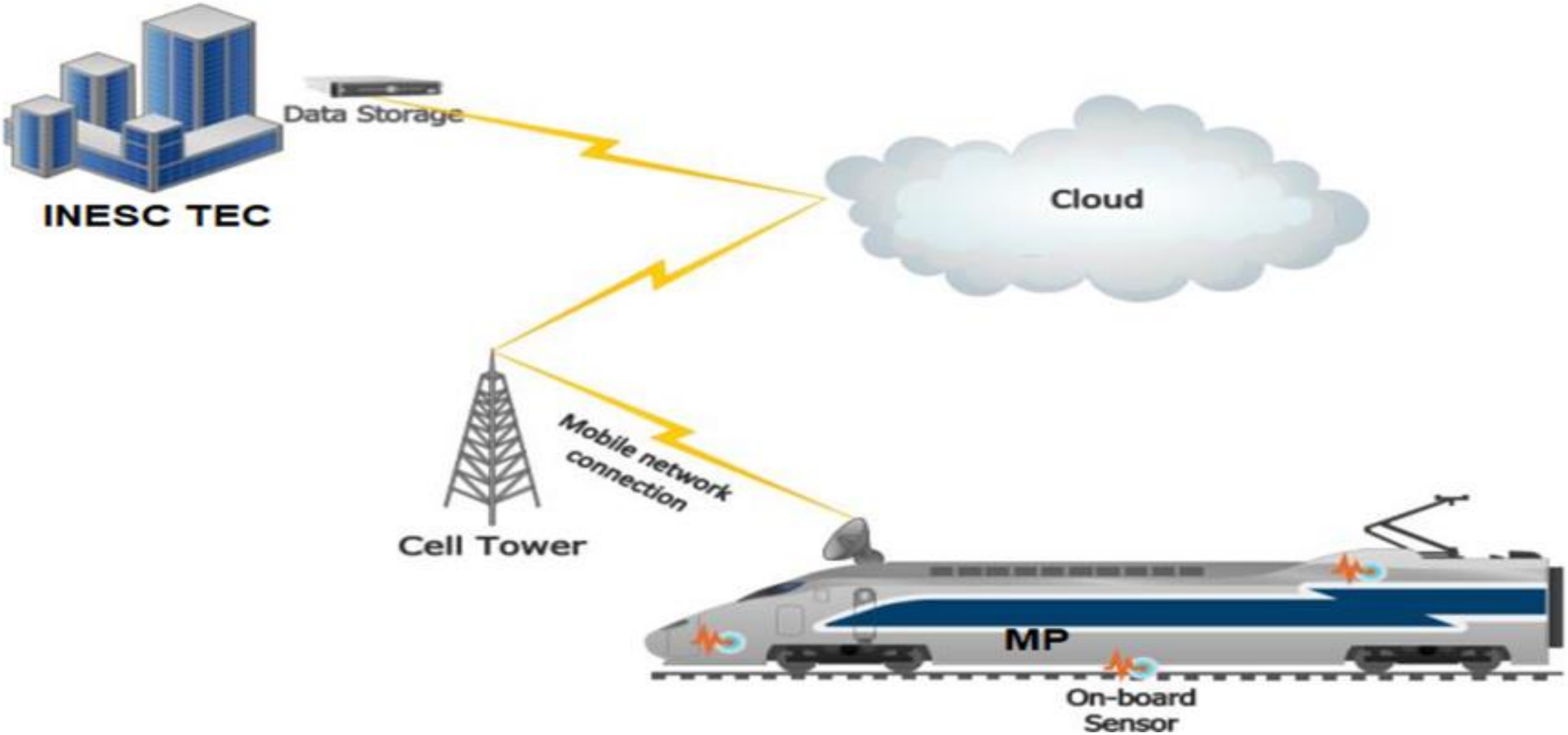
A Neuro-Symbolic Explainer for Rare Events

João Gama, Rita P. Ribeiro, Bruno Veloso

University of Porto & INESC TEC
Portugal

KDD2024
BARCELONA, SPAIN

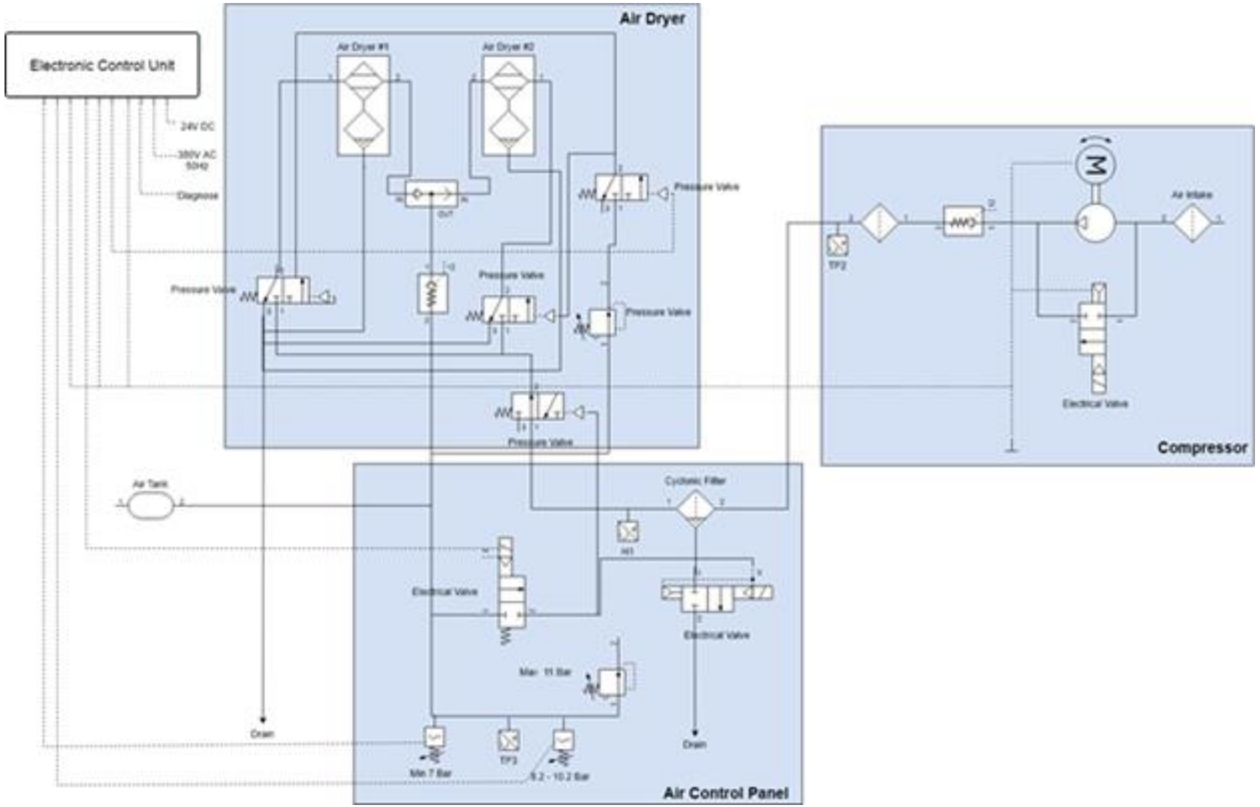
Real Time Failure Detection: The Context



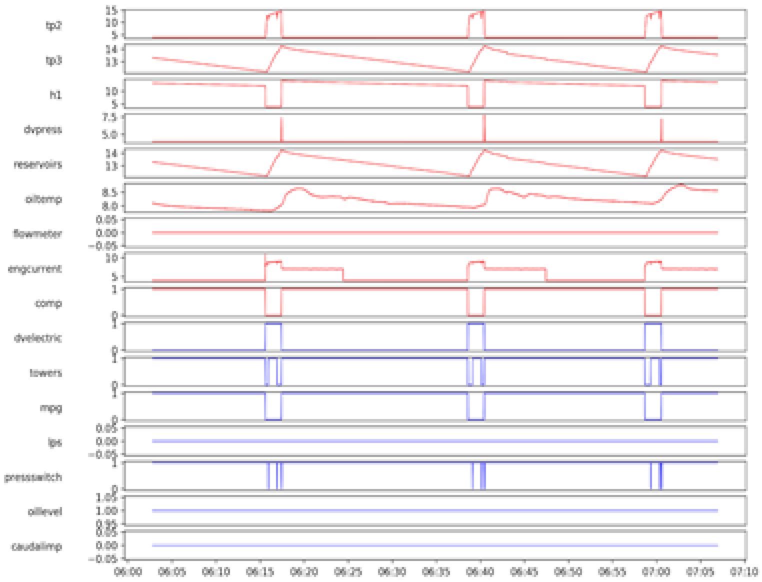
Context

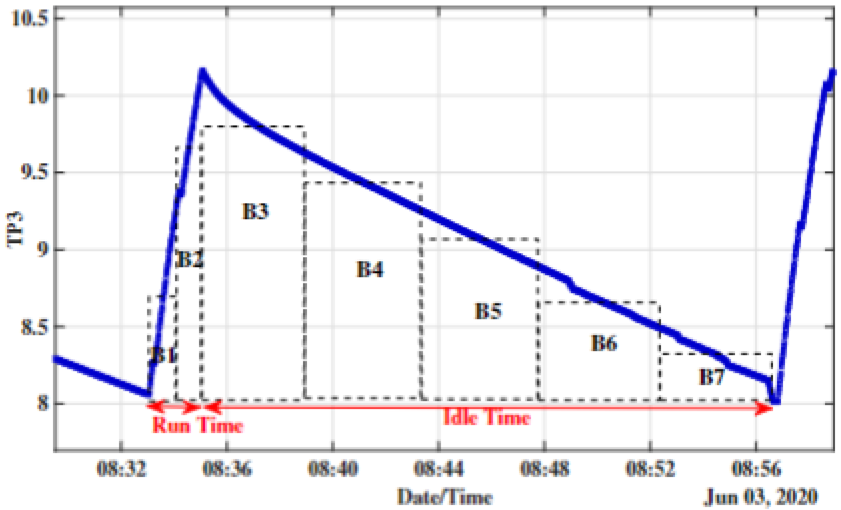
- High-speed Sensor Streaming Data
- The interesting cases are the rare events:
  - Changes in the working regimes
  - Anomalies and Failures
- Explaining the rare events, mainly failures!

The Air Compressor Unity



The Air Compressor Unity Sensors





Analogue Sensors

Table 1: Onboard sensors from APU train [19].

nr.	Module	Description
Analogue		
1	Compressor	TP2 - Compressor Pressure
2	Air Control Panel	TP3 - Pneumatic panel Pressure
3	Air Control Panel	H1 - Pressure above 10.2 Bar
4	Air Dryer	DV - Air Dryer Tower Pressure
5	Air Control Panel	Reservoirs - Pressure
6	Compressor	Oil Temperature
7	Air Control Panel	Flow meter
8	Compressor	Motor Current

Digital Sensors

Digital		
9	Electronic Control Unit	COMP - Compressor on/off
10	Electronic Control Unit	DV electric - Compressor outlet valve
11	Electronic Control Unit	Towers - Active tower number
12	Electronic Control Unit	MPG - Pressure below 8.2 Bar
13	Electronic Control Unit	LPS - Pressure is lower than 7 bars
14	Electronic Control Unit	Towers Pressure
15	Compressor	Oil Level - Level below min
16	Air Control Panel	Caudal impulses

Failure Detection

Autoencoders

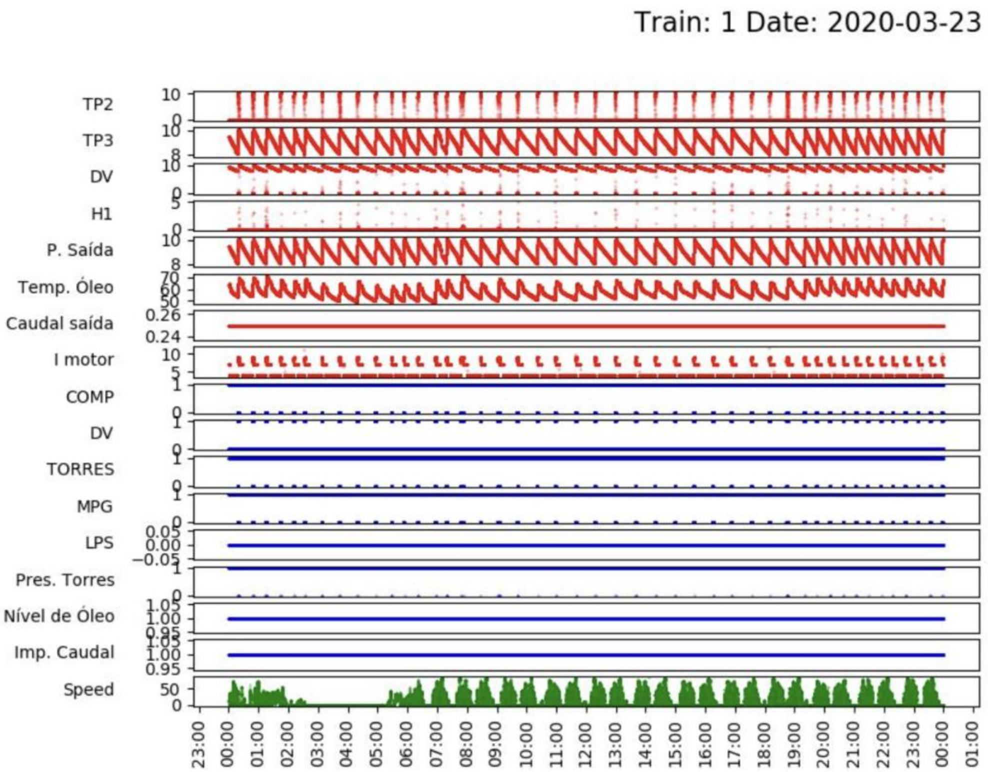
Learn an approximation of the identity function: $f(x) \approx x$.

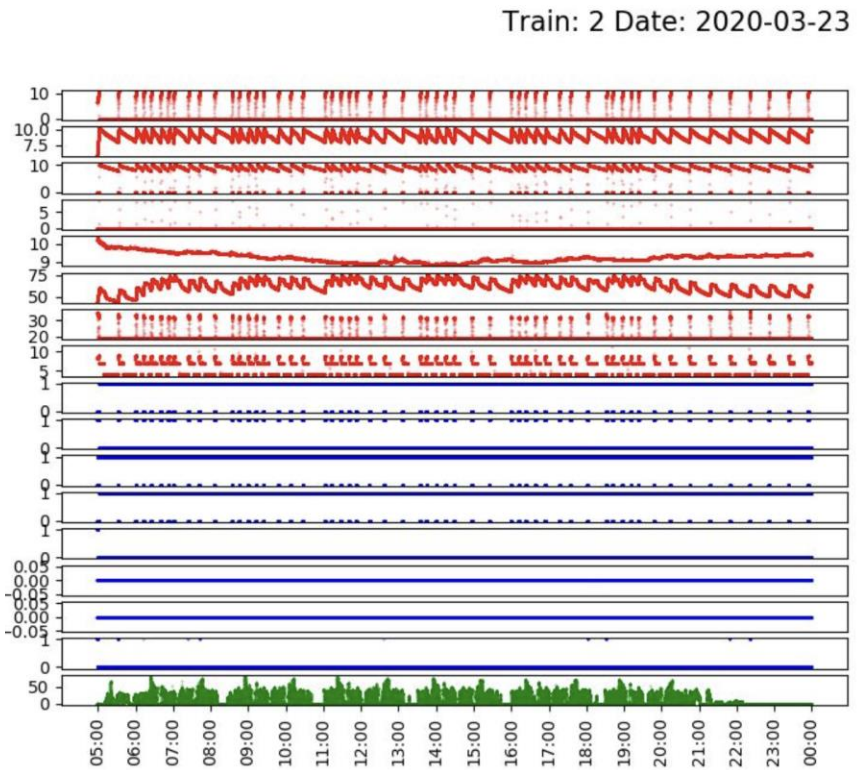
Two function: encoder $E_\phi : \mathcal{X} \to \mathcal{Z}$, decoder $G_\theta : \mathcal{Z} \to \mathcal{X}$,
where $\mathcal{X} = \mathbb{R}^n$ and $\mathcal{Z} = \mathbb{R}^m$.

Output of the encoder is written as: $E_\phi(x) = z$.
Name z as *latent vector* and $\mathcal{Z}$ as *latent space*.

Output of the decoder is written as: $G_\theta(z) = \hat{x}$.
$\hat{x}$ is the reconstruction of x.

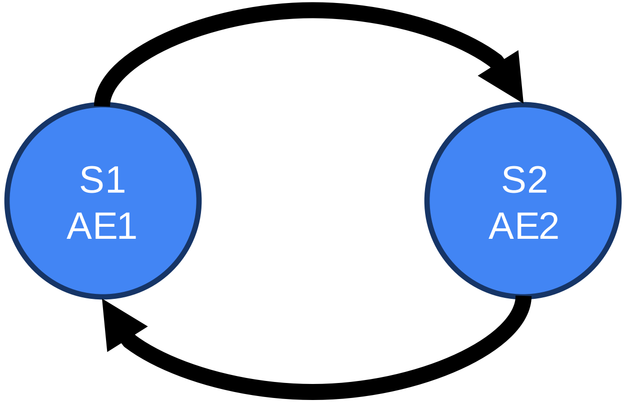
Normal Data


Train: 1 Date: 2020-03-23
TP2
TP3
DV
H1
P. Saída
Temp. Óleo
Caudal saída
I motor
COMP
DV
TORRES
MPG
LPS
Pres. Torres
Nível de Óleo
Imp. Caudal
Speed


Train: 2 Date: 2020-03-23

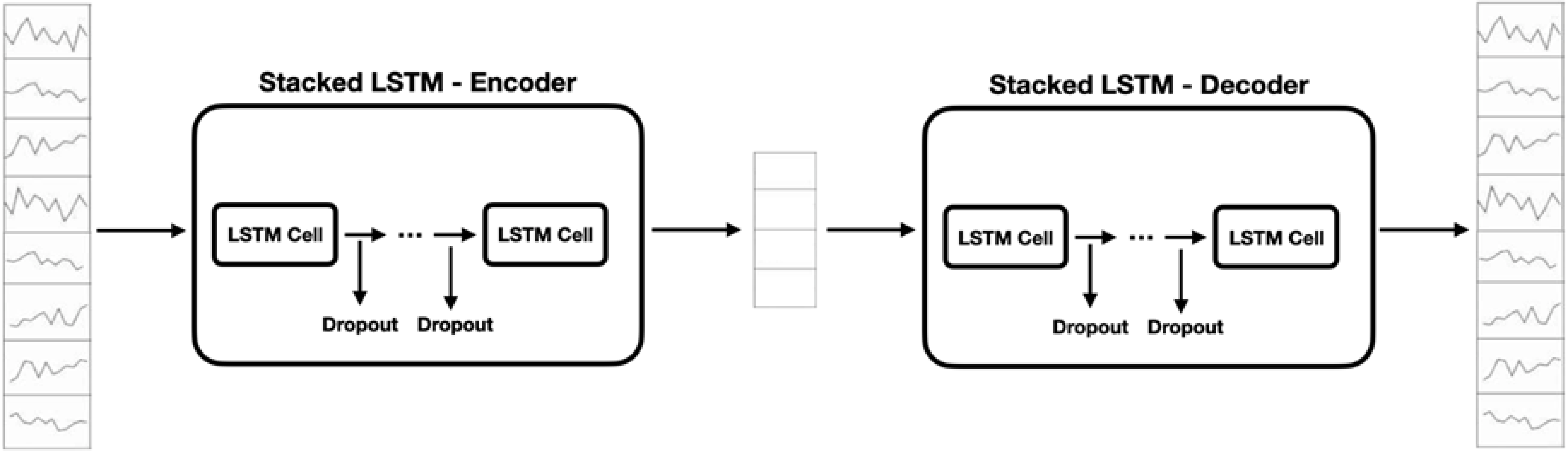
Autoencoders

- AE is trained with data from the normal behaviour of trains
- Different trains have different “normal” behaviours
- There are different normal working regimes for the same train
  - We use change detection algorithms to identify changes in the working regime
  - Each working regime has an AutoEncoer associated

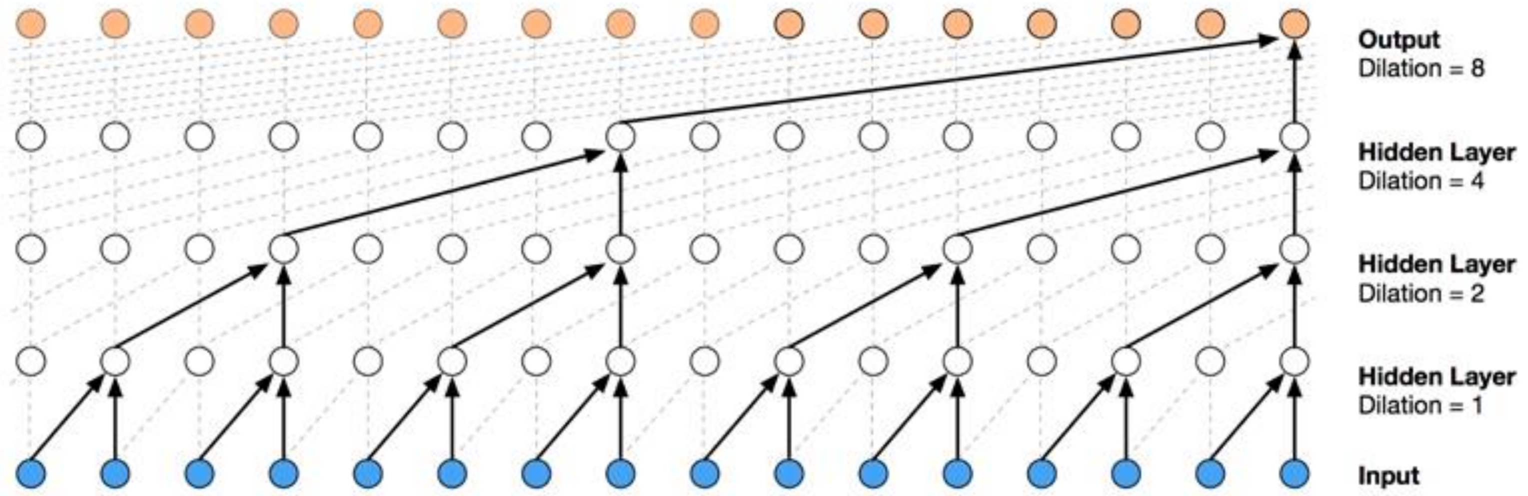


Chiara Balestra, Bin Li, and Emmanuel Muller.
Slidshaps–sliding shapley values for correlation-based change detection in time series.
In *2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2023.

Approach 1: LSTM Autoencoder



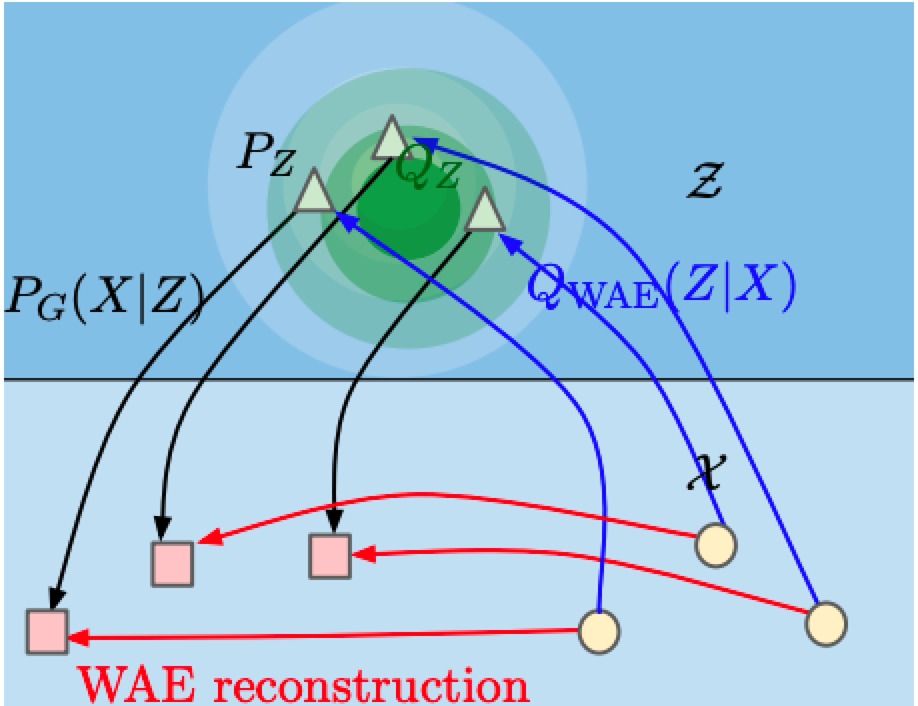
Approach 2: Temporal Convolutional Networks



Receptive field size = *dilation* x (*kernel* - 1) + 1

Exponential dilation: *dilation* = $O(2^{\text{layer}})$

Approach 3; Wasserstein Autoencoders with Generative Adversarial Networks (WAE-GAN)

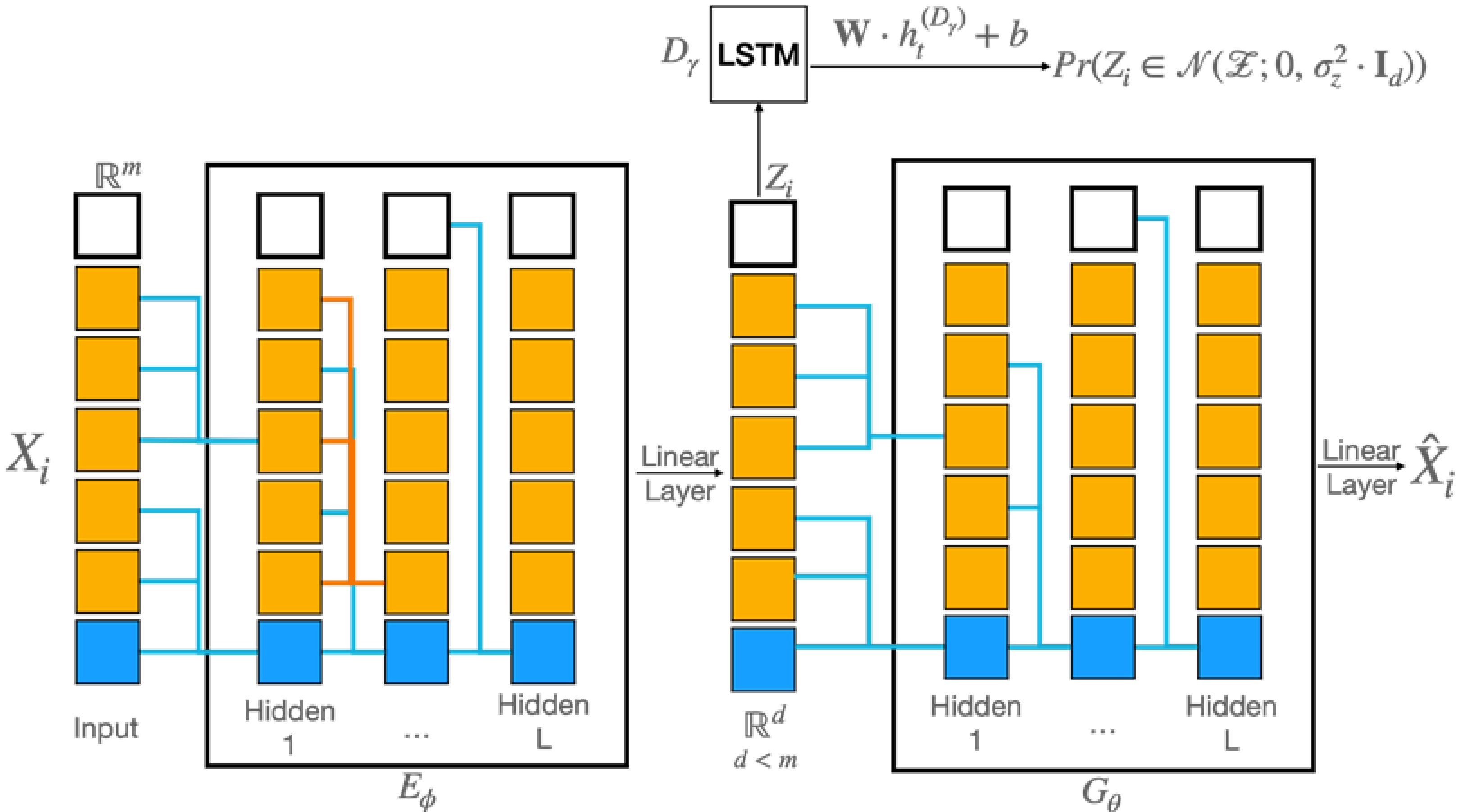


Regularization of latent space using an adversarial training scheme.

Discriminator network D_γ : trained to distinguish samples from E_ϕ from distribution P_Z

Minimax training scheme: E_ϕ is also trained to deceive D_γ

WAE-GAN with LSTM and TCN



Detecting failures

- Reconstruction error: $\frac{1}{n}\sum_{i}^{n}||x_i - G_\theta(E_\phi(x_i))||_2^2$
  - Using critic scores: compute Z-score as normalization - large absolute values of Z-score indicate high anomaly scores. Final output: multiply reconstruction error by z-scored critic score.

- Calculate anomaly threshold from distribution of outputs from the training set. Set the threshold to:

  Q3 +3*IQR

  - Values above anomaly threshold given value of 1: anomaly was detected.

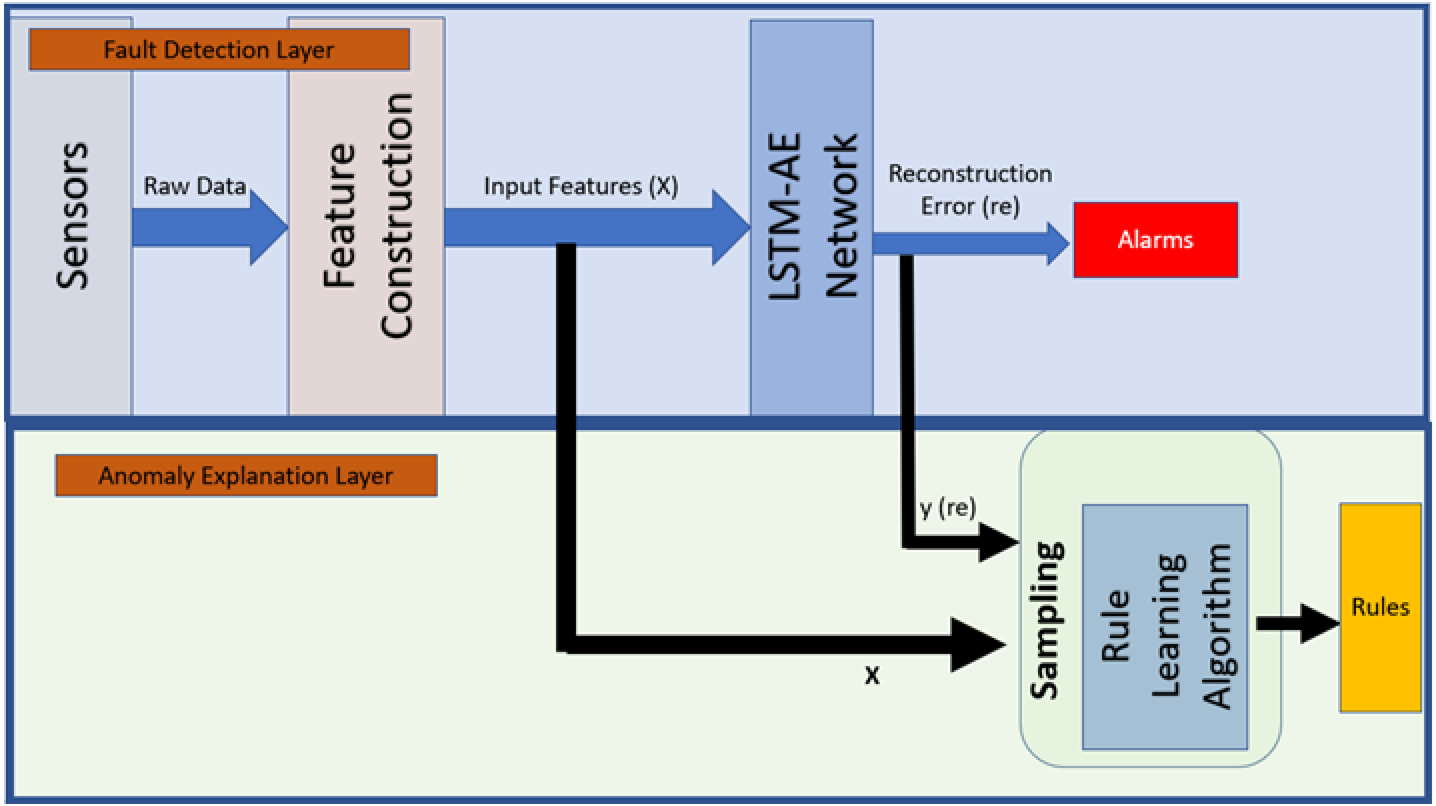
- Run a low pass filter on the resulting sequence of 1s and 0s:

$$y_i = y_{i-1} + \alpha * (x_i - y_{i-1}),\ y_0 = x_0$$

  - Output failure when the output of the low pass filter is consecutively above a decision threshold.

Explaining the Failure

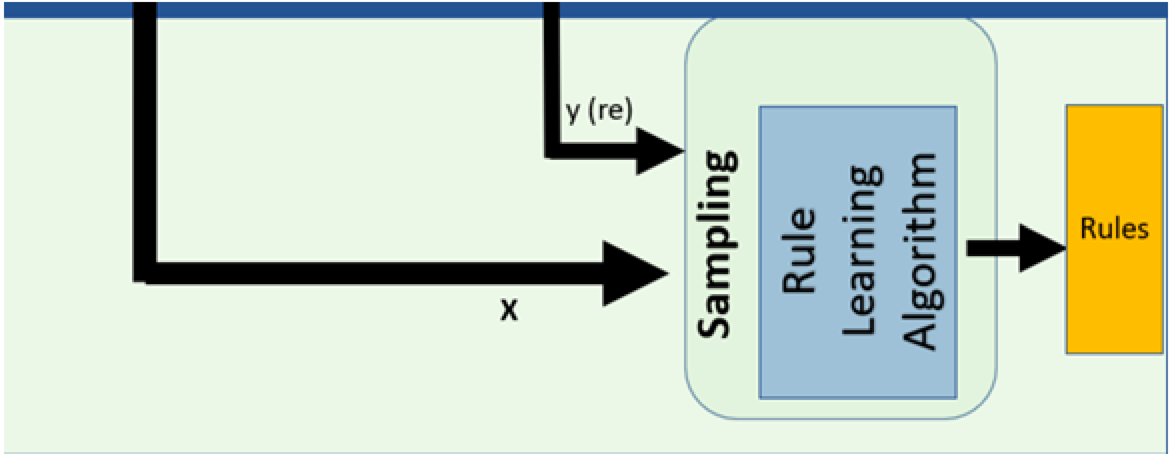
The Neuro-Symbolic Explainer for Rare Cases



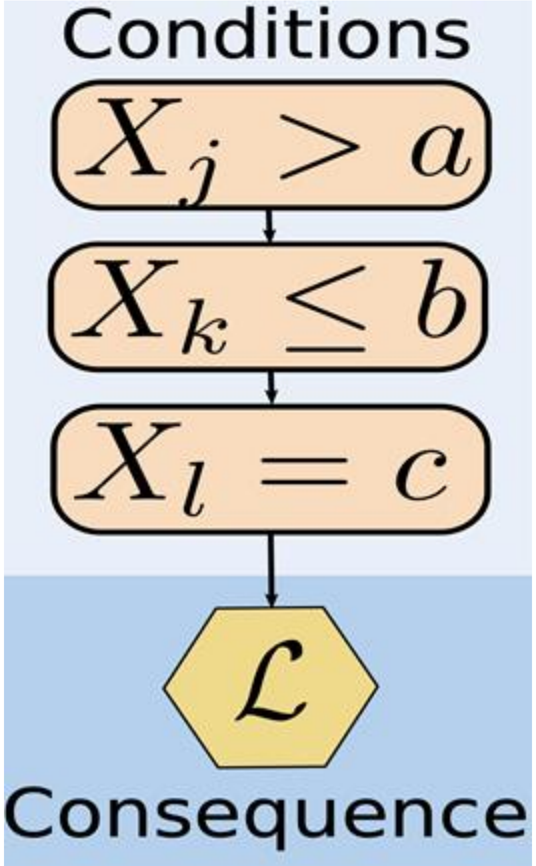
The Anomaly Explanation Layer

The two main components:

- An online regression rules learning system, based on AMRules. Learns a predictive model **y = f(X)**, where **y** is the reconstruction error, and **X** are the input features of the LSTM-AE.
- A sample strategy based on *Chebyshev inequality*: focusing on the examples with high reconstruction error, meaning high probability of being a failure.



Regression Rules



- A rule is an implication of the form
  Antecedent ⇒ Consequent
- The **Antecedent** is a conjunction of conditions based on attribute values.
- If all the conditions are true, a prediction is made based on **Consequent** (L).
- The **Consequent** contains the sufficient statistics to:
  - expand a rule,
  - make predictions,
  - detect changes,

J. Duarte, J. Gama, A. Bifet: Adaptive Model Rules From High-Speed Data Streams. ACM Trans. Knowl. Discov. Data; 2016

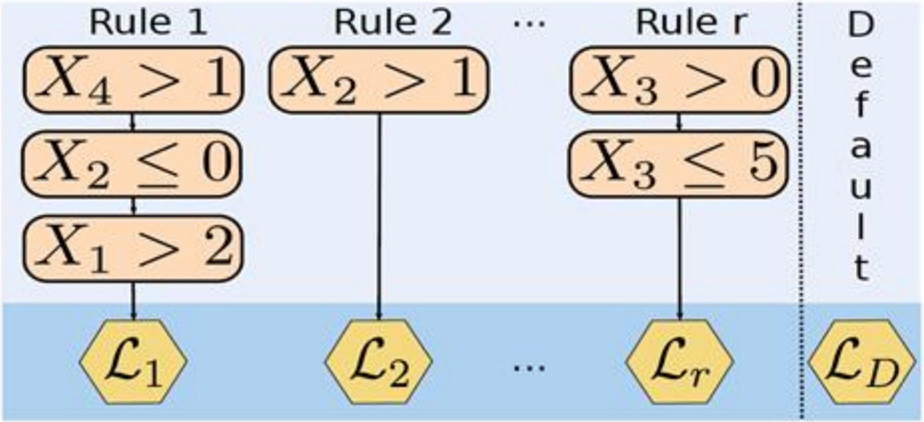
Regression Rules: the AMRules Algorithm

- One-pass algorithm: create, expand, and delete rules online
- Rule expansion: select the literal that most reduce variance of the target
- Uses the Hoeffding bound to decide how many observations are required to create/expand a rule
  - Hoeffding bound $\epsilon = \sqrt{R^2 ln(1/\delta)/(2n)}$
  - Expand when $\sigma_{1st}/\sigma_{2nd} < 1 - \epsilon$
- Evict rule when P-H signals an alarm

```
Input: S: Stream of examples
begin
    R ← {}, D ← 0
    foreach (X, y) ∈ S do
        foreach Rule r ∈ R do
            if ¬IsAnomaly(X, r)
             then
                if PHTest(error_r,
                 λ) then
                    Remove the
                     rule from R
                end
                else
                    Update
                     sufficient
                     statistics L_r
                    ExpandRule(r)
                end
            end
        end
        if S(X) = ∅ then
            Update L_D
            ExpandRule(D)
            if D expanded then
                R ← R ∪ D
                D ← 0
            end
        end
    end
    return (R, L_D)
end
```

Algorithm 1: Training AMRules

Rule Sets



- There are two types of rule sets: **unordered** and **ordered**.
- The support $S^u(X)$ of an unordered rule set given X is the set of rules that cover X.
- The support $S^o(X)$ of an ordered rule set is the first rule of $S^u(X)$.
- Given X, only the rules $R_l \in S(X)$ are used for training/testing. The default rule is used if $S(X) = \emptyset$.

Chebyshev Inequality

Let **Y** be a random variable with finite expected value and finite non-zero variance. Then for any real number **t > 0**

$$P(|y - \bar{y}| \geq t \times \sigma) \leq \frac{1}{t^2}$$

- No more than $1/t^2$ of the distribution's values can be *t* or more standard deviations away from the mean
- The probability of observing values far from the mean is low
- The probability of observing rare cases - the failures - is low[2]

[2] E. Aminian, R. P. Ribeiro, J. Gama: Chebyshev approaches for imbalanced data streams regression models. Data Min. Knowl. Discov. 2021

Chebyshev over-sampling

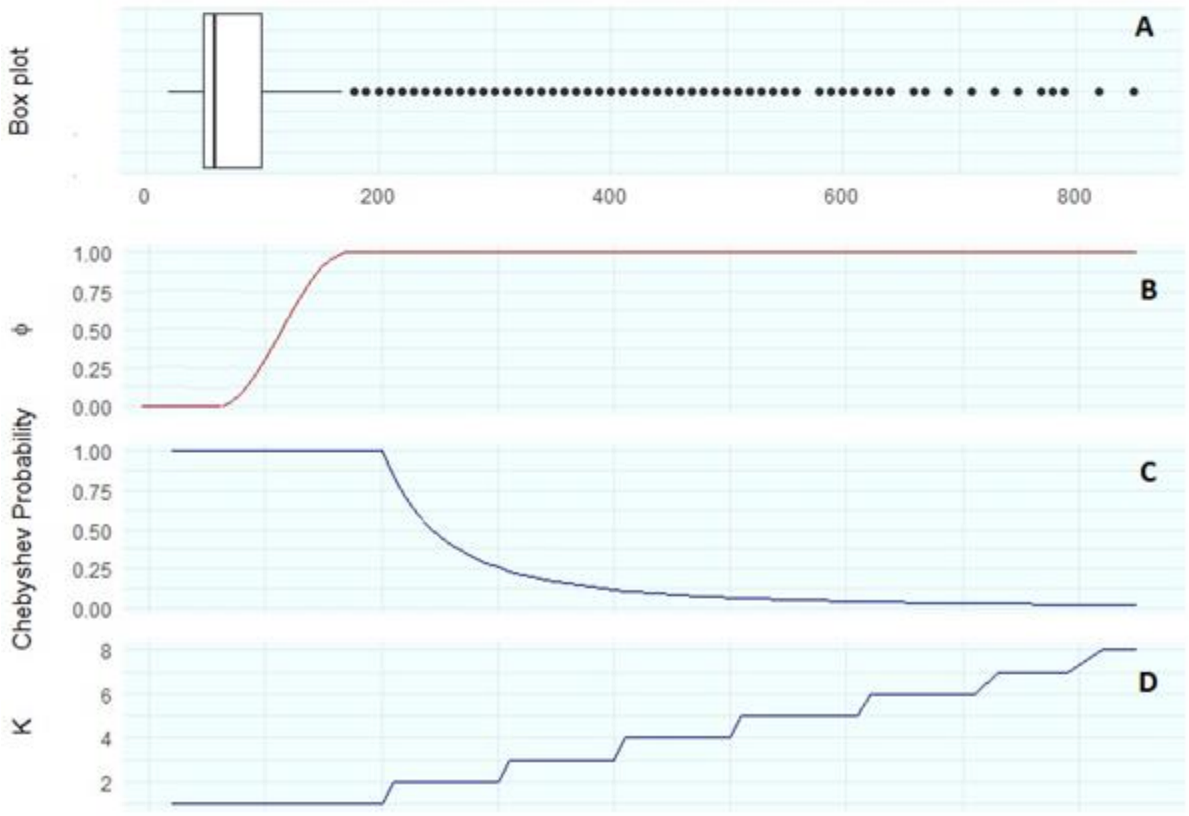
For each example:

- We compute $t = \frac{|y-\overline{y}|}{\sigma}$.
  - **t** is small for values of **y** near the mean
  - **t** is large for values of **y** far from the mean
- The example is passed to the learning algorithm ***K*** times

$$K = \left\lceil \frac{|y-\overline{y}|}{\sigma} \right\rceil.$$

- ***K*** has large values for the rare cases

Chebyshev over-sampling



Experimental Evaluation

The Reported Failures

#	Start Time	End Time	Failure	LPS Time
1	2022-06-04 10:19:24.300	2022-06-04 14:22:39.188	Air Leak	2022-06-04 11:26:01.422
2	2022-07-11 10:10:18.948	2022-07-14 10:22:08.046	Oil Leak	2022-07-13 19:43:52.593

Table 1. Maintenance Report - Failures

Experimental Setup

- **Methods**:
  - LSTM Sparse Autoencoder
  - TCN Autoencoder
  - WAE-GAN
- **Input Sequence of *t* time-stamps**
  - Compressor cycles
  - **Window of 30 m**

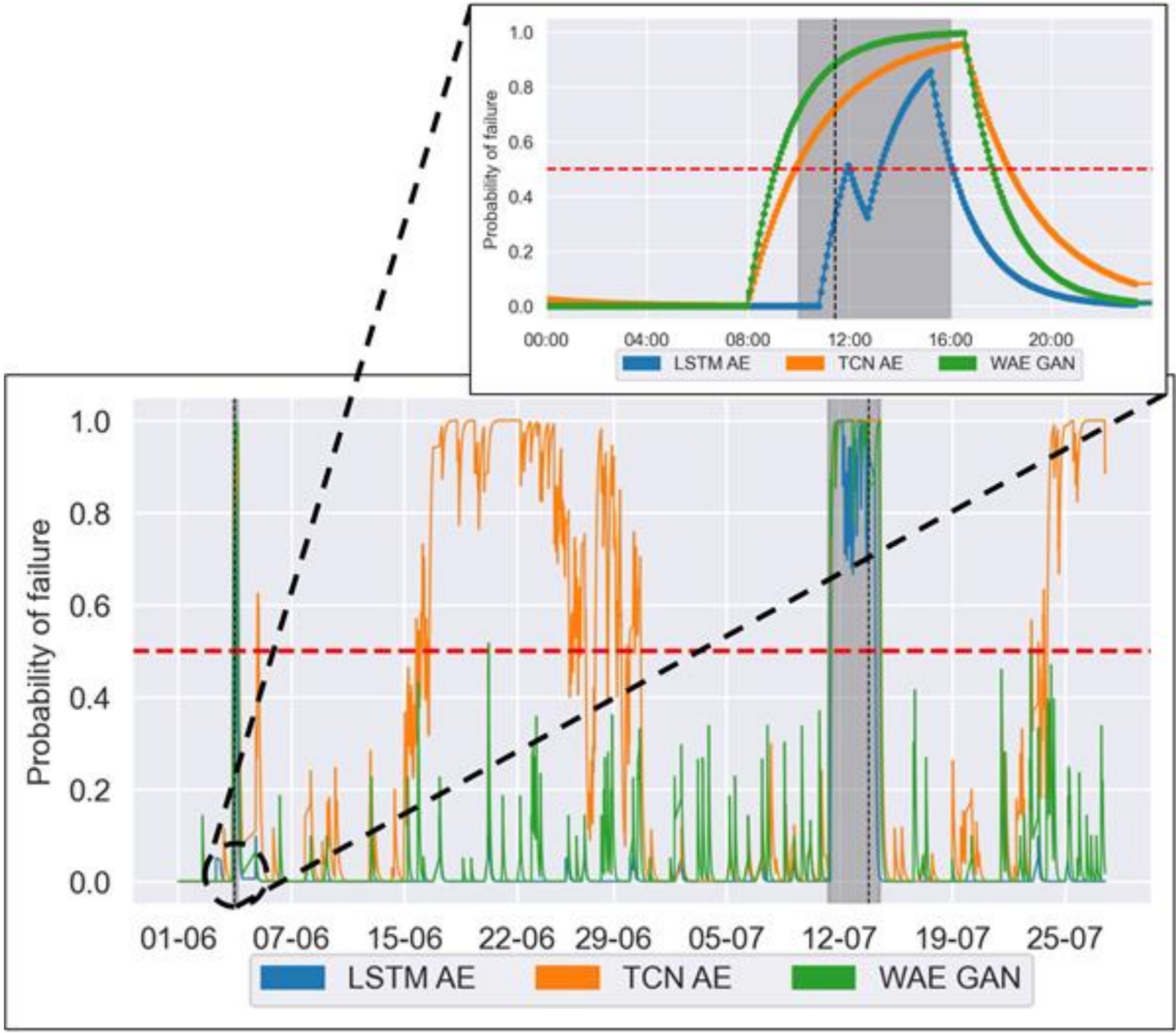
Based on 30-minute data chunks

After hyperparameter tuning:

- LSTM Autoencoder: 5 LSTM layers, 4 neurons per layer.
- TCN Autoencoder: 8 TCN layers, 7-dimensional kernel, 6 hidden units per layer, 4-dimensional latent space (RFS = ~1500).
- WAE-GAN (4-dimensional latent-space):
    - Encoder and decoder: 10 TCN layers, 3-dimensional kernel, 30 hidden units per layer (RFS = ~2000).
    - Discriminator: 3 LSTM layers with 32 neurons per layer.

The red horizontal dotted line is the alarm threshold

The grey bars represent real failures reported by maintenance teams.



Discussion

- The WAE-GAN model is able to identify the two failures at least two hours before the LPS signal is active.
  - without generating any false alarm (achieving a perfect F_1 score).
- The TCN autoencoder is also able to detect both failures early
  - but generates two false alarms (F_1 of 0.67).
- The LSTM autoencoder is able to detect both failures
  - without generating a false alarm,
  - but is unable to detect the first failure before the LPS signal.

Explaining failures generated by the WAE-GAN

- **First failure - air leak:**
  - **H1 <= 8.8** bar and **Oil temperature > 58.5ºC**
    - active 68% of the air leak
  - **Oil temperature > 60.8ºC and TP2 > 9.2 bar and Reservoirs > 9.8 bar**
    - active 0.8% of the air leak
  - **Motor current > 3.8A and TP2 between 7.0 and 7.2 bar and Oil temperature > 58.5ºC**
    - active 7.3% of the air leak

Explaining failures generated by the WAE-GAN

- Second failure - oil leak:
    - **Reservoirs > .8.8 bar and Flowmeter > 0.2 m3/h and H1 <= 9.6 bar and Oil temperature between 65.1ºC and 71.5ºC**
        - active 37% of the oil leak
    - O**il temperature > 65.1ºC and H1 > 0 bar**
        - active 48% of the oil leak.
    - **Oil temperature > 54.6ºC and TP2 > 9.2 bar**
        - active 6.5% of the oil leak.
    - **Flowmeter > 25 m3/h and Oil temperature < 95.8ºC**
        - active 9.1% of the oil leak.

Thank you for your attention.

Any Questions?

XPM - eXplainable Predictive Maintenance
CHIST-ERA-19-XAI-012


Porto, Portugal
Welcome to:
ECMLPKDD 2025
15-19 September