Deep Learning for Fault Detection and Isolation

Hierarchical Networks for PHM

Gabriel Michau[1], Thomas Palmé[2], Olga Fink[1]

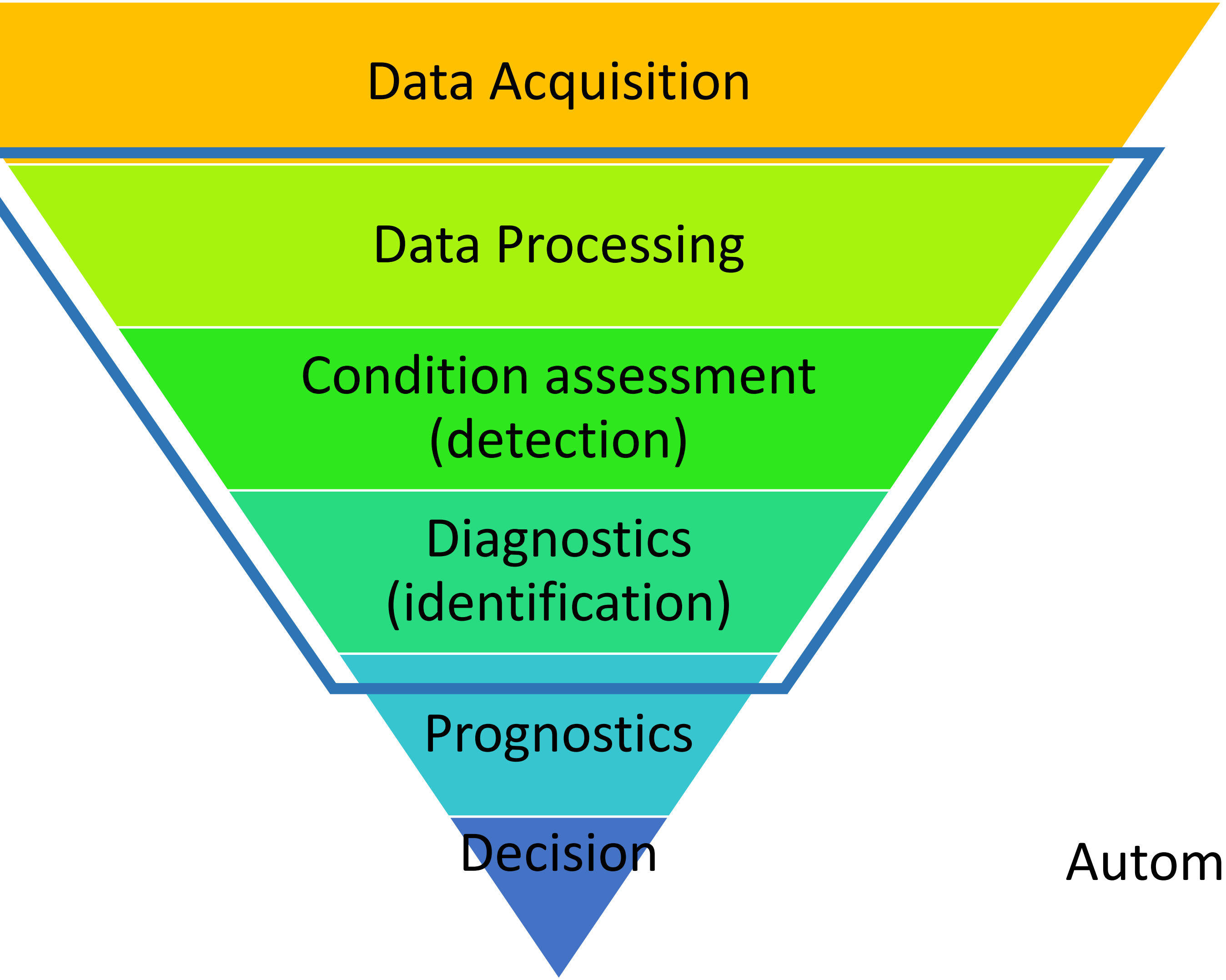
[1] - *Zurich University of Applied Science,* Switzerland

[2] - *General Electric (GE) Switzerland*

Smart Maintenance Conference 2018

Zürich

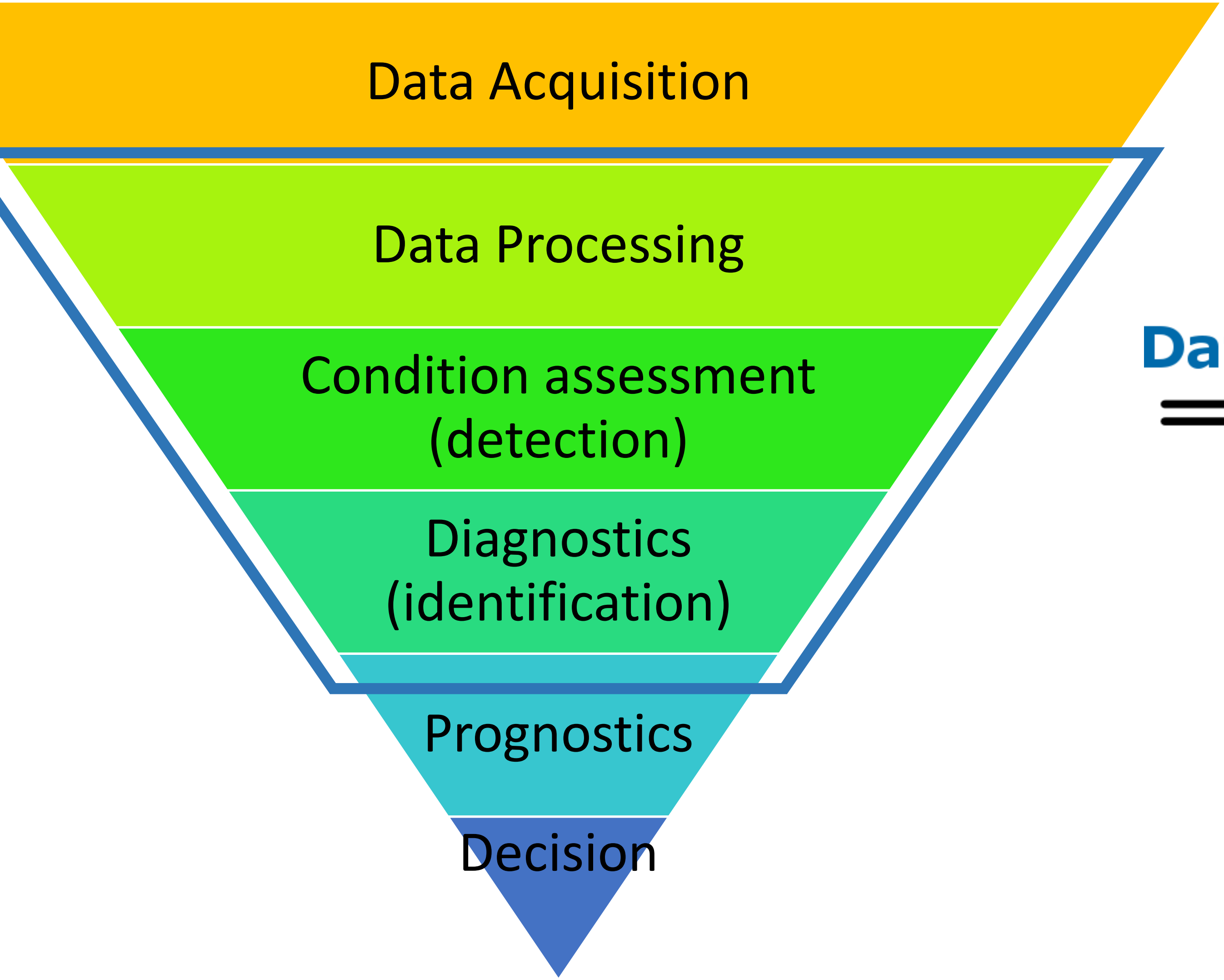
Prognostics and Health Management

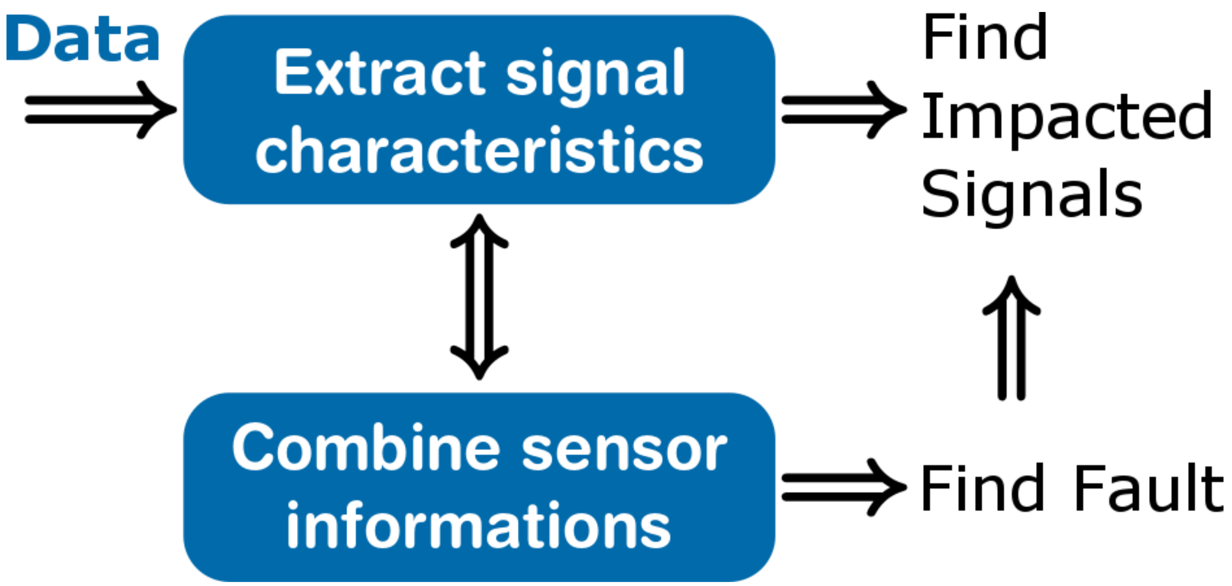


- Features
- Raise early alarm
- Provide clear indication on the cause

Automatic process in a single approach

Prognostics and Health Management





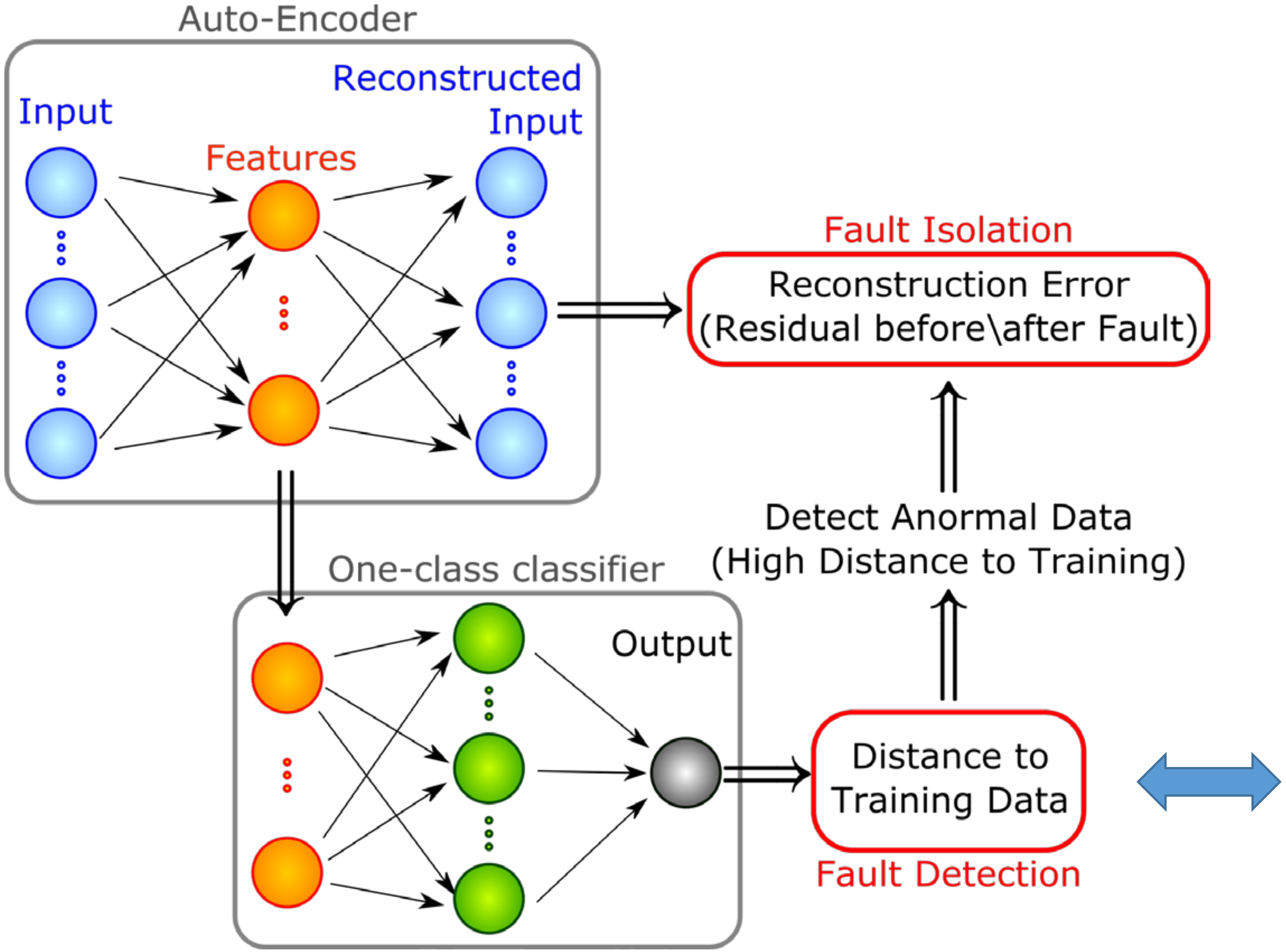
Challenges of Critial Systems

PHM for **critical** (and complex) systems

- Faults:
  - Faults are rare
  - Faults cannot be afforded → preventive maintenance
  - Possible faults are numerous (possibly hitherto unknown)
  - Consequences of faults can be diverse

- System:
  - Heterogeneous data
  - Varied operating condition over long time scale
  - Unit Specificity

Hierarchical Neural Networks

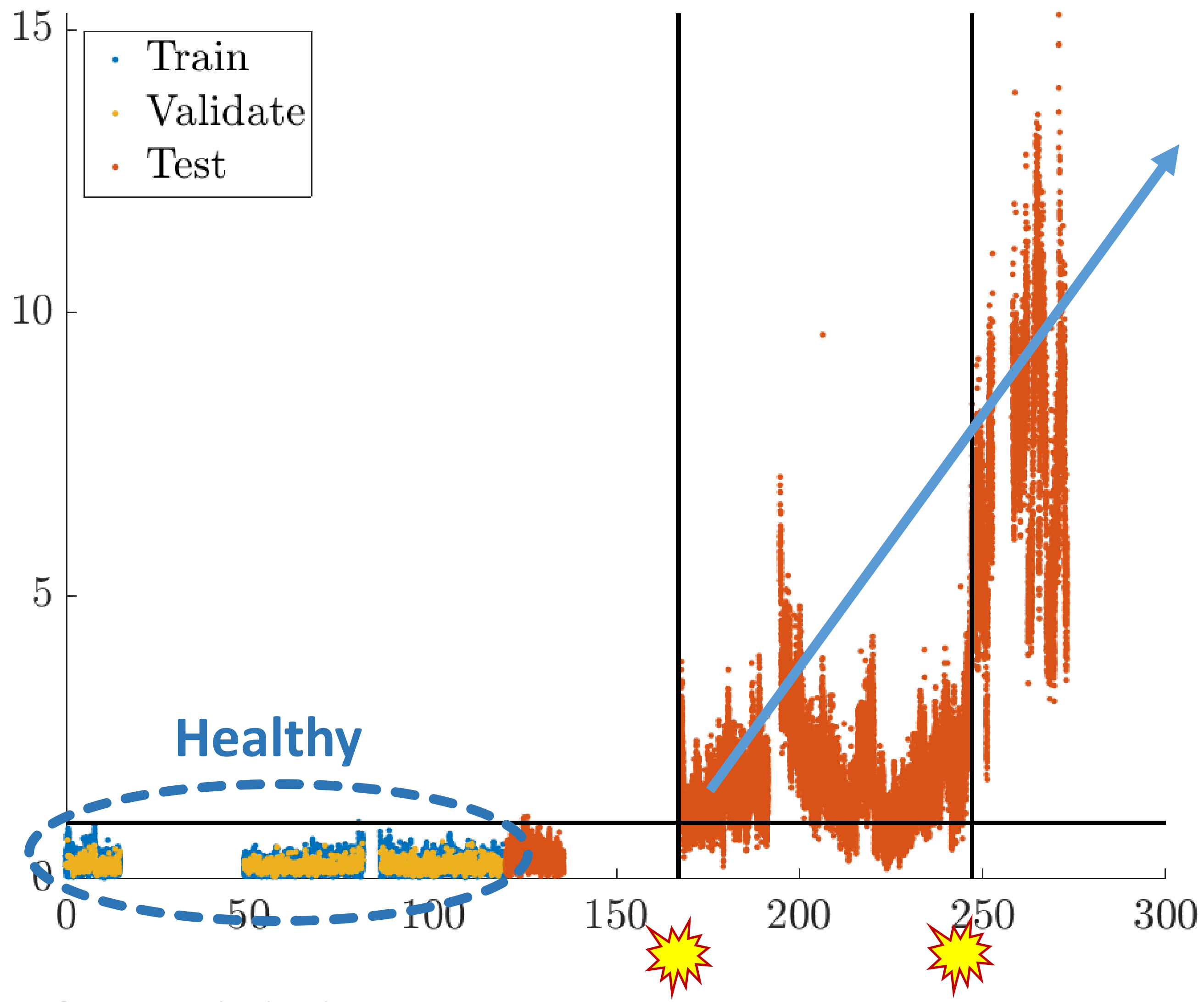


Trained with healthy condition monitoring data only

“Health Indicator”

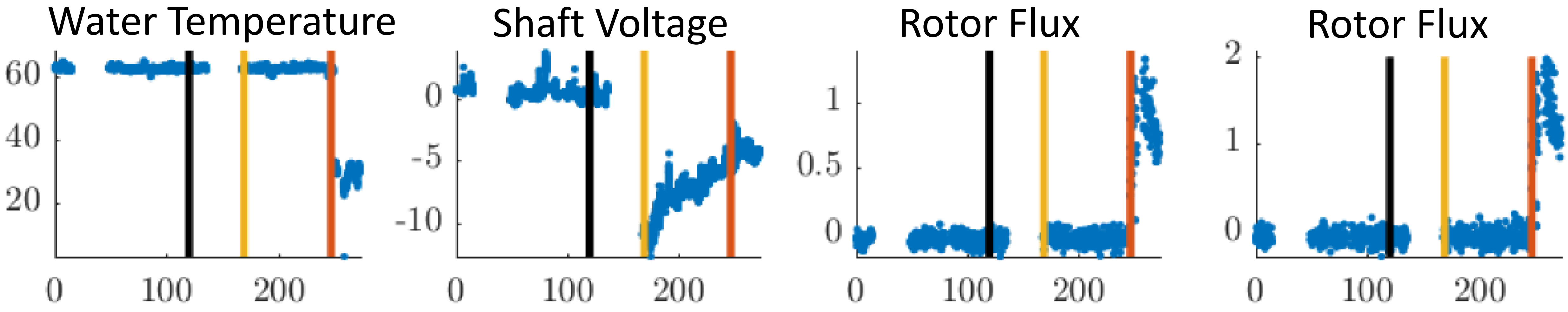
Test in real case studies

Generator with inter-turn failure



Abnormal behavior 100 days before!

Generator Diagnostics



In **agreement** with

- other models (AAKR, PCA)
- with expert knowledge.

At no additional cost!

New Challenges

What if, I don't have enough training data?

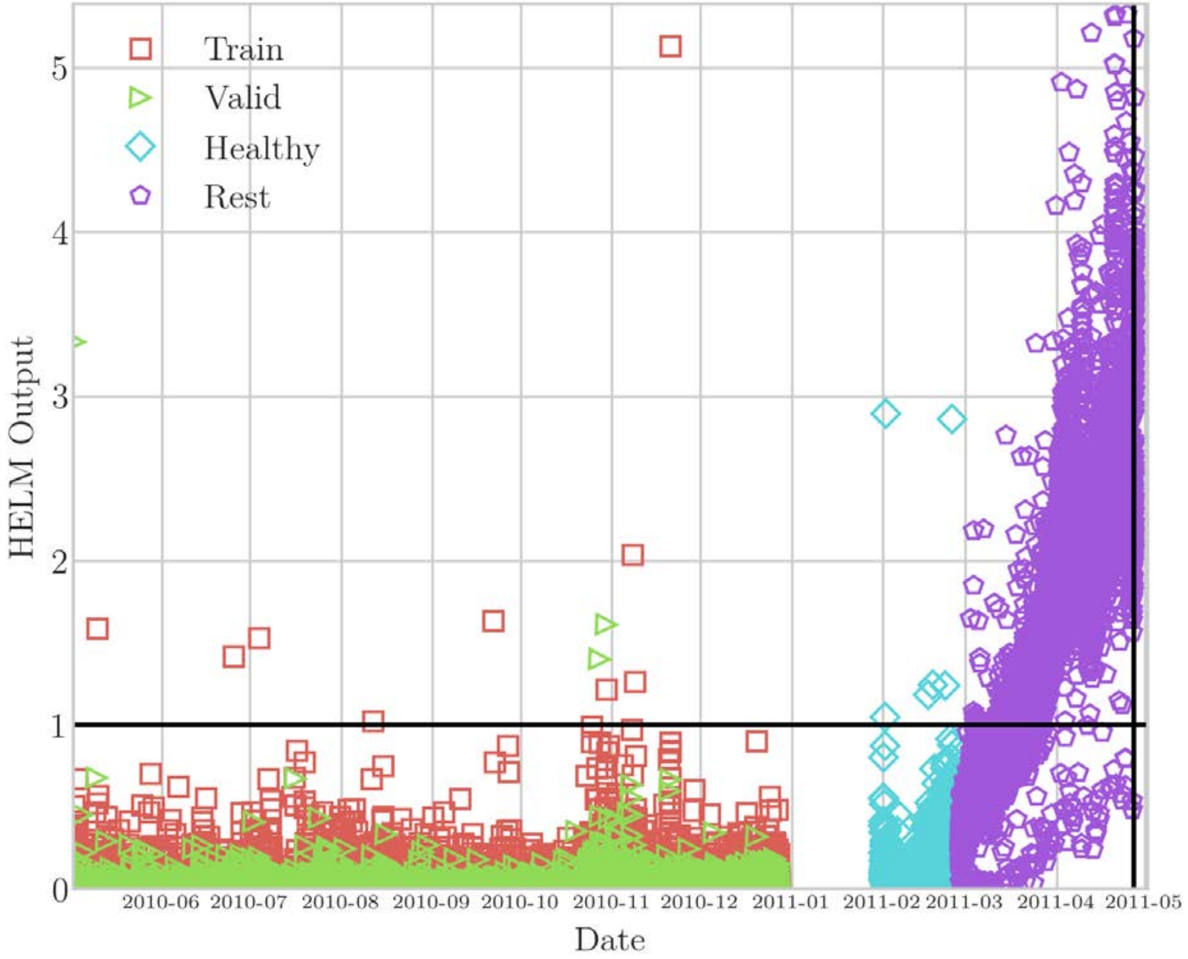
- What if my system is new (or has been refurbished)?
- What if I am expecting operating conditions to change?

$\Rightarrow$Use data from other similar systems

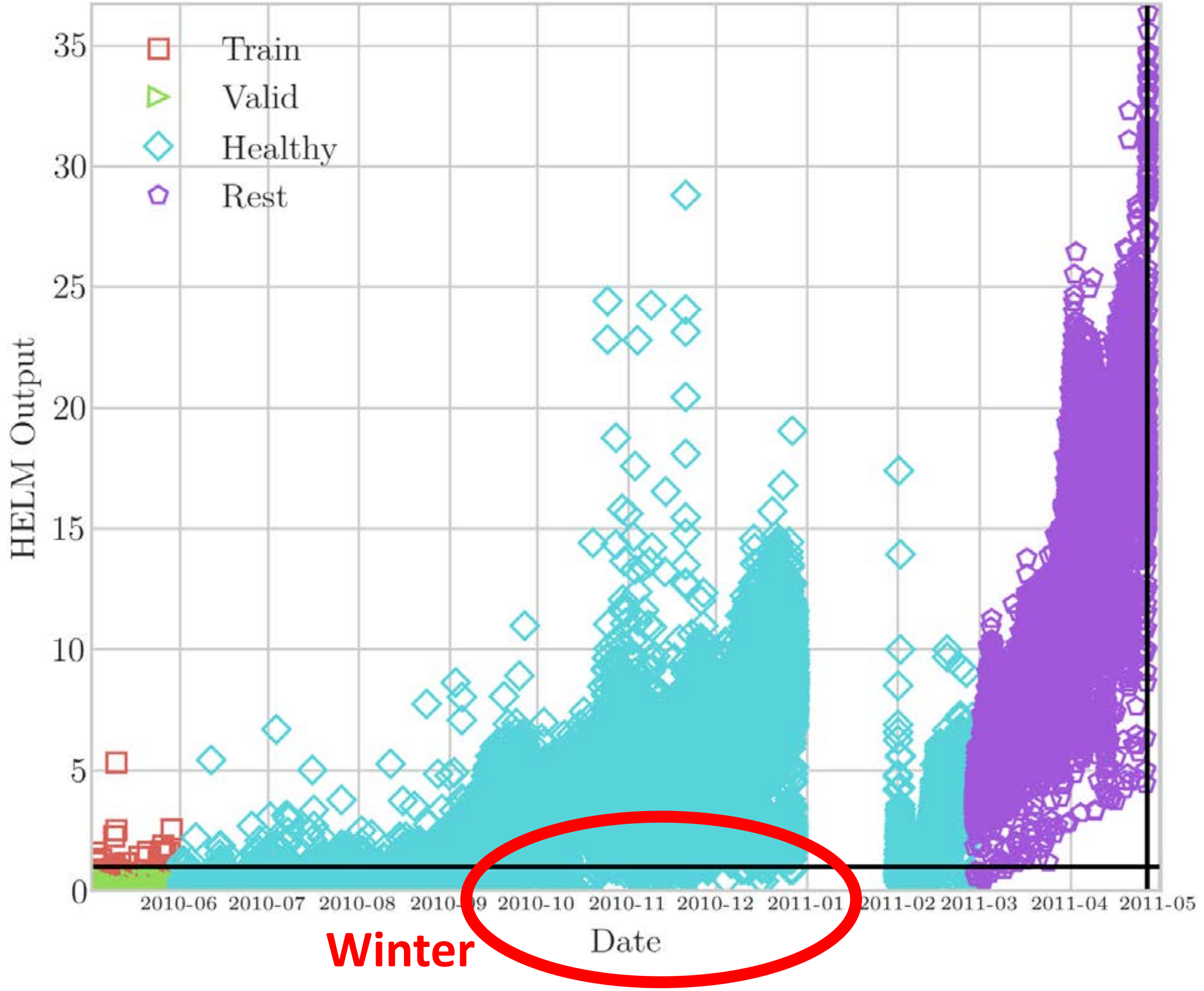
Fleet approach to PHM (from manufacturer perspective)

Example with a stator (vane failure)

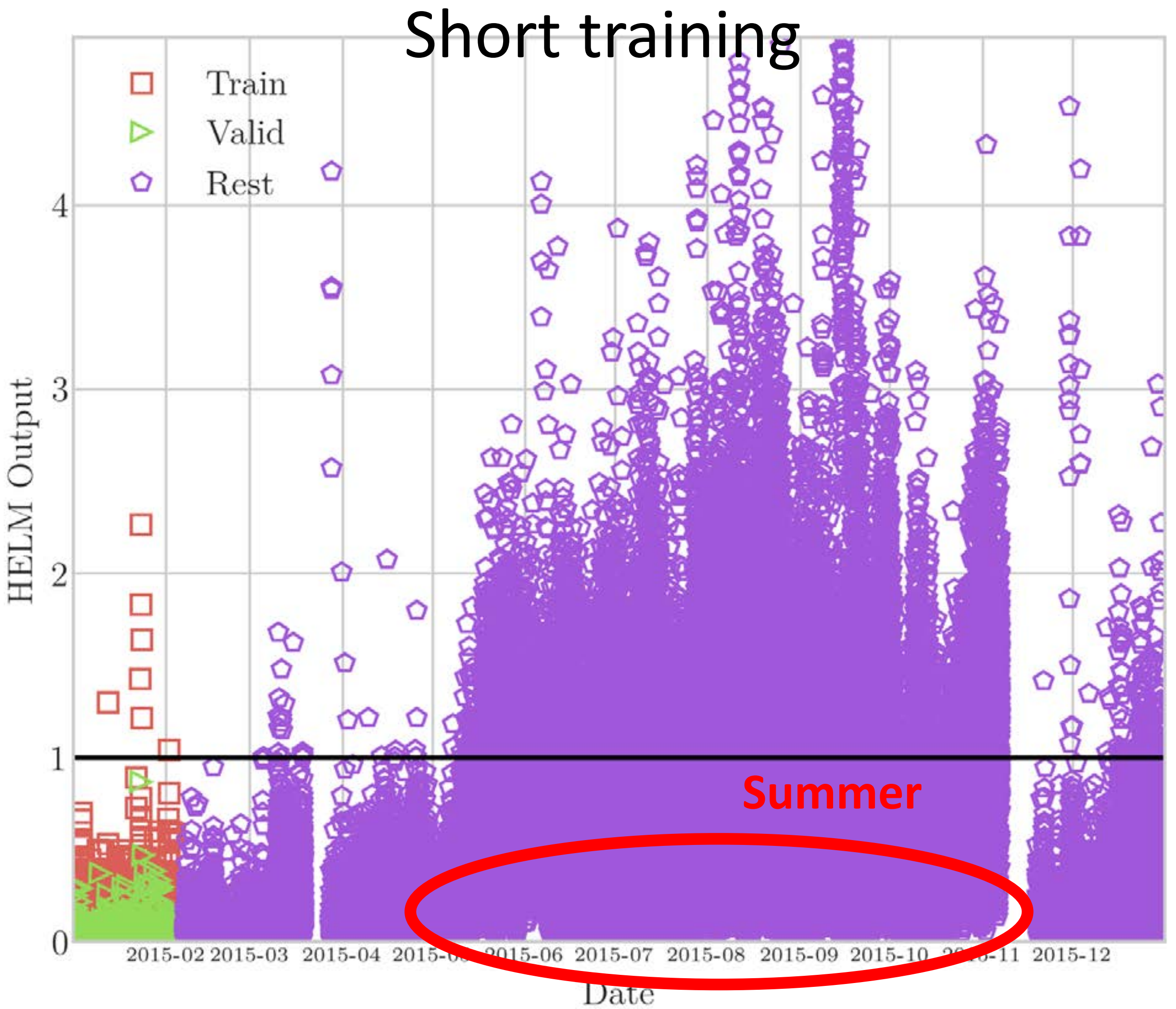
Long training



Short training



Healthy Stator



Subfleet selection

- What data do I need?
  - Not everything at once because
    1. It is too much
    2. It will contains operating points that are not relevant to my unit
    3. Completely different operating conditions might hide faults

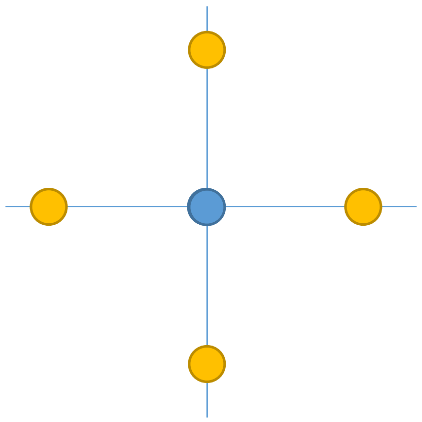
  $\Rightarrow$ I need data relevant for my unit.

- Within a (manufacturer) fleet, all units are similar.

$\Rightarrow$ Need to identify other units with similar operating conditions

Subfleet selection

- Multi-dimensional datasets comparison
  - Resource and time consuming (multi-dimensional distance and distributions)
  - Difficult (number of nearest neighbors evolves as n.D)



Subfleet selection

- **Solution:** The Hierarchical Network has proven to be efficient in measuring how well the test data correspond to the training data.

⇒**Idea**:

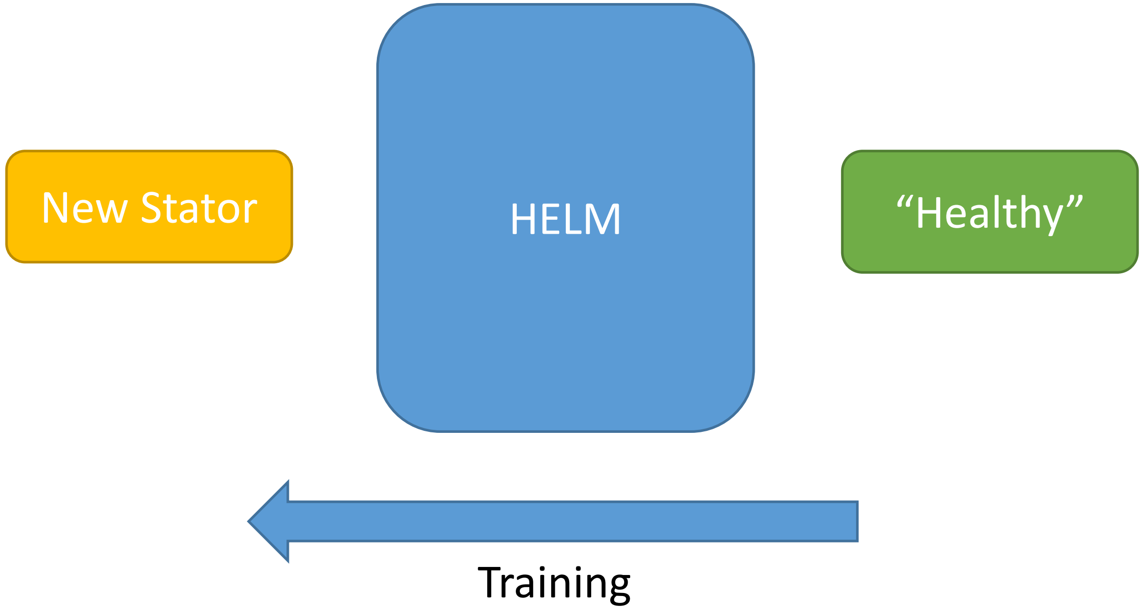
1. Train HELM on one unit, test with data from other units
2. select those that are detected as similar.

⇒Computationally efficient:

- HELM takes all dimensions and output a single indicator
- 1 HELM to train per unit (instead of N^2 dataset comparisons)

Proposed Methodology

1 – Train HELM on "new" stator i

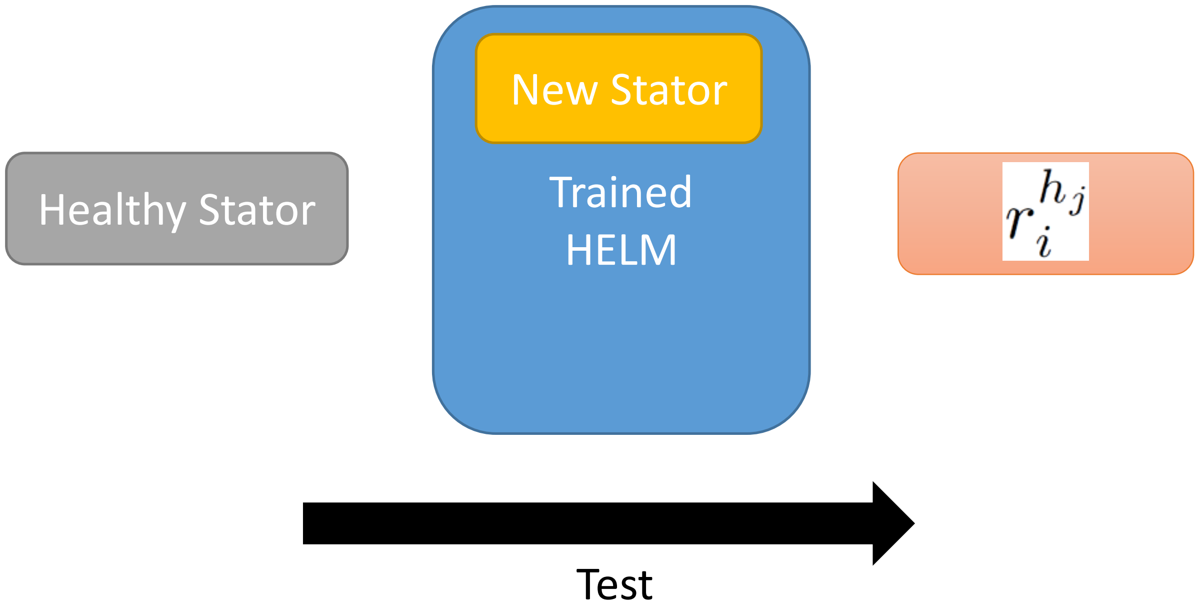


Proposed Methodology

1 – Train HELM on “new” stator i

2 – Test all other Stator h_j

$$r_i^{h_j}$$



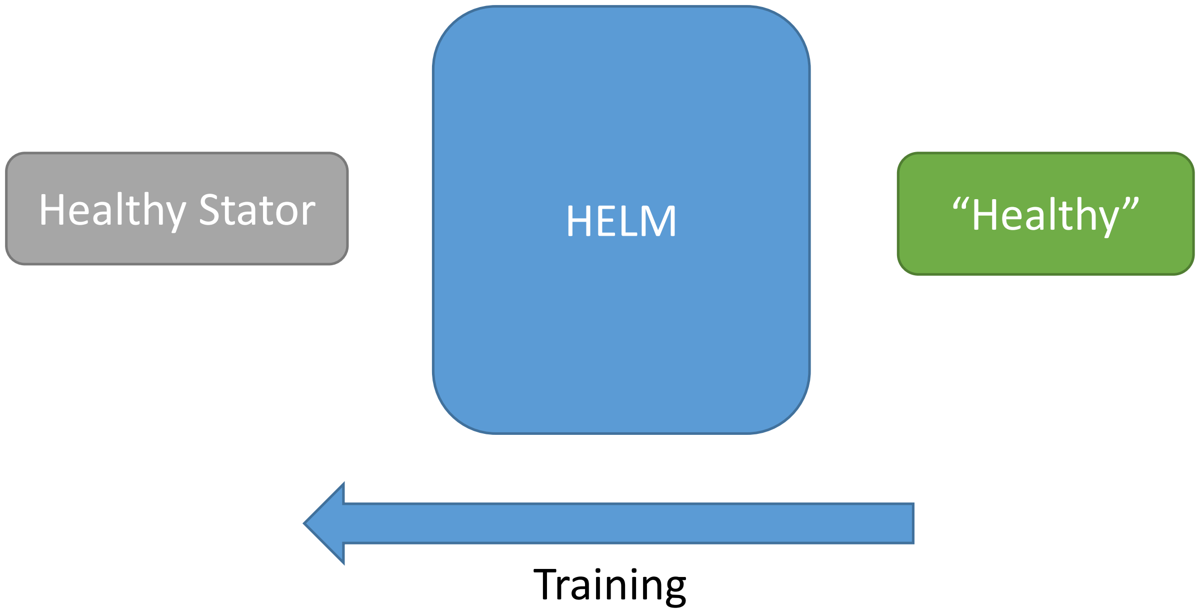
Proposed Methodology

1 – Train HELM on “new” stator i

2 – Test all other Stator h_j

$$r_i^{h_j}$$

3 – Train HELM with other stator



Proposed Methodology

1 – Train HELM on "new" stator i

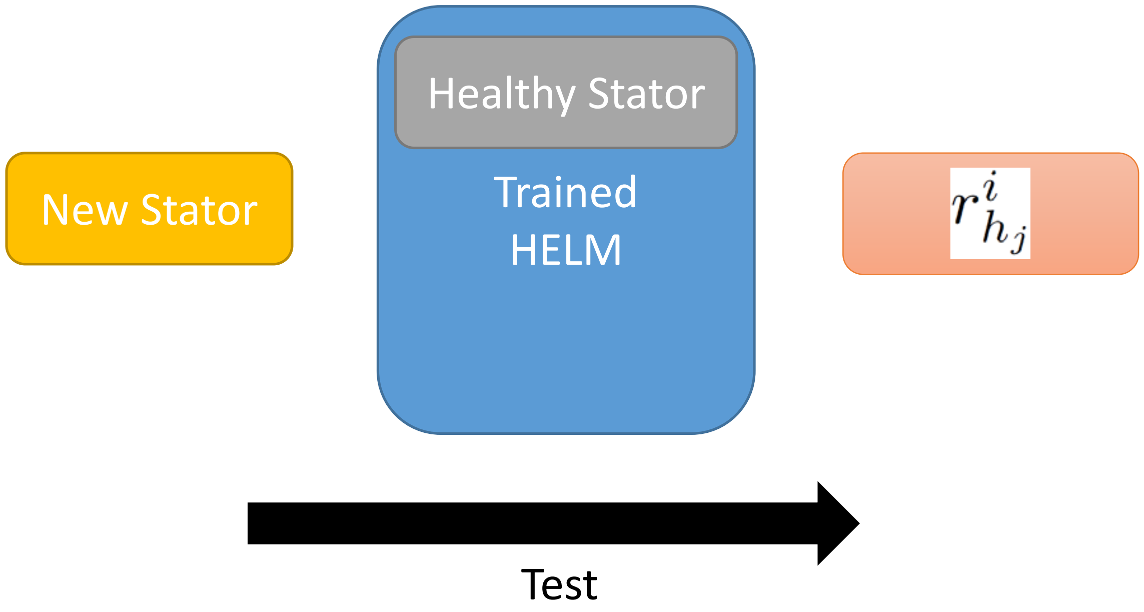
2 – Test all other Stator h_j

$$r_i^{h_j}$$

3 – Train HELM with other stator

4 – Test with "new" stator

$$r_{h_j}^i$$



Proposed Methodology

1 – Train HELM on “new” stator i

2 – Test all other Stator h_j

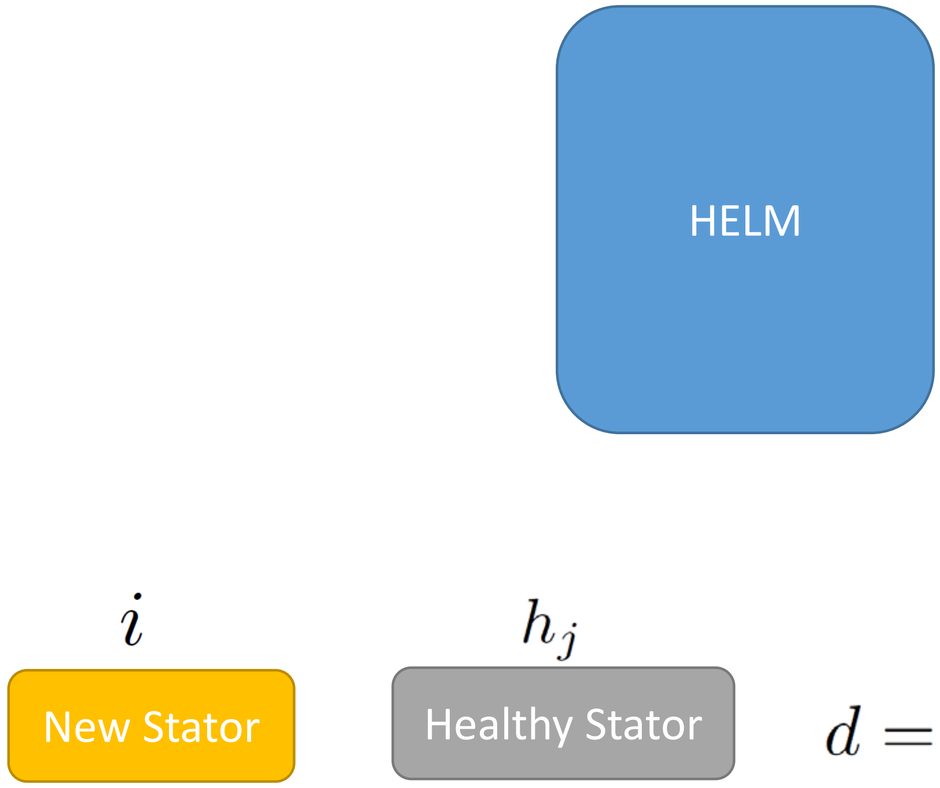
$$r_i^{h_j}$$

3 – Train HELM with other stator

4 – Test with “new” stator

$$r_{h_j}^{i}$$

5 – Dissimilarity measure:



$$d = \frac{r_i^{h_j} + r_{h_j}^{i}}{2}$$

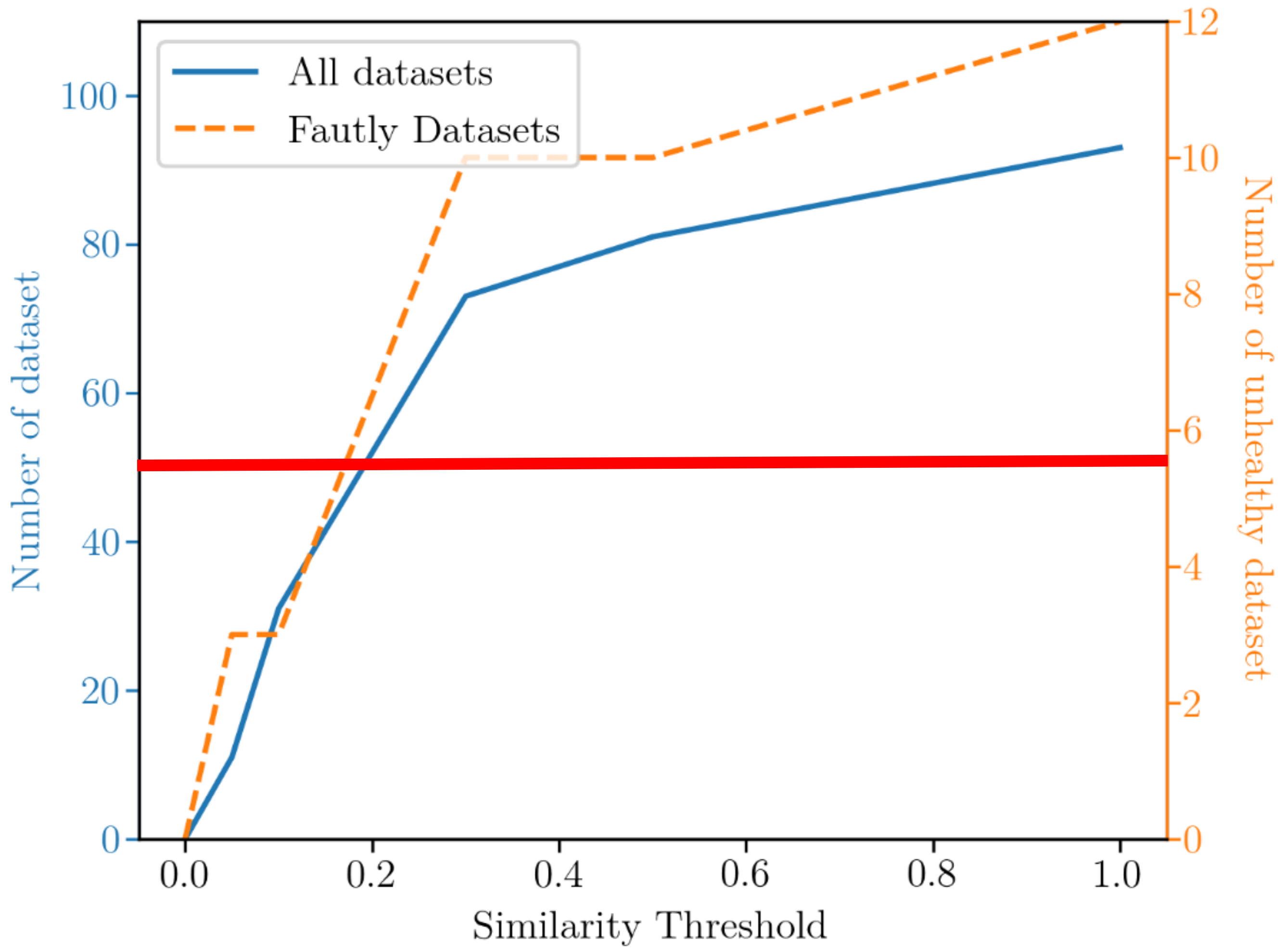
Similar datasets

$$d = \frac{r_i^{h_j} + r_{h_j}^i}{2}$$

Similar dataset if *d* below a threshold

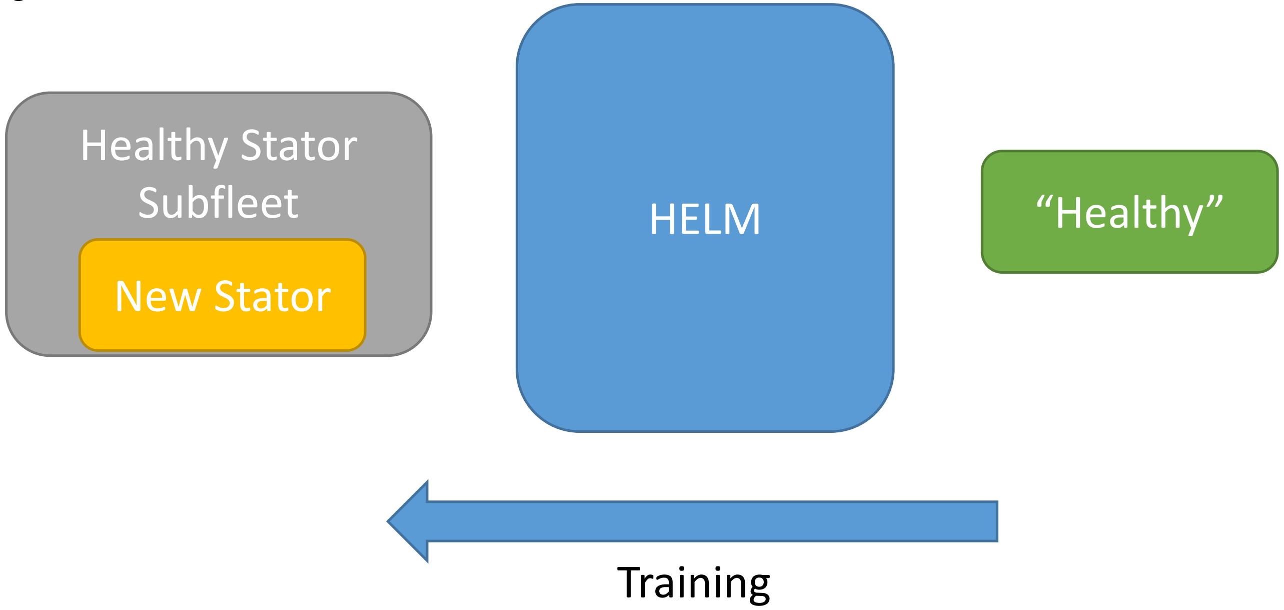
Need to allow for more than 20% of dissimilar point to find one similar dataset for half of the dataset

This fleet has very dissimilar units
Difficulty to find subfleets



Individual Asset Monitoring

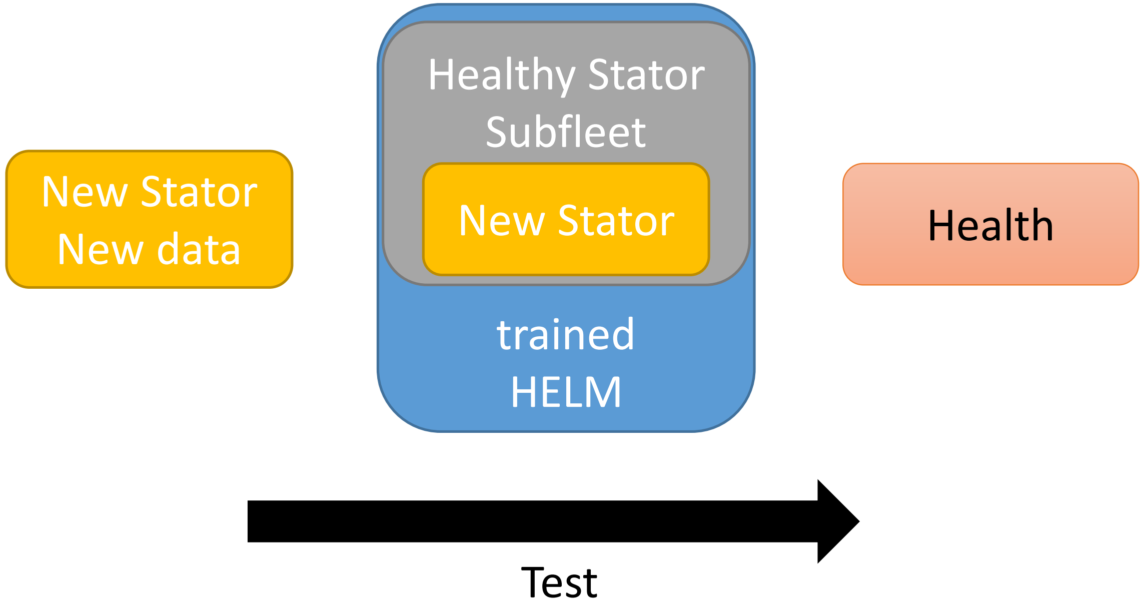
1 – Train HELM on “new” stator subfleet



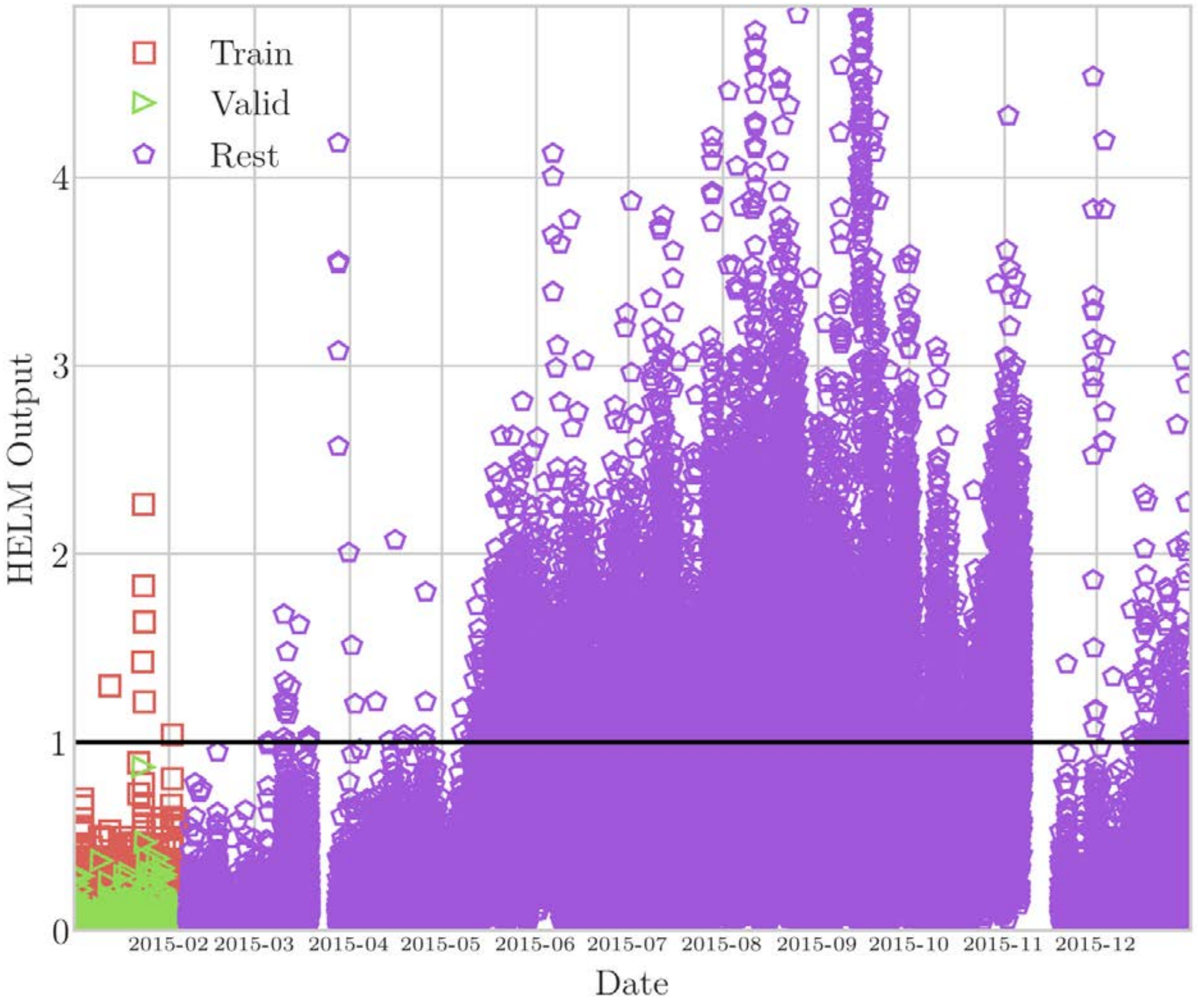
Individual Asset Monitoring

1 – Train HELM on “new” stator subfleet

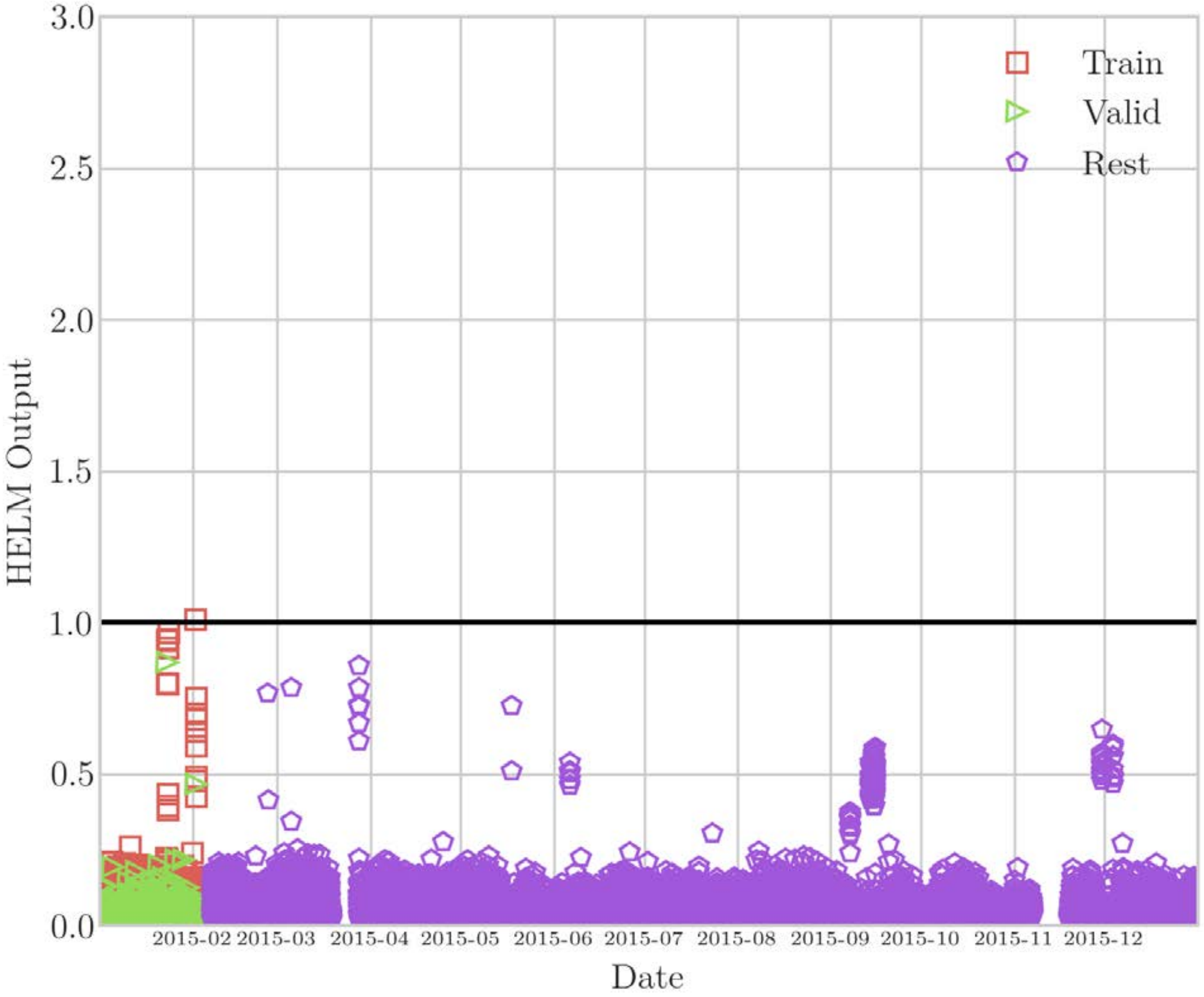
2 – Monitor new data from individual asset



Examples

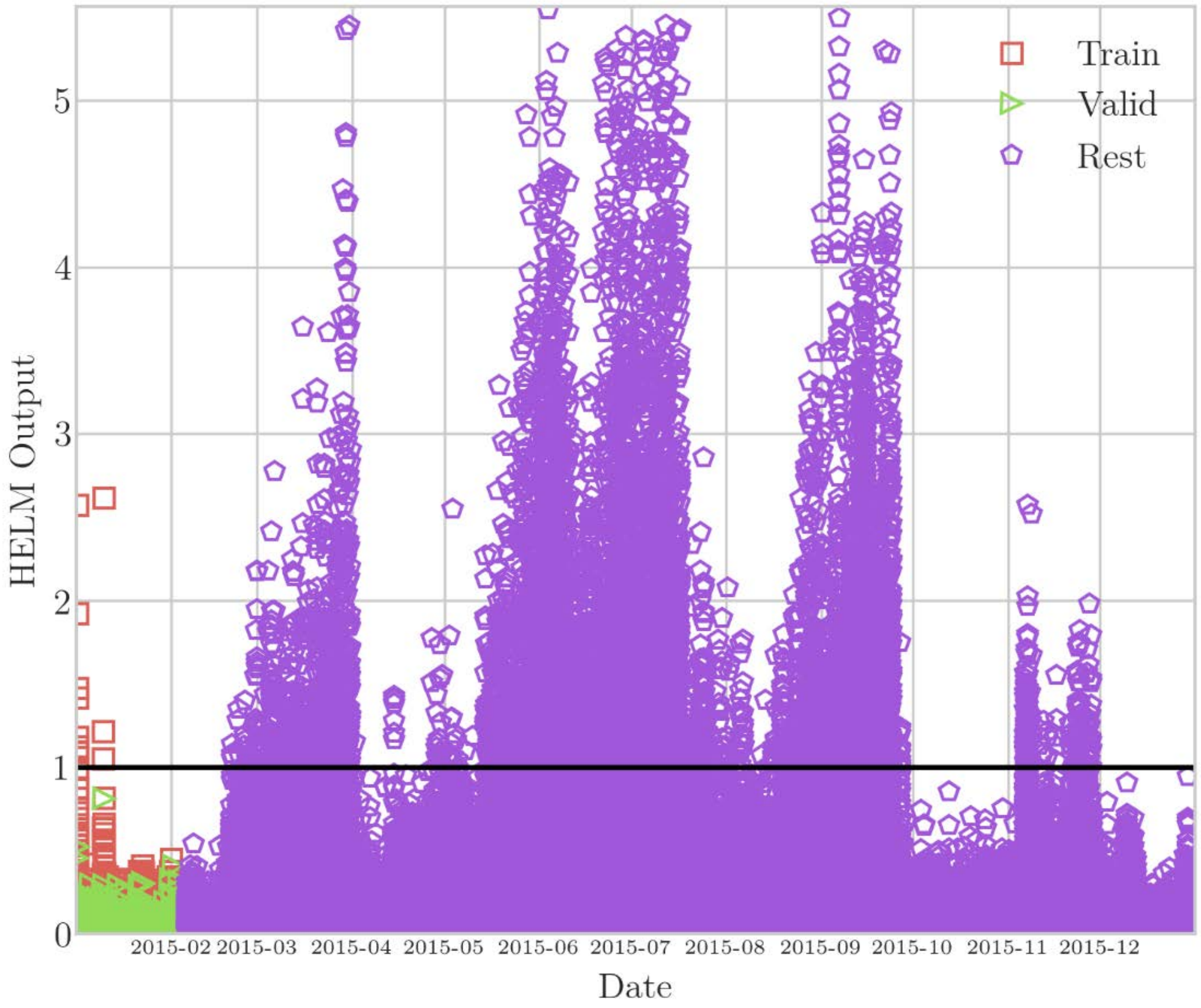


Train
Valid
Rest
HELM Output
0
1
2
3
4
2015-02 2015-03 2015-04 2015-05 2015-06 2015-07 2015-08 2015-09 2015-10 2015-11 2015-12
Date

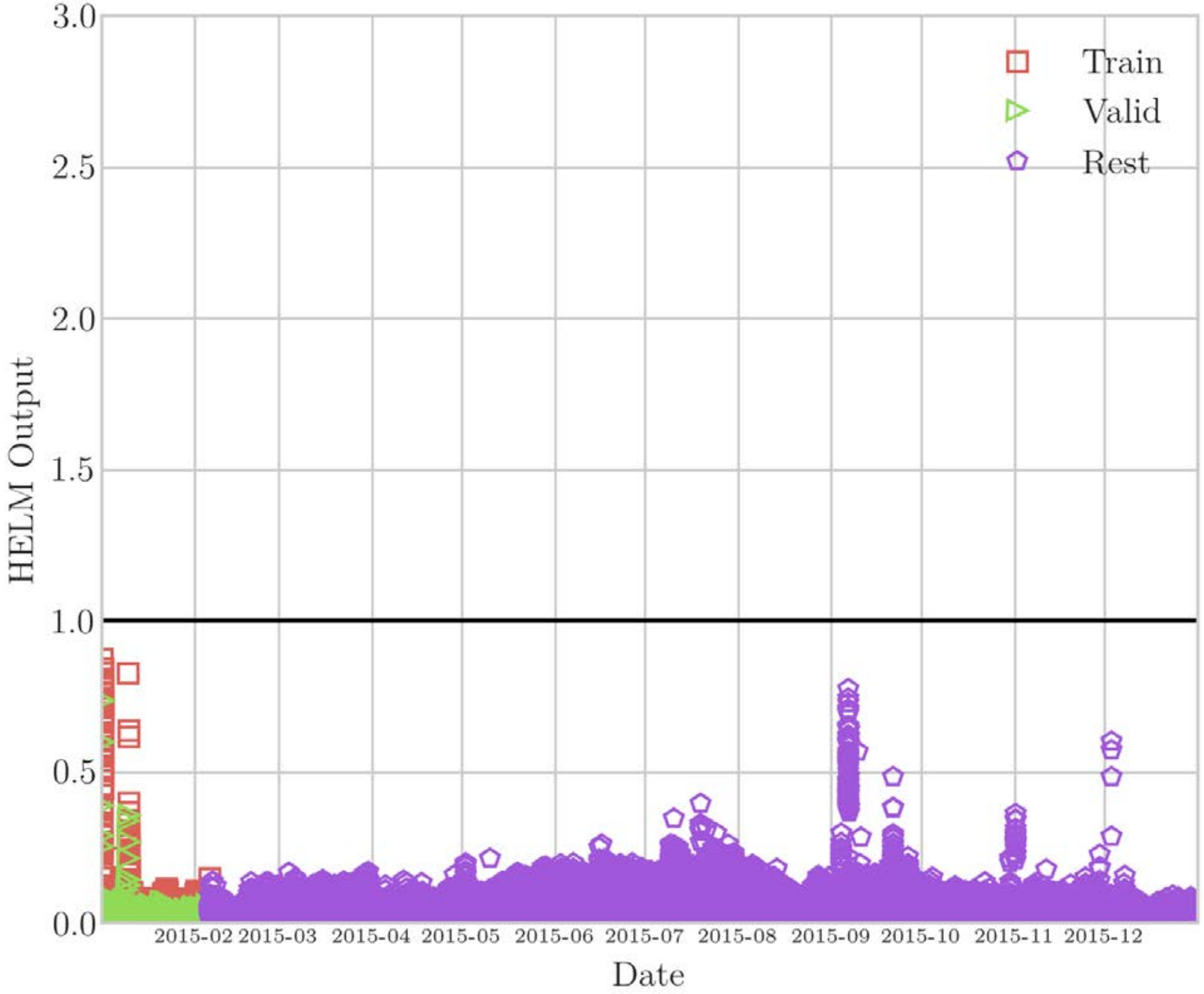


Train
Valid
Rest
HELM Output
0.0
0.5
1.0
1.5
2.0
2.5
3.0
2015-02 2015-03 2015-04 2015-05 2015-06 2015-07 2015-08 2015-09 2015-10 2015-11 2015-12
Date

Examples

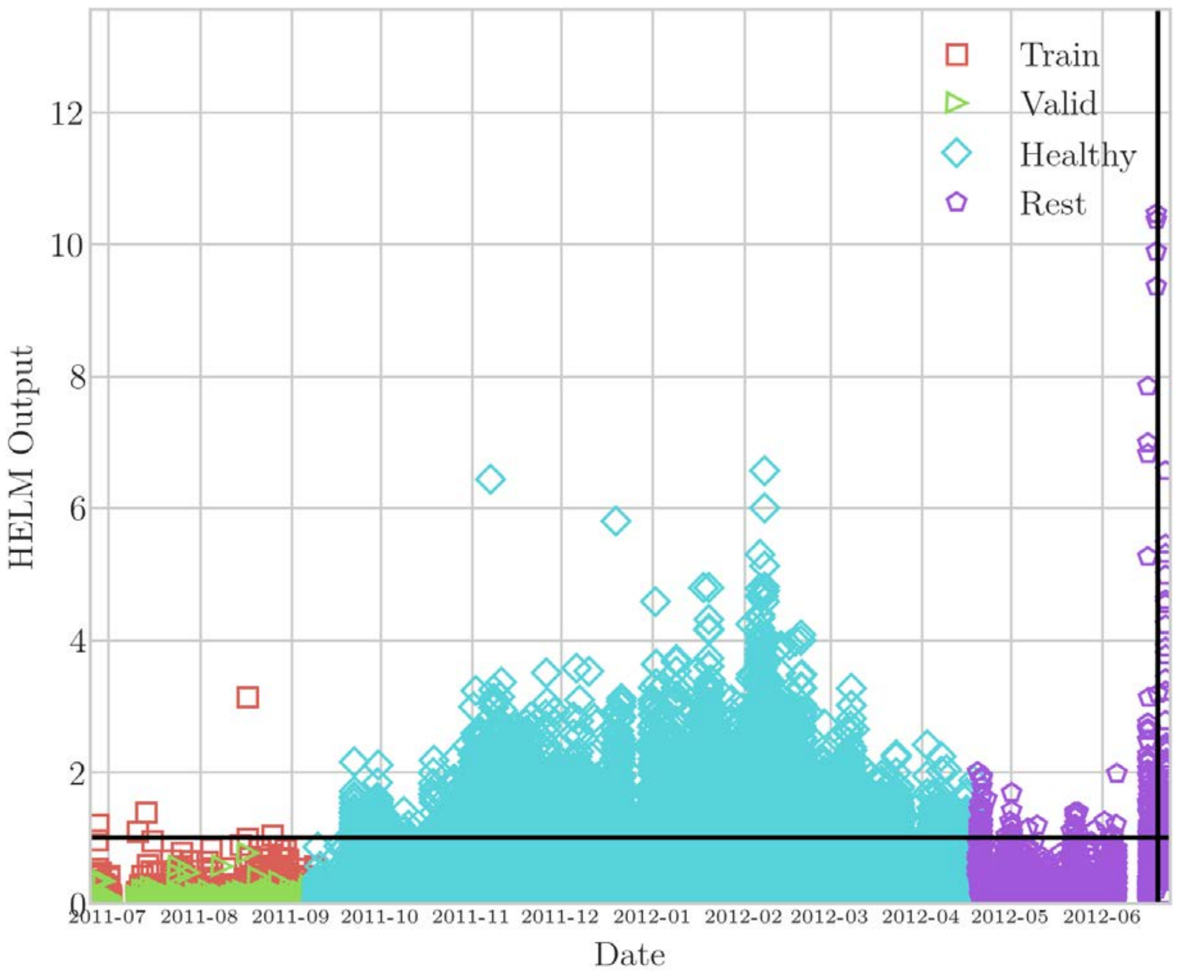


Train
Valid
Rest
HELM Output
Date
0
1
2
3
4
5
2015-02 2015-03 2015-04 2015-05 2015-06 2015-07 2015-08 2015-09 2015-10 2015-11 2015-12

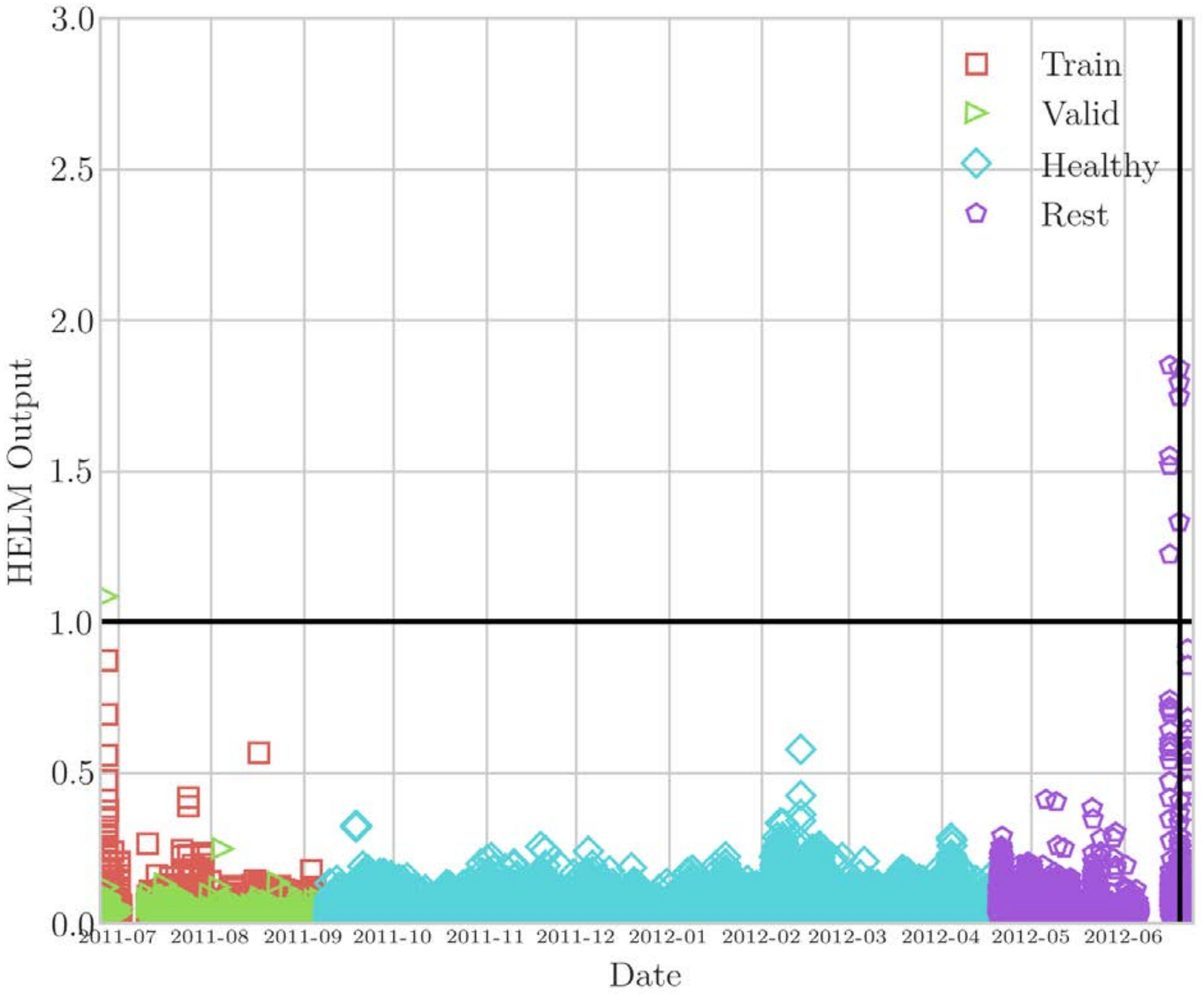


Train
Valid
Rest
HELM Output
Date
0.0
0.5
1.0
1.5
2.0
2.5
3.0
2015-02 2015-03 2015-04 2015-05 2015-06 2015-07 2015-08 2015-09 2015-10 2015-11 2015-12

Examples

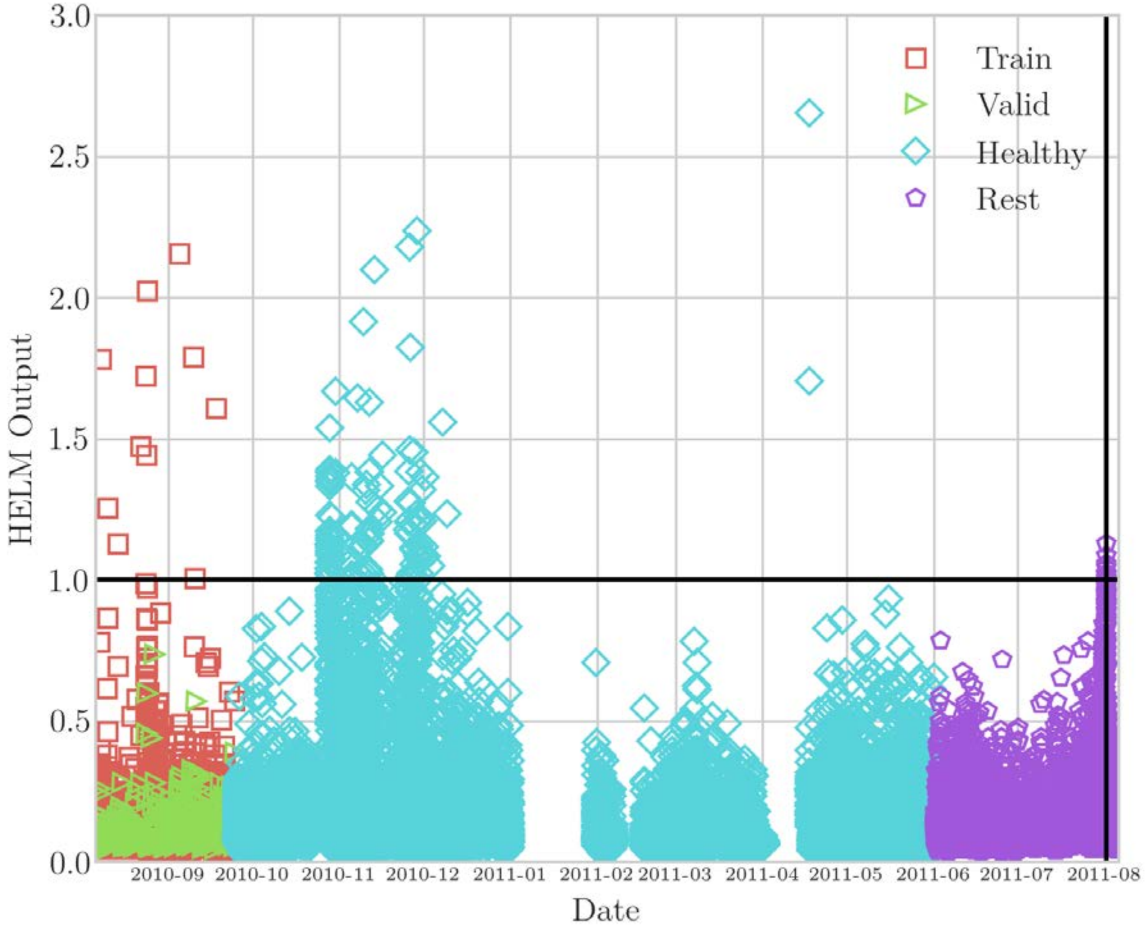


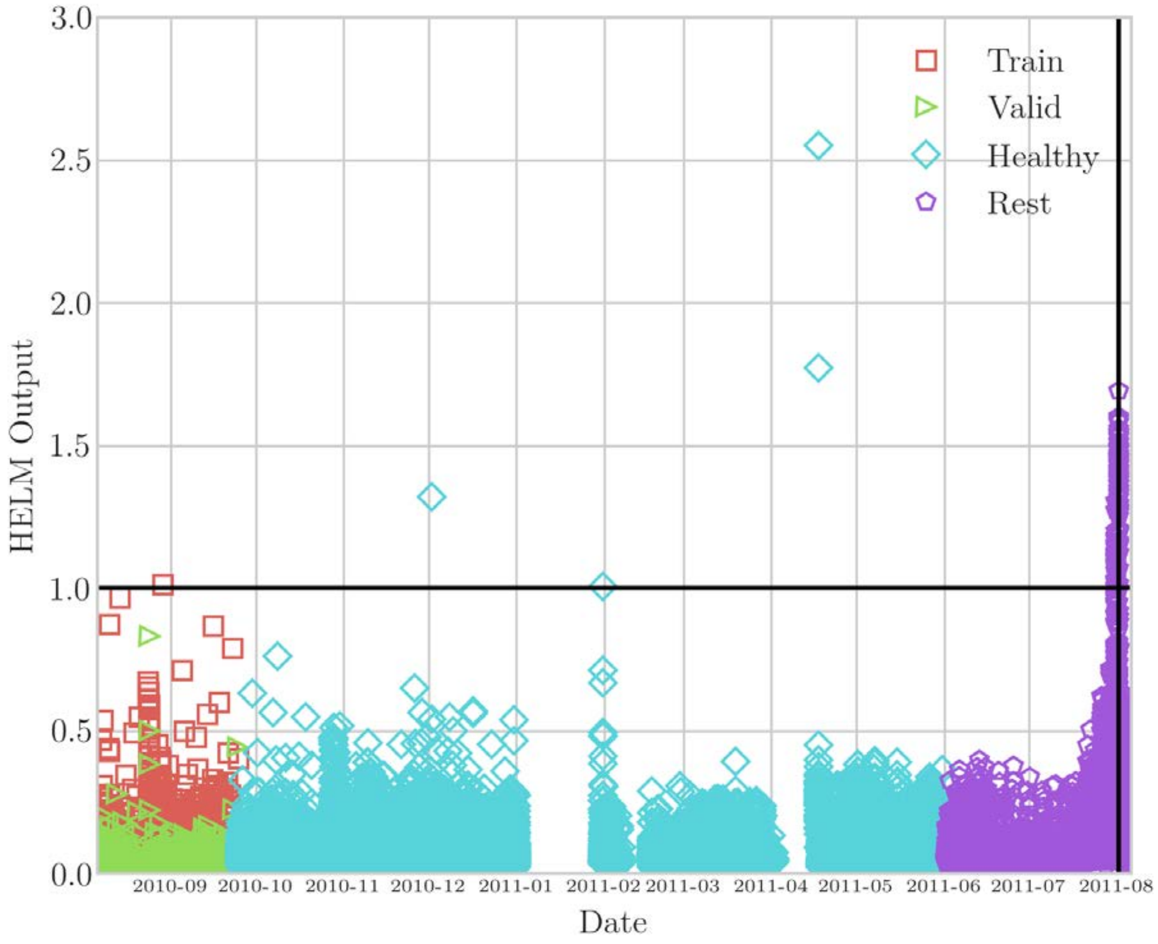
Train
Valid
Healthy
Rest
HELM Output
Date
0
2
4
6
8
10
12
2011-07
2011-08
2011-09
2011-10
2011-11
2011-12
2012-01
2012-02
2012-03
2012-04
2012-05
2012-06



Train
Valid
Healthy
Rest
HELM Output
Date
0.0
0.5
1.0
1.5
2.0
2.5
3.0
2011-07
2011-08
2011-09
2011-10
2011-11
2011-12
2012-01
2012-02
2012-03
2012-04
2012-05
2012-06

Examples


Train
Valid
Healthy
Rest
HELM Output
Date
3.0
2.5
2.0
1.5
1.0
0.5
0.0
2010-09
2010-10
2010-11
2010-12
2011-01
2011-02
2011-03
2011-04
2011-05
2011-06
2011-07
2011-08


Train
Valid
Healthy
Rest
HELM Output
Date
3.0
2.5
2.0
1.5
1.0
0.5
0.0
2010-09
2010-10
2010-11
2010-12
2011-01
2011-02
2011-03
2011-04
2011-05
2011-06
2011-07
2011-08

What is coming next?

- **Fleet Feature analysis**: What can be learned on the fleet?

- **Transfer learning**: What if my fleet have several versions of my system? (eg., not exactly the same data recorded)

Thanks for your attention

Acknowledgements

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Commission for Technology and Innovation CTI