Industrial Blind Anomaly Detection

Characterization of normality in sensors time-series ...

Mazen Alamir

CNRS / Amiral Technologies

Background & point of view!

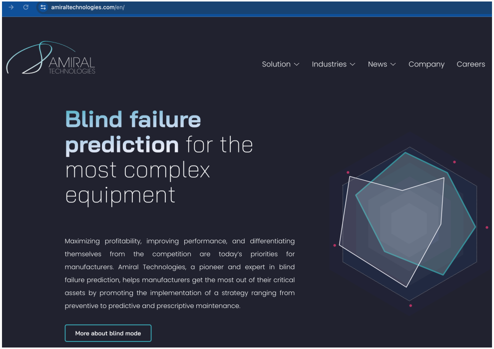
- ✓ Dynamical systems
- ✓ NL inverse problems
- ✓ Optimization
- ✓ **2018**: Creation of

AMIRAL TECHNOLOGIES

Features Generation from time series.

Deep-tech in industrial predictive maintenance (16-persons, Grenoble/Paris)

Background & point of view!



- ✓ Dynamical systems
- ✓ NL inverse problems
- ✓ Optimization
- ✓ **2018**: Creation of AMIRAL TECHNOLOGIES

Features Generation from time series.

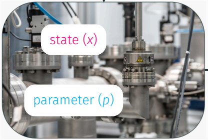
Deep-tech in industrial predictive maintenance (16-persons, Grenoble/Paris)

Problem statement

The specificity of industrial equipments ...

Context (c)

- Set-point
- Load change
- External temperature
- etc.



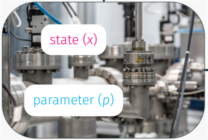
Sensors

- s_1
- s_2
- ...
- s_m

The specificity of industrial equipments ...

Context (c)

- Set-point
- Load change
- External temperature
- etc.



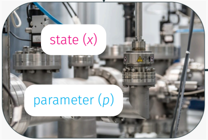
Sensors

- s_1
- s_2
- ...
- s_m

The specificity of industrial equipments ...

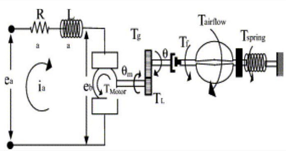
Context (c)

- Set-point
- Load change
- External temperature
- etc.



→ Sensors: s_1, s_2, ..., s_m

Example: The throttle control unit

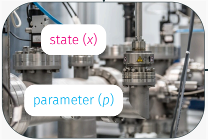


$$\ddot{\theta} = \frac{1}{J}\Big[-K_{sp}(\theta - \theta_0) - K_f\dot{\theta} + NK_t e_a + $$
$$- \pi R_p^2 R_{af}\Delta_p(\theta, P_m, N)\cos^2(\theta)\Big]$$
$$\dot{e}_a = \frac{1}{L_a}\left[-NK_b\dot{\theta} - R_a e_a + i_a\right]$$
$$i_a = Feedback(\theta, \dot{\theta}, \theta_{ref}, \dots)$$

The specificity of industrial equipments ...

Context (c)

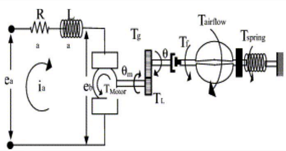
- Set-point
- Load change
- External temperature
- etc.



→ Sensors

- s_1
- s_2
- ...
- s_m

Example: The throttle control unit



$$\ddot{\theta} = \frac{1}{J}\Big[-K_{sp}(\theta - \theta_0) - K_f\dot{\theta} + NK_t e_a + - \pi R_p^2 R_{af}\Delta_p(\theta, P_m, N)\cos^2(\theta)\Big]$$

$$\dot{e}_a = \frac{1}{L_a}\big[-NK_b\dot{\theta} - R_a e_a + i_a\big]$$

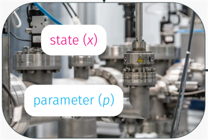
$$i_a = Feedback(\theta, \dot{\theta}, \theta_{ref}, \dots)$$

$x := (\theta, \dot{\theta}, e_a)$	State
$p = (J, K_{sp}, K_f, R_p, R_{af}, N, R_a)$	Parameter
$c = (\theta_{ref}, P_m)$	Context
$s := (\theta, i_a, \theta_{ref})$	Sensors

The specificity of industrial equipments ...

Context (c)

- Set-point
- Load change
- External temperature
- etc.



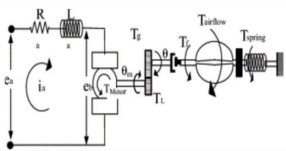
→ Sensors

- s_1
- s_2
- ...
- s_m

$\dot{x} = F(x, c, p)$ (State Equation)

$s = M(x, c, p)$ (Measurement Equation)

Example: The throttle control unit



$$\ddot{\theta} = \frac{1}{J}\big[-K_{sp}(\theta - \theta_0) - K_f\dot{\theta} + NK_t e_a + - \pi R_p^2 R_{af}\Delta_p(\theta, P_m, N)\cos^2(\theta)\big]$$

$$\dot{e}_a = \frac{1}{L_a}\big[-NK_b\dot{\theta} - R_a e_a + i_a\big]$$

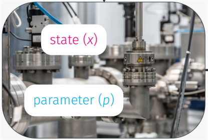
$$i_a = Feedback(\theta, \dot{\theta}, \theta_{ref}, \dots)$$

$x := (\theta, \dot{\theta}, e_a)$	State
$p = (J, K_{sp}, K_f, R_p, R_{af}, N, R_a)$	Parameter
$c = (\theta_{ref}, P_m)$	Context
$s := (\theta, i_a, \theta_{ref})$	Sensors

The specificity of industrial equipments ...

Context (c)

- Set-point
- Load change
- External temperature
- etc.



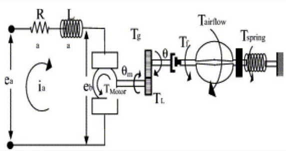
→ Sensors: s_1, s_2, ..., s_m

$$\dot{x} = F(x, c, p) \qquad \text{(State Equation)}$$

$$s = M(x, c, p) \qquad \text{(Measurement Equation)}$$

A digital twin is an algorithm that encodes these equations in a simulator with the appropriate parameters vector p.

Example: The throttle control unit



$$\ddot{\theta} = \frac{1}{J}\Big[-K_{sp}(\theta - \theta_0) - K_f\dot{\theta} + NK_t e_a + - \pi R_p^2 R_{af}\Delta_p(\theta, P_m, N)\cos^2(\theta)\Big]$$

$$\dot{e}_a = \frac{1}{L_a}\big[-NK_b\dot{\theta} - R_a e_a + i_a\big]$$

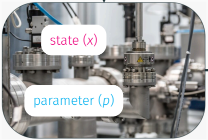
$$i_a = Feedback(\theta, \dot{\theta}, \theta_{ref}, \dots)$$

$x := (\theta, \dot{\theta}, e_a)$	State
$p = (J, K_{sp}, K_f, R_p, R_{af}, N, R_a)$	Parameter
$c = (\theta_{ref}, P_m)$	Context
$s := (\theta, i_a, \theta_{ref})$	Sensors

The time-series-based anomalies detection problem

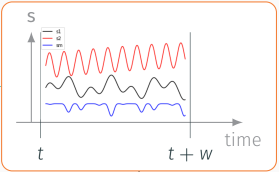
Context (c)

- Set-point
- Load change
- External temperature
- etc.



Sensors

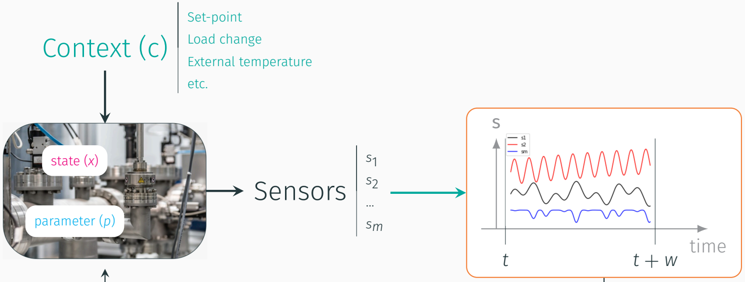
s_1
s_2
...
s_m



$p \neq p_{\text{healthy}}$?

Design an algorithm that detects anomalous variations in the system's parameters p over the interval $[t, t + w]$.

The time-series-based anomalies detection problem



$p \neq p_{\text{healthy}}$?

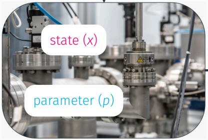
Using only healthy learning data

Design an algorithm that detects anomalous variations in the system's parameters p over the interval $[t, t + w]$.

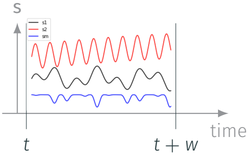
The time-series-based anomalies detection problem

Context (c)

- Set-point
- Load change
- External temperature
- etc.



Sensors: s_1, s_2, ..., s_m



$p \neq p_{\text{healthy}}$?

Using only healthy learning data

Design an algorithm that detects anomalous variations in the system's parameters p over the interval $[t, t+w]$.

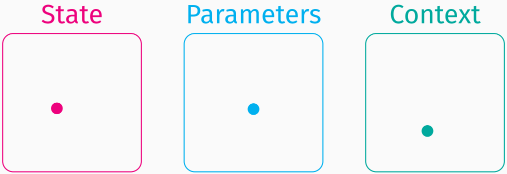
depends on $x(t)$
depends on $p_t^{t+w}(\cdot)$
depends on $c_t^{t+w}(\cdot)$

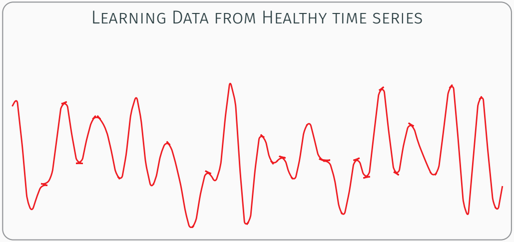
Because

$$\dot{x} = F(x, c, p)$$

$$s = M(x, c, p)$$

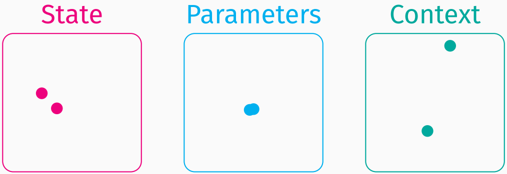
The State-Context induced Ambiguity

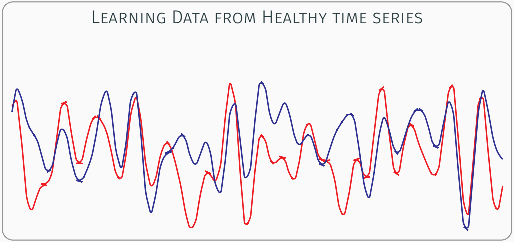




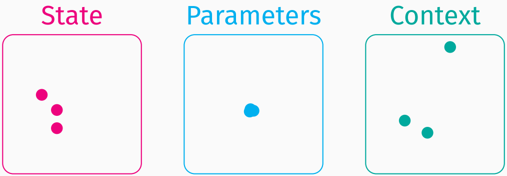
Time series on $\left\{[t_i, t_i + W]\right\}_{i \in \mathcal{I}_{\text{learning}}}$

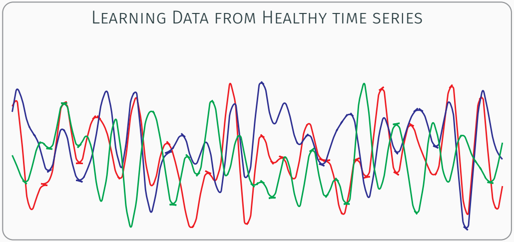
The State-Context induced Ambiguity





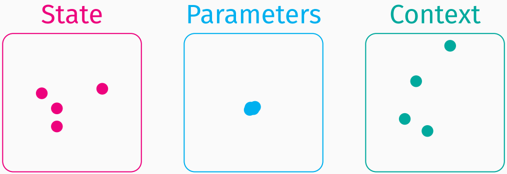
Time series on $\Big\{[t_i, t_i + W]\Big\}_{i\in\mathcal{I}_{\text{learning}}}$

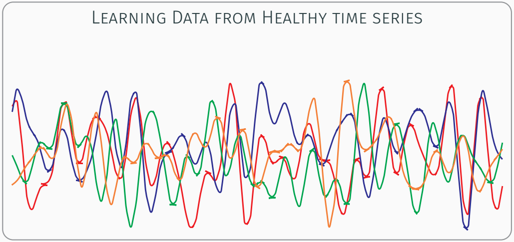




Time series on $\left\{[t_i, t_i + W]\right\}_{i \in \mathcal{I}_{\text{learning}}}$

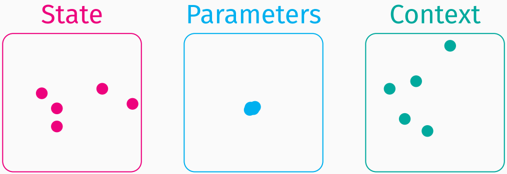
The State-Context induced Ambiguity

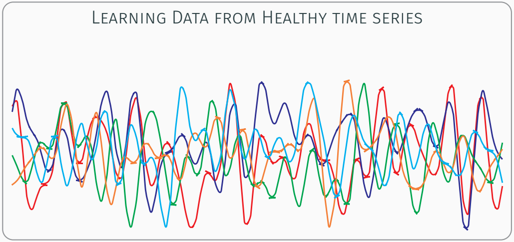




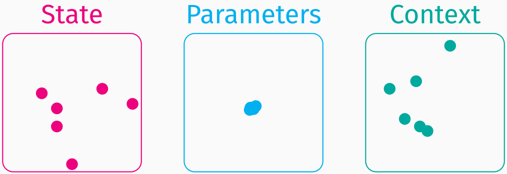
Time series on $\left\{[t_i, t_i + W]\right\}_{i \in \mathcal{I}_{\text{learning}}}$

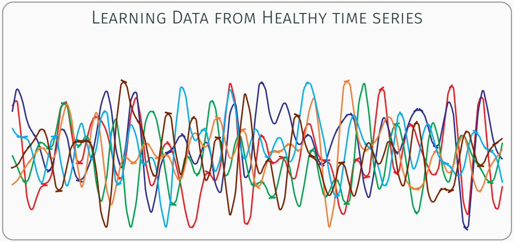
The State-Context induced Ambiguity





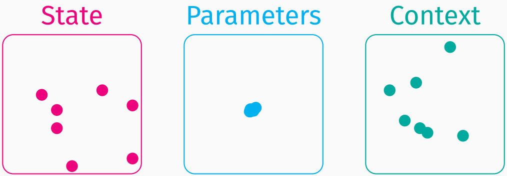
Time series on $\left\{[t_i, t_i + W]\right\}_{i\in\mathcal{I}_{\text{learning}}}$

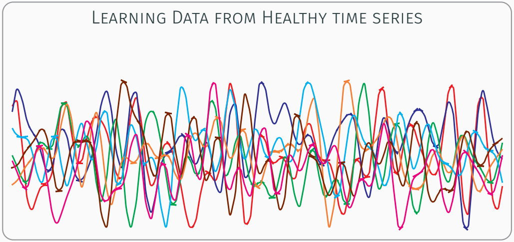




Time series on $\left\{[t_i, t_i + W]\right\}_{i \in \mathcal{I}_{\text{learning}}}$

The State-Context induced Ambiguity

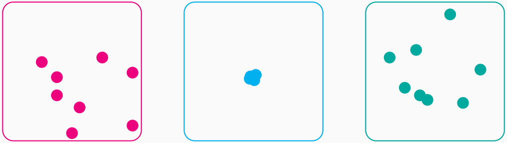


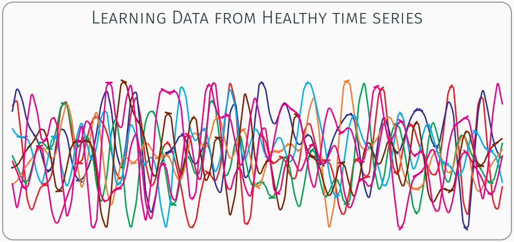


Time series on $\Big\{[t_i, t_i + W]\Big\}_{i \in \mathcal{I}_{\text{learning}}}$

The State-Context induced Ambiguity

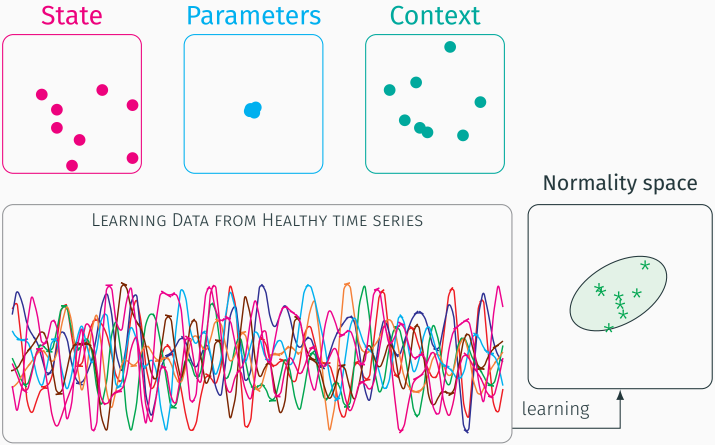
State Parameters Context





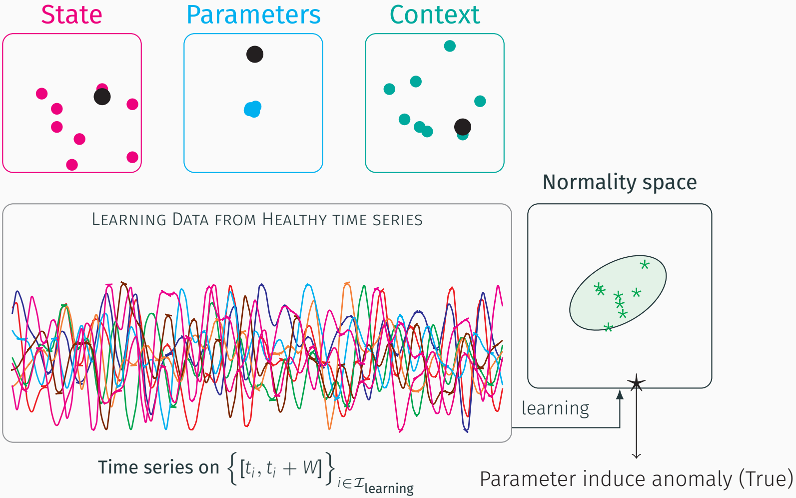
Time series on $\left\{[t_i, t_i + W]\right\}_{i\in\mathcal{I}_{\text{learning}}}$

The State-Context induced Ambiguity

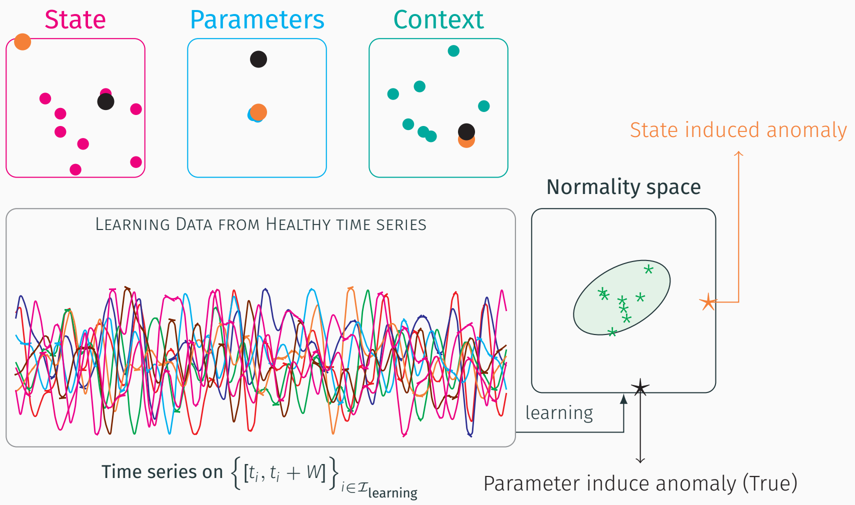


Time series on $\left\{[t_i, t_i + W]\right\}_{i \in \mathcal{I}_{\text{learning}}}$

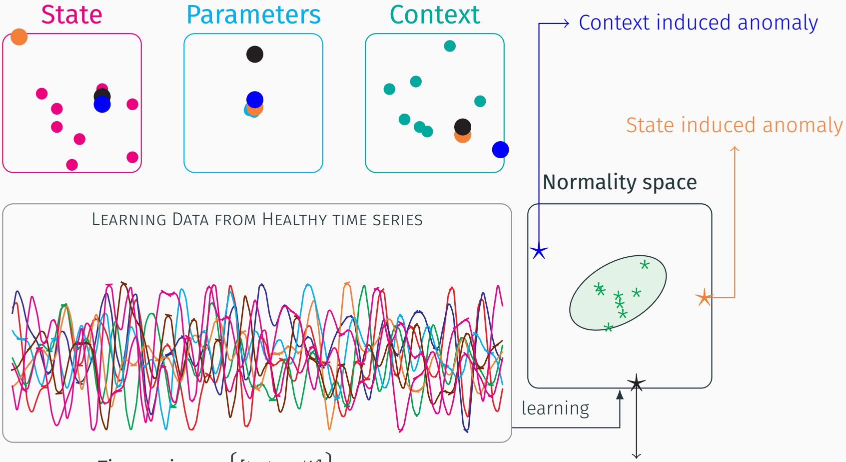
The State-Context induced Ambiguity



The State-Context induced Ambiguity

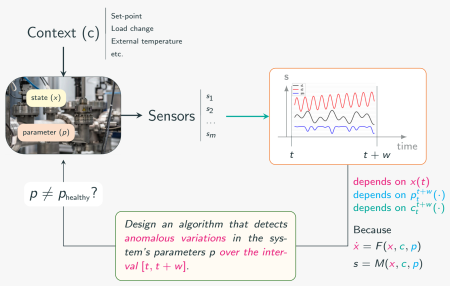


The State-Context induced Ambiguity



Time series on $\left\{[t_i, t_i + W]\right\}_{i\in\mathcal{I}_{\text{learning}}}$

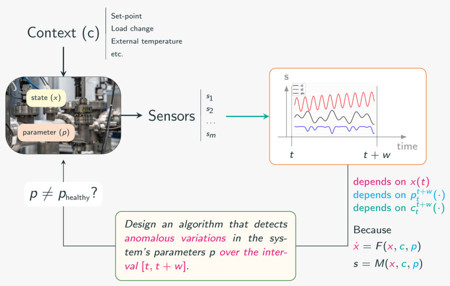
The State/Context ambiguity (SC-Ambiguity)



SC-Ambiguity

Refers to the changes in the time series that are NOT ONLY due to changes in the parameters $p_t^{t+w}(\cdot)$ but also to potentially unseen values of the initial state $x(t)$ or in the context profile $c_t^{t+w}(\cdot)$ or both!

SC-Ambiguity $\leftrightarrow$ *State/Context-induced Ambiguity*!



SC-Restricted Problem

Refers to the changes in the time series that are ONLY due to changes in the parameter $p_t^{t+w}(\cdot)$ since the initial state $x(t)$ and the context profile $c_t^{t+w}(\cdot)$ are almost perfectly reproducible

SC-Restricted ↔ *State/Context-Restricted*!

Coming next

SC-Restricted problems

- ✓ The **Enigma** principle
- ✓ A Toy illustrative example
- ✓ Two industrial examples

SC-Ambiguous problems

- ✓ An introductory example
- ✓ The **Invariance** principle
- ✓ An illustrative example

Coming next

SC-Restricted problems

- ✓ The **Enigma** principle
- ✓ A Toy illustrative example
- ✓ Two industrial examples

SC-Ambiguous problems

- ✓ An introductory example
- ✓ The **Invariance** principle
- ✓ An illustrative example

Two problems
One challenge:

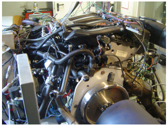
Characterization of normality

What are the properties present in the healthy time series coming from the available sensors that should be considered as relevant set of **characterization of normal behavior** so that when they are not satisfied, alarm should be raised?

SC-R Problems – (Cyclic Data)

Algorithms for SC-R problems: The principle (1)

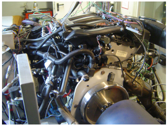
SoH $\leftrightarrow$ p



Measurement $\boldsymbol{s}$ →

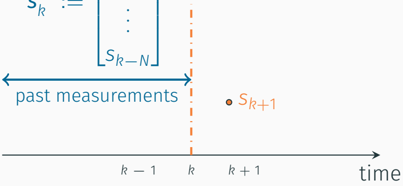
Algorithms for SC-R problems: The principle (1)

SoH $\leftrightarrow p$



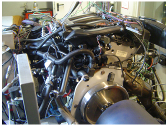
Measurement $\boldsymbol{s}$

$$\mathbf{s}_k^- := \begin{bmatrix} s_k \\ s_{k-1} \\ \vdots \\ s_{k-N} \end{bmatrix}$$



Algorithms for SC-R problems: The principle (1)

SoH $\leftrightarrow p$



Measurement $\boldsymbol{s}$

$$\mathbf{s}_k^- := \begin{bmatrix} s_k \\ s_{k-1} \\ \vdots \\ s_{k-N} \end{bmatrix}$$

past measurements

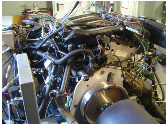
s_{k+1}

$$s_{k+1} = R(\mathbf{s}_k^- \mid p)$$

$k-1$ $\quad k \quad$ $k+1$ $\quad$ time

Algorithms for SC-R problems: The principle (1)

SoH $\leftrightarrow p$



Measurement $\boldsymbol{s}$

NOTA

True for SC-Restricted Problem problems where the initial state and the context are fixed.

$$\boldsymbol{s}_k^- := \begin{bmatrix} s_k \\ s_{k-1} \\ \vdots \\ s_{k-N} \end{bmatrix}$$

$$s_{k+1} = R(\boldsymbol{s}_k^- \mid p)$$

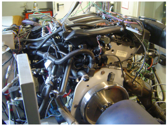
past measurements

s_{k+1}

$k-1$ k $k+1$ time

Algorithms for SC-R problems: The principle (1)

SoH $\leftrightarrow p$

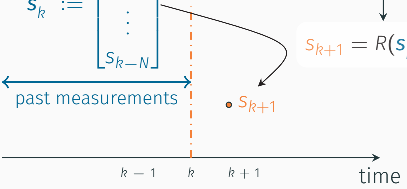


Measurement $\mathbf{s}$

NOTA

True for SC-Restricted Problem problems where the initial state and the context are fixed.

$$\mathbf{s}_k^- := \begin{bmatrix} s_k \\ s_{k-1} \\ \vdots \\ s_{k-N} \end{bmatrix}$$



$$s_{k+1} = R(\mathbf{s}_k^- \mid p)$$

***Prediction Error*-Based Approaches**

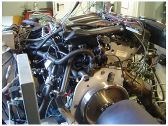
- ✓ Compute $\hat{R}$ via ML / Deep NN
- ✓ Tune th s.t the rule:

$$|s_{k+1} - \hat{R}(\mathbf{s}_k^-)| > th \rightarrow \textit{Alarm}$$

induces acceptable False Alarm rate and high detection rate.

Algorithms for SC-R problems: The principle (1)

SoH $\leftrightarrow p$



Measurement $\boldsymbol{s}$

NOTA

True for SC-Restricted Problem problems where the initial state and the context are fixed.

$$\boldsymbol{s}_k^- := \begin{bmatrix} s_k \\ s_{k-1} \\ \vdots \\ s_{k-N} \end{bmatrix}$$

past measurements

s_{k+1}

$k-1$ k $k+1$ time

$$s_{k+1} = R(\boldsymbol{s}_k^- \mid p)$$

Prediction Error-Based Approaches

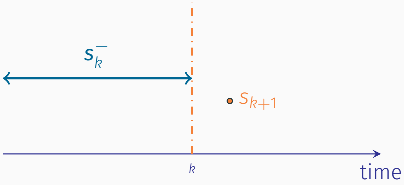
- ✓ Compute $\hat{R}$ via ML / Deep NN
- ✓ Tune th s.t the rule:

$$|s_{k+1} - \hat{R}(\boldsymbol{s}_k^-)| > th \rightarrow \textit{Alarm}$$

induces acceptable False Alarm rate and high detection rate.

Next slide explains this in more details →

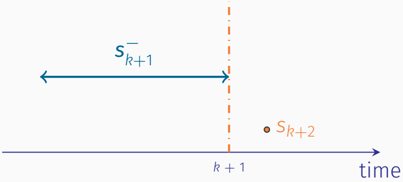
Algorithms for SC-R problems: The principle (2)



SC-Restricted problems

$$s_{k+1} \in R(\mathbf{s}_k^- \mid p)$$

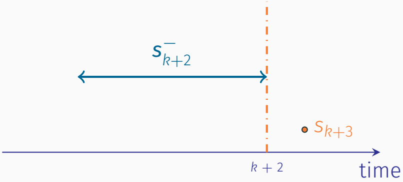
Algorithms for SC-R problems: The principle (2)

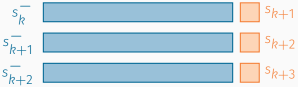


SC-Restricted problems

$$s_{k+1} \in R(\mathbf{s}^-_k \mid p)$$

Algorithms for SC-R problems: The principle (2)

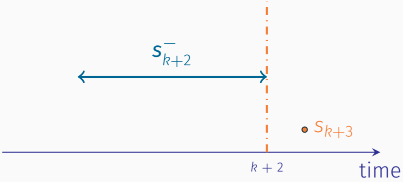


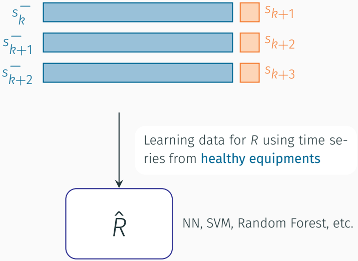


SC-Restricted problems

$$s_{k+1} \in R(\mathbf{s}^-_k \mid p)$$

Algorithms for SC-R problems: The principle (2)

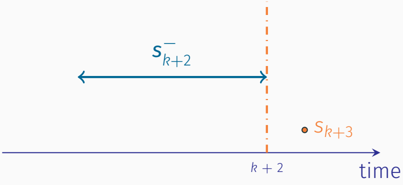


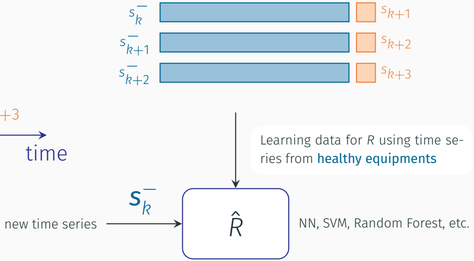


SC-Restricted problems

$$s_{k+1} \in R(\mathbf{s}^{-}_{k} \mid p)$$

Algorithms for SC-R problems: The principle (2)

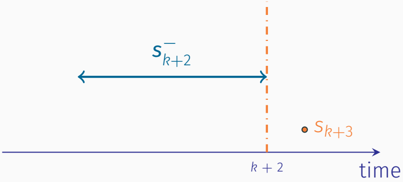


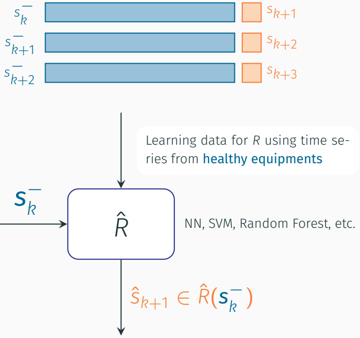


SC-Restricted problems

$$s_{k+1} \in R(\mathbf{s}^{-}_{k} \mid p)$$

Algorithms for SC-R problems: The principle (2)

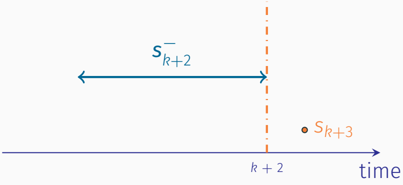


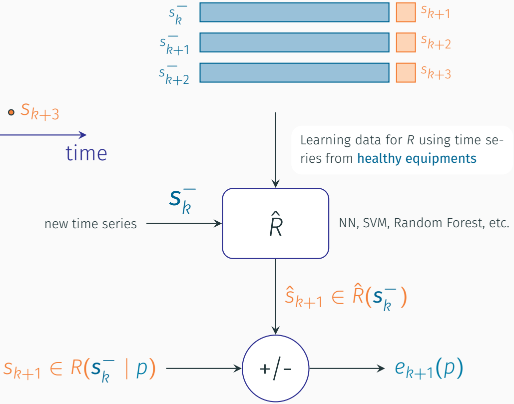


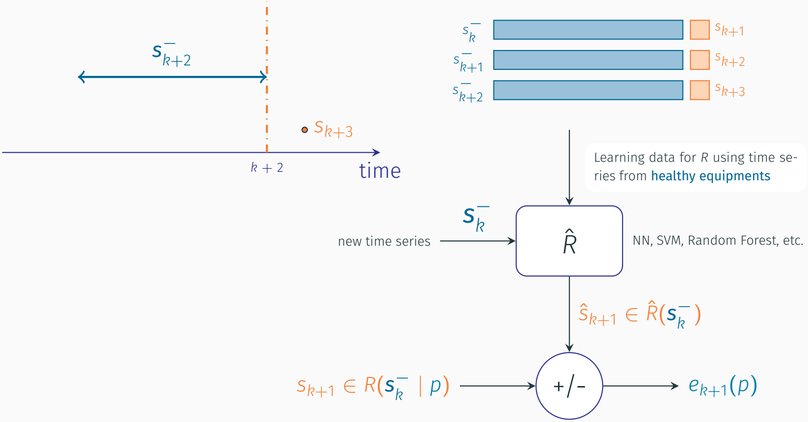
SC-Restricted problems

$$s_{k+1} \in R(s^-_k \mid p)$$

Algorithms for SC-R problems: The principle (2)

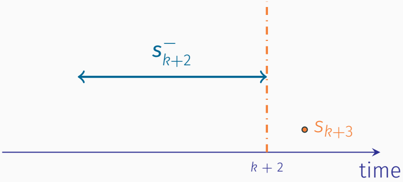




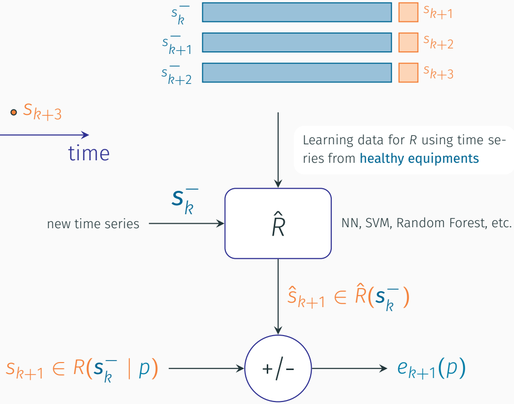


⟶ if e_k is small (OK) otherwise **Raise Alarm** !

Algorithms for SC-R problems: The principle (2)

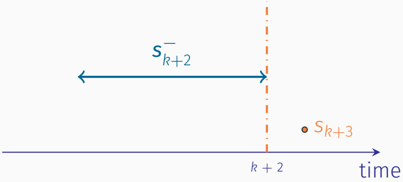


- No many tuning options
- → one single arbitrary view of R

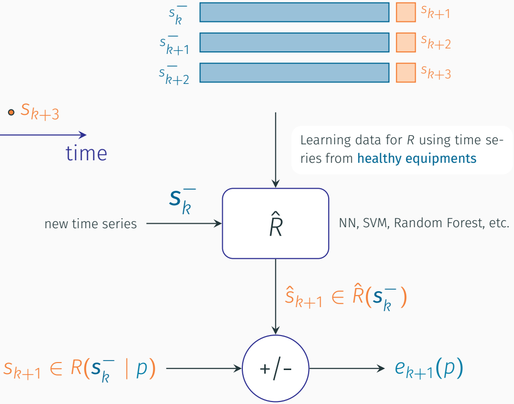


⟶ if e_k is small (OK) otherwise **Raise Alarm** !

Algorithms for SC-R problems: The principle (2)

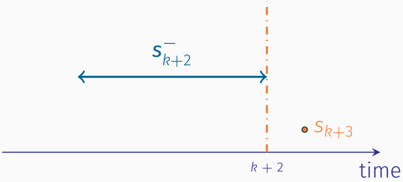


- No many tuning options
- $\longrightarrow$ one single arbitrary view of R

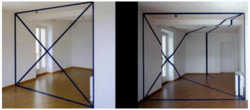


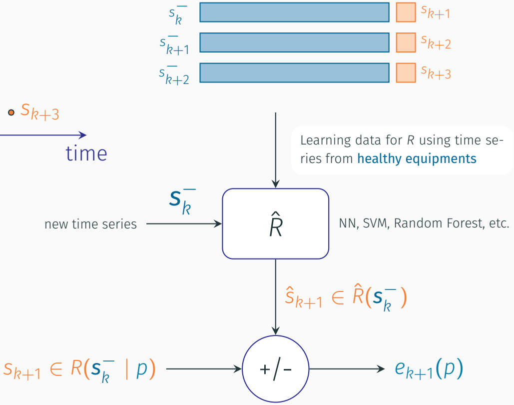
$\longrightarrow$ if e_k is small (OK) otherwise **Raise Alarm** !

Algorithms for SC-R problems: The principle (2)



- No many tuning options
- → one single arbitrary view of R



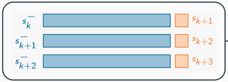


⟶ if e_k is small (OK) otherwise **Raise Alarm** !

Algorithms for SC-R problems: Enigma principle

Seed:σ

The seed defines the map M_σ
There is an ∞ number of values for seed
Each produces $\neq$ map



$$M_\sigma : \mathbb{R}^N \times \mathbb{R} \rightarrow \mathbb{R}^m$$

$$(s^-, s_+) \rightsquigarrow S(p)$$

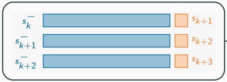
Algorithms for SC-R problems: Enigma principle

$$M_\sigma(s^-, s_+) := \begin{bmatrix} s^- \times s_+{}^{\sigma} \\ \sin(2\sigma(s^- - s_+)) \\ \cos(\sigma |y^-|) \end{bmatrix}$$

$N = 1, m = 3$

Seed: σ

The seed defines the map M_σ
There is an ∞ number of values for seed
Each produces $\neq$ map



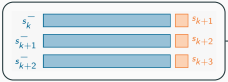
$$M_\sigma : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^m$$

$$(s^-, s_+) \rightsquigarrow S(p)$$

Algorithms for SC-R problems: Enigma principle

Seed:σ

The seed defines the map M_σ
There is an ∞ number of values for seed
Each produces $\neq$ map

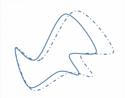


$$M_\sigma : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^m$$

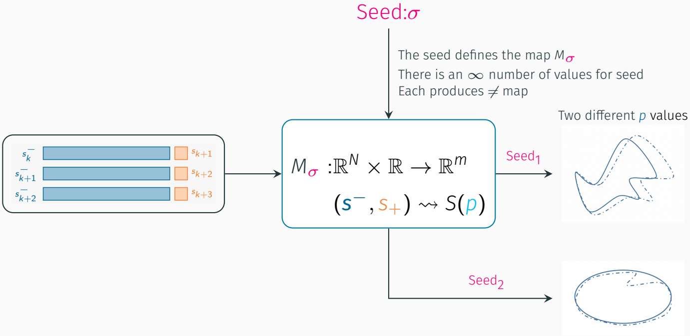
$$(s^-, s_+) \rightsquigarrow S(p)$$

Seed$_1$

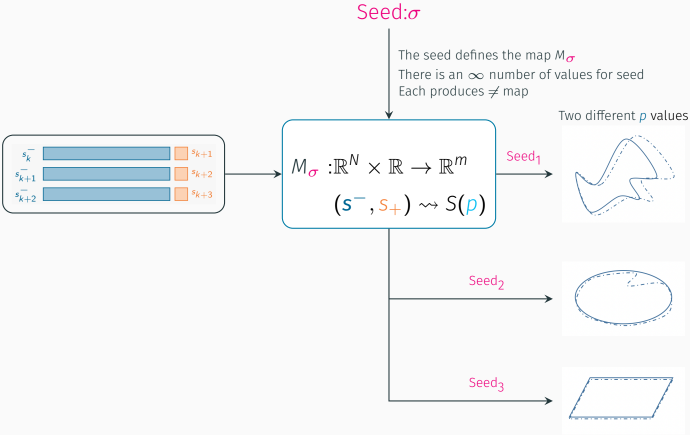
Two different p values



Algorithms for SC-R problems: Enigma principle



Algorithms for SC-R problems: Enigma principle



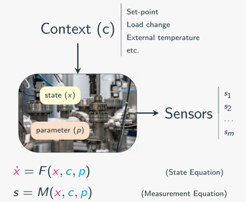
Illustrative example

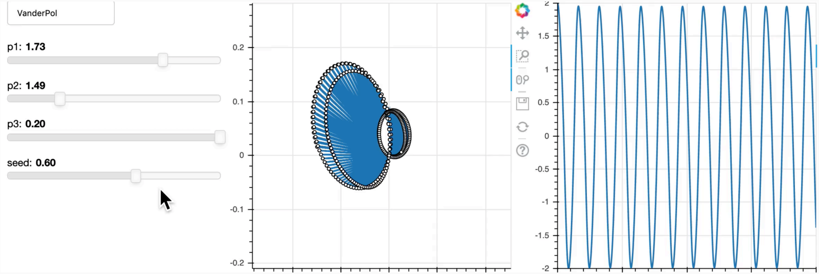
State and measurement equations

$$\dot{x}_1 = p_1 x_1$$

$$\dot{x}_2 = -9x_1 + p_2 x_2 (1 - (x_1 + p_3)^2)$$

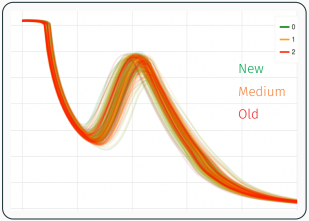
$$s_1 = x_1$$





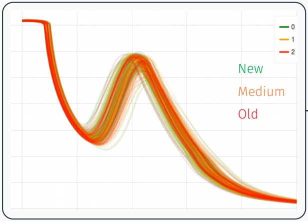
Example 2: Contactors wear example

Note the SCR character of the problem! Same initial state and no context.

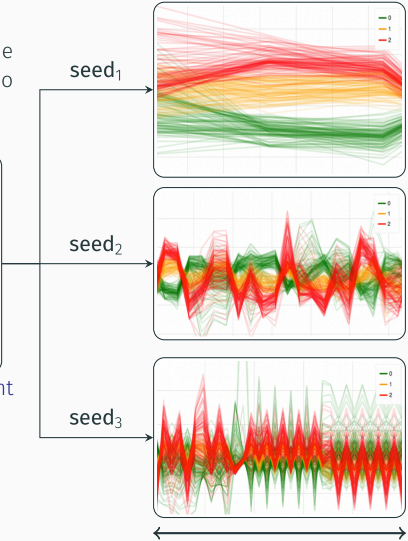


$\approx$ 20 (ms) disconnection current

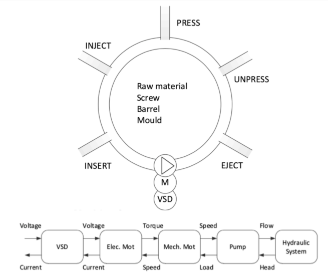
Note the SCR character of the problem! Same initial state and no context.

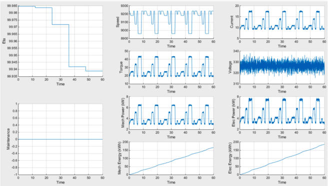


≈ 20 (ms) disconnection current

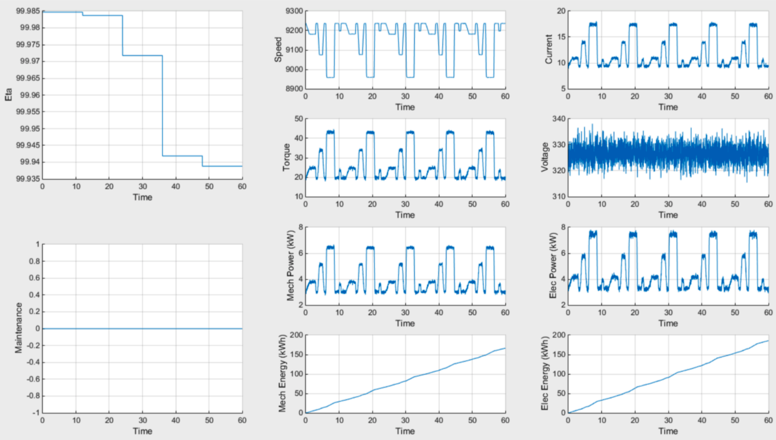


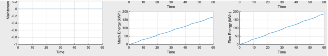
400 Features(**seed**$_3$)


PRESS
INJECT
UNPRESS
Raw material
Screw
Barrel
Mould
INSERT
EJECT
M
VSD
Voltage
Voltage
Torque
Speed
Flow
VSD
Elec. Mot
Mech. Mot
Pump
Hydraulic System
Current
Current
Speed
Load
Head

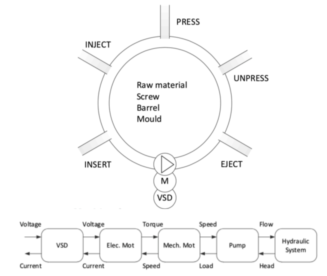


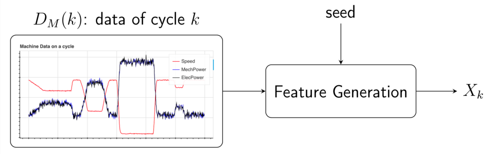
Example 3: Predicting quality & Prescriptive maintenance

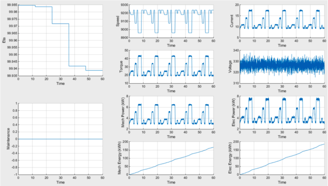




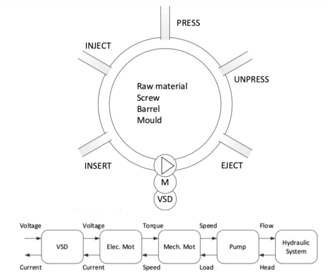
Example 3: Predicting quality & Prescriptive maintenance

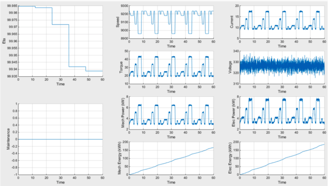


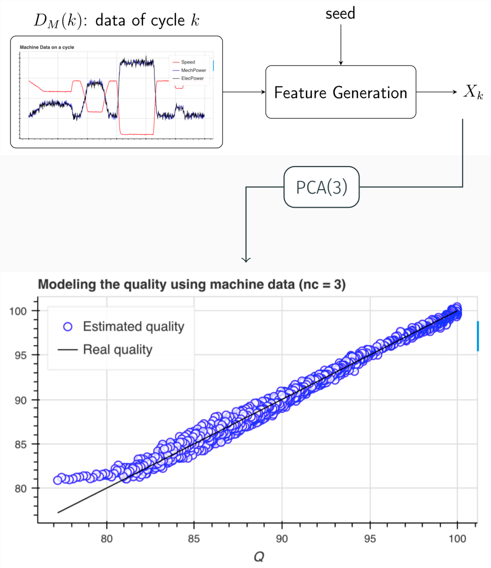




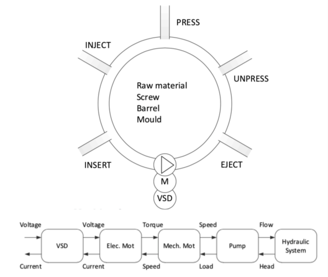
Example 3: Predicting quality & Prescriptive maintenance

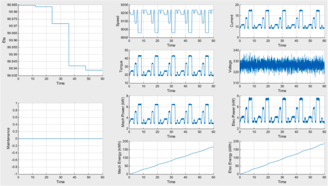


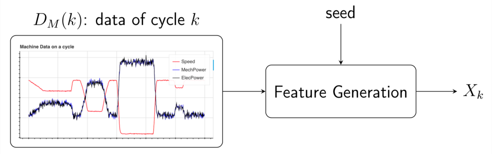


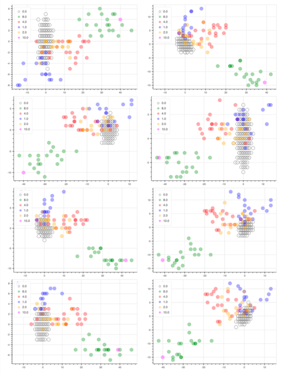


Example 3: Predicting quality & Prescriptive maintenance









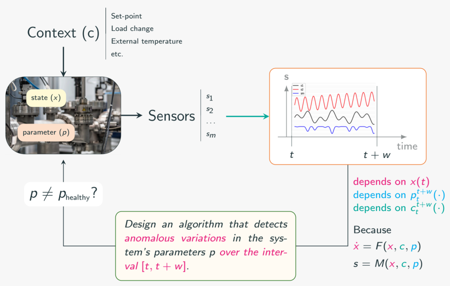
8 Different views in the first two PCA for 8 different seeds used in the features generation.

Why ≠ seed?

Notice how using different seeds enables by combining the analysis on each resulting view, to get rid of the ambiguity that shows on some of the individual views. This improve the quality of the resulting prescriptive maintenance.

Non-cyclic Data (The ultimate challenge)

The SC-Ambiguity



SC-Ambiguity

Refers to the changes in the time series that are NOT due to change in the parameter $p_t^{t+w}(\cdot)$ but to unseen values of the initial state $x(t)$ or in the context profile $c_t^{t+w}(\cdot)$ or both!

SC-Ambiguity ↔ *State/Context-induced Ambiguity*!

Take the simple forced oscillator:

$$\ddot{s} = -\omega^2 s - f\dot{s} + gu$$

Healthy	$(\omega, f, g) = (2.0, 0.05, 1.0)$
Faulty	$(\omega, f, g) = (1.8, 0.08, 1.1)$

An illustrative example

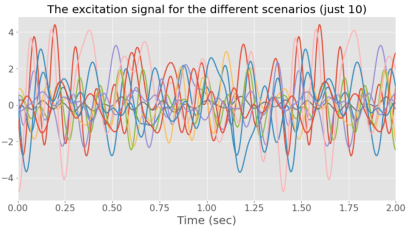
Take the simple forced oscillator:

$$\ddot{s} = -\omega^2 s - f\dot{s} + gu$$

Healthy $(\omega, f, g) = (2.0, 0.05, 1.0)$

Faulty $(\omega, f, g) = (1.8, 0.08, 1.1)$

200 profiles for the excitation scenario **u**



showing only the first 10

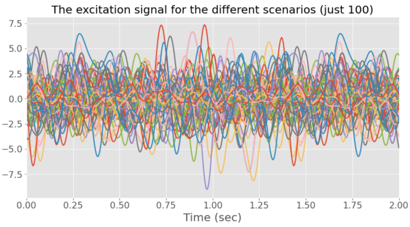
An illustrative example

Take the simple forced oscillator:

$$\ddot{s} = -\omega^2 s - f\dot{s} + gu$$

Healthy	$(\omega, f, g) = (2.0, 0.05, 1.0)$
Faulty	$(\omega, f, g) = (1.8, 0.08, 1.1)$

200 profiles for the excitation scenario **u**



showing the first 100 profiles

An illustrative example

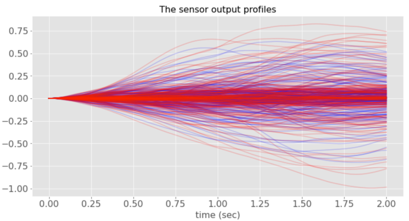
Take the simple forced oscillator:

$$\ddot{s} = -\omega^2 s - f\dot{s} + gu$$

Healthy $(\omega, f, g) = (2.0, 0.05, 1.0)$

Faulty $(\omega, f, g) = (1.8, 0.08, 1.1)$

200 corresponding sensor measurements



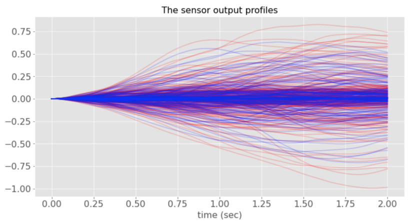
An illustrative example

Take the simple forced oscillator:

$$\ddot{s} = -\omega^2 s - f\dot{s} + gu$$

Healthy	$(\omega, f, g) = (2.0, 0.05, 1.0)$
Faulty	$(\omega, f, g) = (1.8, 0.08, 1.1)$

200 corresponding sensor measurements



An illustrative example

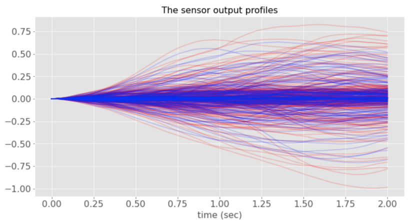
Take the simple forced oscillator:

$$\ddot{s} = -\omega^2 s - f\dot{s} + gu$$

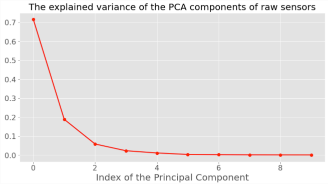
Healthy $(\omega, f, g) = (2.0, 0.05, 1.0)$

Faulty $(\omega, f, g) = (1.8, 0.08, 1.1)$

200 corresponding sensor measurements



3 components → 95% of the variance



An illustrative example

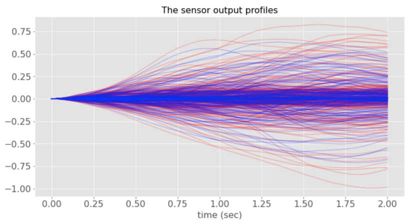
Take the simple forced oscillator:

$$\ddot{s} = -\omega^2 s - f\dot{s} + gu$$

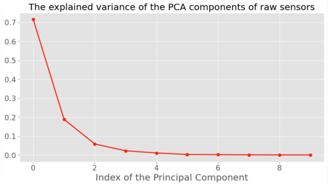
Healthy $(\omega, f, g) = (2.0, 0.05, 1.0)$

Faulty $(\omega, f, g) = (1.8, 0.08, 1.1)$

200 corresponding sensor measurements



3 components → 95% of the variance



Feature generation method 1

Ignoring dynamic systems specificity ...

$s \longrightarrow$ $PCA(3)$ $\longrightarrow$ pca_1, pca_2, pca_3

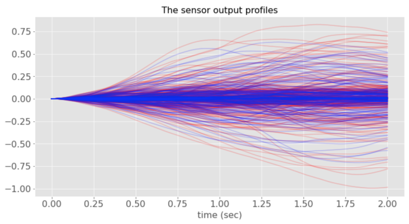
An illustrative example

Take the simple forced oscillator:

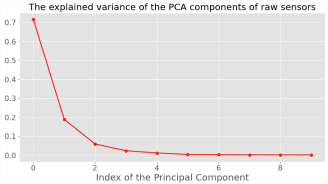
$$\ddot{s} = -\omega^2 s - f\dot{s} + gu$$

Healthy	$(\omega, f, g) = (2.0, 0.05, 1.0)$
Faulty	$(\omega, f, g) = (1.8, 0.08, 1.1)$

200 corresponding sensor measurements



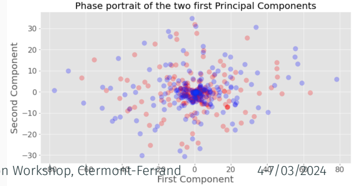
3 components → 95% of the variance



Feature generation method 1

Ignoring dynamic systems specificity ...

$s \longrightarrow PCA(3) \longrightarrow pca_1, pca_2, pca_3$

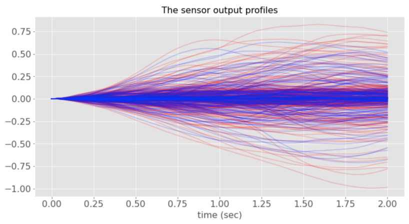


Take the simple forced oscillator:

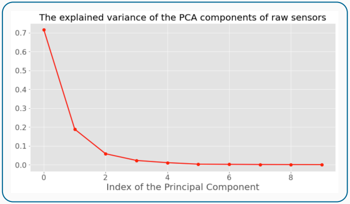
$$\ddot{s} = -\omega^2 s - f\dot{s} + gu$$

Healthy	$(\omega, f, g) = (2.0, 0.05, 1.0)$
Faulty	$(\omega, f, g) = (1.8, 0.08, 1.1)$

200 corresponding sensor measurements

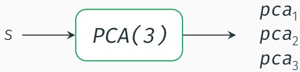


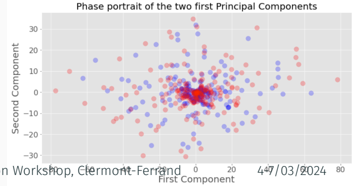
3 components $\rightarrow$ 95% of the variance



Feature generation method 1

Ignoring dynamic systems specificity ...





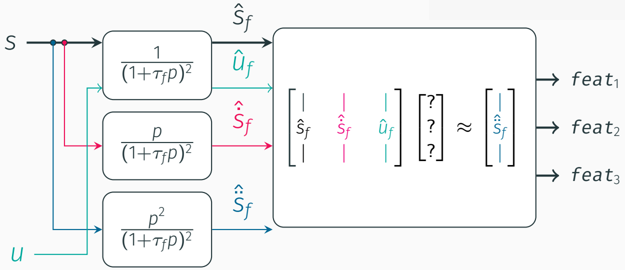
An illustrative example – (continued)

Take the simple forced oscillator:

$$\ddot{s} = -\omega^2 s - f\dot{s} + gu = \begin{bmatrix} s & \dot{s} & u \end{bmatrix} \cdot \begin{bmatrix} -\omega^2 \\ -f \\ g \end{bmatrix}$$

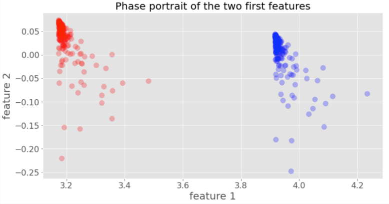
Healthy $(\omega, f, g) = (2.0, 0.05, 1.0)$

Faulty $(\omega, f, g) = (1.8, 0.08, 1.1)$



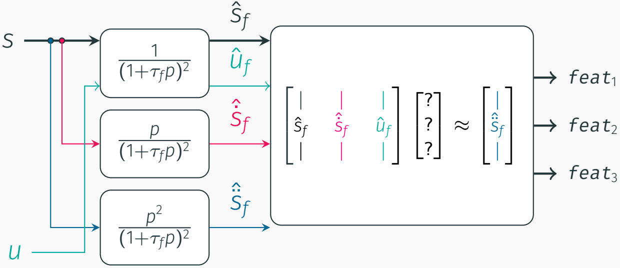
Take the simple forced oscillator:

$$\ddot{s} = -\omega^2 s - f\dot{s} + gu = \begin{bmatrix} s & \dot{s} & u \end{bmatrix} \cdot \begin{bmatrix} -\omega^2 \\ -f \\ g \end{bmatrix}$$



Healthy $(\omega, f, g) = (2.0, 0.05, 1.0)$

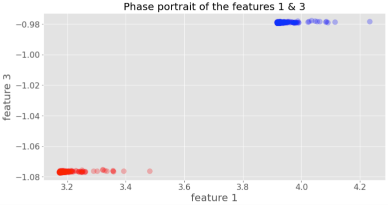
Faulty $(\omega, f, g) = (1.8, 0.08, 1.1)$



An illustrative example – (continued)

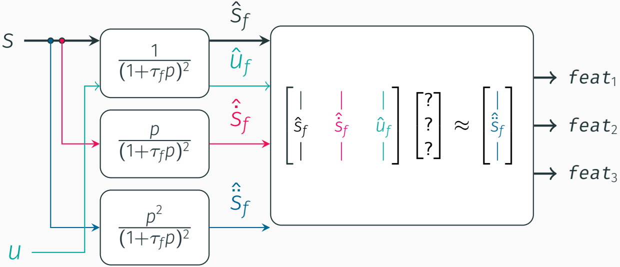
Take the simple forced oscillator:

$$\ddot{s} = -\omega^2 s - f\dot{s} + gu = \begin{bmatrix} s & \dot{s} & u \end{bmatrix} \cdot \begin{bmatrix} -\omega^2 \\ -f \\ g \end{bmatrix}$$



Healthy $(\omega, f, g) = (2.0, 0.05, 1.0)$

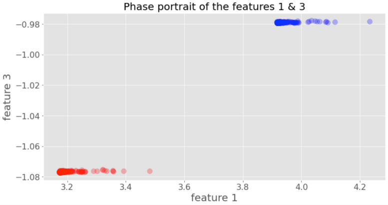
Faulty $(\omega, f, g) = (1.8, 0.08, 1.1)$



An illustrative example – (continued)

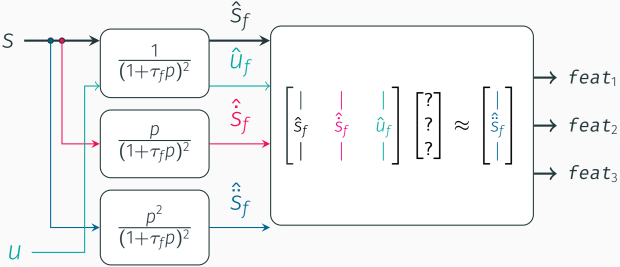
Take the simple forced oscillator:

$$\ddot{s} = -\omega^2 s - f\dot{s} + gu = \begin{bmatrix} s & \dot{s} & u \end{bmatrix} \cdot \begin{bmatrix} -\omega^2 \\ -f \\ g \end{bmatrix}$$



Healthy $(\omega, f, g) = (2.0, 0.05, 1.0)$

Faulty $(\omega, f, g) = (1.8, 0.08, 1.1)$



(Keep in mind)

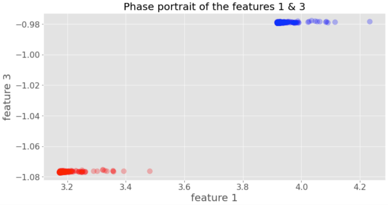
- We knew the model
- Linear model
- SISO model
- We knew the oder (=2)
- We add ad-hoc virtual sensors

- We compare to a basic features generation method

An illustrative example – (continued)

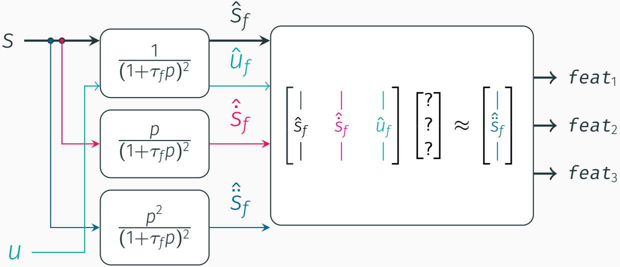
Take the simple forced oscillator:

$$\ddot{s} = -\omega^2 s - f\dot{s} + gu = \begin{bmatrix} s & \dot{s} & u \end{bmatrix} \cdot \begin{bmatrix} -\omega^2 \\ -f \\ g \end{bmatrix}$$



Healthy $(\omega, f, g) = (2.0, 0.05, 1.0)$

Faulty $(\omega, f, g) = (1.8, 0.08, 1.1)$

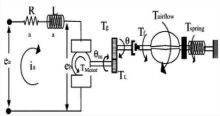


(System Invariants)

In this example, feat_1, feat_2 and feat_3 are almost invariant w.r.t the context of excitation.

While this was easy for this example, generalizing the computation of invariants to all industrial equipments is a challenging task!

Example: The throttle control unit



$$\ddot{\theta} = \frac{1}{J}\big[-K_{sp}(\theta - \theta_0) - K_f\dot{\theta} + NK_t e_a + \\ - \pi R_p^2 R_{af}\Delta_p(\theta, P_m, N)\cos^2(\theta)\big]$$

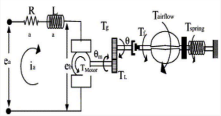
$$\dot{e}_a = \frac{1}{L_a}\left[-NK_b\dot{\theta} - R_a e_a + i_a\right]$$

$$i_a = \texttt{Feedback}(\theta, \dot{\theta}, \theta_{ref}, \dots)$$

$x := (\theta, \dot{\theta}, e_a)$	State
$p = (J, K_{sp}, K_f, R_p, R_{af}, N, R_a)$	Parameter
$c = (\theta_{ref}, P_m)$	Context
$s := (\theta, i_a, \theta_{ref})$	Sensors

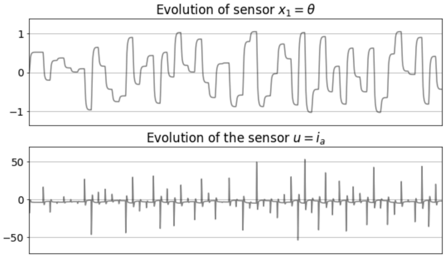
Blind normality characterization of Nonlinear MIMO equipment...

Example: The throttle control unit



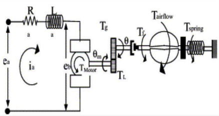
$$\ddot{\theta} = \frac{1}{J}\Big[-K_{sp}(\theta - \theta_0) - K_f\dot{\theta} + NK_t e_a +$$
$$- \pi R_p^2 R_{af}\Delta_p(\theta, P_m, N)\cos^2(\theta)\Big]$$
$$\dot{e}_a = \frac{1}{L_a}\Big[-NK_b\dot{\theta} - R_a e_a + i_a\Big]$$
$$i_a = \texttt{Feedback}(\theta, \dot{\theta}, \theta_{ref}, \dots)$$

$x := (\theta, \dot{\theta}, e_a)$	State
$p = (J, K_{sp}, K_f, R_p, R_{af}, N, R_a)$	Parameter
$c = (\theta_{ref}, P_m)$	Context
$s := (\theta, i_a, \theta_{ref})$	Sensors



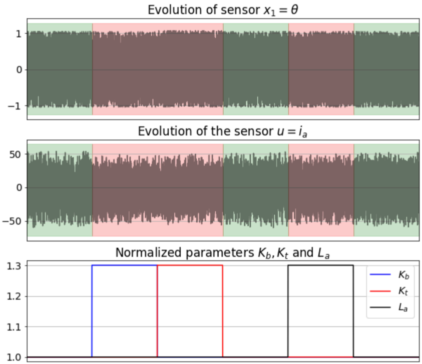
Create learning data using a sequence of changes in the reference values θ_{ref} with the health values of the parameters (zoomed view)

Example: The throttle control unit



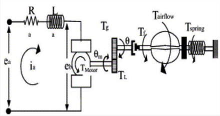
$$\ddot{\theta} = \frac{1}{J}\left[-K_{sp}(\theta - \theta_0) - K_f\dot{\theta} + NK_t e_a + \right.$$
$$\left. - \pi R_p^2 R_{af}\Delta_p(\theta, P_m, N)\cos^2(\theta)\right]$$
$$\dot{e}_a = \frac{1}{L_a}\left[-NK_b\dot{\theta} - R_a e_a + i_a\right]$$
$$i_a = \texttt{Feedback}(\theta, \dot{\theta}, \theta_{ref}, \dots)$$

$x := (\theta, \dot{\theta}, e_a)$	State
$p = (J, K_{sp}, K_f, R_p, R_{af}, N, R_a)$	Parameter
$c = (\theta_{ref}, P_m)$	Context
$s := (\theta, i_a, \theta_{ref})$	Sensors



Now using new reference profiles for θ_{ref}, check now if slight changes in the parameters K_b, K_t and L_a induces significant changes in the designed invariants.

Example: The throttle control unit

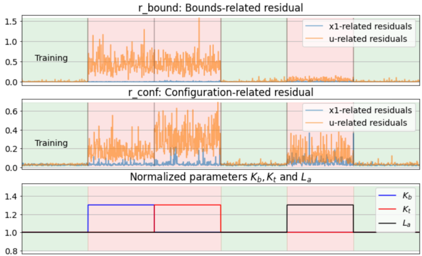


$$\ddot{\theta} = \frac{1}{J}\Big[-K_{sp}(\theta-\theta_0) - K_f\dot{\theta} + NK_t e_a + \\ - \pi R_p^2 R_{af}\Delta_p(\theta, P_m, N)\cos^2(\theta)\Big]$$

$$\dot{e}_a = \frac{1}{L_a}\big[-NK_b\dot{\theta} - R_a e_a + i_a\big]$$

$$i_a = \texttt{Feedback}(\theta, \dot{\theta}, \theta_{ref}, \dots)$$

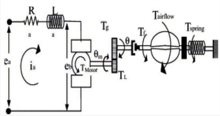
$x := (\theta, \dot{\theta}, e_a)$	State
$p = (J, K_{sp}, K_f, R_p, R_{af}, N, R_a)$	Parameter
$c = (\theta_{ref}, P_m)$	Context
$s := (\theta, i_a, \theta_{ref})$	Sensors



ETC-example of temporal behavior of the blindly constructed normality invariants.

Blind normality characterization of Nonlinear MIMO equipment...

Example: The throttle control unit



$$\ddot{\theta} = \frac{1}{J}\left[-K_{sp}(\theta-\theta_0) - K_f\dot{\theta} + NK_t e_a + \right.$$
$$\left. -\pi R_p^2 R_{af}\Delta_p(\theta, P_m, N)\cos^2(\theta)\right]$$
$$\dot{e}_a = \frac{1}{L_a}\left[-NK_b\dot{\theta} - R_a e_a + i_a\right]$$
$$i_a = \texttt{Feedback}(\theta, \dot{\theta}, \theta_{ref}, \dots)$$

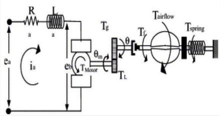
$x := (\theta, \dot{\theta}, e_a)$	State
$p = (J, K_{sp}, K_f, R_p, R_{af}, N, R_a)$	Parameter
$c = (\theta_{ref}, P_m)$	Context
$s := (\theta, i_a, \theta_{ref})$	Sensors

	Lorentz_Attractor_nominal	Automotive_ETC_nominal
SBundle	0.875081	0.951444
SBundle_BFD	0.755362	0.935937
AutoEncoder	0.508832	0.610908
DiagFit	0.533520	0.576509
NBundle	0.509420	0.594133
AutoReg_win100	0.552759	0.495965
MTADVAE	0.546898	0.497290
Autoreg_BFD	0.542559	0.495607
win100_AutoReg	0.535842	0.501373
Filter_Autoreg	0.530481	0.503557
BFD_TS_light	0.529888	0.496942
BFD_None	0.511509	0.498232
LOF_TS_fast	0.499216	0.508092
OCSVM_TS_fast	0.502821	0.499899
IF_TS_fast	0.502562	0.498890
BFD_TS_fast	0.499680	0.501065

Benchmark using different approaches including auto-encoder DNNs and some signal processing approches.

Blind normality characterization of Nonlinear MIMO equipment...

Example: The throttle control unit

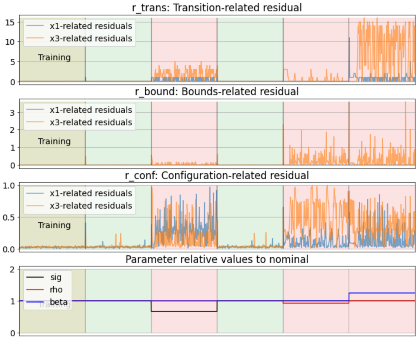


$$\ddot{\theta} = \frac{1}{J}\big[-K_{sp}(\theta - \theta_0) - K_f\dot{\theta} + NK_t e_a + \\ - \pi R_p^2 R_{af}\Delta_p(\theta, P_m, N)\cos^2(\theta)\big]$$

$$\dot{e}_a = \frac{1}{L_a}\big[-NK_b\dot{\theta} - R_a e_a + i_a\big]$$

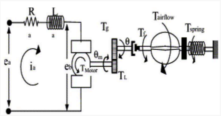
$$i_a = \texttt{Feedback}(\theta, \dot{\theta}, \theta_{ref}, \dots)$$

$x := (\theta, \dot{\theta}, e_a)$	State
$p = (J, K_{sp}, K_f, R_p, R_{af}, N, R_a)$	Parameter
$c = (\theta_{ref}, P_m)$	Context
$s := (\theta, i_a, \theta_{ref})$	Sensors



Lorentz-example of temporal behavior of the blindly constructed normality invariants.

Example: The throttle control unit

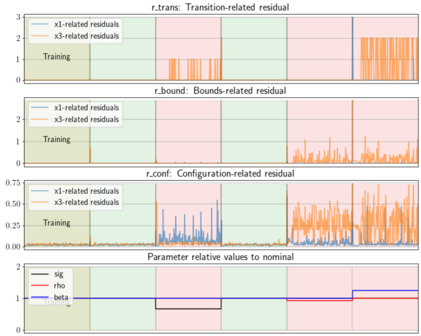


$$\ddot{\theta} = \frac{1}{J}\Big[-K_{sp}(\theta - \theta_0) - K_f\dot{\theta} + NK_t e_a + \\ - \pi R_p^2 R_{af}\Delta_p(\theta, P_m, N)\cos^2(\theta)\Big]$$

$$\dot{e}_a = \frac{1}{L_a}\left[-NK_b\dot{\theta} - R_a e_a + i_a\right]$$

$$i_a = \text{Feedback}(\theta, \dot{\theta}, \theta_{ref}, \dots)$$

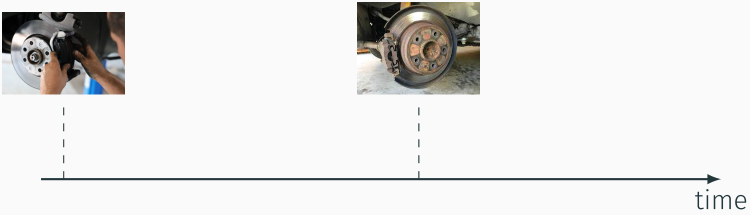
$x := (\theta, \dot{\theta}, e_a)$	State
$p = (J, K_{sp}, K_f, R_p, R_{af}, N, R_a)$	Parameter
$c = (\theta_{ref}, P_m)$	Context
$s := (\theta, i_a, \theta_{ref})$	Sensors



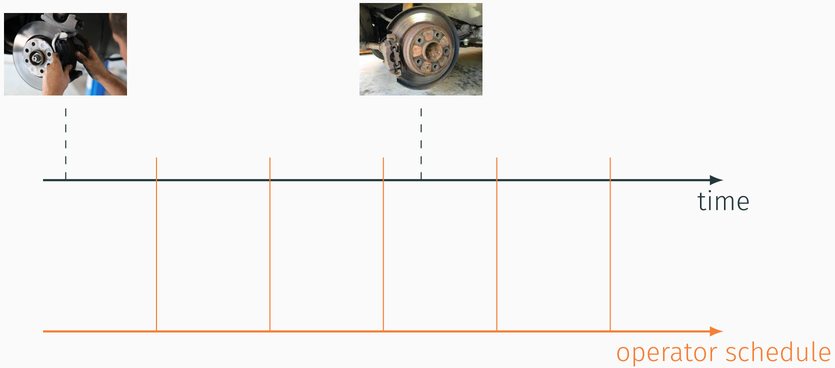
Lorentz-example of temporal behavior of the blindly constructed normality invariants.

General comments

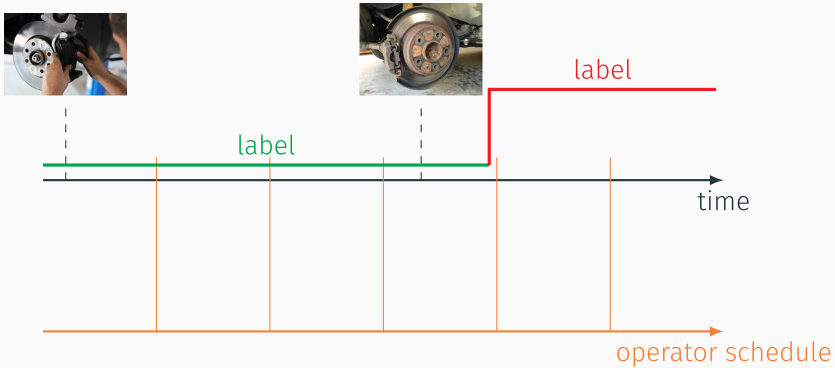
Labelling process in industry



Labelling process in industry



Labelling process in industry



Difficulties of evaluation in industrial context

Example1

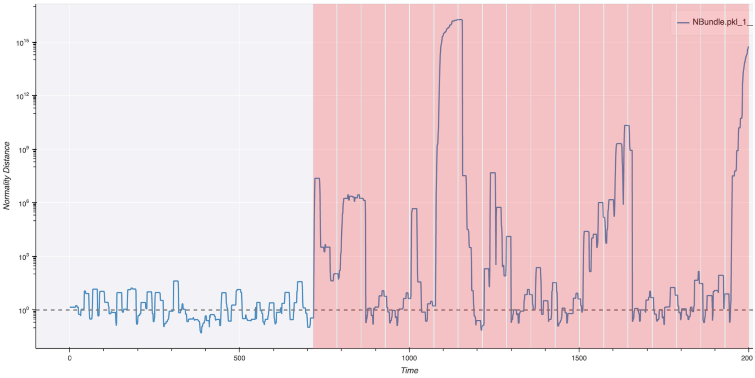


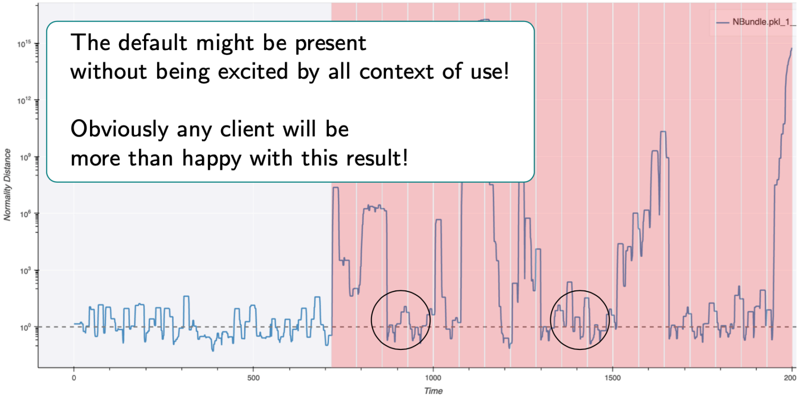
Table results

Sorted by pAUC

	Model	Version	Sensor	F1-Score	Balanced Acc	pAUC	Mean_MCC	AUC	AP	Time I
0	NBundle.pkl	1	__All	nan	nan	0.678151	nan	0.713690	0.832172	

Difficulties of evaluation in industrial context

Example1



The default might be present
without being excited by all context of use!

Obviously any client will be
more than happy with this result!

Table results

Sorted by pAUC

	Model	Version	Sensor	F1-Score	Balanced Acc	pAUC	Mean_MCC	AUC	AP	Time I
0	NBundle.pkl	1	__All	nan	nan	0.678151	nan	0.713690	0.832172	

Example2

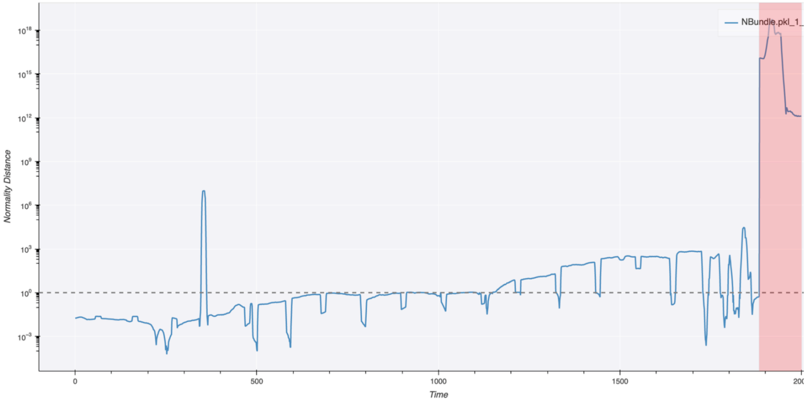


Table results

Sorted by pAUC

	Model	Version	Sensor	F1-Score	Balanced Acc	pAUC	Mean_MCC	AUC	AP	Time
0	NBundle.pkl	1	__All	nan	nan	0.996961	nan	0.998928	0.980854	

Difficulties of evaluation in industrial context

Example2

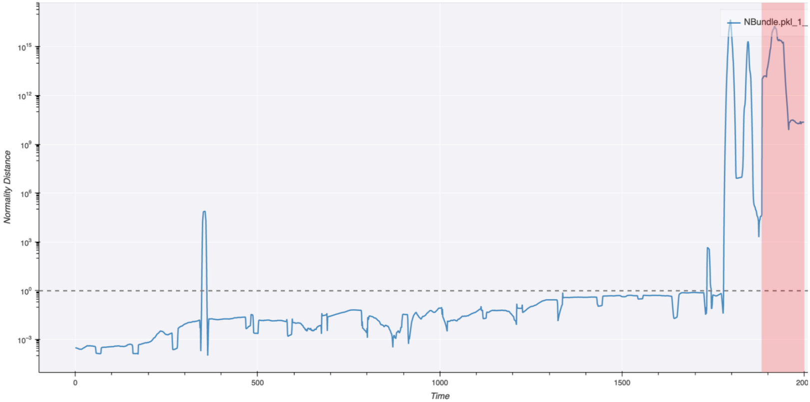


Table results

Sorted by pAUC

	Model	Version	Sensor	F1-Score	Balanced Acc	pAUC	Mean_MCC	AUC	AP	Time I
0	NBundle.pkl	1	__All	nan	nan	0.909475	nan	0.982314	0.629979	

Difficulties of evaluation in industrial context

Example2

A good model that anticipates the default is penalized!

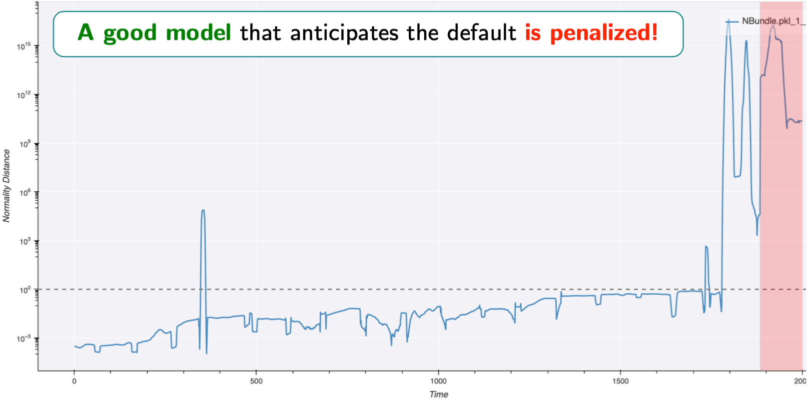


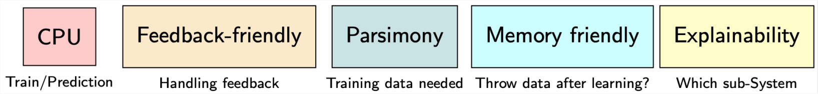
Table results

Sorted by pAUC

	Model	Version	Sensor	F1-Score	Balanced Acc	pAUC	Mean_MCC	AUC	AP	Time
0	NBundle.pkl	1	__All	nan	nan	0.909475	nan	0.982314	0.629979	

Difficulties of evaluation in industrial context

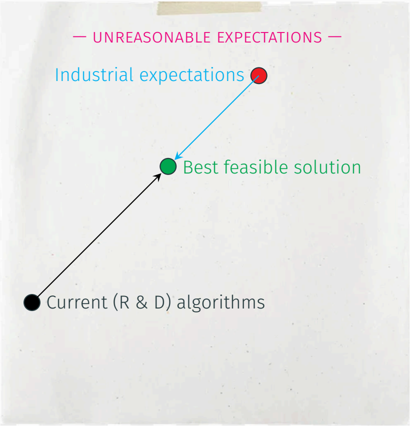
Many real-life concerns need to be accommodated for!



Conclusion


— UNREASONABLE EXPECTATIONS —
Industrial expectations
Best feasible solution
Current (R & D) algorithms

Conclusion


— UNREASONABLE EXPECTATIONS —
Industrial expectations
Best feasible solution
Current (R & D) algorithms

Conclusion


Thank you!