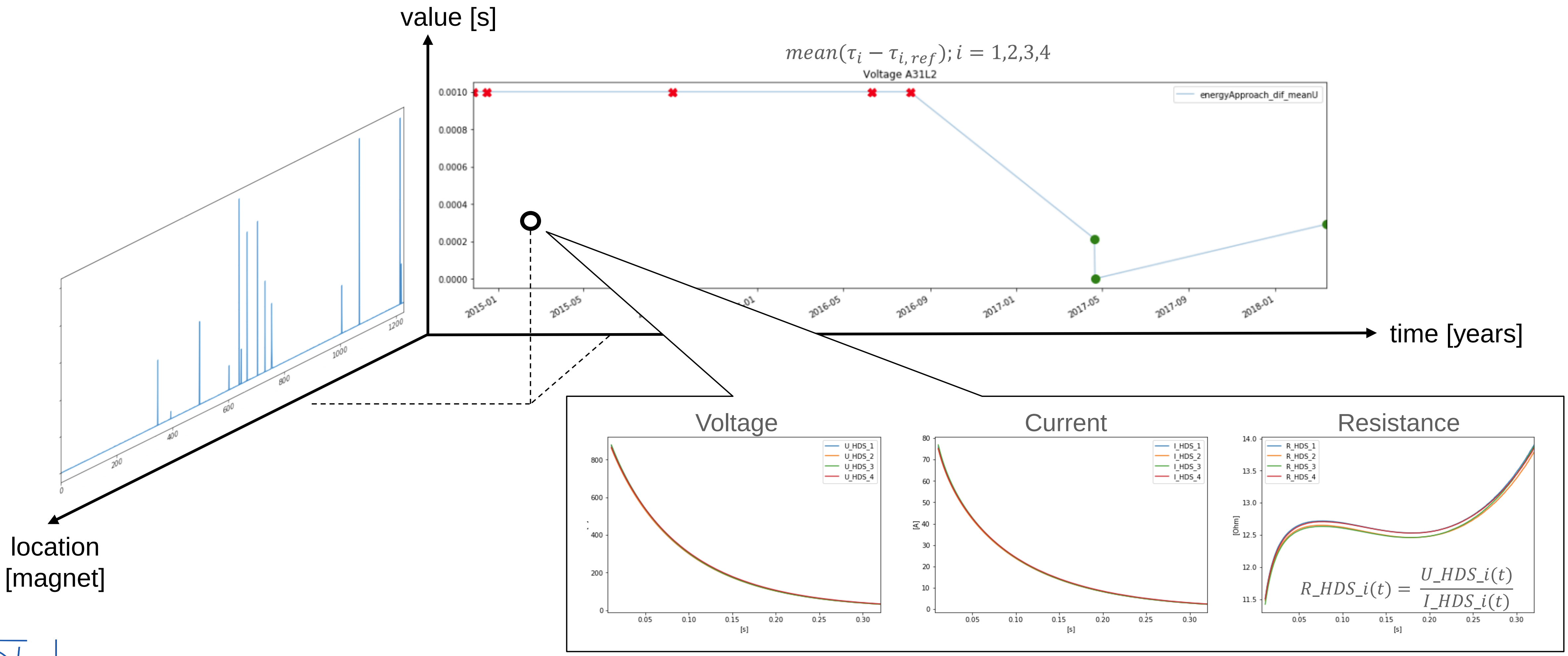
*Actually a support vector machine with a Gaussian RBF kernel was used.
Fine tuning of hyper parameter is still in progress, detailed information will be in my master thesis.

Trend analysis

a. Analysis: Comparison of QH features across time and circuits

3.a Analysis: Comparison across time and circuits

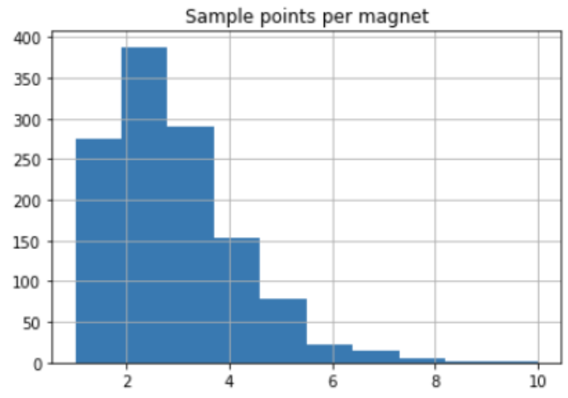
→Browsing through data with dedicated SWAN-notebooks



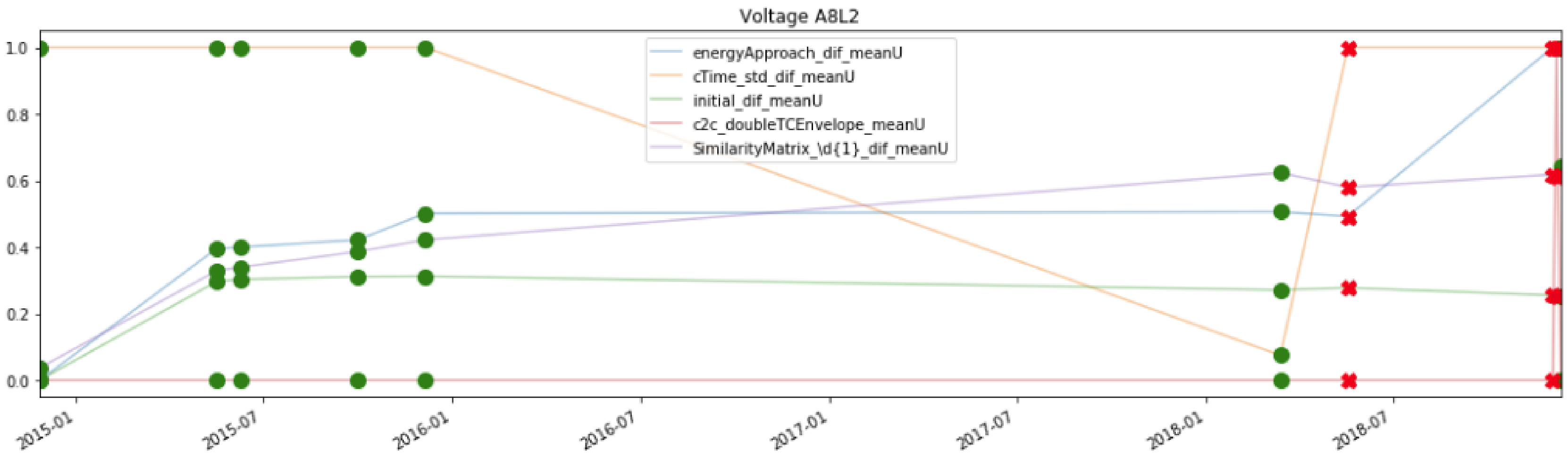
3.a Analysis: Comparison across time and circuits

1. Prefiltering: Which PM entries are eligible for prediction?

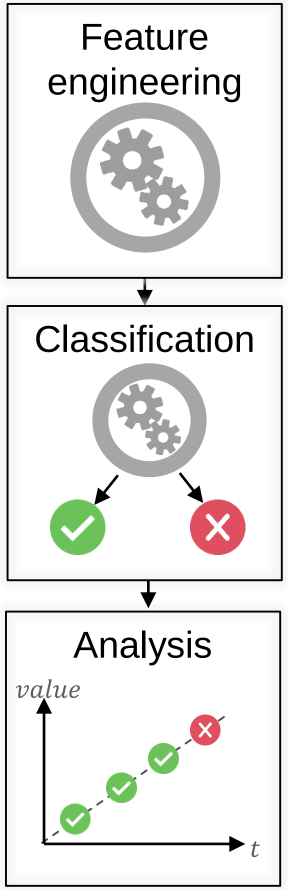
All PM entries 2014-2018	7140
All PM entries 2014-2018 with decay	3246
Entries with up-to-date reference	1929
No Hardware Comissioning tests	1906
Number of Magnets	1232
Average sample points	1.547078



2. Plotting features as a function of time



Conclusion & Outlook



→ Embedding of LabVIEW QH feature calculation

→ Extension with further features

→ Implementation of LabVIEW QH classification approach

→ Comparison of different feature calculation methods

→ New classification concept to incorporate expert knowledge

→ Browsing through data with dedicated SWAN-notebooks

→ Historical data for both quench heaters and busbar resistance available now

→The same approach can be used for further applications

CERN

1. Feature Overview

Feature name	Explanation	Parameter	Nr. Features/Parameter	Nr. Of Features	Upper Threshold	Lower Threshold	Capped at
I_MEAS		I_MEAS	1	1			
initial	mean(medianfilter(data[0:19,w=3]))	UIR	4	12	980,None,None	780,None,None	
last	mean(medianfilter(data[-20:-1,w=3]))	UIR	4	12	70,None,None	15,None,None	
chargeApproach	characteristic time of exp. decay	UI	4	8			
energyApproach	characteristic time of exp. decay	UI	4	8			
linReg	characteristic time of exp. decay	UI	4	8			
cTime_mean	mean characteristic time of exp. decay	UI	4	8			
cTime_std	std characteristic time of exp. decay	UI	4	8			
SimilarityMatrix	eucledian distance within the signals	UIR	6	18			
SimilarityMatrix_normalized	eucledian distance within the signals normalized	UIR	6	18			
initial_Ref	mean(medianfilter(data[0:19,w=3]))	UIR	4	12			
last_Ref	mean(medianfilter(data[-20:-1,w=3]))	UIR	4	12			
chargeApproach_Ref	characteristic time of exp. decay	UI	4	8			
energyApproach_Ref	characteristic time of exp. decay	UI	4	8			
linReg_Ref	characteristic time of exp. decay	UI	4	8			
cTime_mean_Ref	mean characteristic time of exp. decay	UI	4	8			
cTime_std_Ref	std characteristic time of exp. decay	UI	4	8			
SimilarityMatrix_Ref	eucledian distance within the signals	UIR	6	18			
SimilarityMatrix_normalized_Ref	eucledian distance within the signals normalized	UIR	6	18			
c2c	substracted signals have to be within envelope	UI	4	8			
c2c_doubleTCEnvelope	substracted signals have to be within envelope	UIR	4	12	0,0,0		0.1,0.1,1
c2c_doubleTCEnvelope_normalized	substracted signals have to be within envelope	UIR	4	12			
Total				**233**			

1. Feature Compression

Feature comparison							
Feature name	Explanation	Parameter	Nr. Features/Parameter	Nr. Of Features	Upper Threshold	Lower Threshold	Capped at
energyApproach_dif	energyApproach - energyApproach_Ref	UI	4	8			
cTime_mean_dif	cTime_mean - cTime_mean_Ref	UI	4	8			
cTime_std_dif	cTime_std - cTime_std_Ref	UI	4	8	1200,100, None		
initial_dif	initial - initial_Ref	UIR	4	12	0.003		
SimilarityMatrix_dif	SimilarityMatrix - SimilarityMatrix_Ref	UIR	6	18	None,None, 20		
SimilarityMatrix_normalized_dif	SimilarityMatrix_normalized - SimilarityMatrix_normalized_Re	UIR	6	18			
Total				**72**			

Feature compression							
Feature name	Explanation	Parameter	Nr. Features/Parameter	Nr. Of Features	Upper Threshold	Lower Threshold	Capped at
energyApproach_dif_mean	mean(abs(energyApproach_dif))	UI	1	2			0.0025,0.0025
cTime_std_dif_mean	mean(abs(cTime_std_dif))	UI	1	2			0.25,0.0025
initial_dif_mean	mean(abs(initial_dif))	UIR	1	3			25,25,1
c2c_doubleTCEnvelope_mean	substracted signals have to be within envelope	UIR	1	3			0.1,0.1,1
SimilarityMatrix_dif_mean	mean(abs(SimilarityMatrix_dif))	UIR	1	3			1000,150,10
Total				**13**			

Reason for abs():

- Histograms are almost symmetric
- Values could cancel each other out (e.g. Similarity Matrix)