PULSE CORE

DATA-DRIVEN SMART BUILDINGS

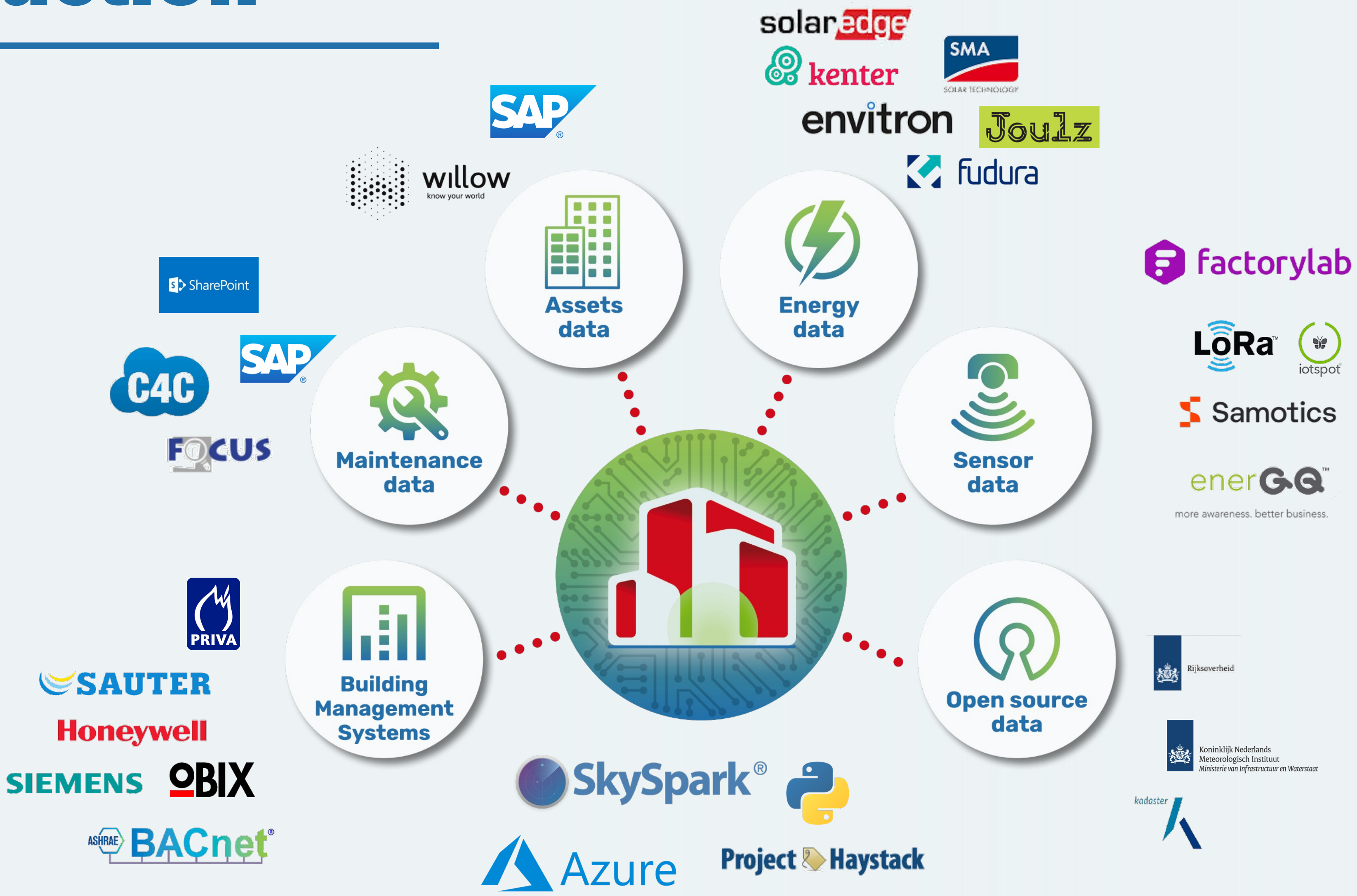
FDD IN THE PULSE CORE PLATFORM

B4B/IEA ANNEX 81 ONLINE SYMPOSIUM 28-02-2024
COEN HOOGERVORST

PULSE Core is powered by SPIE Nederland B.V.

PULSE CORE

Introduction



Introduction

BUILDING BLOCKS

SERVICES

Connectivity

- BMS portal
- Data collection
- Alert processing
- Operator services

Asset data management

- Document management
- Maintenance schedules
- Ticketing
- Quotations
- Building & Asset centre

Analytics

- Summit
- Smart
- Start

Narrowcasting

Introduction

>750	Buildings connected
>4,6 mill.	m^2 managed and analyzed
278.448	Ton CO_2
400.000	BMS notifications / year
>250	Strategic customers
>150.000	Sensors
>62 mld.	measured values
>4000	Users

Why FDD in the PULSE Core platform

CURRENT SITUATION:

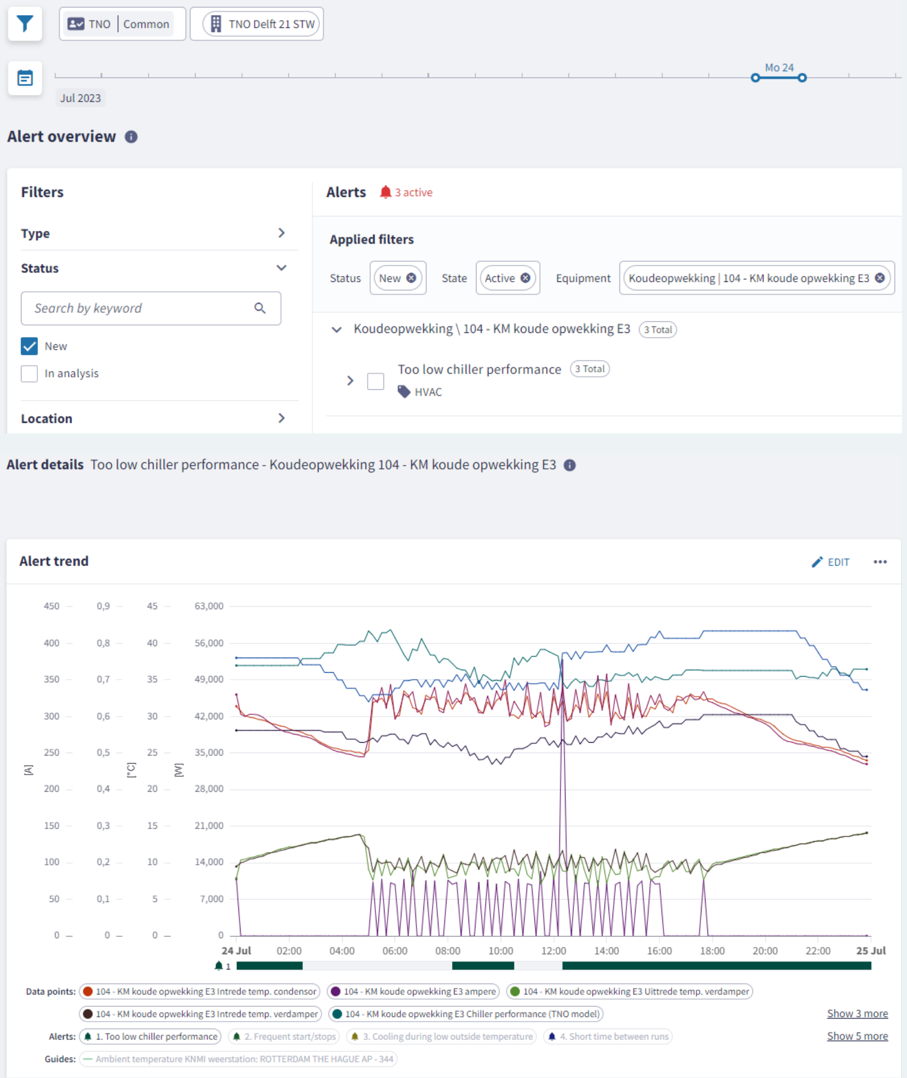
- A large librar… …hms (rule & model based) applied on la…
- Results not … …irectly in operation:
    - Person with analytics skills and domain knowledge is needed to do root cause analysis

These persons are hard to find!

DESIRED SITUATION:

- Create an automated solution which will execute the root cause analysis so a diagnose can directly be shared with people in the operation

>> Fault Detection **and Diagnosis**



Research question

Main question:

Are diagnostic bayesion networks (DBN) a suitable methodology to extend our automated fault detection solution with diagnostics to replace (a part of) our manual analyses?

Requirements and preconditions:

- Low implementation time per project (requirement for positive business case).
- The methodology should be scalable and easy to apply to a wide variety of projects with specific characteristics.
- We do not have labelled datasets.
- For good adoption in operation reliability is key!

Research steps

1. Build automated data flow from sensor data to fault probabilities in our platform for an air handling unit (AHU).

2. Demonstrate the reliability and usability of the DBN methodology on AHU's

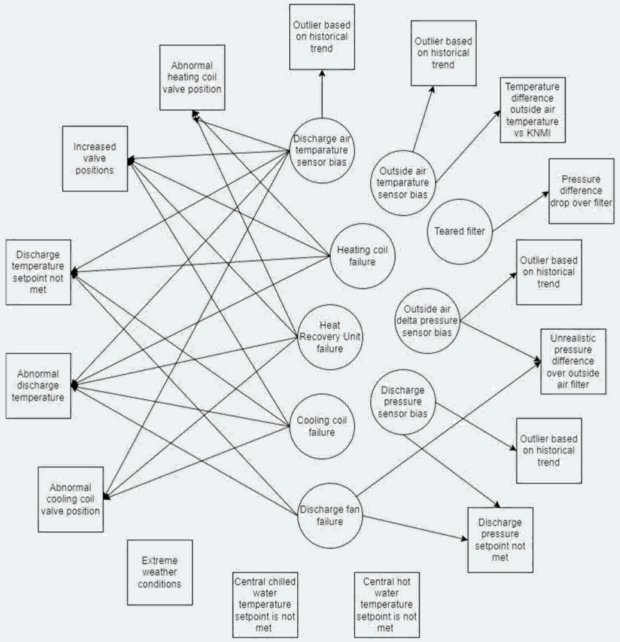
2a. Optional: testing applicability of small sub-component models trade-off between different model results.

3. Define a conclusion on applicability of DBN for FDD

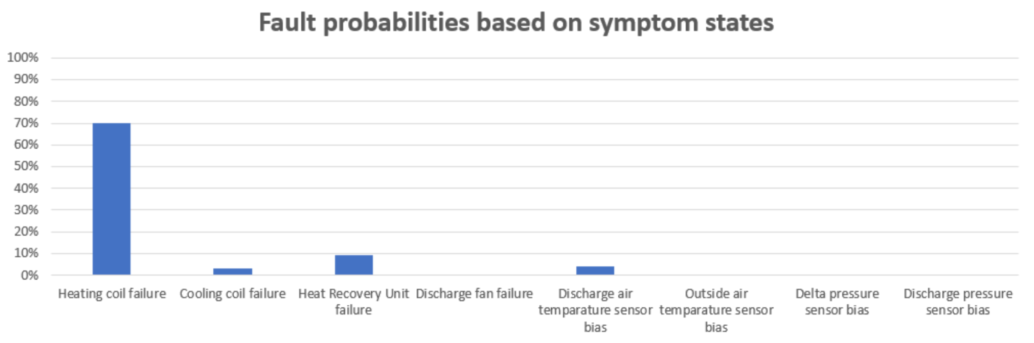
4. Optional: create DBN's for other equipment types

From sensor data to fault probabilities

- A DBN for an AHU on main component level was defined (using Noisy-max).
- This was the first experience in estimating prior and conditional probabilities.



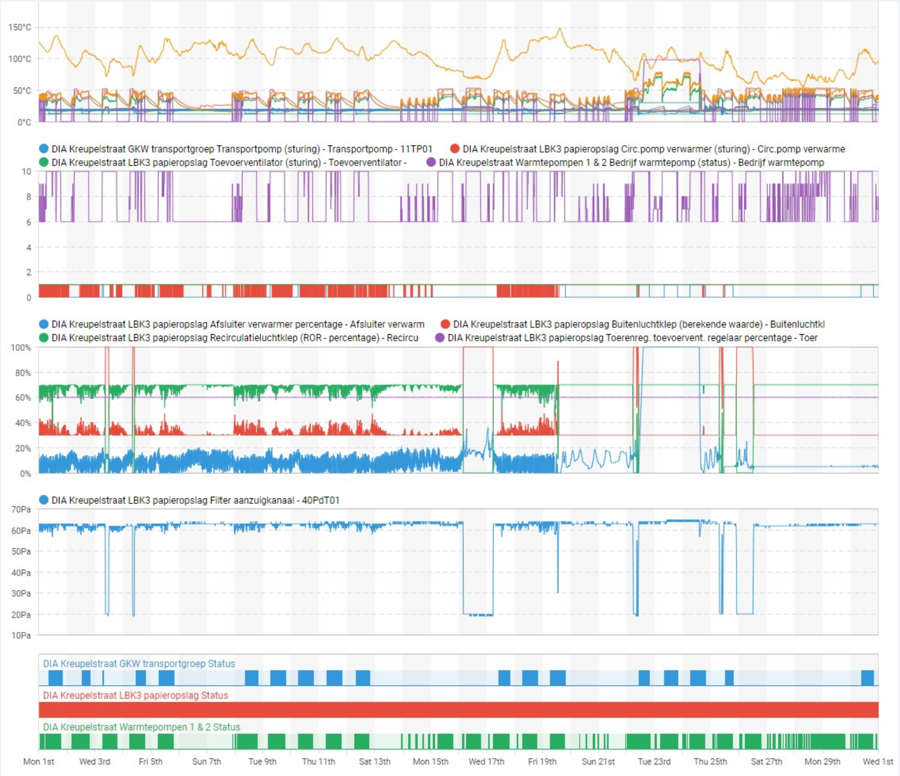
	Symptom state	Heating coil failure	Cooling coil failure	Heat Recovery Unit failure	Discharge fan failure	Discharge air temparature sensor bias	Outside air temparature sensor bias	Delta pressure sensor bias	Discharge pressure sensor bias
		Present	Present	Present	Present	Present	Present	Present	Present
Abnormal heating coil valve position during heat demand	Present								
Abnormal cooling coil valve position during cool demand	Absent								
increased valve positions	Present								
Abnormal discharge temperature	Present								
Discharge temperature not met	Present								
Discharge pressure not met	Absent								
Central hot water temperature is not met	Absent								
Central chilled water temperature is not met	Absent								
Extreme weather conditions	Absent								
Outlier discharge temperature	Absent								
Outlier outside temperature	Absent								
Outlier delta pressure filter	Absent								
Outlier discharge pressure	Absent								
Unrealistic delta pressure filter	Absent								
Big temperature difference between outside air temperature and KNMI	Absent								
FAULT PROBABILITIES		**70%**	**3%**	**9%**	**0%**	**4%**	**0%**	**0%**	**0%**

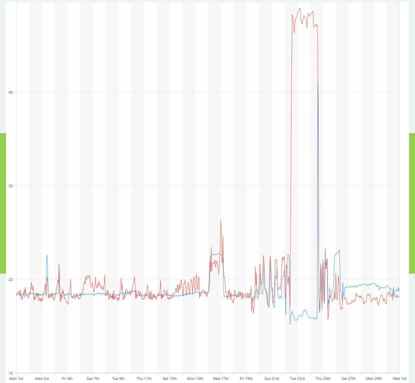


FDD IN THE PULSE CORE PLATFORM

From sensor data to fault probabilities

- Logic for symptoms was created in Skyspark
- Linear regression models for anomaly detection instead of ML models (with not too bad performance)

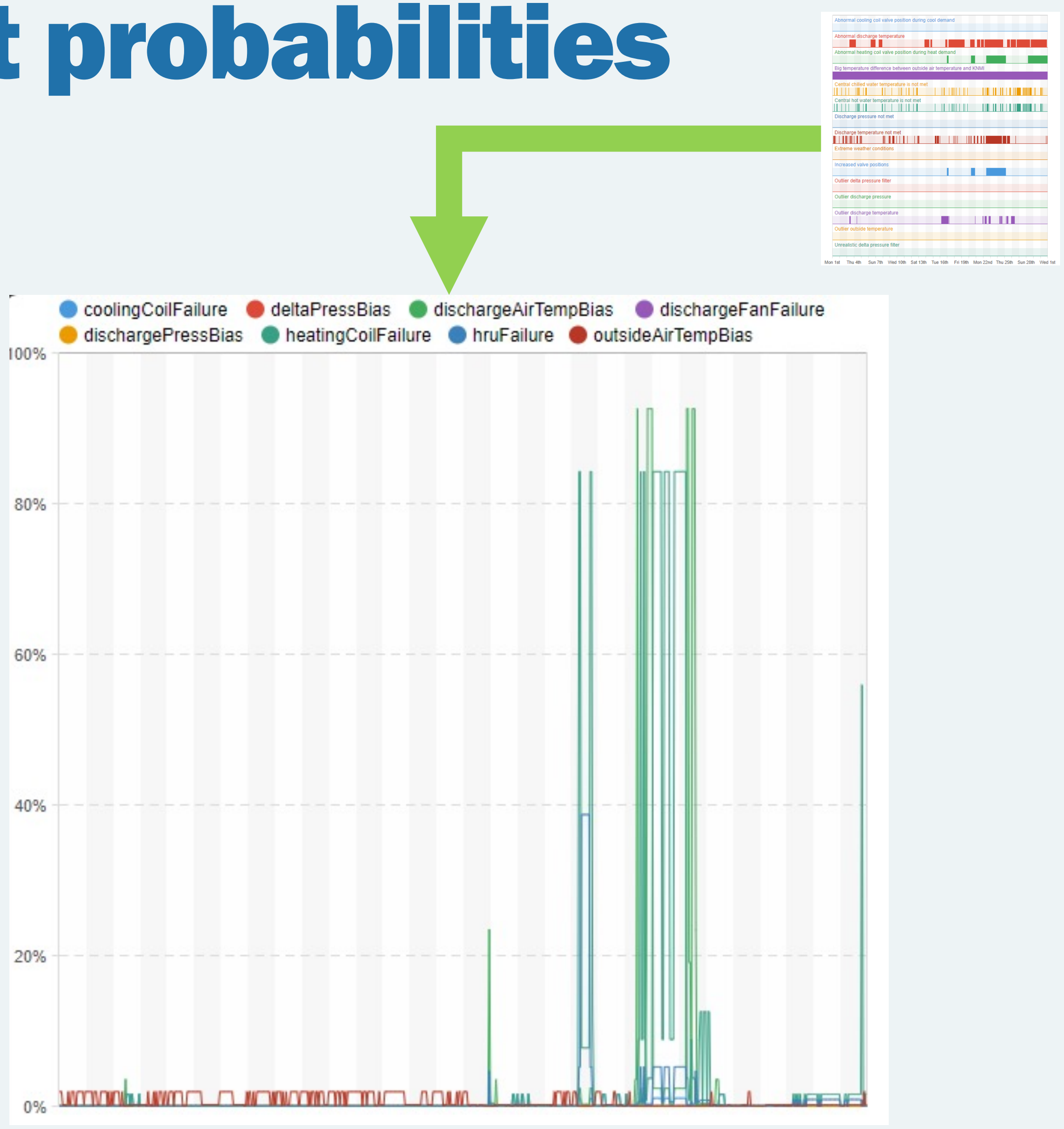




From sensor data to fault probabilities

- Created general logic to calculate fault probabilities based on symptom states.
    - simplified probability calculation (which approaches real methodology)

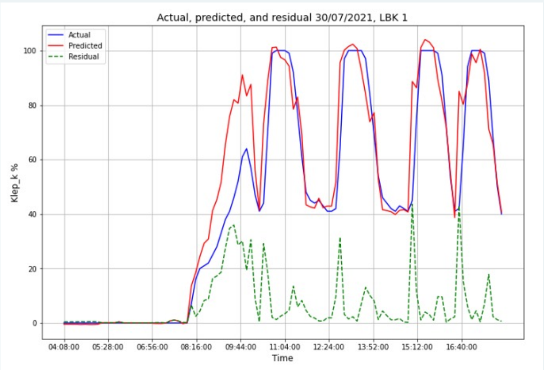
- Current work: apply this logic on multiple buildings to gain experience on scalability and identify issues:
    - How handle different configurations?
    - Is the (regression / ML) model approach reusable?
    - Is the DBN structure with probabilities universal applicable?

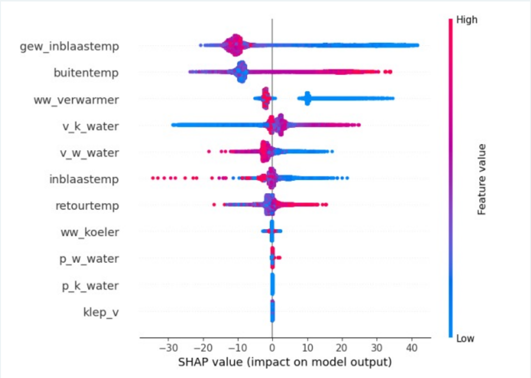


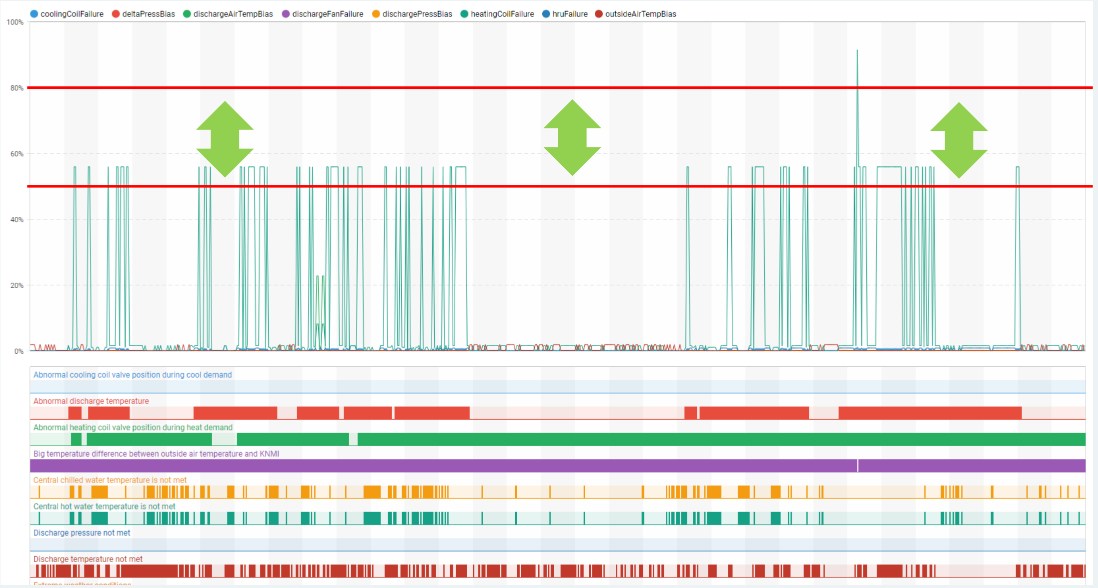
Reliability and scalability of the DBN

- Create an improved DBN:
  - Domain expert interviews in corporation with TU/e for improved DBN
  - Tune the DBN and symptom definitions based on first results in step 1
  - Replace regression models by ML models (if needed)

- Evaluate results of the DBN on multiple buildings with known failures:
  - Based on first threshold evaluate periods with detected faults by comparing them with failure database.
  - Define probability threshold for alerts



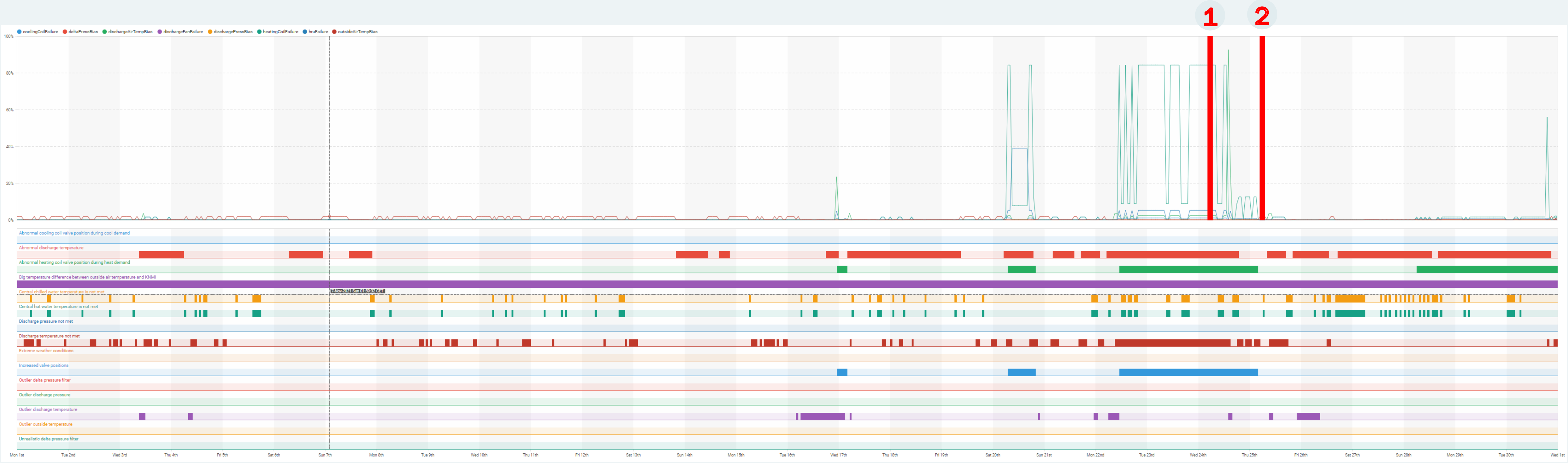




Findings so far

Fault probabilities and symptom states for 1 month:

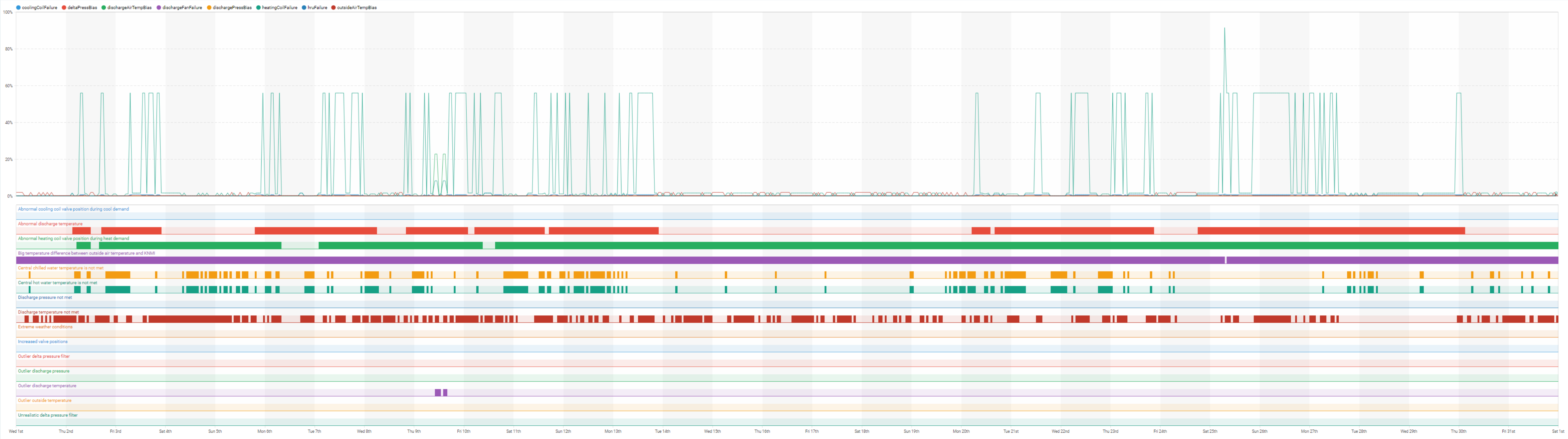
1. Reported failure: stuck heating coil valve and broken valve actuator, temporary fix
2. Replaced valve actuator.



Findings so far

Fault probabilities and symptom states for 1 month after previous failure

1. Fault probabilities at 56% for large period caused by symptoms: abnormal heating coil valve position, discharge temperature not met, abnormal discharge temperature.
2. After fix of failure the prediction models are not reliable anymore: retraining of models is needed?!



Findings so far

DBN definition is complex

- Specific symptom definition is key to make distinction between failures.
- Probabilities estimation (both prior and conditional) is new for domain experts and too difficult. Solution for this:
  - failures are ranked in order of occurrence and then translated to prior probabilities
  - Symptom-failure relations are defined using classes (no, some, strong relation) and then translated to conditional probabilities.
  - Tool to evaluate impact by evaluating different symptom states combination helps to get a better feeling.
- The definition phase is time expensive: this can only be justified if the methodology is reliable and generically applicable.

Generic approach creates new challenges:

- Define minimum set of symptoms of which the state can be determined.
- Create a set of optional symptoms:
  - Should probabilities be adjusted based on the available symptoms?
- ML models for anomaly detection:
  - Automated cleaning of data and feature selection based on available data points and haystack data model

Findings so far

Reason why we are continuing this work is:

- A DBN approach makes it easier to do an **integral** evaluation a large set of possible diagnosis. This is valuable compared to (sub)component level rule-based analysis.
  - Current situation: when central hot water generation has a failure multiple faults are detected. Not only on the hot water generation but also on equipments using the hot water for heating.
  - Desired situation: the central hot water fault is diagnosed and all other equipments do not get diagnosed with a failure.
- The DBN approach is easy to expand with new symptoms and diagnoses because (using the noisy max approach) it is just a 2 dimensional matrix. This which makes it future proof and easy expandable

Planned work

1. Finish step 1 by evaluation the results of the DBN on multiple AHU's in our platform with known failures.
2. Create an improved DBN and integrate in the current logic for a 2nd iteration where we will apply the DBN on a larger scale to:
   1. Tune probabilities and symptom definitions
   2. Define failure probability thresholds for alerting.
3. Draw conclusion on the usability of the methodology

If the conclusion is that the DBN approach is a good approach, we will start working on creating DBN for other type of equipments.

Questions?

Are the any questions!