POLITECNICO
DI TORINO

Center for
Automotive Research
and Sustainable Mobility

AI for damage sensing in composite structures.

Alberto Ciampaglia
PhD Student
Giovanni Belingardi
Professor

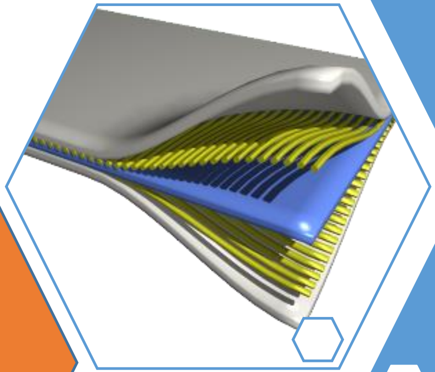
Research motivation

Load data
Improve design

No driver FB
ODM
Failure predict

Data availability

Autonomous driving



Smart design

Failure modes
High Sfactor
In-situ sensors

Composite materials

Research motivation

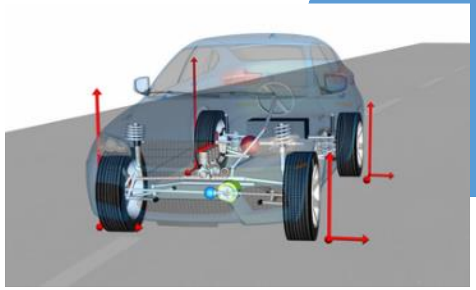
LOADS
Amplitude
Freqeuncy
Spectrum

Model

K&C
Suspension kinematic
Elements Stiffness
Tire behaviour

Assumption

DRIVER
Avarege speed
Curve acceleration



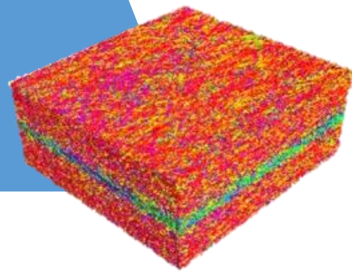
PROPERTIES
Stiffness
Strenght

Model

STRUCTURE
Fiber orientations
Local fiber density
Cohesive behaviour

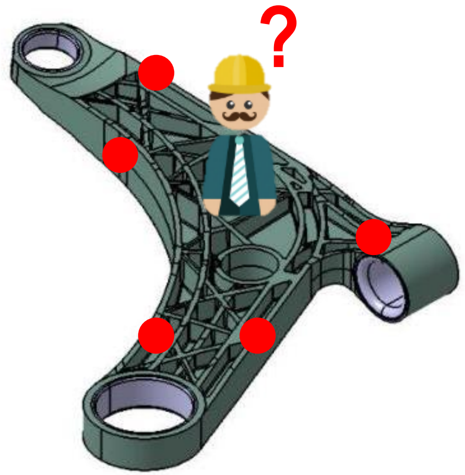
Model

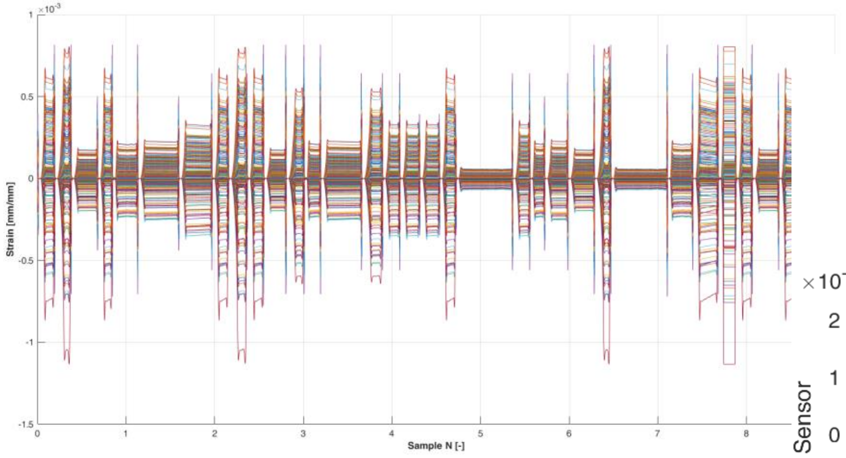
PROCESS
Injection point
Mold temperature
Rehology

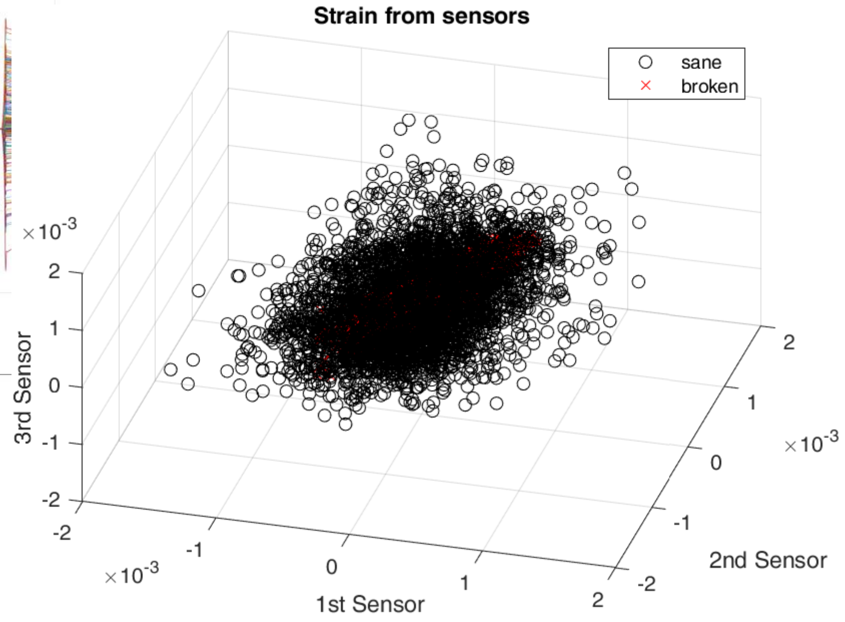


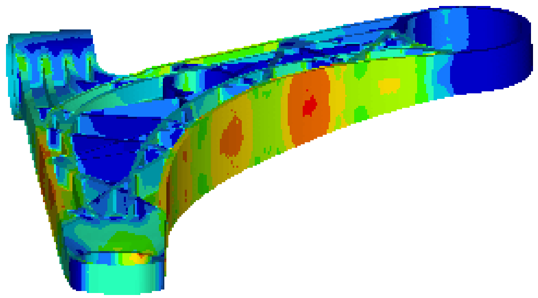
How to deal with data?

SENSOR

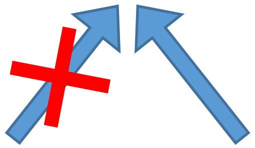








STRAIN MAP



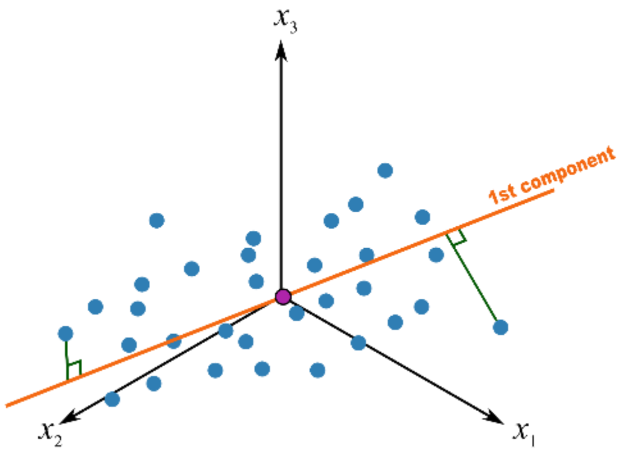
LOADS

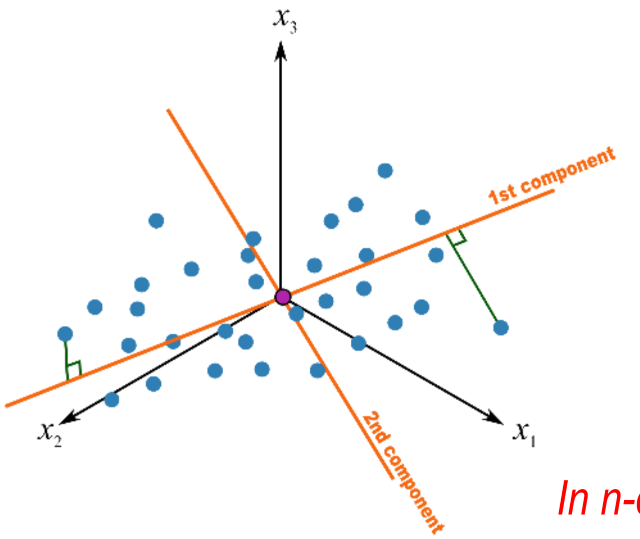
STRUCTURE

<u>How to remove the effect of loads on strains?</u>

Principal Component Analisys (PCA)

Principal Component Analysis (PCA)





In n-dimensions!

Given our dataset

$$X = [\,\{\varepsilon\}_{sens}^{t1}\{\varepsilon\}_{sens}^{t2}\ \dots\{\varepsilon\}_{sens}^{tN}]^T$$

With

$$\{\varepsilon\}_{sens}^{ti}\text{: strains from sensors at time ti}$$

From the co-variance matrix:

$$C_X = \frac{1}{N-1}X^TX$$

We compute the coefficients matrix P so that:

$$C_XP = P\Lambda$$

The principal component will be:

$$T = XP \quad \in R^{NxL}$$

PCA and loads

Given the strain acquisition at time t_i:

$$\{\varepsilon\}_{sens}^{t_i} = g([b][K]^{-1}\{F(t_i)\})$$

With

g(): mapping function

[b]: kinematic matrix

[K]: stiffness matrix

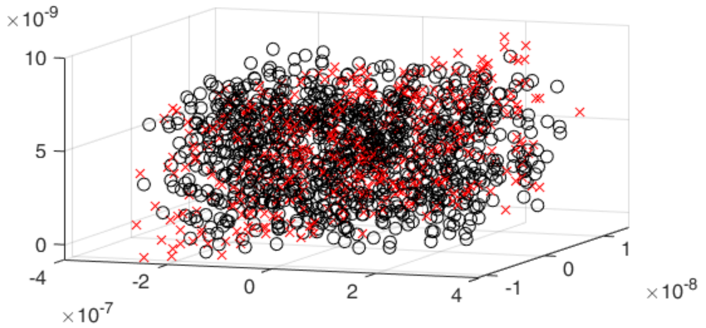
{F(t)}: load vector

The variation of $\{\varepsilon\}_{sens}^{t}$ in time is dominated by {F}

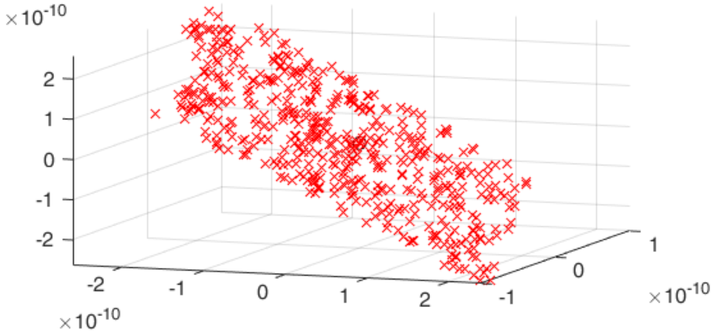
By dropping the first components we somehow remove the influence of loads on strains !

Broken
Sane

Component 1-3



Component 4-6



Support Vector Machine for One Class classification

Given a training set :

$$S = \{s_i^k\}_{i=1}^{N^k}$$

The OC SVM is formulated as:

$$\min_{\theta} \frac{1}{2} \|\theta\|^2 + \frac{1}{\nu N^k} \sum_{i=1}^{N^k} \xi_i - \rho$$

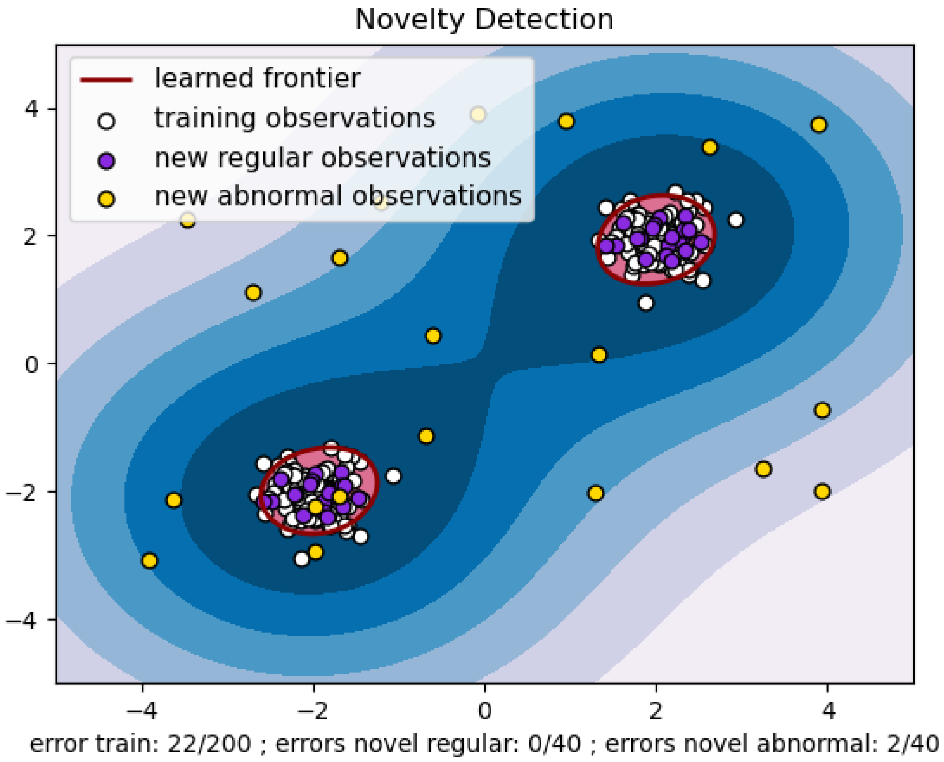
$$s.t. \quad \theta^t \phi\left(s_i{}^k\right) \geq \rho - \xi_i \,, \qquad \xi_i \geq 0 \,, i = 1, \ldots, n$$

where θ is the learned weight vector, ρ is the offset, $\phi(\cdot)$ is the feature map.

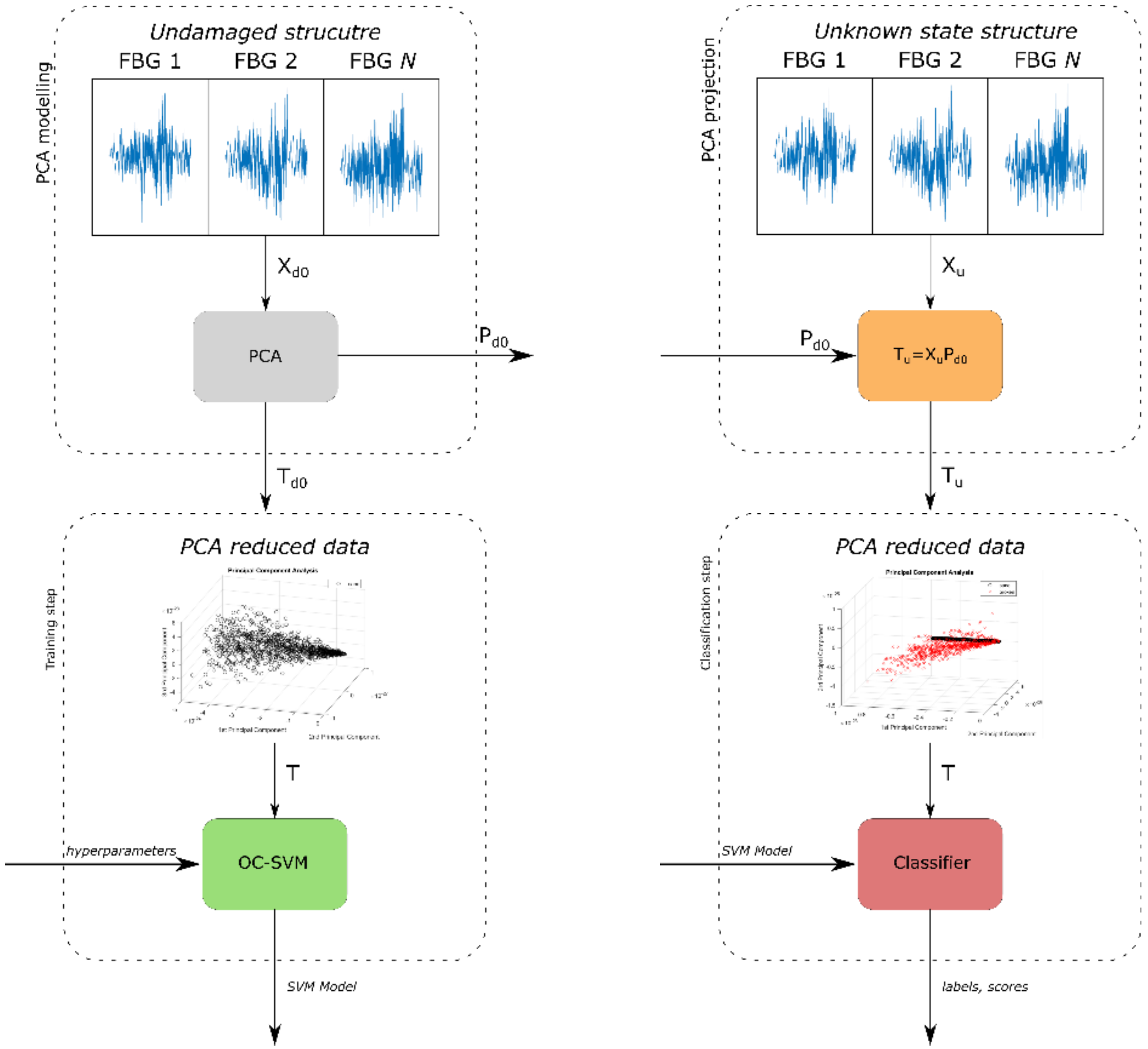
$$k(s_i^k, s_i^k) = e^{\frac{-\|(s_i^k - s_j^k\|^2}{2\sigma^2}}$$

The outlier score is then computed as:

$$A\left(s_t{}^k\right) = \theta^t \phi\left(s_t{}^k\right) - \rho$$



Pipeline

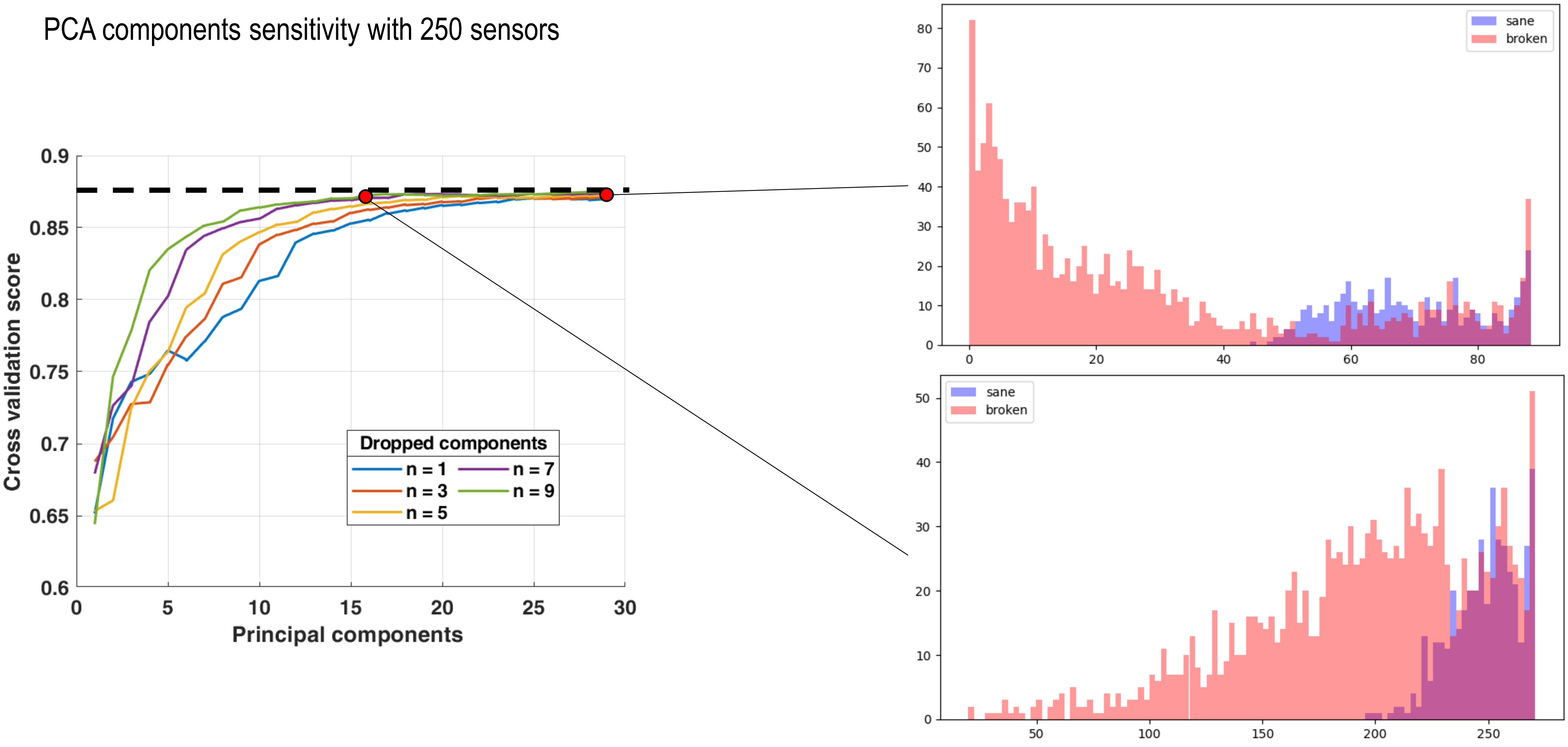


PIPELINE MAIN STEPS

1. **PCA modelling:** compute principal reference system.
2. **Training step:** given the hyperparameter, the OC-SVM learn a hypersphere.
   - *Components (drop_comp, n_comp)*
   - *Train %*
   - *Kernel function, Kernel parameters*
3. **PCA projection:** data from the monitored structure are projected on the principal reference system
4. **Classification step:** the classifier compute scores for the binary classification.

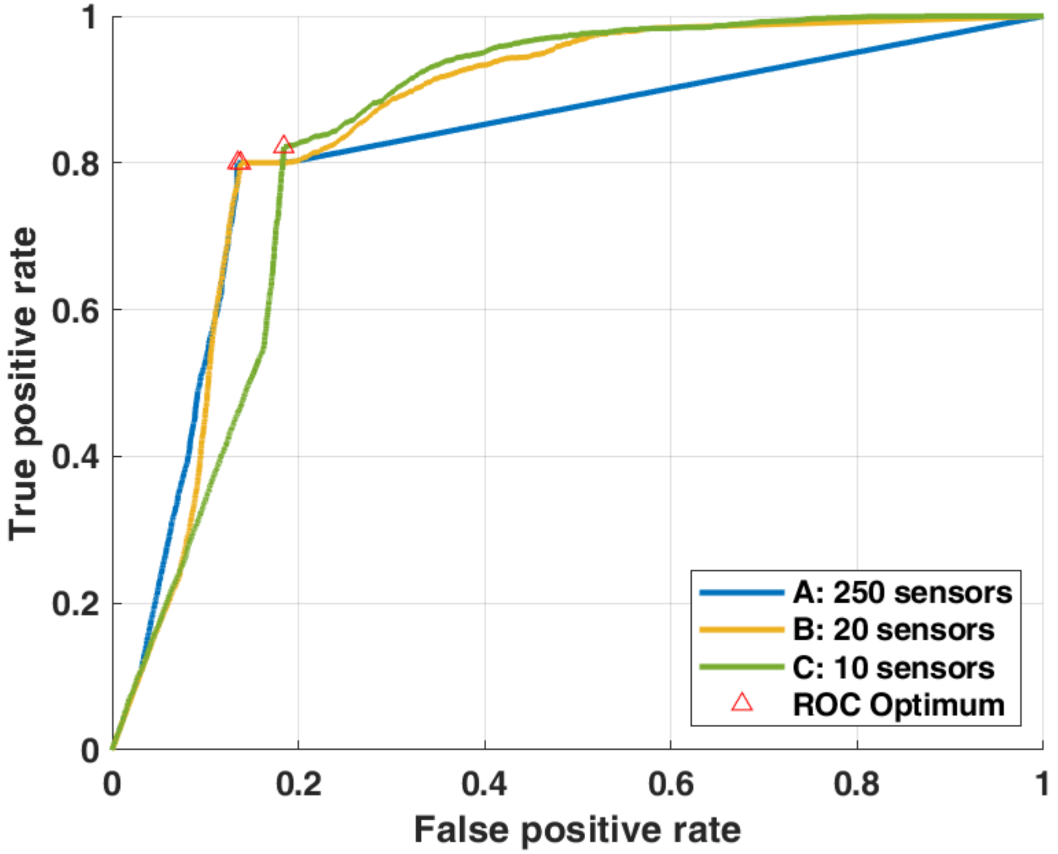
Results

PCA components sensitivity with 250 sensors

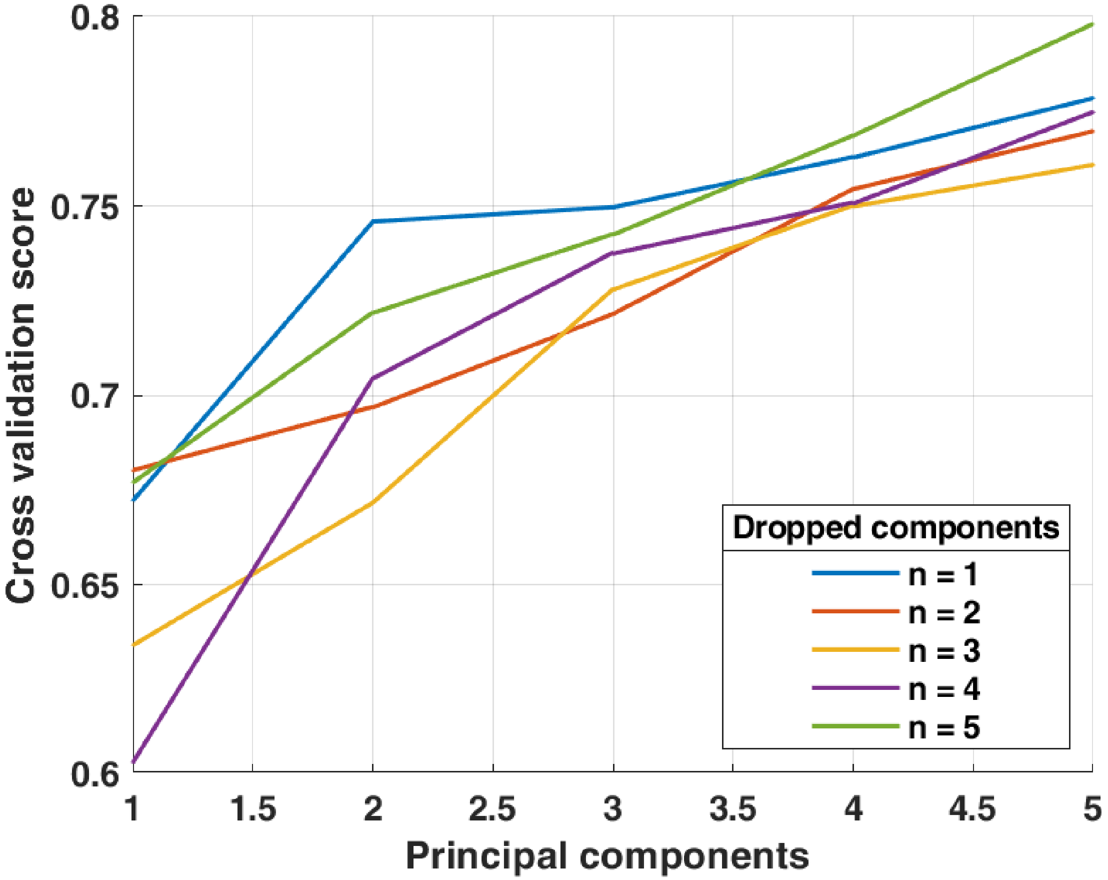


Results

Sensors number sensitivity

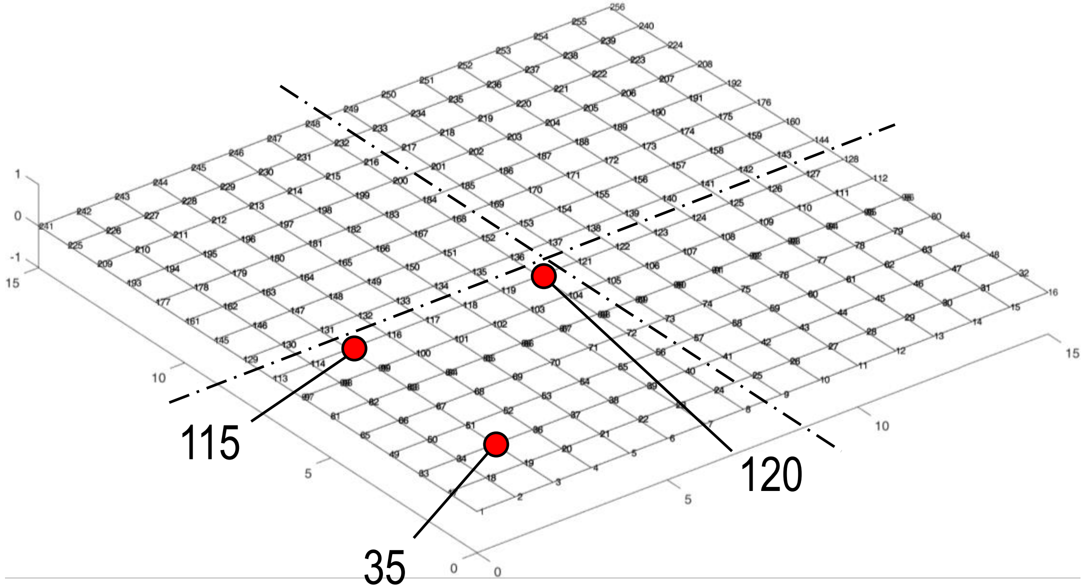


Configuration C



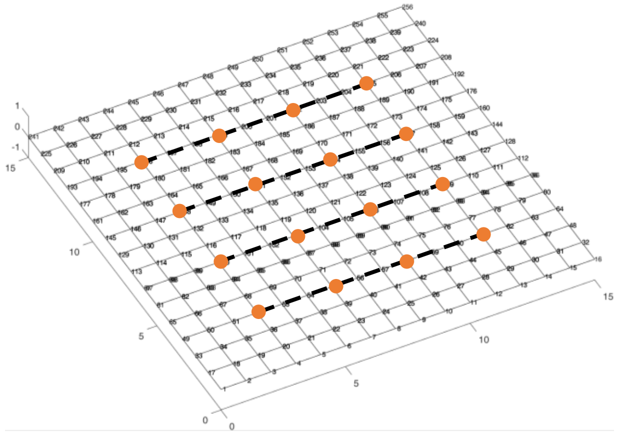
Results – test case

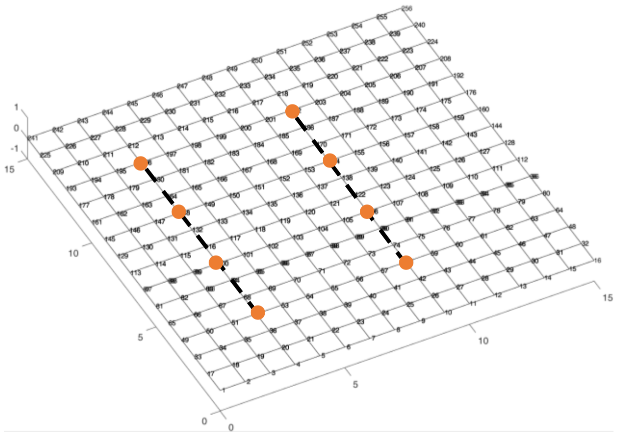
Damage position

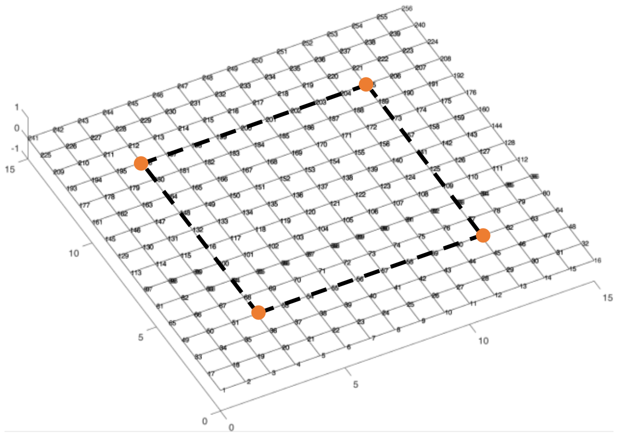


Position / Damage	35	115	120
20%			
30%			
40%			

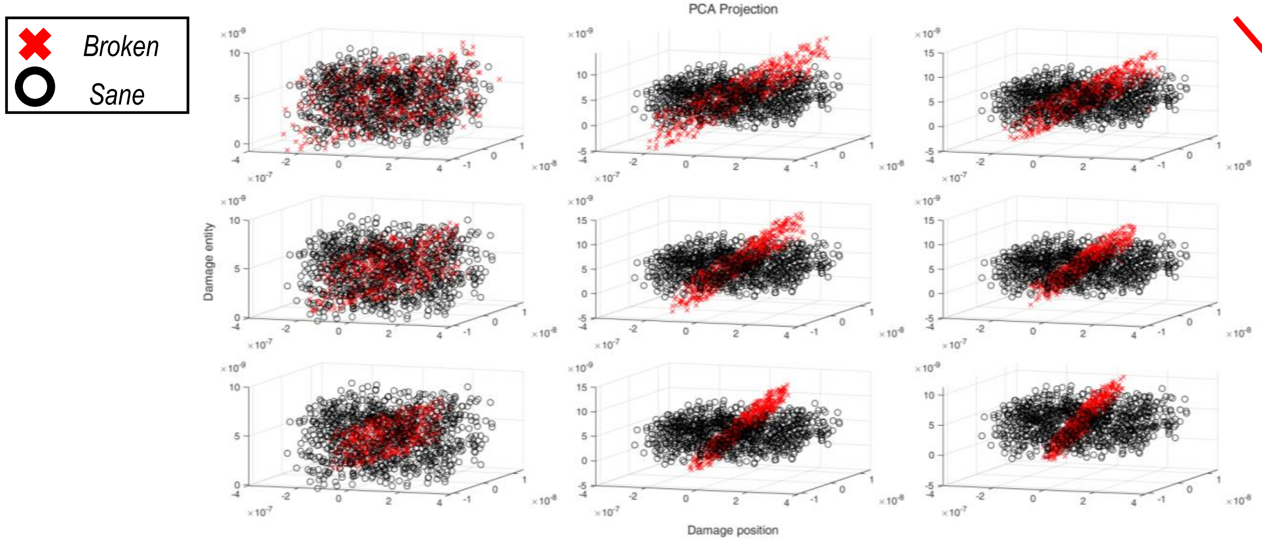
Sensors position



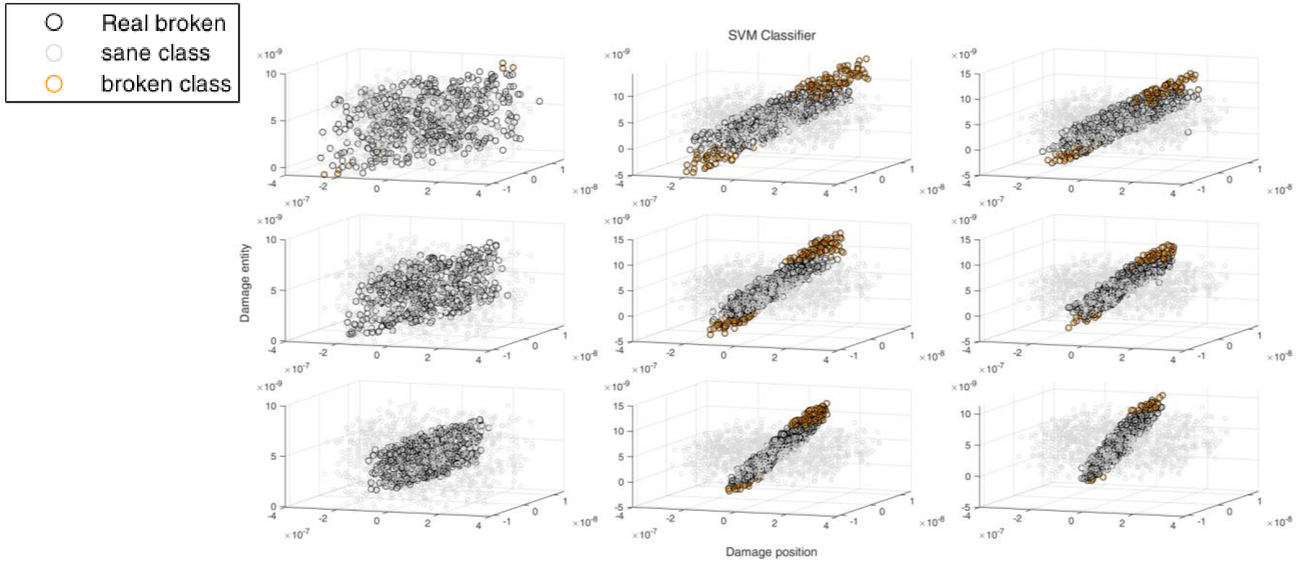


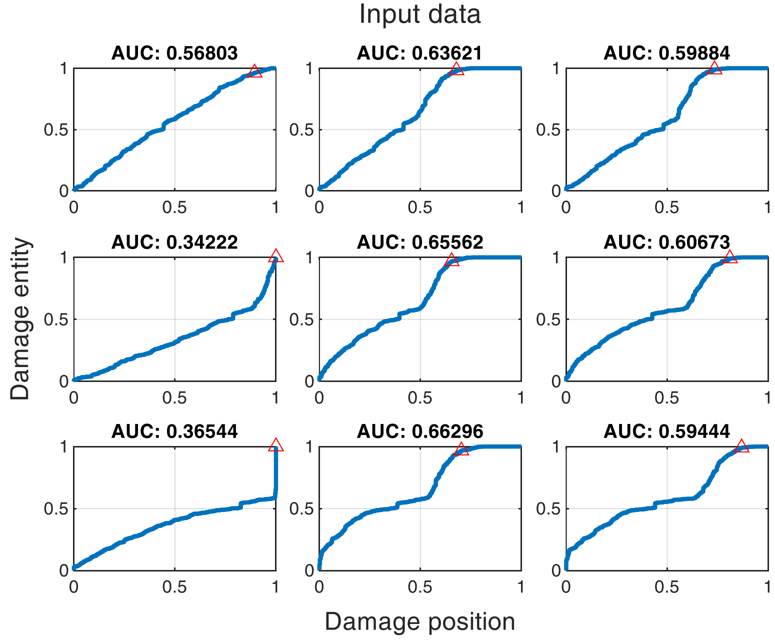


Results – test case

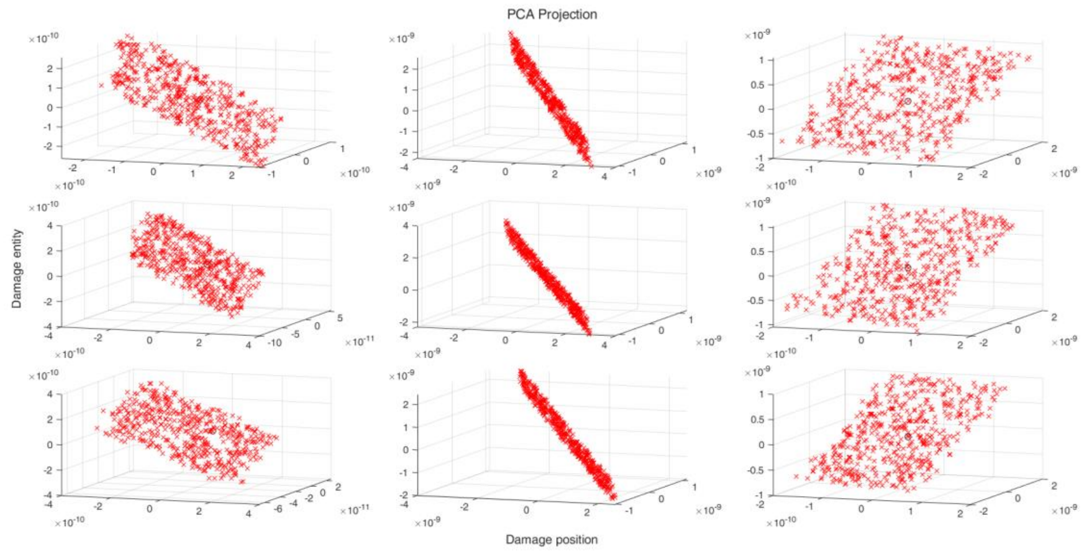


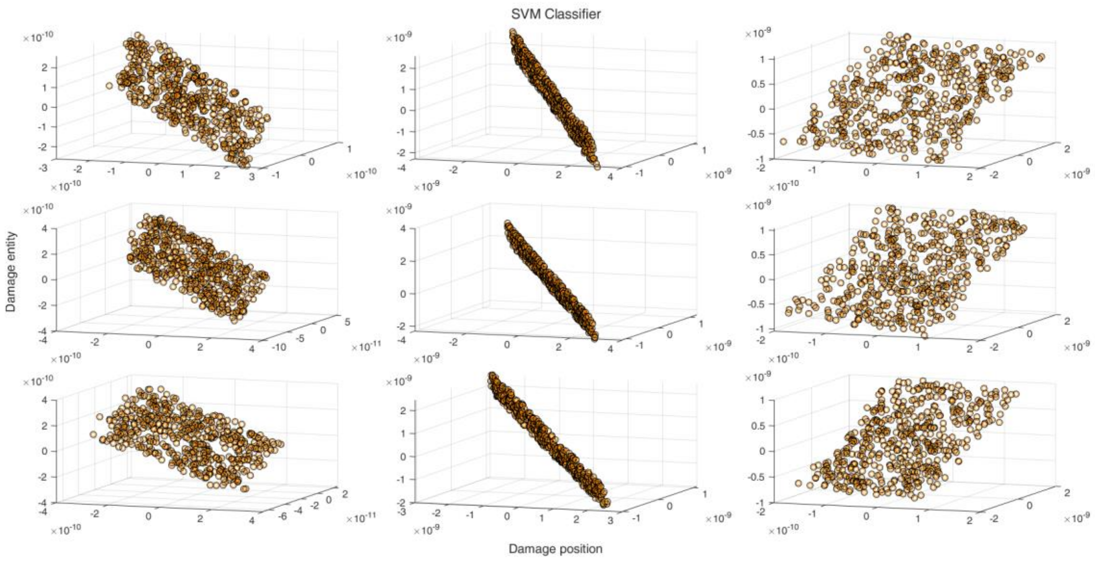
N. comp.	Explained variance[%]	
	Undamaged	*Damaged*
1	99.8	99.7
2	0.17	0.17
3	0.02	0.03
4	1e-25	0.0015



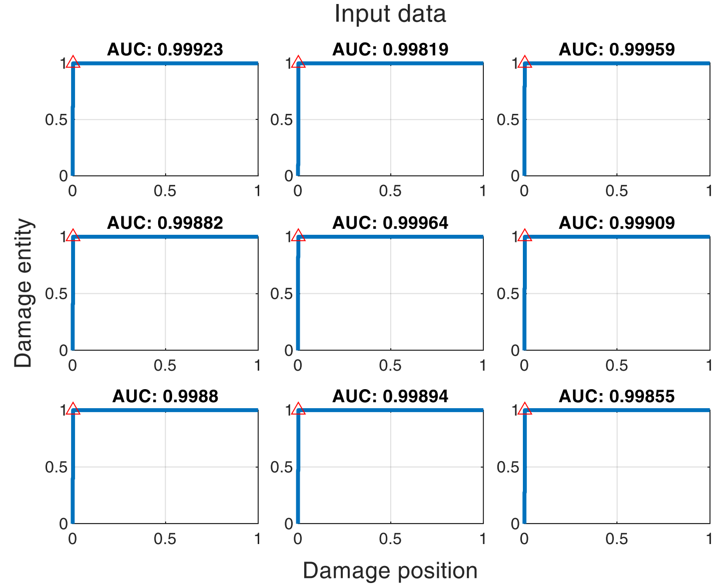


Results – test case





N. comp.	Explained variance[%]	
	Undamaged	*Damaged*
1	99.8	99.7
2	0.17	0.17
3	0.02	0.03
4	1e-25	0.0015
5	1e-25	3e-5
6	1e-25	1e-5



Future development

- K-means cluster on sensors time-history to define the best sensor configuration
- Convolutional Neural Networks on correlation maps
- Experimental tests

Thank you for your attention

Alberto Ciampaglia

PhD student

alberto.ciampaglia@polito.it