CHANGE AND ANOMALY DETECTION IN IMAGES, SIGNALS AND DATASTREAMS

Giacomo Boracchi,

Politecnico di Milano, DEIB.

https://boracchi.faculty.polimi.it/

Diego Carrera,

System Research and Applications, STMicroelectronics, Agrate Brianza

ICPR 2020, January 10, 2021

GIACOMO BORACCHI

Mathematician (Università Statale degli Studi di Milano 2004),

PhD in Information Technology (DEIB, Politecnico di Milano 2008)

Associate Professor since 2019 at DEIB (Computer Science), Polimi

Research Interests are mathematical and statistical methods for:

- Image / Signal analysis and processing
- Unsupervised learning, change / anomaly detection

DIEGO CARRERA

Mathematician (Università Statale degli Studi di Milano 2013),

PhD in Information Technology (DEIB, Politecnico di Milano 2019)

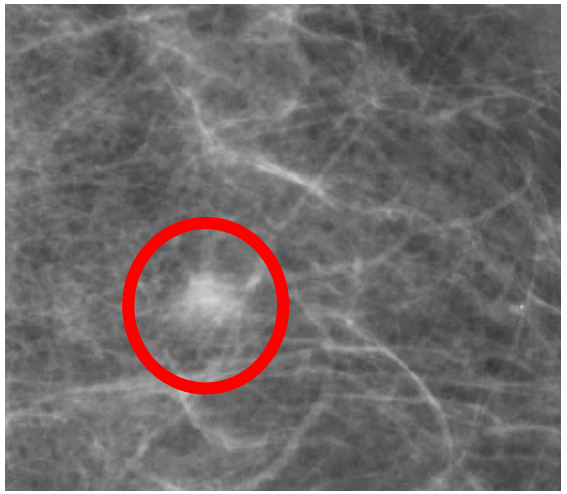
Researcher at STMicroelectronics since 2019

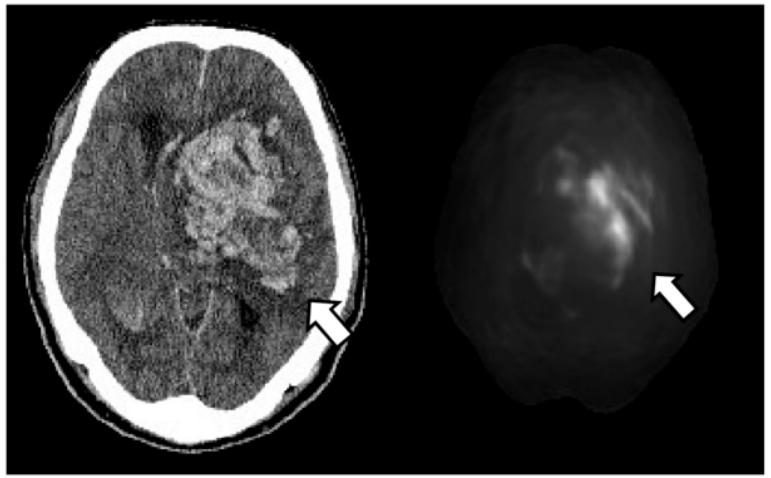
Research Interests are mainly focused on:

- Change detection in high dimensional datastreams
- Anomaly detection in signal and images
- Unsupervised learning algorithms

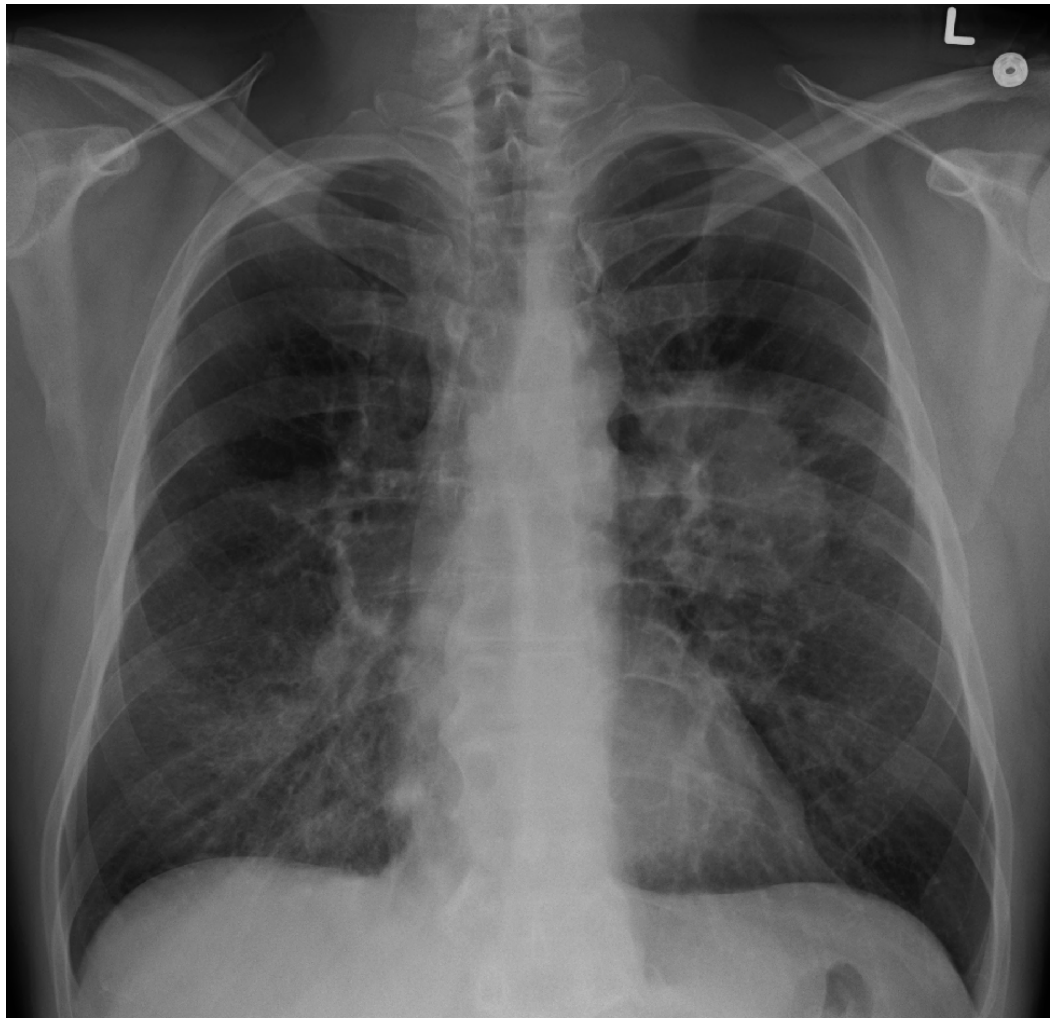
ANOMALY DETECTION IN HEALTH

Mammograms



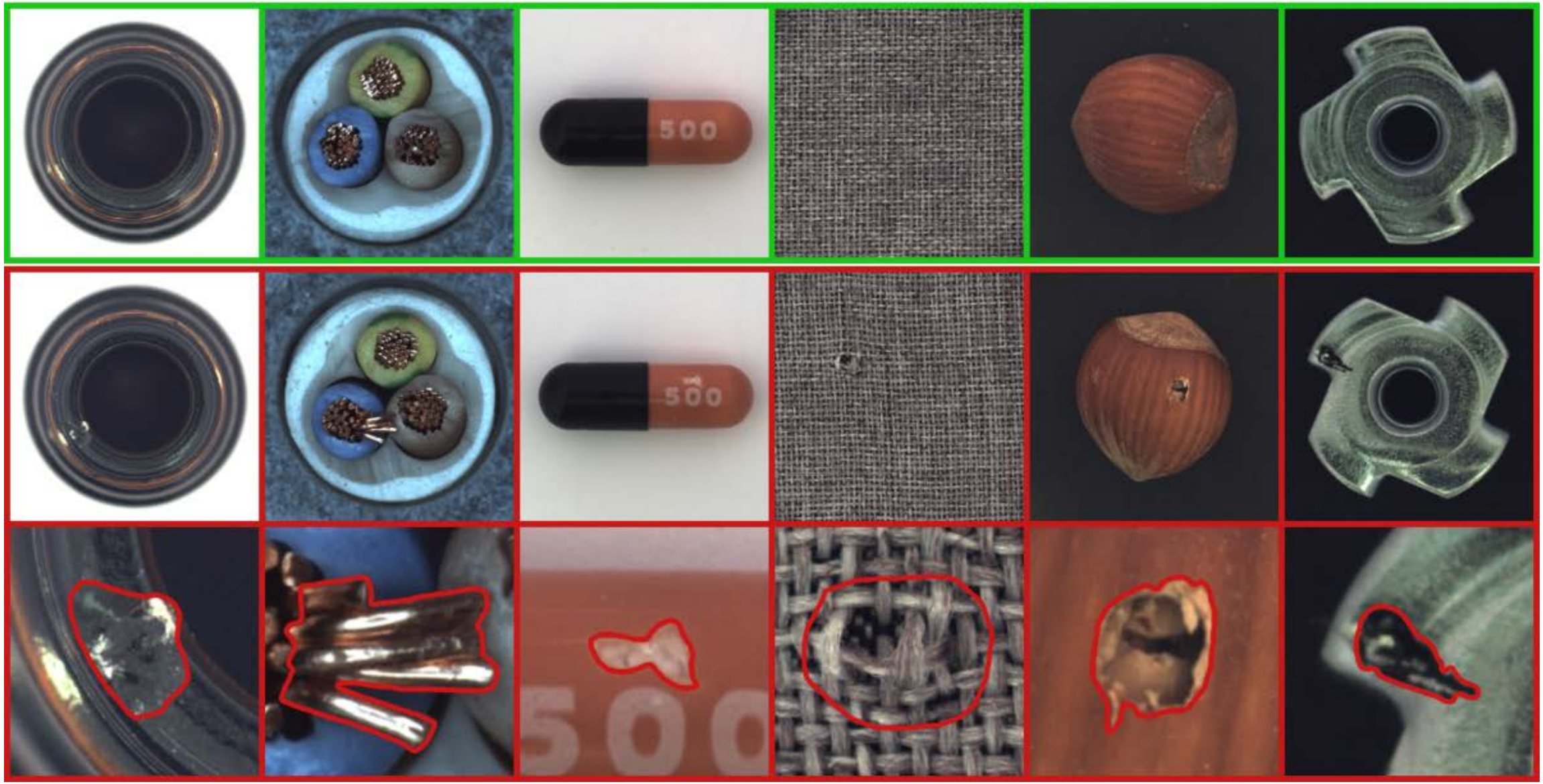


Sato et al, A primitive study on unsupervised anomaly detection with an autoencoder in emergency head CT volumes, SPIE Medical Imaging, 2018



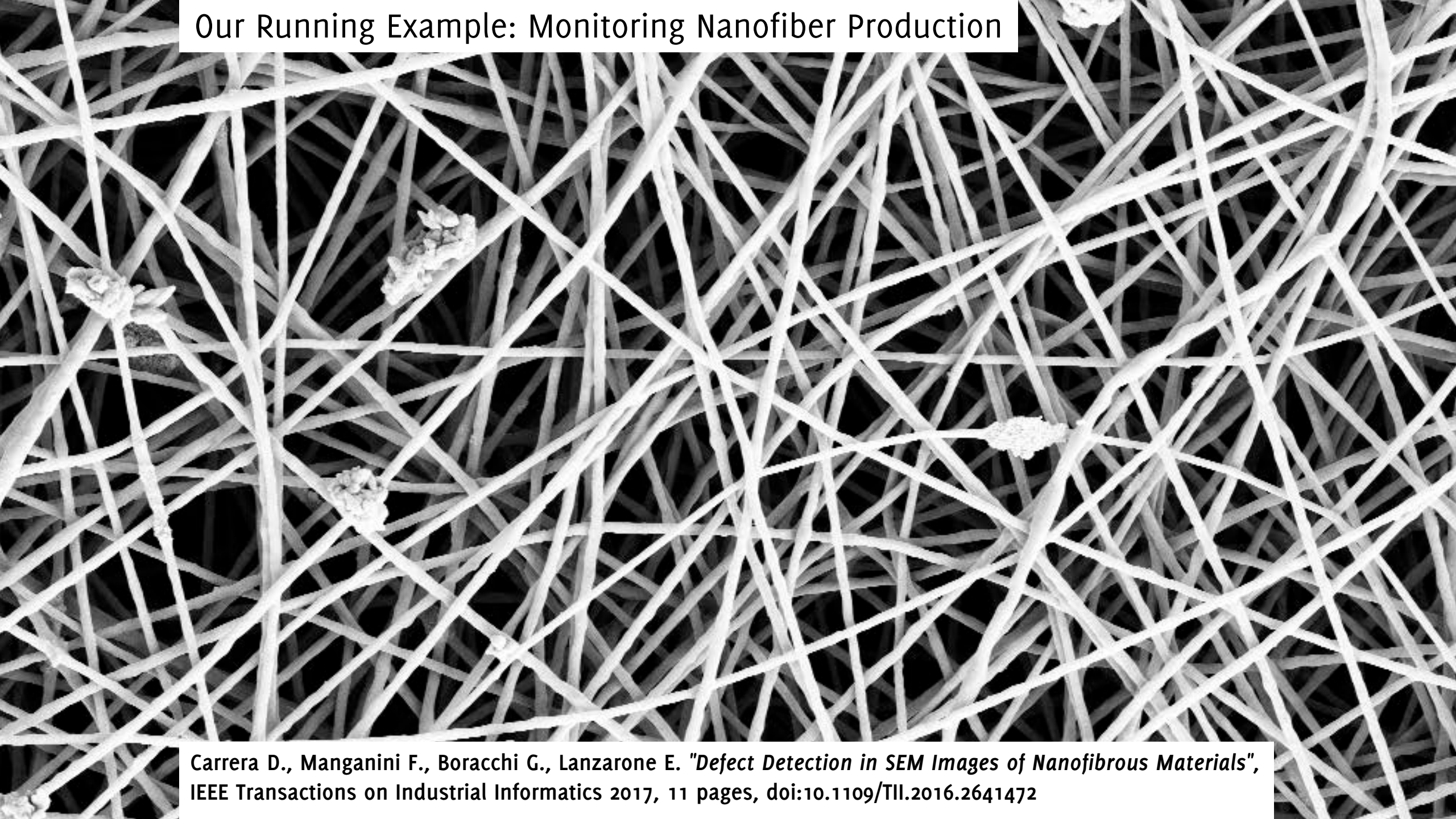
James Heilman, MD / CC BY-SA
(https://creativecommons.org/licenses/by-sa/4.0)

ANOMALY DETECTION FOR AUTOMATIC QUALITY CONTROL



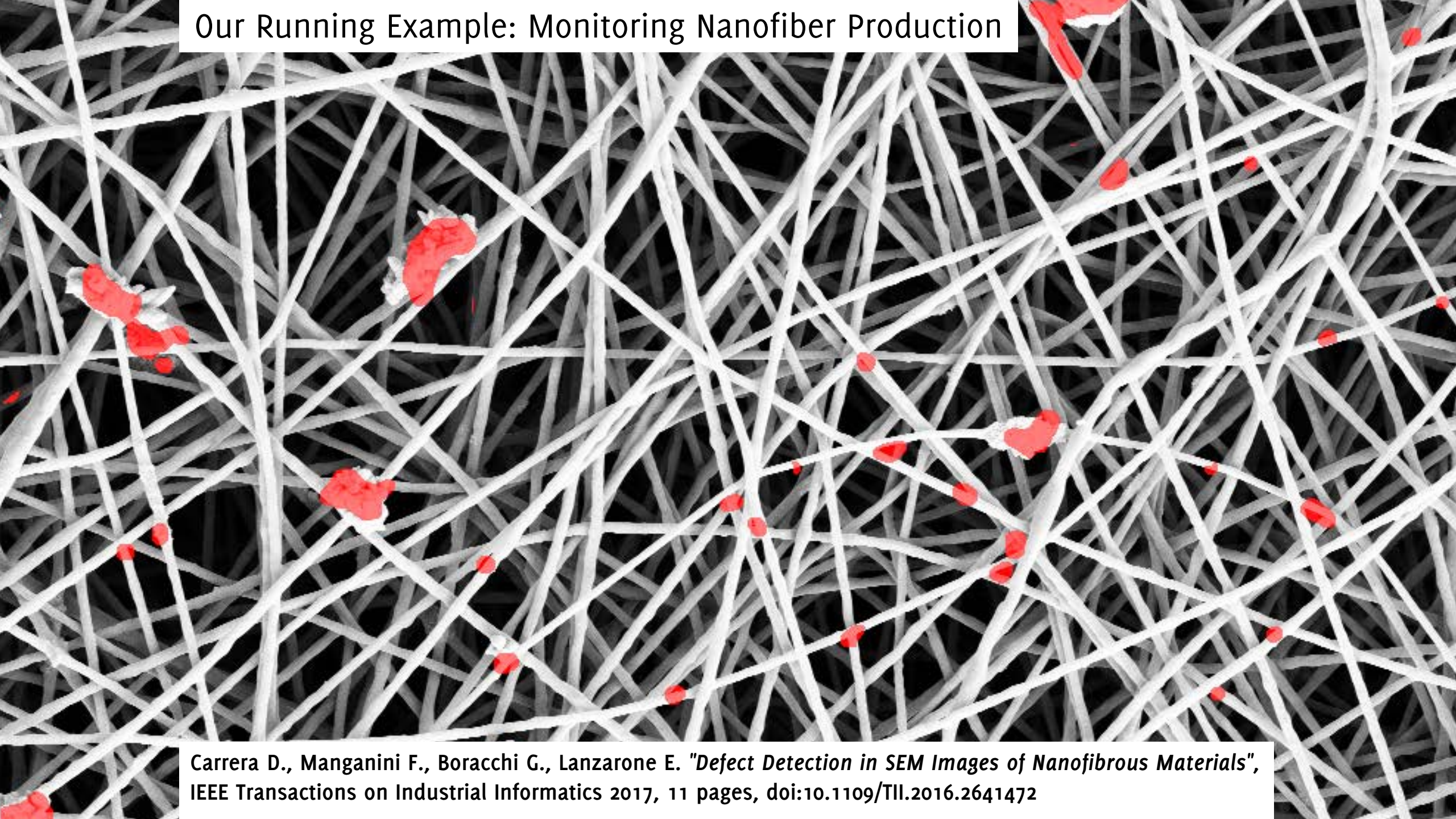
https://www.mvtec.com/company/research/datasets/mvtec-ad/

Our Running Example: Monitoring Nanofiber Production



Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

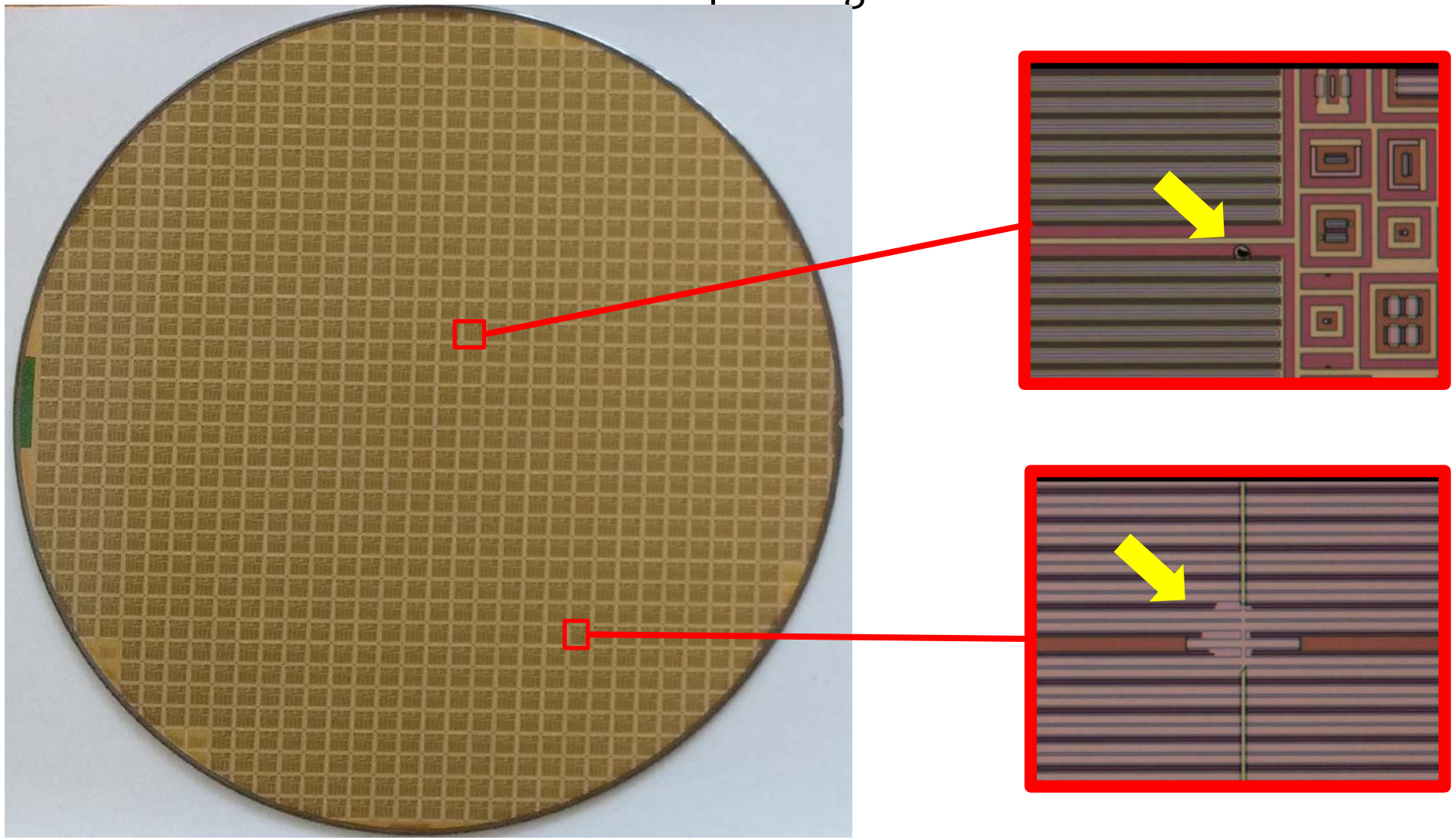
Our Running Example: Monitoring Nanofiber Production



Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

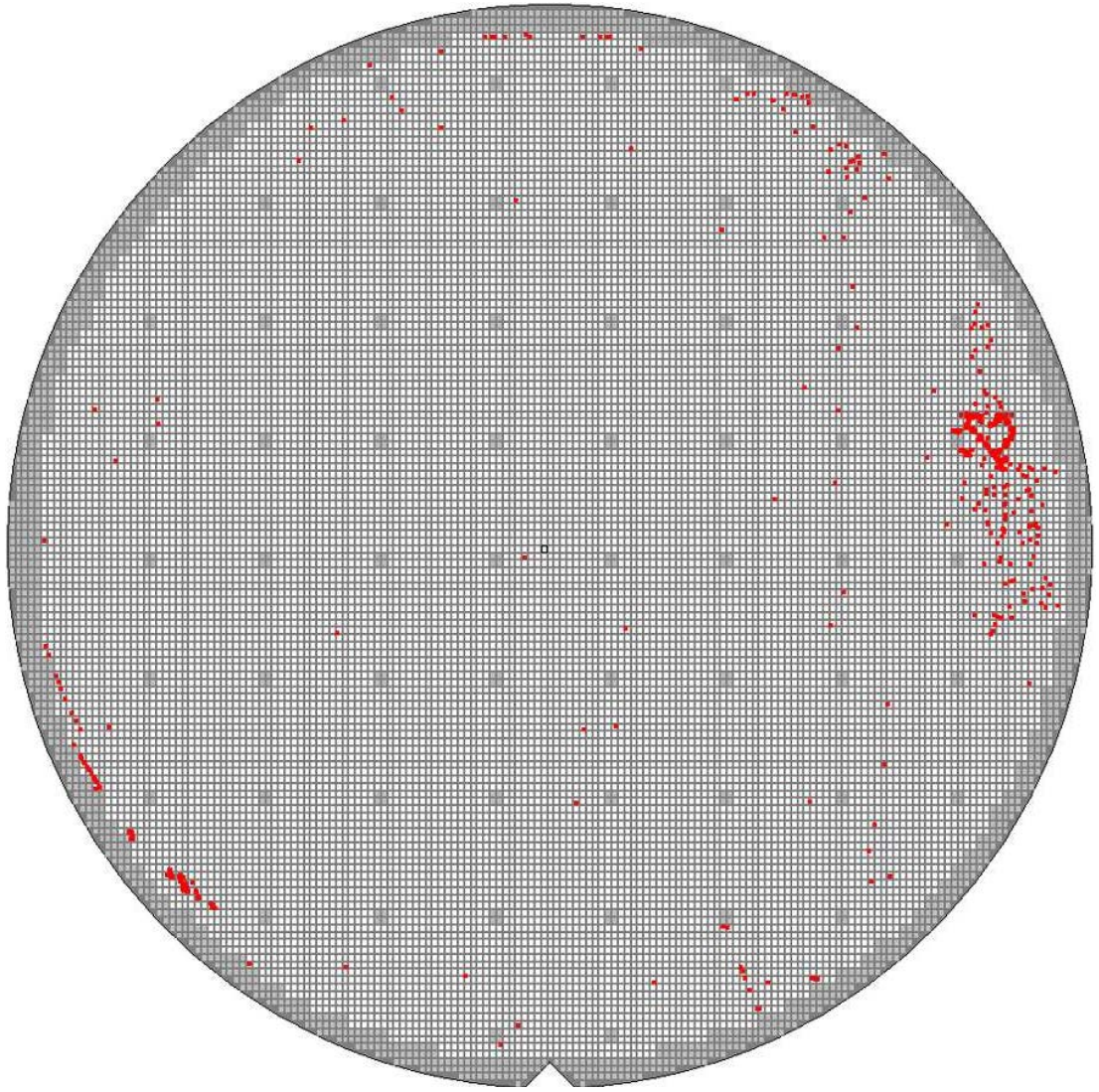
SYLICON WAFER MANUFACTURING

Defects detected as anomalies in microscope images



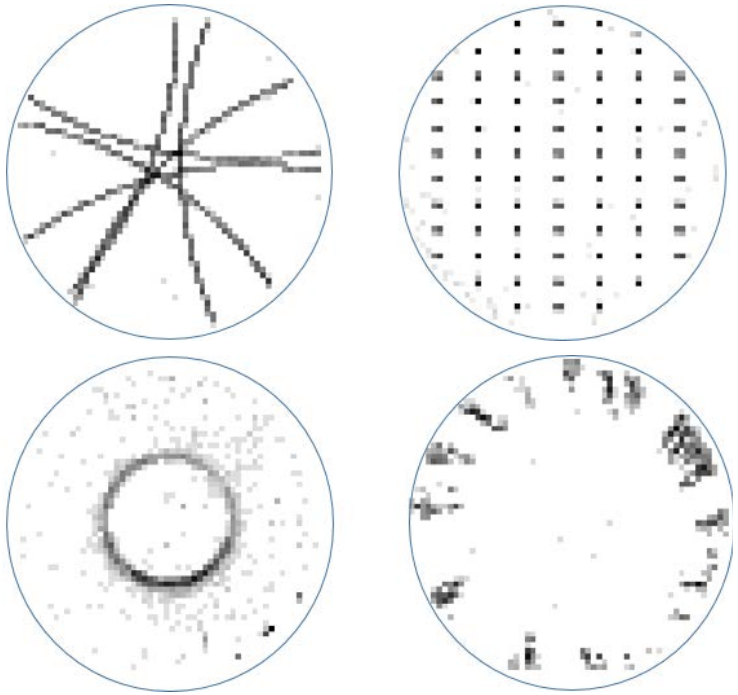
Boracchi, Carrera, ICPR 2020

DETECTION OF ANOMALOUS PATTERNS



Detect/Identify **patterns** in **wafer defect maps**

These might indicate faults, problems or malfunctioning in the chip production.

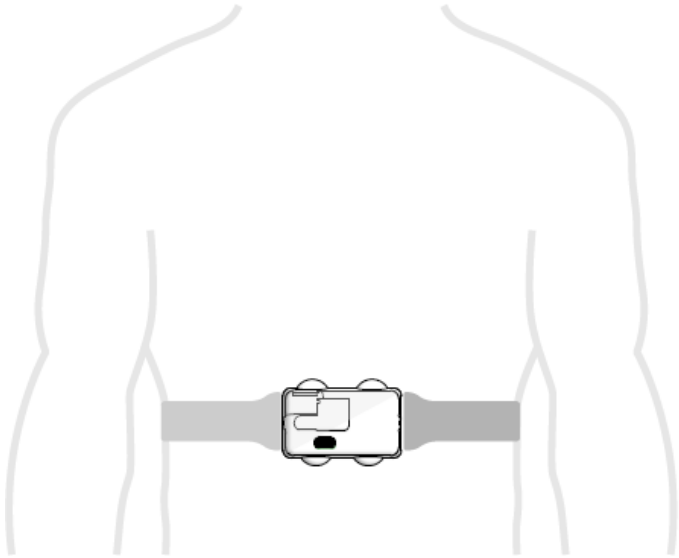


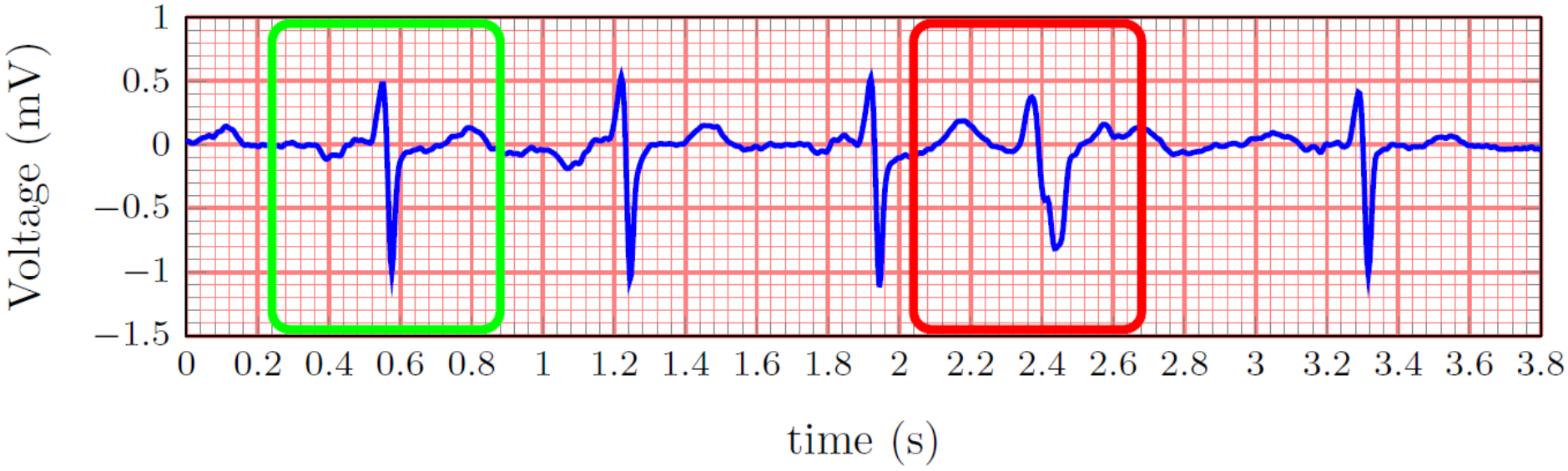
Di Bella, Carrera, Rossi, Fragneto, Boracchi *Wafer Defect Map Classification Using Sparse Convolutional Neural Networks* ICIAP09

AUTOMATIC AND LONG TERM EGC MONITORING

Health monitoring / wearable devices:

Automatically analyze EGC tracings to detect arrhythmias or incorrect device positioning



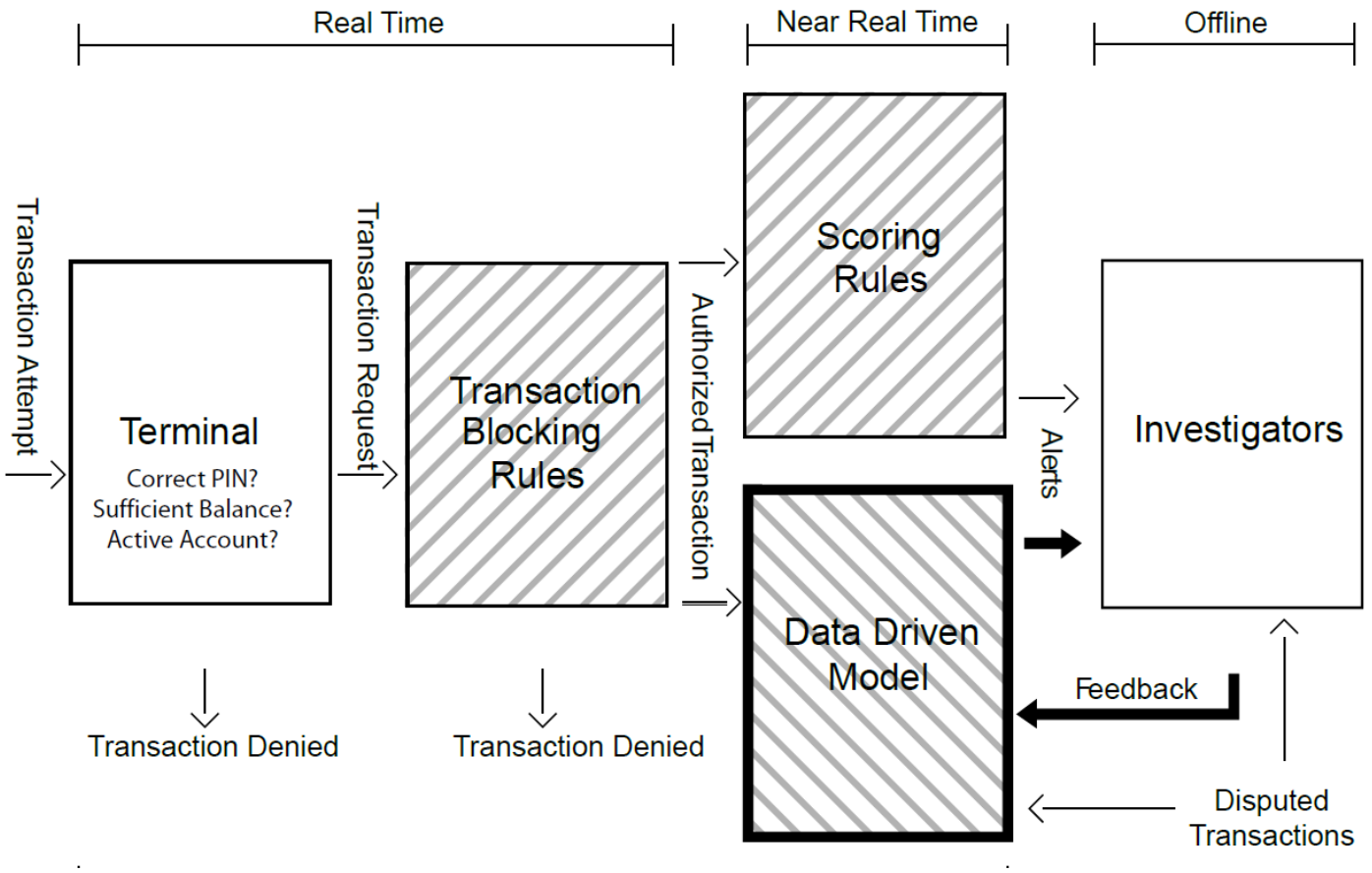


D. Carrera, B. Rossi, D. Zambon, P. Fragneto, and G. Boracchi "*ECG Monitoring in Wearable Devices by Sparse Models*" in ECML-PKDD 2016

ANOMALOUS ACTIVITIES DETECTION IN VIDEOS

USCD Anomaly Dataset http://www.svcl.ucsd.edu/projects/anomaly/dataset.html

FRAUD DETECTION IN CREDIT CARD TRANSACTIONS



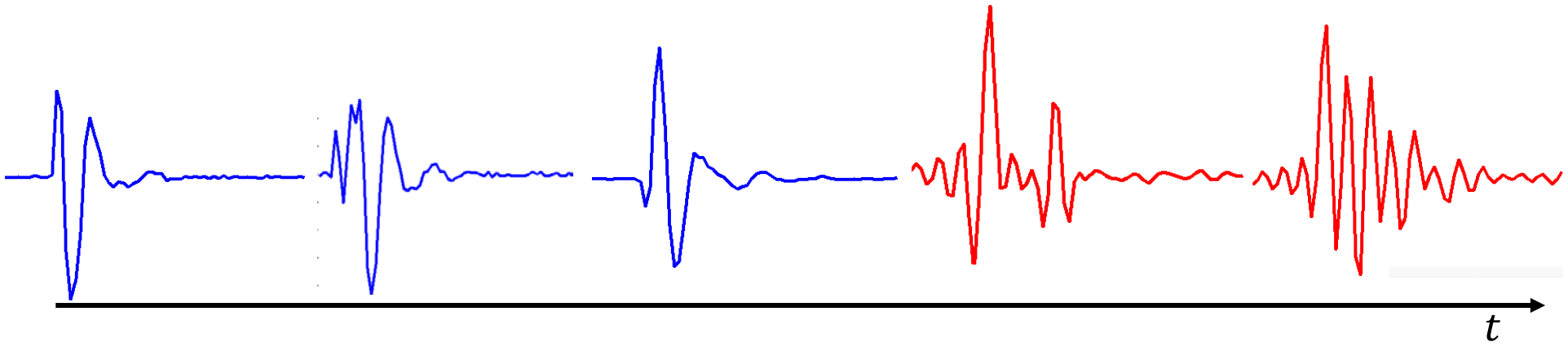
ULB

Dal Pozzolo A., Boracchi G., Caelen O., Alippi C. and Bontempi G., *"Credit Card Fraud Detection: a Realistic Modeling and a Novel Learning Strategy"* , IEEE TNNL 2017

… A CHANGE-DETECTION PROBLEM

Environmental Monitoring

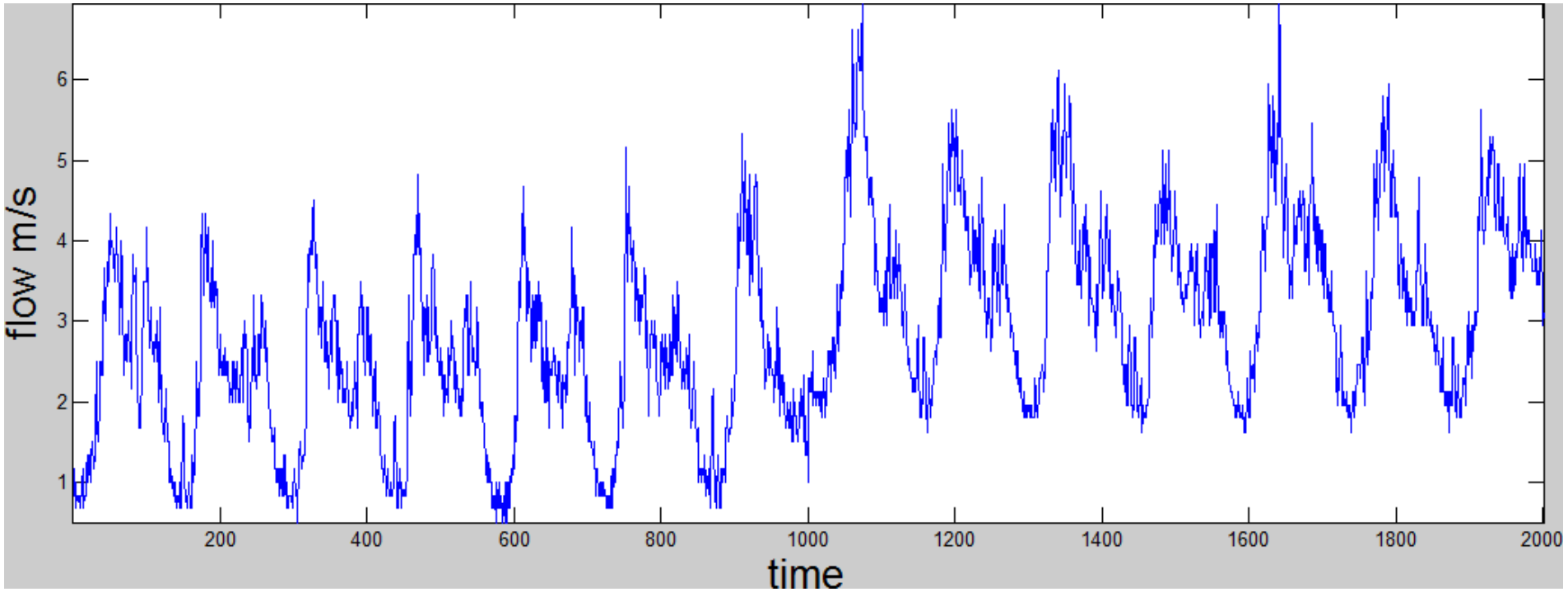
A sensor network monitoring rock faces: detecting changes in the waveforms that are recorded by MEMS sensors in network units.



C. Alippi, G. Boracchi, B. Wohlberg "*Change Detection in Streams of Signals with Sparse Representations*" in Proceedings of IEEE ICASSP 2014

… A CHANGE-DETECTION PROBLEM

Leak detection in Water Distribution Networks

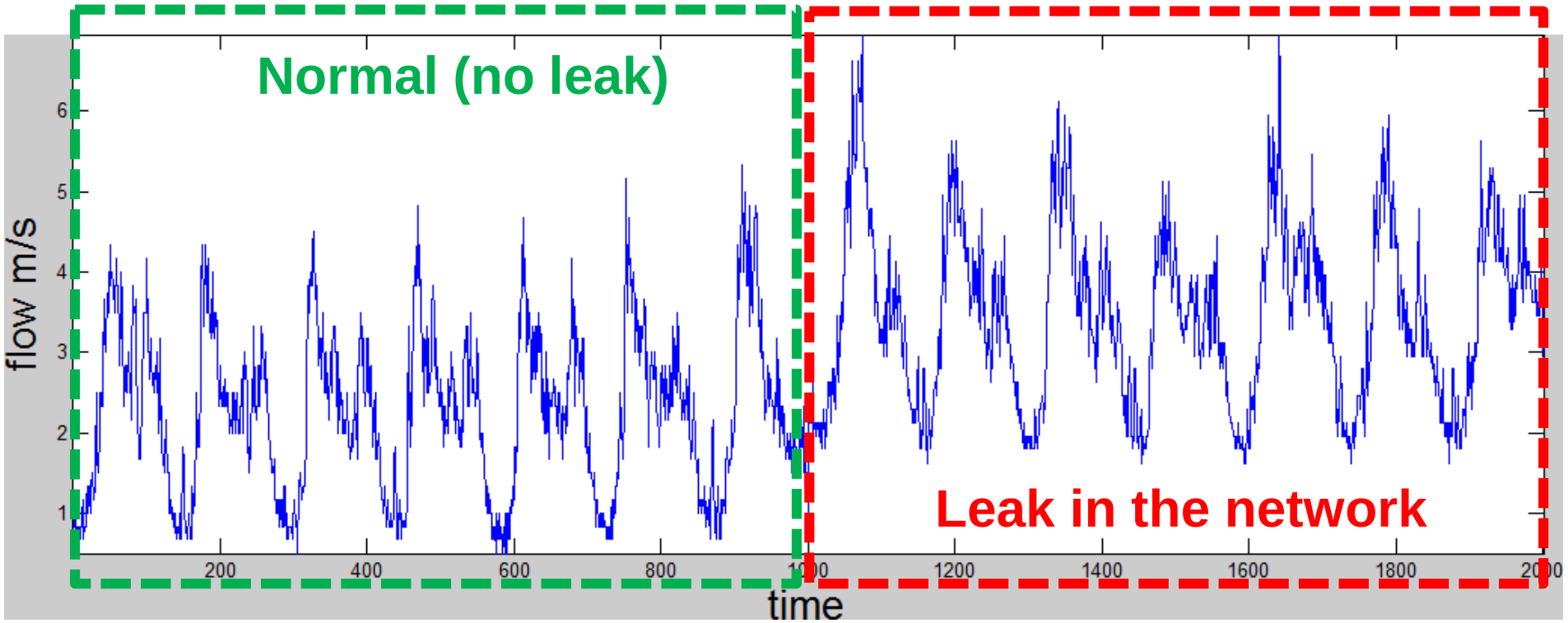


G. Boracchi and M. Roveri *"Exploiting Self-Similarity for Change Detection"* , IJCNN 2014, pp 3339 - 3346

… A CHANGE-DETECTION PROBLEM

Leak detection in Water Distribution Networks

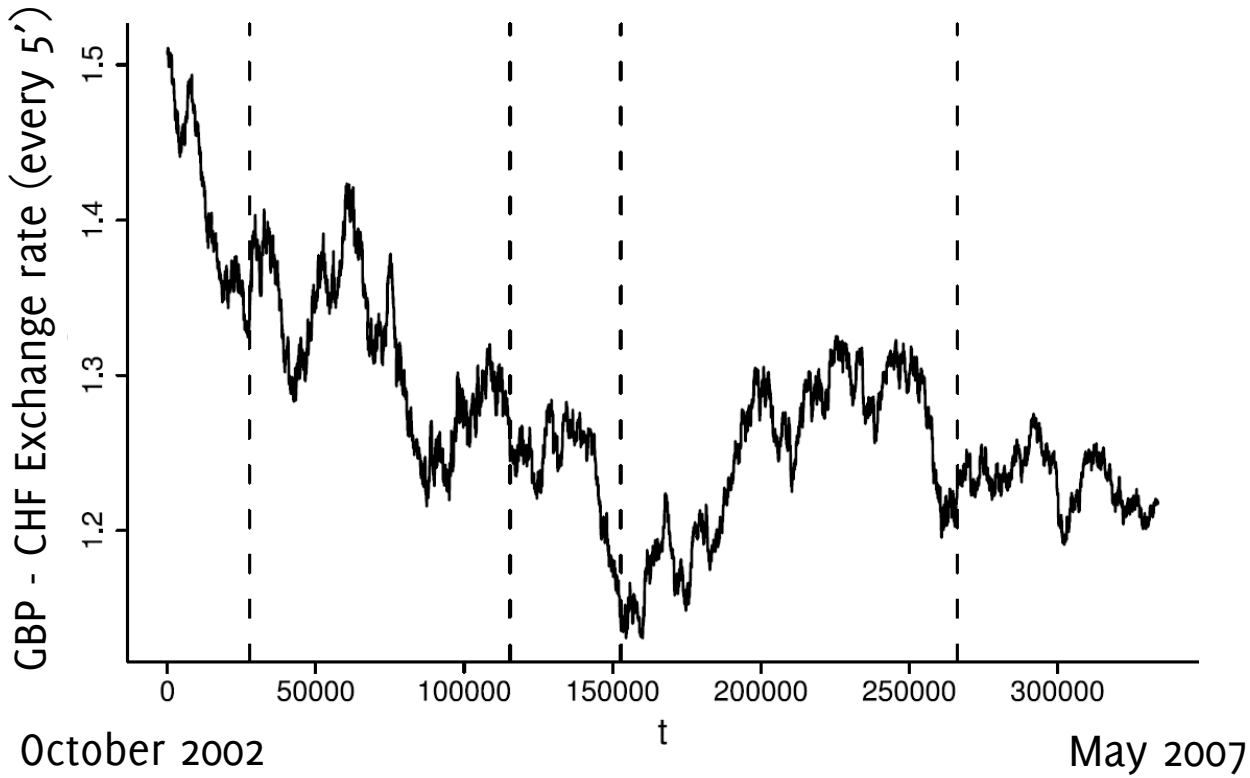
Similar problems arise in other critical infrastructure monitoring scenarios



G. Boracchi and M. Roveri *"Exploiting Self-Similarity for Change Detection"* , IJCNN 2014, pp 3339 - 3346

… A CHANGE-DETECTION PROBLEM

Time-series (including financial ones) are typically subject to changes, as the **data-generating process evolves** over time.



G.J. Ross, D.K. Tasoulis, N.M. Adams *"Nonparametric monitoring of data streams for changes in location and scale"* Technometrics 2011

… A CHANGE-DETECTION PROBLEM

Learning problems related to **predicting user preferences / interests,** such as:

- Recommendation systems
- Spam / email filtering

Changes arise when users change their own preferences.

Changes have to be detected to update the system accordingly

Spam Classification

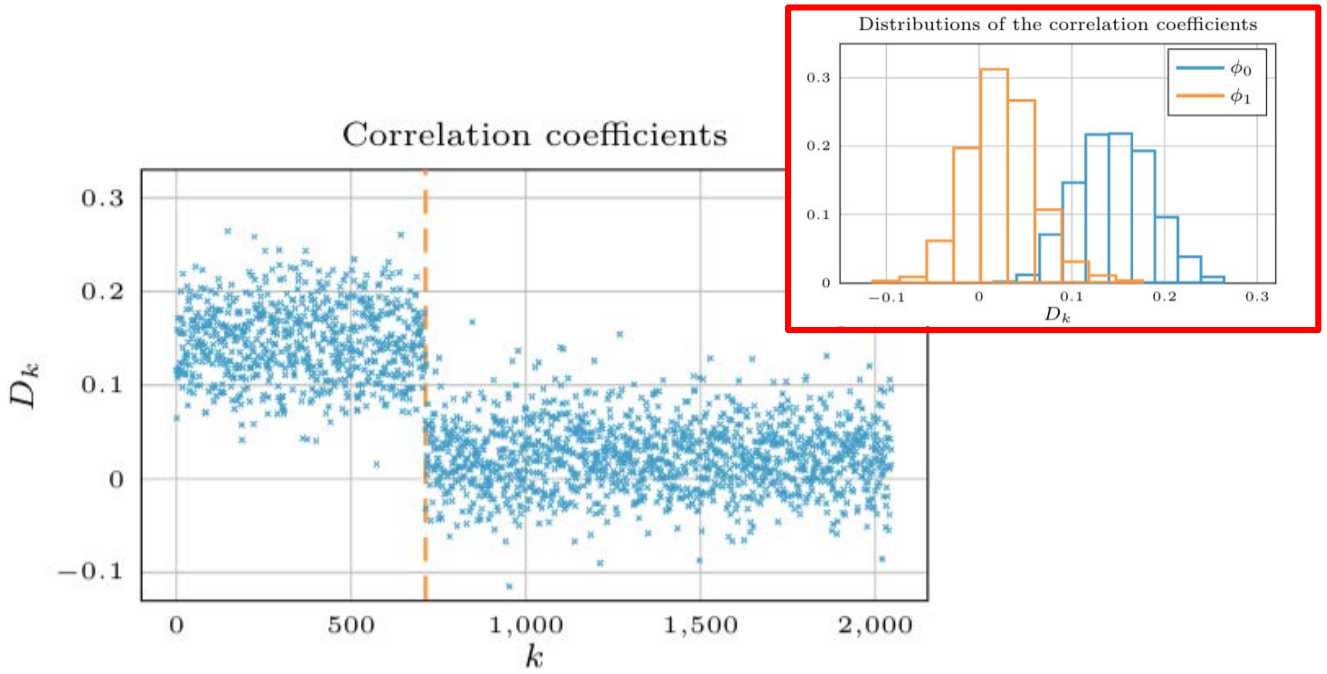
C. Alippi, G. Boracchi, M. Roveri, *"Just-in-time classifiers for recurrent concepts"*. IEEE TNNLS, 24(4), 620-634 (2013).

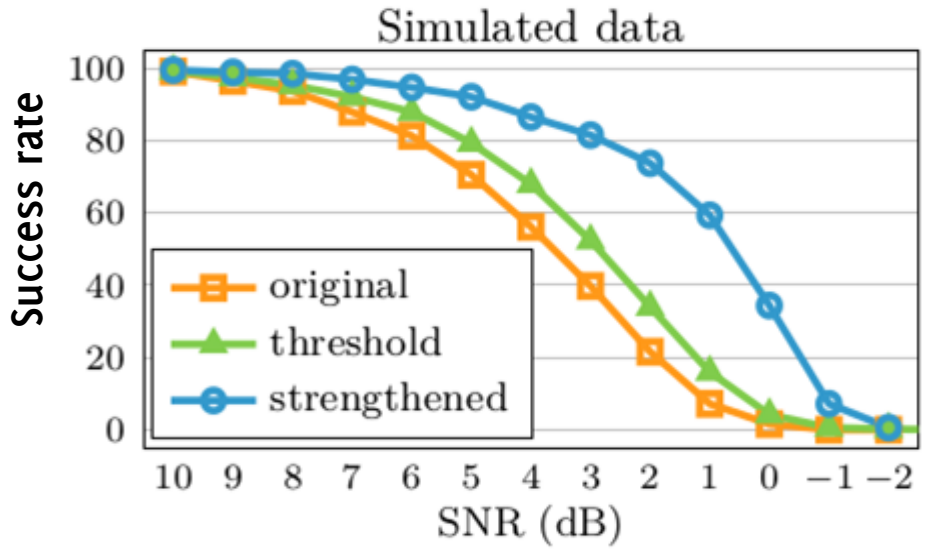
J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, *"A survey on concept drift adaptation"*. ACM Computing Surveys 2014

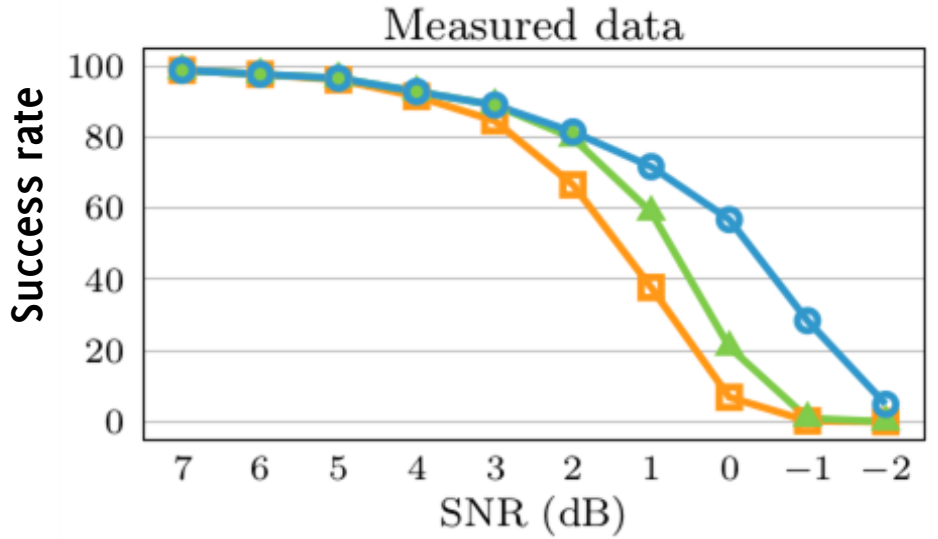
… A CHANGE DETECTION PROBLEM

Distribution changes in side channel attacks indicate a wrong guess for the key being forced.

Carefully designed attacks leverage change-detection algorithms to achieve much higher success rates







L. Frittoli, M. Bocchi, S. Mella, D. Carrera, B. Rossi, F. Fragneto, R. Susella, and G. Boracchi "Strengthening Sequential Side-Channel Attacks Through Change Detection", IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES), 2020(3), pp. 1-21 doi: 10.13154/tches.v2020.i3.1-21

TUTORIAL OUTLINE

PRESENTATION OUTLINE

Background:

- Problem formulation in random variables / signals and images.
- Main ingredients and performance measures.
- Solutions in the ideal settings.

Detection Algorithms for Random Variables:

- Anomaly Detection: the general approach and the most relevant solutions.
  - Supervised, Semi-supervised, unsupervised.
- Change Detection: the general approach and the most relevant solutions.
  - Sequential monitoring and histogram-based monitoring.
  - Change detection in high-dimensional datastreams.

PRESENTATION OUTLINE

Detection Algorithms for Signals and Images

- Detection by learned models:
  - Reconstruction-based methods.
  - Feature-based methods.
  - Reference-based methods.
- Counteracting domain Shift in Detection Problems.
- Anomaly detection by deep learning models:
  - Transfer Learning / Self-supervised.
  - Autoencoders.
  - Domain based.
  - Generative models.

DISCLAIMERS

We will consider **unsupervised and semi-supervised approaches**, as these better conform with anomaly and change detection scenarios.

We will **mainly** consider **numerical data**. In some cases, extensions apply to categorical or ordinal data.

In change detection, we will mainly focus on **datastreams**, which do **not have a fixed length** and that have to be **analyzed while data are being received**.

We will refer to either changes/anomalies according to **our personal experience**

For a **complete overview**, please refer to surveys reported below.

V. Chandola, A. Banerjee, V. Kumar. *"Anomaly detection: A survey"*. ACM Comput. Surv. 41, 3, Article 15 (July 2009), 58 pages.

Pimentel, M. A., Clifton, D. A., Clifton, L., Tarassenko, L. *"A review of novelty detection"* Signal Processing, 99, 215-249 (2014)

A. Zimek, E. Schubert, H.P. Kriegel. *"A survey on unsupervised outlier detection in high-dimensional numerical data"* Statistical Analysis and Data Mining: The ASA Data Science Journal, 5(5), 2012.

T. Ehret, A. Davy, JM Morel, M. Delbracio *"Image Anomalies: A Review and Synthesis of Detection Methods"*, Journal of Mathematical Imaging and Vision, 1-34

L. Ruff, et al. *"A Unifying Review of Deep and Shallow Anomaly Detection" preprint 2020* https://arxiv.org/abs/2009.11732

THE PROBLEM FORMULATION

Anomaly / Change Detection Problems
in a Statistical Framework

ANOMALIES

"Anomalies are patterns in data that do not conform to a well defined notion of normal behavior"

Thus:

- **Normal data** are generated from a **stationary process** $\mathcal{P}_N$
- **Anomalies** are from a **different process** $\mathcal{P}_A \neq \mathcal{P}_N$

Examples:

- **Frauds** in the stream of all the credit card transactions
- **Arrhythmias** in ECG tracings
- **Defective regions in an image**, which do not conform a reference pattern

Anomalies might appear as **spurious** elements, and are typically the most **informative** samples in the stream

V. Chandola, A. Banerjee, V. Kumar. *"Anomaly detection: A survey"*. ACM Comput. Surv. 41, 3, Article 15 (July 2009), 58 pages.

ANOMALY-DETECTION IN A STATISTICAL FRAMEWORK

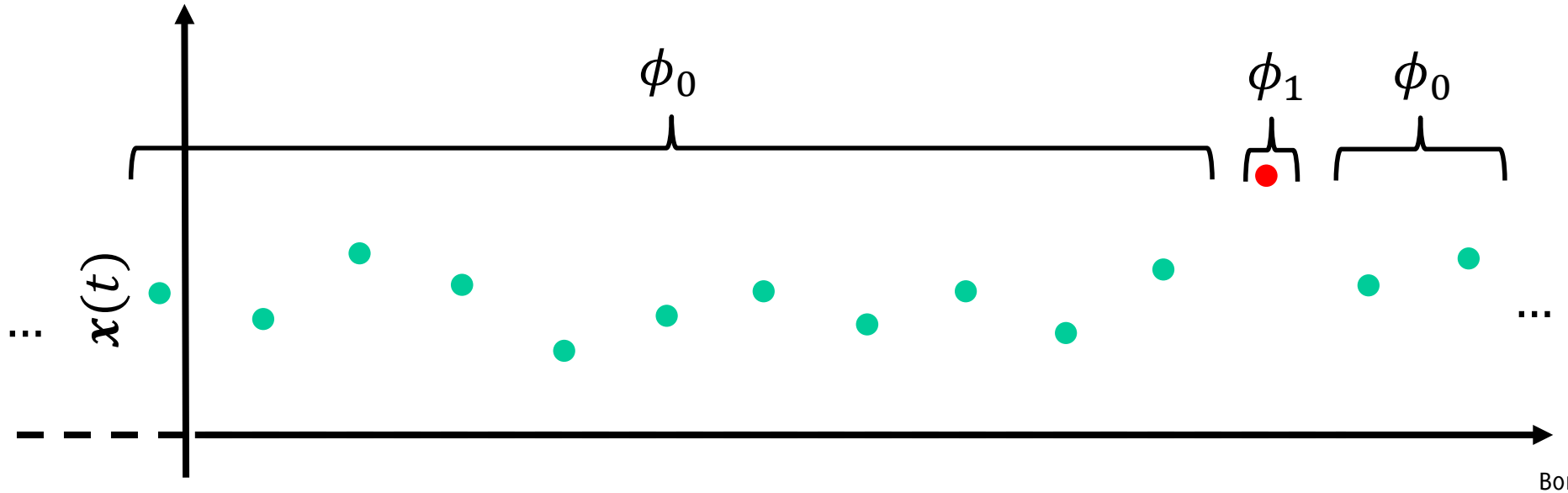
Often, the anomaly-detection problem **boils down to**:

Monitor a set of data (not necessarily a stream)

$$\{\boldsymbol{x}(t),\ t = t_0, \dots\},\quad \boldsymbol{x}(t) \in \mathbb{R}^d$$

where $x(t)$ are realizations of a **random variable having pdf** ϕ_o, and detect **outliers** i.e., those points that do not conform with ϕ_o

$$\boldsymbol{x}(t) \sim \begin{cases} \phi_0 & \text{normal data} \\ \phi_1 & \text{anomalies} \end{cases},$$



Boracchi, Carrera, ICPR 2020

ANOMALY-DETECTION IN A STATISTICAL FRAMEWORK

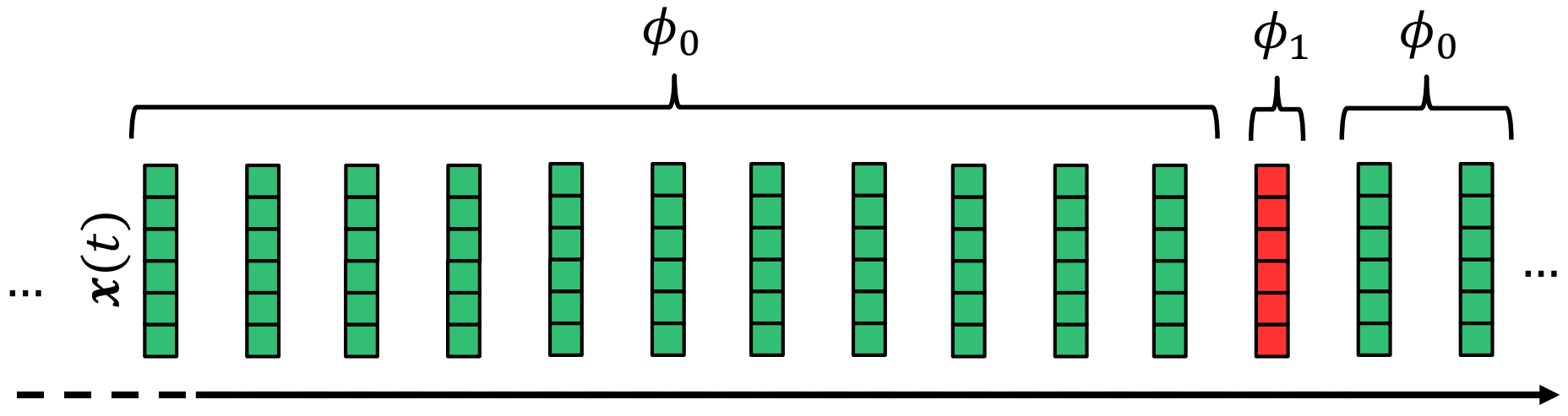
Often, the anomaly-detection problem **boils down to**:

Monitor a set of data (not necessarily a stream)

$$\{\boldsymbol{x}(t),\ t = t_0, \dots\},\quad \boldsymbol{x}(t) \in \mathbb{R}^d$$

where $x(t)$ are realizations of a **random variable having pdf** ϕ_o, and detect **outliers** i.e., those points that do not conform with ϕ_o

$$\boldsymbol{x}(t) \sim \begin{cases} \phi_0 & \text{normal data} \\ \phi_1 & \text{anomalies} \end{cases},$$



Boracchi, Carrera, ICPR 2020

ANOMALY-DETECTION IN A STATISTICAL FRAMEWORK

Often, the anomaly-detection problem **boils down to**:

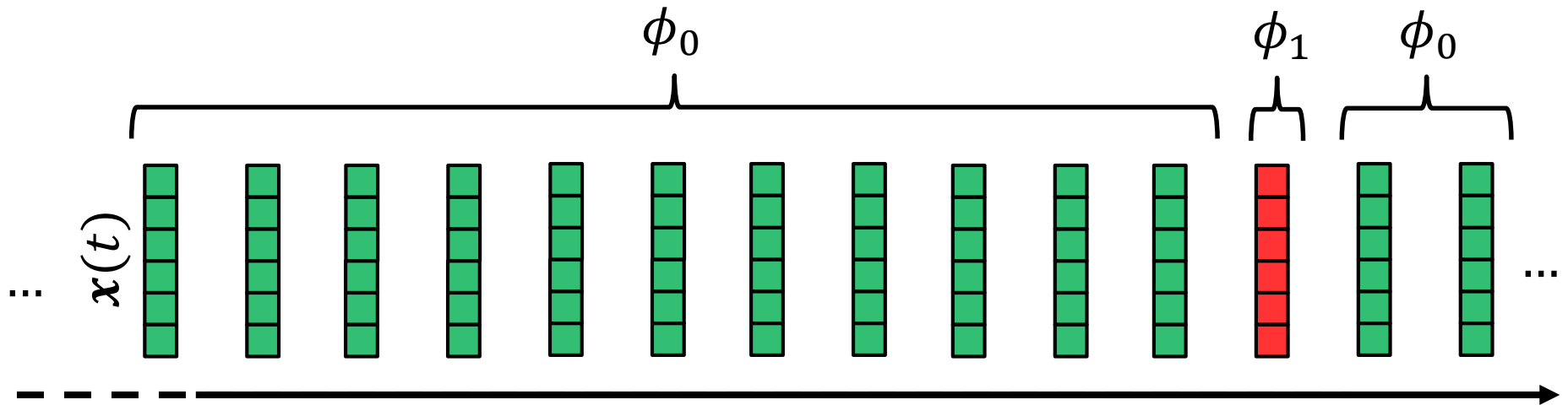
Monitor a set of data (not necessarily a stream)

$$\{\boldsymbol{x}(t),\ t = t_0, \dots\},\quad \boldsymbol{x}(t) \in \mathbb{R}^d$$

where $\boldsymbol{x}(t)$ are realizations of a random variable having pdf f_{ϕ_0} and detect **outliers** i.e., those

***Locate those* samples that do not conform the normal ones *or a* model explaining normal ones**

$$\boldsymbol{x}(t) \sim \begin{cases} f_{\phi_0} & \text{normal} \\ \phi_1 & \text{anomalies} \end{cases},$$



Boracchi, Carrera, ICPR 2020

THE LEGAL CASE OF MR HADLUM V. MRS HADLUM (1949)

The sole evidence of adultery consisted of the birth of a child 349 days after Mr Hadlum had left for military service abroad.

Vic Barnett *"The Study of Outliers: Purpose and Model"* Journal of the Royal Statistical Society. Series C (Applied Statistics) Vol. 27, No. 3 (1978), pp. 242-250

THE LEGAL CASE OF MR HADLUM V. MRS HADLUM (1949)

The sole evidence of adultery consisted of the birth of a child 349 days after Mr Hadlum had left for military service abroad.

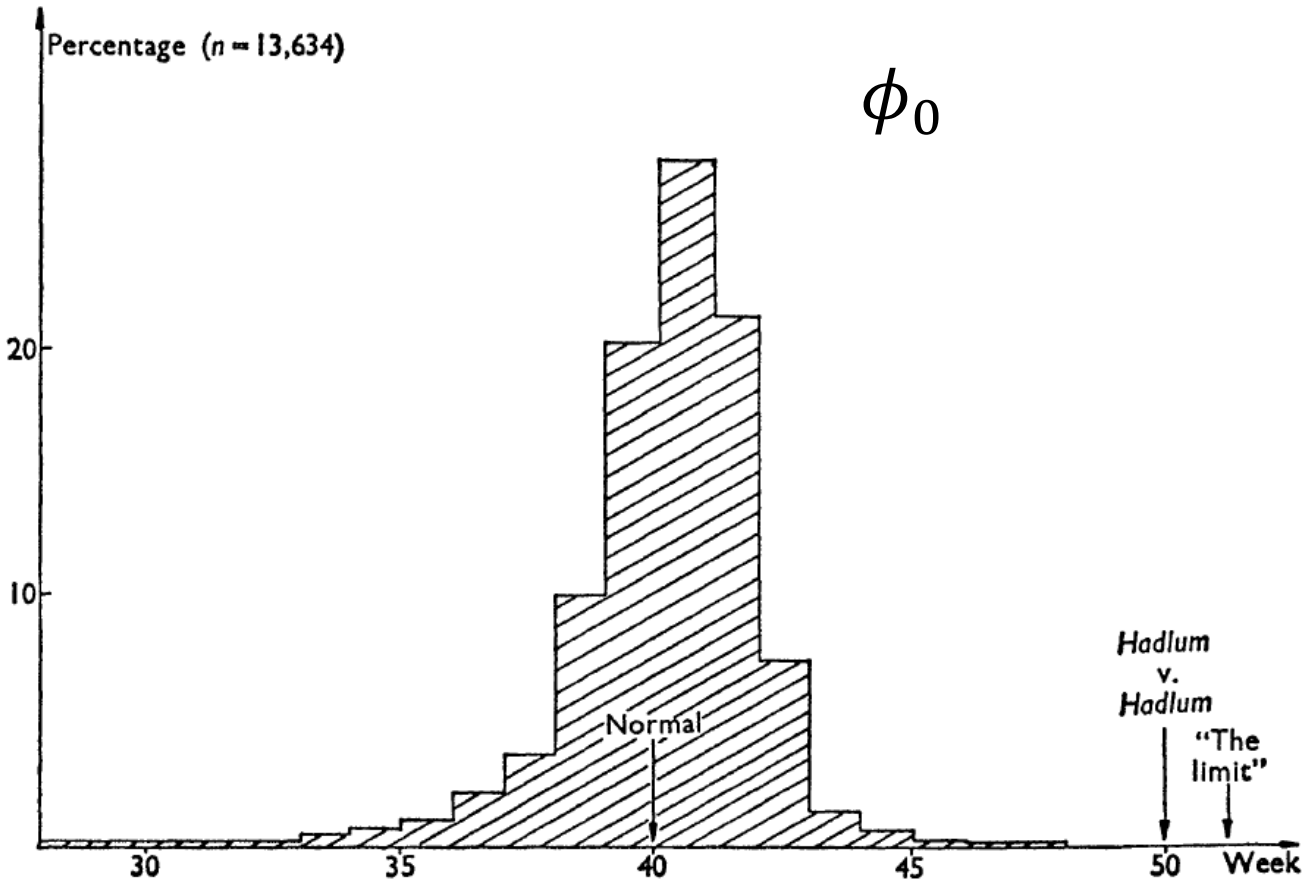


FIG. 1. Distribution of human gestation periods.

Vic Barnett *"The Study of Outliers: Purpose and Model"* Journal of the Royal Statistical Society. Series C (Applied Statistics) Vol. 27, No. 3 (1978), pp. 242-250

THE LEGAL CASE OF MR HADLUM V. MRS HADLUM (1949)

The sole evidence of adultery consisted of the birth of a child 349 days after Mr Hadlum had left for military service abroad.

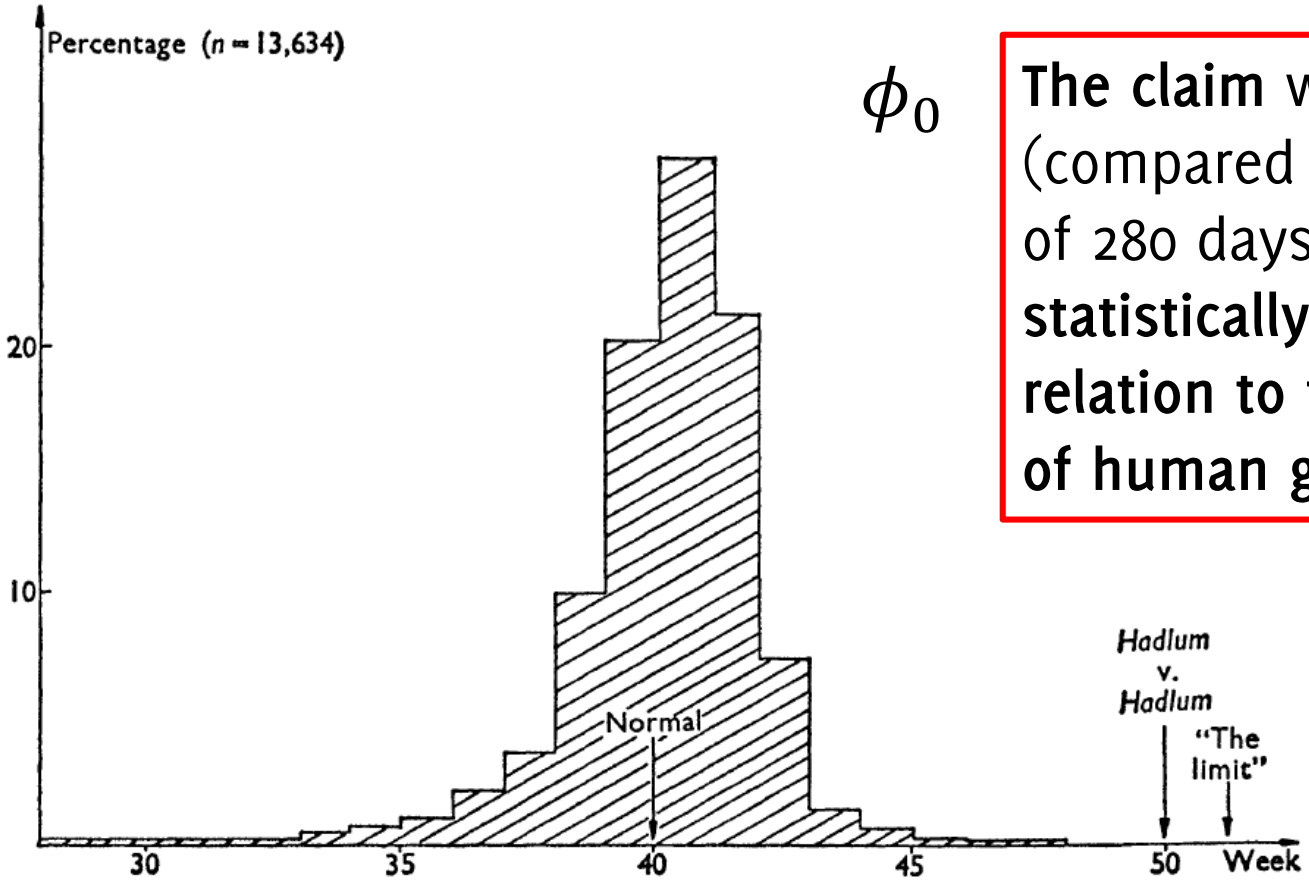


FIG. 1. Distribution of human gestation periods.

ϕ_0

The claim was that 349 days (compared with an average of 280 days) was discordant: **statistically unreasonable in relation to the distribution of human gestation periods.**

Vic Barnett *"The Study of Outliers: Purpose and Model"* Journal of the Royal Statistical Society. Series C (Applied Statistics) Vol. 27, No. 3 (1978), pp. 242-250

THE LEGAL CASE OF MR HADLUM V. MRS HADLUM (1949)

The sole evidence of adultery consisted of the birth of a child 349 days after Mr Hadlum had left for military service abroad.

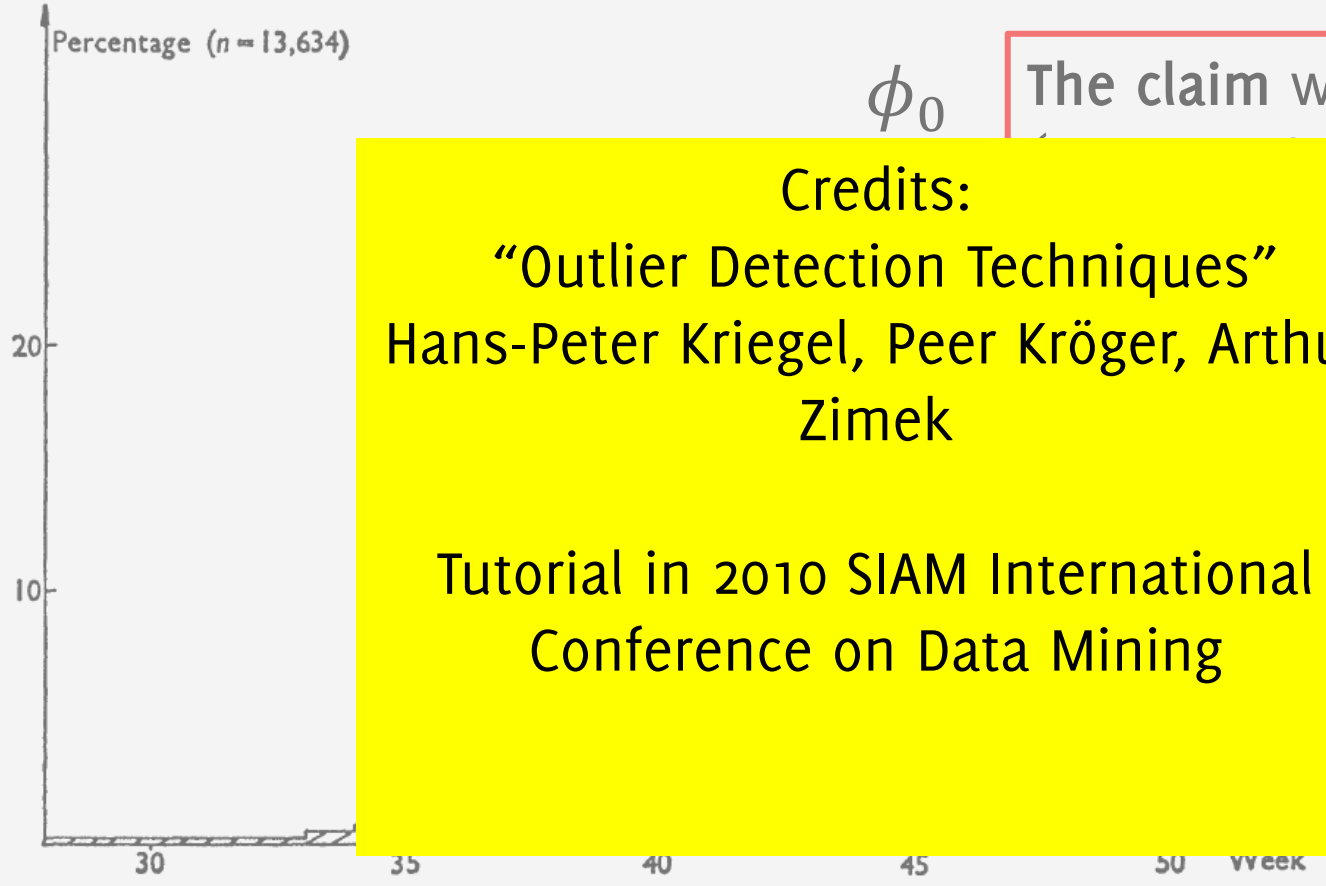


FIG. 1. Distribution of human gestation periods.

ϕ_0

The claim was that 349 days
h an average
as discordant:
reasonable in
distribution
ation periods.

Credits:
"Outlier Detection Techniques"
Hans-Peter Kriegel, Peer Kröger, Arthur Zimek

Tutorial in 2010 SIAM International Conference on Data Mining

Vic Barnett *"The Study of Outliers: Purpose and Model"* Journal of the Royal Statistical Society. Series C (Applied Statistics) Vol. 27, No. 3 (1978), pp. 242-250

PROCESS CHANGES

Normal data are generated in **stationary** conditions, i.e. **are i.i.d. realizations** of a process $\mathcal{P}_N$

After the change, **data are** generated from a **different process** $\mathcal{P}_A \neq \mathcal{P}_N$, which **persists over time**

Examples:

- Quality inspection system: **faults** producing flawed components
- Environmental monitoring: **persistent changes in the morphology** of measured signals
- Change of **user interests** in on-demand platform

CHANGE-DETECTION IN A STATISTICAL FRAMEWORK

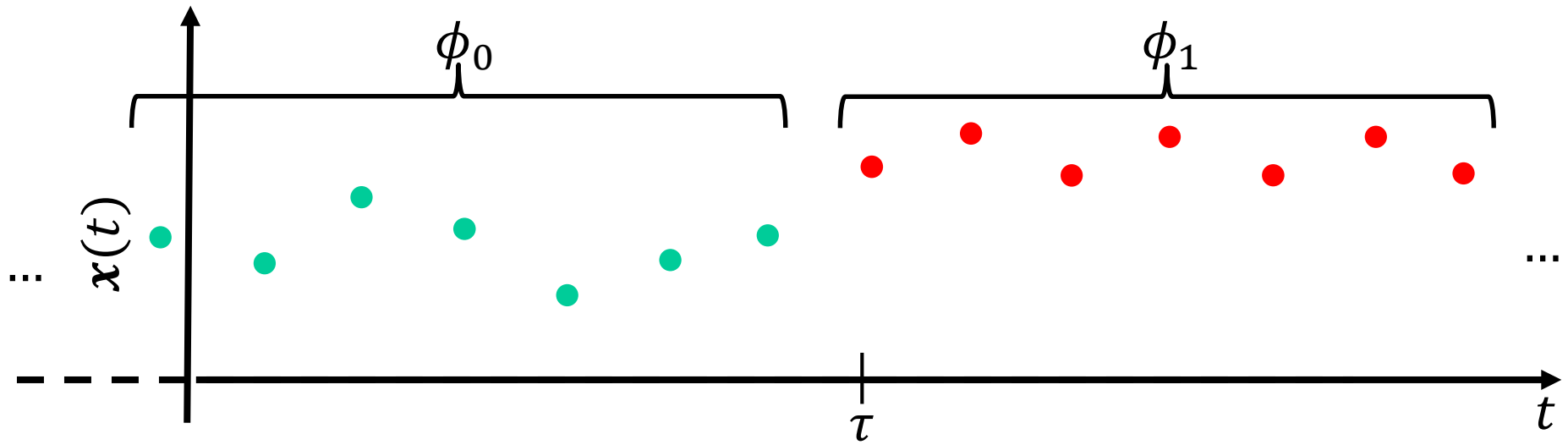
Often, the change-detection problem **boils down to:**

Monitor a **stream** $\{\boldsymbol{x}(t), t = 1, \dots\}$, $\boldsymbol{x}(t) \in \mathbb{R}^d$ of realizations of a **random variable**, and **detect the change-point** τ,

$$\boldsymbol{x}(t) \sim \begin{cases} \phi_0 & t < \tau \\ \phi_1 & t \geq \tau \end{cases} \quad \begin{matrix} \text{in control state} \\ \text{out of control state} \end{matrix},$$

where $\{\boldsymbol{x}(t),\ t < \tau\}$ **are i.i.d.** and $\phi_0 \neq \phi_1$

We denote such change as: $\phi_o \rightarrow \phi_1$



CHANGE-DETECTION IN A STATISTICAL FRAMEWORK

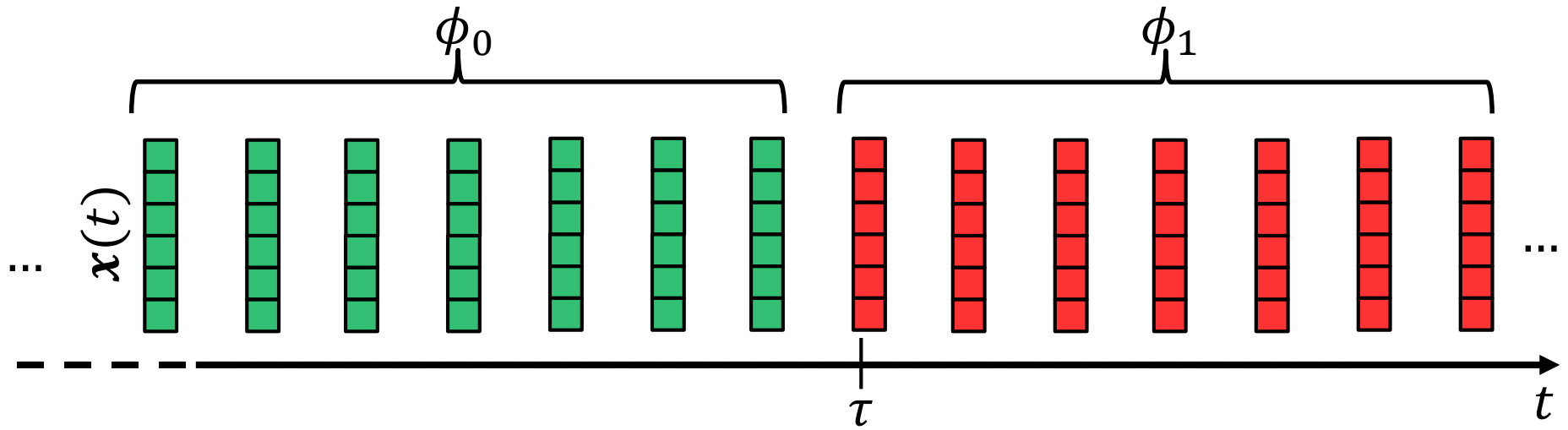
Often, the change-detection problem **boils down to:**

Monitor a **stream** $\{\boldsymbol{x}(t), t = 1, \dots\}$, $\boldsymbol{x}(t) \in \mathbb{R}^d$ of realizations of a **random variable**, and **detect the change-point** τ,

$$\boldsymbol{x}(t) \sim \begin{cases} \phi_0 & t < \tau \\ \phi_1 & t \geq \tau \end{cases} \quad \begin{matrix} \text{in control state} \\ \text{out of control state} \end{matrix},$$

where $\{\boldsymbol{x}(t),\ t < \tau\}$ **are i.i.d.** and $\phi_0 \neq \phi_1$

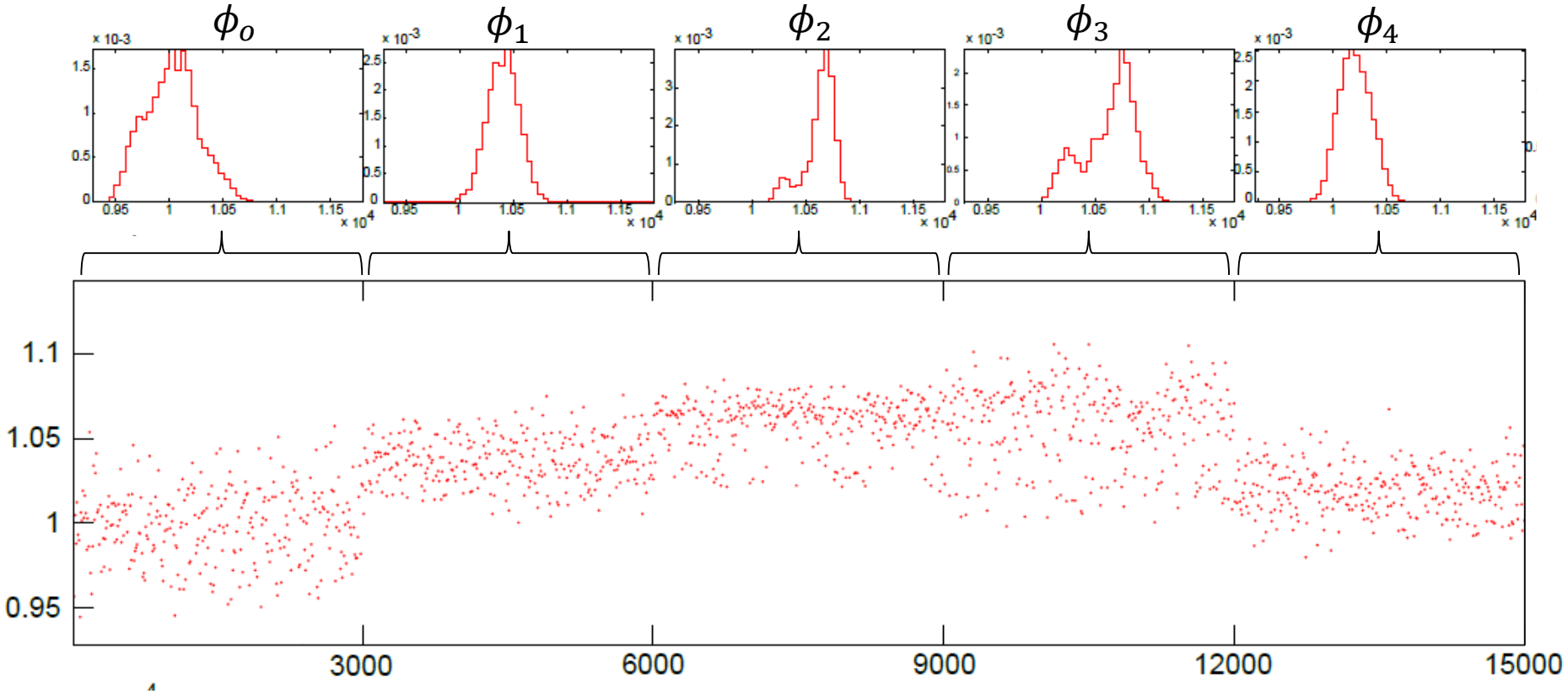
We denote such change as: $\phi_o \rightarrow \phi_1$



CHANGE-DETECTION IN A STATISTICAL FRAMEWORK

Here are data from an X-ray monitoring apparatus.

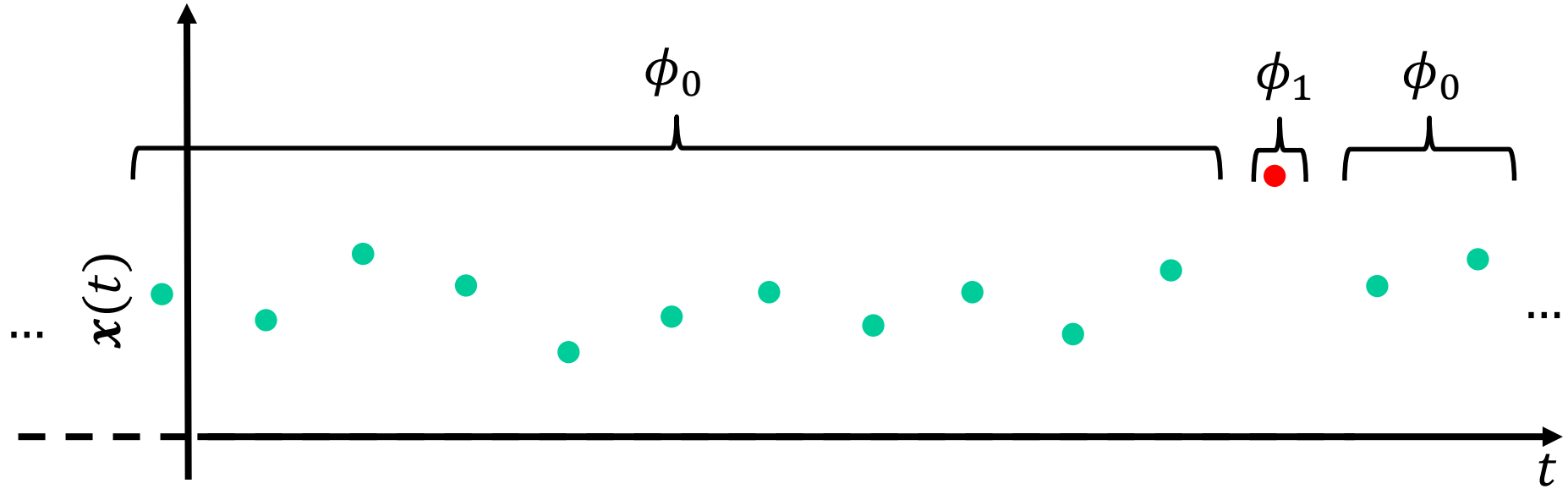
There are 4 changes $\phi_o \rightarrow \phi_1 \rightarrow \phi_2 \rightarrow \phi_3 \rightarrow \phi_4$ corresponding to different monitoring conditions and/or analyzed materials



chi, Carrera, ICPR 2020

PROCESS CHANGES VS ANOMALIES

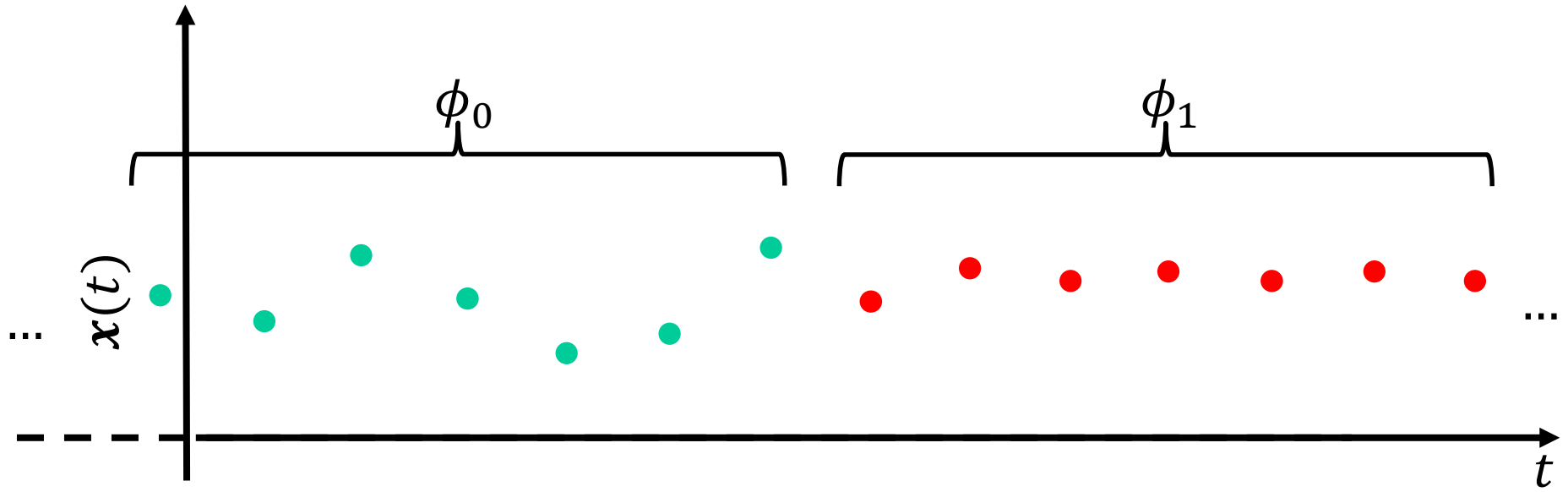
Not all anomalies are due to process changes



Boracchi, Carrera, ICPR 2020

PROCESS CHANGES VS ANOMALIES

Not all process changes result in anomalies



Boracchi, Carrera, ICPR 2020

DETECTION PROBLEMS IN SIGNALS / IMAGES / VIDEOS

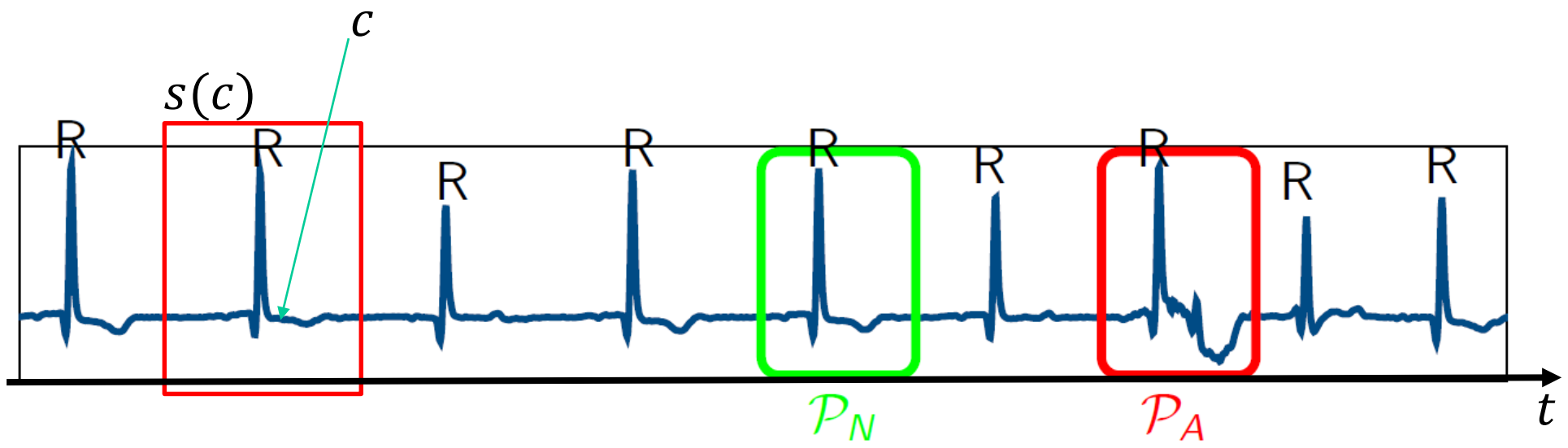
ANOMALY DETECTION IN TIME SERIES

Similar definition holds for detecting anomalies in a time series $s \subset \mathbb{R}^d$

The datastream is partitioned in segments $s(c)$ centered in a specific location c by sliding window or expert-driven algorithms

The goal is to determine whether each segment

$$s(c) \sim \begin{cases} \mathcal{P}_N & \text{normal data} \\ \mathcal{P}_A & \text{anomalies} \end{cases}$$



D. Carrera, B. Rossi, D. Zambon, P. Fragneto, and G. Boracchi "*ECG Monitoring in Wearable Devices by Sparse Models*" in Proceedings of ECML-PKDD 2016, 16 pages

ANOMALY DETECTION IN IMAGES

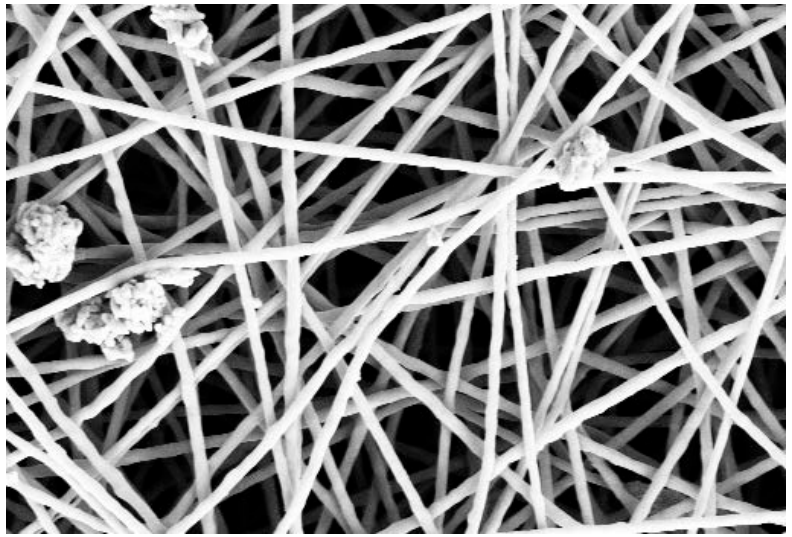
Let s be an image defined over the pixel domain $\mathcal{X} \subset \mathbb{Z}^2$,

let $c \in \mathcal{X}$ be a pixel and $s(c)$ the corresponding intensity.

Our goal is to **locate any anomalous region** in s, i.e. **estimating the unknown anomaly mask $\boldsymbol{\Omega}$** defined as

$$\Omega(c) = \begin{cases} 0 & \text{if } c \text{ falls in a normal region} \\ 1 & \text{if } c \text{ falls in an anomalous region} \end{cases}$$

s



Boracchi, Carrera, ICPR 2020

ANOMALY DETECTION IN IMAGES

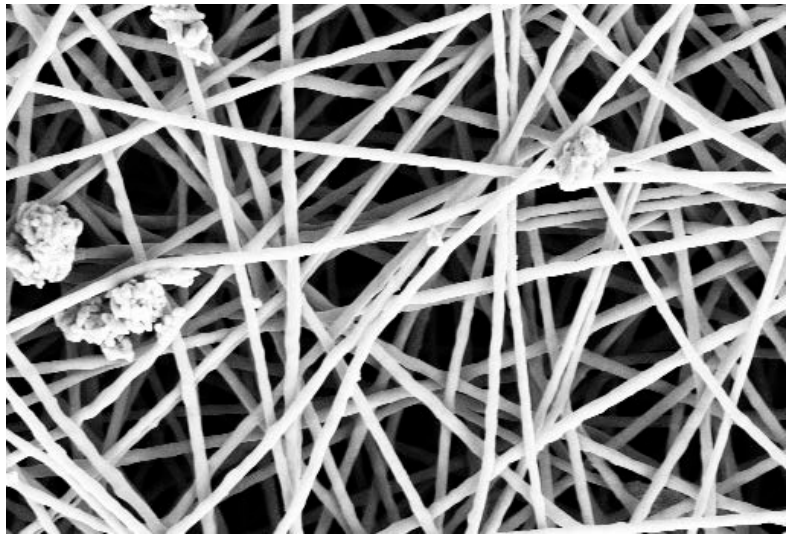
Let s be an image defined over the pixel domain $\mathcal{X} \subset \mathbb{Z}^2$,

let $c \in \mathcal{X}$ be a pixel and $s(c)$ the corresponding intensity.

Our goal is to **locate any anomalous region** in s, i.e. **estimating the unknown anomaly mask $\boldsymbol{\Omega}$** defined as

$$\Omega(c) = \begin{cases} 0 & \text{if } c \text{ falls in a normal region} \\ 1 & \text{if } c \text{ falls in an anomalous region} \end{cases}$$

s



Ω

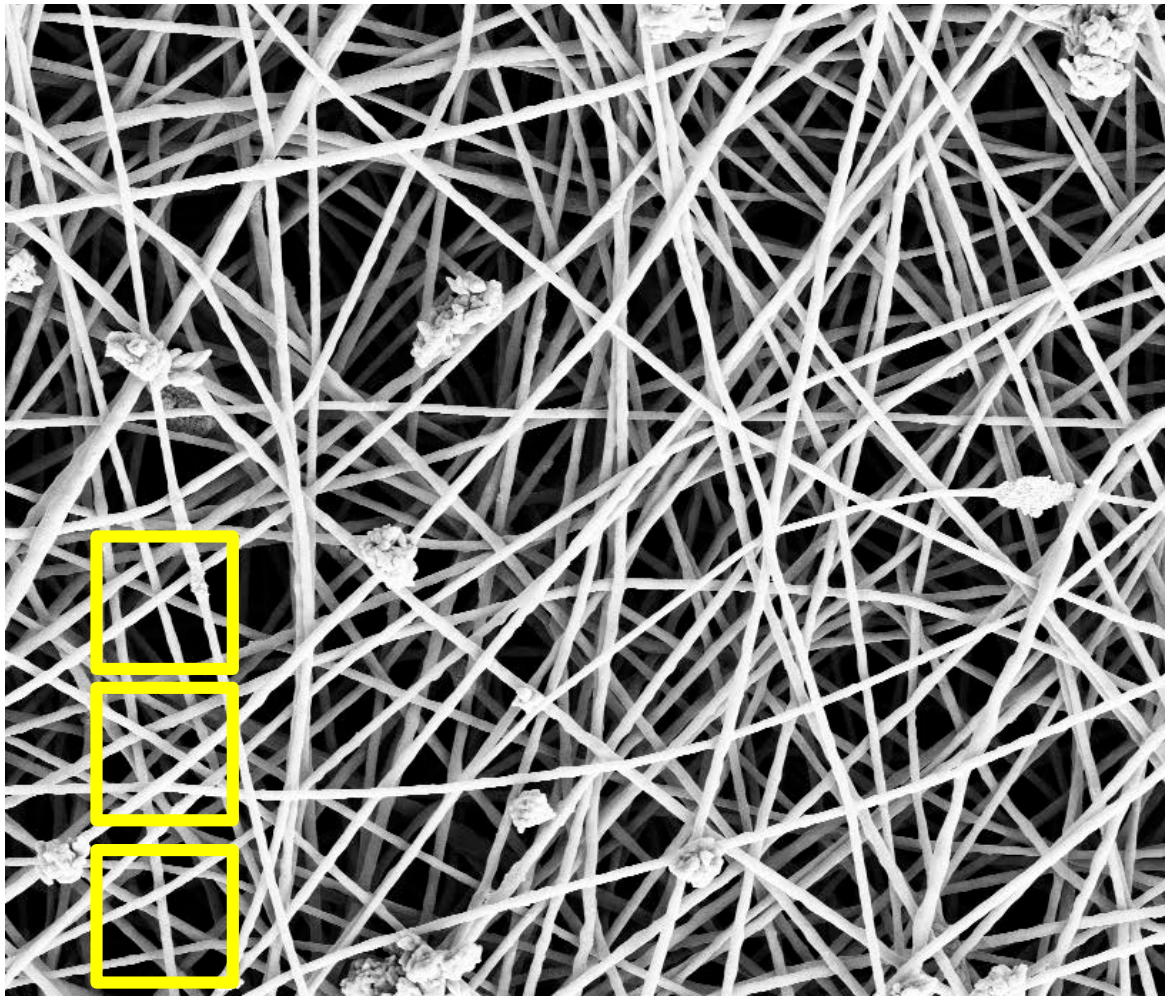
Boracchi, Carrera, ICPR 2020

PATCH-WISE ANOMALY DETECTION

The goal not determining whether the whole image is normal or anomalous, but **locate/segment possible anomalies**

Therefore, it is convenient to

1. **Analyze the image patch-wise**
2. Isolate regions containing patches that are detected as as anomalies

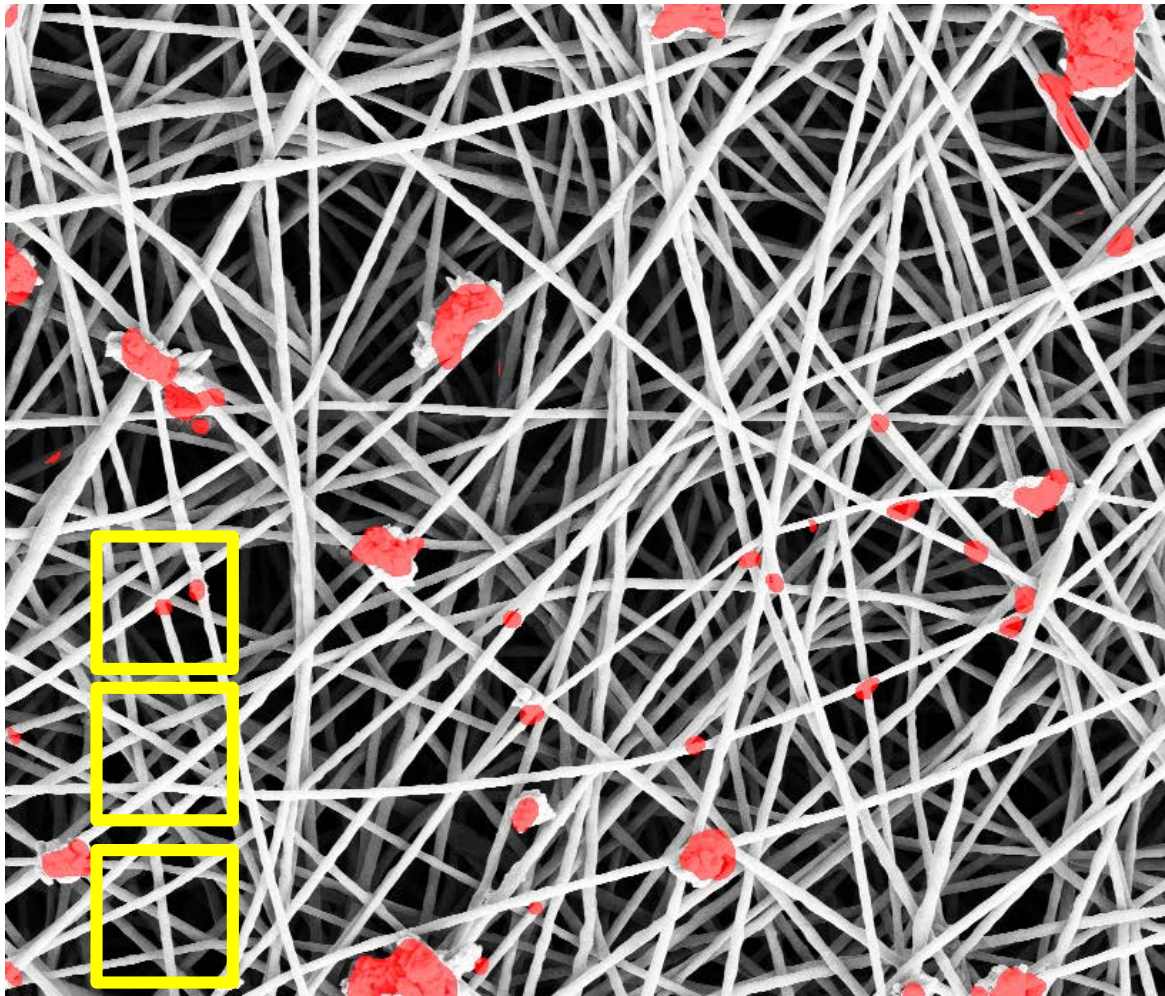


PATCH-WISE ANOMALY DETECTION

The goal not determining whether the whole image is normal or anomalous, but **locate/segment possible anomalies**

Therefore, it is convenient to

1. **Analyze the image patch-wise**
2. Isolate regions containing patches that are detected as as anomalies

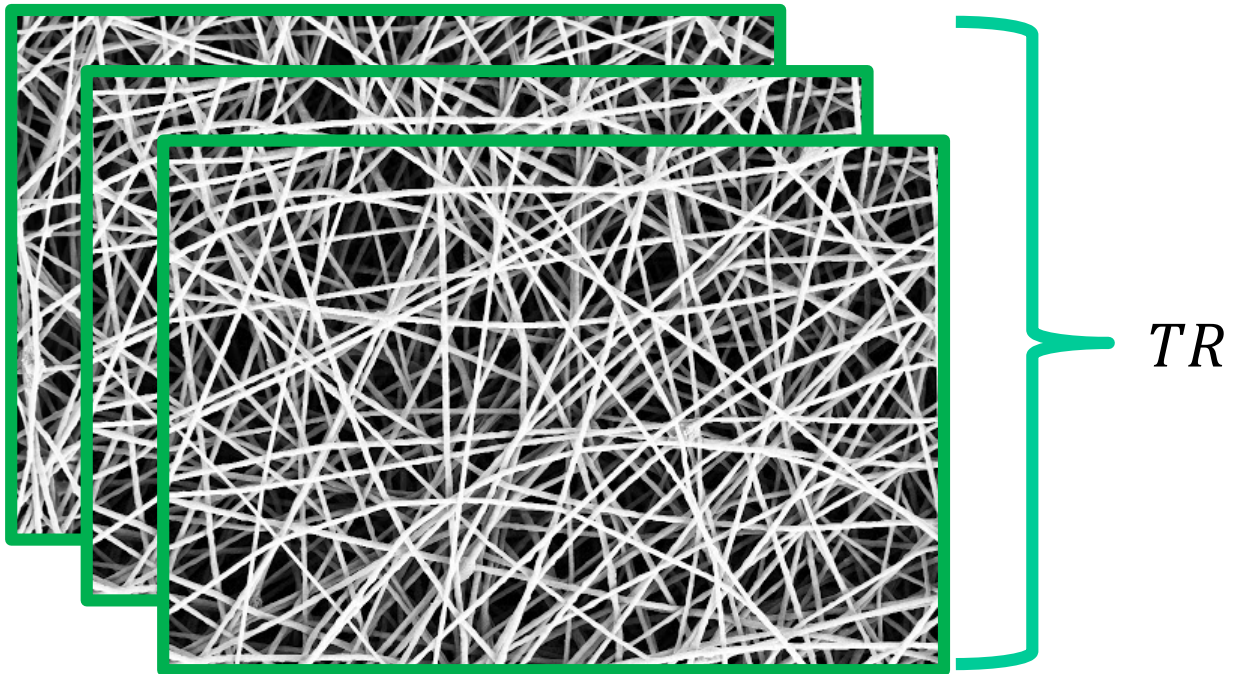


TYPICAL ASSUMPTIONS

A training set TR is provided for configuring the AD algorithm

Depending on the algorithm

- Only normal images -> semi-supervised methods
- Unlabeled images -> unsupervised methods
- Annotated images -> supervised methods



Boracchi, Carrera, ICPR 2020

DETECTING ANOMALOUS ACTIVITIES IN VIDEOS

It is very similar to image settings, but $\mathcal{X} \subset \mathbb{Z}^2 \times \mathbb{R}^+$ is a spatio-temporal domain

The goal is to locate any anomalous region in s, i.e. estimating the unknown anomaly mask $\boldsymbol{\Omega}$ defined as

$$\Omega(c) = \begin{cases} 0 & \text{if } c \text{ falls in a normal region} \\ 1 & \text{if } c \text{ falls in an anomalous region} \end{cases}$$

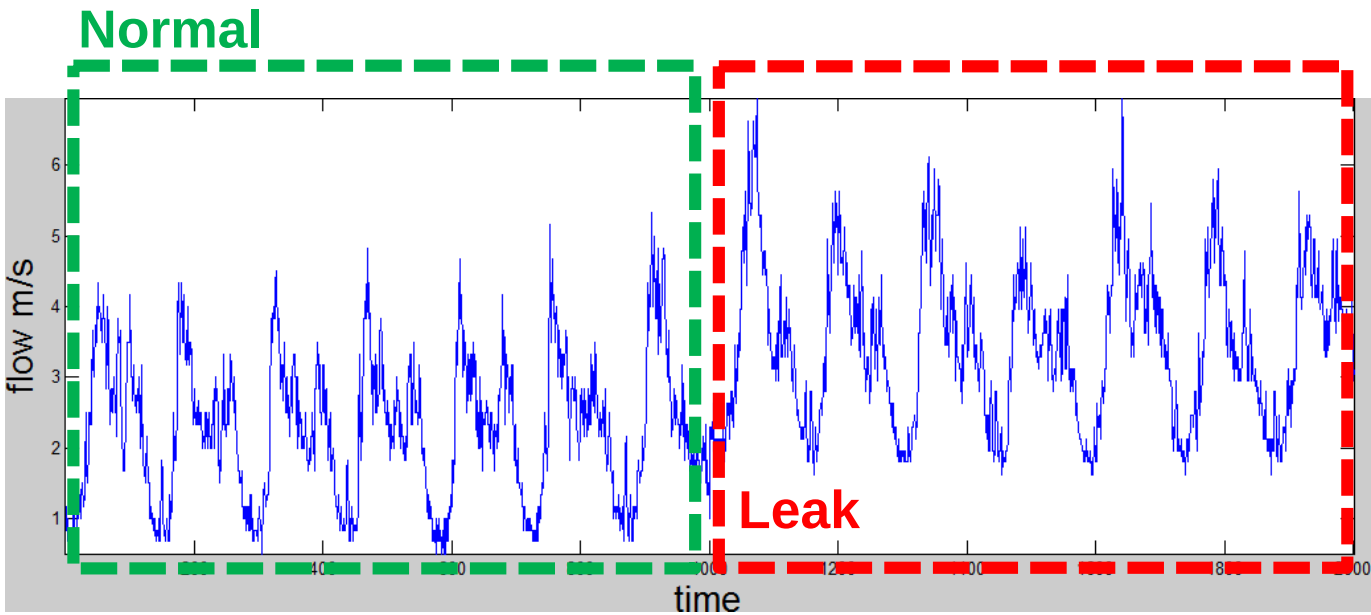
USCD Anomaly Dataset http://www.svcl.ucsd.edu/projects/anomaly/dataset.html

CHANGE-DETECTION IN TIME SERIES

Monitor a **time series** $\{\boldsymbol{x}(t), t = 1, \ldots\}$, $\boldsymbol{x}(t) \in \mathbb{R}^d$ to detect a change-point τ,

$$\boldsymbol{x}(t) \sim \begin{cases} \mathcal{P}_N & t < \tau \quad \text{in control state} \\ \mathcal{P}_A & t \geq \tau \quad \text{out of control state} \end{cases},$$

where $\mathcal{P}_A \neq \mathcal{P}_N$ and $\{\boldsymbol{x}(t),\ t < \tau\}$ typically exhibit some characteristic / pattern that is peculiar for normal data (e.g. similarity, smoothness, …)



G. Boracchi and M. Roveri *"Exploiting Self-Similarity for Change Detection"* , IJCNN 2014, pp 3339 - 3346

DETECTION ALGORITHMS: MAIN INGREDIENTS

THE ANOMALY / CHANGE DETECTION PROBLEMS

Anomaly-detection problem:

Locate those samples that do not conform the normal ones or a model explaining normal ones

Anomalies in data translate to significant information

Change-detection problem:

Given the previously estimated model, the arrival of new data invites the question: "Is yesterday's model capable of explaining today's data?"

Detecting process changes is important to understand the monitored phenomenon

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

C. J. Chu, M. Stinchcombe, H. White "*Monitoring Structural Change*" Econometrica Vol. 64, No. 5 (Sep., 1996), pp. 1045-1065.

THE TYPICAL SOLUTIONS

Most algorithms are composed of:

- A **statistic** that has a known response to normal data (e.g., the average, the sample variance, the log-likelihood, the confidence of a classifier, an “anomaly score”…)
- A **decision rule** to analyze the statistic (e.g., an adaptive threshold, a confidence region)

THE TYPICAL SOLUTIONS

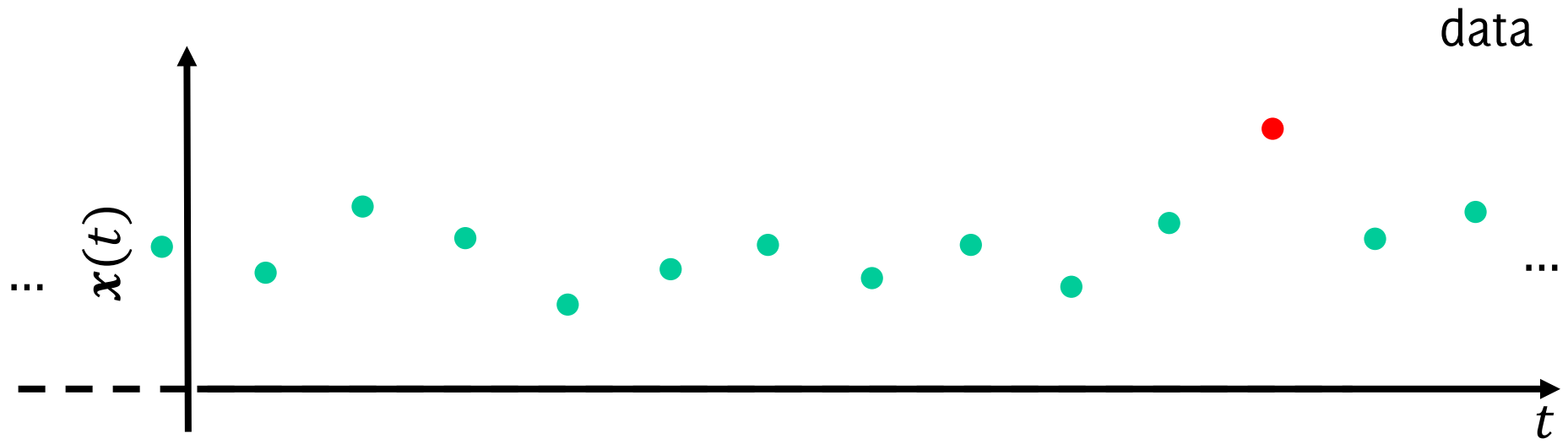
Anomaly-detection algorithms:

Statistics and decision rules are **“one-shot”**, analyzing a set of historical data or each new data (or chunk) independently

Change-detection algorithms:

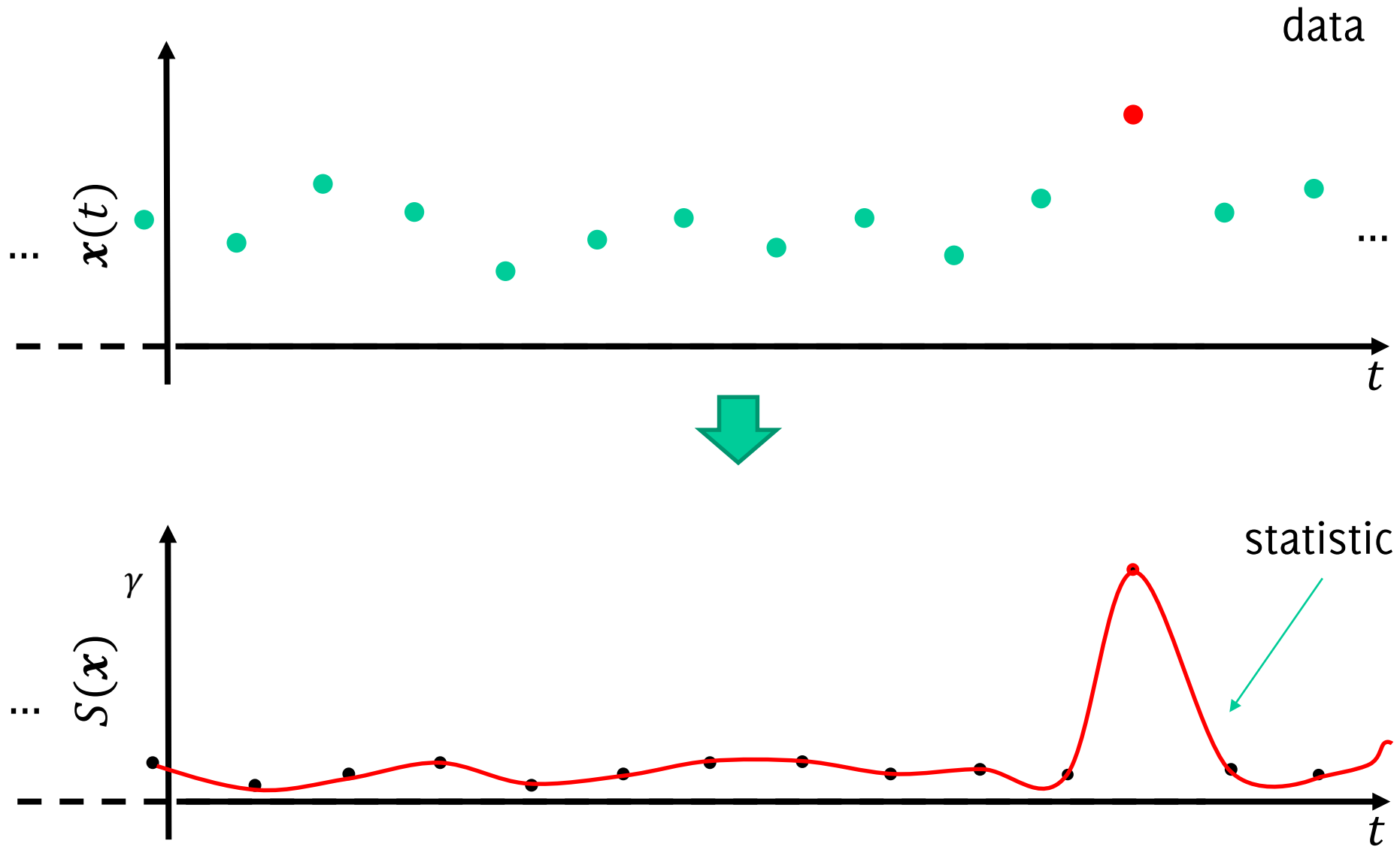
Statistics and decision rules are **sequential**, as they make a decision considering all the data received so far

THE TYPICAL SOLUTIONS



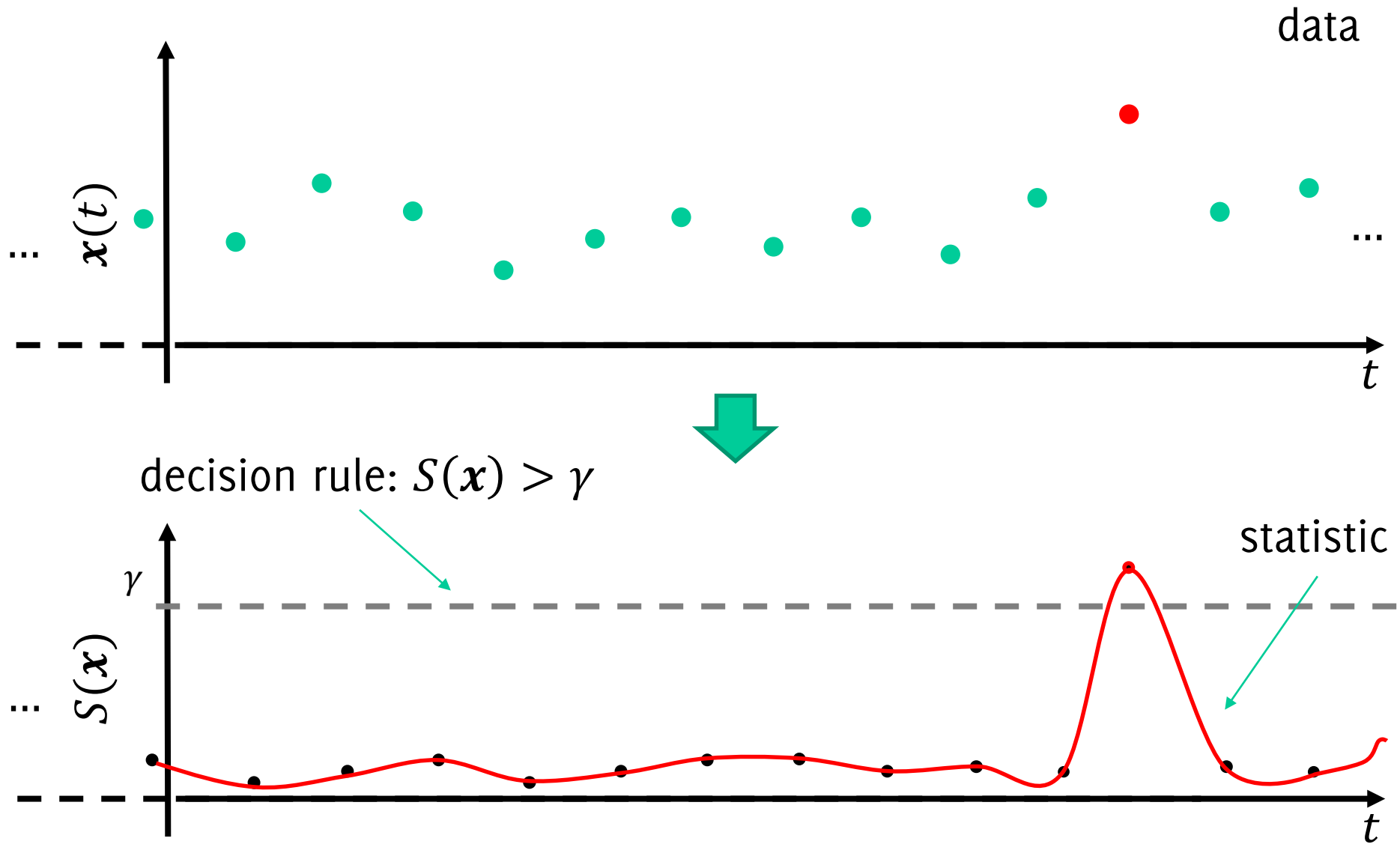
Boracchi, Carrera, ICPR 2020

THE TYPICAL SOLUTIONS



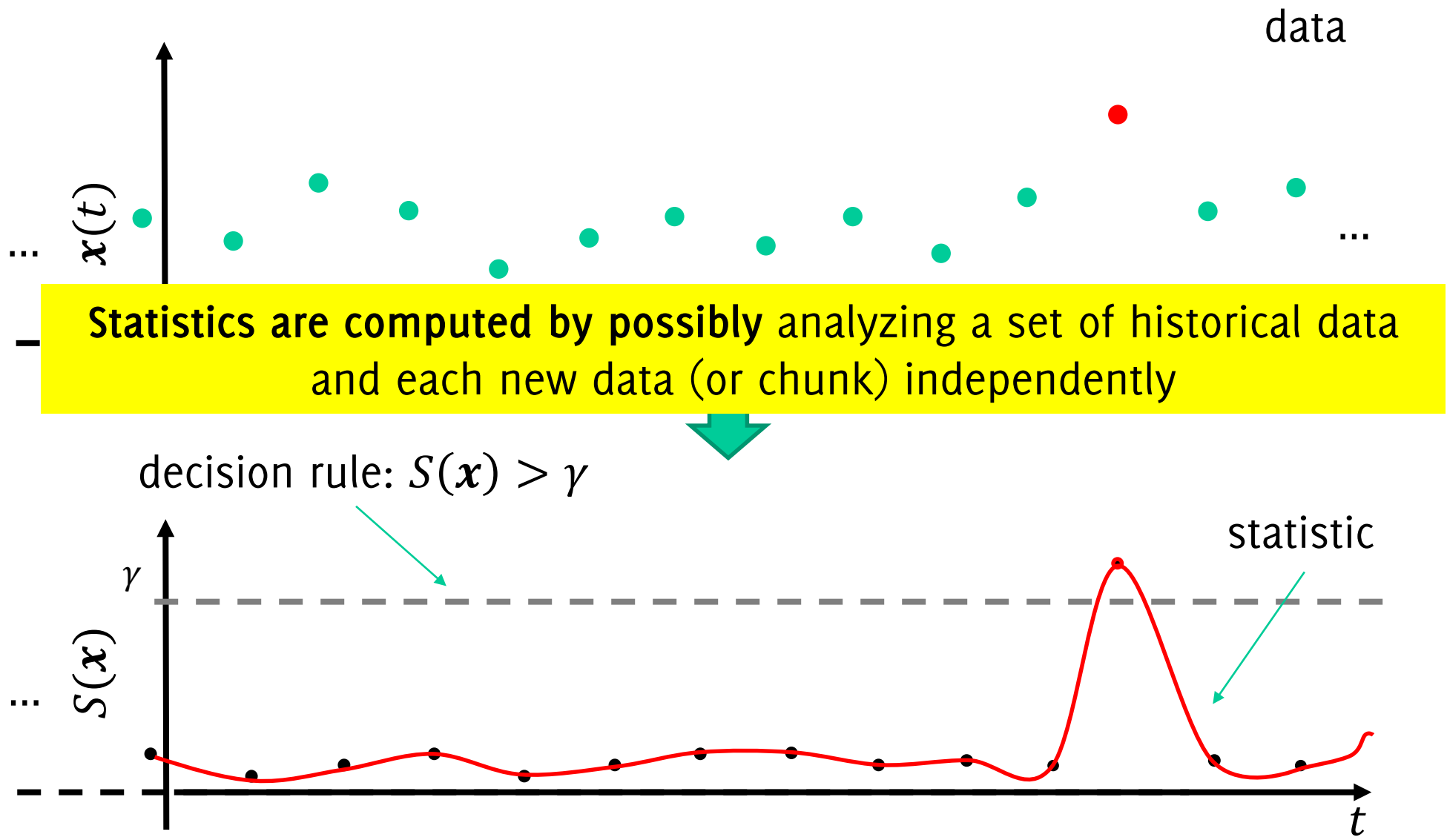
Boracchi, Carrera, ICPR 2020

THE TYPICAL SOLUTIONS



Boracchi, Carrera, ICPR 2020

THE TYPICAL SOLUTIONS



Boracchi, Carrera, ICPR 2020

PERFORMANCE MEASURES

Assessing performance of anomaly detection algorithms

ANOMALY-DETECTION PERFORMANCE

Anomaly detection performance:

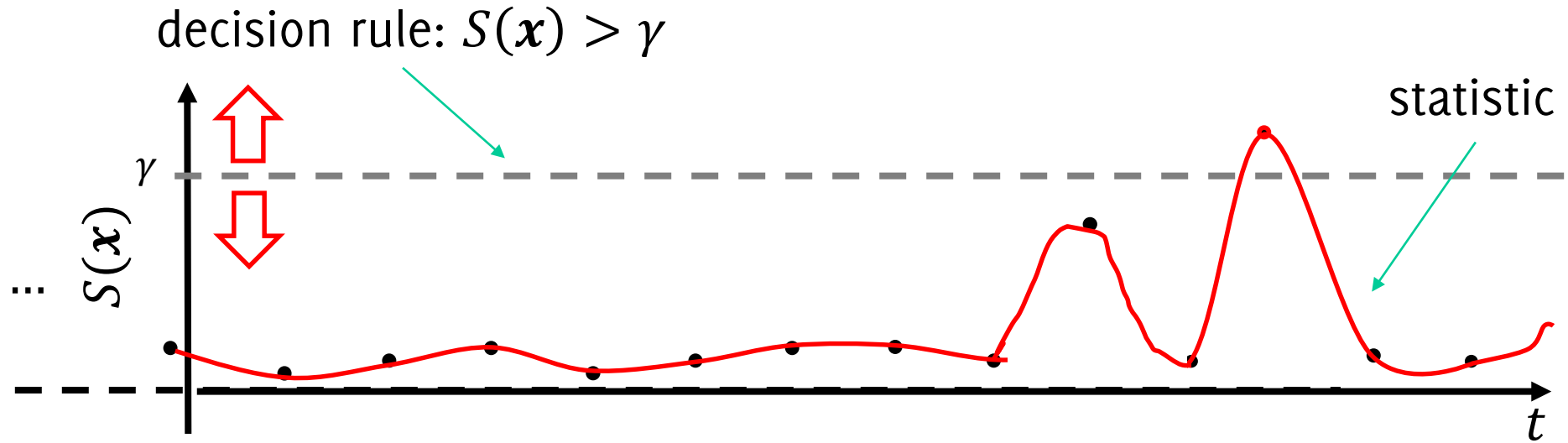
- True positive rate: $TPR = \frac{\#\{\text{anomalies detected}\}}{\#\{\text{anomalies}\}}$
- False positive rate: $FPR = \frac{\#\{\text{normal samples detected}\}}{\#\{\text{normal samples}\}}$

You have probably also heard of

- False negative rate (or miss-rate): $FNR = 1 - TPR$
- True negative rate (or specificity): $TNR = 1 - FPR$
- Precision on anomalies: $\frac{\#\{\text{anomalies detected}\}}{\#\{\text{detections}\}}$
- Recall on anomalies (or sensitivity, hit-rate): TPR

TPR / FPR TRADE-OFF

There is always a **trade-off between TPR and FPR** (and similarly for derived quantities), which is ruled by algorithm parameters



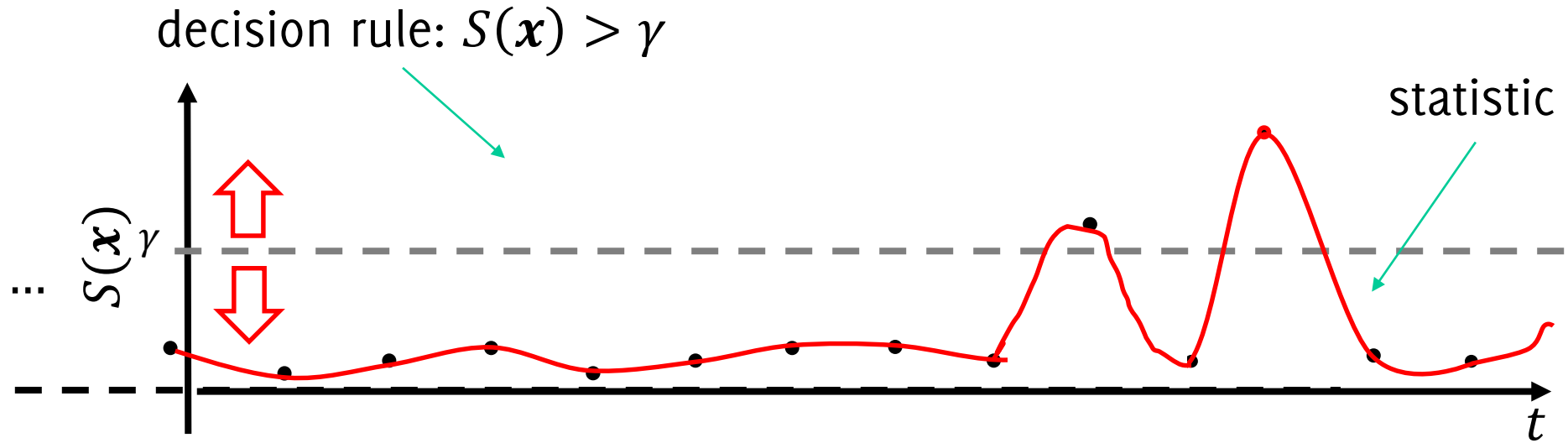
Boracchi, Carrera, ICPR 2020

TPR / FPR TRADE-OFF

There is always a **trade-off between TPR and FPR** (and similarly for derived quantities), which is ruled by algorithm parameters

Decreasing γ increases both true positive rate and false positive rates.

Increasing γ reduces both true positive rate and false positive rates.



ANOMALY-DETECTION PERFORMANCE

There is always a **trade-off between TPR and FPR** (and similarly for derived quantities), which is ruled by algorithm parameters

Thus, to correctly assess performance it is necessary to consider at least **two indicators** (e.g., TPR, FPR)

Indicators combining both TPR and FPR:

$$\text{Accuracy} = \frac{\#\{\text{anomalies detected}\} + \#\{\text{normal samples not detected}\}}{\#\{\text{samples}\}}$$

$$\text{F1 score} = \frac{2\#\{\text{anomalies detected}\}}{\#\{\text{detections}\} + \#\{\text{anomalies}\}}$$

These equal 1 in case of “ideal detector” which detects all the anomalies and has no false positives

ANOMALY-DETECTION PERFORMANCE

Comparing different methods might be tricky since we have to make sure that both have been configured in their best conditions

Testing a large number of parameters lead to the **ROC** (receiver operating characteristic) **curve**

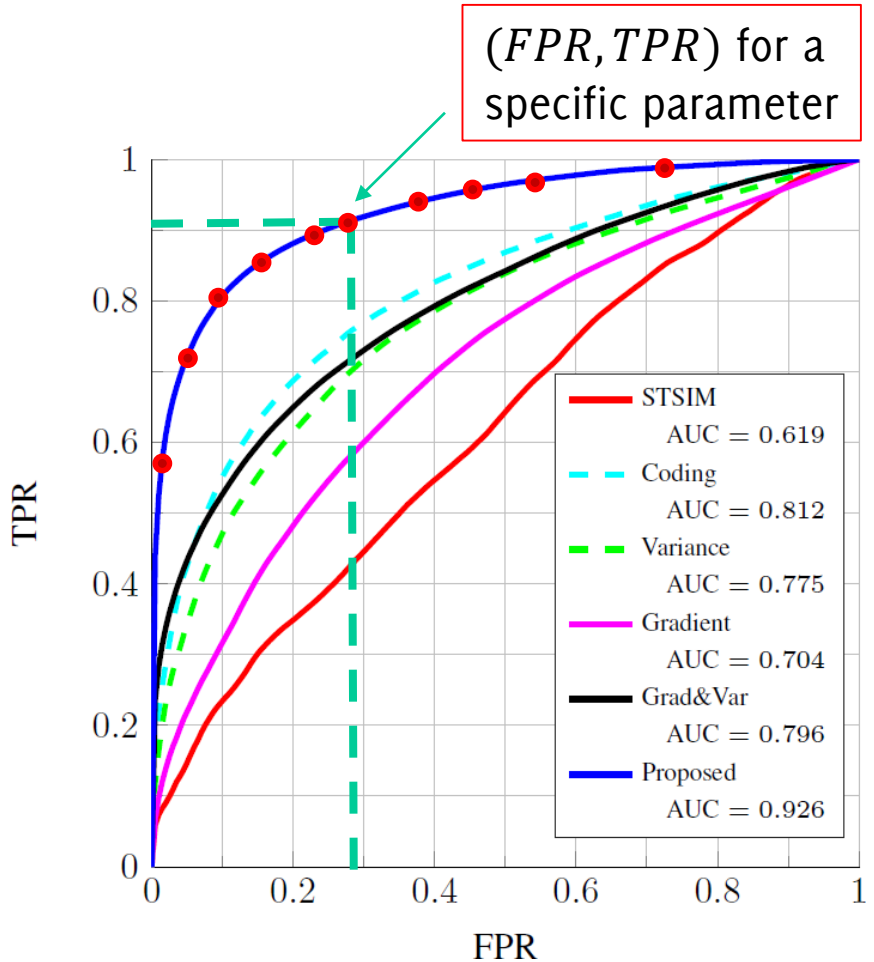
The ideal detector would achieve:

- $FPR = 0\%$,
- $TPR = 100\%$

Thus, the closer to (0,1) the better

The largest the **Area Under the Curve** (AUC), the better

The optimal parameter is the one yielding the point closest to (0,1)



CHANGE-DETECTION PERFORMANCE

In a sequential monitoring scenarios, performance are assessed in terms of the Average Run Length.

In particular, we denote by $\hat{T}$ the detection time and define

$$ARL_0 = \mathop{\mathrm{E}}_{x}\left[\hat{T} \mid \phi_0\right]$$

which is the **expected number of samples before a false alarm** and

$$ARL_1 = \mathop{\mathrm{E}}_{x}\left[\hat{T} \mid \phi_1\right]$$

which is the **expected delay for a detection**

ARL_0 and ARL_1 still depend on the algorithm parameters.

In particular, one configures the CDT to operate at a given ARL_0

M. Basseville, I. V. Nikiforov, "*Detection of Abrupt Changes - Theory and Application*", Prentice-Hall, Inc. April 1993

G. J. Ross, D. K. Tasoulis, N. M. Adams *"Nonparametric monitoring of data streams for changes in location and scale"* Technometrics, 53(4), 379-389, 2012.

CHANGE-DETECTION PERFORMANCE

Unfortunately, it is not always possible to compute ARL_0 and/or ARL_1, in particular for nonparametric CDTs.

Then, one resorts to **performing several simulations** on finite sequences with a change at a known location τ, and computing

The **detection delay**,

$$DD = \mathop{\mathrm{E}}_{x}\left[\hat{T} - \tau \mid \hat{T} \geq \tau, \phi_1\right]$$

and

- $FPR = \frac{\#\{\text{normal sequences where a change was detected}\}}{\#\{\text{normal sequences}\}}$
- $FNR = \frac{\#\{\text{sequences where change was not detected }\}}{\#\{\text{changed sequences}\}}$

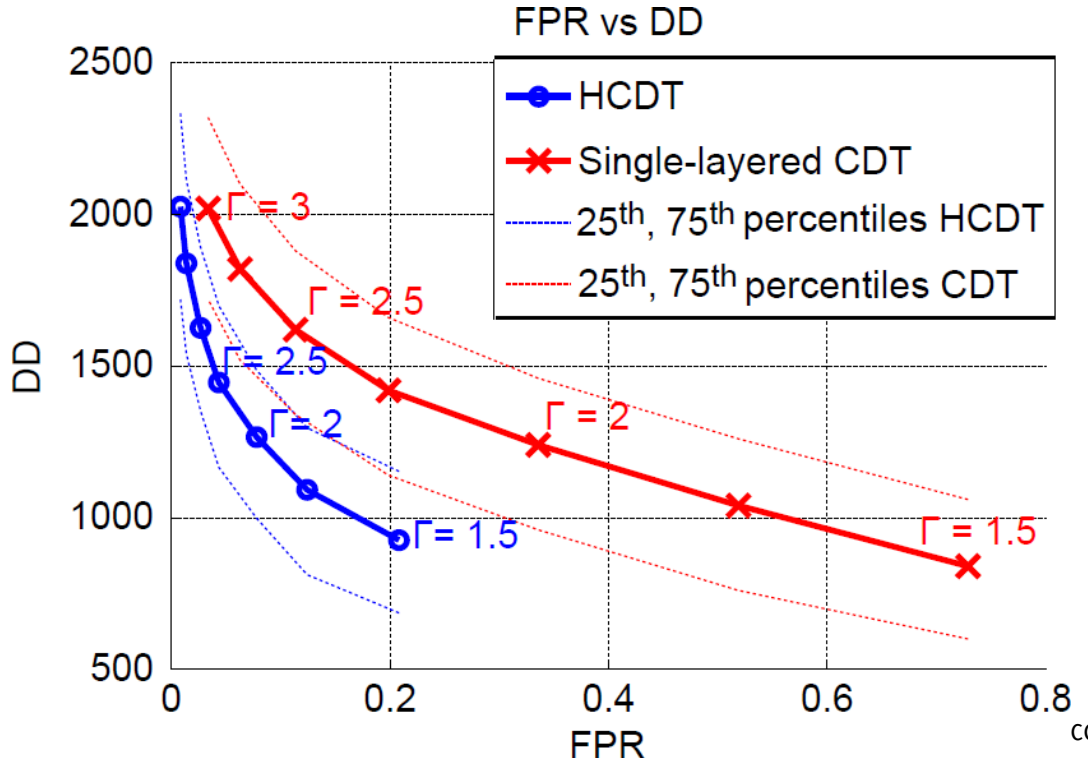
which are defined as in the anomaly detection case, but depend on the sequence length

CHANGE-DETECTION PERFORMANCE

These figures of merit also depend on algorithm parameters.

To perform a fair comparison among different methods one can:

- Generate long enough sequences to have $FNR = 0\%$
- Consider few parameters settings
- Draw FPR-DD curves (similar to ROC): the lower the better



cchi, Carrera, ICPR 2020

ANOMALY/CHANGE DETECTION IN THE IDEAL SETTINGS

...when ϕ_0 and ϕ_1 are known

ONE-SHOT DETECTOR: NEWMAN PEARSON TEST

Assume data are generated from a parametric distribution ϕ_θ and formulate the following hypothesis test

$$H_0: \theta = \theta_0 \text{ vs } H_1: \theta = \theta_1$$

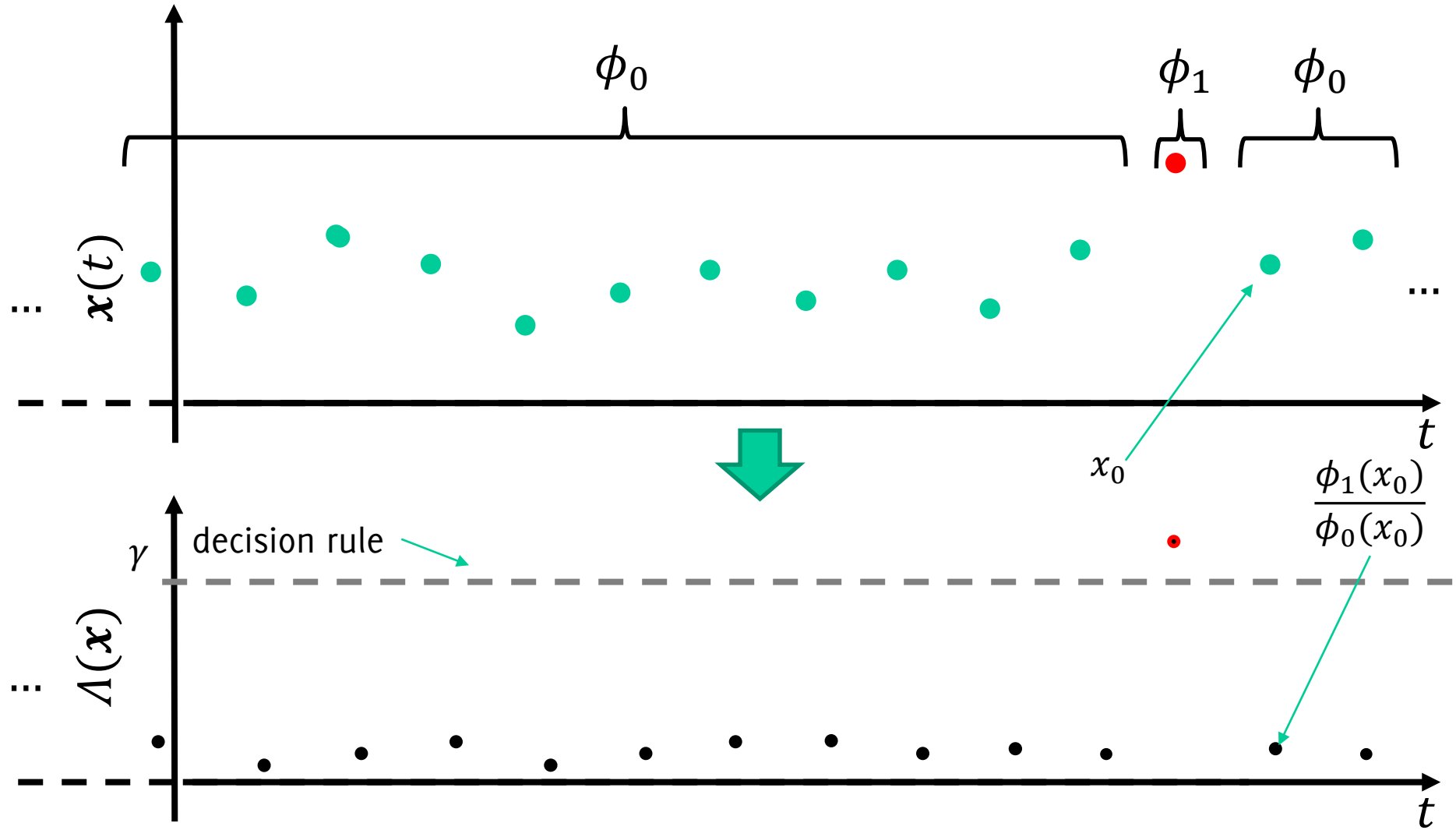
According to the Neumann Pearson lemma, the most powerful **statistic** to detect changes is the **likelihood ratio**

$$\Lambda(x) = \frac{\phi_1(x)}{\phi_0(x)}$$

and the **detection rule** is $\Lambda(x) > \gamma$, where γ is set to **control the false alarm rate** (type I errors of the test).

ONE-SHOT DETECTOR: NEWMAN PEARSON TEST

Outliers can be detected by a threshold on $\Lambda(\boldsymbol{x})$



Boracchi, Carrera, ICPR 2020

THE CUSUM TEST ON THE LIKELIHOOD RATIO

CUSUM involves the calculation of a **CU**mulative **SUM,** which makes it a sequential monitoring scheme.

It can be applied to the log-likelihood ratio:

$$\log\big(\Lambda(x)\big) = \log\left(\frac{\phi_1(x)}{\phi_0(x)}\right) = \begin{cases} < 0 & \text{when} \quad \phi_0(x) > \phi_1(x) \\ > 0 & \text{otherwise} \end{cases}$$

The CUSUM statistic is:

$$S(t) = \max\Big(0, S(\mathrm{t}-1) + \log\big(\Lambda(x(t))\big)\Big)$$

And the decision rule is

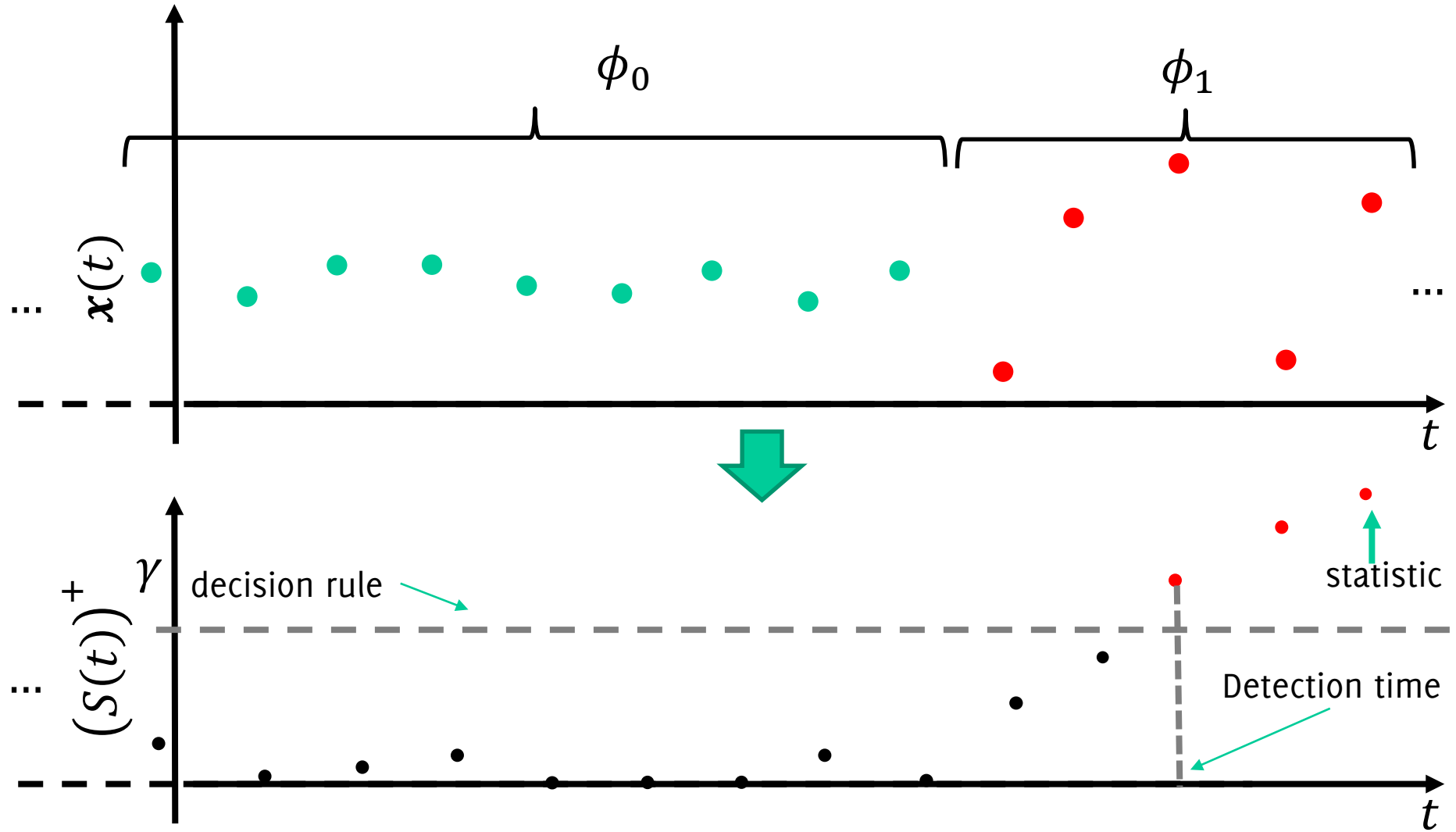
$$S(t) > \gamma$$

M. Basseville, I. V. Nikiforov, "*Detection of Abrupt Changes - Theory and Application*", Prentice-Hall, Inc. April 1993

E. S. Page "*Continuous Inspection Scheme*". Biometrika. 41 (1/2): 100–115 (June 1954).

CUSUM TEST

Outliers can be detected by a threshold on $\Lambda(\boldsymbol{x})$



Boracchi, Carrera, ICPR 2020

PARAMETRIC SEQUENTIAL MONITORING

Quickest Change-Point Detection:

- Detection policies that minimize the expected delay to detection, subject to a fixed ARL_0.
- The CUSUM test is the optimal change-detection test (CDT) when minimizing the maximum delay (at a given ARL_0).
- Other procedures are optimal if we use a different measure for the detection delay or different prior information

M. Basseville, I. V. Nikiforov, "*Detection of Abrupt Changes - Theory and Application*", Prentice-Hall, Inc. April 1993

A. Polunchenko and A. Tartakovsky, *"State-of-the-art in sequential change-point detection,"* Methodology and Computing in Applied Probability, 2012

STATISTICAL APPROACHES TO DETECT ANOMALIES

...when ϕ_0 and ϕ_1 are unknown

ANOMALY DETECTION WHEN ϕ_0 AND ϕ_1 ARE UNKNOWN

Most often, **only a training set** TR is **provided**:

There are three scenarios:

- **Supervised:** Both normal and anomalous training data are provided in TR.
- **Semi-Supervised:** Only normal training data are provided, i.e. no anomalies in TR.
- **Unsupervised:** TR is provided without label.

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

SUPERVISED ANOMALY DETECTION – DISCLAIMER

Most papers and reviews agree that **supervised methods have not to be considered part of anomaly detection,** because:

- Anomalies in general lacks of a statistical coherence
- Not (enough) training samples are provided for anomalies

However,

- Some supervised problems are often referred to as «detection», in case of **severe class imbalance** (e.g. fraud detection)
- **Supervised models can be transferred** in unsupervised settings, in particular for deep learning

Boracchi, Carrera, ICPR 2020

SUPERVISED ANOMALY DETECTION - SOLUTIONS

In **supervised methods** training data are annotated and divided in normal (+) and anomalies (−) :

$$TR = \{(\boldsymbol{x}(t), y(t)), \boldsymbol{x} \in \mathbb{R}^d, y \in \{+, -\} \text{ and } t < t_0\}$$

Solution:

- **Train a two-class classifier** to distinguish normal vs anomalous data.

During training:

- Train a classifier $\mathcal{K}$ from TR.

During testing:

- Compute the classifier output $\mathcal{K}(\boldsymbol{x})$, or
- Set a threshold on the posterior $p_{\mathcal{K}}(-|\boldsymbol{x})$, or
- Select the k −most likely anomalies

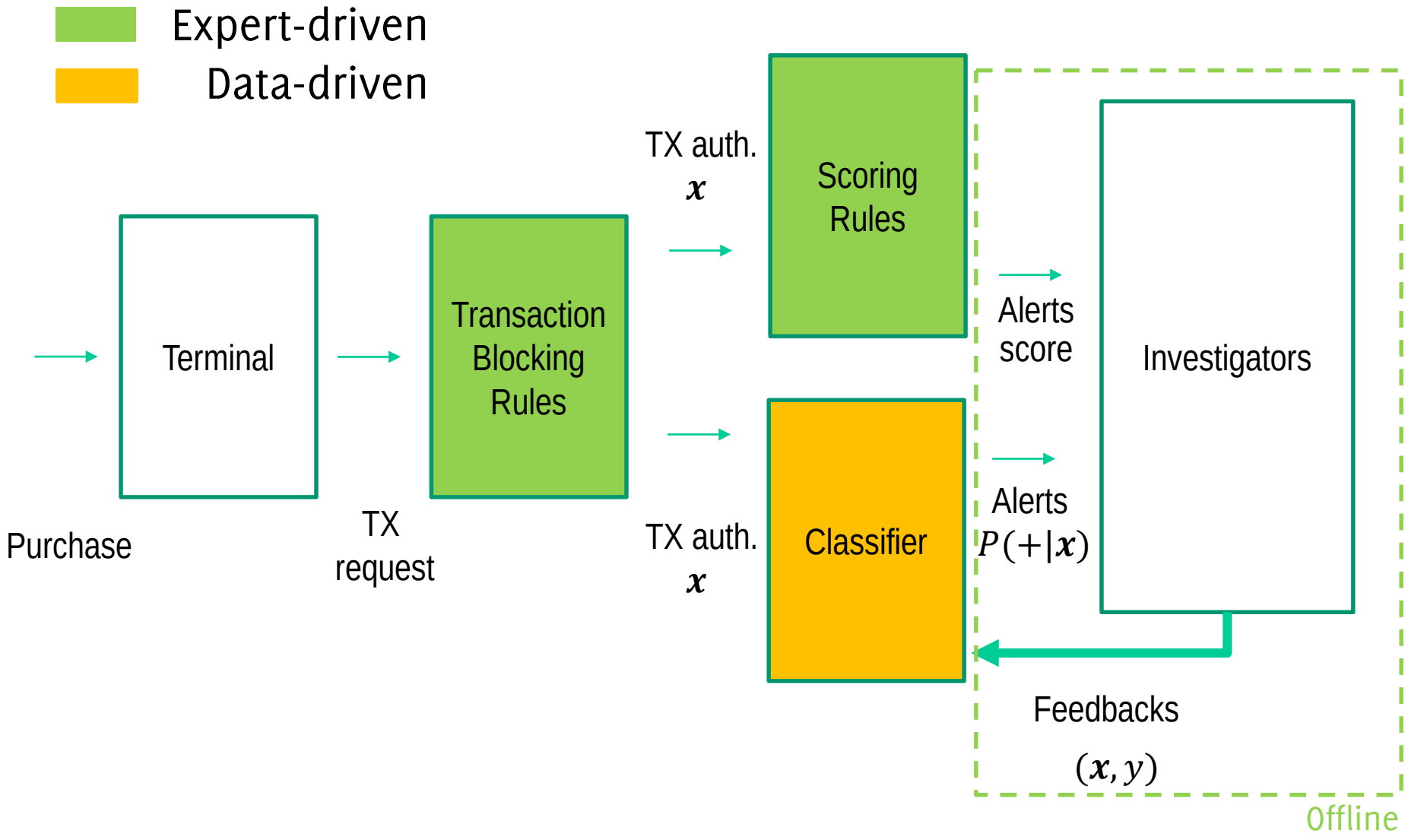
SUPERVISED ANOMALY DETECTION – CHALLENGES

These **classification problems are challenging** because these anomaly-detection settings typically imply:

- **Class Imbalance:** Normal data far outnumber anomalies
- **Concept Drift:** Anomalies might **evolve** over time, thus the few annotated anomalies might not be representative of anomalies occurring during operations
- **Selection Bias:** Training samples are typically selected through a **closed-loop and biased procedure**. Often **only detected anomalies are annotated**, and the vast majority of the stream remain unsupervised. This biases the selection of training samples.

A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi and G. Bontempi, “Credit Card Fraud Detection: a Realistic Modeling and a Novel Learning Strategy”, IEEE TNNL 2017

FRAUD DETECTION: SUPERVISED ANOMALY DETECTION



A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi and G. Bontempi, "Credit Card Fraud Detection: a Realistic Modeling and a Novel Learning Strategy", IEEE TNNL 2017

FRAUD DETECTION: SUPERVISED ANOMALY DETECTION

This is **what typically happens in fraud detection**.

Class Imbalance:

- Frauds are typically less than 1% of genuine transactions

Concept Drift:

- Fraudster always implement new strategies

Sampling Selection Bias:

- Only alerted / reported transactions are controlled and annotated
- Old transactions that have not been disputed are considered genuine transactions

A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi and G. Bontempi, “Credit Card Fraud Detection: a Realistic Modeling and a Novel Learning Strategy”, IEEE TNNL 2017

ANOMALY DETECTION WHEN ϕ_0 AND ϕ_1 ARE UNKNOWN

Most often, **only a training set** TR is **provided**:

There are three scenarios:

- **Supervised:** Both normal and anomalous training data are provided in TR.
- **Semi-Supervised:** Only normal training data are provided, i.e. no anomalies in TR.
- **Unsupervised:** TR is provided without label.

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

SEMI-SUPERVISED ANOMALY DETECTION

In semi-supervised methods the TR is composed of normal data

$$TR = \{x(t), x \sim \phi_0 \text{ and } t < t_0, \}$$

Very practical assumptions:

- **Normal data** are **easy to gather** and the vast majority
- **Anomalous data** are **difficult**/costly **to collect**/select and it would be **difficult to gather a representative training set**
- Training examples in TR might not be **representative of all the possible anomalies** that can occur

All in all, it is often **safer** to **detect any data departing from** the **normal conditions**

Semi-supervised anomaly-detection methods are also referred to as **novelty-detection methods**

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

M. A. Pimentel, D. A. Clifton, L. Clifton, L. Tarassenko. *"A review of novelty detection"* Signal Processing, 99, 215-249 (2014)

DENSITY-BASED METHODS

Density-Based Methods: ***Normal*** *data occur in* ***high probability regions*** *of a stochastic model, while* ***anomalies*** *occur in the* ***low probability regions*** *of the model*

During training: $\hat{\phi}_0$ can be **estimated** from the training set

$$TR = \{x(t), x \sim \phi_0 \text{ and } t < t_0, \}$$

- parametric models (e.g., Gaussian mixture models)
- nonparametric models (e.g. KDE, histograms)

During testing:

- Anomalies are detected as data yielding $\hat{\phi}_0(\boldsymbol{x}) < \eta$

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

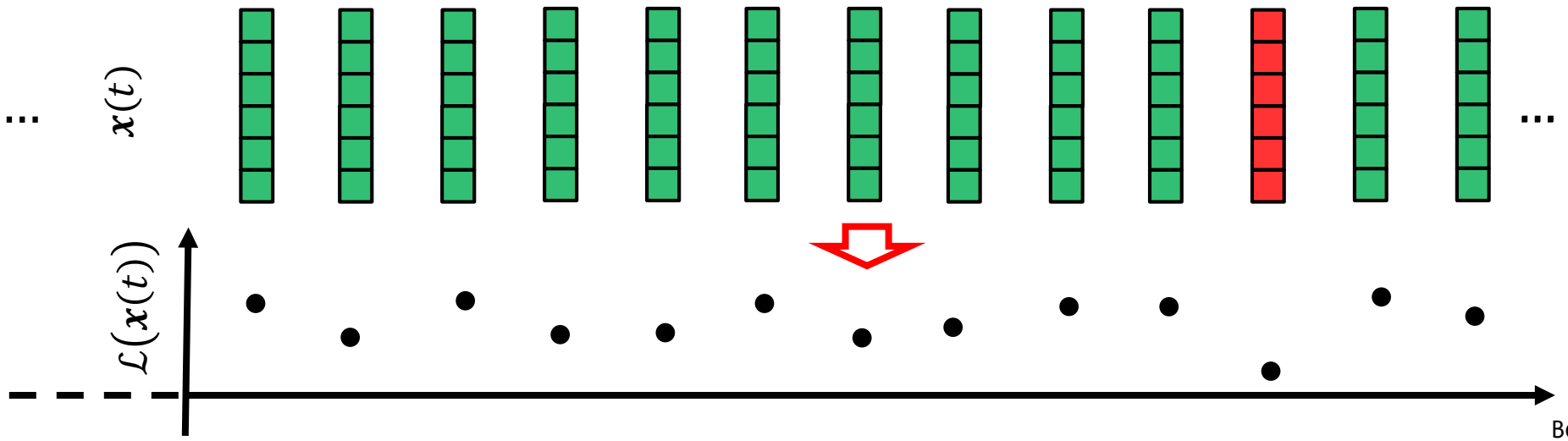
DENSITY-BASED METHODS: MONITORING THE LOG-LIKELIHOOD

Monitoring the log-likelihood of data w.r.t $\hat{\phi}_0$ allow to address anomaly-detection problems in multivariate data

1. During training, estimate $\hat{\phi}_0$ from TR
2. During testing, compute

$$\mathcal{L}\big(\boldsymbol{x}(t)\big) = \log(\hat{\phi}_0(\boldsymbol{x}(t)))$$

3. Monitor $\{\mathcal{L}\big(\boldsymbol{x}(t)\big),\ t = 1, \ldots\}$



Boracchi, Carrera, ICPR 2020

DENSITY-BASED METHODS: MONITORING THE LOG-LIKELIHOOD

Monitoring the log-likelihood of data w.r.t $\hat{\phi}_0$ allow to address anomaly-detection problems in multivariate data

1. During training, estimate $\hat{\phi}_0$ from TR
2. During testing, compute

$$\mathcal{L}\big(\boldsymbol{x}(t)\big) = \log(\hat{\phi}_0(\boldsymbol{x}(t)))$$

3. Monitor $\big\{\mathcal{L}\big(\boldsymbol{x}(t)\big),\ t = 1, \ldots\big\}$

This is quite a popular approach in either anomaly and change detection algorithms

L. I. Kuncheva, "*Change detection in streaming multivariate data using likelihood detectors*," IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 5, 2013.

X. Song, M. Wu, C. Jermaine, and S. Ranka, "*Statistical change detection for multidimensional data*," in Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD), 2007.

J. H. Sullivan and W. H. Woodall, "*Change-point detection of mean vector or covariance matrix shifts using multivariate individual observations*," IIE transactions, vol. 32, no. 6, 2000.

C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "*Change Detection in Multivariate Datastreams: Likelihood and Detectability Loss*" IJCAI 2016, New York, USA, July 9 - 13

DENSITY-BASED METHODS

Advantages:

- $\hat{\phi}_0(\boldsymbol{x})$ indicates how safe a detection is (like a p-value)
- If the density estimation process is robust to outliers, it is possible to tolerate few anomalous samples in TR
- in relatively small dimensions, you might use non-parametric models like histograms

Challenges:

- **It is challenging to fit models for high-dimensional data**
- Histograms traditionally suffer of **curse of dimensionality** when d increases
- Often the **1D histograms** of the marginals are monitored, **ignoring the correlations** among components

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

DOMAIN-BASED METHODS

Domain-based methods: ***Estimate a boundary around normal data****, rather than the density of normal data.*

A **drawback of density-estimation methods** is that they are meant to be accurate in high-density regions, while anomalies live in low-density ones.

One-Class SVM are domain-based methods defined by **the normal samples at the periphery of the distribution**.

Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., Platt, J. C. "*Support Vector Method for Novelty Detection*". In NIPS 1999 (Vol. 12, pp. 582-588).

Tax, D. M., Duin, R. P. "*Support vector domain description*". Pattern recognition letters, 20(11), 1191-1199 (1999)

M. A. Pimentel, D. A. Clifton, L. Clifton, L. Tarassenko. *"A review of novelty detection"* Signal Processing, 99, 215-249 (2014)

ONE-CLASS SVM (SCHÖLKOPF ET AL. 1999)

Idea: define boundaries by estimating a **binary function** f that **captures regions of the input space where density is higher**.

As in support vector methods, f is **defined in the feature space** F and **decision boundaries are defined by a few support vectors** (i.e., a few normal data).

Let $\psi(\boldsymbol{x})$ the feature associated to $\boldsymbol{x}$, f is defined as

$$f(\boldsymbol{x}) = \text{sign}(< w, \psi(\boldsymbol{x}) > -\rho)$$

Where the hyperplane parameters w, ρ are optimized to yield a **function that is positive on most training samples.** Thus in the feature space, normal points can be separated from the origin.

A linear separation in the feature space corresponds to a **variety of nonlinear boundaries in the space of $\boldsymbol{x}$**.

B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, J. C. Platt "*Support Vector Method for Novelty Detection*". In NIPS 1999 (Vol. 12, pp. 582-588).

ONE-CLASS SVM (TAX AND DUIN 1999)

Boundaries of normal region can be also defined by an **hypersphere that, in the feature space, encloses most of the normal data** i.e., $\psi(\boldsymbol{x})$ for $\boldsymbol{x} \in TR$.

Similar detection formulas hold, measuring the **distance in the feature space from the sphere center**.

The sphere center can be defined in terms of support vectors.

Remarks: In both one-class approaches, the amount of samples that falls within the margin (outliers) is controlled by regularization parameters.

This parameter regulates the number of outliers in the training set and the detector sensitivity.

D. M. Tax, R. P. Duin, *"Support vector domain description"*. Pattern recognition letters, 20(11), 1191-1199 (1999)

ANOMALY DETECTION WHEN ϕ_0 AND ϕ_1 ARE UNKNOWN

Most often, **only a training set** TR is **provided**:

There are three scenarios:

- **Supervised:** Both normal and anomalous training data are provided in TR.
- **Semi-Supervised:** Only normal training data are provided, i.e. no anomalies in TR.
- **Unsupervised:** TR is provided without label.

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

UNSUPERVISED ANOMALY-DETECTION

The training set TR might contain **both normal and anomalous data**. However, **no labels** are provided

$$TR = \{x(t), t < t_0\}$$

Underlying assumption: ***anomalies are rare*** *w.r.t. normal data* TR

In principle:

- Density/Domain based methods that are **robust to outliers** can be applied in an unsupervised scenario
- Unsupervised methods can be **improved whenever labels are available**

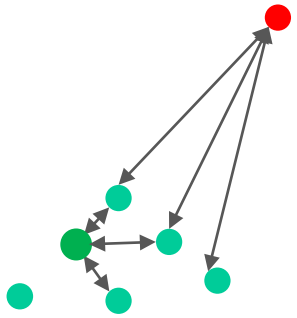
DISTANCE-BASED METHODS

Distance-based methods: ***normal*** *data fall in* ***dense neighborhoods****, while* ***anomalies*** *are* ***far from their closest neighbors****.*

A critical aspect is the **choice of the similarity measure** to use.

Anomalies are detected by **monitoring**:

- **distance** between each data and its k **–nearest neighbor**
- the **density** of each data **relatively to its neighbors**



M. Zhao, V. Saligrama, *"Anomaly detection with score functions based on nearest neighbor graphs"*. NIPS 2009

A. Zimek, E. Schubert, H. Kriegel. *"A survey on unsupervised outlier detection in high-dimensional numerical data"* SADM 2012

M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers", in International Conference on Management of data, 2000

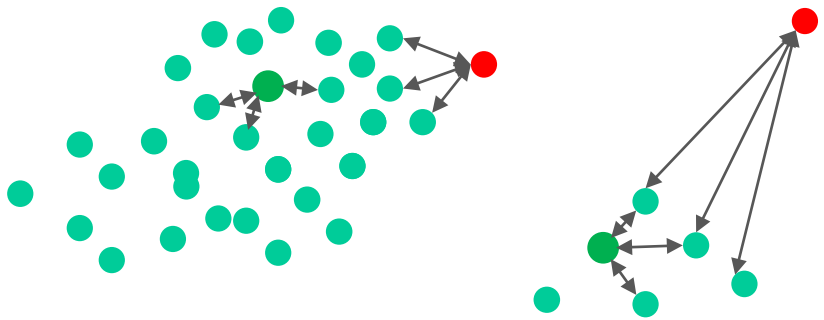
DISTANCE-BASED METHODS

Distance-based methods: ***normal*** *data fall in* ***dense neighborhoods****, while* ***anomalies*** *are* ***far from their closest neighbors****.*

A critical aspect is the **choice of the similarity measure** to use.

Anomalies are detected by **monitoring**:

- **distance** between each data and its k **–nearest neighbor**
- the **above distance** considered **relatively to neighbors**
- whether they do not belong to **clusters**, or are at the cluster periphery, or belong to small and sparse clusters



V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

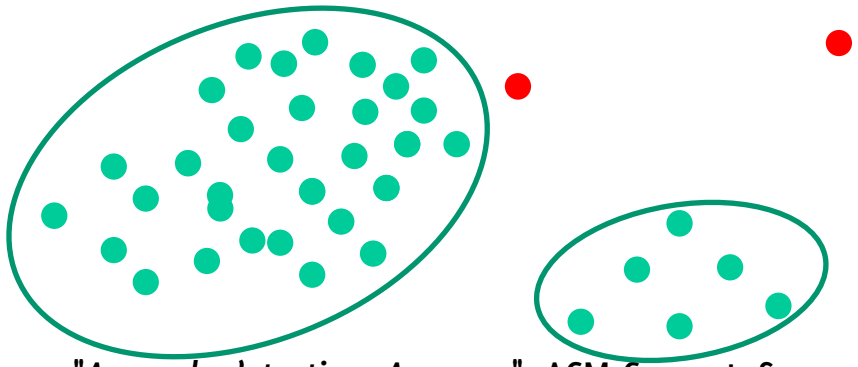
DISTANCE-BASED METHODS

Distance-based methods: ***normal*** *data fall in* ***dense neighborhoods****, while* ***anomalies*** *are* ***far from their closest neighbors****.*

A critical aspect is the **choice of the similarity measure** to use.

Anomalies are detected by **monitoring**:

- **distance** between each data and its k **–nearest neighbor**
- the **above distance** considered **relatively to neighbors**
- whether they do not belong to **clusters**, or are at the cluster periphery, or belong to small and sparse clusters

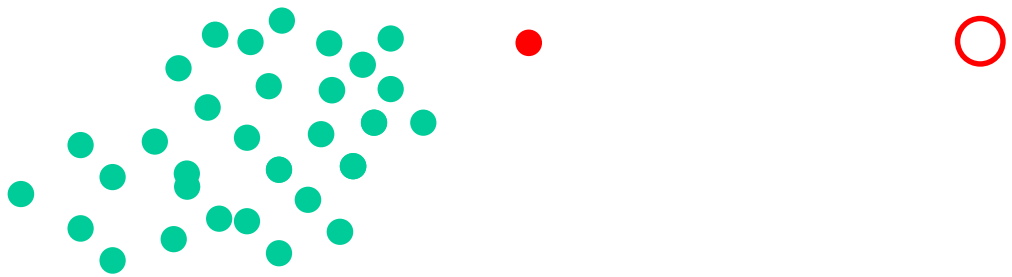


V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

ISOLATION FOREST

Builds upon the rationale that
"*anomalies are easier to separate from the rest of normal data*"

This idea is implemented very efficiently through **a forest of binary trees** that are constructed via an iterative procedure



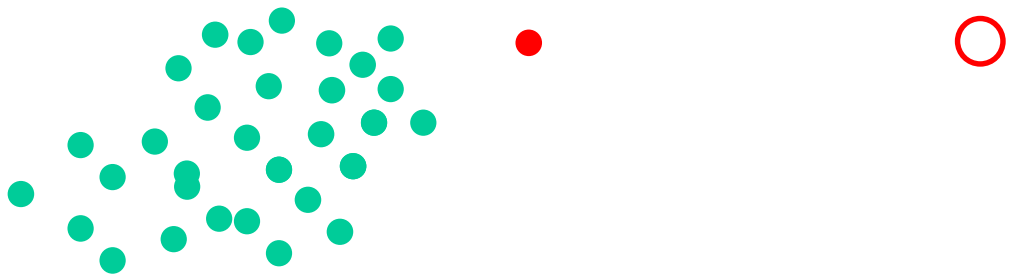
F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

Builds upon the rationale that

"*anomalies are easier to separate from the rest of normal data*"

This idea is implemented very efficiently through **a forest of binary trees** that are constructed via an iterative procedure



Randomly choose

1. a component x_i

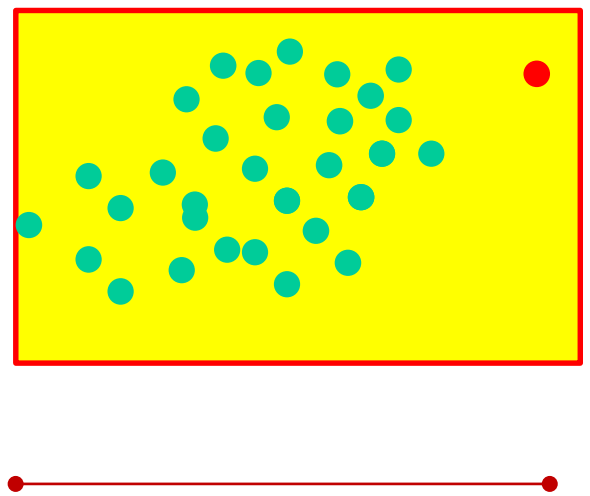
F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

Builds upon the rationale that

"*anomalies are easier to separate from the rest of normal data*"

This idea is implemented very efficiently through **a forest of binary trees** that are constructed via an iterative procedure



Randomly choose

1. a component x_i

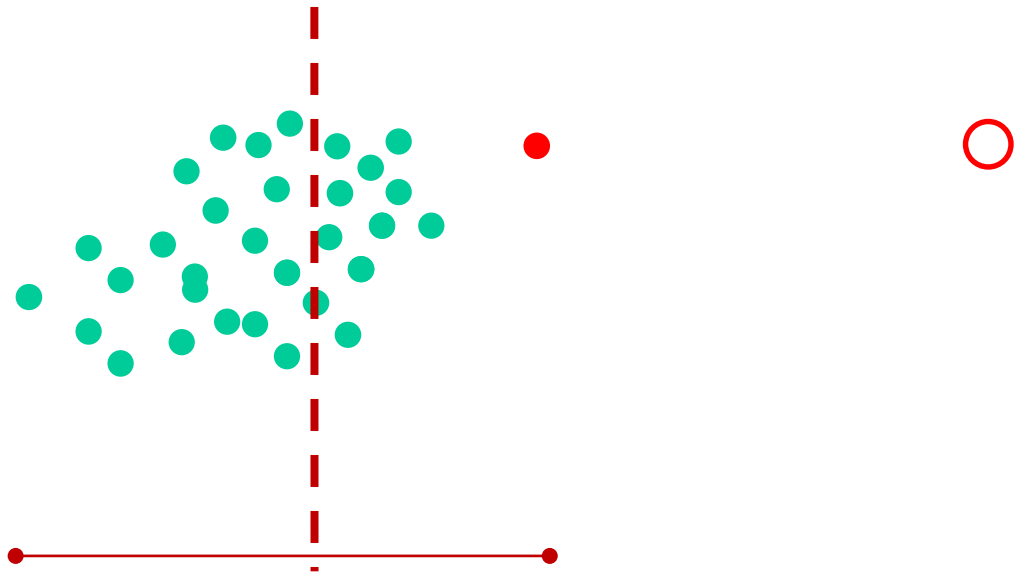
F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

Builds upon the rationale that

"*anomalies are easier to separate from the rest of normal data*"

This idea is implemented very efficiently through **a forest of binary trees** that are constructed via an iterative procedure



Randomly choose

1. a component x_i
2. a value in the range of projections of TR over the i-th component

This yields a splitting

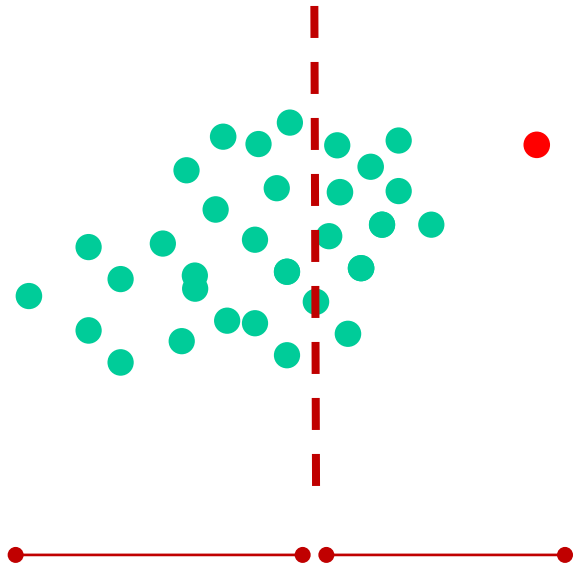
F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

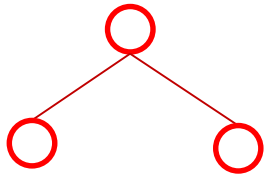
ISOLATION FOREST

Builds upon the rationale that

"*anomalies are easier to separate from the rest of normal data*"

This idea is implemented very efficiently through **a forest of binary trees** that are constructed via an iterative procedure





This yields a splitting criteria

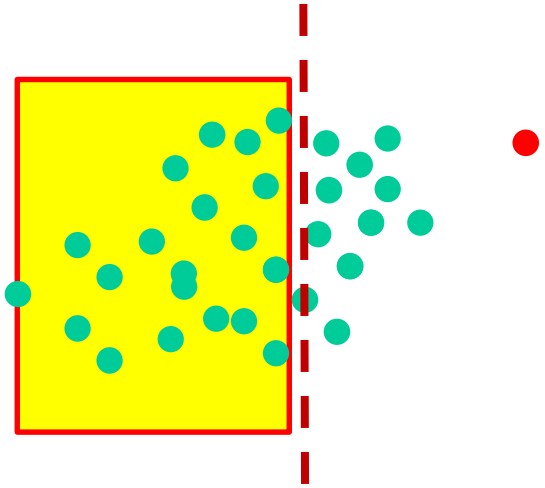
F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

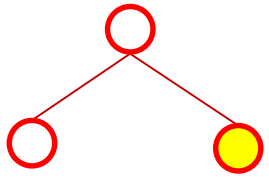
ISOLATION FOREST

Builds upon the rationale that

"*anomalies are easier to separate from the rest of normal data*"

This idea is implemented very efficiently through **a forest of binary trees** that are constructed via an iterative procedure





Repeat the procedure on each node:

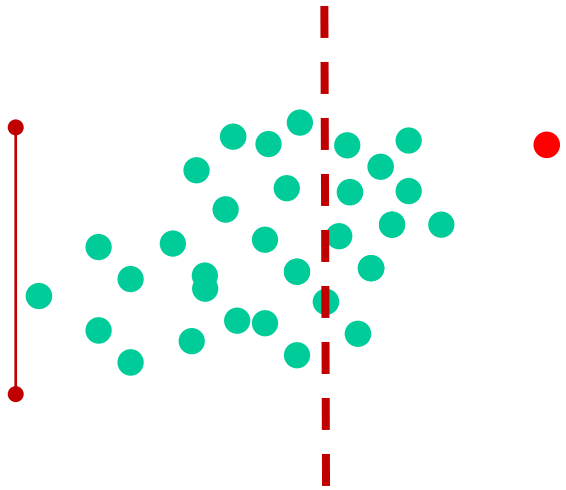
F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

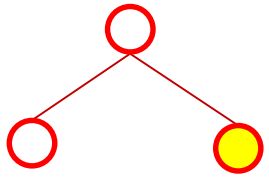
ISOLATION FOREST

Builds upon the rationale that

"*anomalies are easier to separate from the rest of normal data*"

This idea is implemented very efficiently through **a forest of binary trees** that are constructed via an iterative procedure





Repeat the procedure on each node: Randomly select a component

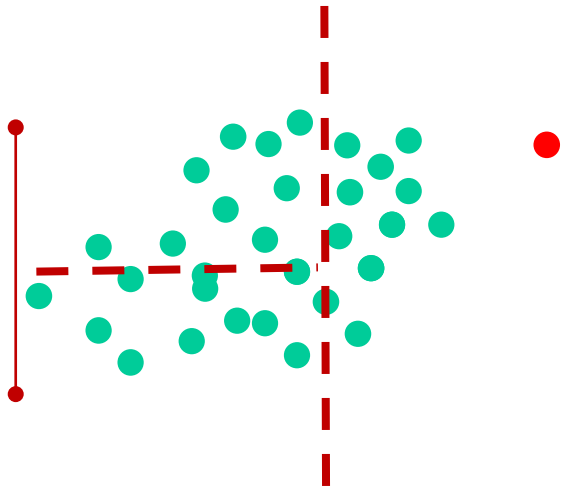
F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

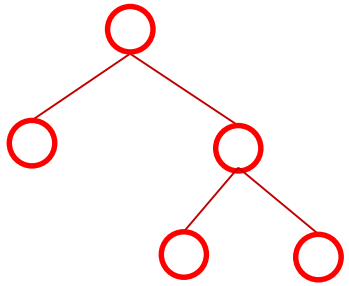
ISOLATION FOREST

Builds upon the rationale that

"*anomalies are easier to separate from the rest of normal data*"

This idea is implemented very efficiently through **a forest of binary trees** that are constructed via an iterative procedure





Repeat the procedure on each node: Randomly select a component and a cut point

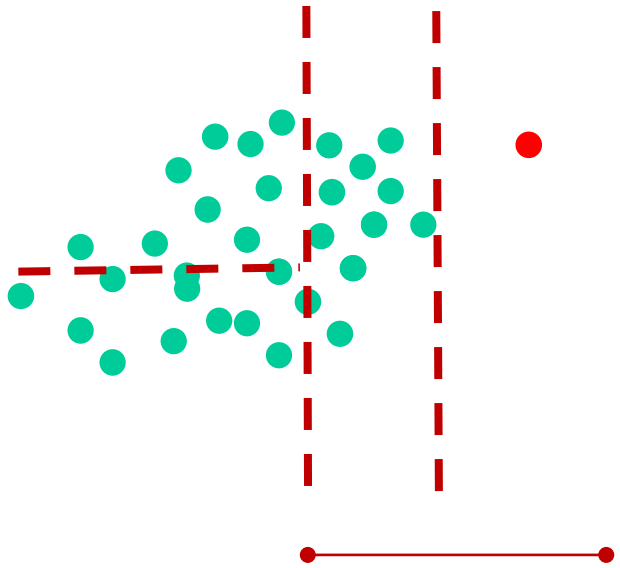
F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

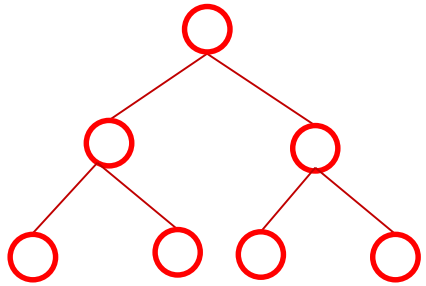
ISOLATION FOREST

Builds upon the rationale that

"*anomalies are easier to separate from the rest of normal data*"

This idea is implemented very efficiently through **a forest of binary trees** that are constructed via an iterative procedure





Randomly choose a component and a value within the range and define a splitting criteria

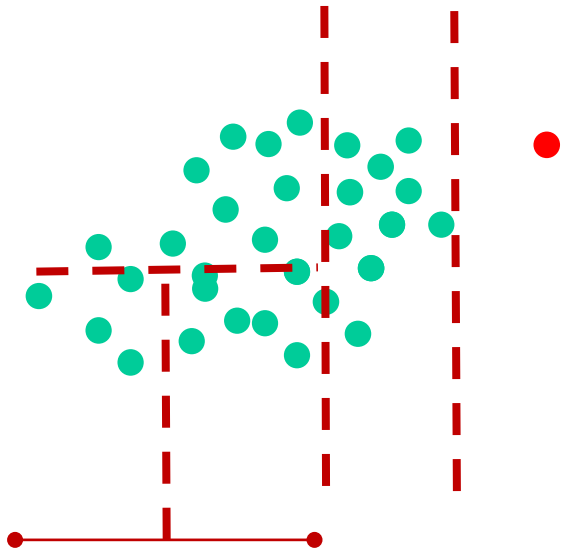
F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

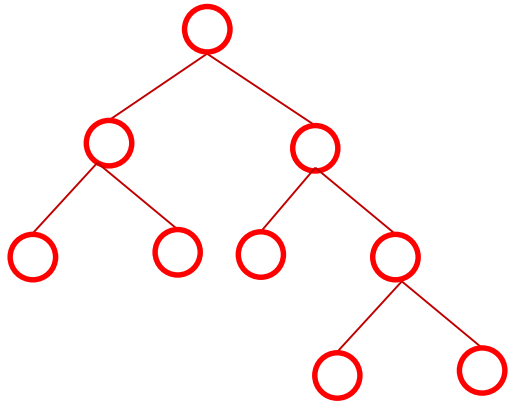
ISOLATION FOREST

Builds upon the rationale that

"*anomalies are easier to separate from the rest of normal data*"

This idea is implemented very efficiently through **a forest of binary trees** that are constructed via an iterative procedure





Repeat the procedure on the nodes: Randomly select a component and a cut point

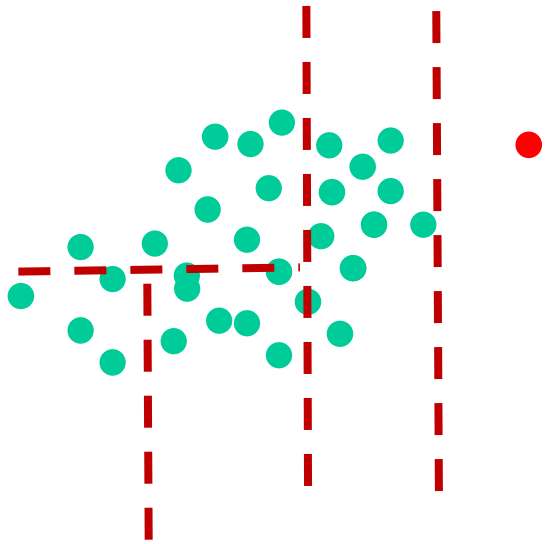
F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

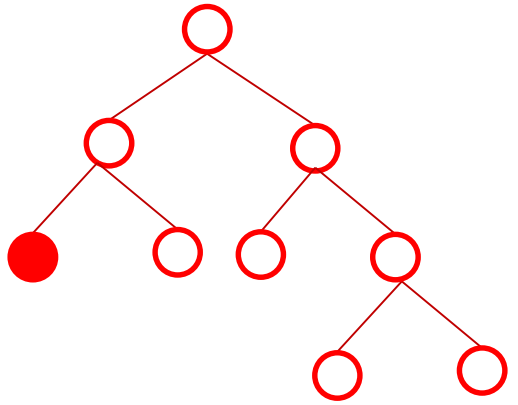
ISOLATION FOREST

Builds upon the rationale that

"*anomalies are easier to separate from the rest of normal data*"

This idea is implemented very efficiently through **a forest of binary trees** that are constructed via an iterative procedure





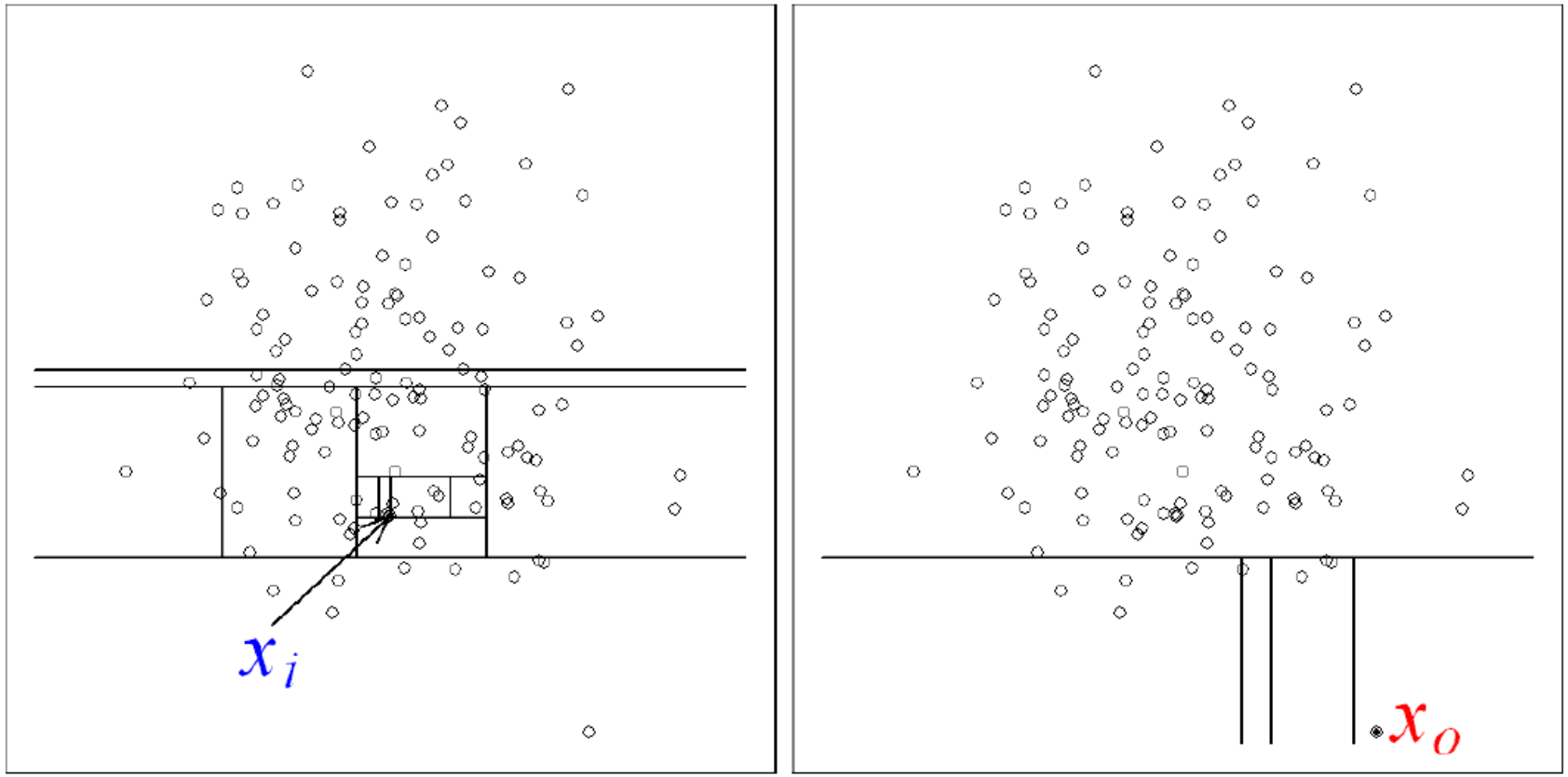
Anomalies lies in leaves close to the root.

F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

An anomalous point (x_0) can be easily isolated

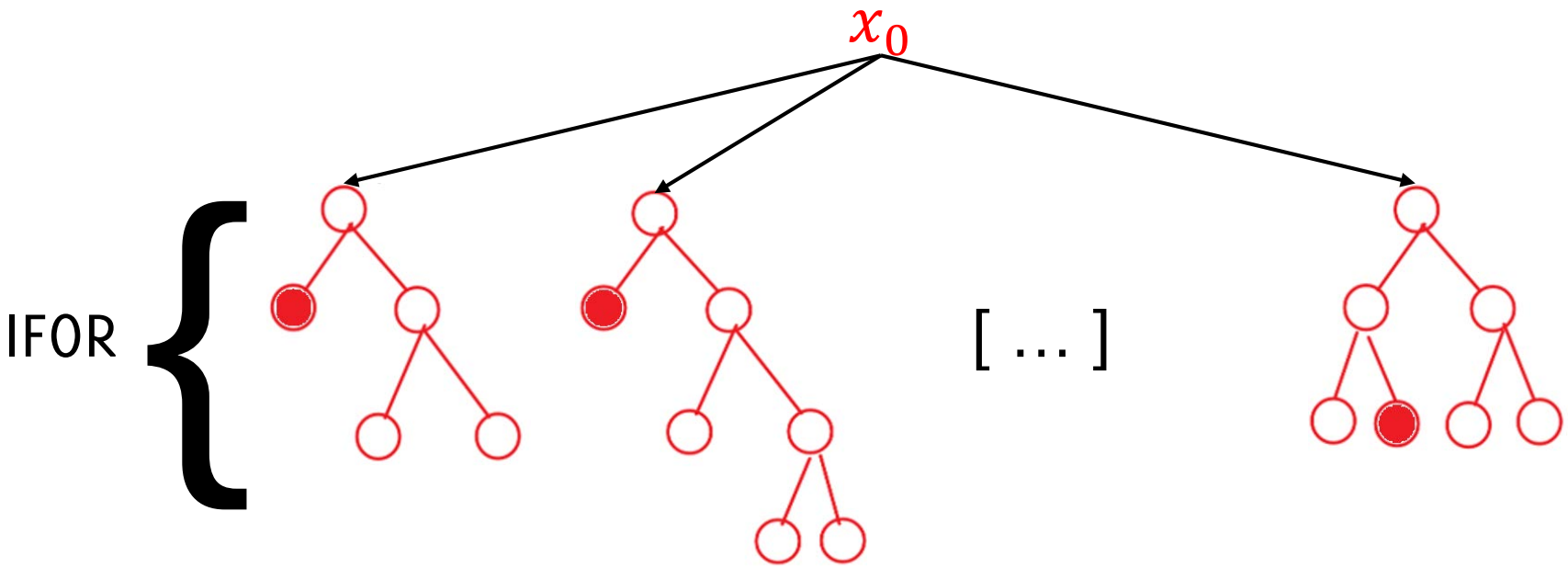
Genuine points (x_i) are instead difficult to isolate.



F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

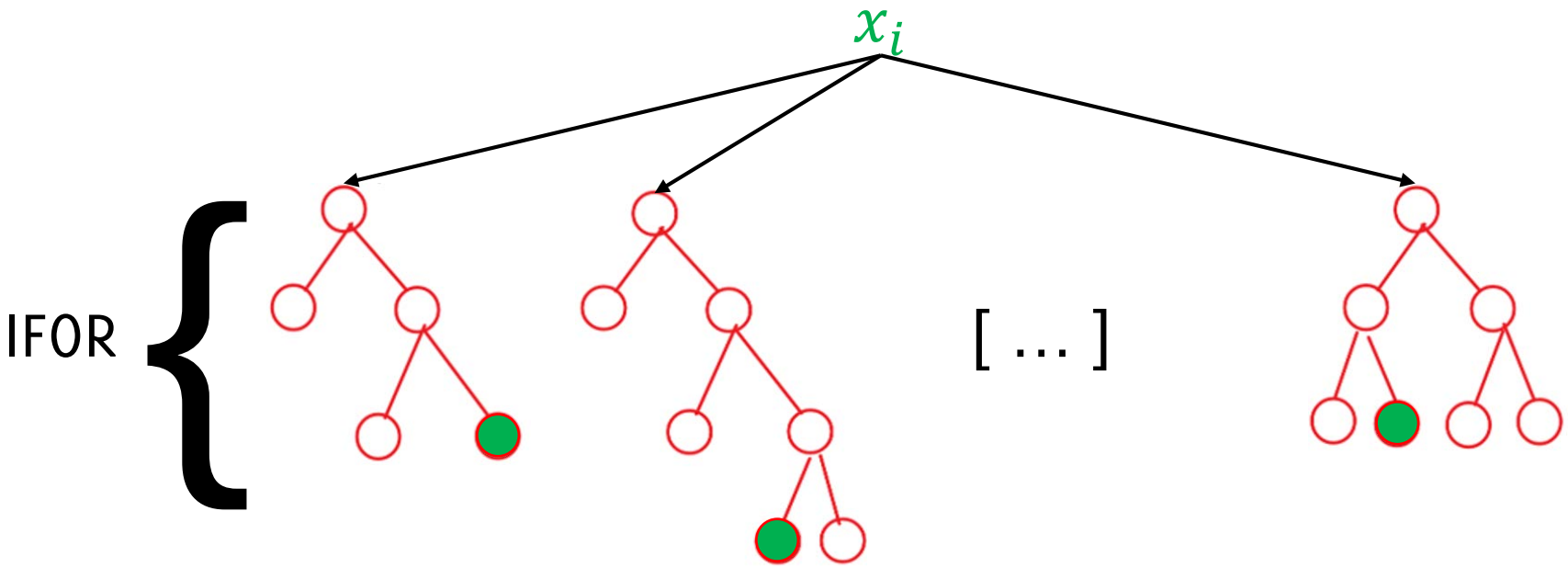
Anomalies



F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

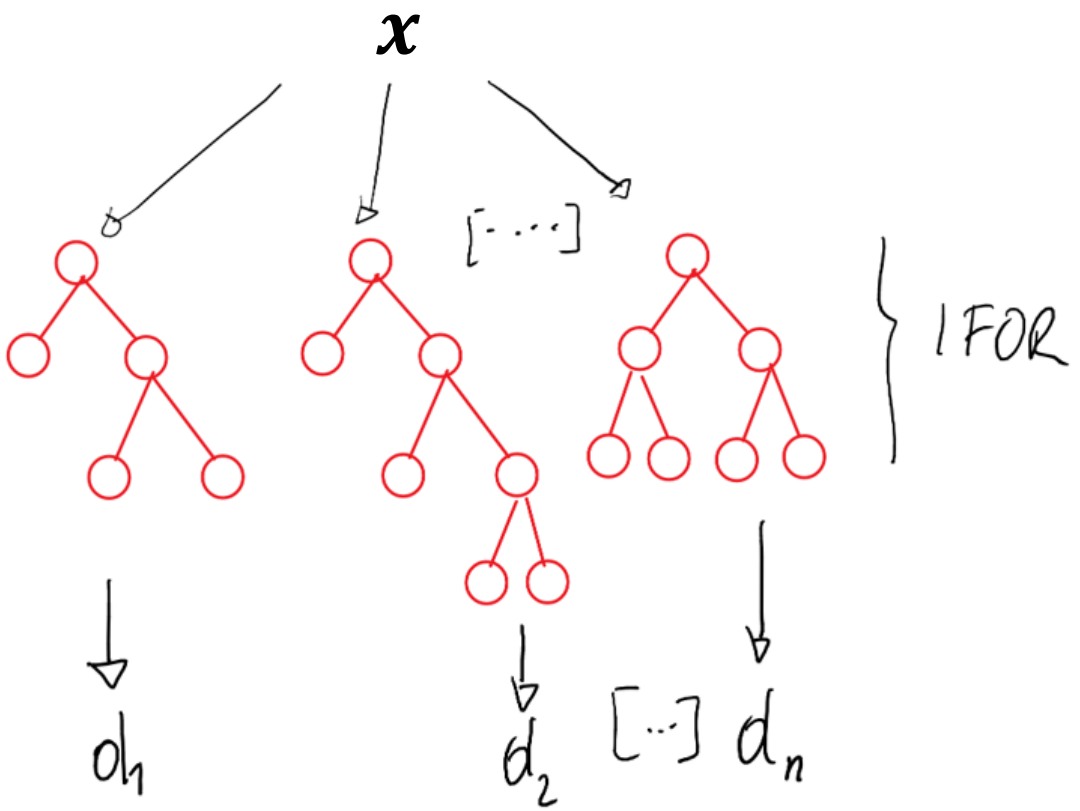
Normal data



F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST: TESTING

Compute $E\big(h(\boldsymbol{x})\big)$, the **average path length** among all the trees in the forest, of a test sample $\boldsymbol{x}$



F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

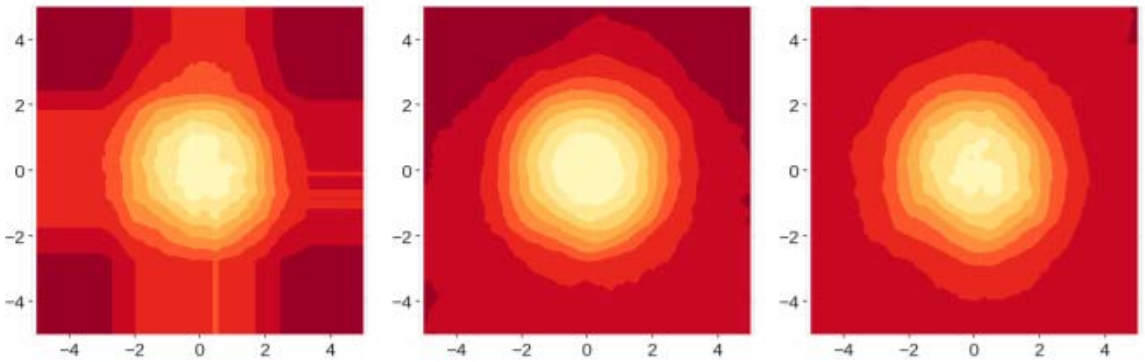
ISOLATION FOREST: TESTING

A test sample is identified as **anomalous** when:

$$\mathcal{A}(\boldsymbol{x}) = 2^{-\frac{E(h(\boldsymbol{x}))}{c(n)}} > \gamma$$

- n : number of sessions in TR
- $c(n)$: average path length of unsuccessful search in Binary

Several extensions including **EIF** (Extendend Isolation Forest) modify the splitting criteria to yield anomaly scores map that better conform to normal data



Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

S. Hariri, M. C. Kind, and R. J. Brunner, "Extended isolation forest", TKDE, 2019.

Any Questions?

STATISTICAL APPROACHES TO DETECT CHANGES

...change detection when ϕ_0 and ϕ_1 are unknown

THE CHANGE AND ANOMALY DETECTION PROBLEMS

Anomaly detection:

$$\boldsymbol{x}(t) \sim \begin{cases} \phi_0 & \text{normal data} \\ \phi_1 & \text{anomalies} \end{cases}$$

Change detection:

$$\boldsymbol{x}(t) \sim \begin{cases} \phi_0 & t < \tau \quad \text{in control state} \\ \phi_1 & t \geq \tau \quad \text{out of control state} \end{cases}$$

In change detection we have to take into account the temporal dimension when monitoring the stream $\{\boldsymbol{x}(t), t = 1, \ldots\}$

CHANGE DETECTION APPROACHES

Parametric Settings:

- The Change-Point Formulation

Non-parametric Settings:

- The Change-Point Formulation
- Change-Detection by Histograms
- Change-Detection by Monitoring Features
- Hierarchical Change-Detection Tests

CHANGE DETECTION IN PARAMETRIC SETTINGS: CPM

Parametric settings:

ϕ_0 and ϕ_1 are known up to their parameters (θ_0 and θ_1), thus the change $\phi_0 \rightarrow \phi_1$ corresponds to a change $\theta_0 \rightarrow \theta_1$

Change-Point Methods (CPM) are **sequential** monitoring schemes that **extend** traditional **parametric hypothesis tests**

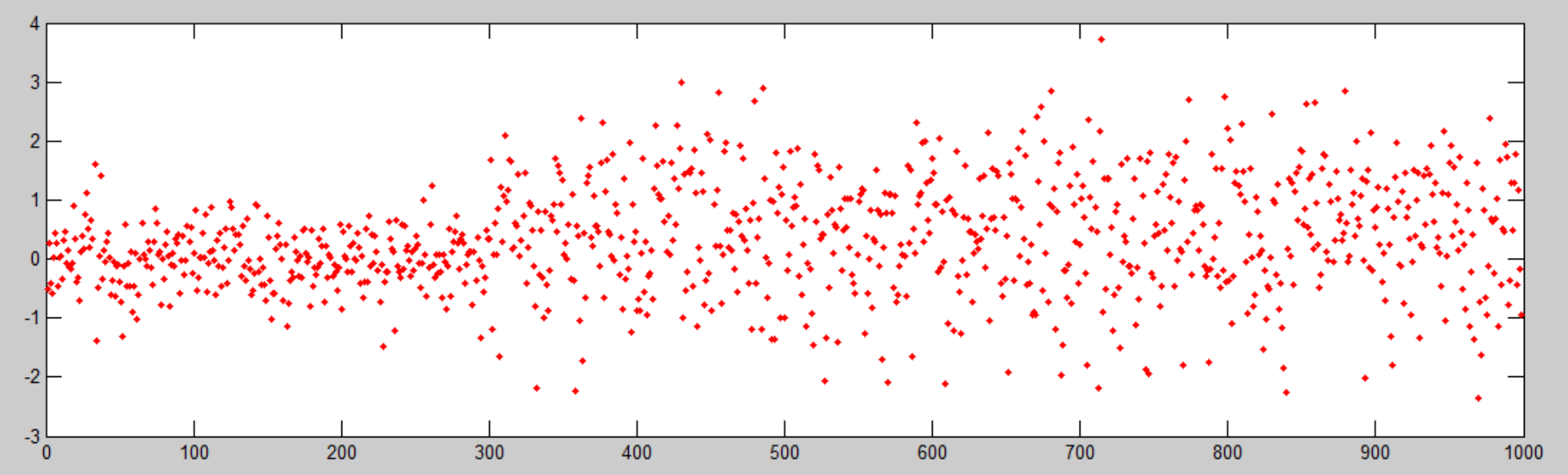
These assumptions hold in some **quality control** application, but sometimes the **change is unpredictable** (e.g. θ_1 it is unknown)

The basic functioning of CPM is illustrated for offline monitoring, but CMP can be **iterated to perform online change detection** (sequential monitoring).

Hawkins, D. M., and Zamba, K. D. *"Statistical process control for shifts in mean or variance using a changepoint formulation"* Technometrics 2005

Ross, G. J. "Sequential change detection in the presence of unknown parameters". Statistics and Computing, 24(6), 1017-1030, 2014

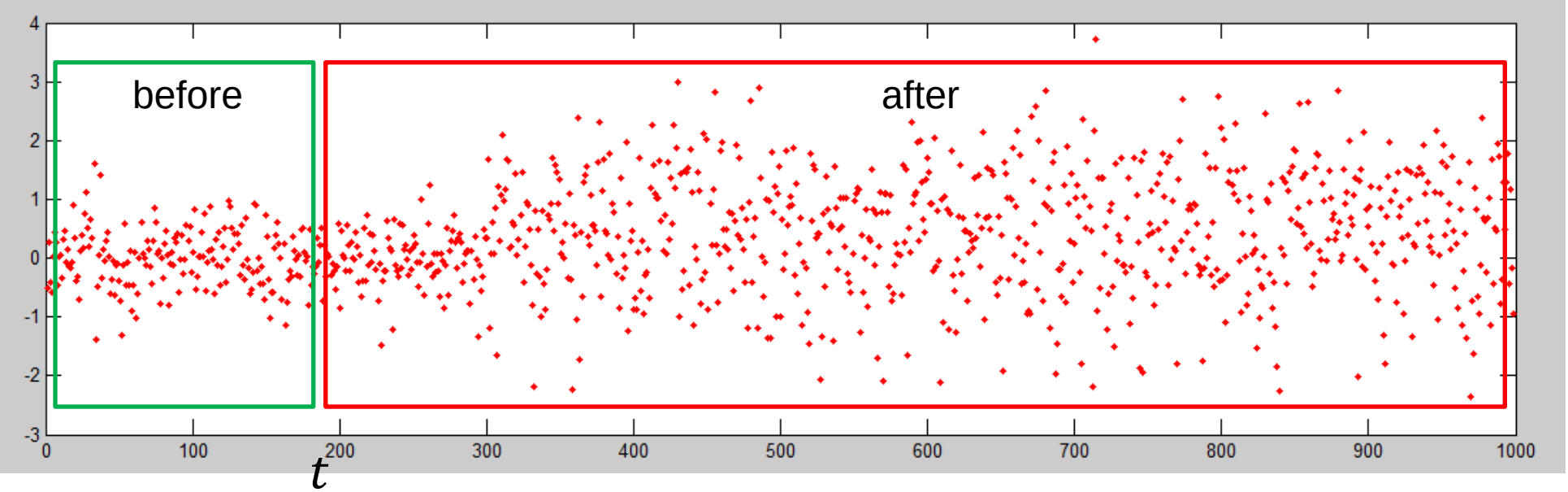
ILLUSTRATION OF CHANGE POINT METHOD (CPM)



Assume a sequence of 1000 points is given and we want to find the change point τ inside (offline analysis)

Boracchi, Carrera, ICPR 2020

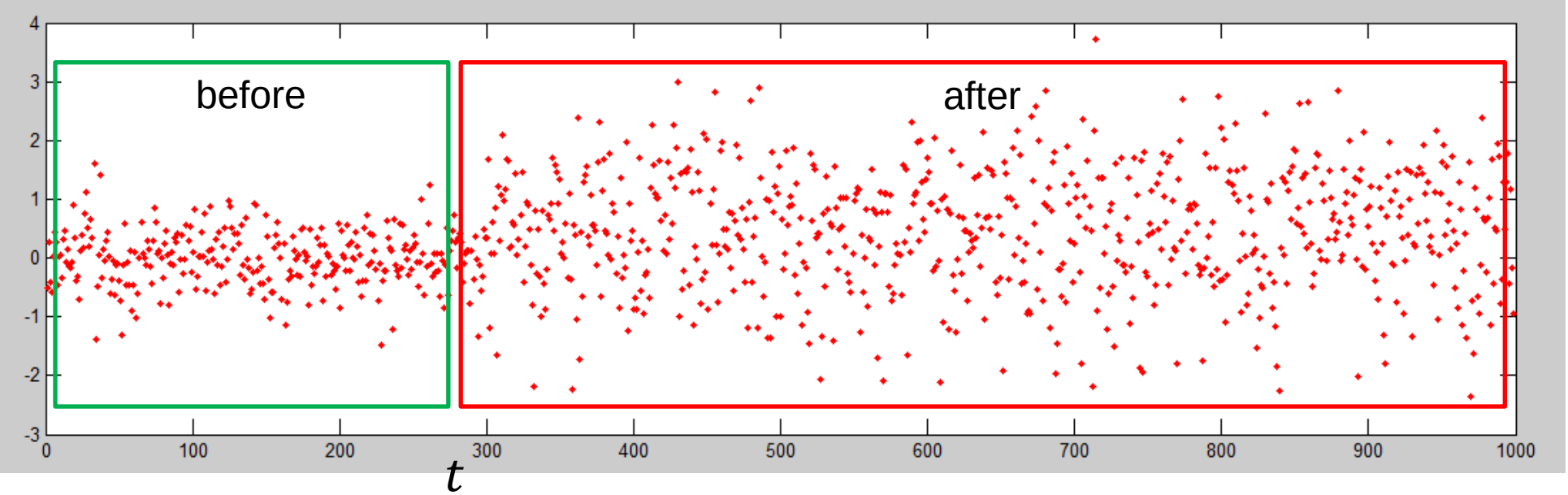
ILLUSTRATION OF CHANGE POINT METHOD (CPM)



- Test a single point t to be a change point
- Split the dataset in two sets «before» and «after»
- Compute a test statistic $\mathcal{T}$ to determine whether the two sets are from the same distribution (e.g. same mean)
- Repeat the procedure and store the value of the statistic

Boracchi, Carrera, ICPR 2020

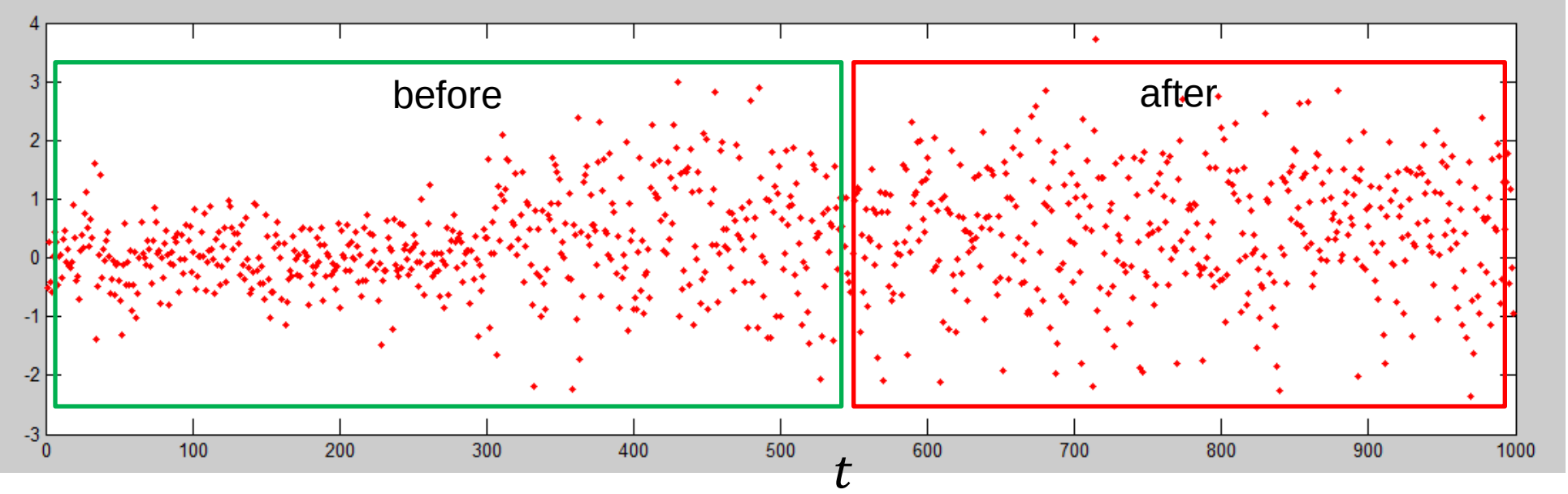
ILLUSTRATION OF CHANGE POINT METHOD (CPM)



- Test a single point t to be a change point
- Split the dataset in two sets «before» and «after»
- Compute a test statistic $\mathcal{T}$ to determine whether the two sets are from the same distribution (e.g. same mean)
- Repeat the procedure and store the value of the statistic

Boracchi, Carrera, ICPR 2020

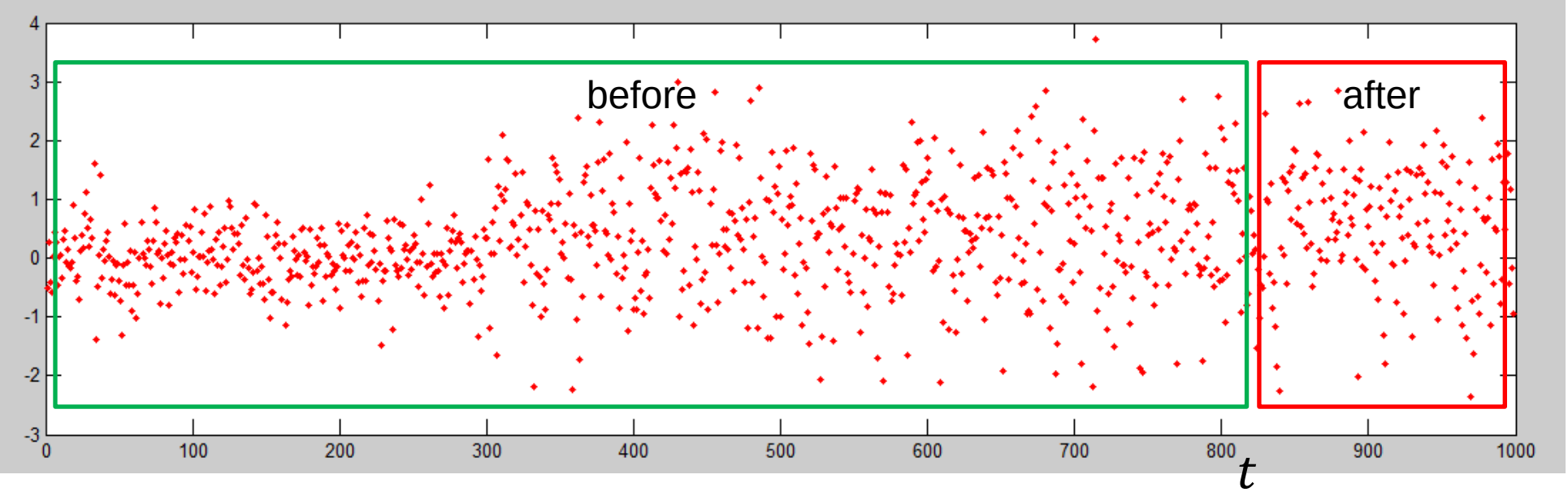
ILLUSTRATION OF CHANGE POINT METHOD (CPM)



- Test a single point t to be a change point
- Split the dataset in two sets «before» and «after»
- Compute a test statistic $\mathcal{T}$ to determine whether the two sets are from the same distribution (e.g. same mean)
- Repeat the procedure and store the value of the statistic

Boracchi, Carrera, ICPR 2020

ILLUSTRATION OF CHANGE POINT METHOD (CPM)



- Test a single point t to be a change point
- Split the dataset in two sets «before» and «after»
- Compute a test statistic $\mathcal{T}$ to determine whether the two sets are from the same distribution (e.g. same mean)
- Repeat the procedure and store the value of the statistic

Boracchi, Carrera, ICPR 2020

ILLUSTRATION OF CHANGE POINT METHOD (CPM)

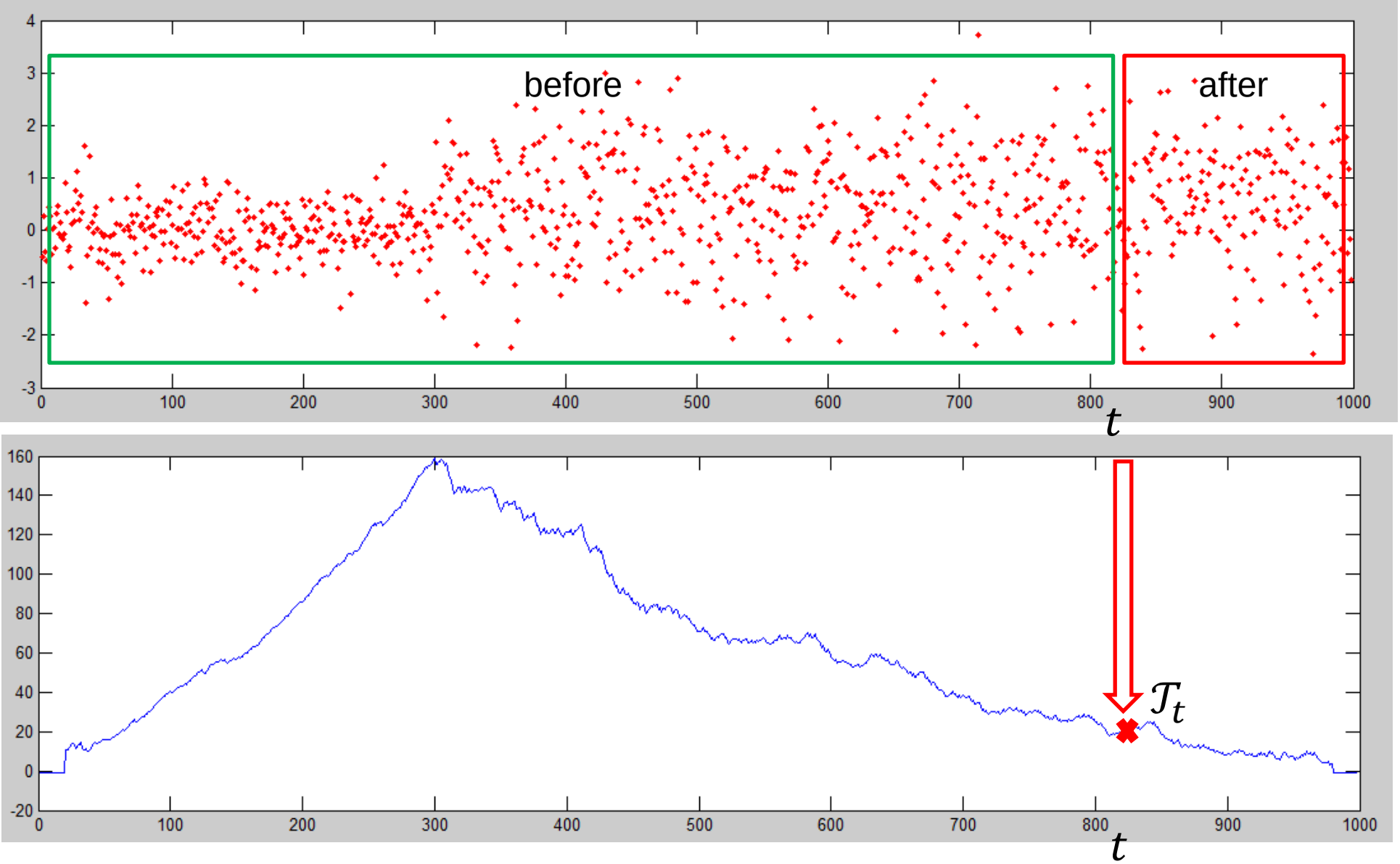


ILLUSTRATION OF CHANGE POINT METHOD (CPM)

The point where the statistic achieves its maximum is the most likely position of the change-point

As in hypothesis testing, it possible to set a threshold $h_{1000,\alpha}$ for $\mathcal{T}_{\max,1000}$ by setting to α the probability of type I errors.

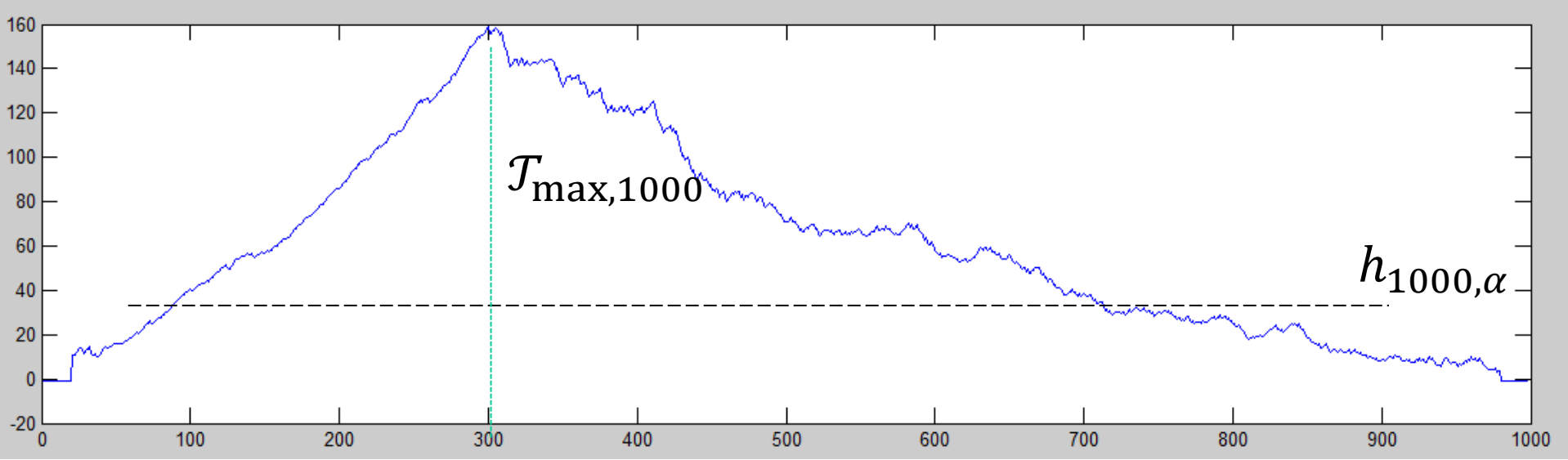


ILLUSTRATION OF CHANGE POINT METHOD (CPM)

It is possible to extend the CPM framework to online monitoring.

At each time t:

- we compute the statistic $\mathcal{T}_{\max,t}$
- we detect a change if $\mathcal{T}_{\max,t} > h_t$

The thresholds $\{h_t\}$ have to be set to guarantee the ARL_0

Hawkins, D. M., and Zamba, K. D. *"Statistical process control for shifts in mean or variance using a changepoint formulation"* Technometrics 2005

CHANGE DETECTION APPROACHES

Parametric Settings:

- The Change-Point Formulation

Non-parametric Settings:

- The Change-Point Formulation
- Change-Detection by Histograms
- Change-Detection by Monitoring Features
- Hierarchical Change-Detection Tests

CPM IN NONPARAMETRIC SETTINGS

Both ϕ_0 and ϕ_1 are unknown, thus **the change $\phi_0 \to \phi_1$ is completely unpredictable**

One viable option consists in using **nonparametric statistics**, like:

- Mann-Whitney,
- Mood,
- Lepage,
- Kolmogorov-Smirnov,
- Cramer von Mises,

which do not require any information about ϕ_0 or ϕ_1.

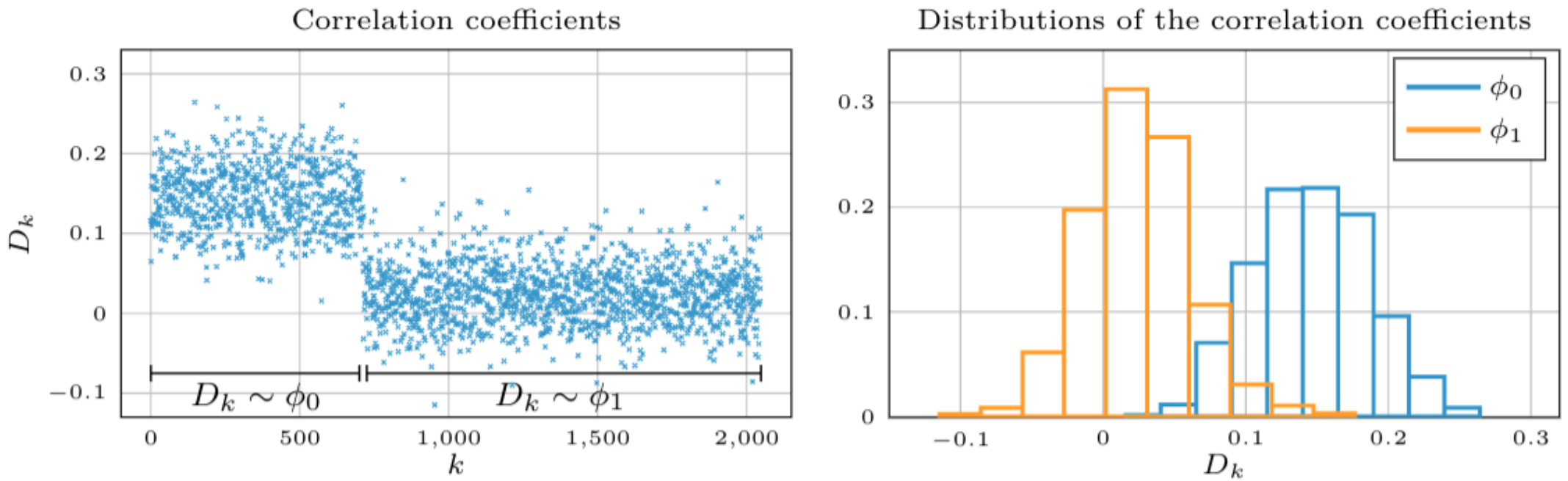
Ross, G. J., Tasoulis, D. K., Adams, N. M. *"Nonparametric monitoring of data streams for changes in location and scale"* Technometrics, 53(4), 379-389, 2012.

A REAL WORLD MONITORING EXAMPLE

In cryptography sequential side channel attacks reconstruct the secret key one bit at a time.

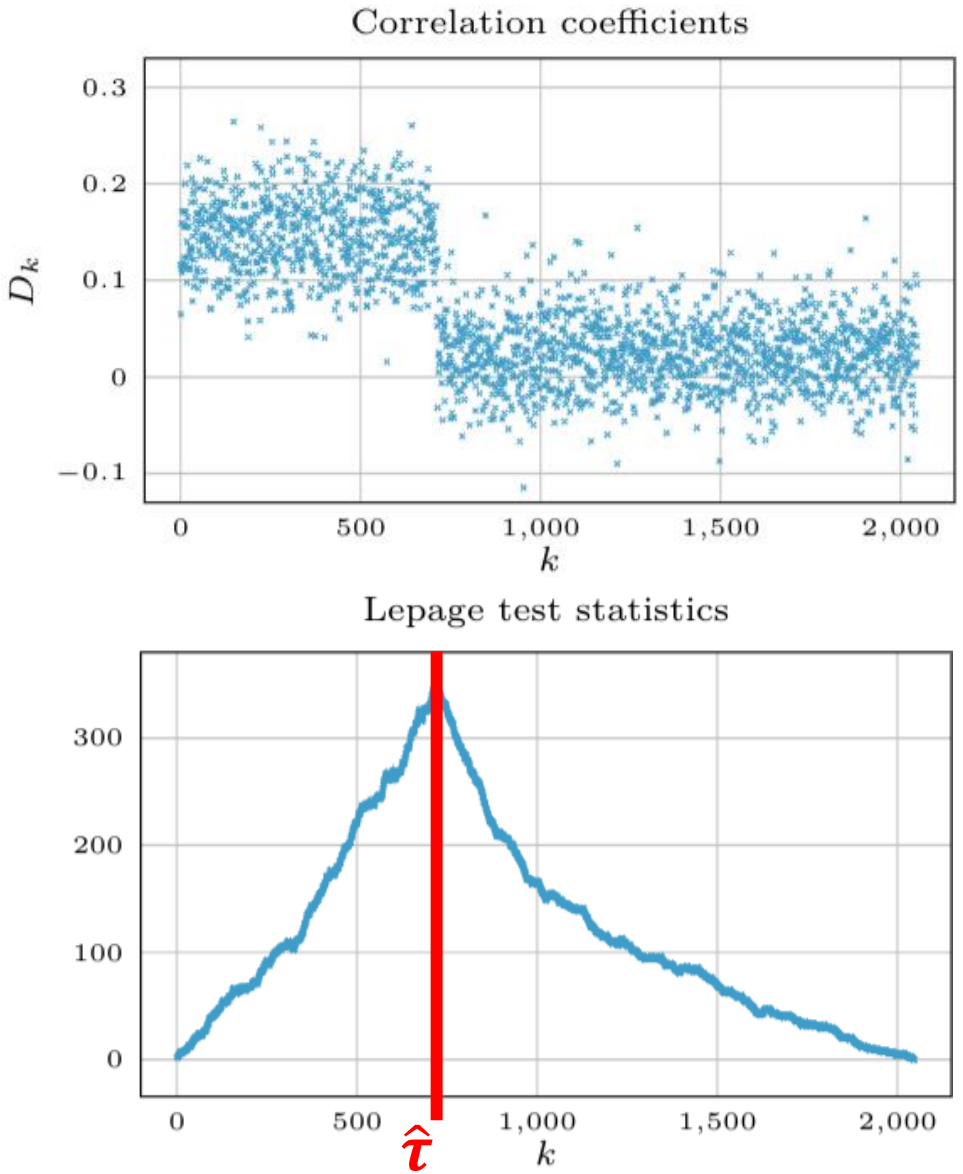
At each step a bit is reconstructed by looking at the value of a distinguisher function

Errors in sequential attacks propagate in the following steps



Frittoli, L., Matteo, B., Silvia, M., Carrera, D., Beatrice, R., Fragneto, P., Susella, R., Boracchi G. " Strengthening Sequential Side-Channel Attacks Through Change Detection". *CHES 2020*

A REAL WORLD MONITORING EXAMPLE



The CPM allows to identify the change in the distribution of the distinguisher

One the change point $\hat{\tau}$ has been estimated the right value of the bit can be retrieved

Frittoli, L., Matteo, B., Silvia, M., Carrera, D., Beatrice, R., Fragneto, P., Susella, R., Boracchi G. " Strengthening Sequential Side-Channel Attacks Through Change Detection". *CHES 2020*

CPM IN NONPARAMETRIC SETTINGS

The CPM is a very powerful framework that can be used either in online and sequential monitoring.

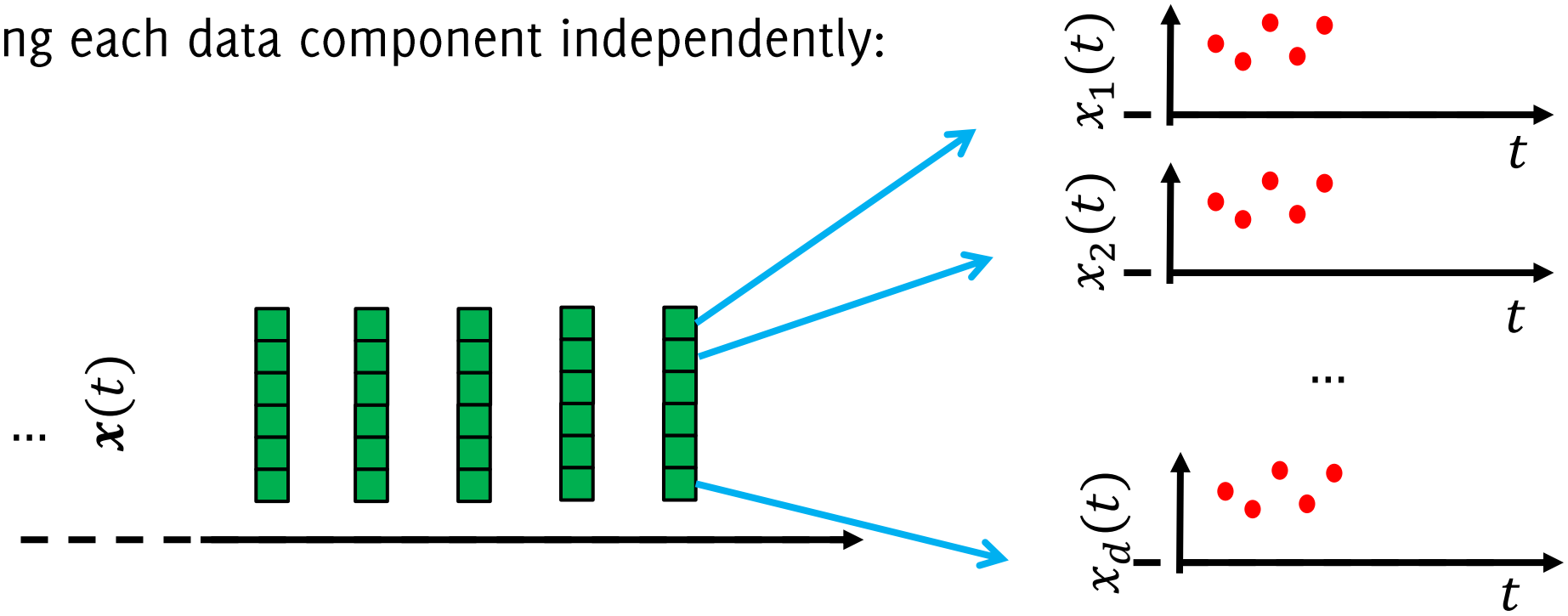
Pro: CPMs do not require training samples

Con: Non parametric statistics (Mahn-Whitney, Lepage,…) are difficult to extend to handle multivariate data.

Ross, G. J., Tasoulis, D. K., Adams, N. M. *"Nonparametric monitoring of data streams for changes in location and scale"* Technometrics, 53(4), 379-389, 2012.

CAN'T I GO BACK TO A FEW UNIVARIATE PROBLEMS?

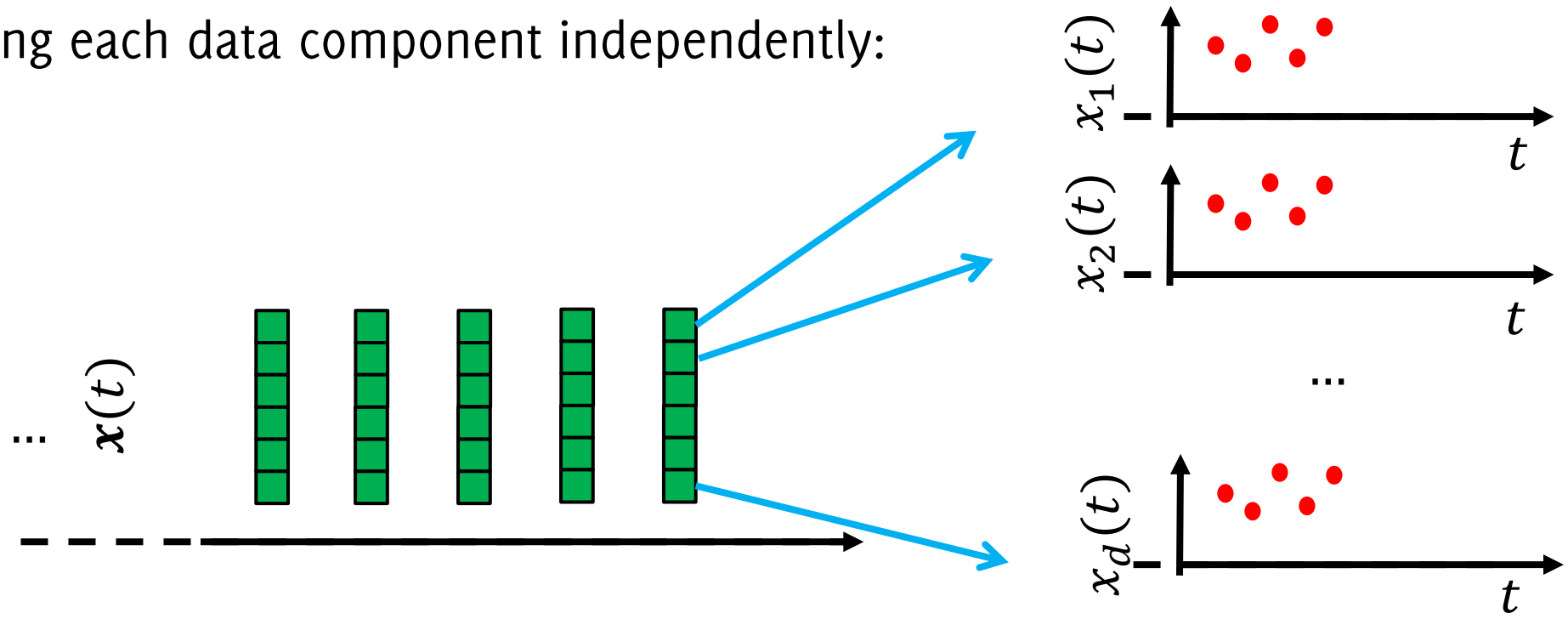
Monitoring each data component independently:



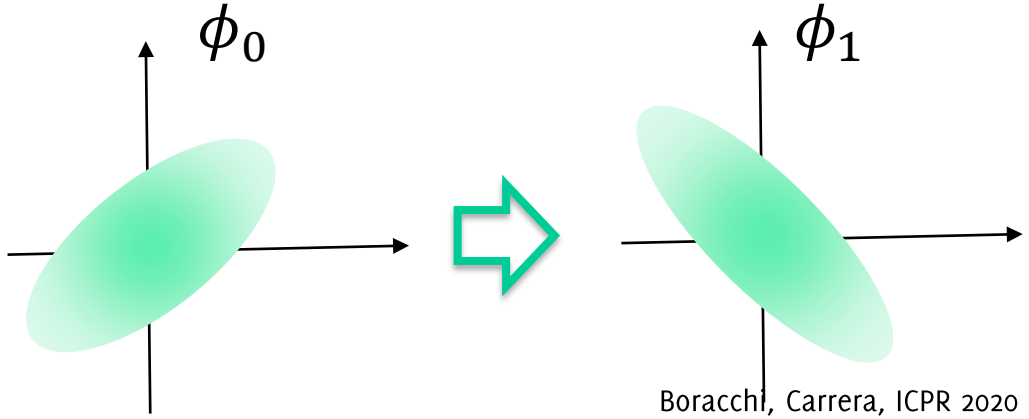
Boracchi, Carrera, ICPR 2020

CAN'T I GO BACK TO A FEW UNIVARIATE PROBLEMS?

Monitoring each data component independently:



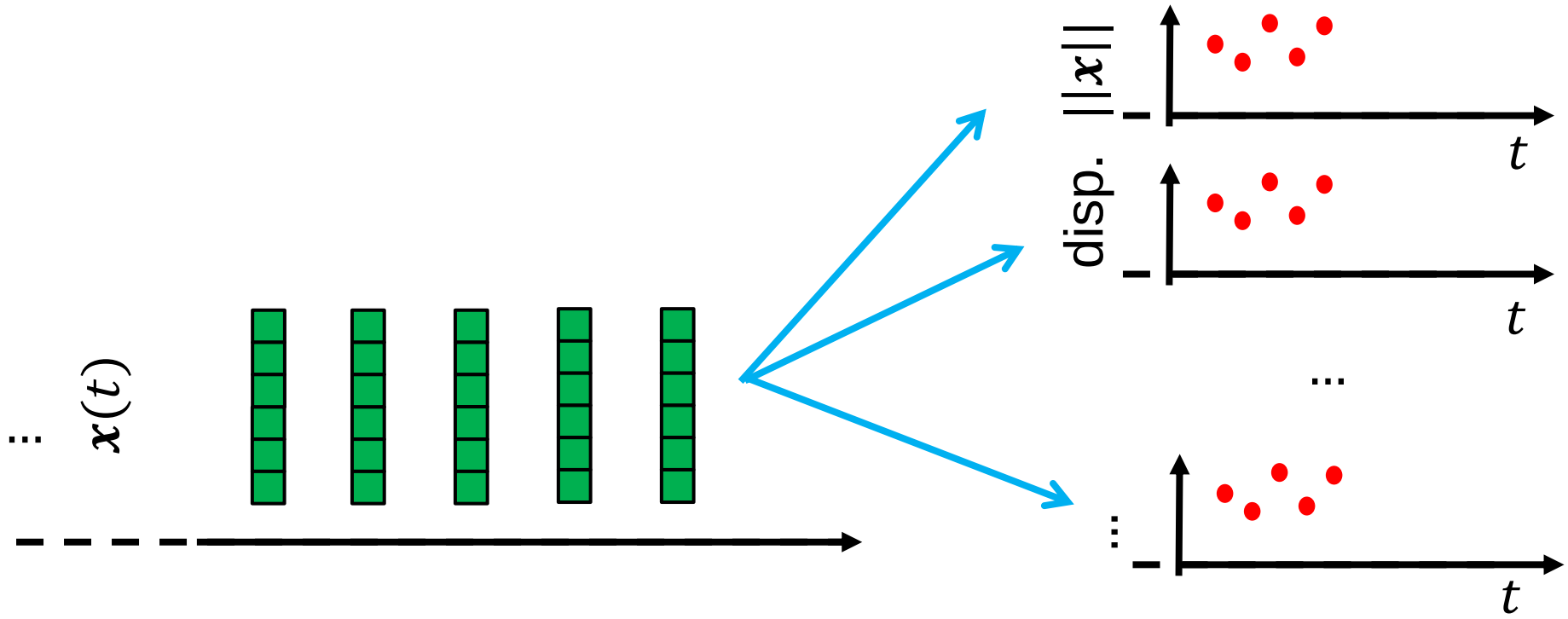
This is **not a truly multivariate** monitoring scheme.
for instance you would not be able to detect
changes affecting correlation



Boracchi, Carrera, ICPR 2020

CAN'T I GO BACK TO A FEW UNIVARIATE PROBLEMS?

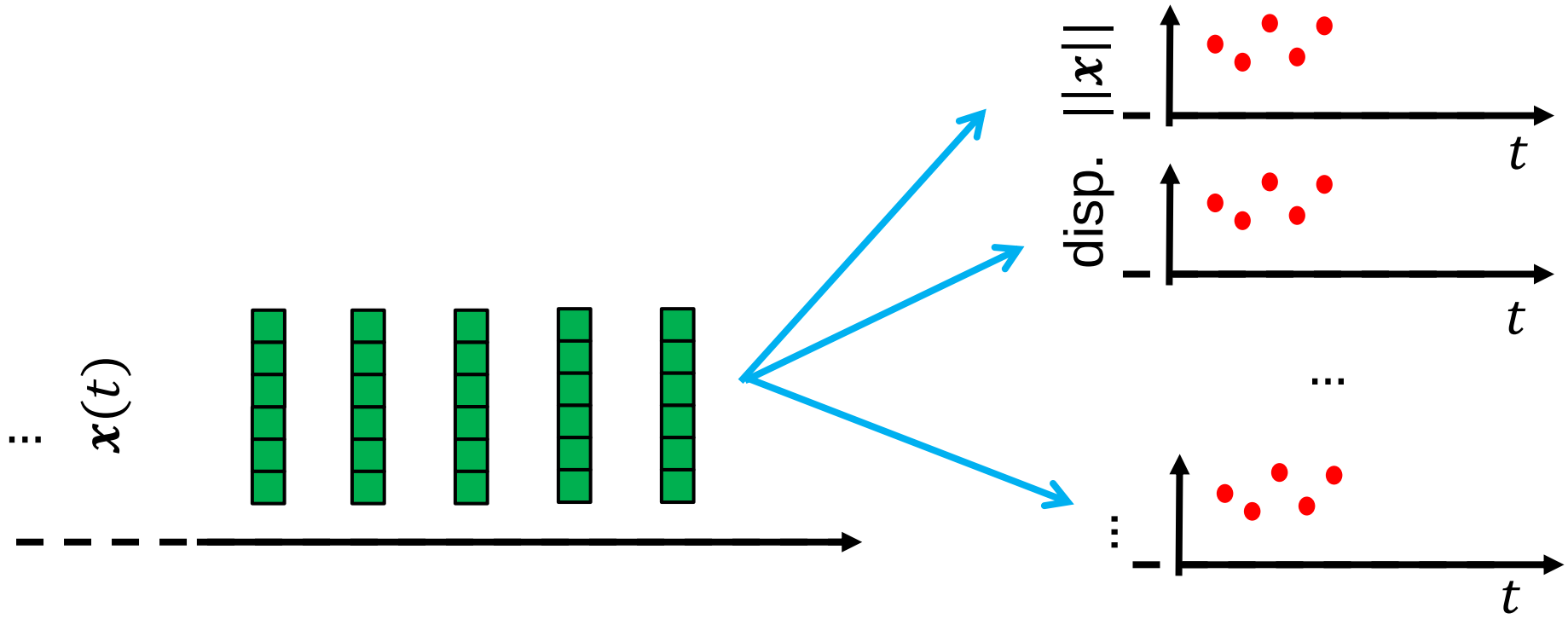
Extracting a few features / indicators that are **expected to change** when $\phi_0 \rightarrow \phi_1$ and which **distribution is known** under ϕ_0



Alippi, C., Boracchi, G., Roveri, M. *"Change detection tests using the ICI rule"* IJCNN 2010 (pp. 1-7).

CAN'T I GO BACK TO A FEW UNIVARIATE PROBLEMS?

Extracting a few features / indicators that are **expected to change** when $\phi_0 \rightarrow \phi_1$ and which **distribution is known** under ϕ_0



Not truly multivariate: **only changes affecting features are detectable**

Alippi, C., Boracchi, G., Roveri, M. *"Change detection tests using the ICI rule"* IJCNN 2010 (pp. 1-7).

CHANGE DETECTION APPROACHES

Parametric Settings:

- The Change-Point Formulation

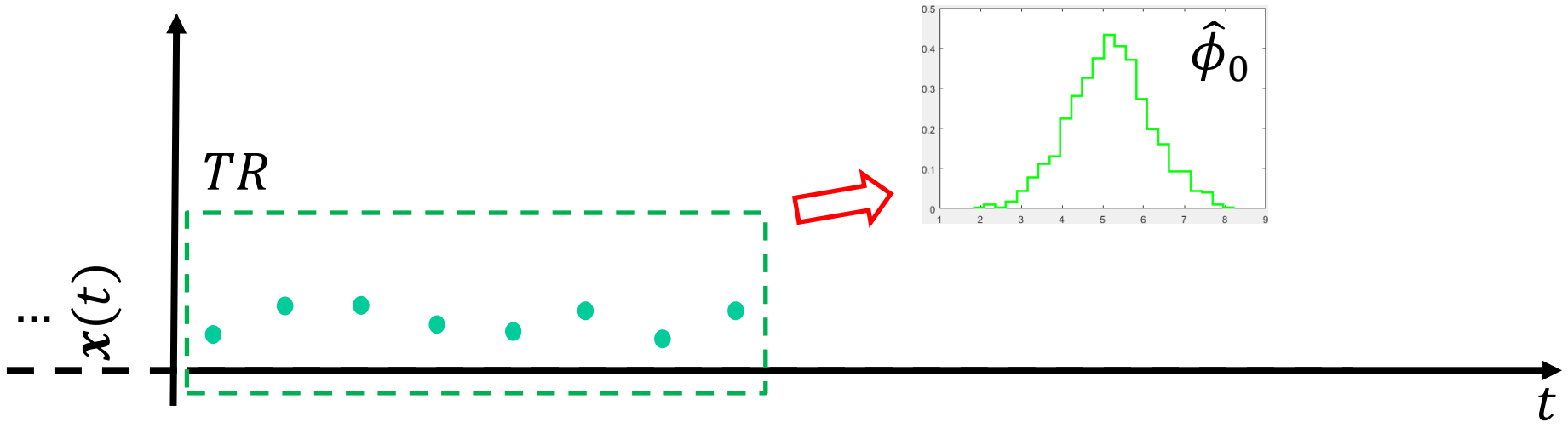
Non-parametric Settings:

- The Change-Point Formulation
- Change-Detection by Histograms
- Change-Detection by Monitoring Features
- Hierarchical Change-Detection Tests

CHANGE DETECTION BY MEANS OF HISTOGRAMS

Very often, **a training set** TR containing **stationary data** is provided, as in semi-supervised anomaly detection methods.

The distribution of stationary data can be approximated by a histogram $\hat{\phi}_0$ estimated from TR



T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi. *"An information-theoretic approach to detecting changes in multi-dimensional data streams"*. Symposium on the Interface of Statistics, Computing Science, and Applications. 2006

R. Sebastião, J. Gama, P. P. Rodrigues, and J. Bernardes, *"Monitoring incremental histogram distribution for change detection in data streams,"* Lecture Notes on Computer in Knowledge Discovery from Sensor Data, 2017.

HISTOGRAMS

An histogram h^0 defined over the input domain $\mathcal{X} \subset \mathbb{R}^d$ is

$$h^0(\mathcal{X}) = \left\{\left(S_k, p_k^0\right)\right\}_{k=1,\dots,K}$$

Where $\{S_k\}_k$ is a partitioning of $\mathcal{X}$, namely $S_k \subset \mathcal{X}$

$$\bigcup_k S_k = \mathcal{X} \text{ and } S_j \cap S_i = \delta_{i,j}$$

and $p_k^0 \in [0,1]$ is the probability (estimated from X) for a sample drawn from ϕ_0 to fall inside S_k, i.e.

$$p_k^0 = \frac{m_k}{N}$$

and $N = \#X$

MONITORING APPROACHES

Two **major monitoring approaches** using histograms:

- **Likelihood-based** methods
- **Distance-based** methods

whose applicability also depends on the partitioning scheme

M. Muja and D. G. Lowe: ***"Scalable Nearest Neighbor Algorithms for High Dimensional Data"***. TPAMI, Vol. 36, 2014

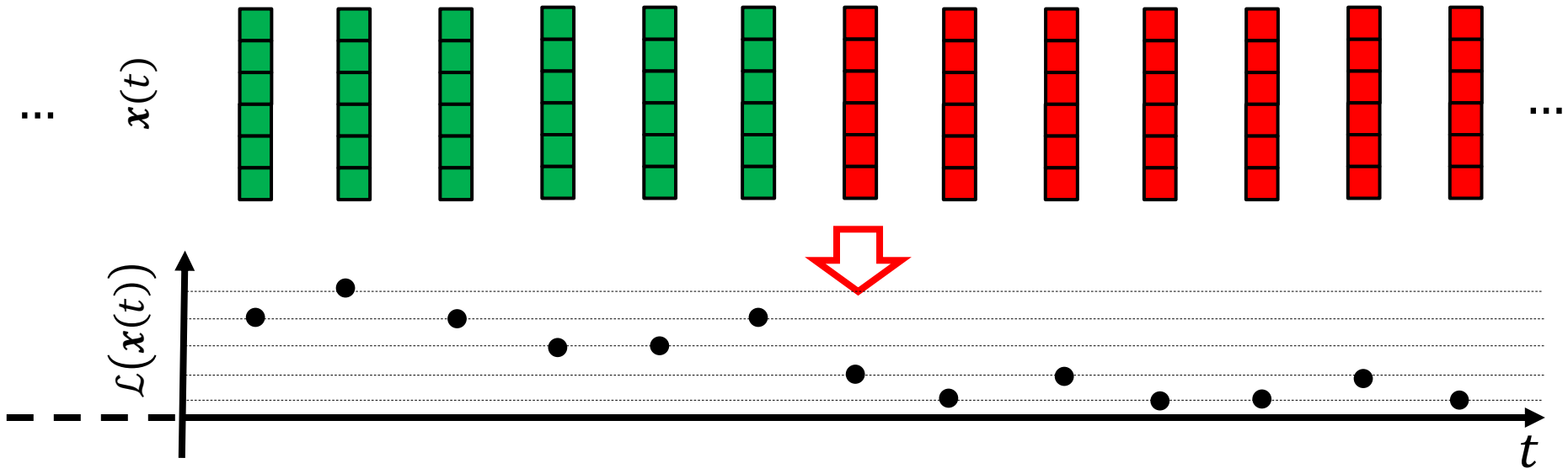
LOG-LIKELIHOOD – BASED MONITORING SCHEME

As in density-based methods, $\hat{\phi}_0$ can be used to compute the log-likelihood, which can be then monitored by univariate CDT

1. During training, estimate $\hat{\phi}_0 = \{(S_k, p_k^0)\}_{k=1,\dots,K}$ from TR
2. During testing, compute

$$\mathcal{L}(\boldsymbol{x}(t)) = \log(\hat{\phi}_0(\boldsymbol{x}(t)))$$

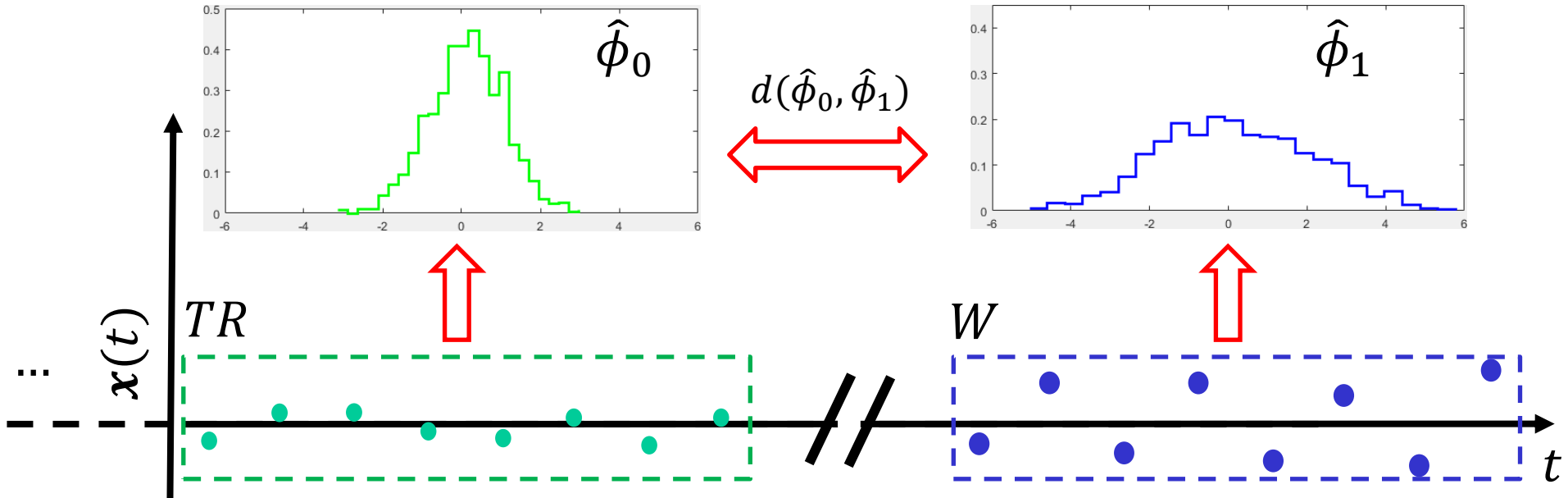
3. Monitor $\{\mathcal{L}(\boldsymbol{x}(t)),\ t = 1, \dots\}$ which is discrete



DISTANCE – BASED MONITORING SCHEME

$\hat{\phi}_0$ can be used to monitor the datastream window-wise:

- During training, estimate $\hat{\phi}_0 = \{(S_k, p_k^0)\}_{k=1,\dots,K}$ from TR
- Crop a window W over the most recent data
- Estimate $\{p_k^1\}_{k=1,\dots,K}$ from W to obtain $\hat{\phi}_1 = \{(S_k, p_k^1)\}_{k=1,\dots,K}$
- Compare $\hat{\phi}_0$ and $\hat{\phi}_1$ by a distance d between distributions
- Monitor $d(\hat{\phi}_0, \hat{\phi}_1)$



DISTANCE – BASED MONITORING SCHEME: STATISTICS

Example of distances d between distributions are:

- Kullback-Leibler divergence
- Total variation distance, Pearson chi-square test
- Kolmogorov-Smirnov, Cramer-Von-Mises distance
- Possibly a kernel approximation of these distances

DISTANCE – BASED MONITORING SCHEME: STOPPING RULE

Thresholding the distance is the typical stopping rule.

$$d(\hat{\phi}_0, \hat{\phi}_1) \gtrless \gamma$$

Thresholds:

- are defined from the empirical distribution of $d(\hat{\phi}_0, \hat{\phi}_1)$, which is computed through a **Bootstrap procedure**.
- are given from **approximation of the statistic**, which typically **holds asymptotically**, as in case the of Pearson

Similar approaches can be used to compare features extracted in different data-windows.

Dasu, T., Krishnan, S., Venkatasubramanian, S., Yi, K. *"An information-theoretic approach to detecting changes in multi-dimensional data streams"*. Symp. on the Interface of Statistics, Computing Science, and Applications, 2006.

Ditzler G., Polikar R., *"Hellinger distance based drift detection for nonstationary environments"*, IEEE SSCI 2011.

Boracchi G., Cervellera C., and Maccio D. *"Uniform Histograms for Change Detection in Multivariate Data"* IJCNN 2017

Sebastião R., Gama J. Mendonça T. *"Fading histograms in detecting distribution and concept changes"* IJDSA, 2017

Bu L., Alippi C., Zhao D. *"A pdf-free change detection test based on density difference estimation"* TNNLS 2016

S. Liu, M. Yamada, N. Collier, and M. Sugiyama, *"Change-point detection in time-series data by relative density-ratio estimation,"* Neural Networks, vol. 43, pp. 72–83, Jul. 2013

AN EXAMPLE OF DISTANCE-BASED MONITORING SCHEME

1. Compute the probabilities for an incoming batch W over $\{S_k\}$

$$p_k^W = \frac{\#\{\boldsymbol{x}_i \in S_k \cap W\}}{\nu}$$

2. Compare h^0 and h^W by a suitable distance, e.g.

$$d_{TV}(h^0, h^W) = \frac{1}{2}\sum_k |p_k^0 - p_k^W| \quad \text{(total variation)}$$

or

$$d_{PS}(h^0, h^W) = \nu \sum_k \frac{\left(p_k^0 - p_k^W\right)^2}{p_k^0} \quad \text{(Pearson)}$$

3. Run an HT on d_{TV} (having estimated its p-values empirically) or d_P (this follows a χ^2 distribution)

PROS AND CONS OF USING HISTOGRAMS

Pros:

- Histograms are very **general and flexible models**.
- Some partitioning schemes can be associated with **a tree having splits along a single component**. This enable very fast searches through the histogram.

Cons:

- When d increases, some partitioning schemes are not viable as they require q^d bins.
- In general, the distribution of test statistic is unknown (in particular in multivariate settings)

However, there is quite a lot of freedom in designing $\{S_k\}_k$

HISTOGRAMS YIELDING UNIFORM VOLUME

This is the most common way of constructing histograms.

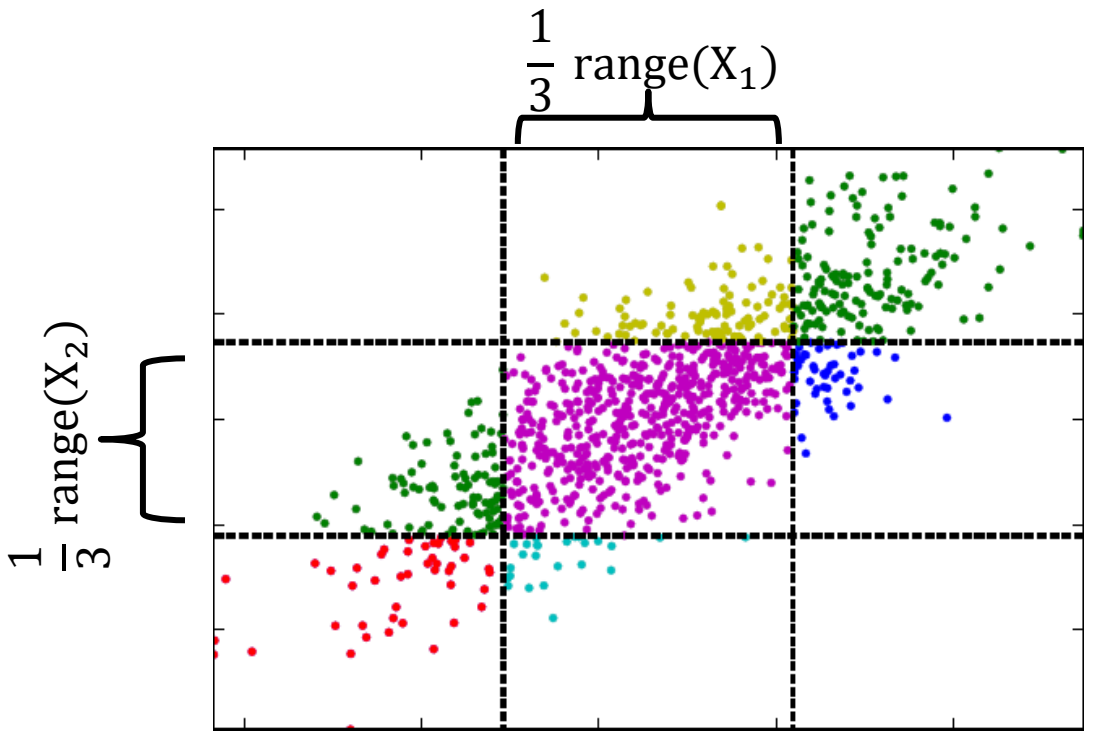
Build a tessellation of $\mathrm{supp}(X)$ by splitting each component in q equally sized parts.

This yields q^d hyper-rectangles $\{S_k\}$ having the **same volume**

Add to the histogram a region to gather points that during operation, won't fall in $\mathrm{supp}(X)$

$$S_K = \bar{X}, p_K^0 = 0$$

being $K = q^d + 1$



An example of 2D histogram $q = 1/3$

HISTOGRAMS YIELDING UNIFORM DENSITY

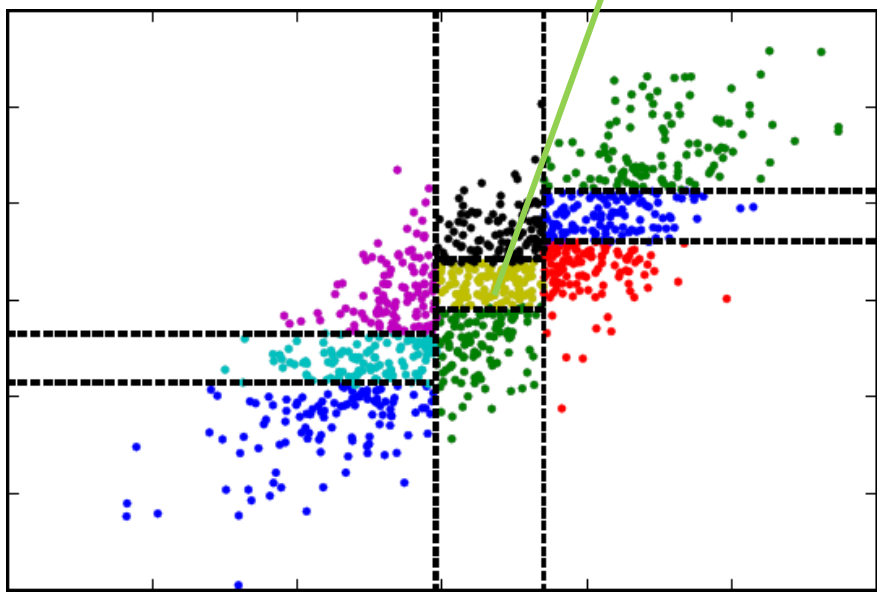
Define the partition $\{S_k\}_k$ in such a way that all the subsets have the uniform density, i.e.,

$$p_k^0 \approx \frac{1}{K}, k = 1, .., K$$

Such that each of the q^d hyper-rectangles contains the same number of points

No need to consider a separate region for $\bar{X}$

This is an example of k-d trees, there are many alternatives…

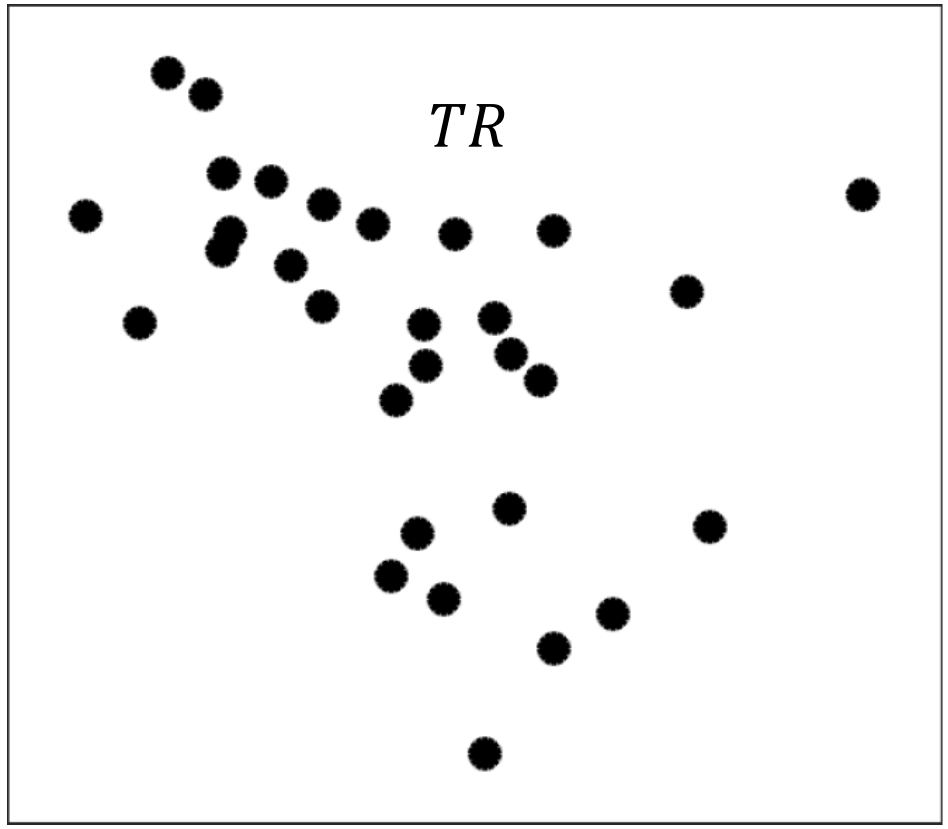


An example of 2D histogram $q = 1/3$

Boracchi G., Cervellera C., and Maccio D. "Uniform Histograms for Change Detection in Multivariate Data" IJCNN 2017

QUANTTREES: HISTOGRAMS FOR CHANGE DETECTION

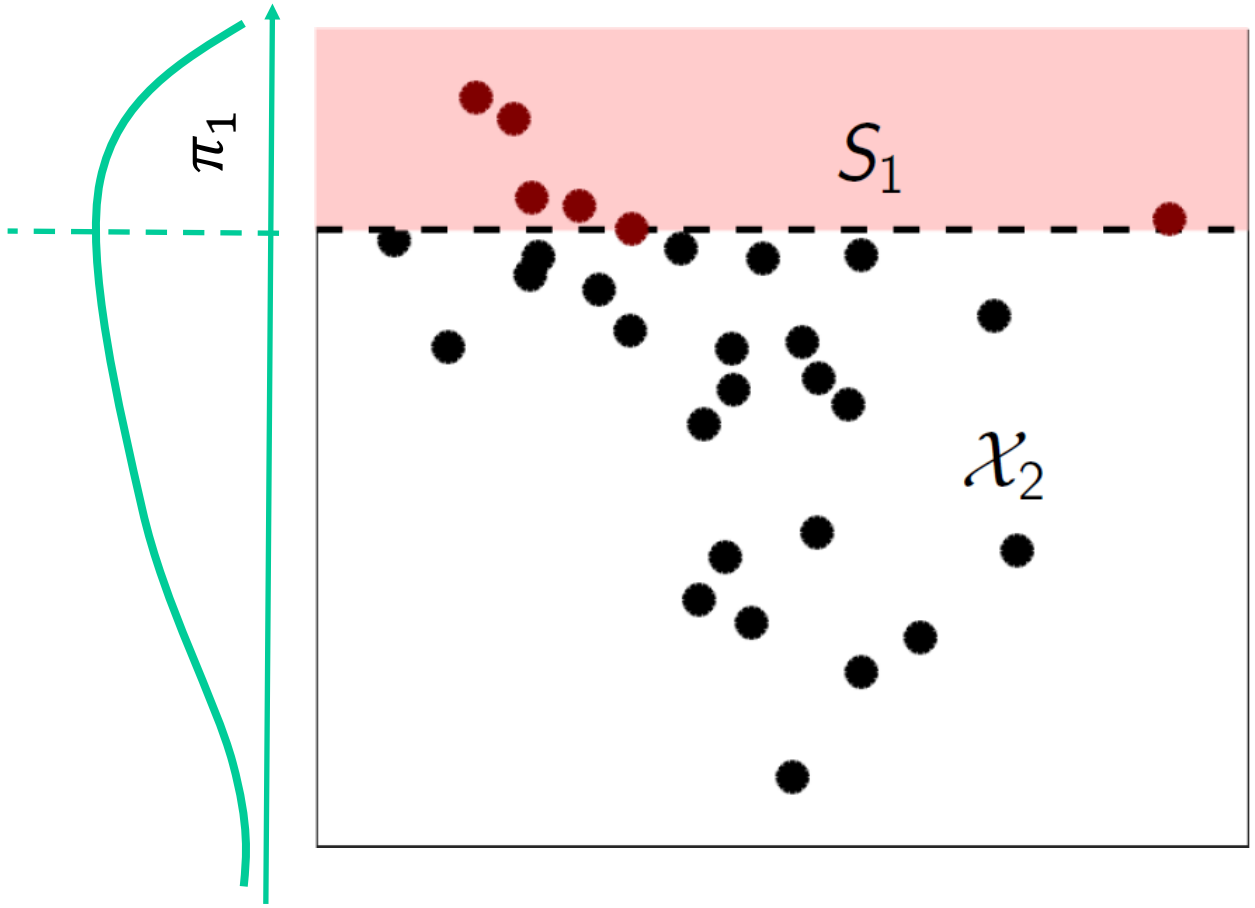
Assume you are given a set of target probabilities $\{\pi_i\}_{i=1,..,K}$ and a training set TR

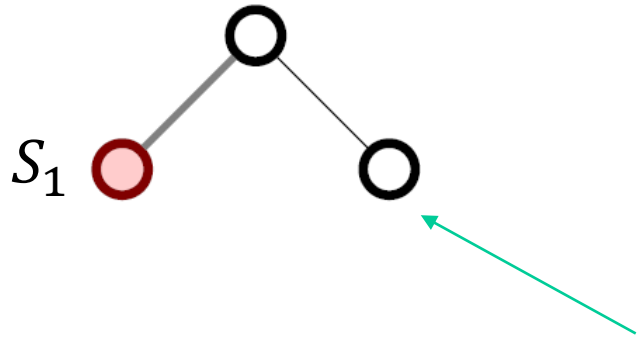


G. Boracchi, D. Carrera, C. Cervellera, D. Macciò "QuantTree: Histograms for Change Detection in Multivariate Data Streams" ICML 2018

QUANTTREES: HISTOGRAMS FOR CHANGE DETECTION

Choose a dimension j at random, define the S_1 as the set containing the $1 - \pi_1$ quantile of the marginal distribution of training samples along j



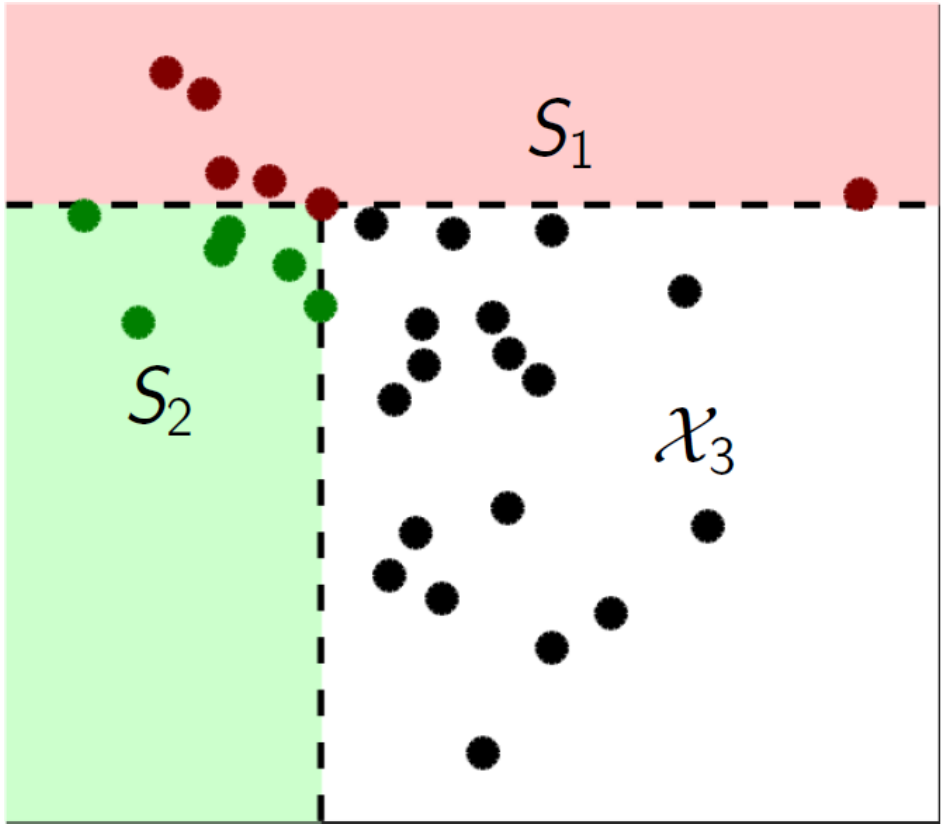


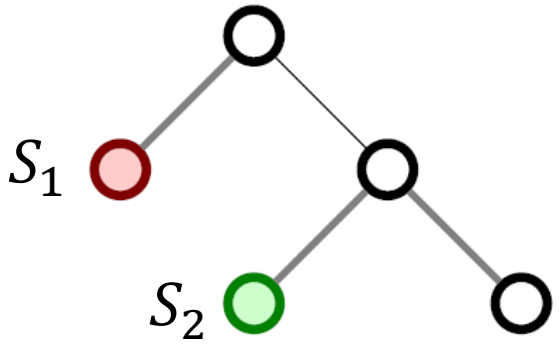
Call $\mathcal{X}_2$ the remaining samples

G. Boracchi, D. Carrera, C. Cervellera, D. Macciò "QuantTree: Histograms for Change Detection in Multivariate Data Streams" ICML 2018

QUANTTREES: HISTOGRAMS FOR CHANGE DETECTION

The procedure is iterated on the training samples that have not been included in a bin.

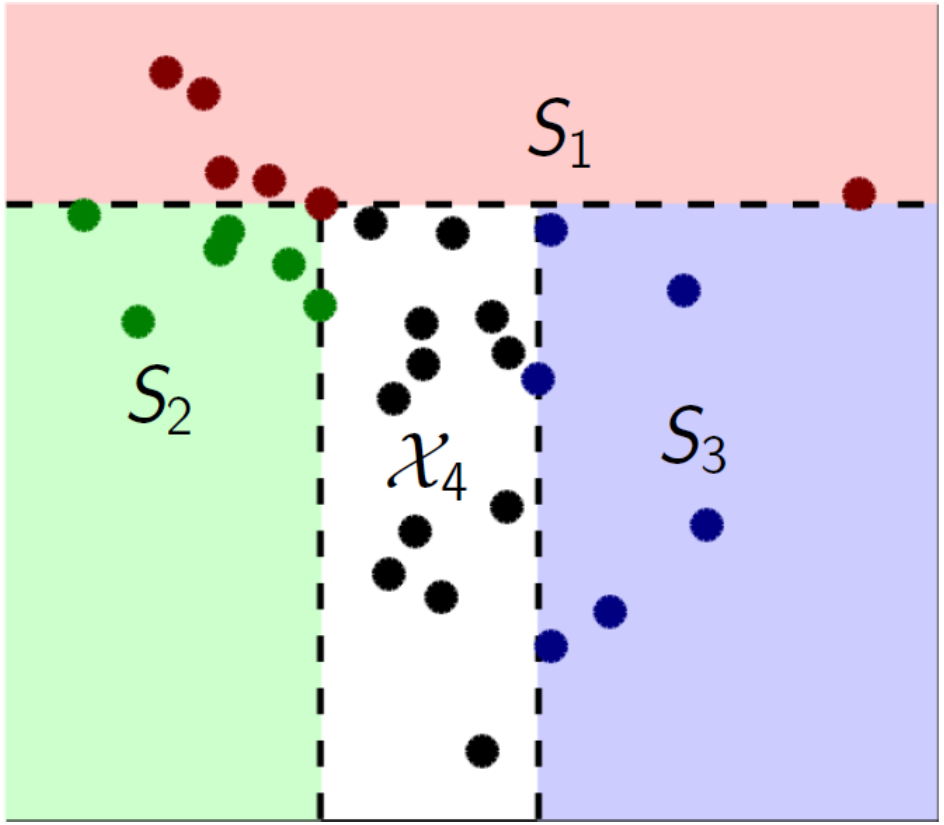


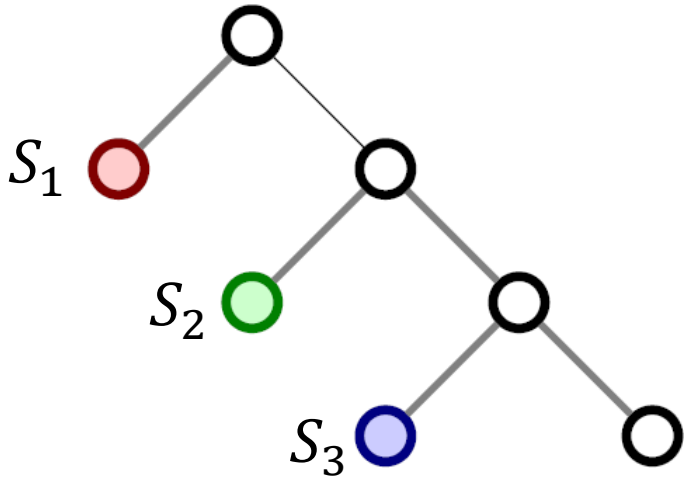


G. Boracchi, D. Carrera, C. Cervellera, D. Macciò "QuantTree: Histograms for Change Detection in Multivariate Data Streams" ICML 2018

QUANTTREES: HISTOGRAMS FOR CHANGE DETECTION

The procedure is iterated on the training samples that have not been included in a bin.

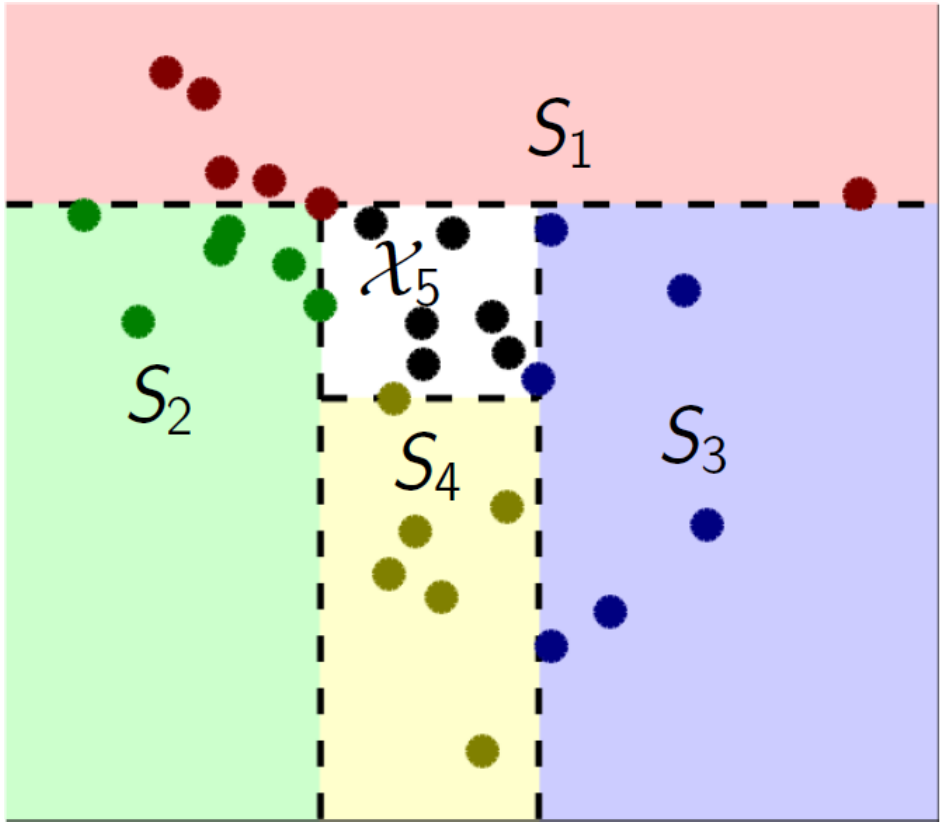


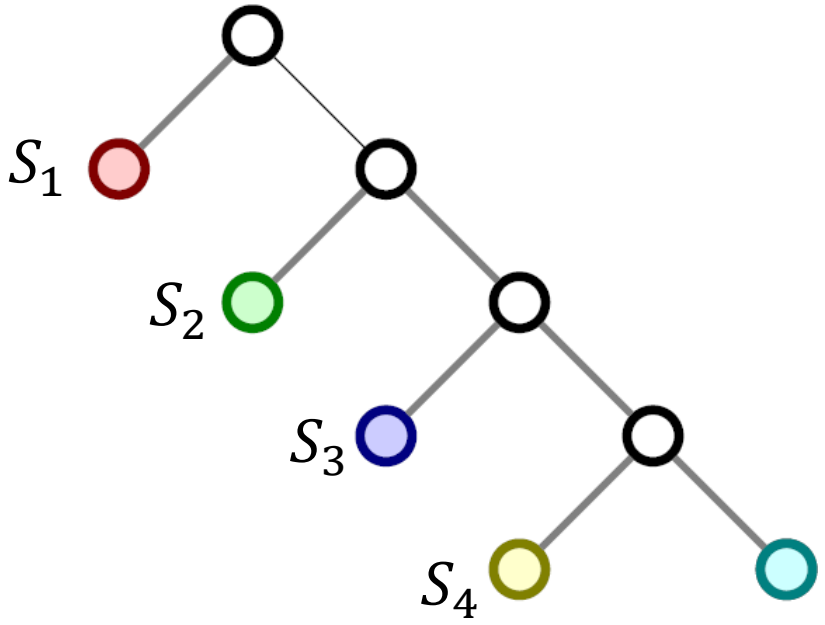


G. Boracchi, D. Carrera, C. Cervellera, D. Macciò "QuantTree: Histograms for Change Detection in Multivariate Data Streams" ICML 2018

QUANTTREES: HISTOGRAMS FOR CHANGE DETECTION

The procedure is iterated on the training samples that have not been included in a bin.

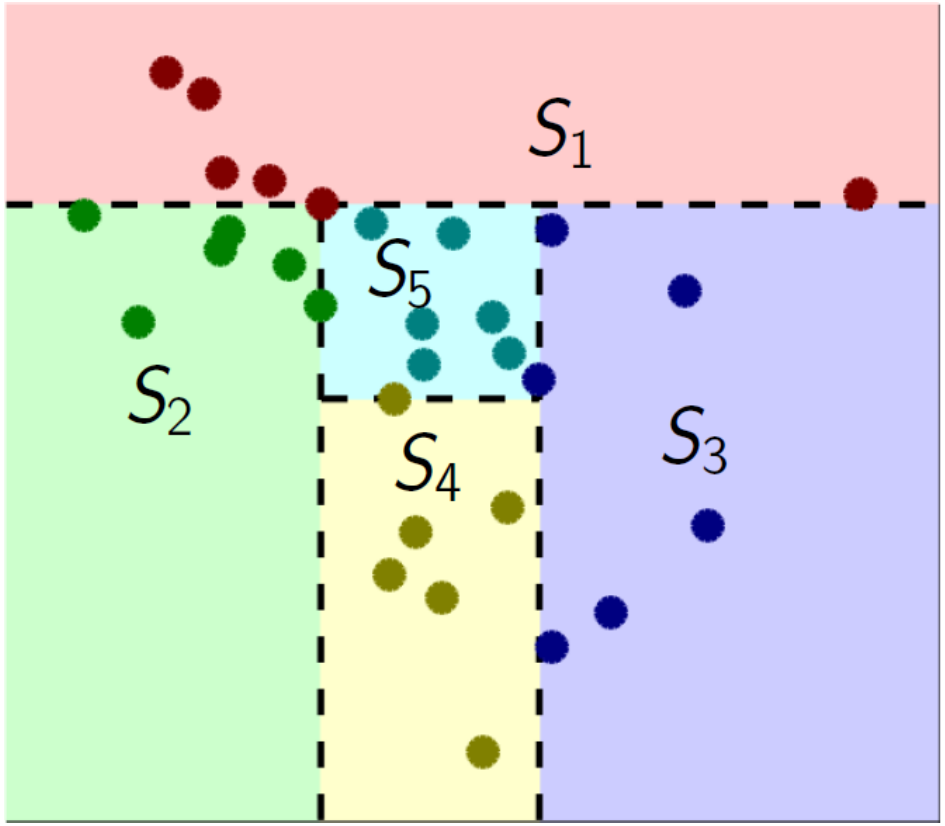


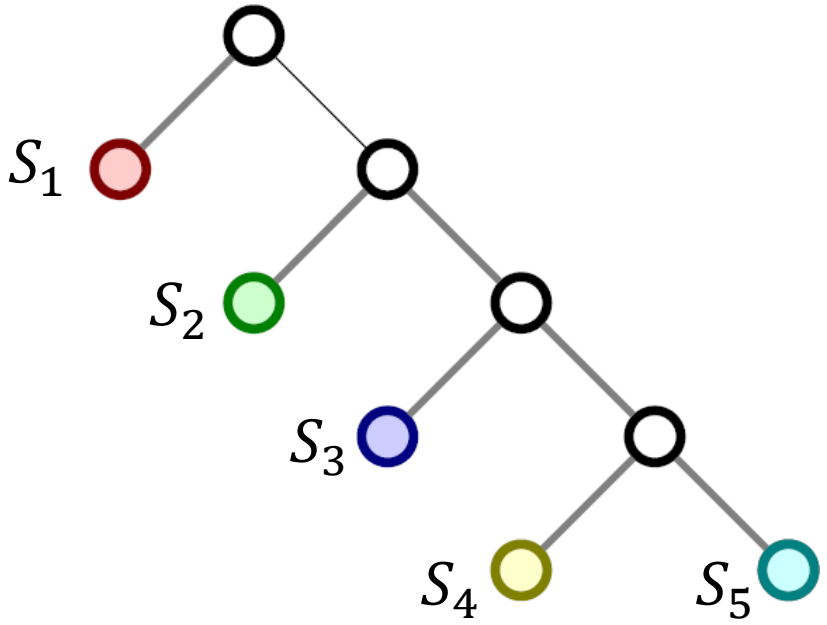


G. Boracchi, D. Carrera, C. Cervellera, D. Macciò "QuantTree: Histograms for Change Detection in Multivariate Data Streams" ICML 2018

QUANTTREES: HISTOGRAMS FOR CHANGE DETECTION

The procedure is iterated on the training samples that have not been included in a bin.





G. Boracchi, D. Carrera, C. Cervellera, D. Macciò "QuantTree: Histograms for Change Detection in Multivariate Data Streams" ICML 2018

QUANTTREES: HISTOGRAMS FOR CHANGE DETECTION

QuantTree iteratively divides the input space by binary splits along a single covariate, where the cutting points are defined by the quantiles of the marginal distributions

Algorithm 1 QuantTree

Input: Training set TR containing N stationary points in $\mathcal{X}$; number of bins K; target probabilities $\{\pi_k\}_k$.

Output: The histogram $h = \{(S_k, \widehat{\pi}_k)\}_k$.

Set $N_0 = N$, $L_0 = 0$.
for $k = 1, \ldots, K$ **do**
    Set $N_k = N_{k-1} - L_{k-1}$, $\mathcal{X}_k = \mathcal{X} \setminus \bigcup_{j<k} S_j$, and $L_k = \text{round}(\pi^k N)$.
    Choose a random component $i \in \{1, \ldots, d\}$.
    Define $z_n = [\mathbf{x}_n]_i$ for each $\mathbf{x}_n \in \mathcal{X}_k$.
    Sort $\{z_n\}$: $z_{(1)} \leq z_{(2)} \leq \ldots z_{(N_k)}$.
    Draw $\gamma \in \{0, 1\}$ from a Bernoulli(0.5).
    if $\gamma = 0$ **then**
        Define $S_k = \{\mathbf{x} \in \mathcal{X}_k \quad [\mathbf{x}]_i \leq z_{(L_k)}\}$.
    else
        Define $S_k = \{\mathbf{x} \in \mathcal{X}_k \quad [\mathbf{x}]_i \geq z_{(N_k - L_k + 1)}\}$.
    end if
    Set $\widehat{\pi}_k = L_k / N$.
end for

G. Boracchi, D. Carrera, C. Cervellera, D. Macciò "QuantTree: Histograms for Change Detection in Multivariate Data Streams" ICML 2018

QUANTTREES: HISTOGRAMS FOR CHANGE DETECTION

Theorem (ICML18)

Let $T_h(\cdot)$ be a statistic defined over the bin probabilities of an histogram h computed by QuantTree.

When $W \sim \phi_0$, the distribution of $T_h(W)$ depends only on:

- the number of training samples N,
- the size of window W,
- the expected probabilities in each bin $\{\pi_i\}_{i=1,\dots,K}$

G. Boracchi, D. Carrera, C. Cervellera, D. Macciò "QuantTree: Histograms for Change Detection in Multivariate Data Streams" ICML 2018

IMPLICATIONS

In histograms constructed by QuantTrees, the bin probabilities do not depend on ϕ_0, nor data dimension d.

Thus, **thresholds** of tests statistics **can be numerically computed from univariate data** that have been synthetically generated, yet guaranteeing a controlled false positive rate.

CHANGE DETECTION APPROACHES

Parametric Settings:

- The Change-Point Formulation

Non-parametric Settings:

- The Change-Point Formulation
- Change-Detection by Histograms
- Change-Detection by Monitoring Features
- Hierarchical Change-Detection Tests

HIERARCHICAL CHANGE-DETECTION TESTS

In nonparametric sequential monitoring it is convenient to

- **online sequential CDTs** for detection purposes
- **offline hypothesis tests** for validation purposes.

This results in two-layered (hierarchical) CDTs

Alippi, C., Boracchi, G., Roveri, M. *"Hierarchical Change-Detection Tests"* TNNLS 2016 pp 1- 13

HIERARCHICAL CHANGE-DETECTION TESTS

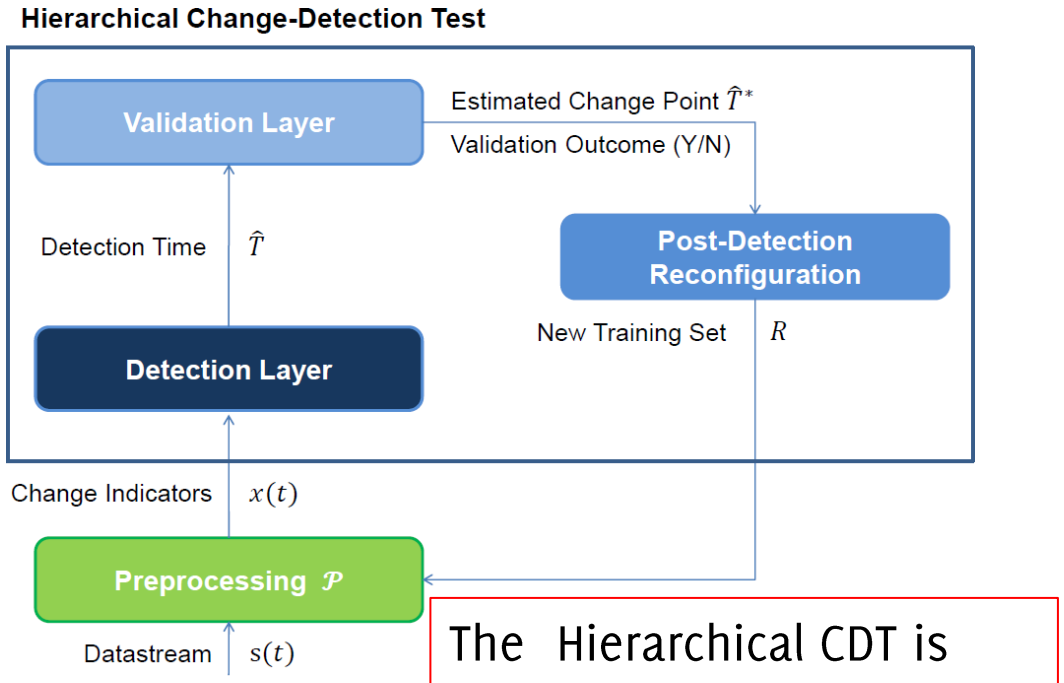
In nonparametric sequential monitoring it is convenient to

- **online sequential CDTs** for detection purposes
- **offline hypothesis tests** for validation purposes.

This results in two-layered (hierarchical) CDTs

Offline HT is activated to validate any detection

Online CDT detects process changes in the input datastream

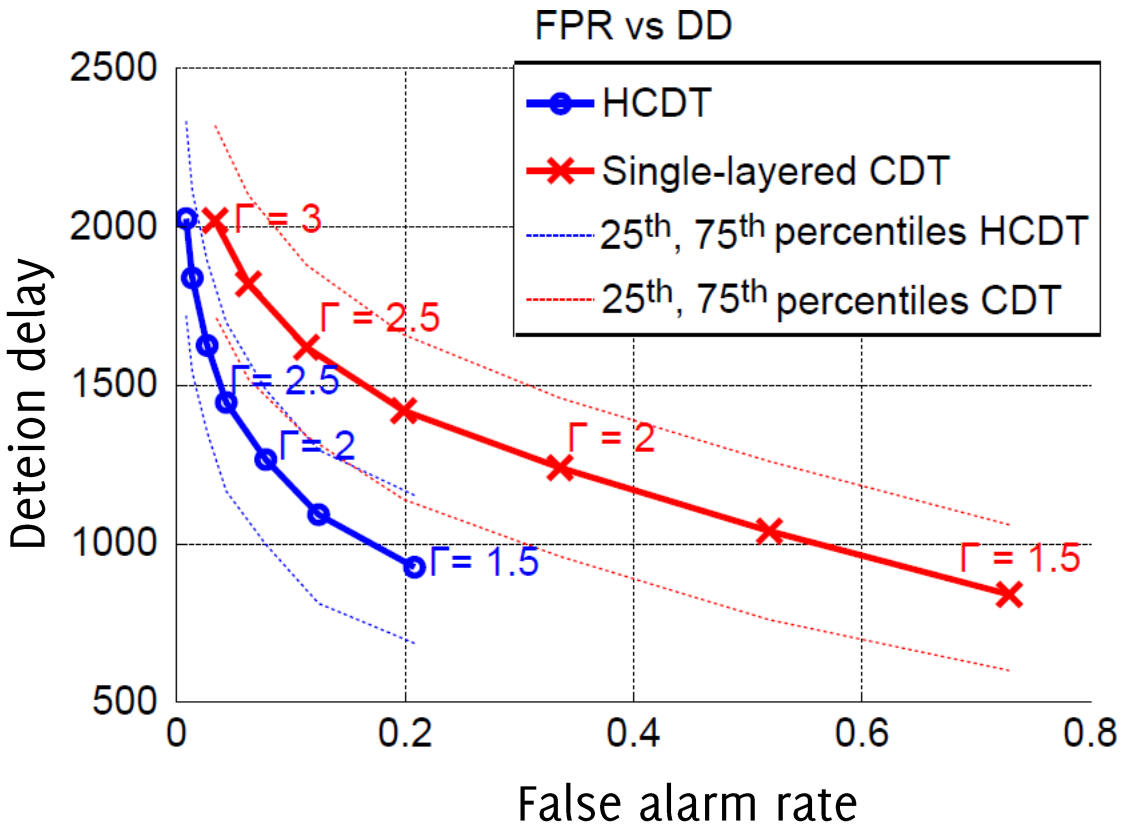


The Hierarchical CDT is automatically reconfigured

Alippi, C., Boracchi, G., Roveri, M. *"Hierarchical Change-Detection Tests"* TNNLS 2016 pp 1- 13

HIERARCHICAL CHANGE-DETECTION TESTS

Hierarchical CDTs can achieve a far **more advantageous trade-off** between false-positive rate and detection delay **than their single-layered**, more traditional, **counterpart.**



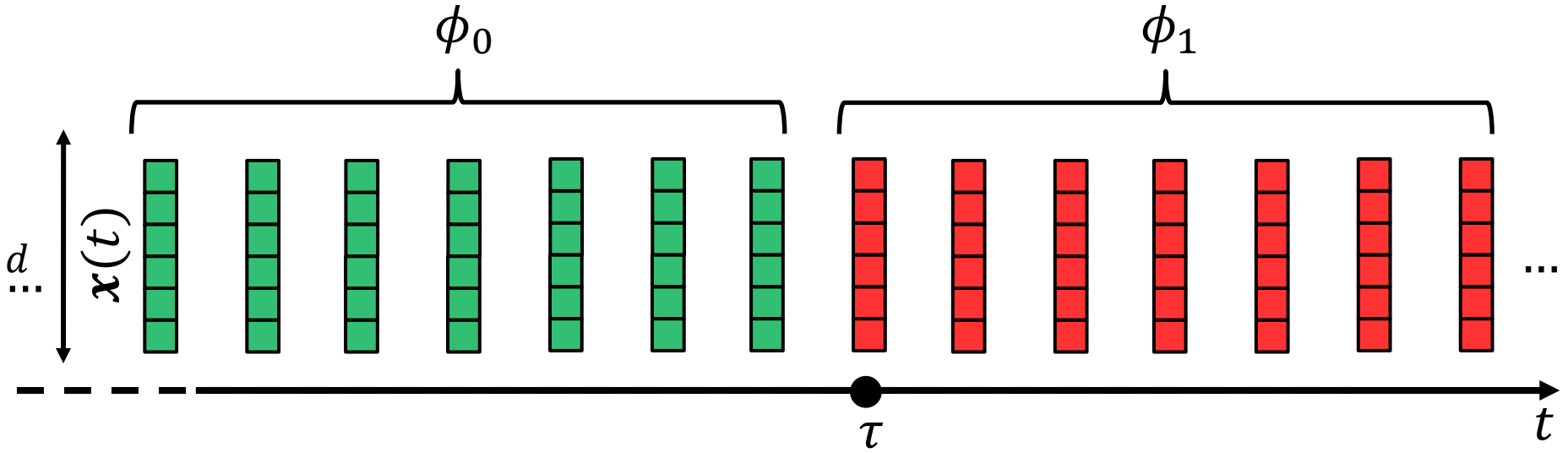
Alippi, C., Boracchi, G., Roveri, M. *"Hierarchical Change-Detection Tests"* TNNLS 2016 pp 1- 13

DETECTABILITY LOSS IN HIGH-DIMENSIONAL DATA

How data dimension affects monitoring the Log-likelihood

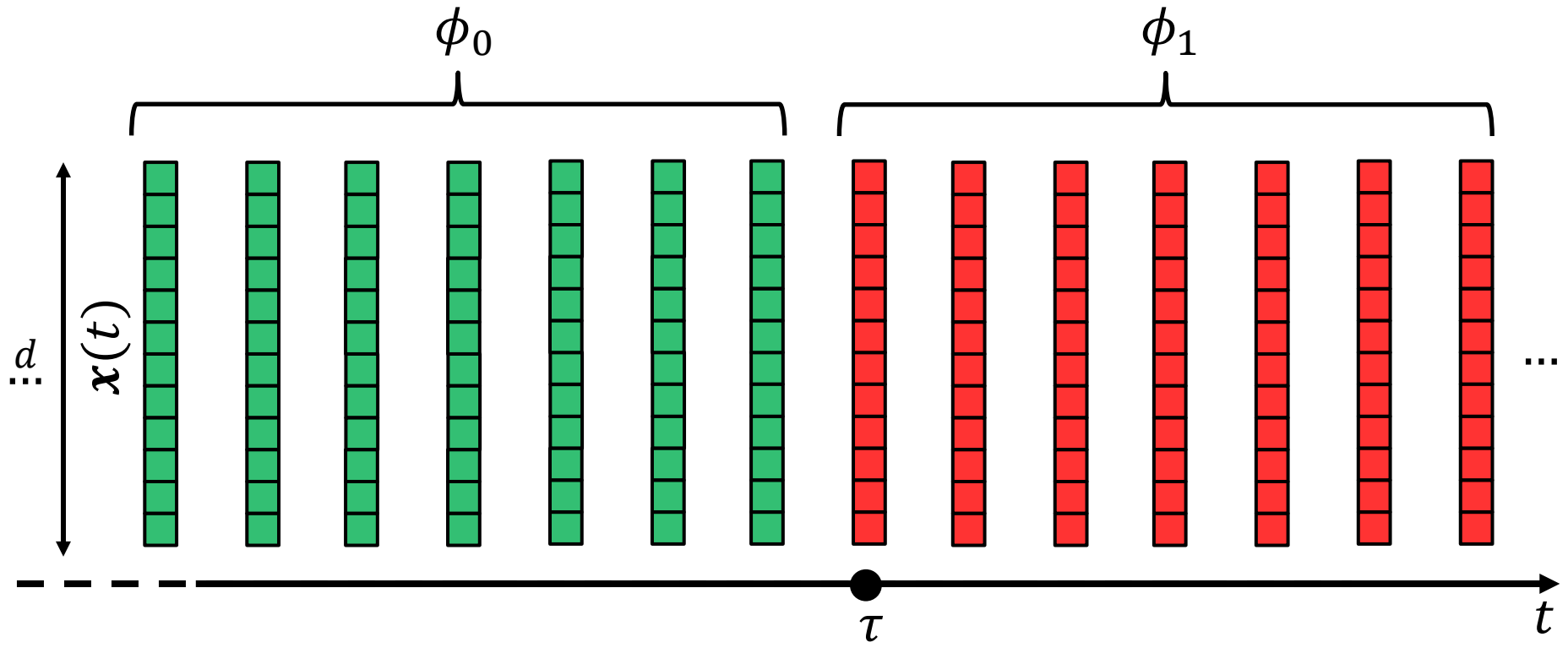
OUR GOAL

*Study how the **data dimension** d **influences** the **change detectability**, i.e., how difficult is to solve change/anomaly detection problems*



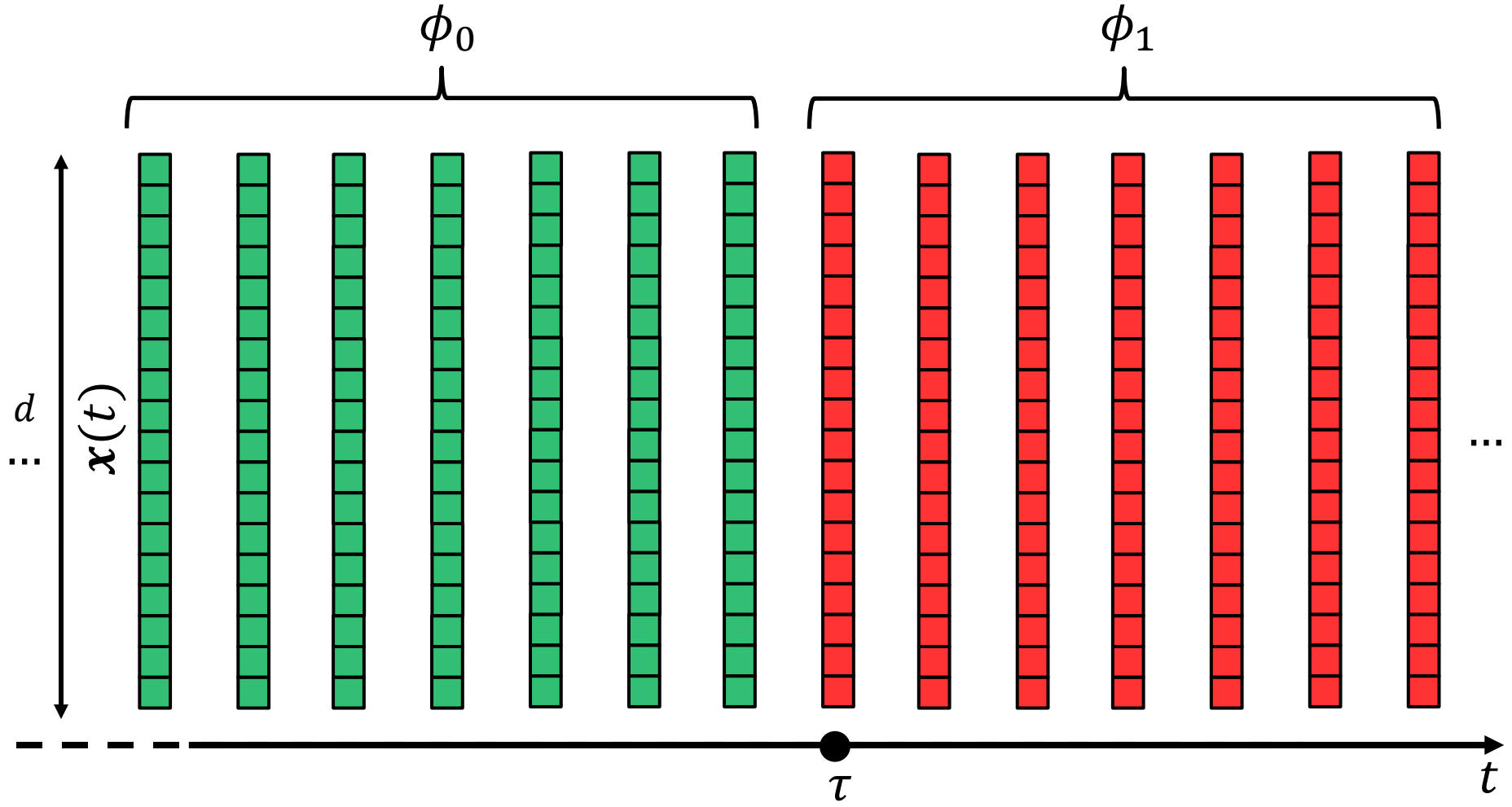
OUR GOAL

*Study how the **data dimension** d **influences** the **change detectability**, i.e., how difficult is to solve change/anomaly detection problems*



OUR GOAL

*Study how the **data dimension** d **influences** the **change detectability**, i.e., how difficult is to solve change/anomaly detection problems*



OUR APPROACH

To study the impact of the **sole data dimension** d in **change-detection problems**:

1. Consider a **change-detection approach**
2. Define a measure of **change detectability** that well correlates with traditional performance measures
3. Define a measure of **change magnitude** that refers only to differences between ϕ_0 and ϕ_1

OUR RESULT

We show there is a **detectability loss** problem, i.e. that change **detectability** steadily **decreases** when d increases.

Detectability loss is shown by:

- Analytical derivations: when ϕ_0 and ϕ_1 are **Gaussians**
- Empirical analysis on real data: measuring the **power of hypothesis tests**

ROADMAP TO DETECTABILITY LOSS

Preliminaries:

- The change magnitude
- The change-detection approach
- The measure of change detectability

The *detectability loss*

- Analytical results
- Empirical analysis

THE CHANGE MAGNITUDE

We measure the **magnitude of a change** $\phi_0 \rightarrow \phi_1$ by the *symmetric Kullback-Leibler divergence*

$$\mathrm{sKL}(\phi_0, \phi_1) = \mathrm{KL}(\phi_0, \phi_1) + \mathrm{KL}(\phi_1, \phi_0) =$$
$$= \int \log\left(\frac{\phi_0(\boldsymbol{x})}{\phi_1(\boldsymbol{x})}\right)\phi_0(\boldsymbol{x})d\boldsymbol{x} + \int \log\left(\frac{\phi_1(\boldsymbol{x})}{\phi_0(\boldsymbol{x})}\right)\phi_1(\boldsymbol{x})d\boldsymbol{x}$$

In practice, **large values** of $\mathrm{sKL}(\phi_0, \phi_1)$ correspond to **changes** $\phi_0 \rightarrow \phi_1$ that are very apparent, since $\mathrm{sKL}(\phi_0, \phi_1)$ identifies an upperbound of the power of hypothesis tests designed to detect either $\phi_0 \rightarrow \phi_1$ or $\phi_1 \rightarrow \phi_0$

T. Dasu, K. Shankar, S. Venkatasubramanian, K. Yi, *"An information-theoretic approach to detecting changes in multi-dimensional data streams"* In Proc. Symp. on the Interface of Statistics, Computing Science, and Applications, 2006

ROADMAP TO DETECTABILITY LOSS

Preliminaries:

- The change magnitude
- The change-detection approach
- The measure of change detectability

The *detectability loss*

- Analytical results
- Empirical analysis

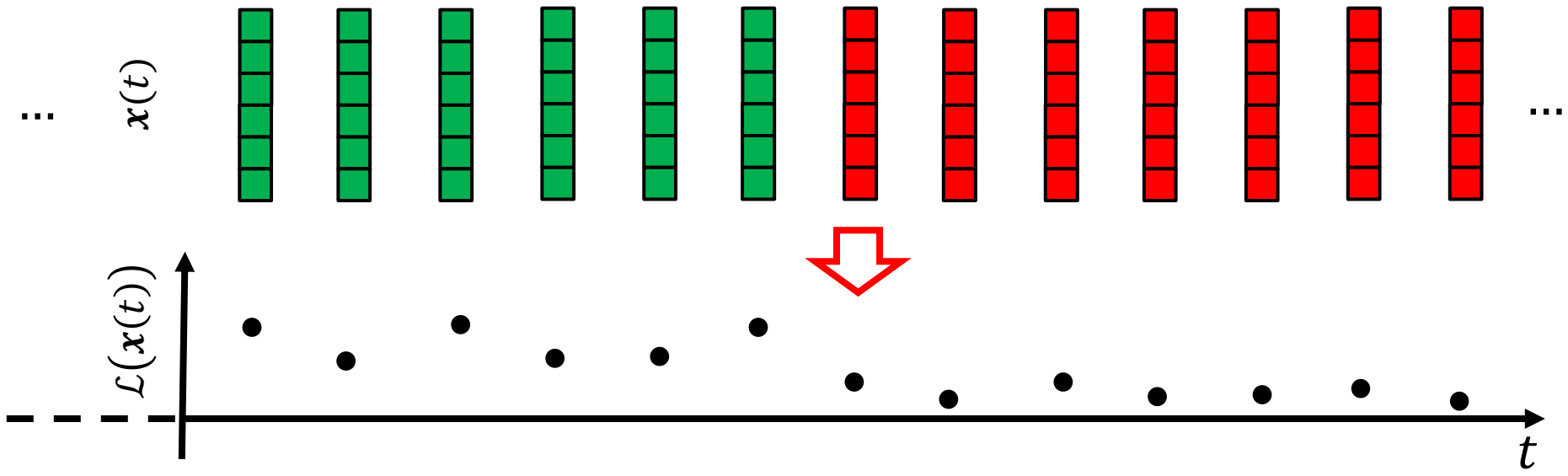
MONITORING THE LOG-LIKELIHOOD

A typical approach to monitor the log-likelihood

1. During training, estimate $\hat{\phi}_0$ from TR
2. During testing, compute

$$\mathcal{L}\big(\boldsymbol{x}(t)\big) = -\log(\hat{\phi}_0(\boldsymbol{x}(t)))$$

3. Monitor $\{\mathcal{L}\big(\boldsymbol{x}(t)\big),\ t = 1, \ldots\}$



ROADMAP TO DETECTABILITY LOSS

Preliminaries:

- The change magnitude
- The change-detection approach
- The measure of change detectability

The *detectability loss*

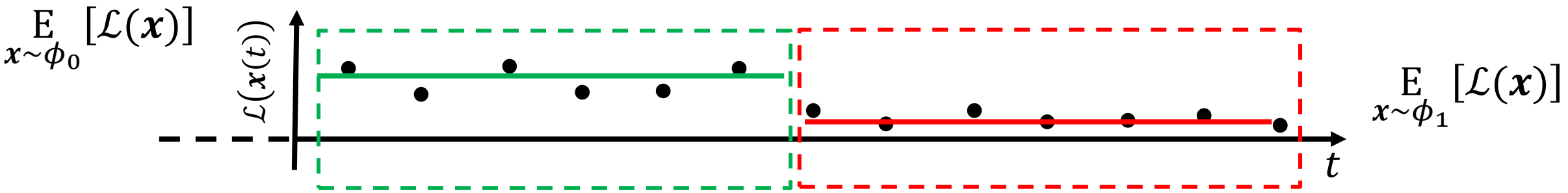
- Analytical results
- Empirical analysis

THE CHANGE DETECTABILITY

The ***Signal to Noise Ratio of the change***

$$\mathrm{SNR}(\phi_0 \to \phi_1) = \frac{\left(\underset{x\sim\phi_0}{\mathrm{E}} [\mathcal{L}(\boldsymbol{x})] - \underset{x\sim\phi_1}{\mathrm{E}} [\mathcal{L}(\boldsymbol{x})] \right)^2}{\underset{x\sim\phi_0}{\mathrm{var}} [\mathcal{L}(\boldsymbol{x})] + \underset{x\sim\phi_1}{\mathrm{var}} [\mathcal{L}(\boldsymbol{x})]}$$

measures the extent to which $\phi_0 \to \phi_1$ is **detectable by statistical tools** designed to **detect changes** in $\mathrm{E}[\mathcal{L}(\boldsymbol{x})]$



OUR APPROACH

To study the impact of the **sole data dimension** d in **change-detection problems** we need to:

1. Consider a **change-detection approach**
2. Define a measure of **change detectability** that well correlates with traditional performance measures
3. Define a measure of **change magnitude** that refers only to differences between ϕ_0 and ϕ_1

Our goal (reformulated):
Studying how the **change detectability** $\mathrm{SNR}(\phi_0 \to \phi_1)$ **varies** in **change-detection problems** that have

- **different data dimensions** d
- **constant change magnitude** $\mathrm{sKL}(\phi_0, \phi_1)$

ROADMAP TO DETECTABILITY LOSS

Preliminaries:

- The change magnitude
- The change-detection approach
- The measure of change detectability

The *detectability loss*

- Analytical results
- Empirical analysis

THE DETECTABILITY LOSS

Theorem (IJCAI16)

Let $\phi_0 = \mathcal{N}(\mu_0, \Sigma_0)$ *and let* $\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v})$ *where* $Q \in \mathbb{R}^{d \times d}$ *and orthogonal ,* $\boldsymbol{v} \in \mathbb{R}^d$*, then*

$$\mathrm{SNR}(\phi_0 \to \phi_1) < \frac{C}{d}$$

Where C *is a constant that depends only on* $\mathrm{sKL}(\phi_0, \phi_1)$

C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "*Change Detection in Multivariate Datastreams: Likelihood and Detectability Loss*" IJCAI 2016

THE DETECTABILITY LOSS

Theorem (IJCAI16)

Let $\phi_0 = \mathcal{N}(\mu_0, \Sigma_0)$ *and let* $\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v})$ *where* $Q \in \mathbb{R}^{d \times d}$ *and orthogonal ,* $\boldsymbol{v} \in \mathbb{R}^d$*, then*

$$\mathrm{SNR}(\phi_0 \to \phi_1) < \frac{C}{d}$$

Where C *is a constant that depends only on* $\mathrm{sKL}(\phi_0, \phi_1)$

Remarks:

- Changes of a given magnitude, $\mathrm{sKL}(\phi_0, \phi_1)$, become more difficult to detect when d increases
- DL does not depend on the change parameters
- DL does not depend on the specific detection rule
- DL does not depend on estimation errors on $\hat{\phi}_0$

THE DETECTABILITY LOSS: THE CHANGE MODEL

Theorem (IJCAI16)

Let $\phi_0 = \mathcal{N}(\mu_0, \Sigma_0)$ *and let* $\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v})$ *where* $Q \in \mathbb{R}^{d \times d}$ *and orthogonal ,* $\boldsymbol{v} \in \mathbb{R}^d$*, then*

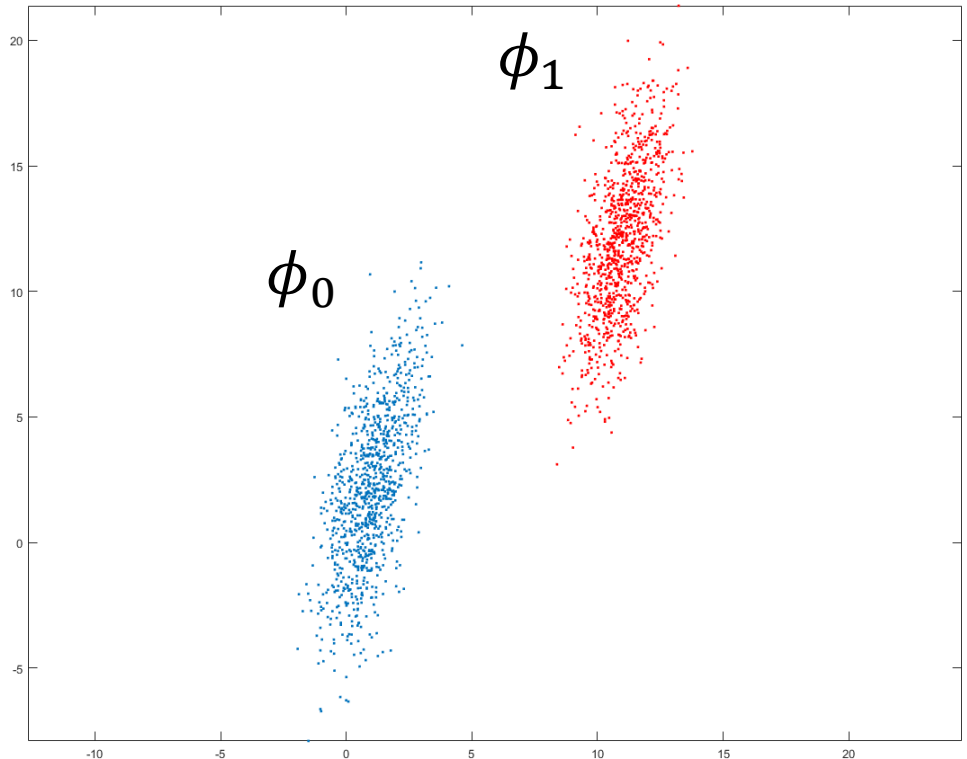
$$\mathrm{SNR}(\phi_0 \to \phi_1) < \frac{C}{d}$$

Where C *is a constant that depends only on* $\mathrm{sKL}(\phi_0, \phi_1)$

THE DETECTABILITY LOSS: THE CHANGE MODEL

The change model $\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v})$ includes:

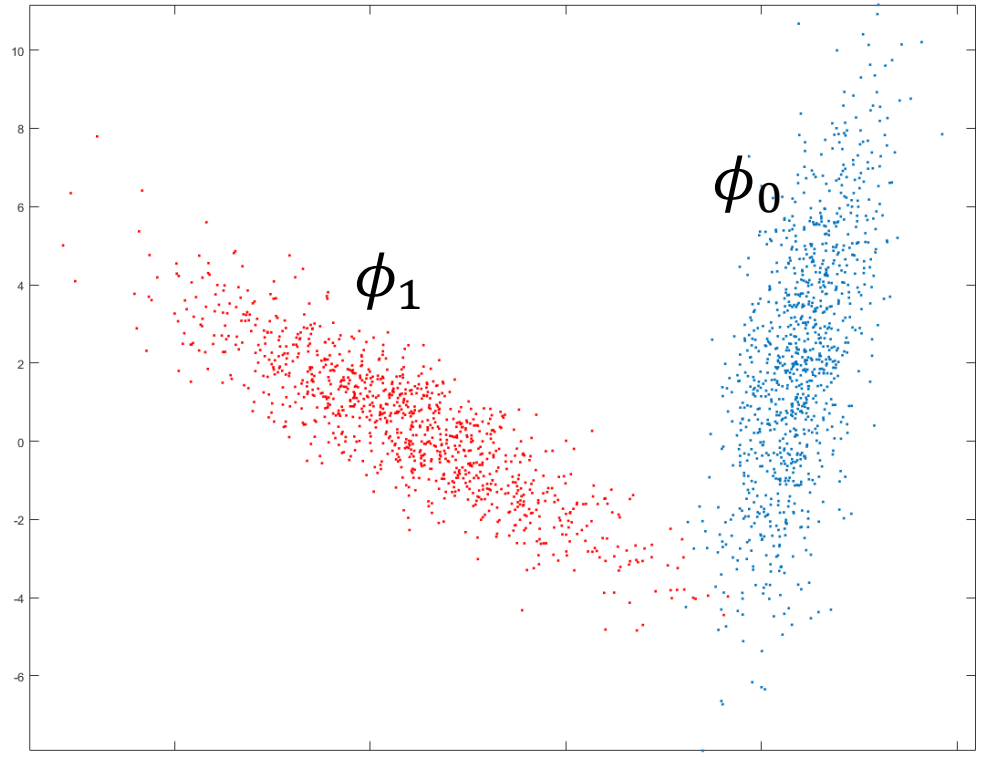
- Changes in the location of ϕ_0 (i.e, $+\boldsymbol{v}$)



THE DETECTABILITY LOSS: THE CHANGE MODEL

The change model $\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v})$ includes:

- Changes in the location of ϕ_0 (i.e, $+\boldsymbol{v}$)
- Changes in the correlation of $\boldsymbol{x}$ (i.e, $Q\boldsymbol{x}$)

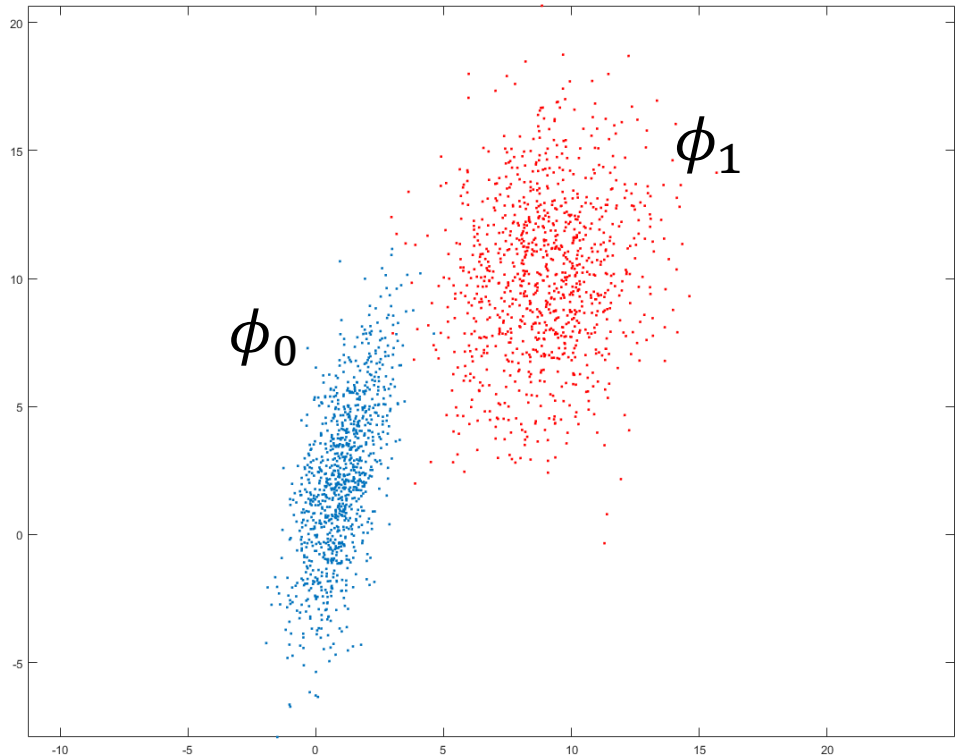


THE DETECTABILITY LOSS: THE CHANGE MODEL

The change model $\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v})$ includes:

- Changes in the location of ϕ_0 (i.e, $+\boldsymbol{v}$)
- Changes in the correlation of $\boldsymbol{x}$ (i.e, $Q\boldsymbol{x}$)

It does not include changes in the scale of ϕ_0 that can be however detected monitoring $||\boldsymbol{x}||$



THE DETECTABILITY LOSS: THE GAUSSIAN ASSUMPTION

Assuming $\phi_0 = \mathcal{N}(\mu_0, \Sigma_0)$ looks like a severe limitation.

- Other distributions are not easy to handle analytically
- We can prove that DL occurs **also in random vectors having independent components**
- The result have been **empirically confirmed** in case of approximations of $\mathcal{L}(\cdot)$ typically used **for Gaussian mixtures**

ROADMAP TO DETECTABILITY LOSS

Preliminaries:

- The change magnitude
- The change-detection approach
- The measure of change detectability

The *detectability loss*

- Analytical results
- Empirical analysis

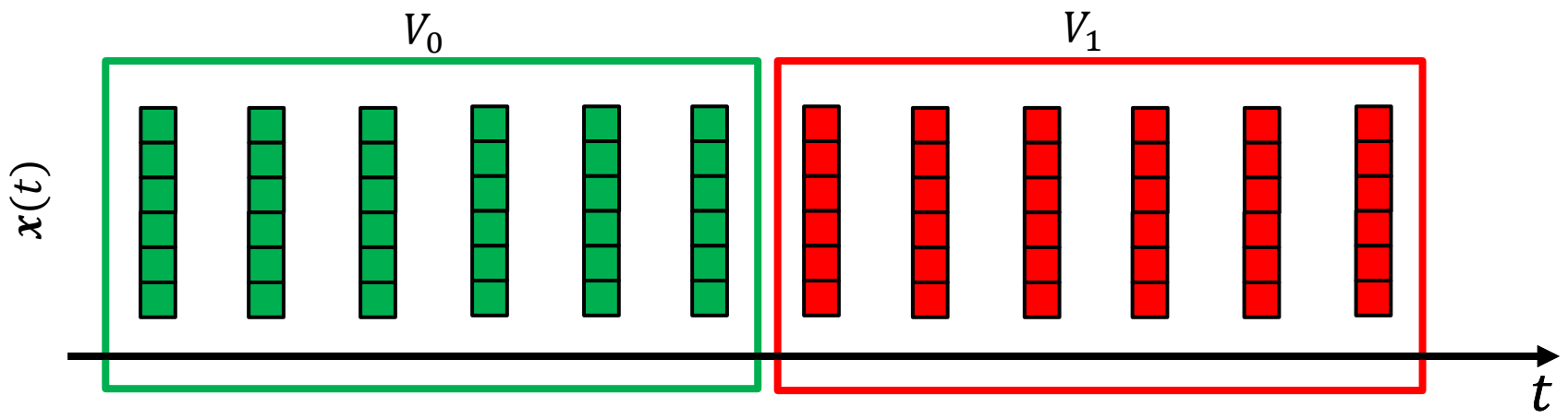
THE DETECTABILITY LOSS: EMPIRICAL ANALYSIS

The data

- Synthetically generate streams with different dimensions d
- Estimate $\hat{\phi}_0$ by Gaussian Mixture from a **stationary training set**
- In each stream we introduce $\phi_0 \to \phi_1$ such that

$$\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v}) \text{ and } \mathrm{sKL}(\phi_0, \phi_1) = 1$$

- **Test data: two windows** V_0 and V_1 (500 samples each) selected before and after the change.



THE DETECTABILITY LOSS: EMPIRICAL ANALYSIS

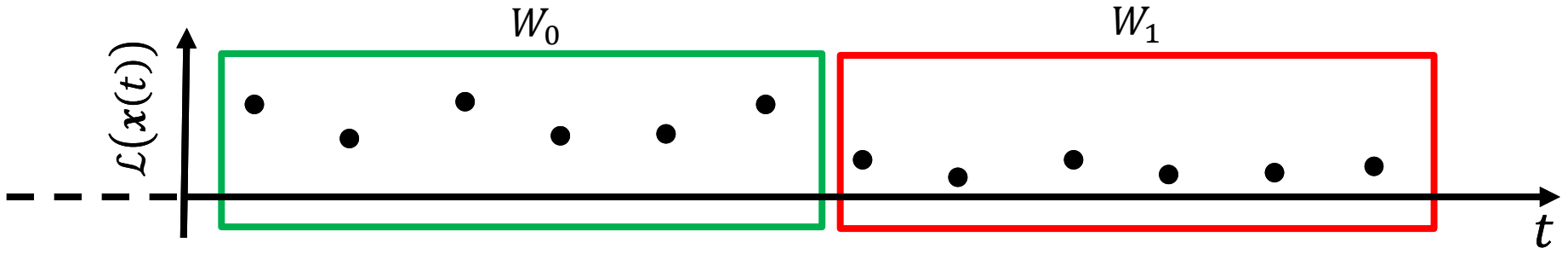
We measure the change-detectability as:

- Compute $\mathcal{L}\left(\hat{\phi}_0(\boldsymbol{x})\right)$ from V_0 and V_1, obtaining W_0 and W_1
- Compute a test statistic $\mathcal{T}(W_0, W_1)$ to compare the two
- Detect a change by an hypothesis test

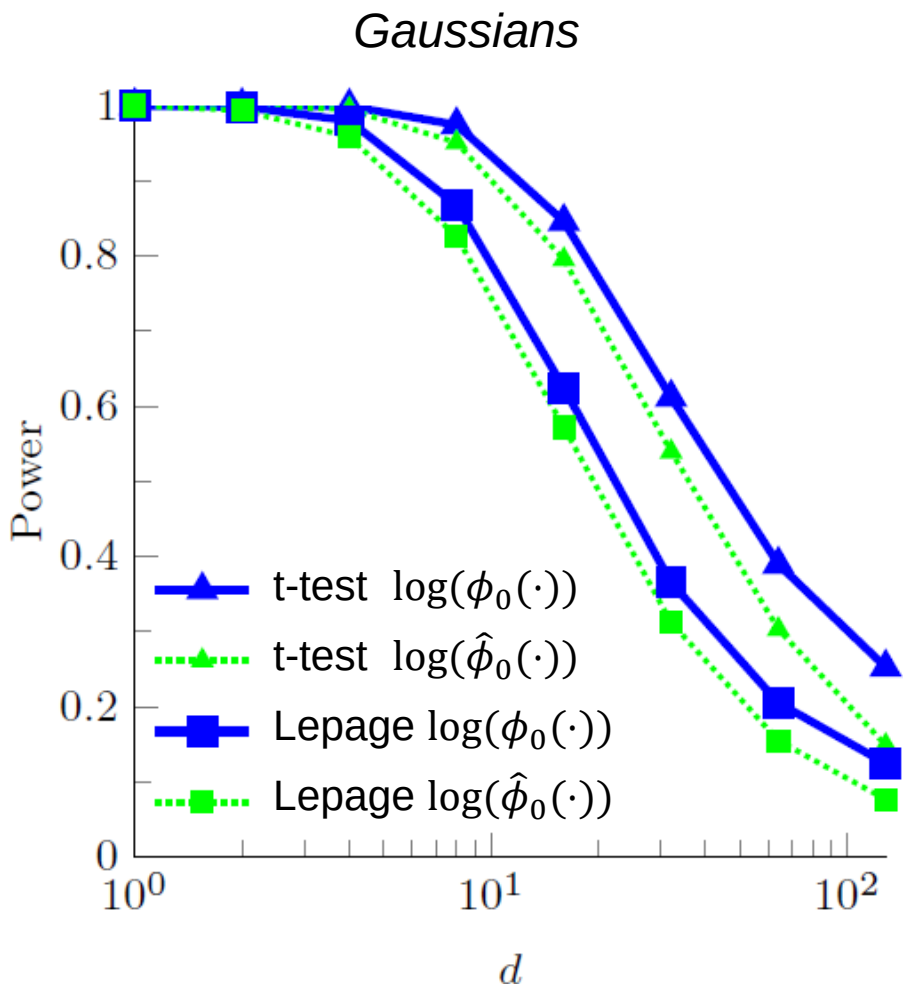
$$\mathcal{T}(W_0, W_1) \lessgtr h$$

where h controls the amount of false positives

- Use the **power** of this **test** to assess change detectability



THE HYPOTHESIS TESTS POWER ON GAUSSIAN STREAMS



Remarks:

- ϕ_1 is defined analytically
- The t-test detects changes in the expectation of $\log(\phi_0(\cdot))$
- The Lepage test detects changes in the location and scale of $\log(\phi_0(\cdot))$

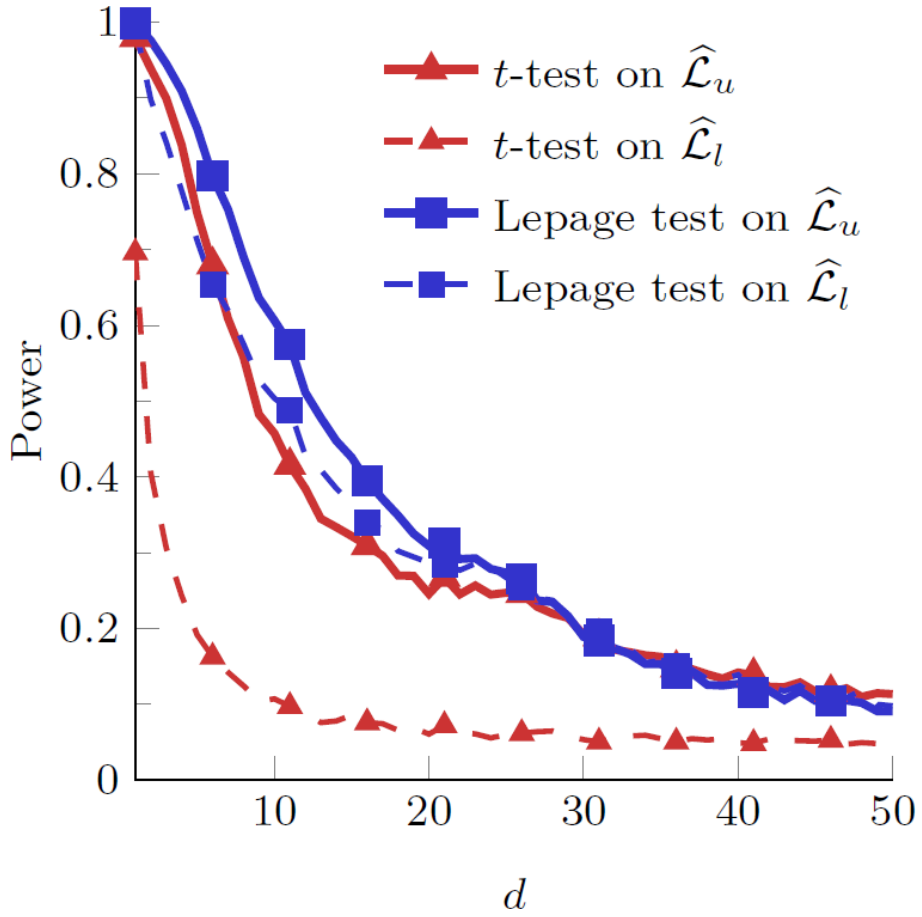
Results

- The HT power decays with d: DL does not only concern the upperbound of SNR.
- DL is not due to estimation errors, but these make things worst.
- Also the power of the Lepage HT decreases, which indicates that the change is more difficult to detect even when monitoring the variance

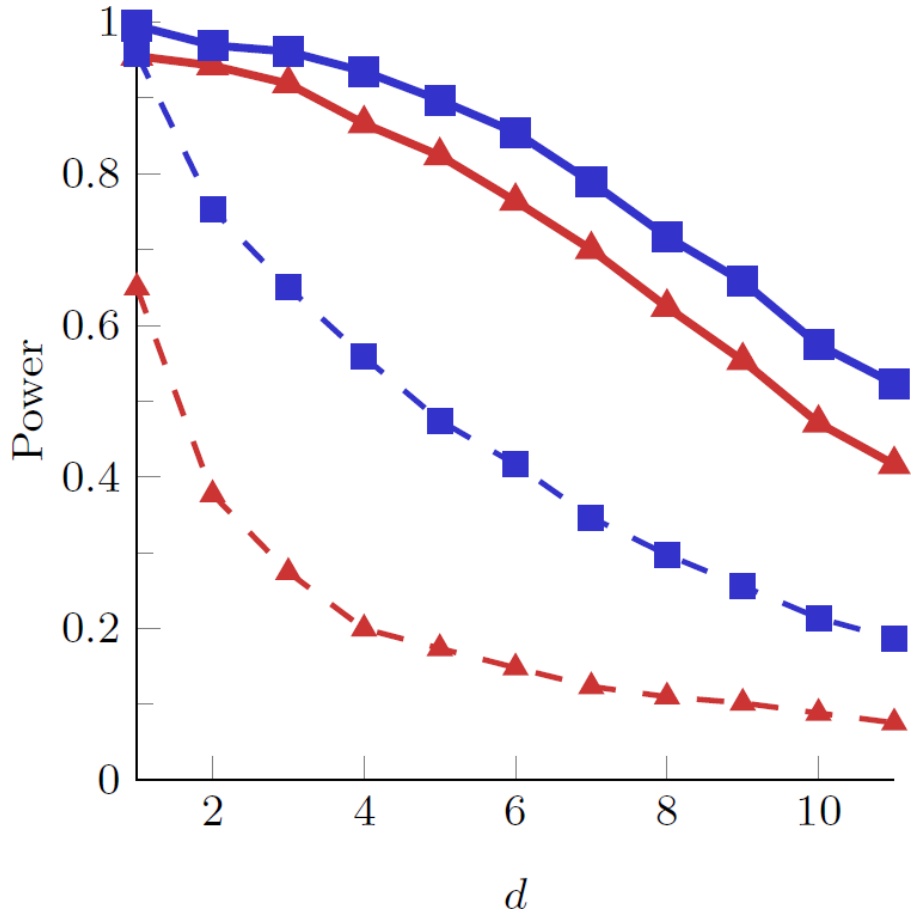
Boracchi, Carrera, ICPR 2020

THE HYPOTHESIS TESTS POWER ON UCI DATASETS

Particle

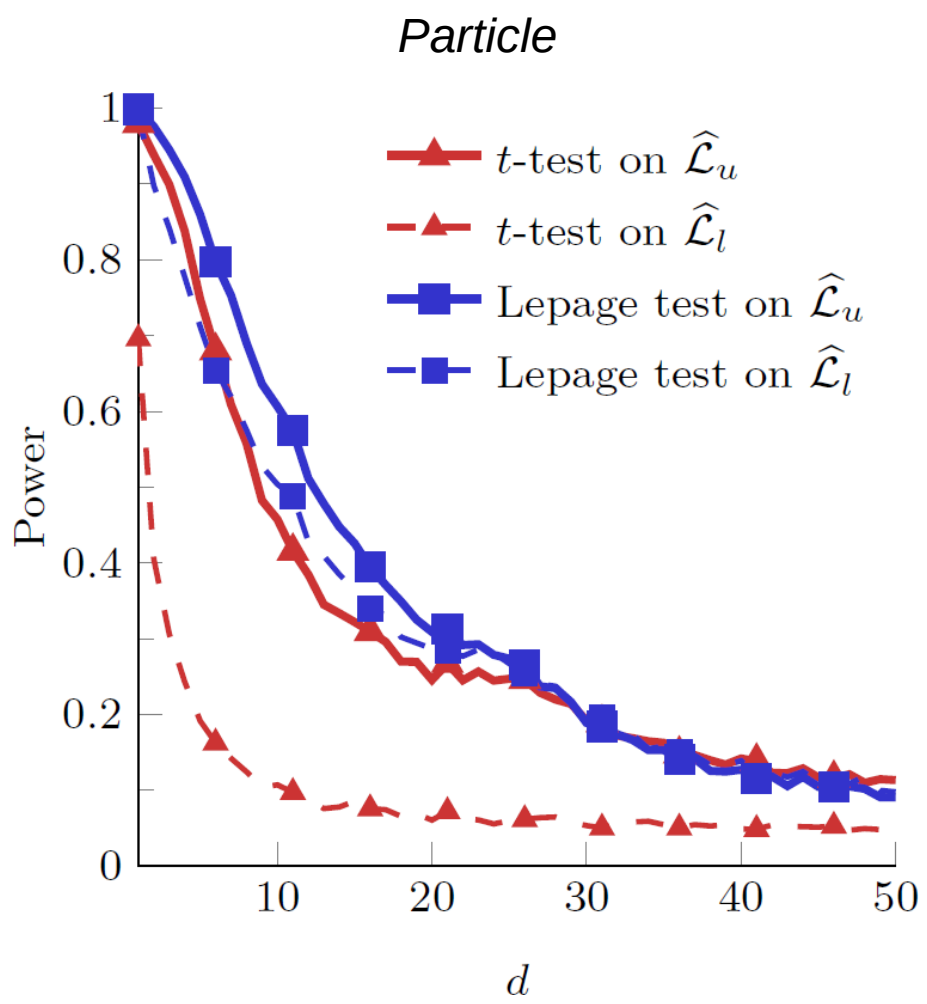


Wine



Boracchi, Carrera, ICPR 2020

THE HYPOTHESIS TESTS POWER ON PARTICLE DATASET



Remarks:

- ϕ_1 is defined through CCM a framework to control the change magnitude and yield $\mathrm{sKL}(\phi_0, \phi_1) \approx 1$
- $\hat{\phi}_0$ is a Gaussian Mixture where k is selected by cross-validation
- Approximated expression of $\mathcal{L}(\cdot)$ to prevent numerical approximations

Results:

- DL occurs also in non-Gaussian data approximated by GM
- DL is clearly visible at quite a low dimensions

C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "*Change Detection in Multivariate Datastreams: Likelihood and Detectability Loss*" IJCAI 2016

Any Questions?

CHANGE AND ANOMALY DETECTION IN IMAGES, SIGNALS AND DATASTREAMS

Giacomo Boracchi,

Politecnico di Milano, DEIB.

https://boracchi.faculty.polimi.it/

Diego Carrera,

System Research and Applications, STMicroelectronics, Agrate Brianza

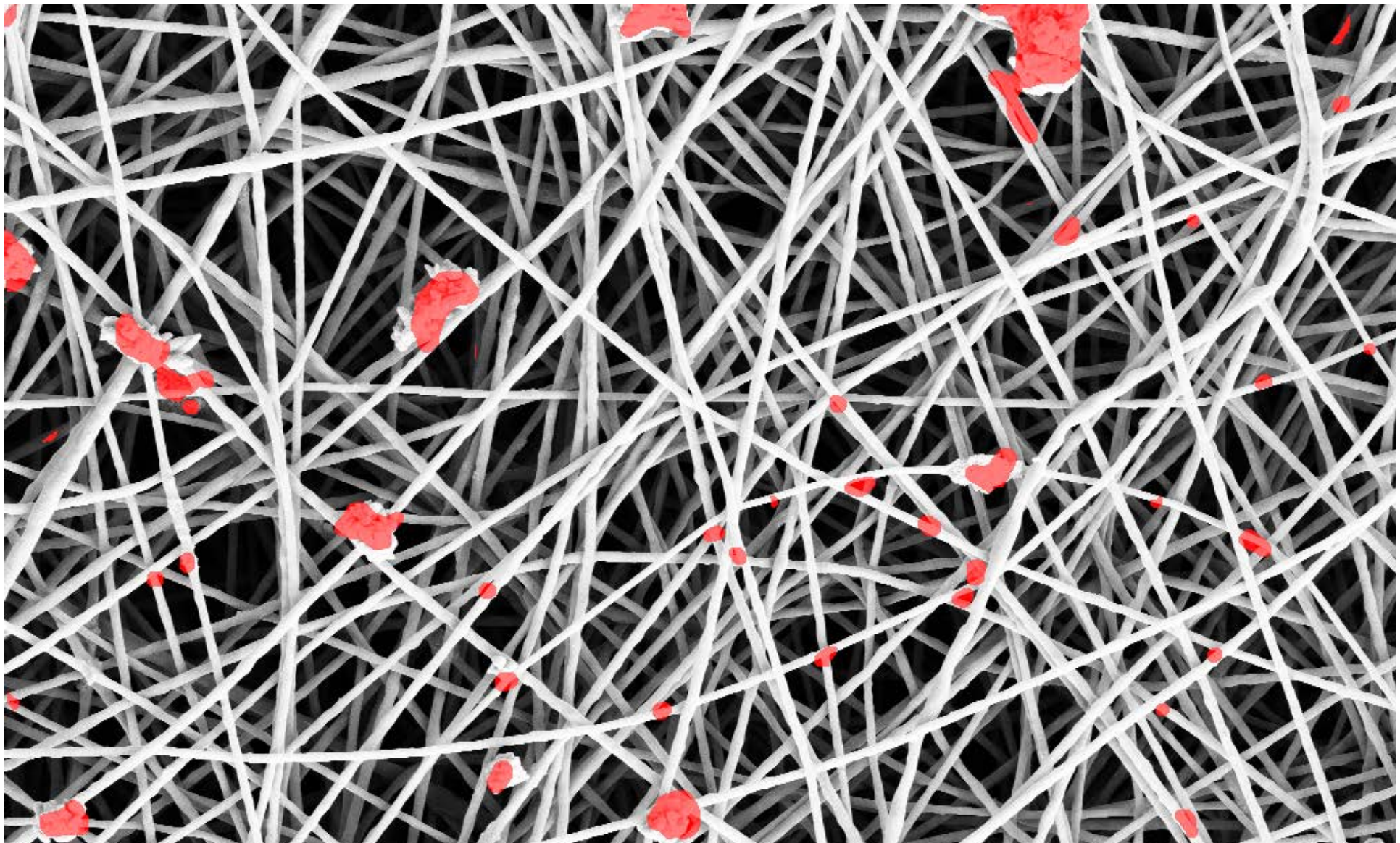
ICPR 2020, January 10, 2021

LET'S MOVE OUT OF THE RANDOM VARIABLE WORLD!

Applying statistical methods to signals / image patches

OUR RUNNING EXAMPLE

Goal: Automatically measure area covered by defects



Boracchi, Carrera, ICPR 2020

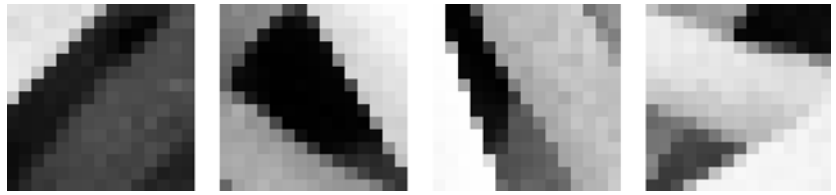
ANOMALY DETECTION IN IMAGES

The goal not determining whether the whole image is normal or anomalous, but **locate/segment possible anomalies**

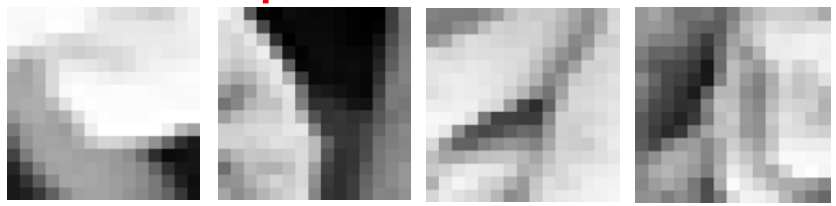
Therefore, it is convenient to

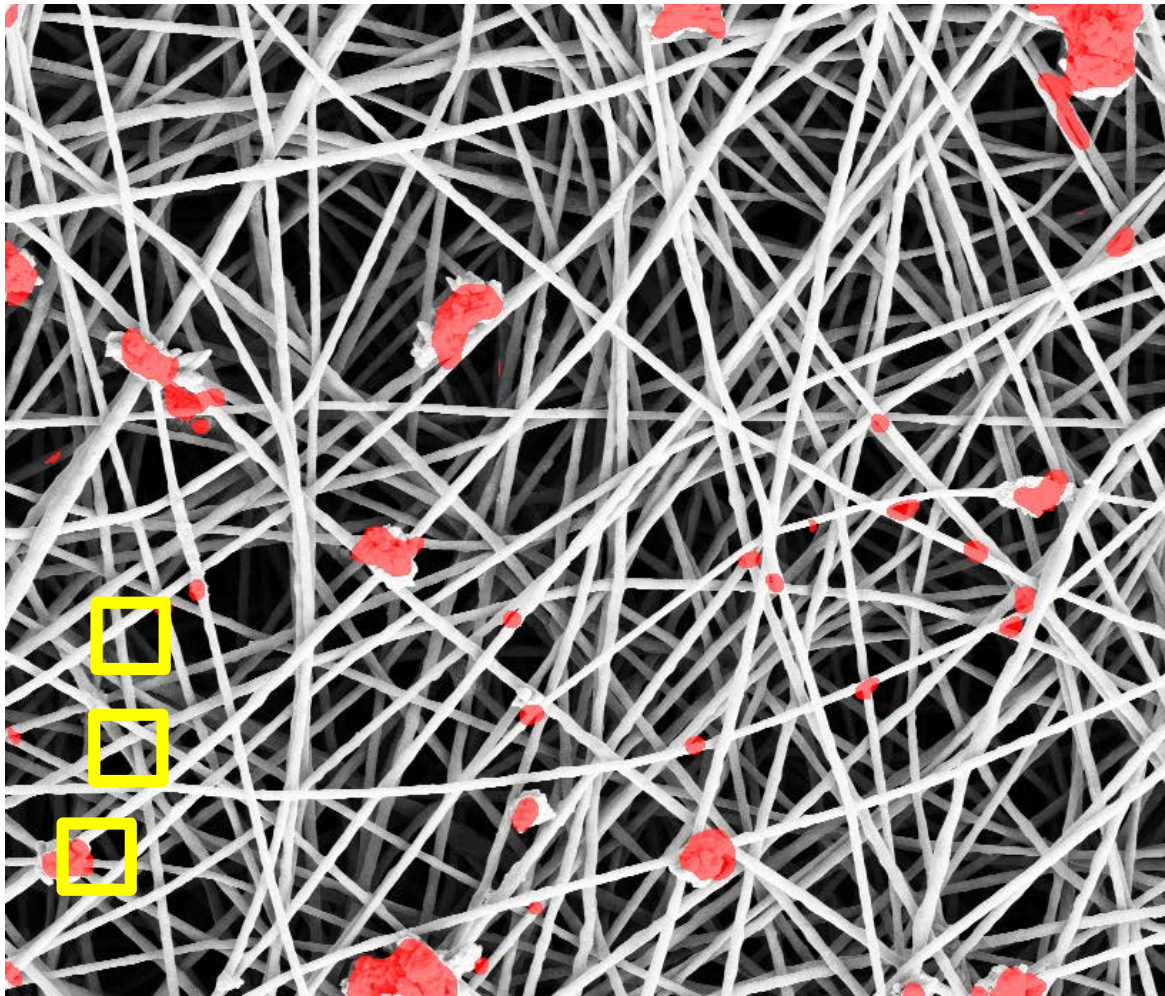
1. **Analyze the image patch-wise**
2. Isolate regions containing patches that are detected as as anomalies

Normal patches



Anomalous patches

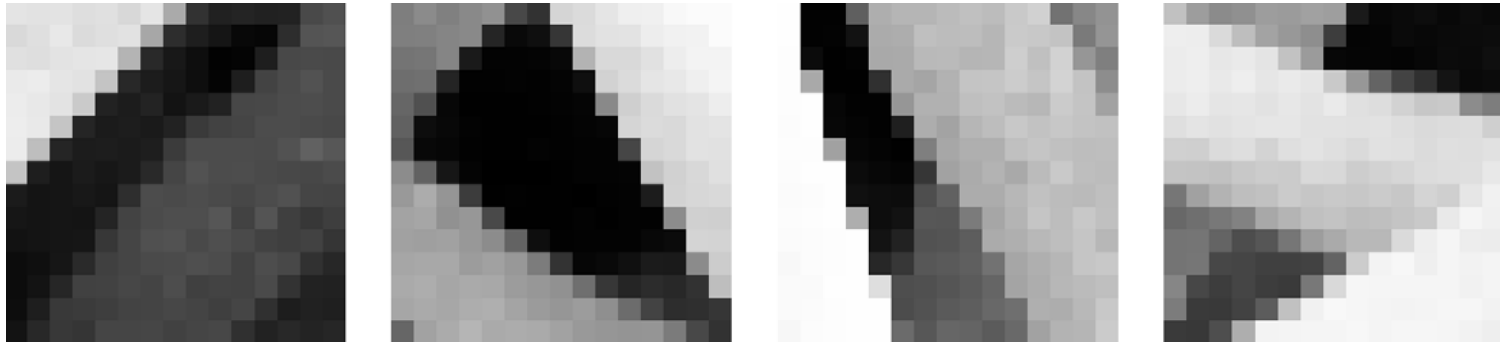




REAL WORLD DETECTION PROBLEMS

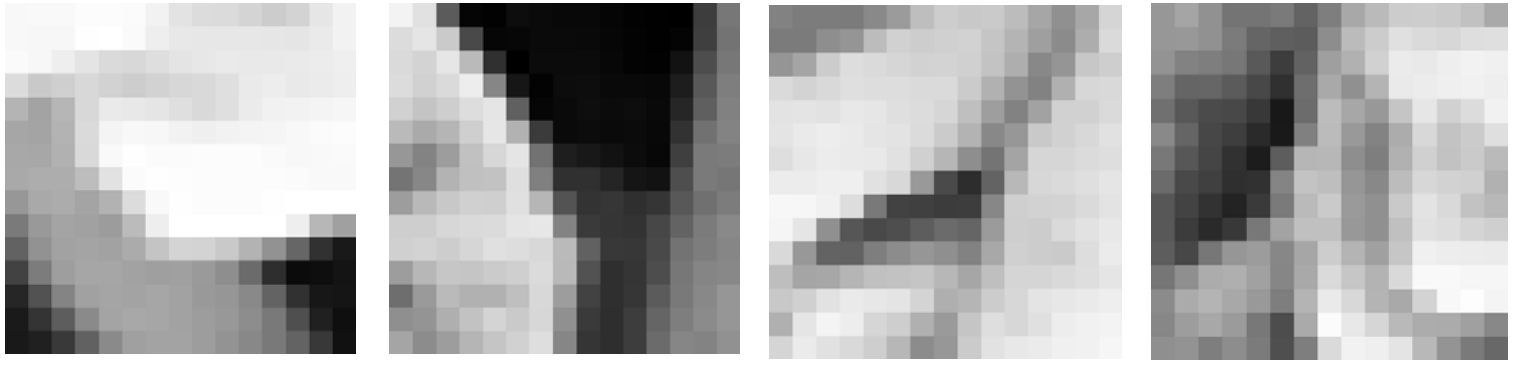
Normal patches -> background

- Exhibit a specific structure (geometry) or intensities



Anomalous patches:

- Are rare elements that do not confrom with the background



Boracchi, Carrera, ICPR 2020

Can we pursue approaches designed for random variables on image patches?

DENSITY-BASED APPROACH ON IMAGE PATCHES

A density-based approach to AD would be:

Training

i. Split the normal image in patches $\boldsymbol{s}$

ii. Fit a statistical model $\hat{\phi}_0 = \mathcal{N}(\mu, \Sigma)$ describing normal patches.

Testing

i. Split the test image in patches

ii. Compute $\hat{\phi}_0(\boldsymbol{s})$ the likelihood of each test patch $\boldsymbol{s}$

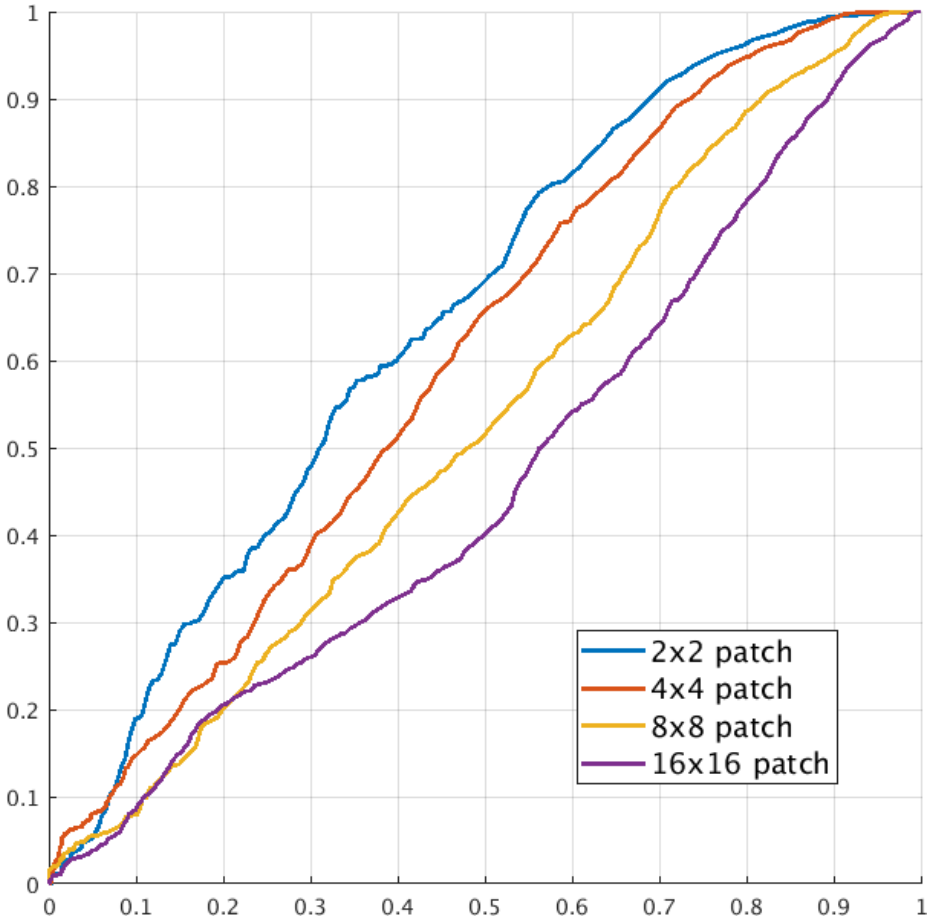
iii. Detect anomalies by thresholding the likelihood

This model is rarely accurate on natural images. Small patches (e.g. 2×2 or 5×5) are typically preferred

Du, B., Zhang, L.: *Random-selection-based anomaly detector for hyperspectral imagery*. IEEE Transactions on Geoscience and Remote sensing

X Xie, M Mirmehdi *"Texture exemplars for defect detection on random textures"* – ICPR 2005

THE LIMITATIONS OF THE RANDOM VARIABLE MODEL



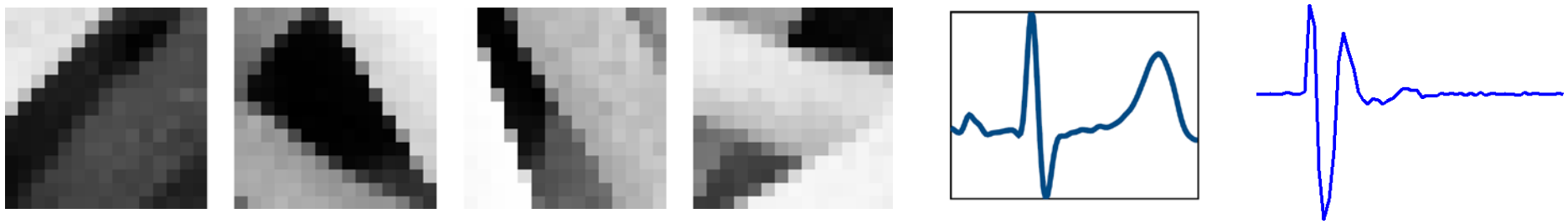
- ROC curve indicates very low performance on the nanofiber images
- Density model is not accurate on these images (and rarely is on natural images)
- Small patches (e.g. 2×2 or 5×5) are typically preferred. The model becomes even more unfit as the patch size increases

Du, B., Zhang, L.: *Random-selection-based anomaly detector for hyperspectral imagery*. IEEE Transactions on Geoscience and Remote sensing

X Xie, M Mirmehdi *"Texture exemplars for defect detection on random textures"* – ICPR 2005

REAL WORLD DETECTION PROBLEMS

Random variable model **does not successfully apply to signals or images** (not even small portions)

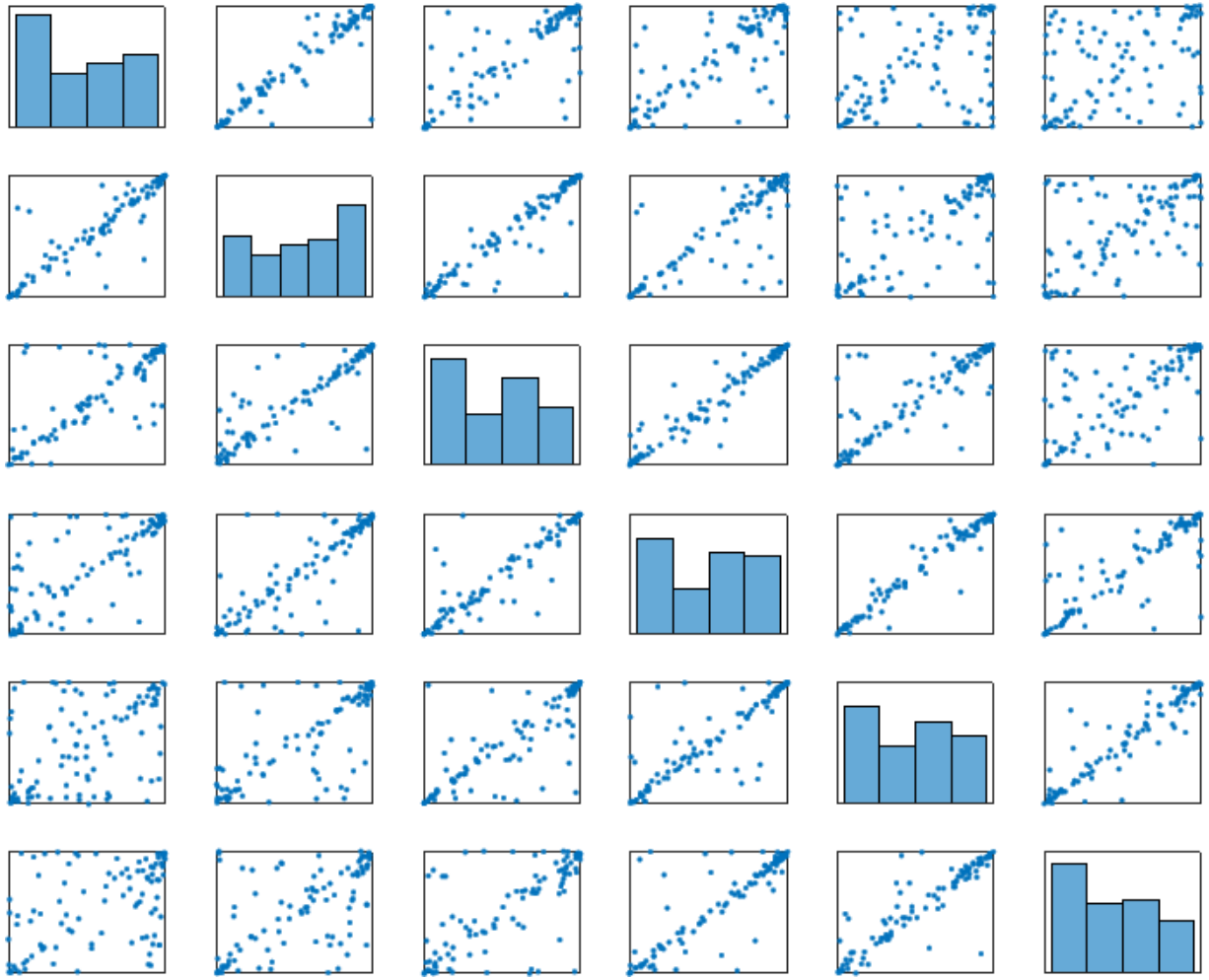


Stacking each patch/signal $\boldsymbol{s} \in \mathbb{R}^d$ in a vector $\boldsymbol{x}$ is not convenient:

- **Data dimension** d becomes **huge**
- **Strong correlations** among components, **difficult to directly model** by a probability density function ϕ_0

THE LIMITATIONS OF THE RANDOM VARIABLE MODEL

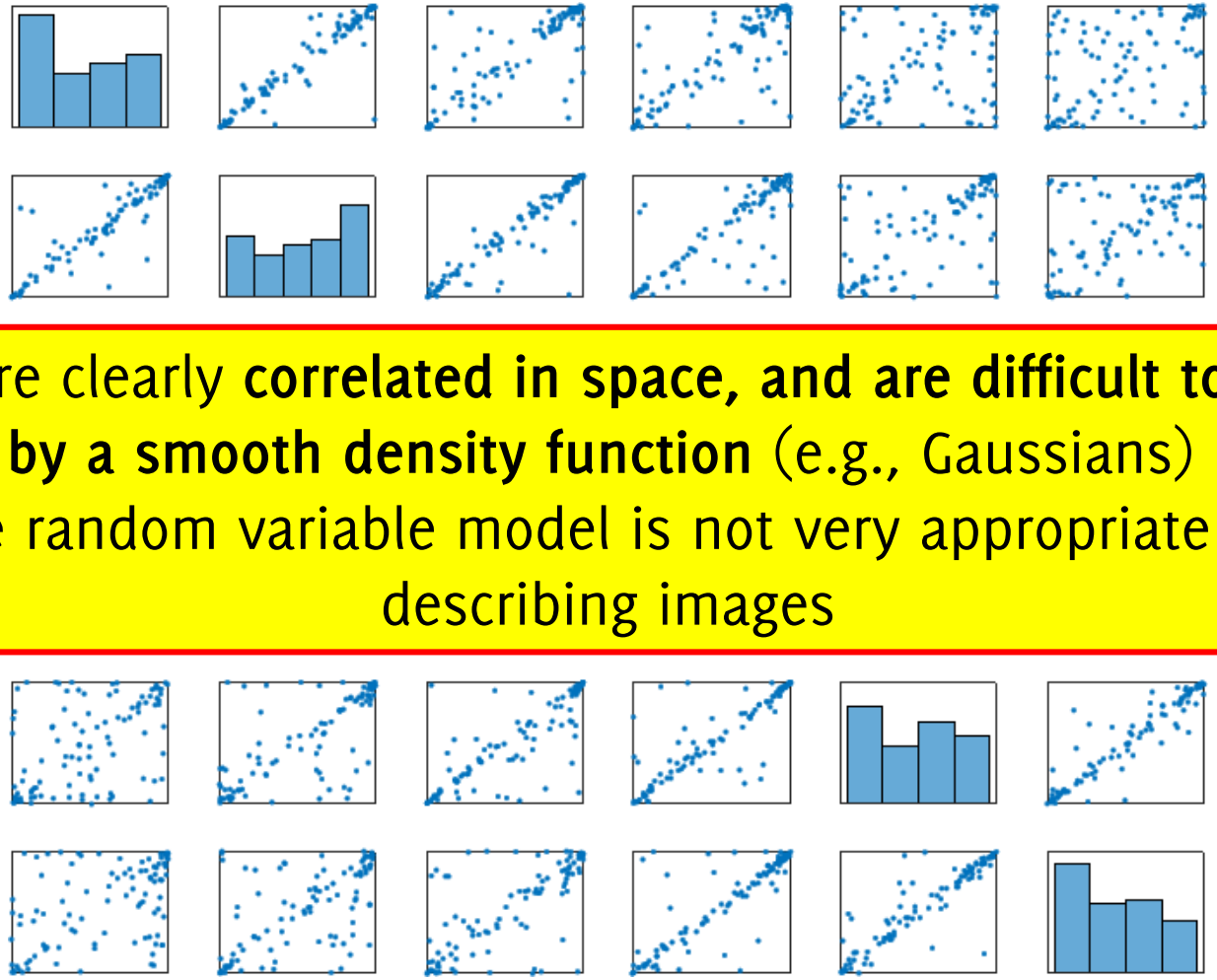
Distribution of adjacent pixel values inside a patch:



Boracchi, Carrera, ICPR 2020

THE LIMITATIONS OF THE RANDOM VARIABLE MODEL

Distribution of adjacent pixel values inside a patch:

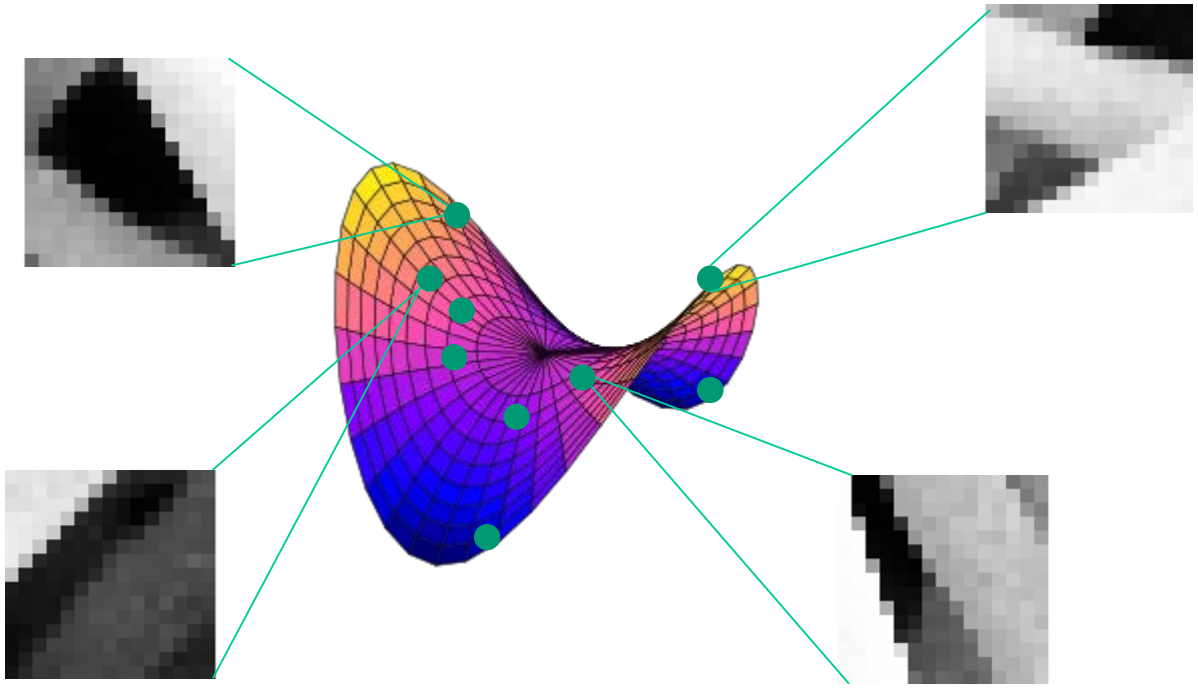


Data are clearly **correlated in space, and are difficult to model by a smooth density function** (e.g., Gaussians)

The random variable model is not very appropriate for describing images

THE LIMITATIONS OF THE RANDOM VARIABLE MODEL

Patches from natural images leave close to a low dimensional manifold

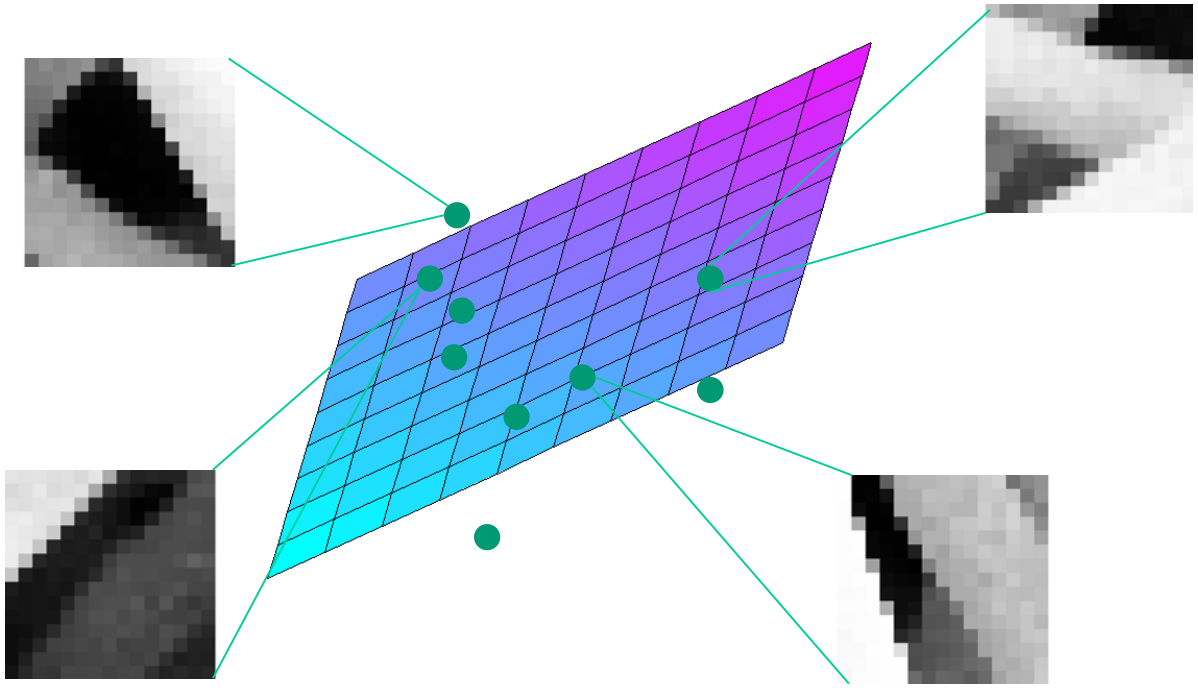


These means that patches can be well described by few latent variables

Bengio, Courville, Vincent, “Representation Learning: A Review and New Perspectives”, IEEE Pattern Analysis and Machine Intelligence 2013

A SIMPLE EXPERIMENT

Let’s approximate this manifold with the simplest one: a linear subspace

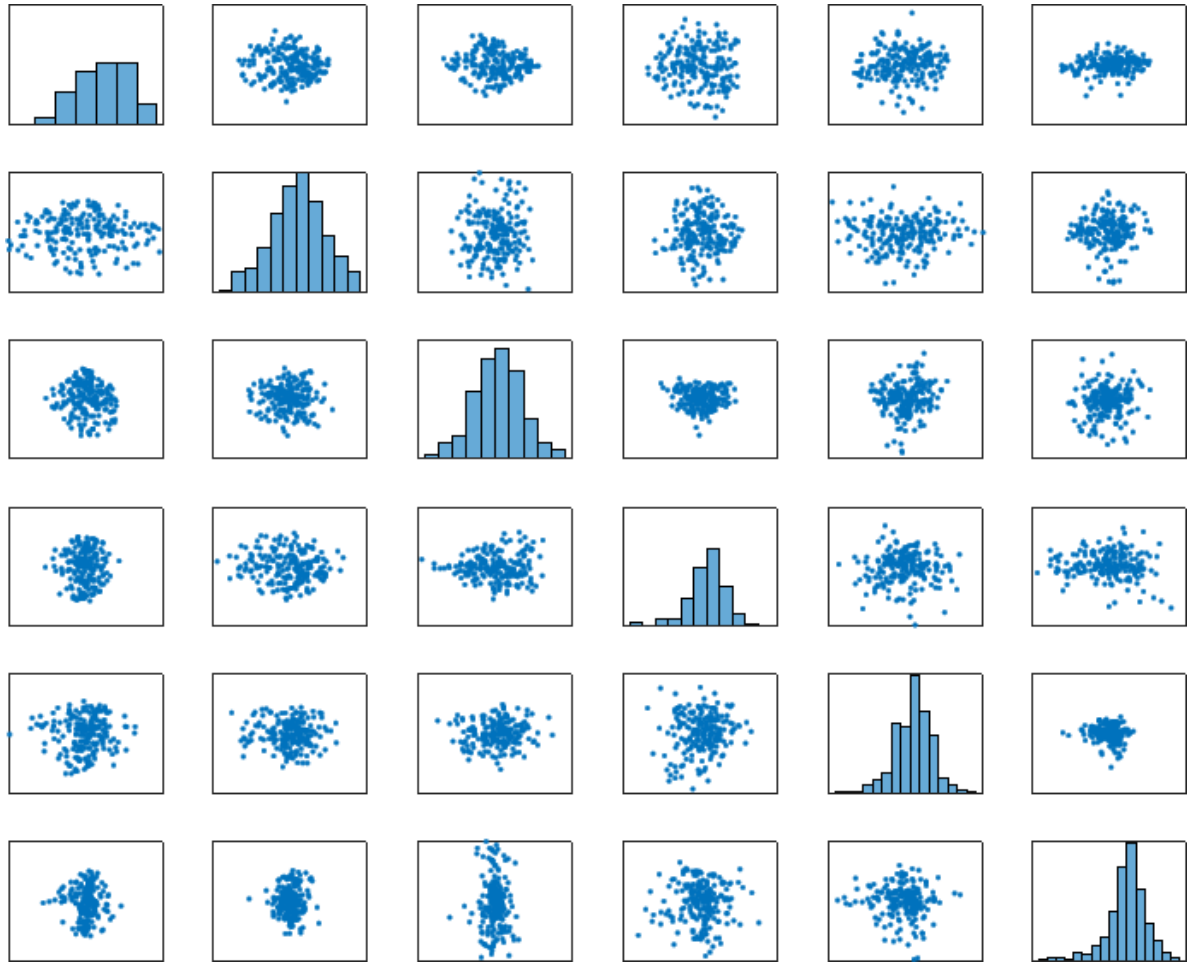


In practice, we compute the PCA of training patches. Consider the PCA score as latent variables, which means projecting each patch over the linear subspace spanned by the first components.

Boracchi, Carrera, ICPR 2020

A SIMPLE EXPERIMENT

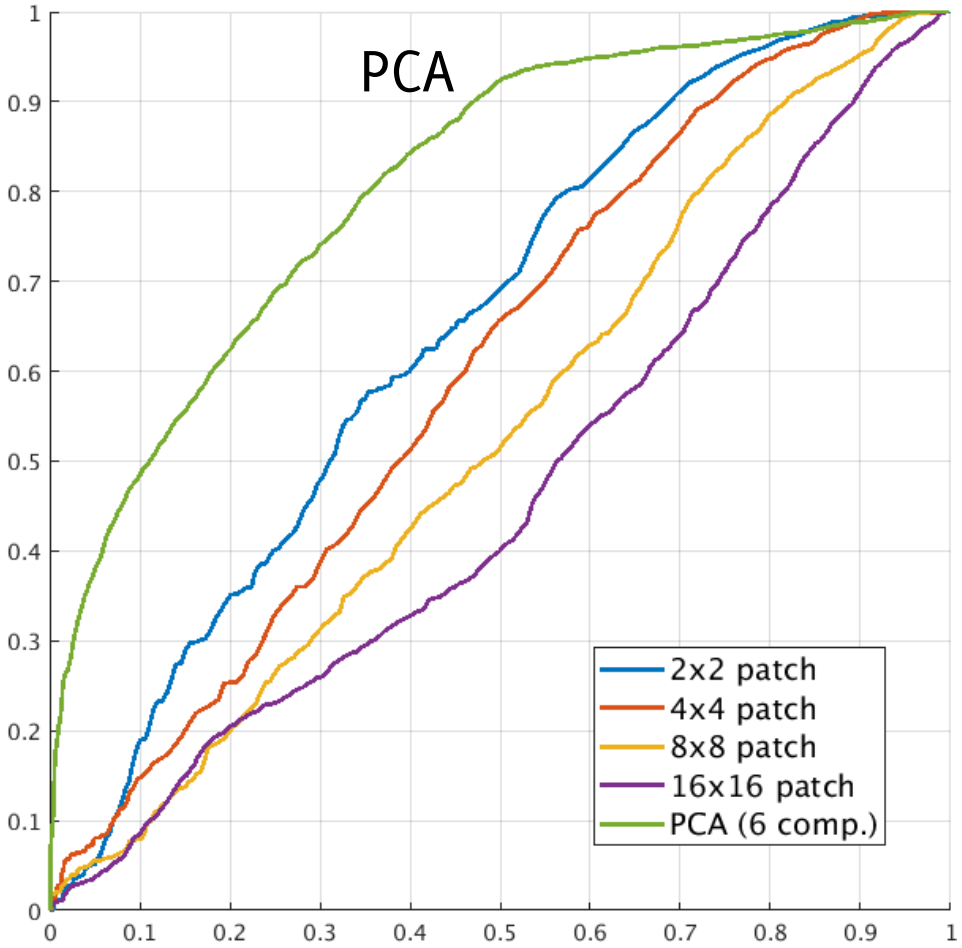
Distribution of first 6 PCA coefficients:



Boracchi, Carrera, ICPR 2020

A SIMPLE EXPERIMENT

We fit a $\hat{\phi}_0 = \mathcal{N}(\mu, \Sigma)$ on PCA components to describe normal patches, and perform anomaly detection



Boracchi, Carrera, ICPR 2020

ANOMALY DETECTION IN IMAGES AND SIGNALS

Out of the “Random Variable” World:
signal-based models for images

THE TYPICAL APPROACH

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Use the background model to provide, for each test signal/patch, an **anomaly score,** or measure of rareness
3. Apply a **decision rule** to the anomaly score to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth detections and avoid isolated pixels that are not consistent with neighbourhoods

Remark: Statistical-based approaches seen before uses as background model the statistical distribution $\hat{\phi}_0$ and a statistic as anomaly score

Boracchi, Carrera, ICPR 2020

THE TYPICAL APPROACH

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Use the background model to provide, for each test signal/patch, an **anomaly score,** or measure of rareness
3. Apply a **decision rule** to the anomaly score to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth detections and avoid isolated pixels that are not consistent with neighbourhoods

Remark: Statistical-ba ckground model the statistical distribution

The background model is used to **bring an image patch into the "random variable world"**

Boracchi, Carrera, ICPR 2020

THE TYPICAL APPROACH

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Use the background model to provide, for each test signal/patch, an **anomaly score,** or measure of rareness
3. Apply a **decision rule** to the anomaly score to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth detections and avoid isolated pixels that are not consistent with neighbourhoods

Remark: Statistical ound model the statistical distribut

Once "having applied" the background model, one can use **anomaly detection methods for the "random variable world"**. This might require **fitting an additional model**

Boracchi, Carrera, ICPR 2020

THE THREE MAJOR INGREDIENTS

Most detection algorithms have three major ingredients:

- The background model $\mathcal{M}$, learned from normal data
- The statistic / anomaly score: $\mathrm{err}(\boldsymbol{s}), \mathcal{L}(\boldsymbol{s}), \mathcal{A}(\boldsymbol{s}), \ldots$
- Decision rule to detect, e.g. $\mathrm{err}(\boldsymbol{s}) \gtrless \gamma$

SEMI-SUPERVISED ANOMALY-DETECTION IN IMAGES

Out of the "Random Variable" world

- Reconstruction-based methods
    - Subspace methods
- Feature-based monitoring
    - Expert-driven Features
    - Data-driven Features

SEMI-SUPERVISED ANOMALY-DETECTION IN IMAGES

Out of the "Random Variable" world

- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features

Boracchi, Carrera, ICPR 2020

RECONSTRUCTION-BASED METHODS

Fit a statistical model to the observation to ***describe dependence, apply anomaly detection*** *on the independent* ***residuals****.*

Detection is performed by **a model** $\mathcal{M}$ which **encodes and reconstructs normal data**:

- **During training:** learn the model $\mathcal{M}$ from training set S
- **During testing:**
  - Encode and reconstruct each test signal $\boldsymbol{s}$ through $\mathcal{M}$.
  - Assess $\mathrm{err}(\boldsymbol{s})$, the **residual** between $\boldsymbol{s}$ and its reconstruction through $\mathcal{M}$

$$\mathrm{err}(\boldsymbol{s}) = \|\boldsymbol{s} - \hat{\boldsymbol{s}}\|$$

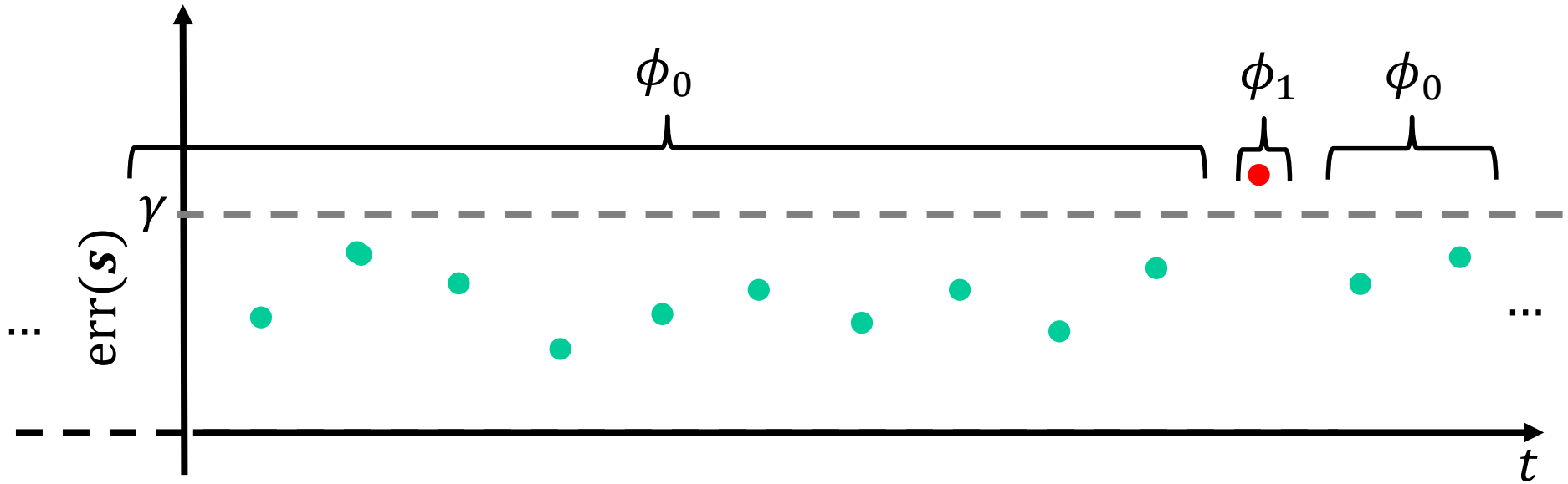
The rationale is that $\mathcal{M}$ **can reconstruct only normal data**, thus anomalies are expected to yield large reconstruction errors.

Boracchi, Carrera, ICPR 2020

MONITORING THE RECONSTRUCTION ERROR

Normal data are expected to yield values of $\mathrm{err}(\boldsymbol{s})$ that **are low**, while anomalies do not. This holds when the model $\mathcal{M}$ was specifically learned to describe normal data

Outliers can be detected by thresholding $\mathrm{err}(\boldsymbol{s})$



Boracchi, Carrera, ICPR 2020

RECONSTRUCTION-BASED METHODS

Popular models are:

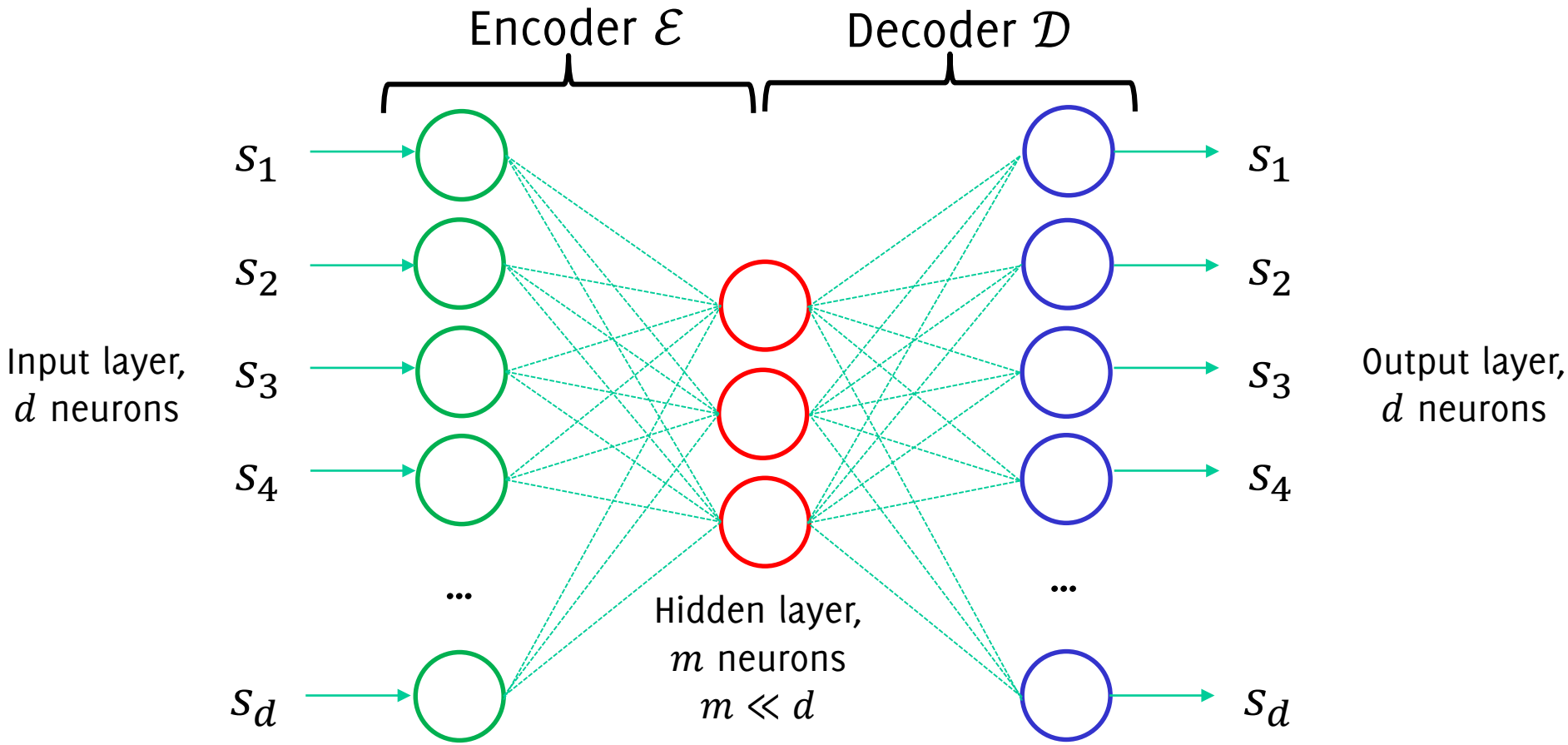
- **neural networks**, in particular auto-encoders, for higher dimensional data
- **projection on subspaces** / manifolds
- **dictionaries yielding sparse-representations**
- **autoregressive models** for time series (ARMA, ARIMA…)

Methods based on projections and dictionaries can be also interpreted as subspace methods

RECONSTRUCTION-BASED METHODS

Autoencoders are neural networks used for data reconstruction (they learn the identity function)

The typical structure of an autoencoder is:



Boracchi, Carrera, ICPR 2020

RECONSTRUCTION-BASED METHODS

Autoencoders are trained to reconstruct all the samples in the training set. The reconstruction loss over the training set S is

$$\mathcal{L}(S) = \sum_{\boldsymbol{s} \in S} \left\| \boldsymbol{s} - \mathcal{D}\big(\mathcal{E}(\boldsymbol{s})\big) \right\|_2$$

$\mathcal{D}\big(\mathcal{E}(\cdot)\big)$ is trained via standard backpropagation algorithms (e.g. SGD)

Remarks

- Typically $\mathcal{D}\big(\mathcal{E}(\cdot)\big)$ does not provide perfect reconstruction, since $m \ll d$.
- **Regularization terms** might be included in the loss function for the latent representation $\mathcal{E}(\boldsymbol{s})$ to feature specific properties

Bengio, Y., Courville, A., Vincent, P. *"Representation learning: A review and new perspectives"*. IEEE TPAMI 2013

Mishne, G., Shaham, U., Cloninger, A., & Cohen, I. *Diffusion nets*. Applied and Computational Harmonic Analysis (2017).

MONITORING THE RECONSTRUCTION ERROR

Detection by reconstruction error monitoring (AE notation)

Training (Monitoring the Reconstruction Error):

1. Train the model $\mathcal{D}(\mathcal{E}(\cdot))$ from the training set S
2. Learn the distribution of reconstruction errors

$$\mathrm{err}(\boldsymbol{s}) = \left\| \boldsymbol{s} - \mathcal{D}\left(\mathcal{E}(\boldsymbol{s})\right) \right\|_2, \qquad \boldsymbol{s} \in V$$

over a validation set V, such that $V \cap S = \emptyset$, and define a suitable threshold γ

Testing (Monitoring the Reconstruction Error):

1. Perform encoding and compute the reconstruction error

$$\mathrm{err}(\boldsymbol{s}) = \left\| \boldsymbol{s} - \mathcal{D}\left(\mathcal{E}(\boldsymbol{s})\right) \right\|_2$$

2. Consider $\boldsymbol{s}$ anomalous when $\mathrm{err}(\boldsymbol{s}) > \gamma$

OUTLINE ON SEMI-SUPERVISED APPROACHES

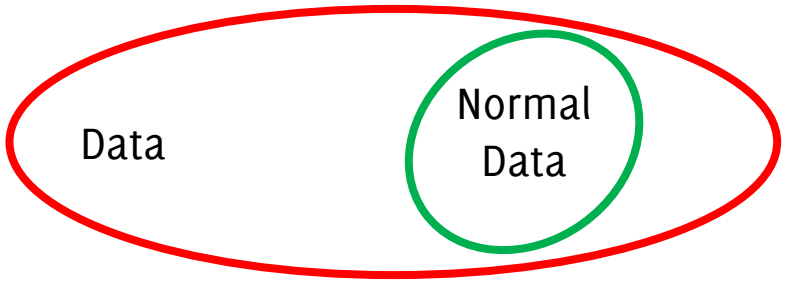
Out of the "Random Variable" world

- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features
  - Extended models

SUBSPACE METHODS

The underlying assumption is that

- **normal patches live in a subspace** that can be identified by S
- **anomalies can be detected by projecting test patches** in such subspace and by monitoring the reconstruction error (distance with the projection)

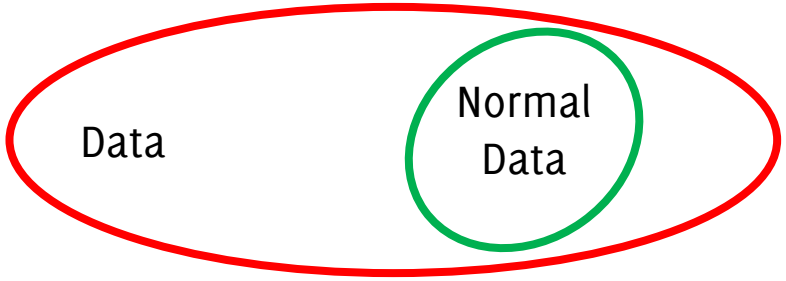


V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

SUBSPACE METHODS

A few example of **models used for describing normal patches**:

- Orthogonal basis: normal patches can be expressed by **a few selected basis elements** (Fourier, Wavelets..)
- PCA: normal patches live in the **linear subspace of the first components**
- Robust PCA: defined on the ℓ^1 distance to be **insensitive to outliers in normal data**
- Kernel PCA: normal patches live in **a non-linear manifold**
- **Dictionaries** yielding **sparse representations**
- Random projections



V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

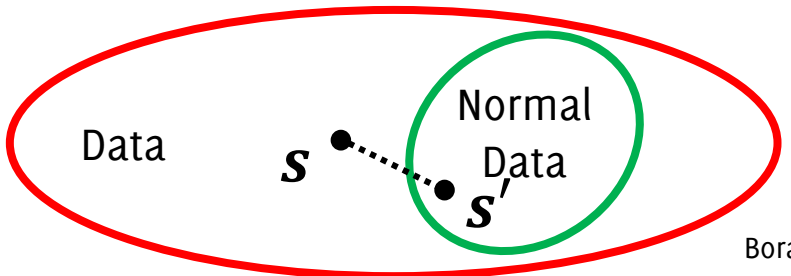
SUBSPACE METHODS: PCA-BASED MONITORING

Anomaly detection based on PCA (and similar techniques):

1. Compute the **projection on the subspace**,
$$\boldsymbol{s}' = P^T \boldsymbol{s}, \qquad P \in \mathbb{R}^{d \times m}, \qquad m \ll d$$
which is the projection over the first m principal components and a way to reduce data-dimensionality.

2. Monitor the **reconstruction error**:
$$\mathrm{err}(\mathbf{s}) = \|\boldsymbol{s} - PP^T \boldsymbol{s}\|_2$$
which is the distance between $\boldsymbol{s}$ and its projection $PP^T \boldsymbol{s}$ over the subspace of normal patches

2. [bis] The **projection along the last principal component**, is also a good anomaly score, as it becomes
large at anomalies.



Boracchi, Carrera, ICPR 2020

SUBSPACE METHODS: SPARSE REPRESENTATIONS

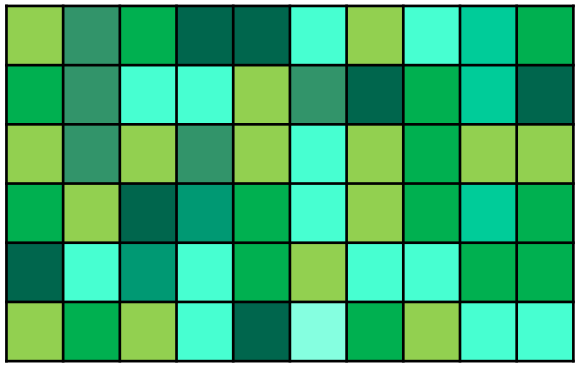
Basic assumption: normal data live in **a union of low-dimensional subspaces** of the input space

- The model learned from S is a matrix: the **dictionary** D.
- Each signal is decomposed as **the sum of a few dictionary atoms** (representation is constrained to be **sparse**).
- **Atoms** represent the many **building blocks** that can be used to reconstruct normal signals.
- There are typically **more atoms** than the signal dimension (redundant dictionaries).
- Effective as long as the learned **dictionary** D is **very specific for normal data**

M. Elad *"Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing"*, Springer, 2010

DICTIONARIES YIELDING SPARSE REPRESENTATIONS

Dictionaries are just matrices: $D \in \mathbb{R}^{d \times m}$

$D =$ 

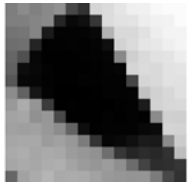
Boracchi, Carrera, ICPR 2020

DICTIONARIES YIELDING SPARSE REPRESENTATIONS

Dictionaries are just matrices: $D \in \mathbb{R}^{d \times m}$

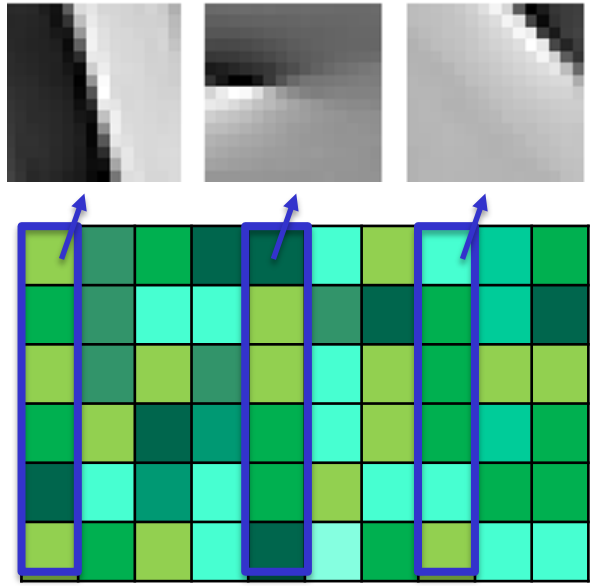
Each column of D is an atom:

- lives in the input space $\mathbb{R}^d$
- it is one of the building blocks that have been learned to reconstruct the input signal in the training set S



s

$D =$



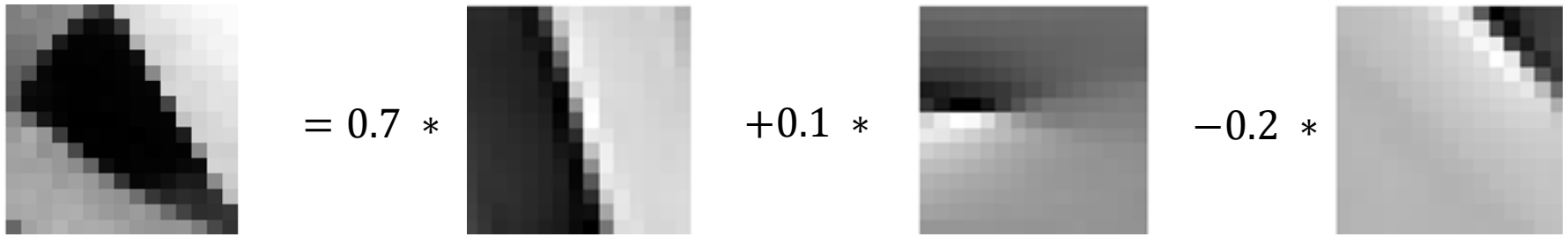
SPARSE REPRESENTATIONS

Let $\boldsymbol{s} \in \mathbb{R}^d$ be the input signal, a sparse representation is

$$\boldsymbol{s} = \sum_{i=1}^{m} \alpha_i \, \boldsymbol{d_i}$$

a linear combination of **few dictionary atoms** $\{\boldsymbol{d_i}\}$, i.e., most of coefficients are such that $\alpha_i = 0$

An illustrative example in case of our patches

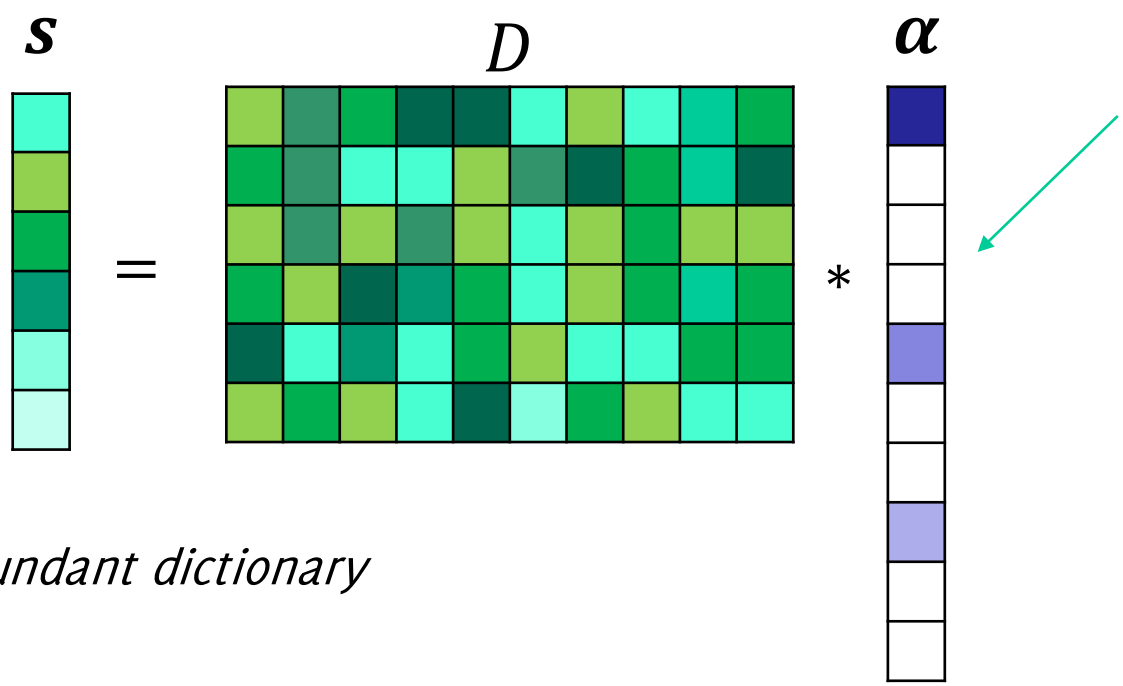


SPARSE REPRESENTATIONS IN MATRIX EXPRESSION

Let $\boldsymbol{s} \in \mathbb{R}^d$ be the input signal, a sparse representation is

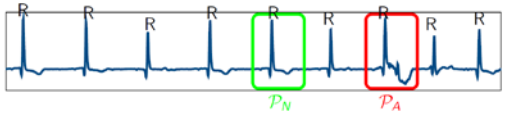
$$\boldsymbol{s} = \sum_{i=1}^{m} \alpha_i \, \boldsymbol{d_i} = D\boldsymbol{\alpha}, \qquad D \in \mathbb{R}^{d \times m}$$

a linear combination of **few dictionary atoms** $\{\boldsymbol{d_i}\}$ and $\|\boldsymbol{\alpha}\|_0 < L$, i.e. only a few coefficients are nonzero, i.e. $\boldsymbol{\alpha}$ is sparse.



Overcomplete / Redundant dictionary

Boracchi, Carrera, ICPR 2020

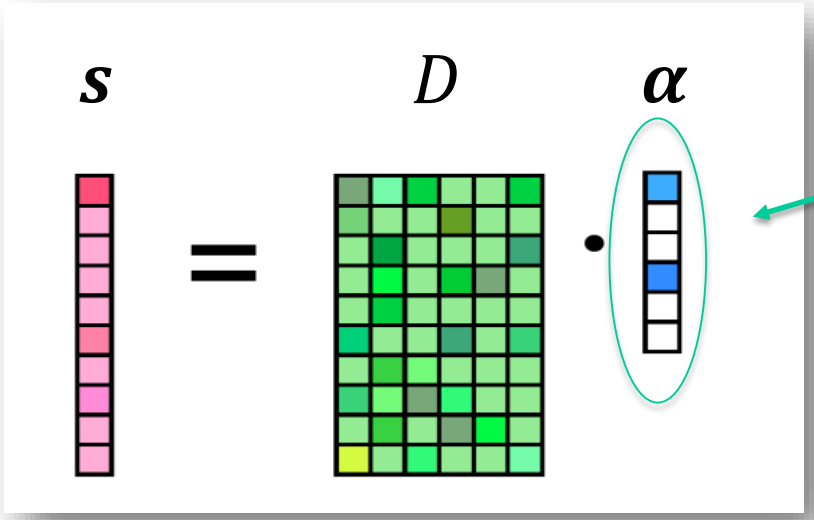


SPARSE REPRESENTATIONS IN MATRIX EXPRESSION

Let $\boldsymbol{s} \in \mathbb{R}^d$ be the input signal, a sparse representation is

$$\boldsymbol{s} = \sum_{i=1}^{m} \alpha_i \, \boldsymbol{d_i} = D\boldsymbol{\alpha}, \qquad D \in \mathbb{R}^{d \times m}$$

a linear combination of **few dictionary atoms** $\{\boldsymbol{d_i}\}$ and $\|\boldsymbol{\alpha}\|_0 < L$, i.e. only a few coefficients are nonzero, i.e. $\boldsymbol{\alpha}$ is sparse.



This vector
$\boldsymbol{\alpha} = [\alpha_1, \quad \ldots, \quad \alpha_m]$
is **sparse**

Undercomplete dictionary

D. Carrera, B. Rossi, D. Zambon, P. Fragneto, G. Boracchi "*ECG Monitoring in Wearable Devices by Sparse Models*" in Proceedings of ECML-PKDD 2016, 16 pages

D. Carrera, B. Rossi, P. Fragneto, G. Boracchi "*Online Anomaly Detection for Long-Term ECG Monitoring using Wearable Devices*", Pattern Recognition 2019

THE SPARSE CODING PROBLEM…

Sprase Coding: computing the sparse representation for an input signal $\boldsymbol{s}$ w.r.t. D

$$\boldsymbol{s} \in \mathbb{R}^d \quad \Rightarrow \quad \boldsymbol{\alpha} \in \mathbb{R}^m$$

Since $\boldsymbol{\alpha}$ has to be **sparse**, some **sparsity-promoting prior** need to be included. Most popular formulation for this **optimization problem** are:

- ℓ^0 constrained,

$$\boldsymbol{\alpha} = \underset{\boldsymbol{a}\in\mathbb{R}^m}{\operatorname{argmin}} \|D\boldsymbol{a} - s\|_2 \;\; \text{s.t.} \;\; \|\boldsymbol{a}\|_0 < L$$

solved for instance by Matching Pursuit (MP), Orthogonal Matching Pursuit (OMP),…

- ℓ^1 penalized (or constrained),

$$\boldsymbol{\alpha} = \underset{\boldsymbol{a}\in\mathbb{R}^n}{\operatorname{argmin}} \|D\boldsymbol{a} - \mathbf{s}\|_2^2 + \lambda\|\boldsymbol{a}\|_1, \qquad \lambda > 0$$

solved by BPDN, ISTA (proximal mapping), IRSL, ADMM… or any convex optimization tool.

Y. Pati, R. Rezaiifar, P. Krishnaprasad. *"Orthogonal Matching Pursuit: recursive function approximation with application to wavelet decomposition"*. Asilomar Conf. on Signals, Systems and Comput. 1993

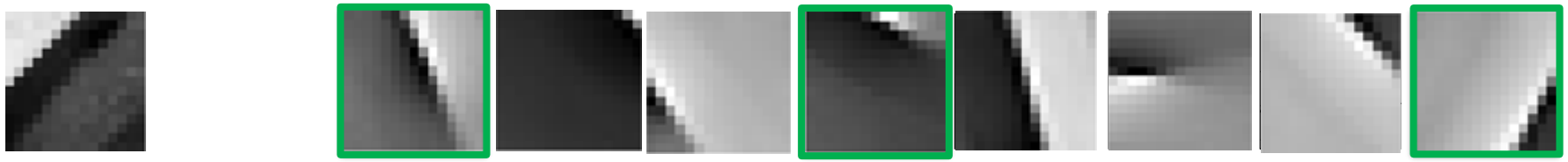
THE SPARSE CODING PROBLEM…

Sprase Coding: computing the sparse representation for an input signal $\boldsymbol{s}$ w.r.t. D

$$\boldsymbol{s} \in \mathbb{R}^d \quad \Rightarrow \quad \boldsymbol{\alpha} \in \mathbb{R}^m$$

It is solved as the following optimization problem, (e.g. via the Orthogonal Matching Pursuit, OMP)

$$\boldsymbol{\alpha} = \underset{\boldsymbol{a} \in \mathbb{R}^m}{\operatorname{argmin}} \|D\boldsymbol{a} - s\|_2 \quad \text{s.t.} \quad \|\boldsymbol{a}\|_0 < L$$



In this illustration $\boldsymbol{\alpha} = [0.7, 0, 0, 0.1, 0, 0, 0, -0.2]$

Boracchi, Carrera, ICPR 2020

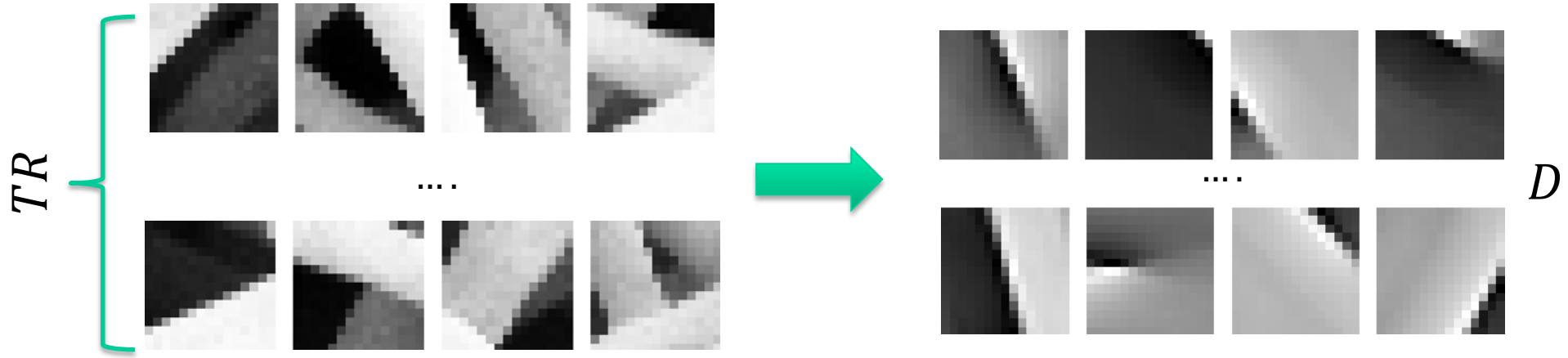
… AND DICTIONARY LEARNING

Dictionary Learning: estimate D from a training set of normal signals $S \subset \mathbb{R}^d$

$$S = \{\boldsymbol{s_1}, \dots \boldsymbol{s_n}\} \quad \Rightarrow \quad D \in \mathbb{R}^{d \times m}$$

It is solved as the following optimization problem typically via **block-coordinates descent** (e.g. KSVD algorithm)

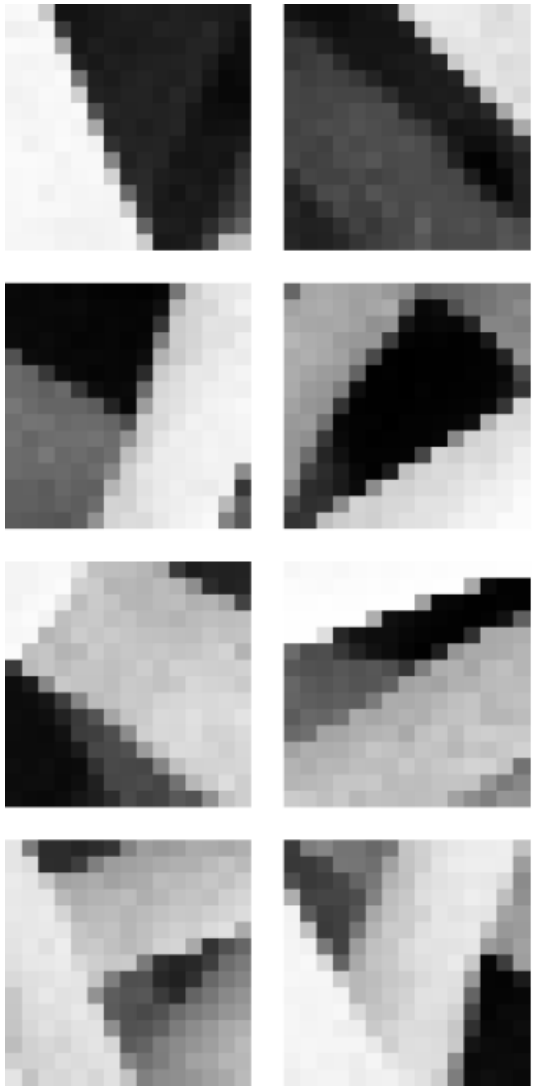
$$[D, X] = \underset{A \in \mathbb{R}^{d \times n},\, Y \in \mathbb{R}^{n \times m}}{\operatorname{argmin}} \|AY - S\|_2 \ \text{ s.t. } \ \|\boldsymbol{y_i}\|_0 < L, \qquad \forall \boldsymbol{y_i}$$



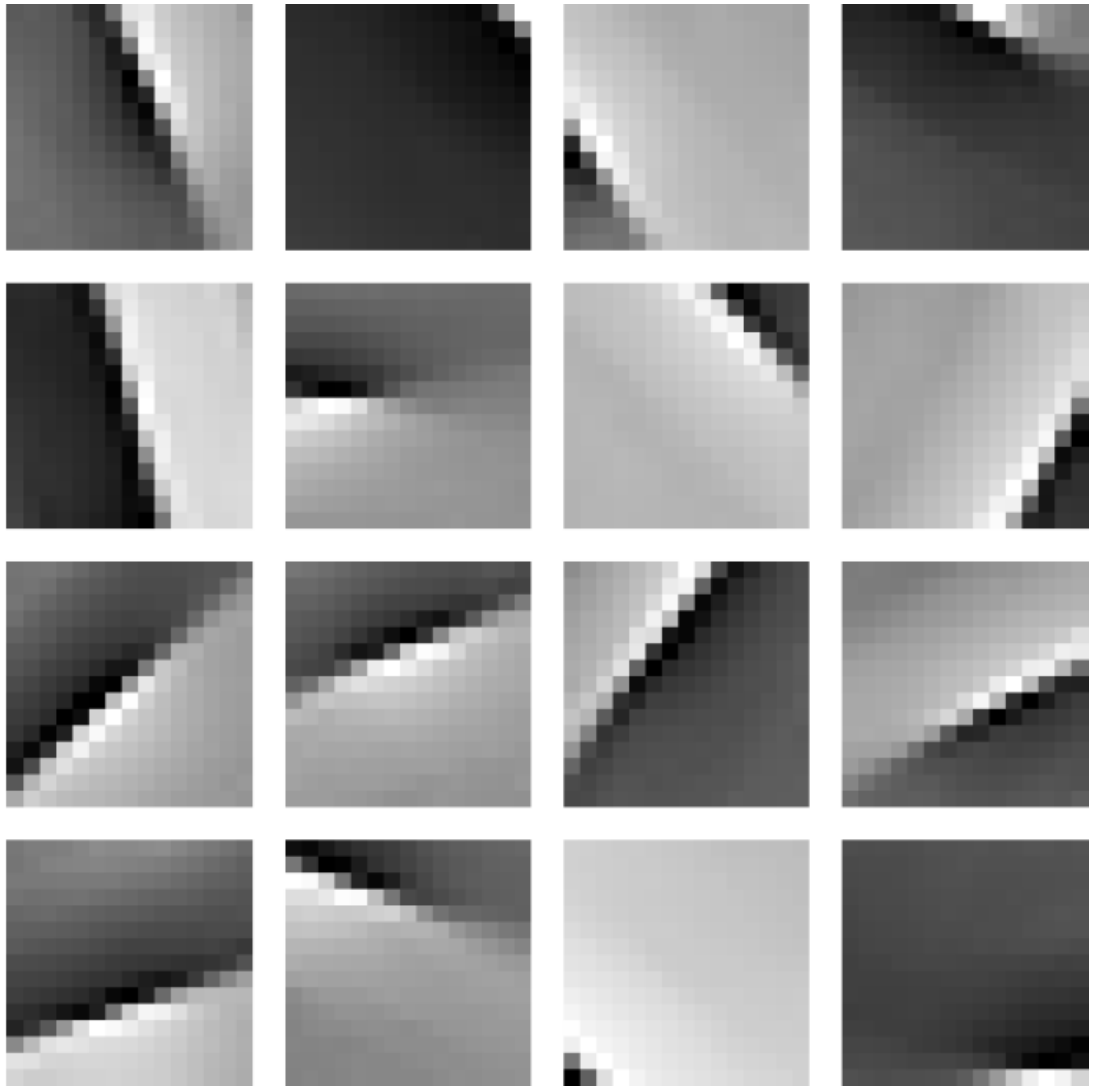
Aharon, M.; Elad, M. Bruckstein, A. *K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation* IEEE TSP, 2006

A DICTIONARY LEARNED FROM NORMAL PATCHES

Example of training patches

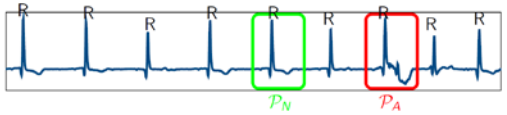


Few learned atoms (BPDN-based learning)

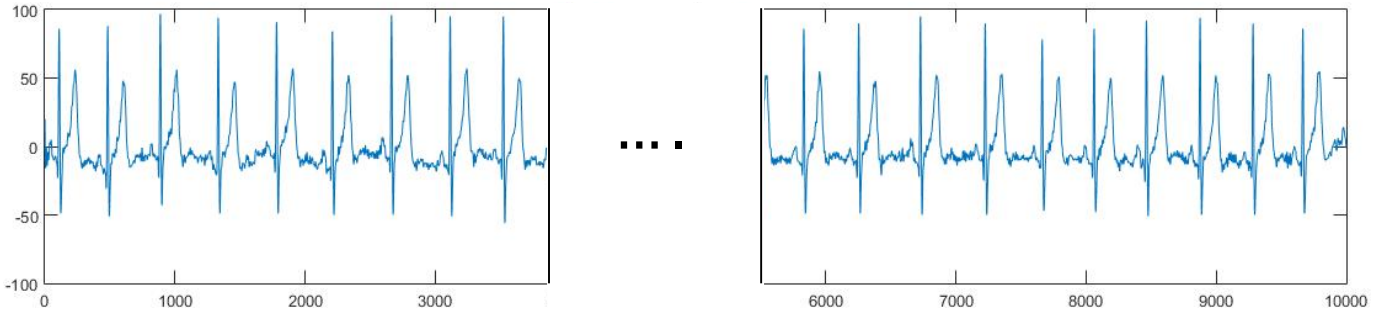


Boracchi, Carrera, ICPR 2020

A DICTIONARY LEARNED FROM NORMAL ECG TRACINGS

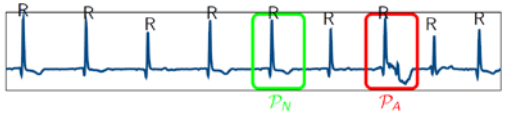


$$S = \{\boldsymbol{s_1}, \ldots \boldsymbol{s_M}\}$$



A few minutes of ECG signals in resting conditions

$$D \in \mathbb{R}^{n \times m}$$



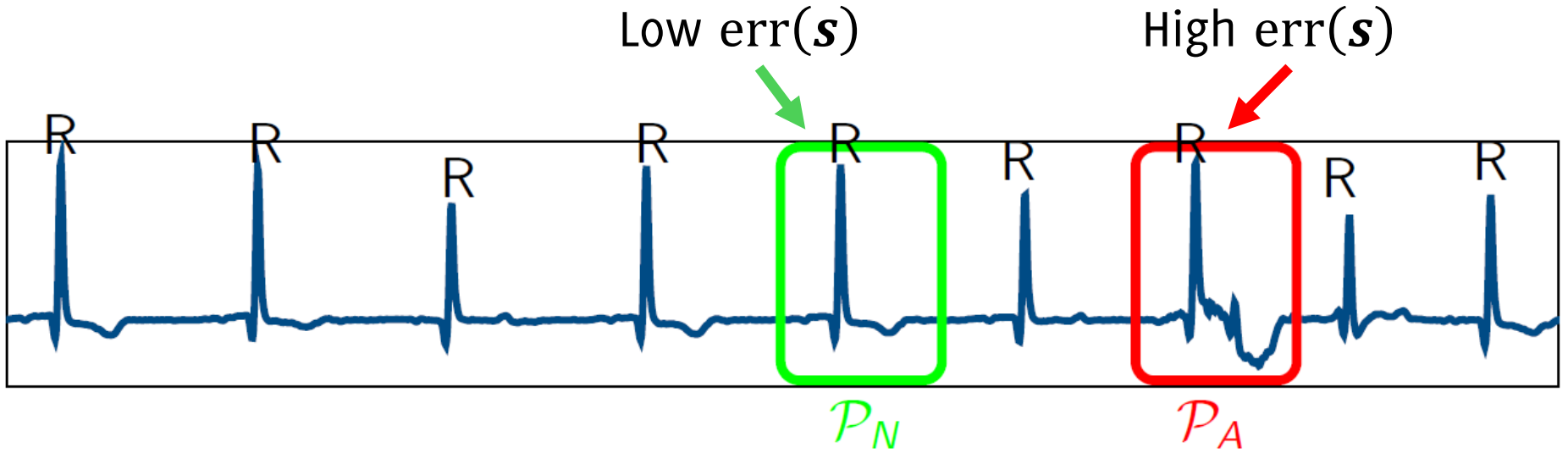
ANOMALY DETECTION BY MONITORING THE RECONSTRUCTION ERROR

Anomalies can be directly **detected by performing the sparse coding** of test signals and then analysing the reconstruction error

$$\text{err}(\boldsymbol{s}) = \|D\boldsymbol{\alpha} - \boldsymbol{s}\|_2$$

And as in reconstruction-based techniques, compare it against a threshold γ

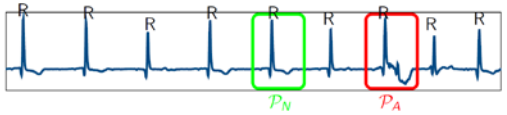
$$\|D\boldsymbol{x} - \mathbf{s}\|_2^2 < \gamma \rightarrow \mathbf{s} \text{ is normal}$$
$$\|D\boldsymbol{x} - \mathbf{s}\|_2^2 \geq \gamma \rightarrow \mathbf{s} \text{ is anomalous}$$



D. Carrera, B. Rossi, D. Zambon, P. Fragneto, G. Boracchi "*ECG Monitoring in Wearable Devices by Sparse Models*" in Proceedings of ECML-PKDD 2016, 16 pages

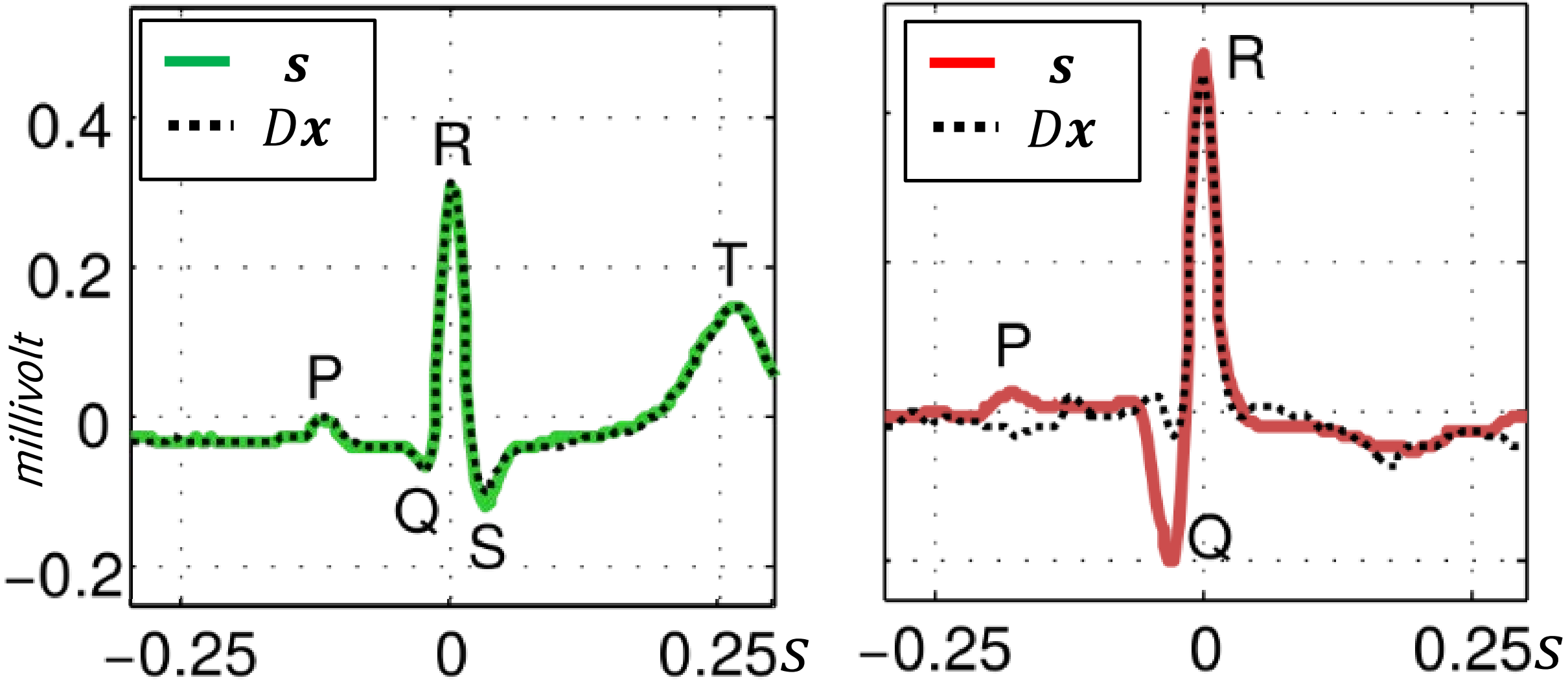
D. Carrera, B. Rossi, P. Fragneto, G. Boracchi "*Online Anomaly Detection for Long-Term ECG Monitoring using Wearable Devices*", Pattern Recognition 2019

ONLINE MONITORING THROUGH SPARSE REPRESENTATIONS



Normal beat: $\|D\boldsymbol{x} - \mathbf{s}\|_2^2 < \gamma$

Anomalies $\|D\boldsymbol{x} - \mathbf{s}\|_2^2 > \gamma$



Boracchi, Carrera, ICPR 2020

ANOMALY DETECTION DURING SPARSE CODING

Anomalies can be directly **detected during the sparse coding** stage, by adopting a special loss during optimization.

A set of test signals is modeled as:

$$S = DX + E + V$$

where X is sparse, V is a noise term, and E is a matrix having most columns set to zero. Columns $\boldsymbol{e}_i \neq \mathbf{0}$ indicate anomalies, as they do not admit a sparse representation w.r.t. D

A. Adler, M. Elad, Y. Hel-Or, and E. Rivlin, "*Sparse coding with anomaly detection*" Journal of Signal Processing Systems, vol. 79, no. 2, pp. 179–188, 2015.

ANOMALY DETECTION DURING SPARSE CODING

Anomalies can be detected by solving (through ADMM) the following sparse coding problem

$$\underset{X,E}{\operatorname{argmin}} \left(\frac{1}{2} \|S - DX - E\|_F^2 + \lambda \|X\|_1 + \mu \|E\|_{2,1} \right)$$

Data-fidelity for normal data

Sparsity

Group sparsity regularization, only a few columns can be nonzero

.. and identifying as anomalies the signals corresponding to columns of E that are nonzero.

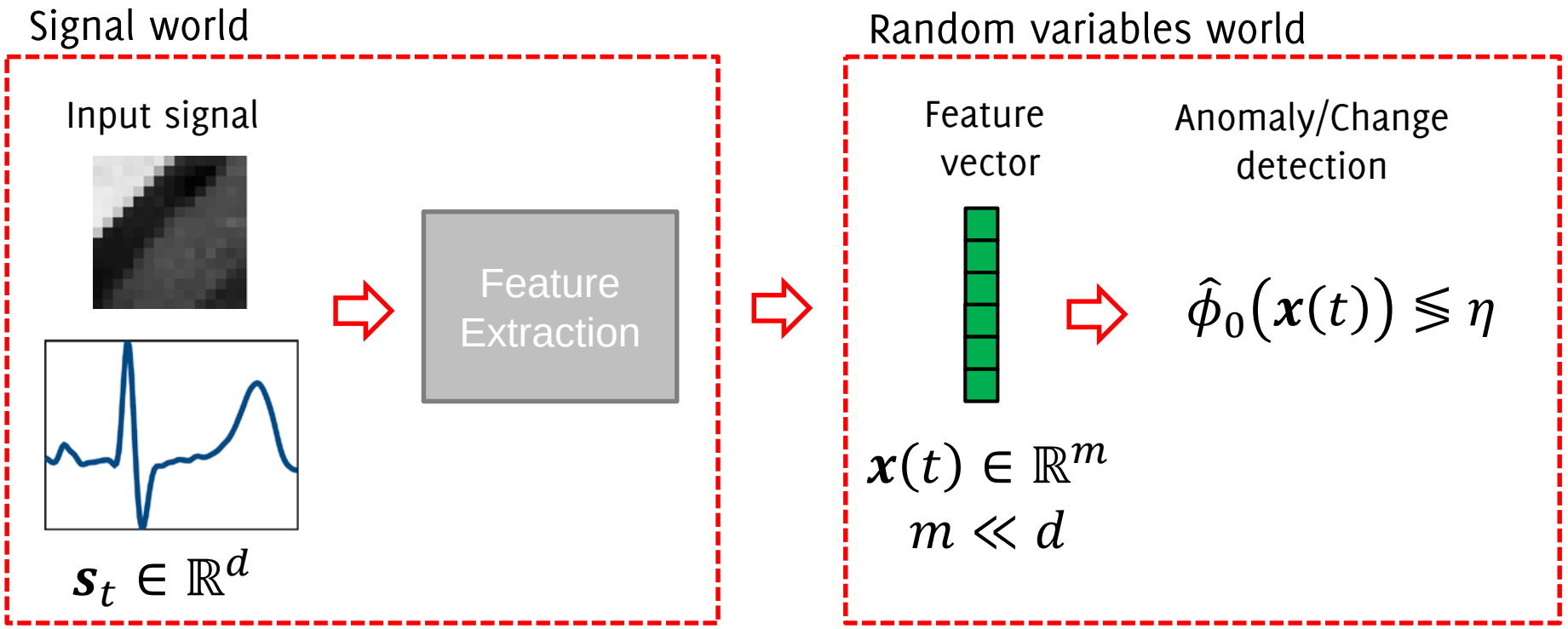
A. Adler, M. Elad, Y. Hel-Or, and E. Rivlin, "*Sparse coding with anomaly detection*" Journal of Signal Processing Systems, vol. 79, no. 2, pp. 179–188, 2015.

OUTLINE ON SEMI-SUPERVISED APPROACHES

- Detrending/Filtering for time-series
- Reconstruction-based methods
    - Subspace methods
- Feature-based monitoring
    - Expert-driven Features
    - Data-driven Features
    - Extended models

TYPICAL APPROACH: MONITORING FEATURES

Feature extraction: meaningful indicators to be monitored which have a known / controlled response w.r.t. normal data



Feature Extraction: signal processing, a priori information, learning methods

The customary statistical framework for anomaly detection

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

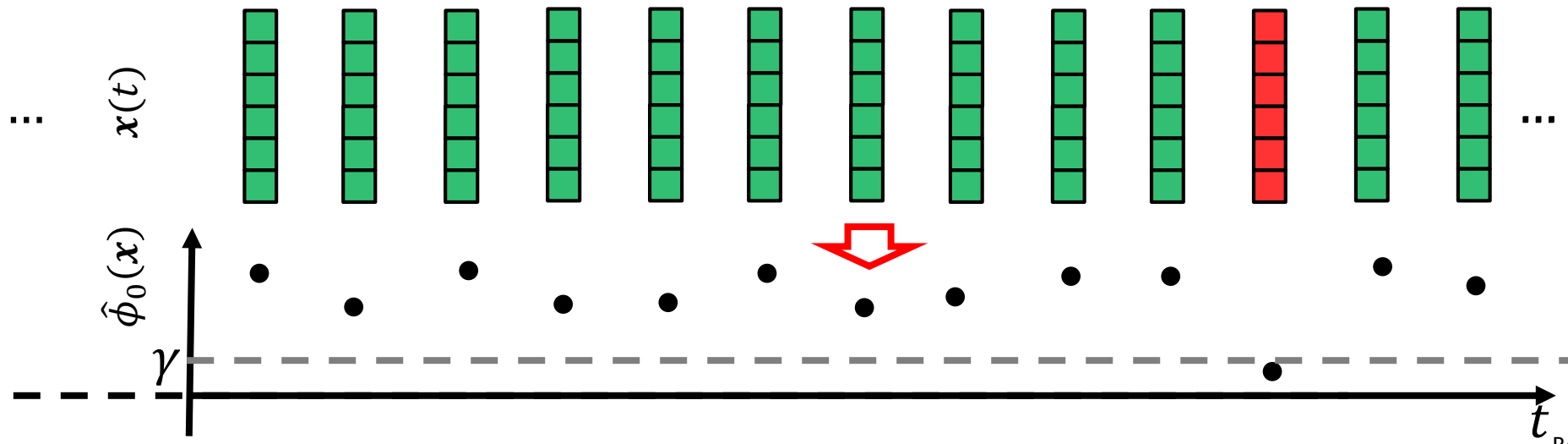
MONITORING FEATURE DISTRIBUTION

Normal data are expected **to yield features** $\boldsymbol{x}$ that are **i.i.d.** and **follow an** unknown **distribution** ϕ_0.

Anomalous data do not, as they follow $\phi_1 \neq \phi_0$.

We are **back to our statistical framework** and we can

- **learn** $\hat{\phi}_0$ from a set features extracted from normal data
- detect anomalous data by **extracting features** $\boldsymbol{x}$ associated to each input s, and then **testing whether** $\hat{\phi}_0(x) < \gamma$



Boracchi, Carrera, ICPR 2020

MONITORING FEATURE DISTRIBUTION

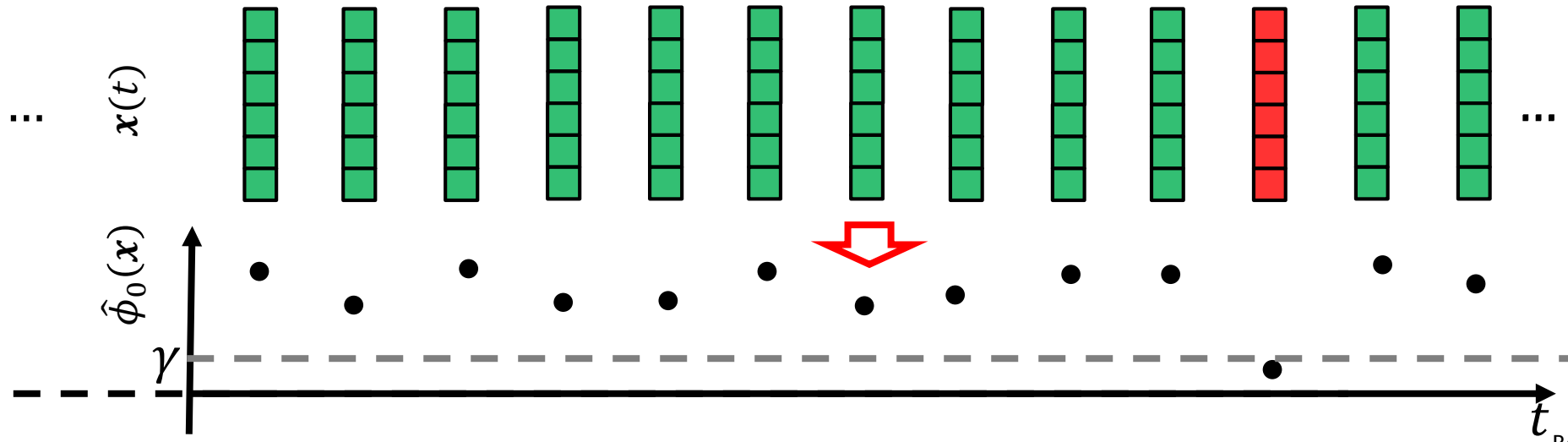
Normal data are expected **to yield features** $\boldsymbol{x}$ that are **i.i.d.** and **follow an** unknown **distribution** ϕ_0.

Anomalous data do not, as they follow $\phi_1 \neq \phi_0$.

We are **back to our statistical framework** and we can

- **learn** $\hat{\phi}$ from a set features extracted from normal data
- detect [illegible] out s, and then **testing whether** $\phi_0(x) < \gamma$

Or by adopting **any other statistical tool** to detect anomalies in $\boldsymbol{x}$



Boracchi, Carrera, ICPR 2020

FEATURE EXTRACTION

Data dimensionality can be reduced by extracting features

Good features should:

- Yield a **stable response** w.r.t. **normal data**
- Yield **unusual response** on **anomalies**

Examples of features seen so far:

- Reconstruction error $\mathrm{err}(\boldsymbol{s})$
- representation coefficients $(P^T\boldsymbol{s}, \boldsymbol{\alpha}, \dots)$

… but these are not the only

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

FEATURE EXTRACTION APPROACHES

There are two major approaches for extracting features:

Expert-driven (hand-crafted) features: computational expressions that are **manually designed by experts** to distinguish between normal and anomalous data

Data-driven features: **features** characterizing normal data are automatically **learned from training set of normal samples** S

OUTLINE ON SEMI-SUPERVISED APPROACHES

- Detrending/Filtering for time-series
- Reconstruction-based methods
    - Subspace methods
- Feature-based monitoring
    - Expert-driven Features
    - Data-driven Features
    - Data-driven Features: extended models

EXAMPLES OF EXPERT-DRIVEN FEATURES

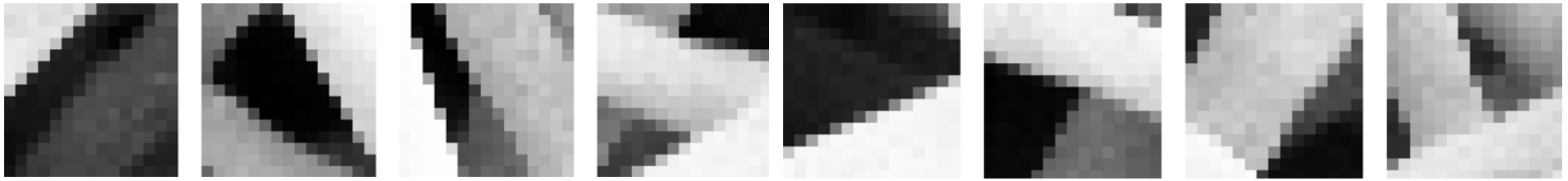
Expert-driven features: each patch of an image s

$$\mathbf{s}_c = \{s(c+u), u \in \mathcal{U}\}$$

Example of features are:

- the average,
- the variance,
- the total variation (the energy of gradients)

These can hopefully **distinguish normal** and **anomalous** patches, considering also how **anomalous regions will be** (e.g. flat or without edges)



D. Carrera, F. Manganini, G. Boracchi, E. Lanzarone *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

OUTLINE ON SEMI-SUPERVISED APPROACHES

- Detrending/Filtering for time-series
- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features
  - Data-driven Features: extended models

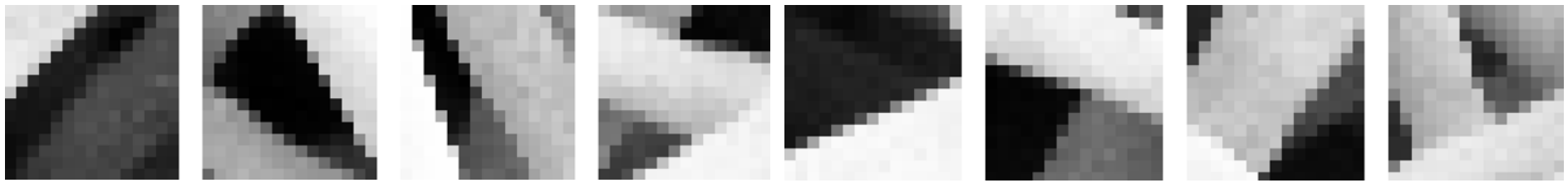
EXAMPLES OF DATA-DRIVEN FEATURES

Analyze each patch of an image s

$$\mathbf{s}_c = \{s(c+u), u \in \mathcal{U}\}$$

and determine whether it is normal or anomalous.

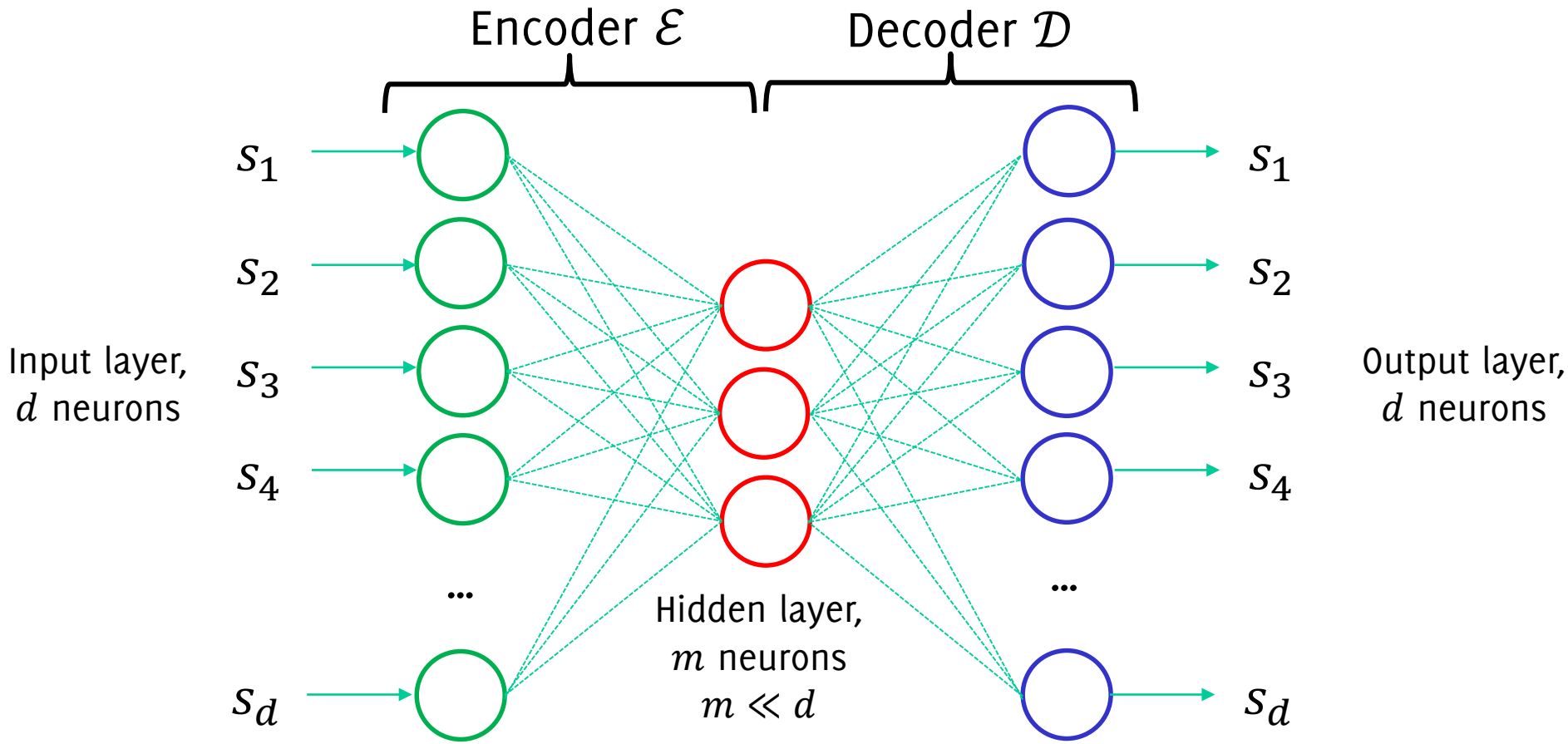
Data driven features: expressions to **quantitatively assess whether test patches conform** or not **with the model**, learned from normal data.



D. Carrera, F. Manganini, G. Boracchi, E. Lanzarone *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

AUTOENCODERS AS FEATURE EXTRACTORS

Autoencoders can be also used in feature-based monitoring schemes, monitoring as **feature** the hidden/**latent representation** of the input signal



Boracchi, Carrera, ICPR 2020

ANOMALY DETECTION BY MONITORING FEATURE DISTRIBUTION

Detection by **feature monitoring** (AE notation)

Training (Monitoring Feature Distribution):

- Learn the autoencoder $\mathcal{D}(\mathcal{E}(\cdot))$ from the training set S
- Fit a density model $\hat{\phi}_0$ to the encoded features

$$\{\mathcal{E}(\boldsymbol{s}), \boldsymbol{s} \in V\}$$

  over a validation set V, such that $V \cap S = \emptyset$

- Define a suitable threshold γ for $\hat{\phi}_0(\boldsymbol{s})$

Testing (Monitoring Feature Distribution):

- Encode each incoming signal $\boldsymbol{s}$ through $\mathcal{E}$
- Detect anomalies when the anomaly score $\mathcal{A}(\boldsymbol{s}) = \hat{\phi}_0(\mathcal{E}(\boldsymbol{s})) < \gamma$

Boracchi, Carrera, ICPR 2020

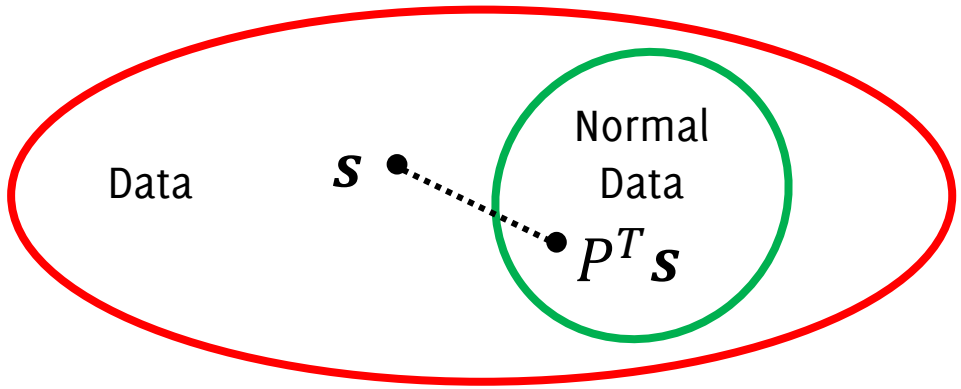
ANOMALY DETECTION BY MONITORING PCA PROJECTIONS

Compute the **projection on the subspace**,

$$\boldsymbol{s}' = P^T \boldsymbol{s}, \qquad P \in \mathbb{R}^{d \times m}, \qquad m \ll d$$

which is the projection over the first m principal components and a way to reduce data-dimensionality.

Monitor the projections $P^T \boldsymbol{s}$ by a suitable statistical technique (e.g. density based), as when monitoring $\mathcal{E}(\boldsymbol{s})$



Boracchi, Carrera, ICPR 2020

SPARSE REPRESENTATIONS AS FEATURE EXTRACTORS

To assess the conformance of $\boldsymbol{s}_c$ with D we solve the following

Sparse coding:

$$\boldsymbol{\alpha} = \underset{\boldsymbol{a}\in\mathbb{R}^n}{\operatorname{argmin}} \|D\boldsymbol{a} - \mathbf{s}\|_2^2 + \lambda\|\boldsymbol{a}\|_1, \qquad \lambda > 0$$

which is the BPDN formulation and we solve using ADMM.

The penalized ℓ^1 formulation has more degrees of freedom in the reconstruction, **the conformance of $\boldsymbol{s}$ with $\boldsymbol{D}$ have to be assessed monitoring both terms of the functional**

S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein. *"Distributed optimization and statistical learning via the alternating direction method of multipliers"* 2011

FEATURES EXTRACTED FROM SPARSE CODING

Features then include both the **reconstruction error**

$$\mathrm{err}(\boldsymbol{s}) = \|D\boldsymbol{\alpha} - \mathbf{s}\|_2^2$$

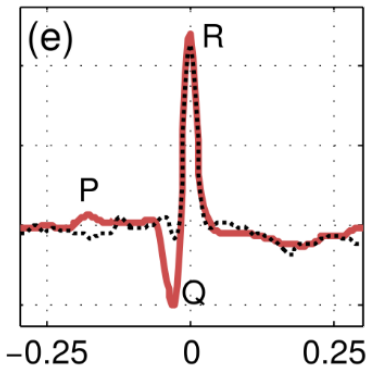
and **the sparsity** of the representation

$$\|\boldsymbol{\alpha}\|_1$$

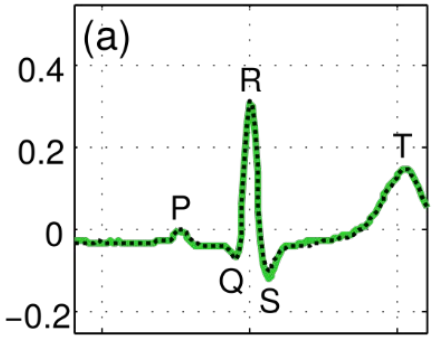
Thus obtaining **a data-driven feature vector** $\boldsymbol{x} = \begin{bmatrix} \|D\boldsymbol{\alpha} - \mathbf{s}\|_2^2 \\ \|\boldsymbol{\alpha}\|_1 \end{bmatrix}$

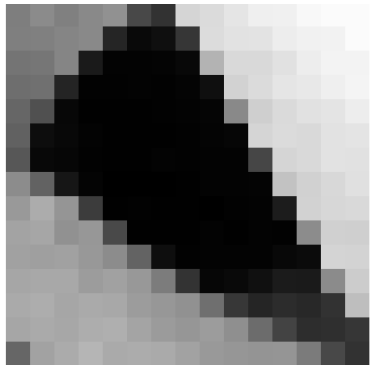
DENSITY-BASED MONITORING

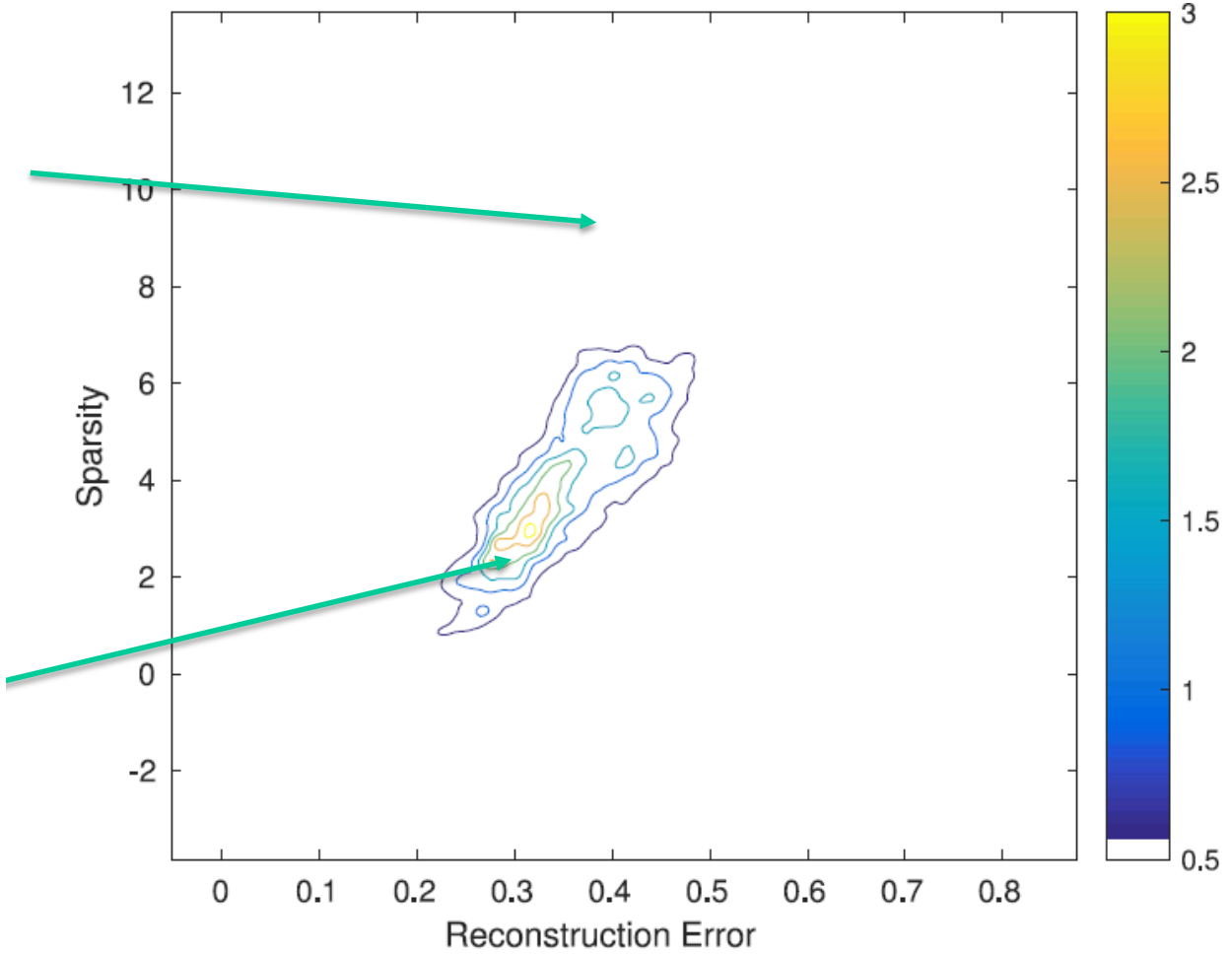
Anomalies



Normal data







Boracchi, Carrera, ICPR 2020

FEATURES EXTRACTED FROM SPARSE CODING

Training:

- **Learn** from S the dictionary D
- **Compute** the sparse representation w.r.t. D, thus features $\boldsymbol{x}$ over the validation set V, such that $V \cap S = \emptyset$
- **Learn** from V, the distribution $\hat{\phi}_0$ of normal features vectors $\boldsymbol{x}$ and the threshold γ.

The model for anomaly detection is $(D, \hat{\phi}_0, \gamma)$

Testing:

- Perform sparse coding of a test signal $\boldsymbol{s}$, thus get the feature vector $\boldsymbol{x}$
- Detect anomalies when $\mathcal{A}(\boldsymbol{s}) = \hat{\phi}_0(\boldsymbol{x}) < \gamma$

D. Carrera, F. Manganini, G. Boracchi, E. Lanzarone *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

FEATURES EXTRACTED FROM SPARSE CODING

Training:

- **Learn** from S the dictionary D
- **Compute** the sparse representation w.r.t. D, thus features $\boldsymbol{x}$ over the validation set V, such that $V \cap S = \emptyset$
- **Learn** from V, the distribution $\hat{L}$ of normal features vectors x and the threshold γ.

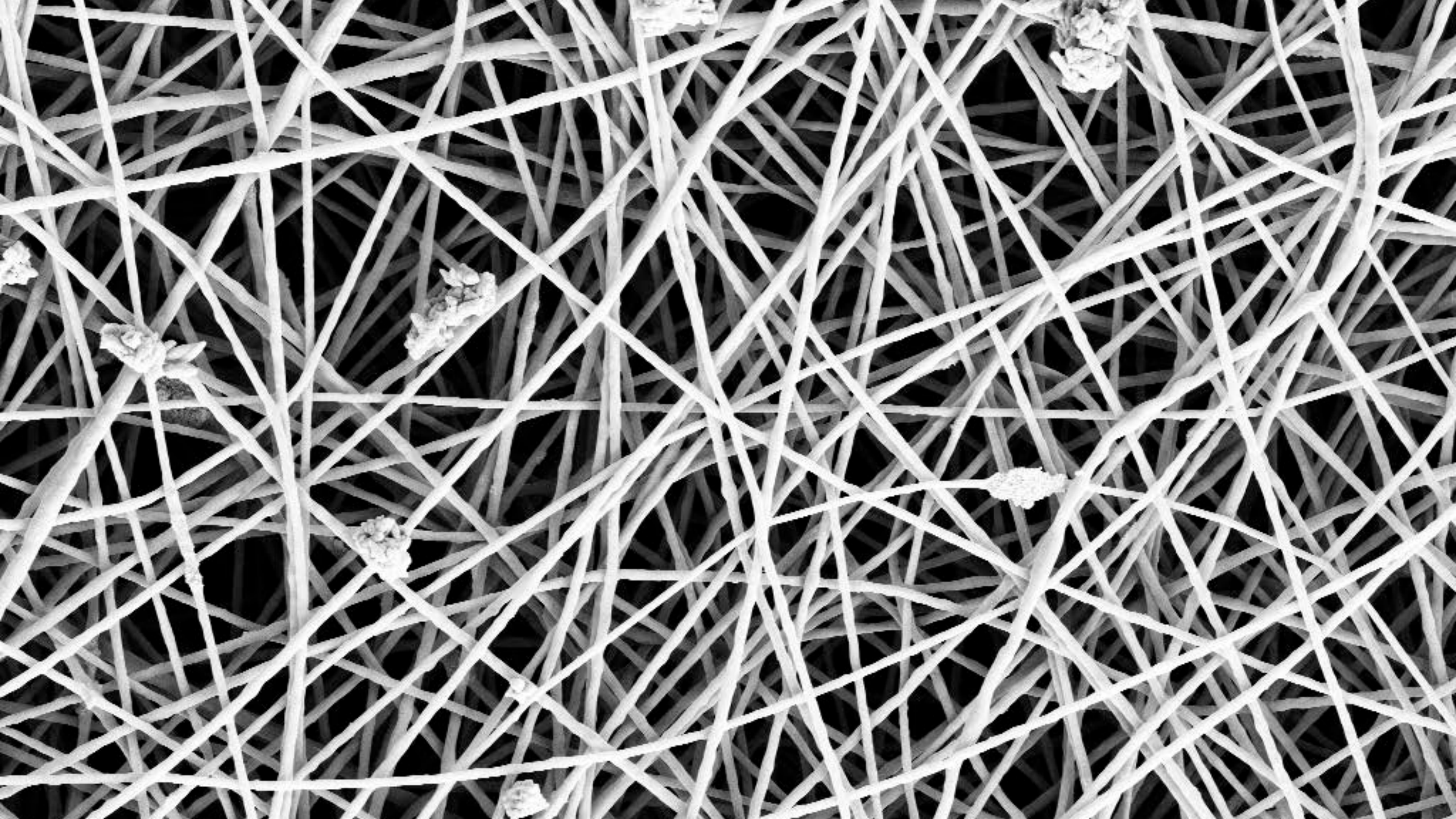
The mode

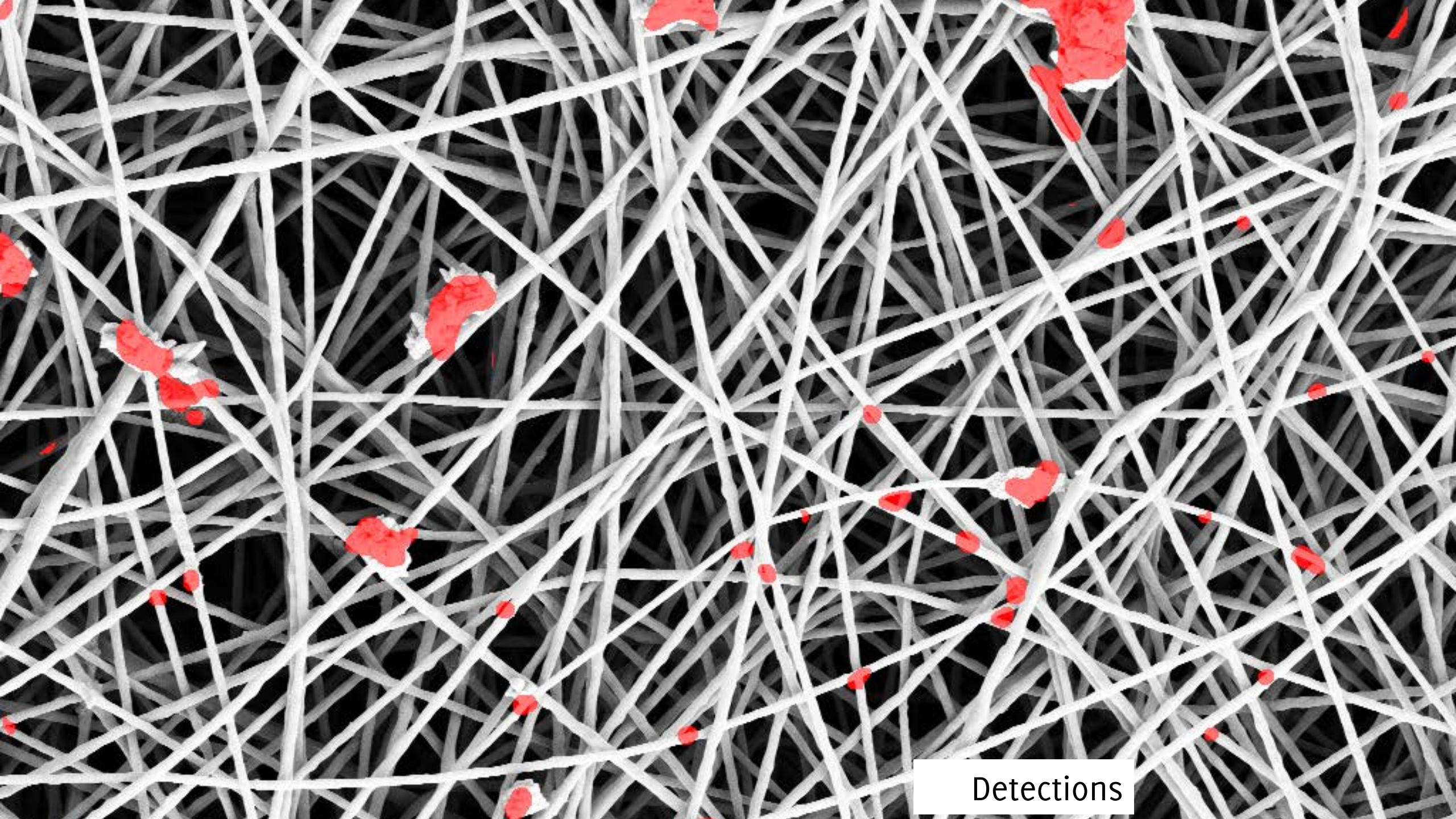
Testing:

- Perfor
- Detect

This is rather a flexible solution and can be adapted when operating conditions changes (e.g. heartrate changes, images are acquired at different zooming level)

D. Carrera, F. Manganini, G. Boracchi, E. Lanzarone *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

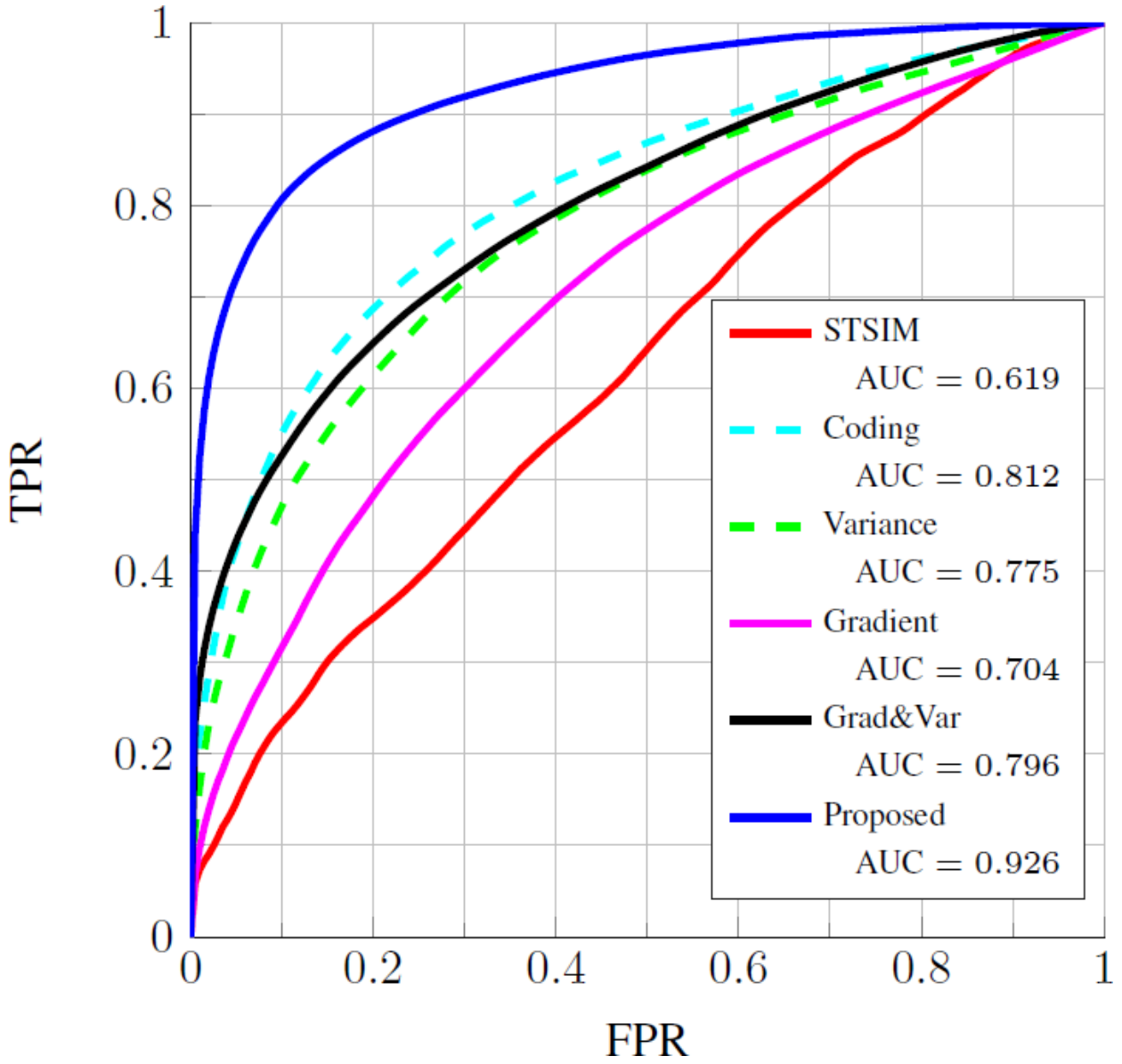



Detections

THE ROC CURVES

Tests on 40 images with anomalies manually annotated by an expert

The proposed anomaly detection algorithm outperforms expert-driven features and other methods based on sparse representations



DETECTABILITY LOSS ON THESE NANOFIBERS

Selecting the good features is obviously important.

Why not stacking data-driven and expert-driven features?

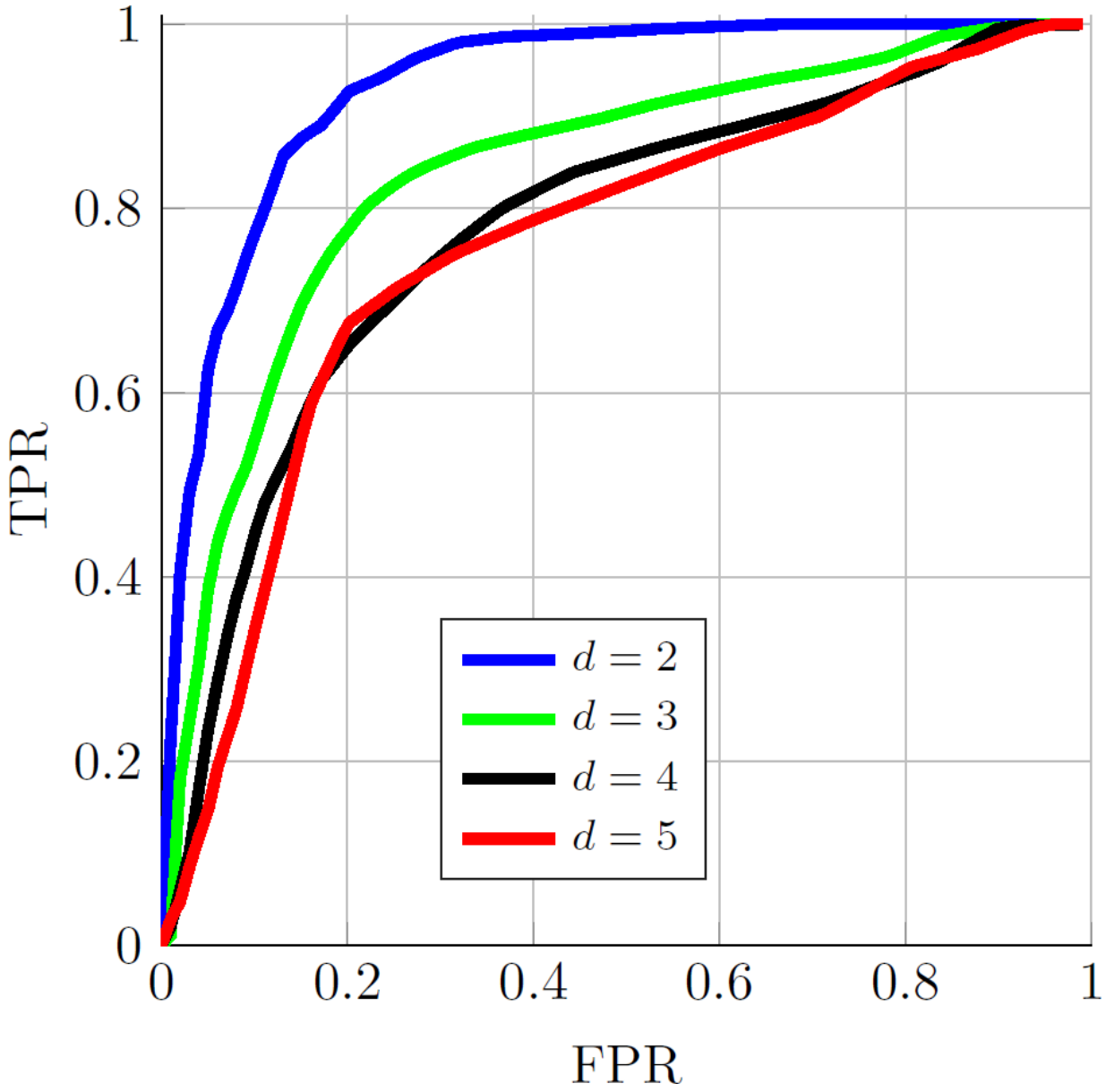
Consider $d = 3, 4, 5$ dimensional features

- We selectively add the three expert-driven features to the two data-driven ones (average gradient and variance)
- We always fit a GM model to a large-enough number of training data

Boracchi, Carrera, ICPR 2020

DETECTABILITY LOSS ON THESE NANOFIBERS

Anomaly detection performance **progressively decay when d increases**



A different test set than TII paper was here used

Boracchi, Carrera, ICPR 2020

DETECTABILITY LOSS AND IRRELEVANT FEATURES

We believe this because **added features are irrelevant**, namely features that:

- are not directly affected by the change
- do not provide any additional information for change detection purposes (i.e. leave $\mathrm{sKL}(\phi_0, \phi_1)$ constant)

Adding irrelevant feature yields detectability loss.

Other issues might affect detection performance

- A biased denisty function for $\hat{\phi}_0$
- Scarcity of training samples when d increases

However, we are inclined to conclude that

- In this case these expert-driven features do not add enough relevant information on top of the data-driven ones (for anomaly-detection purposes).

OUTLINE ON SEMI-SUPERVISED APPROACHES

- Detrending/Filtering for time-series
- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features
  - Data-driven Features: extended models

CONVOLUTIONAL SPARSITY

Data-driven features from **convolutional sparse models**

$$\boldsymbol{s} \approx \sum_{i=1}^{m} \boldsymbol{d_i} \circledast \boldsymbol{\alpha_i}\,, \qquad \text{s.t. } \boldsymbol{\alpha_i} \text{ is sparse}$$

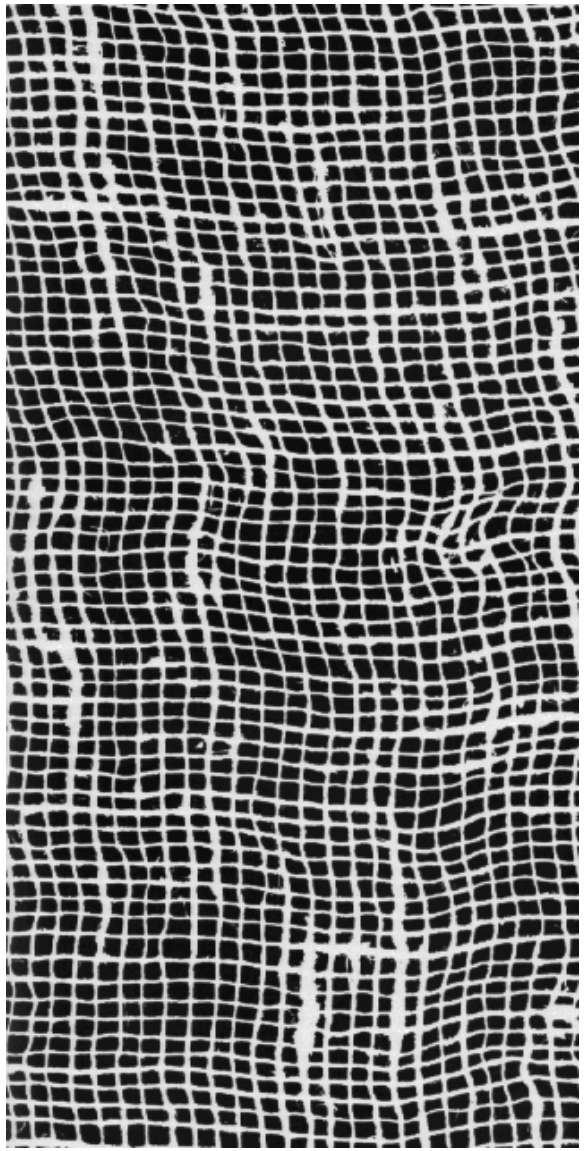
where **the whole image $\boldsymbol{s}$** is **entirely encoded** as the sum of n convolutions between a filter $\boldsymbol{d_i}$ and a coefficient map $\boldsymbol{\alpha_i}$

- Translation invariant representation
- Few small filters are typically required
- Filters exhibit very specific image structures
- Easy to use filters having different size

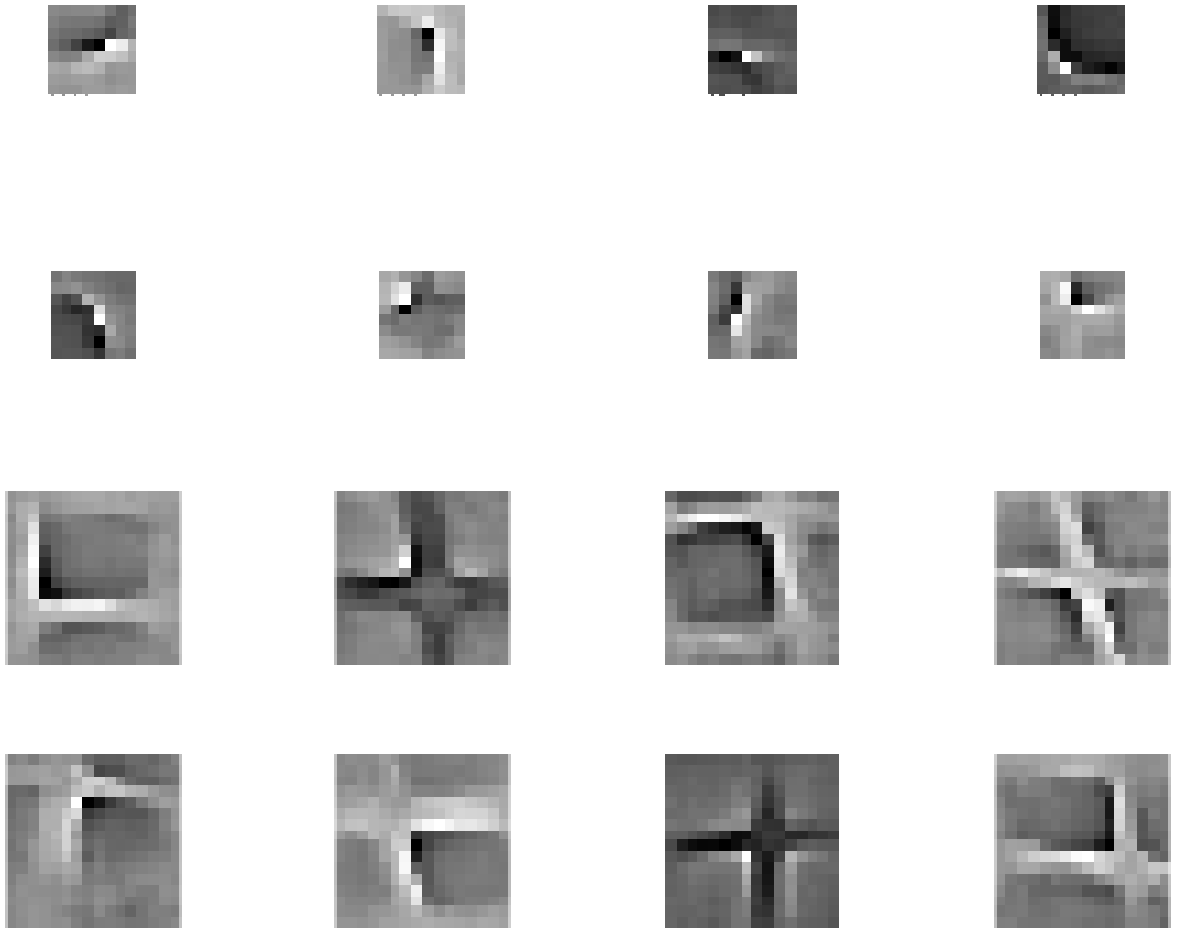
M. D. Zeiler, D. Krishnan, G. W. Taylor and R. Fergus *"Deconvolutional networks"* CVPR 2010

EXAMPLE OF LEARNED FILTERS

Training Image



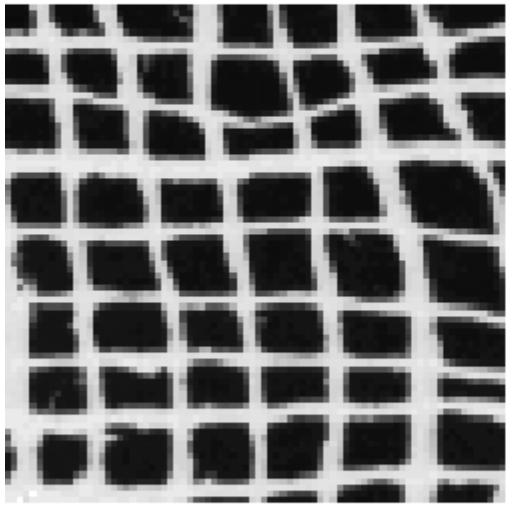
Learned Filters



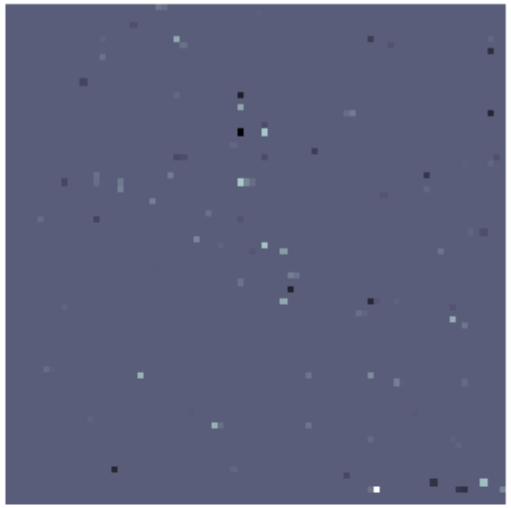
Boracchi, Carrera, ICPR 2020

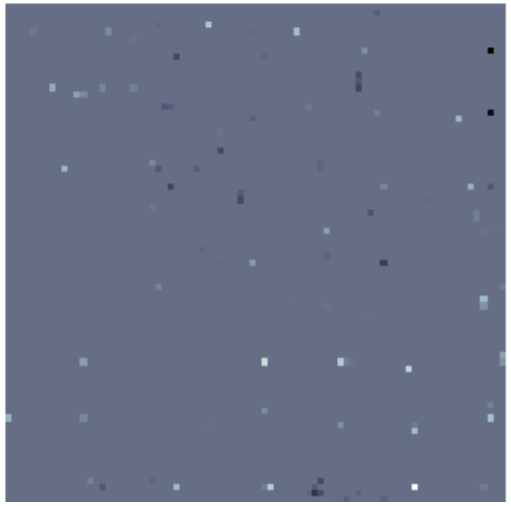
EXAMPLE OF FEATURE MAPS

Test Image

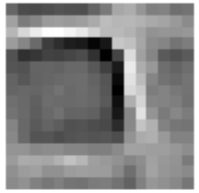


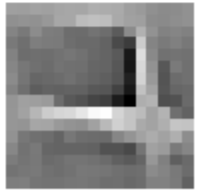
Coefficient maps





Filters





Boracchi, Carrera, ICPR 2020

FEATURES FROM CONVOLUTIONAL SPARSE MODEL

The standard convolutional sparse coding

$$\{\hat{\boldsymbol{\alpha}}\} = \underset{\{\boldsymbol{\alpha}\}_n}{\operatorname{argmin}} \left\| \sum_{i=1}^{n} \boldsymbol{d_i} \circledast \boldsymbol{\alpha_i} - \mathbf{s} \right\|_2^2 + \lambda \sum_{i=1}^{n} \|\boldsymbol{\alpha}\|_1$$

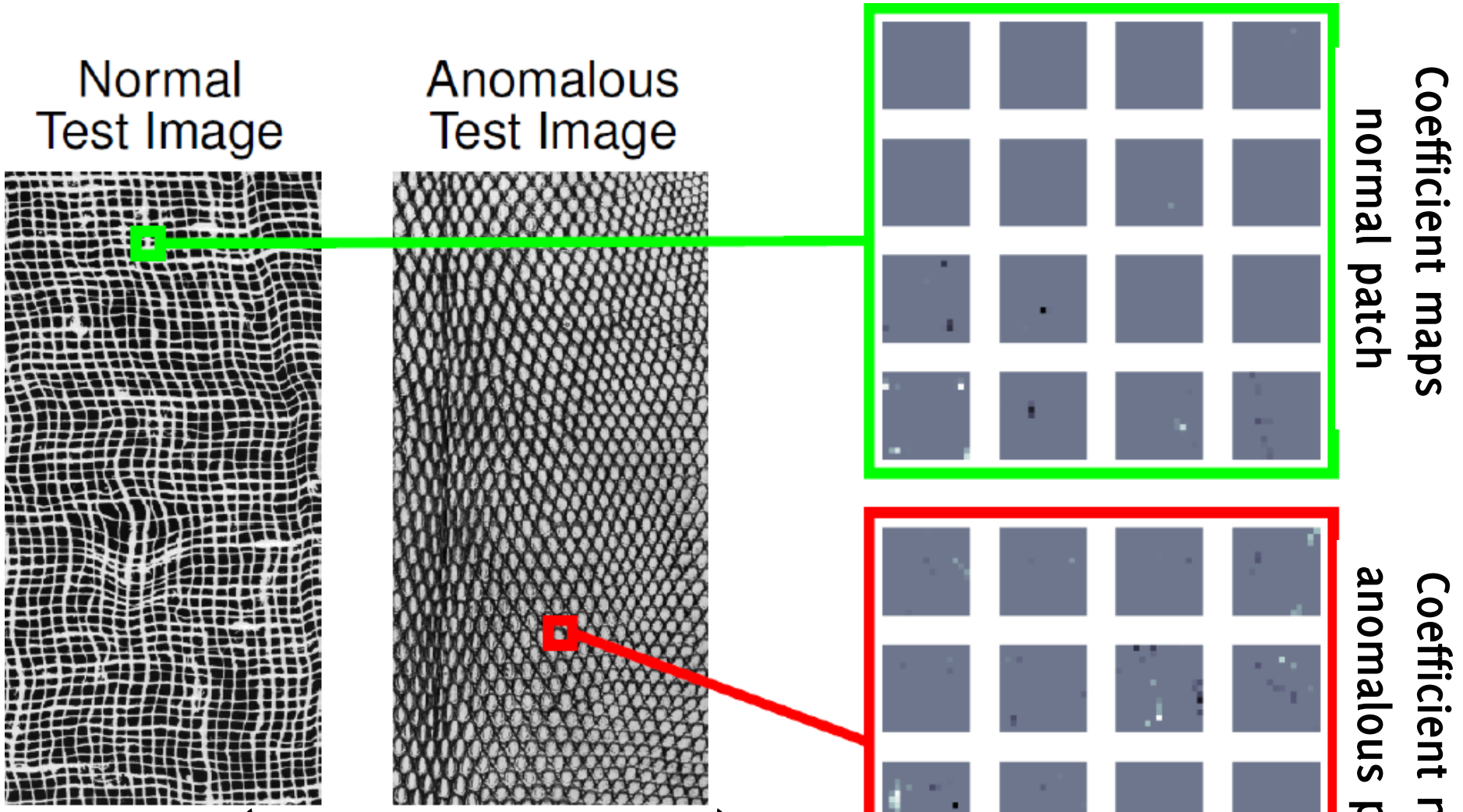
leads to the following feature vector for each image region in c:

$$\boldsymbol{x}_c = \begin{bmatrix} \left\| \prod_{\boldsymbol{c}} \left(\sum_{i=1}^{n} \boldsymbol{d_i} \circledast \hat{\boldsymbol{\alpha}}_i - \mathbf{s} \right) \right\|_2^2 \\ \sum_{i=1}^{n} \left\| \prod_{\boldsymbol{c}} \hat{\boldsymbol{\alpha}} \right\|_1 \end{bmatrix}$$

…but, anomaly detection performance are rather poor

Boracchi, Carrera, ICPR 2020

SPARSITY IS TOO LOOSE A CRITERION FOR DETECTION



These two (normal and anomalous) patches exhibit same sparsity and reconstruction error

Boracchi, Carrera, ICPR 2020

FEATURES FROM CONVOLUTIONAL SPARSE MODEL

Add the **group sparsity** of the maps on the patch support as an **additional feature**

$$x_c = \begin{bmatrix} \left\| \prod_{\boldsymbol{c}} \left(\sum_{i=1}^{m} \boldsymbol{d_i} \circledast \hat{\boldsymbol{\alpha}}_{\boldsymbol{i}} - \mathbf{s} \right) \right\|_{\mathbf{2}}^{\mathbf{2}} \\ \sum_{i=1}^{m} \left\| \prod_{\boldsymbol{c}} \hat{\boldsymbol{\alpha}} \right\|_1 \\ \sum_{i=1}^{m} \left\| \prod_{\boldsymbol{c}} \hat{\boldsymbol{\alpha}} \right\|_2 \end{bmatrix}$$

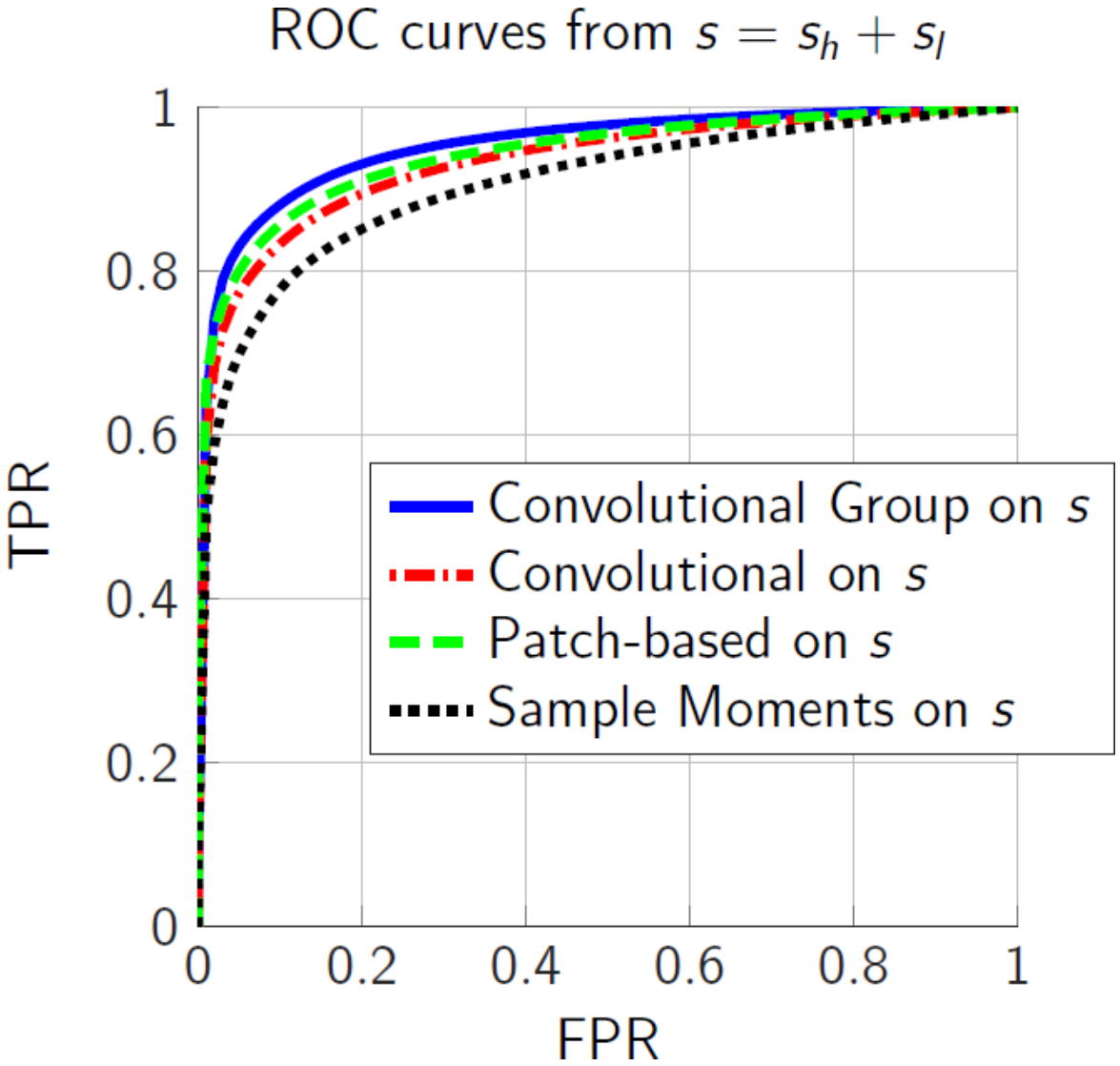
D. Carrera, G. Boracchi, A. Foi and B. Wohlberg , *"Detecting Anomalous Structures by Convolutional Sparse Models "* IEEE IJCNN 2015

ANOMALY-DETECTION PERFORMANCE

On 25 different textures and 600 test images (pair of textures to mimic normal/anomalous regions)

Best performance achieved by the 3-dimensional feature indicators

Achieve similar performance than steerable pyramid specifically designed for texture classification



Boracchi, Carrera, ICPR 2020

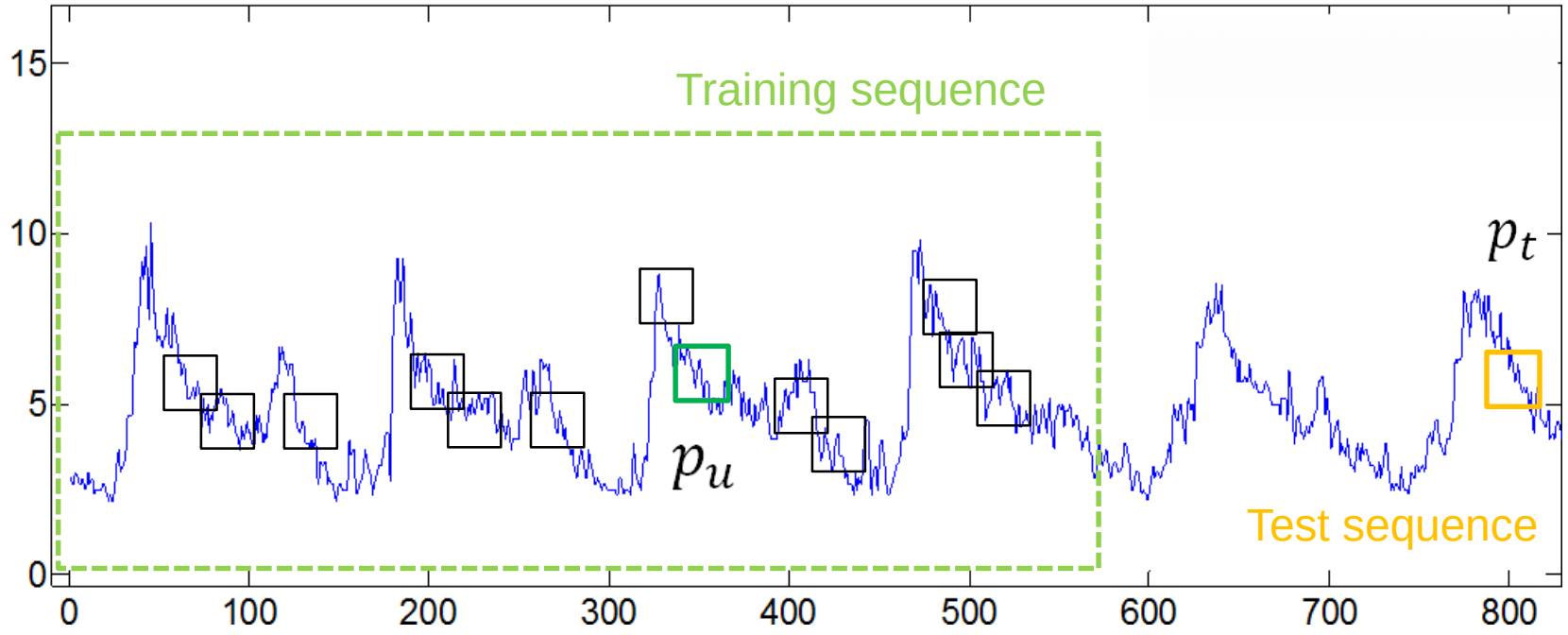
SIMILARITY AND REFERENCE-BASED APPROACHES

Used for monitoring time series and images

SIMILARITY-BASED METHODS

When **normal data exhibit a periodic behaviour**, anomalies are detected as **unusual patterns that are not similar to training ones.**

Euclidean distance between portions of training and test data



E. Keogh, J. Lin, A. Fu *"Hot sax: Efficiently finding the most unusual time series subsequence"* IEEE ICDM 2005 8 pp

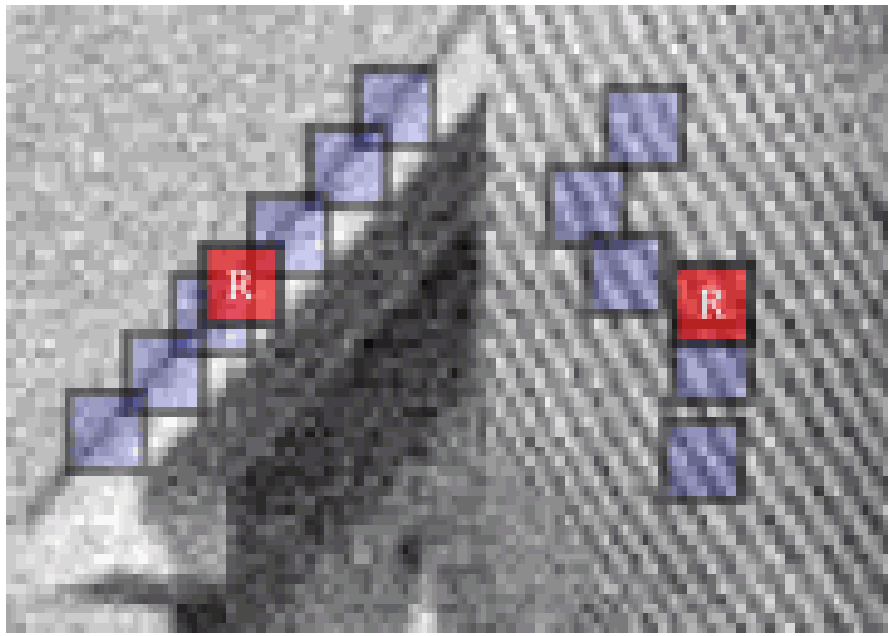
G. Boracchi and M. Roveri *"Exploiting Self-Similarity for Change Detection"* , IJCNN 2014, pp 3339 - 3346

SELF SIMILARITY IS A POWERFUL PRIOR

Texture completion

Denoising (Regression)

Inpainting (Reconstruction)



Images courtesy of Alessandro Foi http://www.cs.tut.fi/~foi/

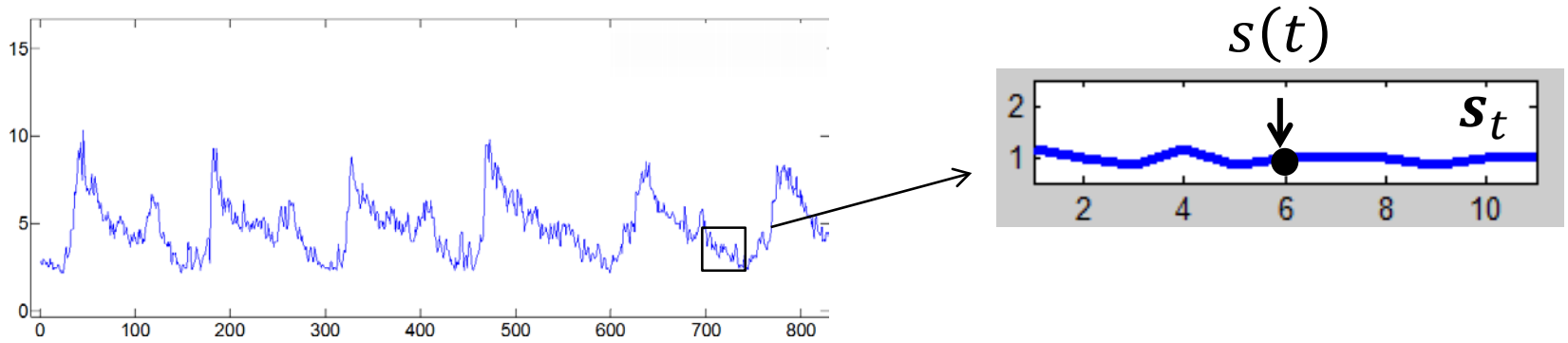
SELF SIMILARITY IS A POWERFUL PRIOR

Self-similarity is measured patch-wise

We consider 1D datastreams $\{s(\tau), \tau = 1, \dots\}, \; s(\tau) \in \mathbb{R}$

We define a patch centered at t having size ν as

$$\boldsymbol{s}_t = \{s(t-\nu), \dots, s(t), \dots, s(t+\nu)\}$$



Distance between patches is the ℓ_2 norm of their difference

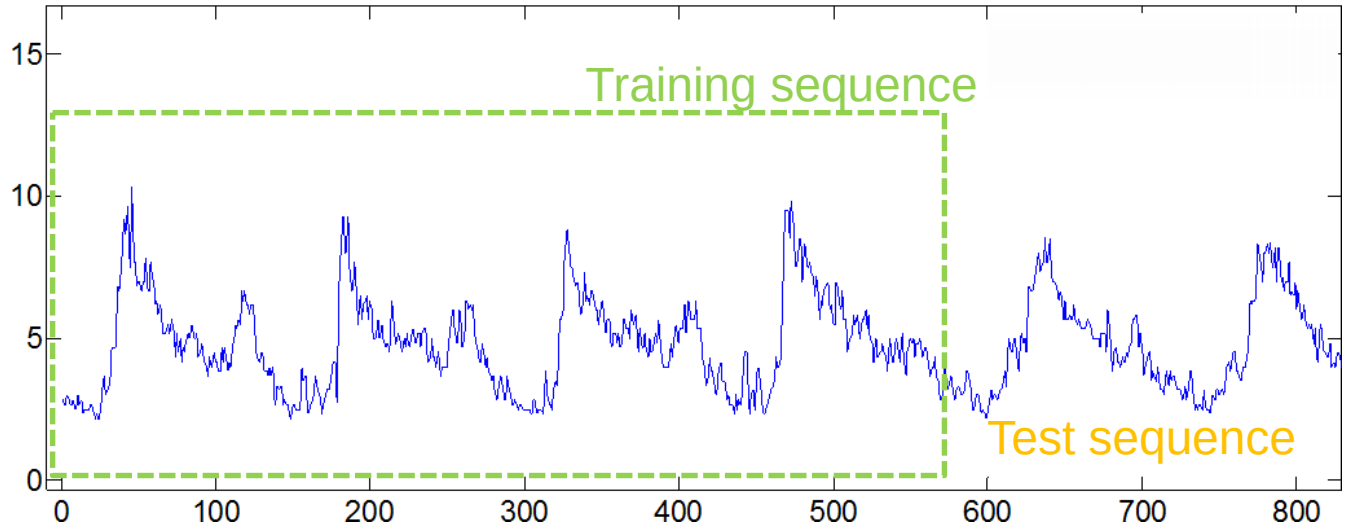
$$\|\boldsymbol{s}_t - \boldsymbol{s}_\tau\|_2 = \sqrt{\sum_{i=-\nu}^{\nu} \big(s(t+i) - s(\tau+i)\big)^2}$$

Boracchi, Carrera, ICPR 2020

MEASURING SIMILARITY FOR DETECTION PURPOSES

We build a training set for *normal patches*

$$\mathbf{P} = \{\boldsymbol{s}_t, t = \nu, \ldots, M - \nu\}$$

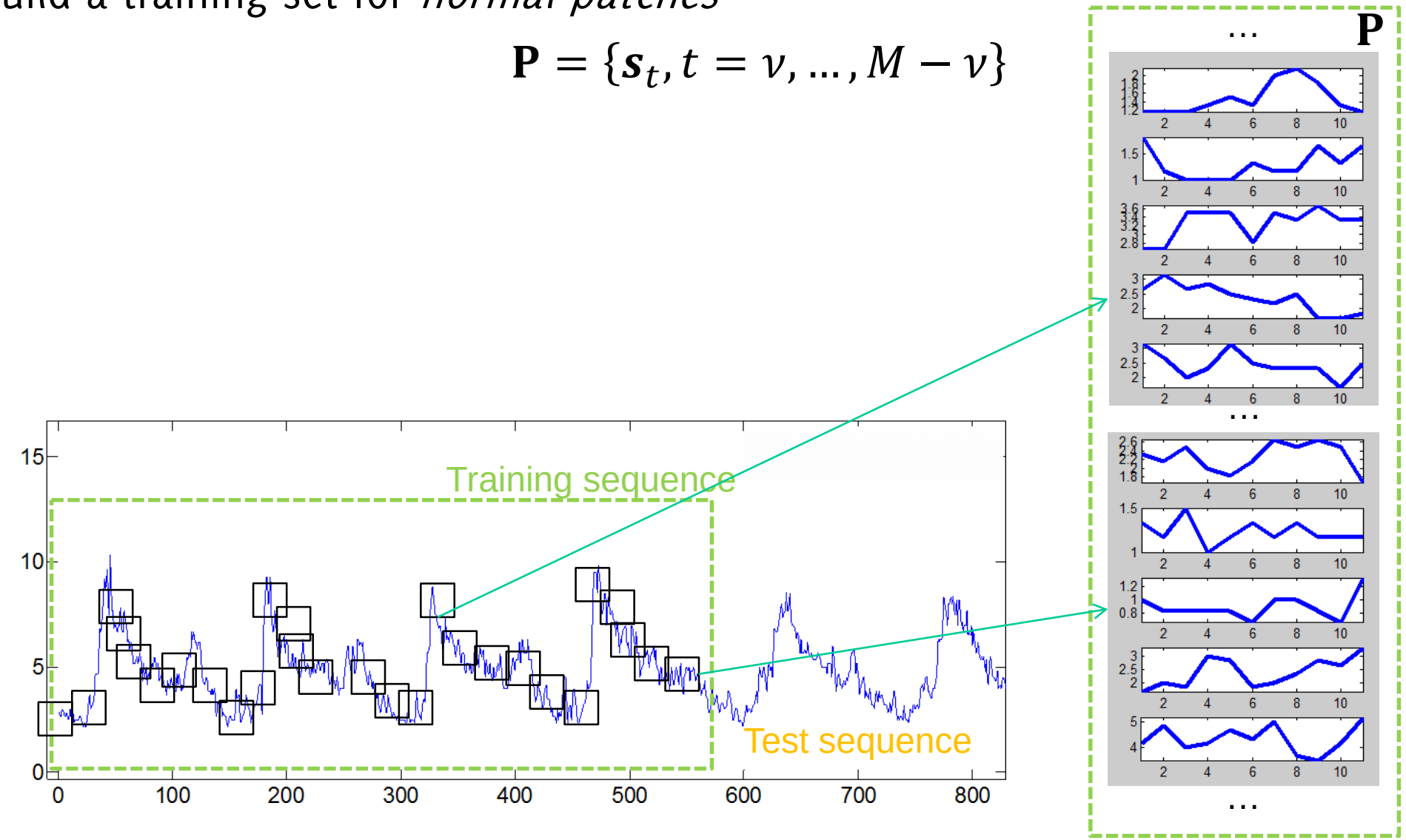


Boracchi, Carrera, ICPR 2020

MEASURING SIMILARITY FOR DETECTION PURPOSES

We build a training set for *normal patches*

$$\mathbf{P} = \{\boldsymbol{s}_t, t = \nu, \ldots, M - \nu\}$$



Boracchi, Carrera, ICPR 2020

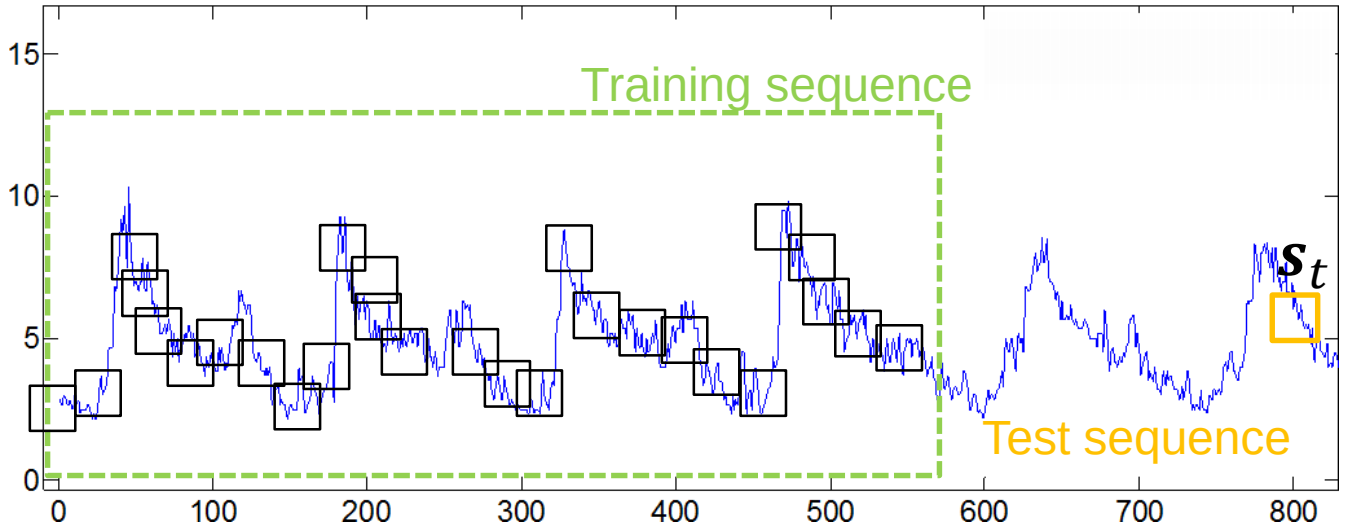
MEASURING SIMILARITY FOR DETECTION PURPOSES

We build a training set for *normal patches*

$$\mathbf{P} = \{\boldsymbol{s}_t, t = \nu, \dots, M - \nu\}$$

Intuition:

$$\begin{cases} \exists\ \boldsymbol{s}_u \in \mathbf{P} \text{ similar to } \boldsymbol{s}_t, \forall t < T^* & \text{Normal} \\ \nexists\ \boldsymbol{s}_u \in \mathbf{P} \text{ similar to } \boldsymbol{s}_t, \forall t \geq T^* & \text{Out of Control} \end{cases}$$



Boracchi, Carrera, ICPR 2020

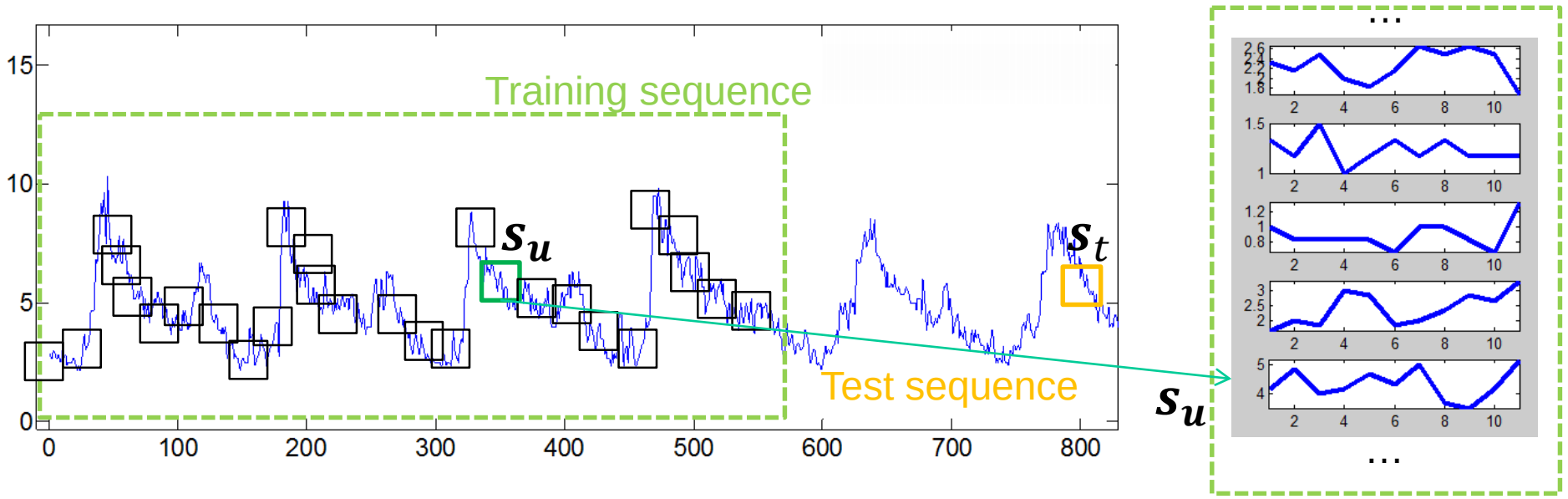
MEASURING SIMILARITY FOR DETECTION PURPOSES

We build a training set for *normal patches*

$$\mathbf{P} = \{\boldsymbol{s}_t, t = \nu, \dots, M - \nu\}$$

Intuition:

$$\begin{cases} \exists\ \boldsymbol{s}_u \in \mathbf{P} \text{ similar to } \boldsymbol{s}_t, \forall t < T^* & \text{Normal} \\ \nexists\ \boldsymbol{s}_u \in \mathbf{P} \text{ similar to } \boldsymbol{s}_t, \forall t \geq T^* & \text{Out of Control} \end{cases}$$



Boracchi, Carrera, ICPR 2020

THE ANOMALY SCORE / CHANGE INDICATOR AND DATA-DRIVEN FEATURES

A **feature** $x(t)$ to **quantitatively assess** similarity to training data

We expect $x(t)$ to satisfy

- $\{x(t), t < T^*\}$ should be **i.i.d.** realizations of an **unknown random variable**
- $\{x(t), t \geq T^*\}$ should come from a **different distribution**

Out of **control states** can be **detected** as **changes in the distribution of x**

- We can use any statistical process control technique

Boracchi, Carrera, ICPR 2020

FEATURE TO ASSESS SELF-SIMILARITY

The feature $\boldsymbol{x(t)}$ is computer after having identified the most similar patch to $\boldsymbol{s_t}$ in $\mathbf{P}$.

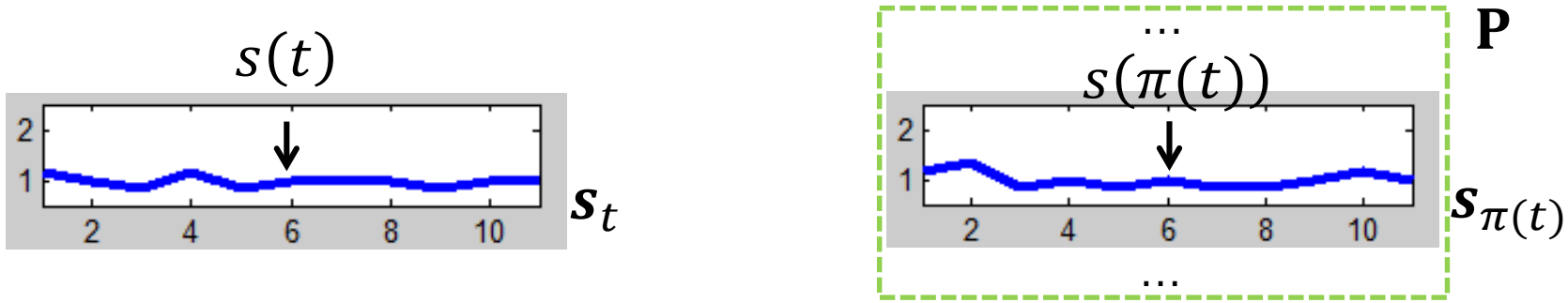
We define $\pi(\cdot)$ as the map that associate to t the location $\pi(t)$ of the patch $\mathbf{P}$ of that is most similar to $\boldsymbol{s_t}$

$$\pi(t) = \underset{\tau=\nu,\dots,M-\nu}{\operatorname{argmin}} ||\boldsymbol{s_t} - \boldsymbol{s_\tau}||_2$$

$\boldsymbol{x(t)}$ is the difference between the centers of $\boldsymbol{s_t}$ and $\boldsymbol{s_{\pi(t)}}$

$$x(t) = s(t) - s(\pi(t))$$

In ideal conditions $\boldsymbol{x(t)}$ should be i.i.d. noise



Thus **can be monitored by any Change Detection Test** (or anomaly detection method)

G. Boracchi and M. Roveri *"Exploiting Self-Similarity for Change Detection"* , IJCNN 2014, pp 3339 - 3346

DATA DRIVEN FEATURE TO ASSESS SIMILARITY

In the real life, perfect matches are rare

- Patches do not differ only because of noise
- Noise affects also the association function $\pi(\cdot)$

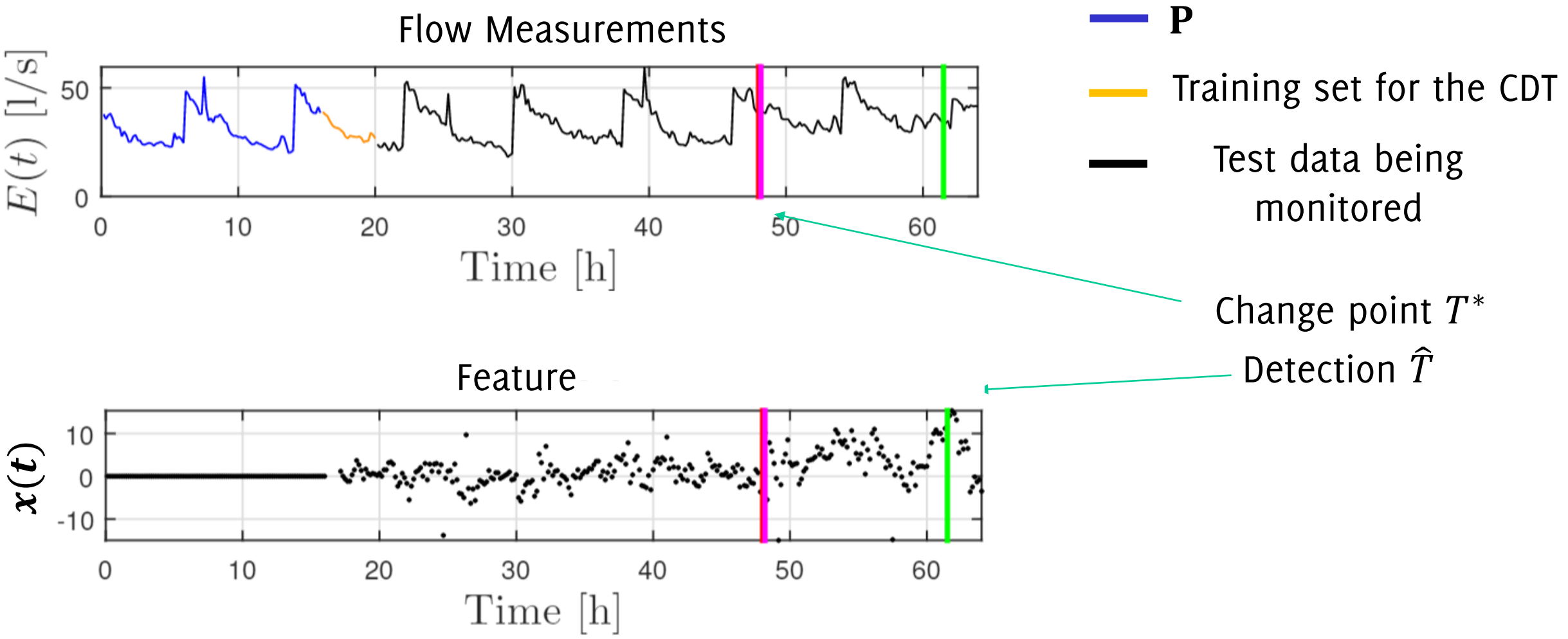
However, there is an **experimental evidence** that **patch similarity** well correlates with the **similarity between** their **central pixels**

- This is the idea behind *Non Local Means filter* [Buades et al 2005], which introduced a well established paradigm in signal/image processing

Therefore, as long as similarity assumption holds, it is possible to monitor the sequence of $x(t)$ to detect changes / anomalies

A. Buades, B. Coll, and J. Morel, "A review of image denoising algorithms, with a new one," Multiscale Modeling Simulation, vol. 4, no. 2, p. 490, 2005.

FEATURE TO ASSESS SELF-SIMILARITY



A. Soldevila, G. Boracchi, M. Roveri, S. Tornil-Sin and V. Puig, "Detecting and Localizing Leaks in Water Distribution Networks by Combining Expert Knowledge and Data-Driven Models", Under Revision

HOT SAX: FINDING DISCORDS

Discords are sequences that are *least* similar to all the others

Discords are located by analyzing the whole sequence and comparing test each patch with all the others

The most unusual patch, i.e. the one having the largest distance w.r.t. its closest neighbohoord, is reported as a discord

In such a comparison it is important to:

- Avoid self-matches (i.e. comparison between each patch and overlapping ones)
- Adopt some efficient search criteria instead of “brute-force” search the most similar match

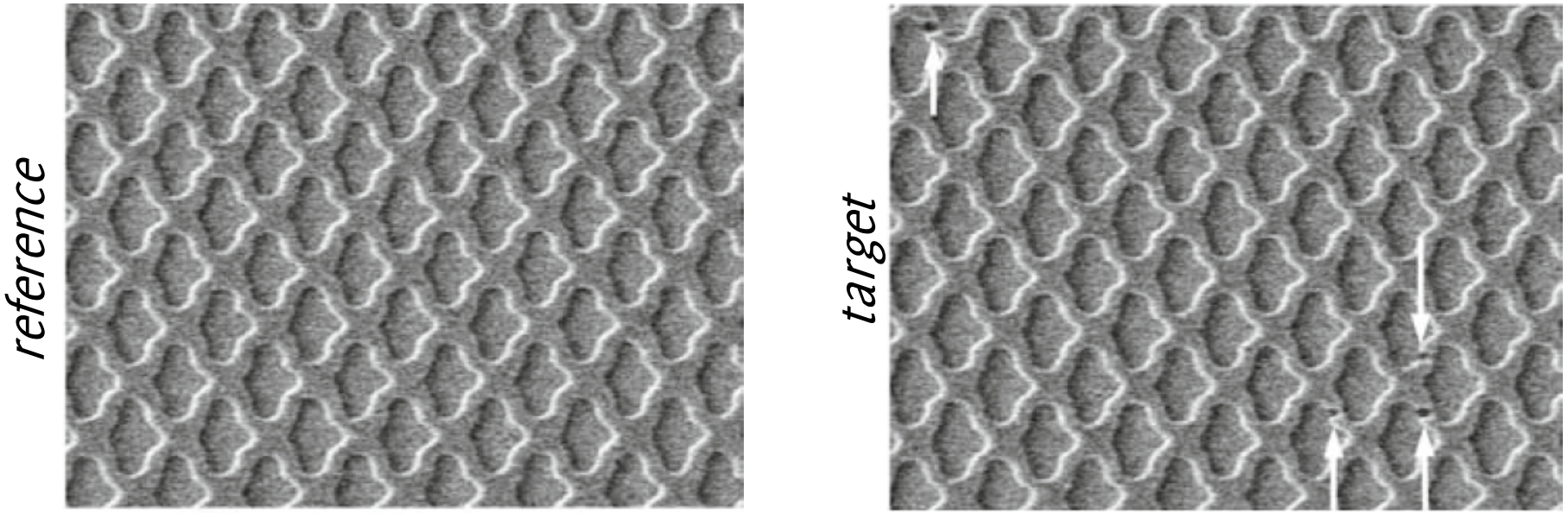
To this purpose, optimized search procedure are proposed in (Keogh et al 2005)

E. Keogh, J. Lin, A. Fu *"Hot sax: Efficiently finding the most unusual time series subsequence"* IEEE ICDM 2005 8 pp

REFERENCE BASED-METHODS IN QUALITY INSPECTION

In some cases anomalies can be detected by comparing

- the **target**, namely the image to be tested
- against a **reference**, namely an anomaly-free image



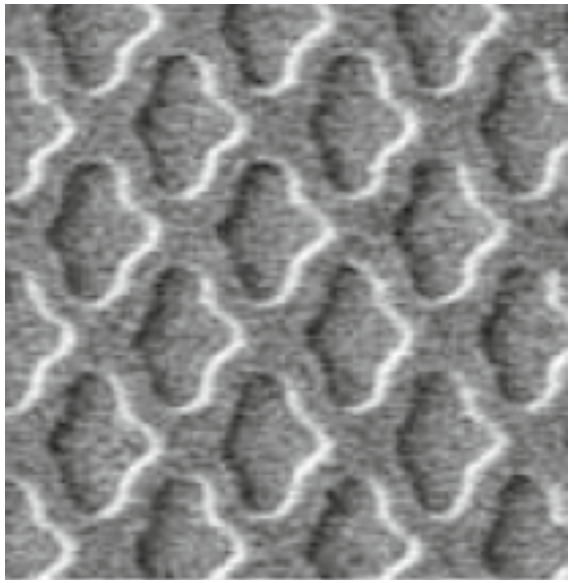
M. Zontak, I. Cohen: *Defect detection in patterned wafers using anisotropic kernels.*" Machine Vision and Applications 21(2), 129{141 (2010)

REFERENCE BASED-METHODS IN QUALITY INSPECTION

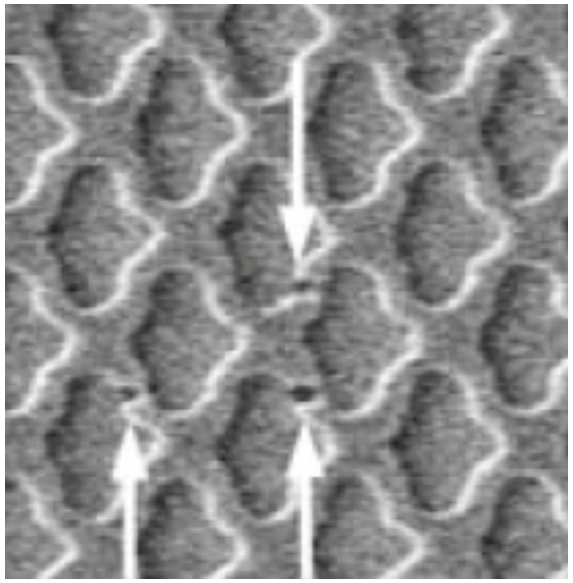
In some cases anomalies can be detected by comparing

- the **target**, namely the image to be tested
- against a **reference**, namely an anomaly-free image

reference



target

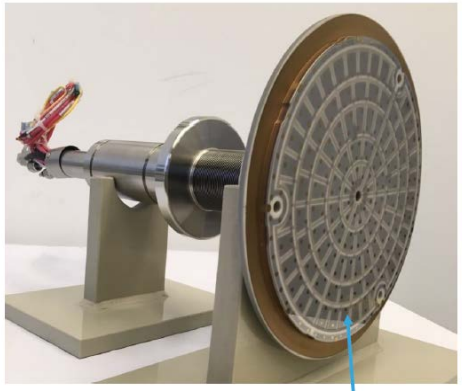


M. Zontak, I. Cohen: *Defect detection in patterned wafers using anisotropic kernels.*" Machine Vision and Applications 21(2), 129{141 (2010)

REFERENCE BASED-METHODS: E-CHUCKING

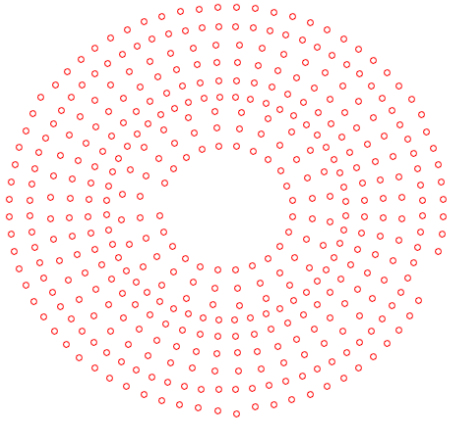
The e-chuck is adopted to safely maneuver, position and block the silicon wafer during production.

The device is configured by means of marking materials and the markers should conform to a reference template

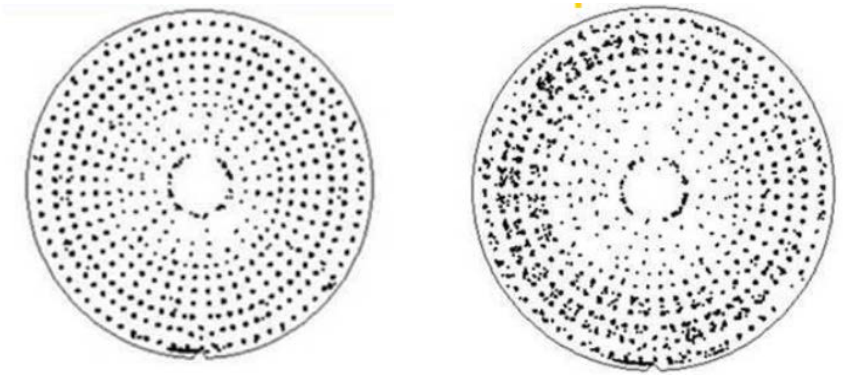


Electrodes

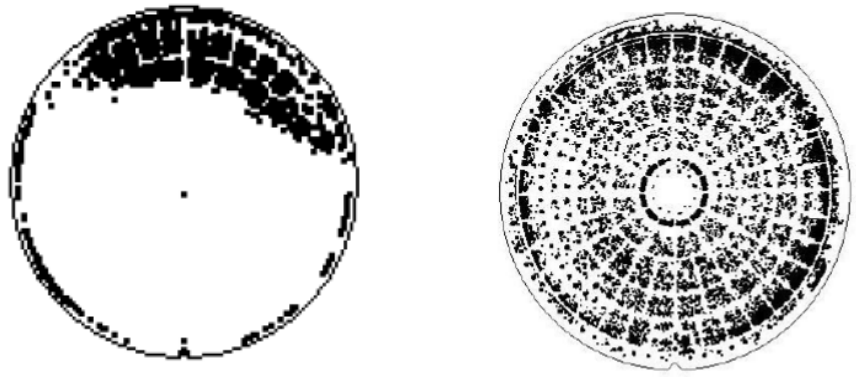
Template



Good (normal)



Bad (anomalous)



Boracchi, Carrera, ICPR 2020

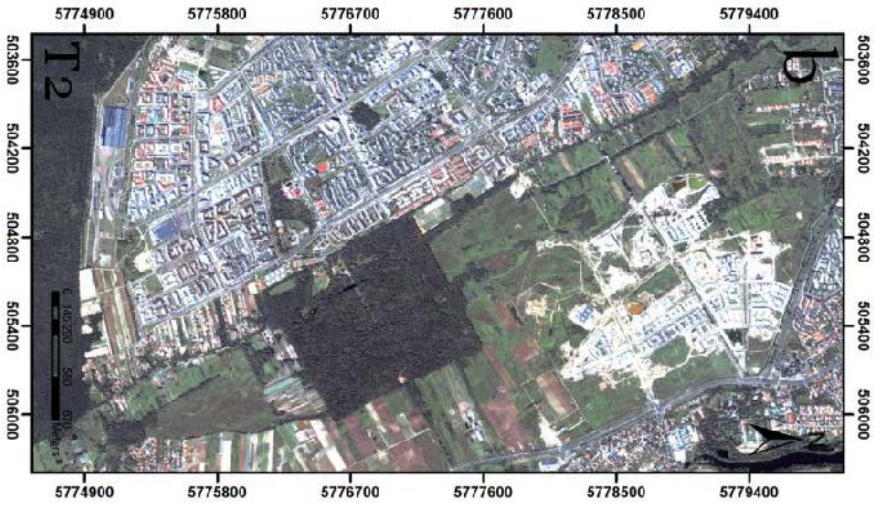
CHANGE DETECTION IN REMOTE SENSING

A key problem **in remote sensing** is to **detect changes, due to manmade or natural phenomenon,** by comparing **multiple (satellite) images at different times.**

In the remote sensing literature this is typically referred to as **change detection**

reference

target



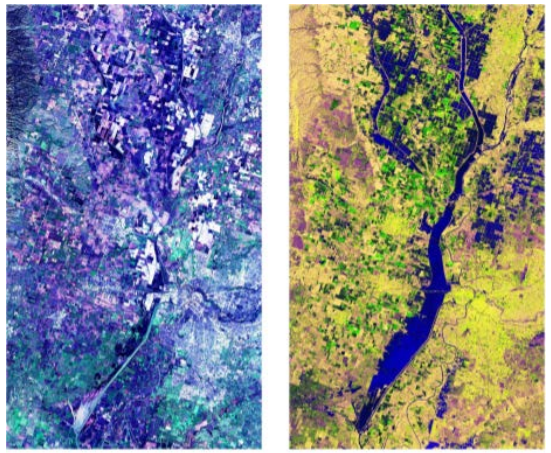
Boracchi, Carrera, ICPR 2020

MULTIMODAL, REFERENCE BASED ANOMALY DETECTION

Non trivial when direct comparison is prevented:

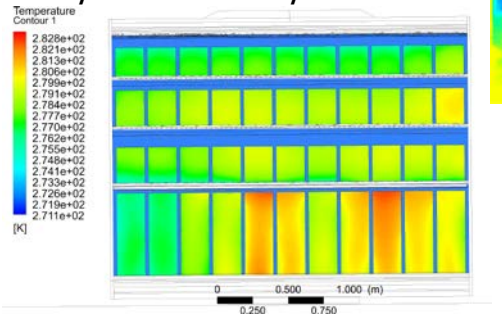
- Reference and **target might not be aligned nor easy to register with a global transformation**
- Reference and target might be **from different modalities / resolution / view**

Multispectral vs SAR images
California flood 2017

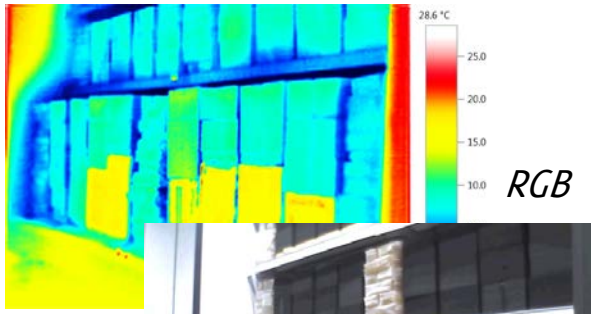


(a) Landsat 8 (t_1) (b) Sentinel-1A (t_2)

Temperature Map from CFD



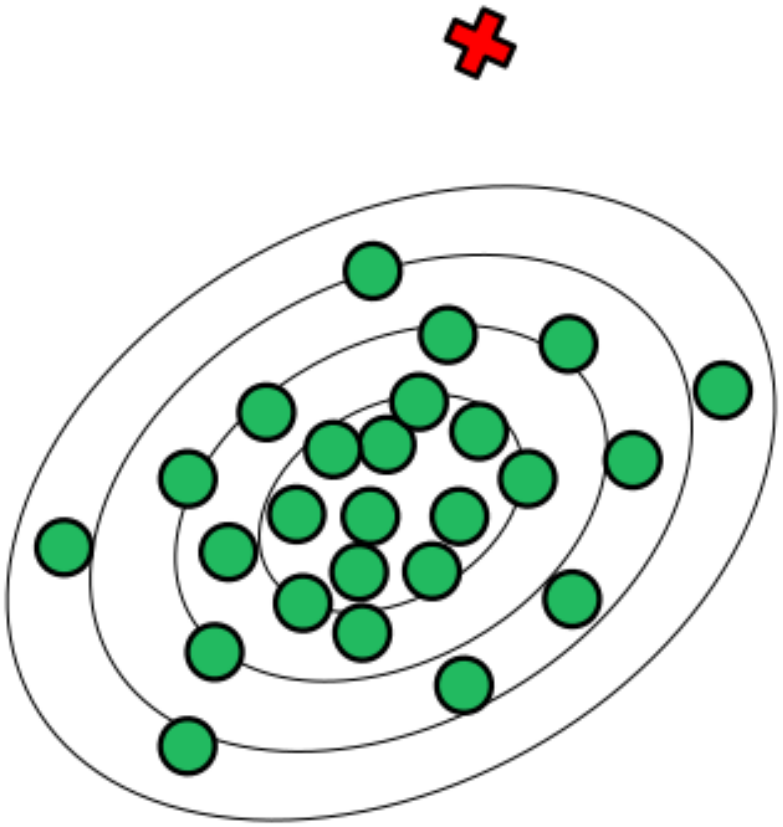
IR



RGB

L. T. Luppino, F. M. Bianchi, G. Moser, S. N. Anfinsen, *"Unsupervised Image Regression for Heterogeneous Change Detection"* IEEE Trans. on Geoscience and Remote Sensing (2019)

TRADITIONAL ANOMALY DETECTION SETUP

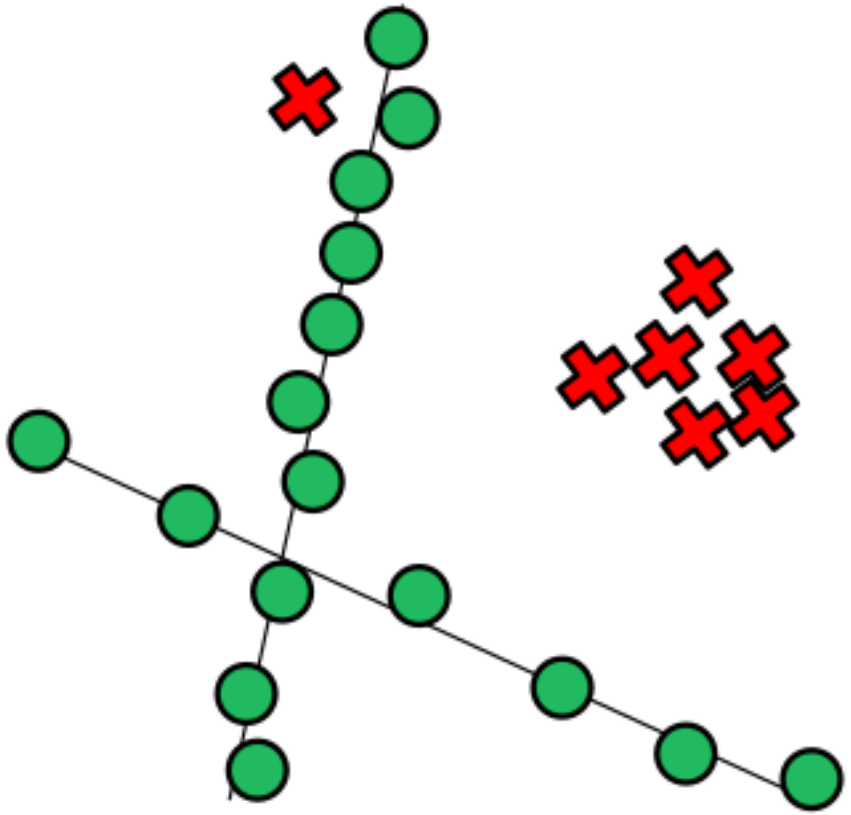


Anomaly detection deals with the problem of **identifying data** that **do not conform to an expected behavior**.

In the statistical and data-mining literature, anomalies are typically detected as **samples falling in low-density regions** of a probability density model describing the data.

Boracchi, Carrera, ICPR 2020

ANOMALY DETECTION IN A PATTERN RECOGNITION SETUP



Density-based or distance-based techniques are not effective to detect anomalies in a **pattern-recognition** setup, where anomalies are samples that **deviate from unknown structures or patterns**.

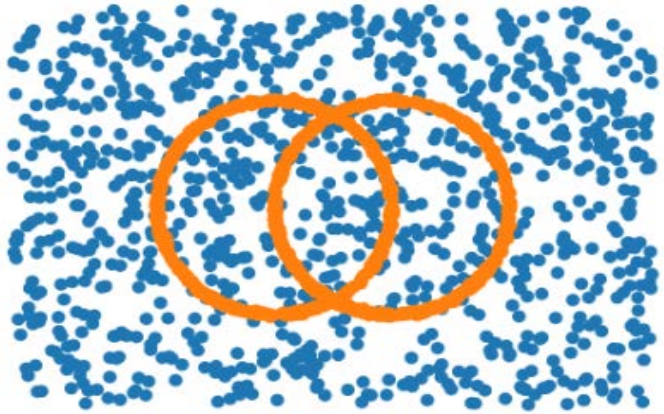
PIF: Preference Isolation Forest, detects these kind of anomalies thanks to

- Embedding in Preference Space
- An ad-hoc forest for detecting outliers in the preference space

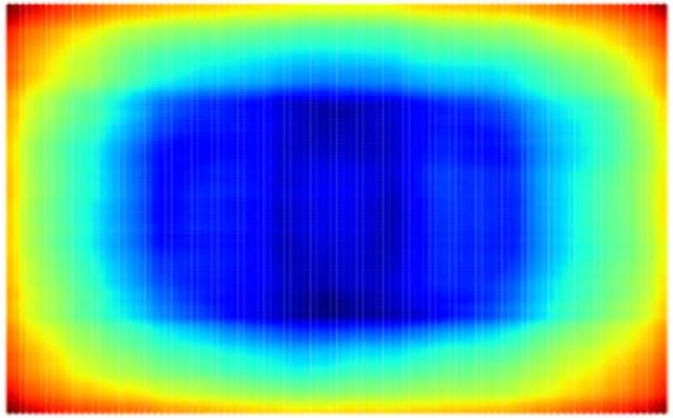
F. Leveni, L. Magri, G. Boracchi, C. Alippi, "Anomaly detection via preference embedding" ICPR 2020

ANOMALY DETECTION

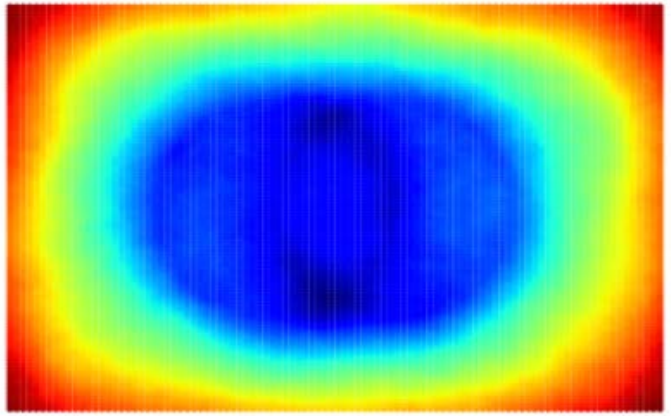
Anomaly Scores



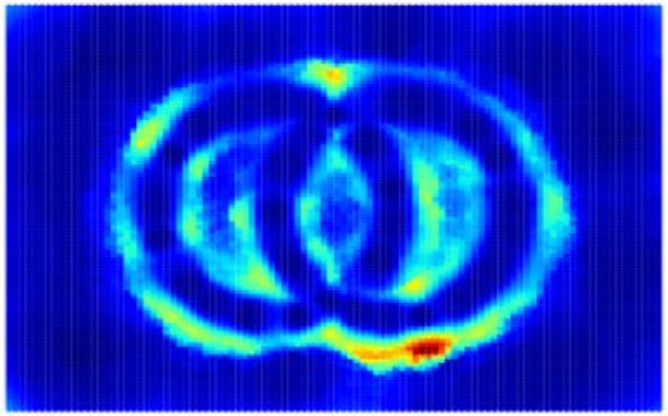
Ground truth



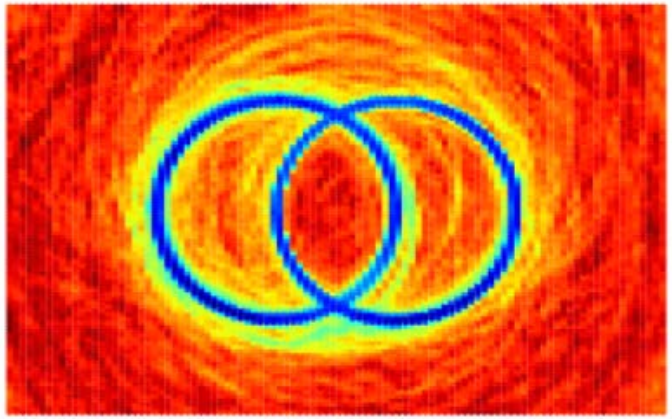
IFOR [1]



EIFOR [2]



LOF [3]



PIF [4]

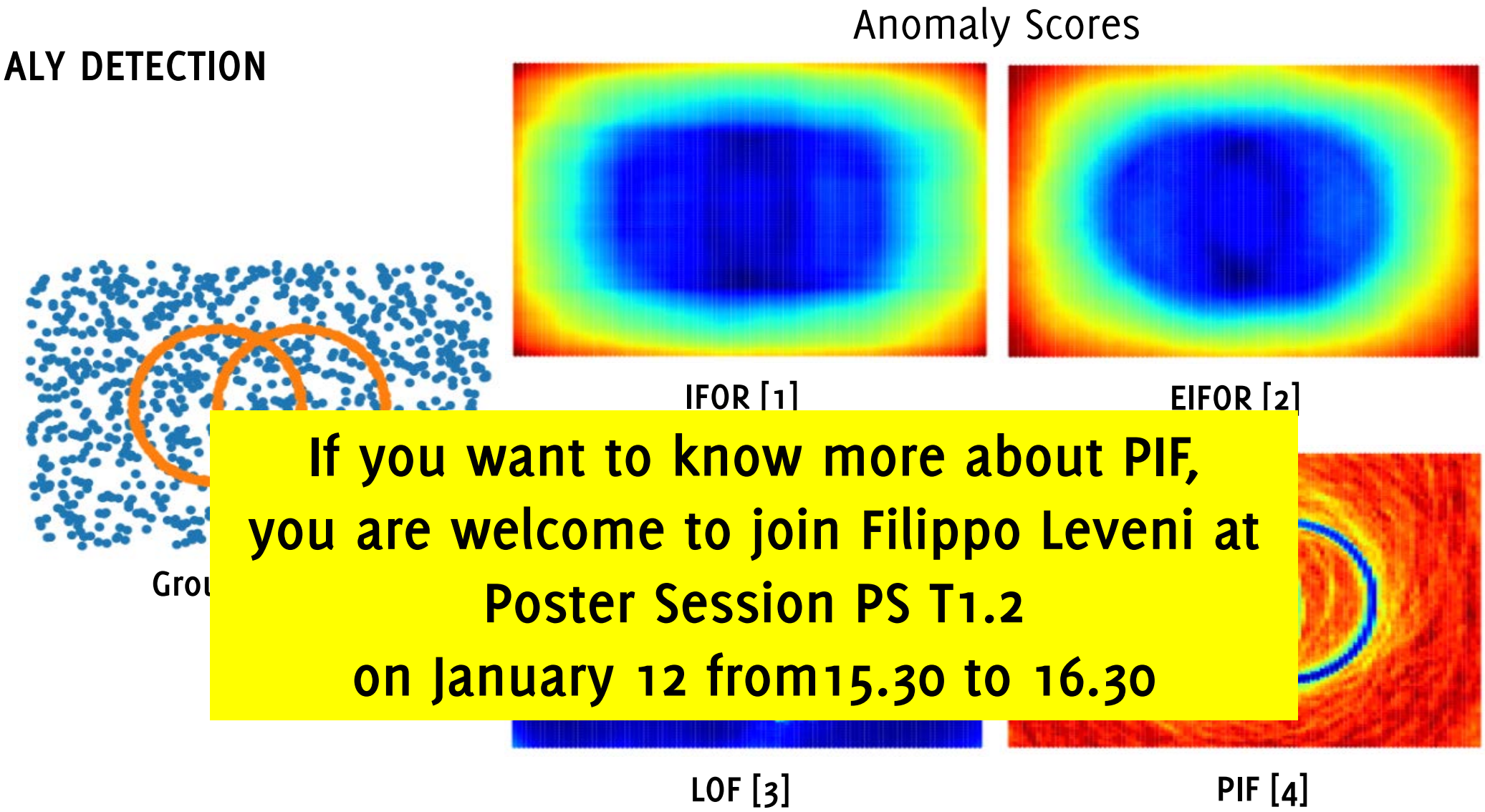
[1] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest”, in International Conference on Data Mining, IEEE, 2008, pp. 413–422.

[2] S. Hariri, M. C. Kind, and R. J. Brunner, “Extended isolation forest”, TKDE, 2019.

[3] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “Lof: Identifying density-based local outliers”, in International Conference on Management of data, 2000

[4] F. Leveni, L. Magri, G. Boracchi, C. Alippi, “Anomaly detection via preference embedding” ICPR 2020

ANOMALY DETECTION



If you want to know more about PIF, you are welcome to join Filippo Leveni at Poster Session PS T1.2 on January 12 from15.30 to 16.30

[1] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest", in International Conference on Data Mining, IEEE, 2008, pp. 413–422.

[2] S. Hariri, M. C. Kind, and R. J. Brunner, "Extended isolation forest", TKDE, 2019.

[3] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers", in International Conference on Management of data, 2000

[4] F. Leveni, L. Magri, G. Boracchi, C. Alippi, "Anomaly detection via preference embedding" ICPR 2020

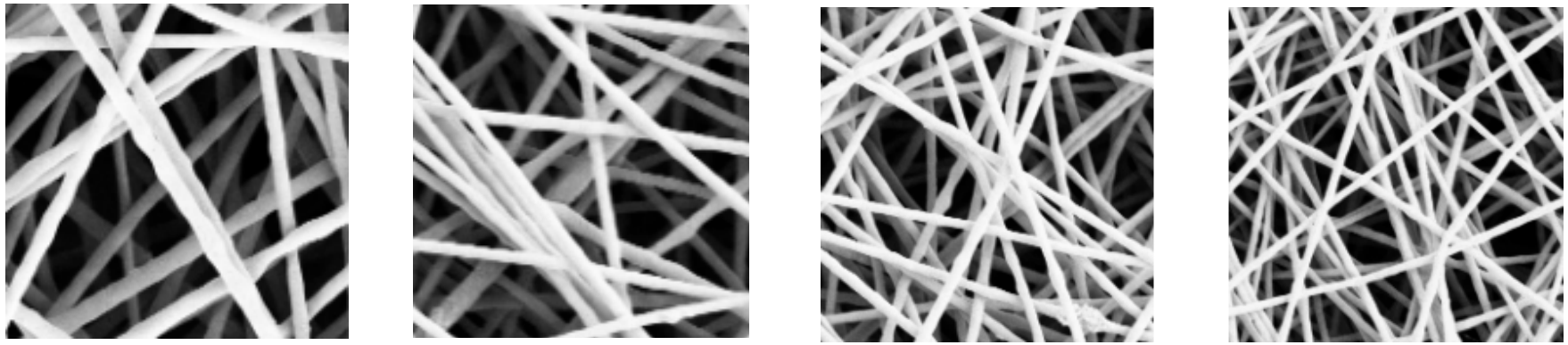
COUNTERACTING DOMAIN SHIFT IN ANOMALY DETECTION

Adaptation Strategies

NEED FOR ADAPTATION

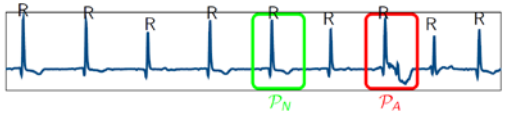
A challenge often occurring when performing online monitoring

Test data might differ from training data: **need of adaptation, otherwise anomaly detection methods would be ineffective**



Defects have to be detected at different zooming levels, that might not be present in the training set.

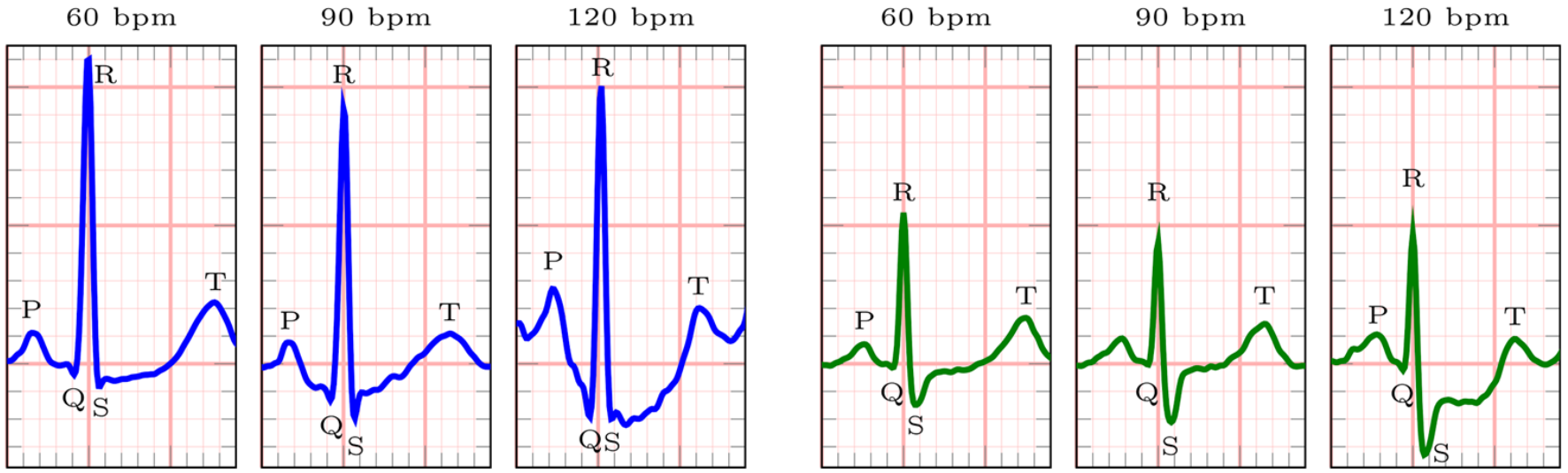
Boracchi, Carrera, ICPR 2020



NEED FOR ADAPTATION

A challenge often occurring when performing online monitoring

Test data might differ from training data: **need of adaptation, otherwise anomaly detection methods would be ineffective**



The **heartbeats** get **transformed** when the **heart rate changes**: **learned models have to be adapted** according to the heart rate.

Boracchi, Carrera, ICPR 2020

MODEL ADAPTATION

In the machine-learning literature these problems go under the name of transfer learning / domain adaptation

Transfer Learning (TL): adapt a model learned in the *source domain* (e.g. heartbeats at a given heartrate / fibers at a certain zoom level) to a *target domain* (e.g. heartbeats at an higher heartrate / fibers zoomed in or out)

Many TL methods have been designed for supervised / semi-supervised / unsupervised methods, depending on the availability of (annotated) data in the source and target domains.

In most anomaly detection settings, **no labels in the target data are provided** (typically they are not even provided in the source domain)

S. J. Pan and Q. Yang *"A survey on transfer learning"* IEEE TKDE 2010

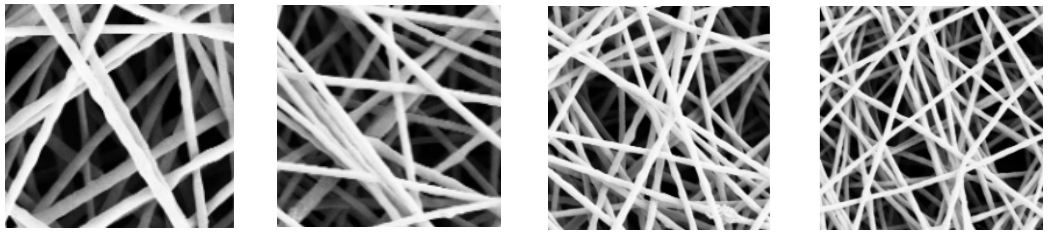
S. Shekhar, V. M. Patel, H. V. Nguyen, & R. Chellappa, *"Generalized domain-adaptive dictionaries,"* CVPR 2013

DOMAIN ADAPTATION ON QUALITY INSPECTION

SEM images can be acquired at different zooming levels

Solution:

- **Synthetically generate** training images at **different zooming levels**
- **Learn a dictionary** D_i at each scale
- Combine the learned dictionaries in a **multiscale dictionary** D



$$D = [\ D_1 \quad D_2 \quad D_3 \quad D_4\ ]$$

D. Carrera, G. Boracchi, A. Foi and B. Wohlberg *"Scale-invariant Anomaly Detection With multiscale Group-sparse Models"* ICIP 2016

DOMAIN ADAPTATION ON QUALITY INSPECTION

SEM images can be acquired at different zooming levels

Solution:

- **Synthetically generate** training images at **different zooming levels**
- **Learn a dictionary** D_i at each scale
- Combine the learned dictionaries in a **multiscale dictionary** D
- **Sparse-coding** including a penalized, **group sparsity term**

$$\boldsymbol{\alpha} = \underset{\boldsymbol{a}\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2}\left|\left|\boldsymbol{s} - D\boldsymbol{a}\right|\right|_2^2 + \lambda\left|\left|\boldsymbol{a}\right|\right|_1 + \mu\sum_{\boldsymbol{i}}\left|\left|\boldsymbol{a}\right|\right|_2$$

D. Carrera, G. Boracchi, A. Foi and B. Wohlberg *"Scale-invariant Anomaly Detection With multiscale Group-sparse Models"* ICIP 2016

DOMAIN ADAPTATION ON QUALITY INSPECTION

SEM images can be acquired at different zooming levels

Solution:

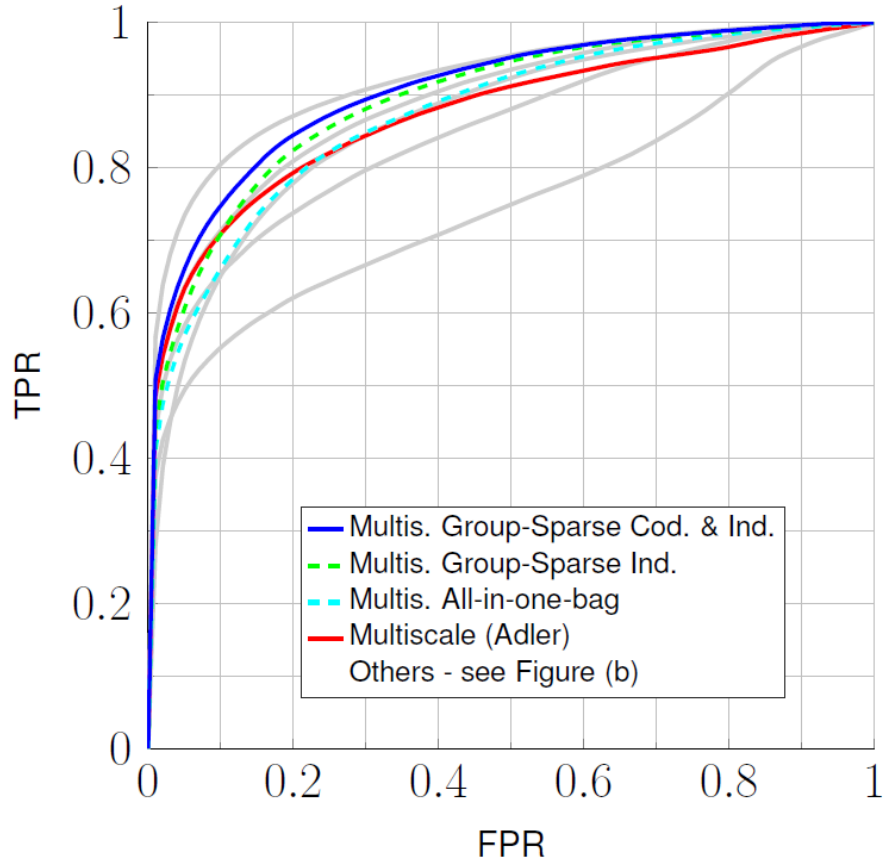
- **Synthetically generate** training images at **different zooming levels**
- **Learn a dictionary** D_i at each scale
- Combine the learned dictionaries in a **multiscale dictionary** D
- **Sparse-coding** including a penalized, **group sparsity term**
- Monitor a tri-variate feature vector

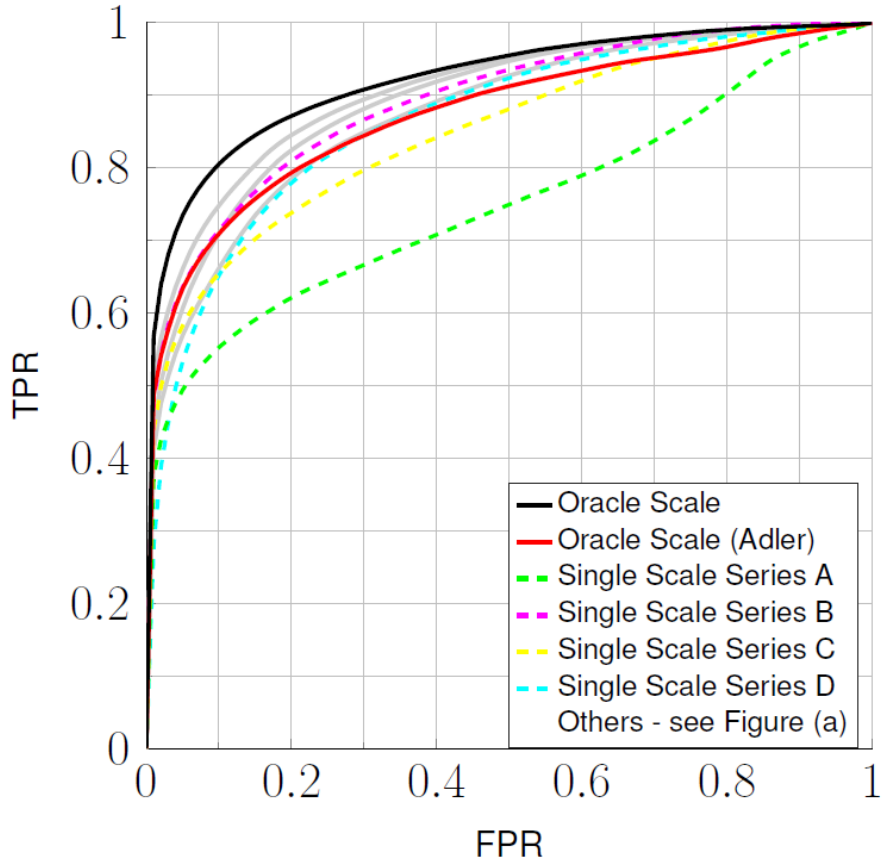
$$\boldsymbol{x} = \begin{bmatrix} \left|\left|\boldsymbol{s} - D\boldsymbol{\alpha}\right|\right|_2^2 \\ \left|\left|\boldsymbol{\alpha}\right|\right|_1 \\ \sum_i \left|\left|\boldsymbol{\alpha}_i\right|\right|_2 \end{bmatrix}$$

D. Carrera, G. Boracchi, A. Foi and B. Wohlberg *"Scale-invariant Anomaly Detection With multiscale Group-sparse Models"* ICIP 2016

DOMAIN ADAPTATION ON QUALITY INSPECTION

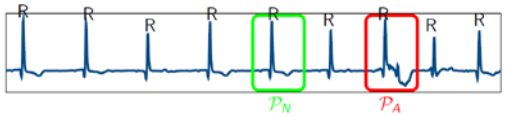
Performance on SEM image dataset acquired at 4 different zooming levels (A,B,C,D). It is important to include group-sparsity regularization also in the sparse coding stage





D. Carrera, G. Boracchi, A. Foi and B. Wohlberg "Scale-invariant Anomaly Detection With multiscale Group-sparse Models" ICIP 2016

DOMAIN ADAPTATION FOR ONLINE ECG MONITORING



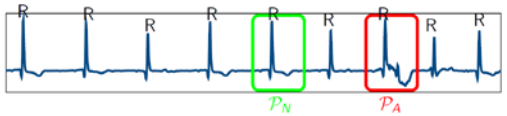
We propose to design linear transformations F_{r_1,r_0} to adapt user-specific dictionaries

$$D_{u,r_1} = F_{r_1,r_0} \cdot D_{u,r_0}, \qquad F_{r_0,r_1} \in \mathbb{R}^{m \times m}$$

D. Carrera, B. Rossi, P. Fragneto and G. Boracchi *"Domain Adaptation for Online ECG Monitoring"* ICDM 2017,

D. Carrera, B. Rossi, P. Fragneto and G. Boracchi "*Online Anomaly Detection for Long-Term ECG Monitoring using Wearable Devices*", Pattern Recognition 2019

DOMAIN ADAPTATION FOR ONLINE ECG MONITORING



We propose to design linear transformations F_{r_1,r_0} to adapt user-specific dictionaries

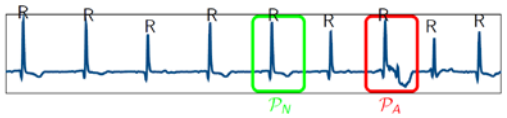
$$D_{u,r_1} = F_{r_1,r_0} \cdot D_{u,r_0}, \qquad F_{r_0,r_1} \in \mathbb{R}^{m \times m}$$

Surprisingly **these transformations can be learned from a publicly available dataset** containing ECG recordings at different heart rates from several users.

D. Carrera, B. Rossi, P. Fragneto and G. Boracchi *"Domain Adaptation for Online ECG Monitoring"* ICDM 2017,

D. Carrera, B. Rossi, P. Fragneto and G. Boracchi "*Online Anomaly Detection for Long-Term ECG Monitoring using Wearable Devices*", Pattern Recognition 2019

LEARNING TRANSFORMATIONS



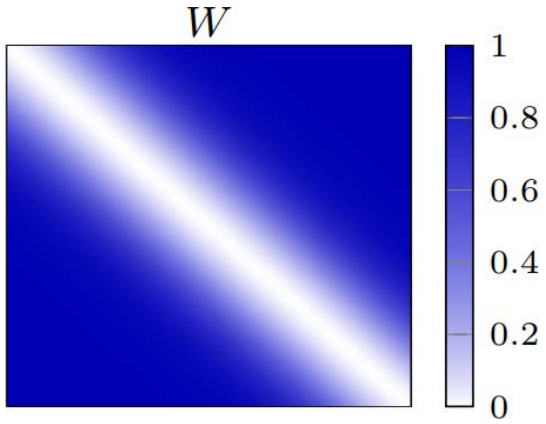
For each pair of heartrates (r_0, r_1) we learn F_{r_0,r_1} by solving the following optimization problem (involving data from L users of the LS-ST Dataset)

$$F_{r_1,r_0} = \underset{F,\{X_u\}}{\operatorname{argmin}} \left(\frac{1}{2} \sum_{u=1}^{L} \left\| S_{u,r_1} - F\, D_{u,r_0}\, X_u \right\|_F^2 + \mu \sum_{u=1}^{L} \|X_u\|_1 + \frac{\lambda}{2} \|W \odot F\|_2^2 + \xi \|W \odot F\|_1 \right)$$

Data-fidelity for heartbeats transformed by F, computed over all the L users

Sparsity

Weighted elastic net regularization to add stability and steer F towards desirable properties

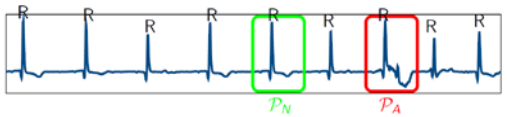


The matrix W is penalyzing less values along the diagonal of F, thus assuming transformation to be local, i.e., involging only neighbouring samples

D. Carrera, B. Rossi, P. Fragneto and G. Boracchi *"Domain Adaptation for Online ECG Monitoring"* ICDM 2017,

D. Carrera, B. Rossi, P. Fragneto and G. Boracchi "*Online Anomaly Detection for Long-Term ECG Monitoring using Wearable Devices*", Pattern Recognition 2019

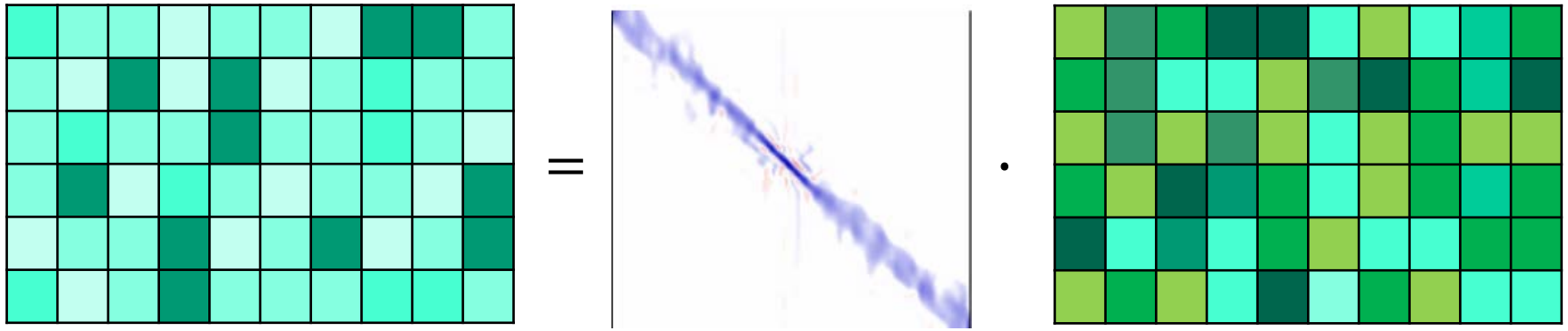
DOMAIN ADAPTATION FOR ONLINE ECG MONITORING



We adapt user-specific dictionaries through F_{r_1,r_0}

$$D_{u,r_1} = F_{r_1,r_0} \cdot D_{u,r_0}, \qquad F_{r_0,r_1} \in \mathbb{R}^{m \times m}$$

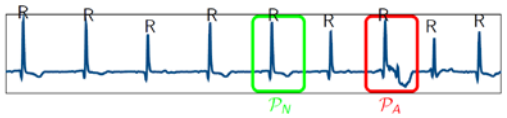
User-independent transformations enable accurate mapping of user-specific dictionaries



D. Carrera, B. Rossi, P. Fragneto and G. Boracchi *"Domain Adaptation for Online ECG Monitoring"* ICDM 2017,

D. Carrera, B. Rossi, P. Fragneto and G. Boracchi "*Online Anomaly Detection for Long-Term ECG Monitoring using Wearable Devices*", Pattern Recognition 2019

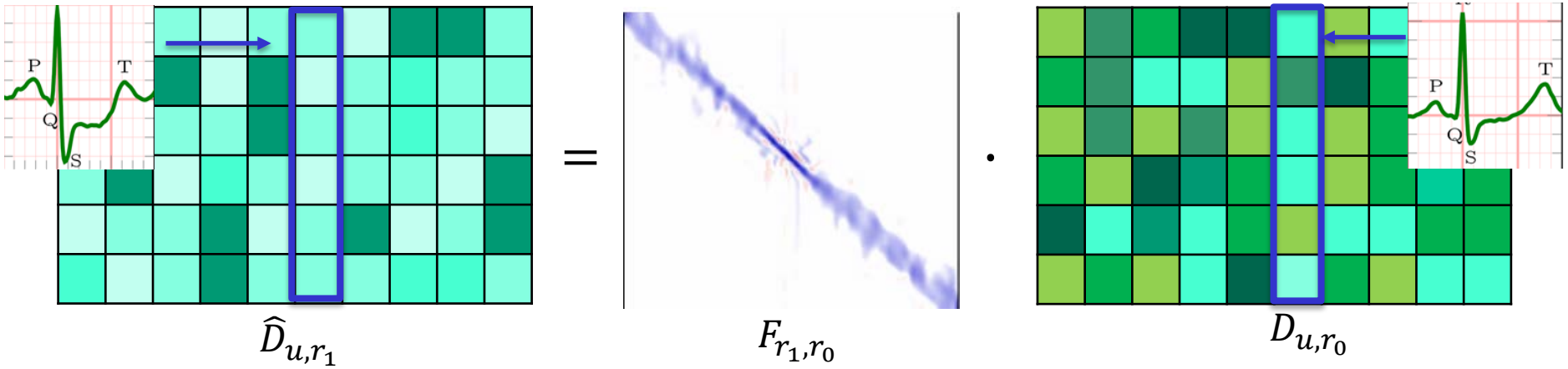
DOMAIN ADAPTATION FOR ONLINE ECG MONITORING



We adapt user-specific dictionaries through F_{r_1,r_0}

$$D_{u,r_1} = F_{r_1,r_0} \cdot D_{u,r_0}, \qquad F_{r_0,r_1} \in \mathbb{R}^{m \times m}$$

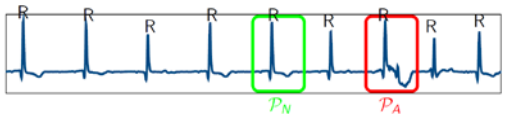
User-independent transformations enable accurate mapping of user-specific dictionaries



$\widehat{D}_{u,r_1}$ F_{r_1,r_0} D_{u,r_0}

Carrera D., Rossi B., Fragneto P., and Boracchi G. "Domain Adaptation for Online ECG Monitoring" ICDM 2017,

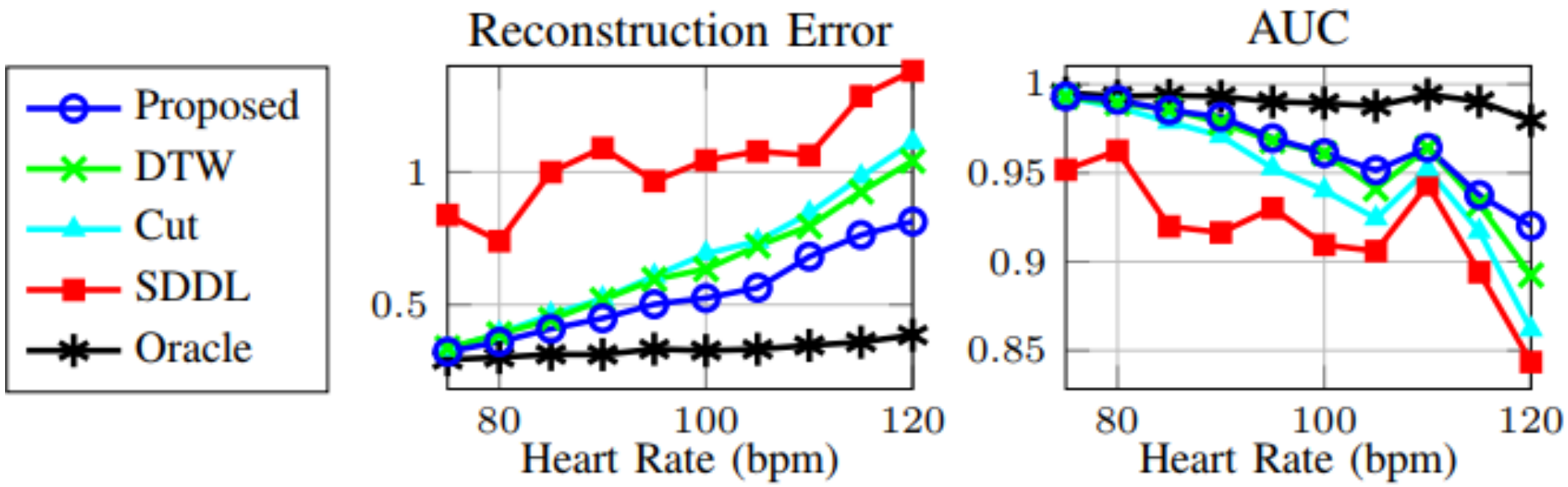
DICTIONARY ADAPTATION PERFORMANCE



The proposed domain adaptation solution achieves:

- lowest signal reconstruction error
- best anomaly detection performance (AUC)

Among alternative methods for dictionary adaptation

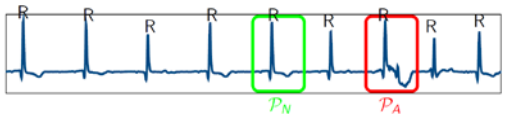


Carrera D., Rossi B., Fragneto P., and Boracchi G. "Domain Adaptation for Online ECG Monitoring" ICDM 2017,

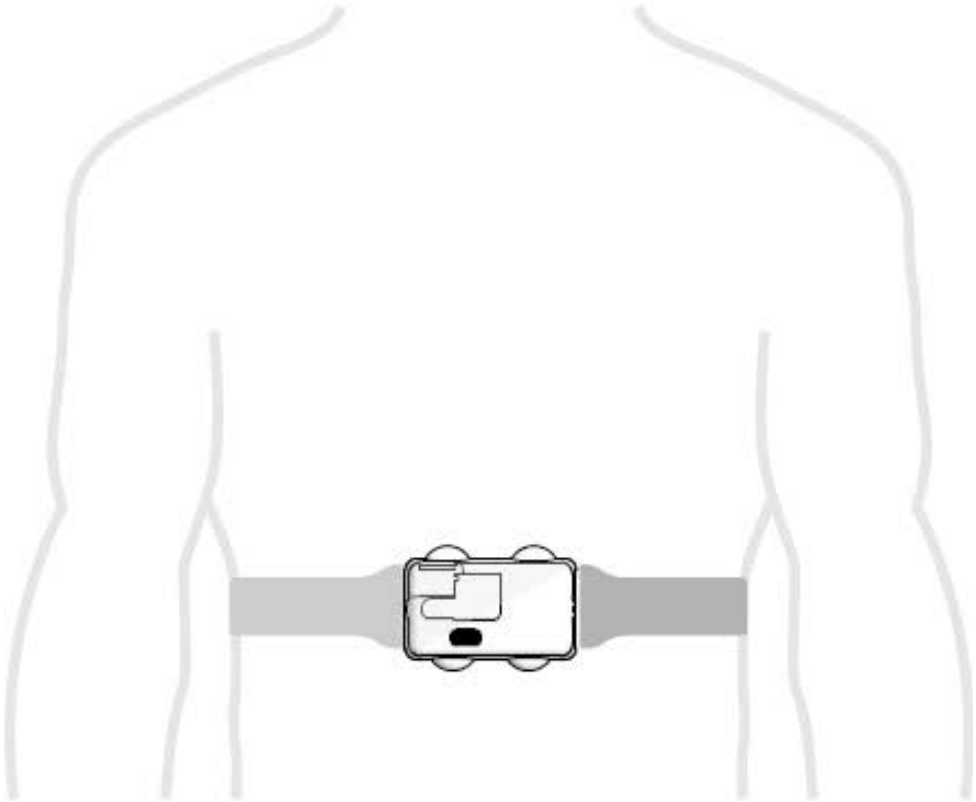
ONLINE ECG MONITORING BY WEARABLE DEVICES

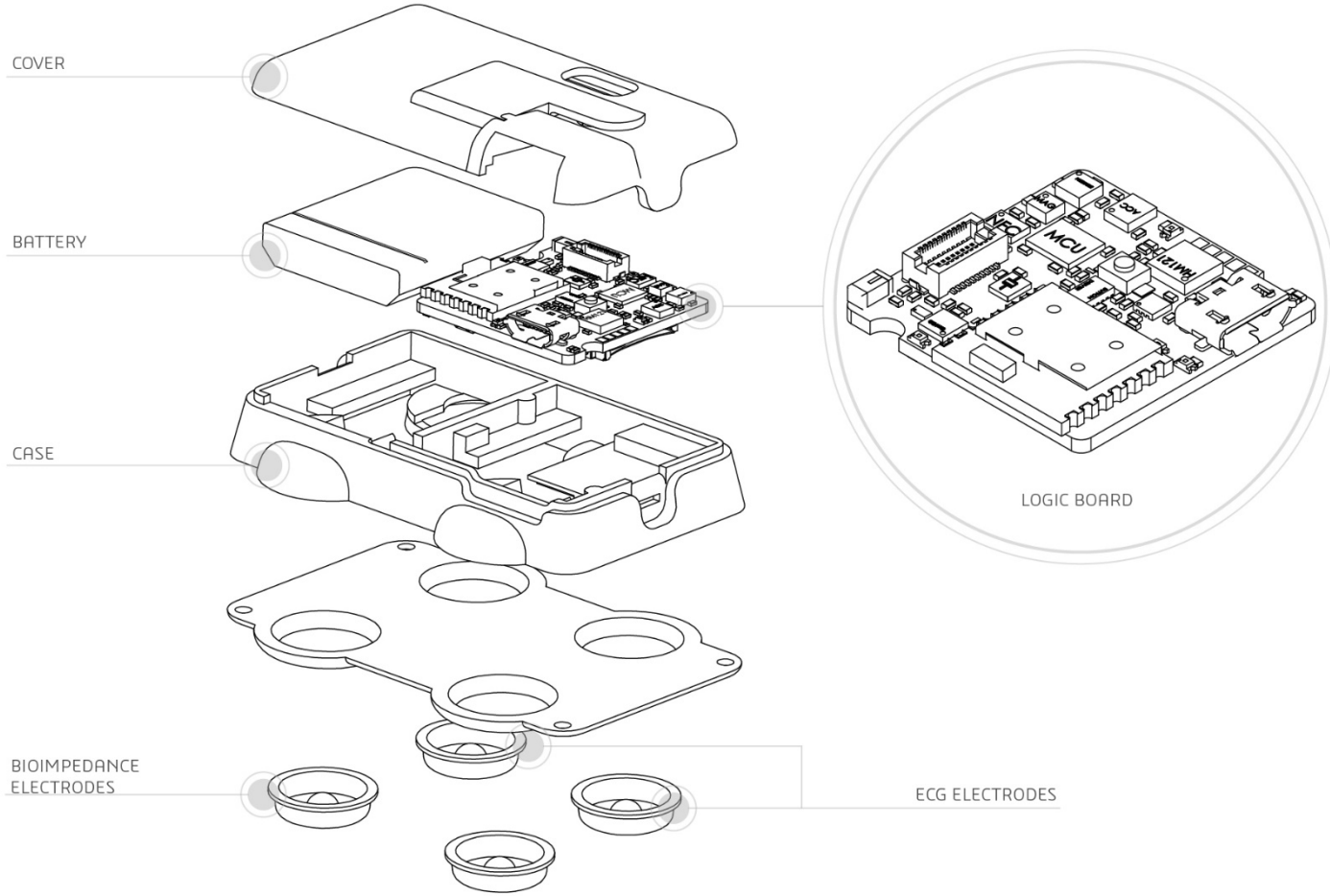
Boracchi, Carrera, ICPR 2020

THE BIO2BIT DEVICE

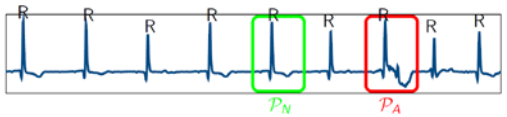


ECG signals are recorded by the BIO2BIT device

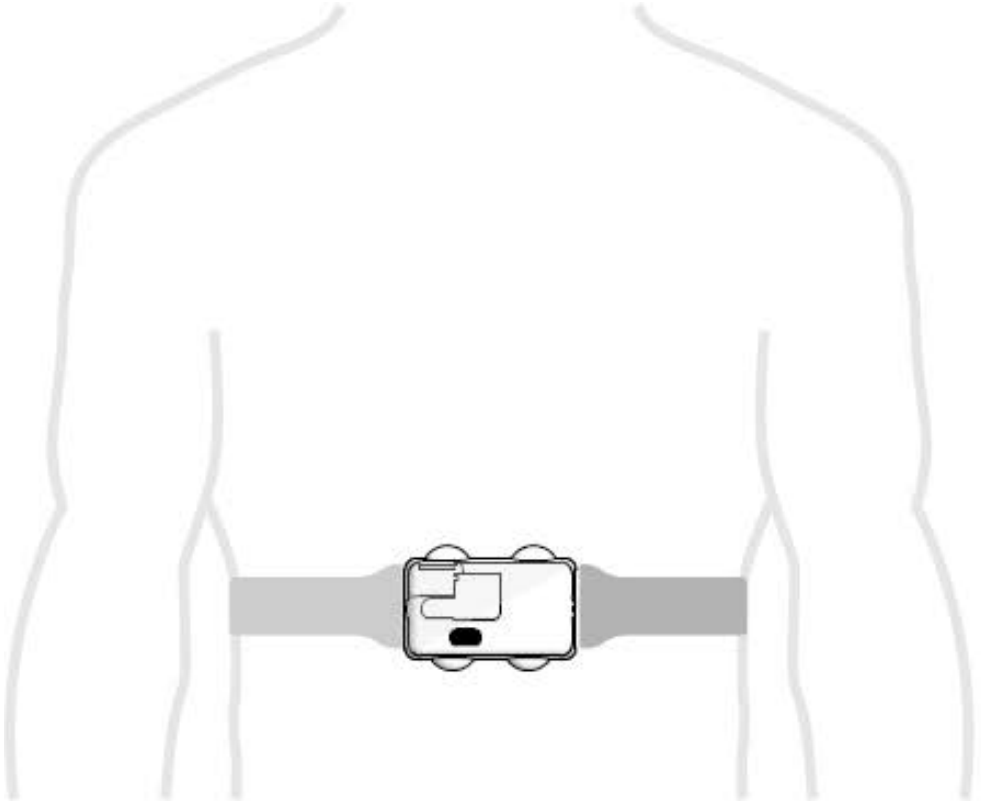




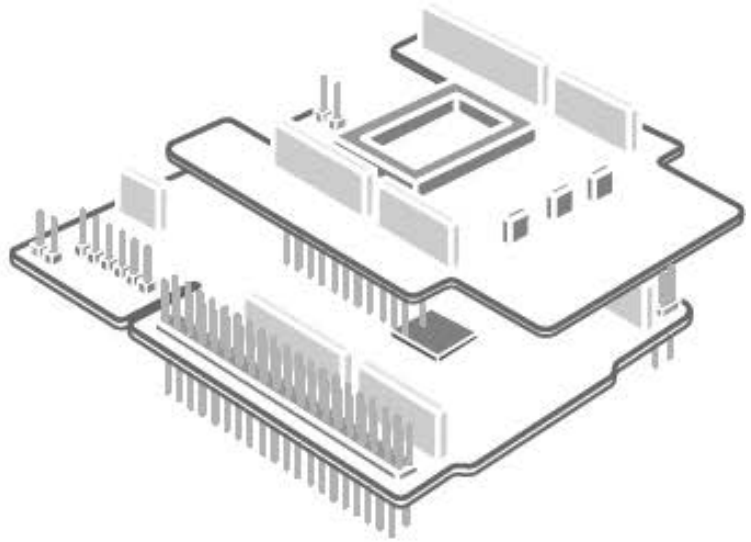
ONLINE ECG SIGNALS BY WEARABLE DEVICES



ECG signals are recorded by the BIO2BIT device



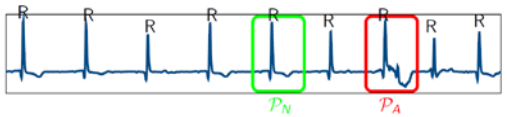
ECG are steadily transmitted via Bluethooth low-energy to a Dongle mounting a Nucleo



Nucleo
STM32L476RG

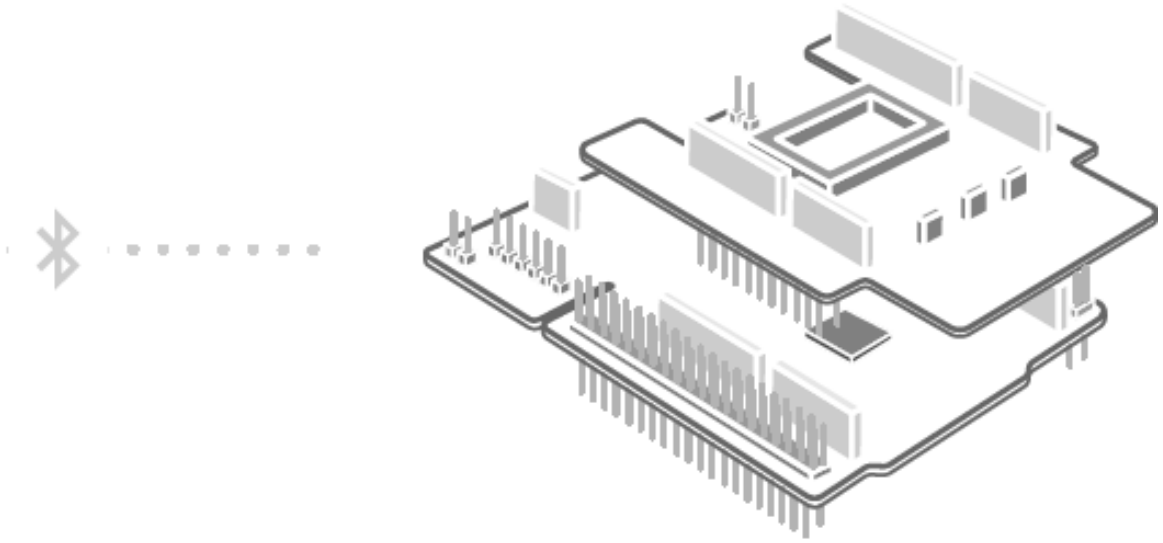
Boracchi, Carrera, ICPR 2020

ONLINE ECG SIGNALS BY WEARABLE DEVICES



Sparse Coding

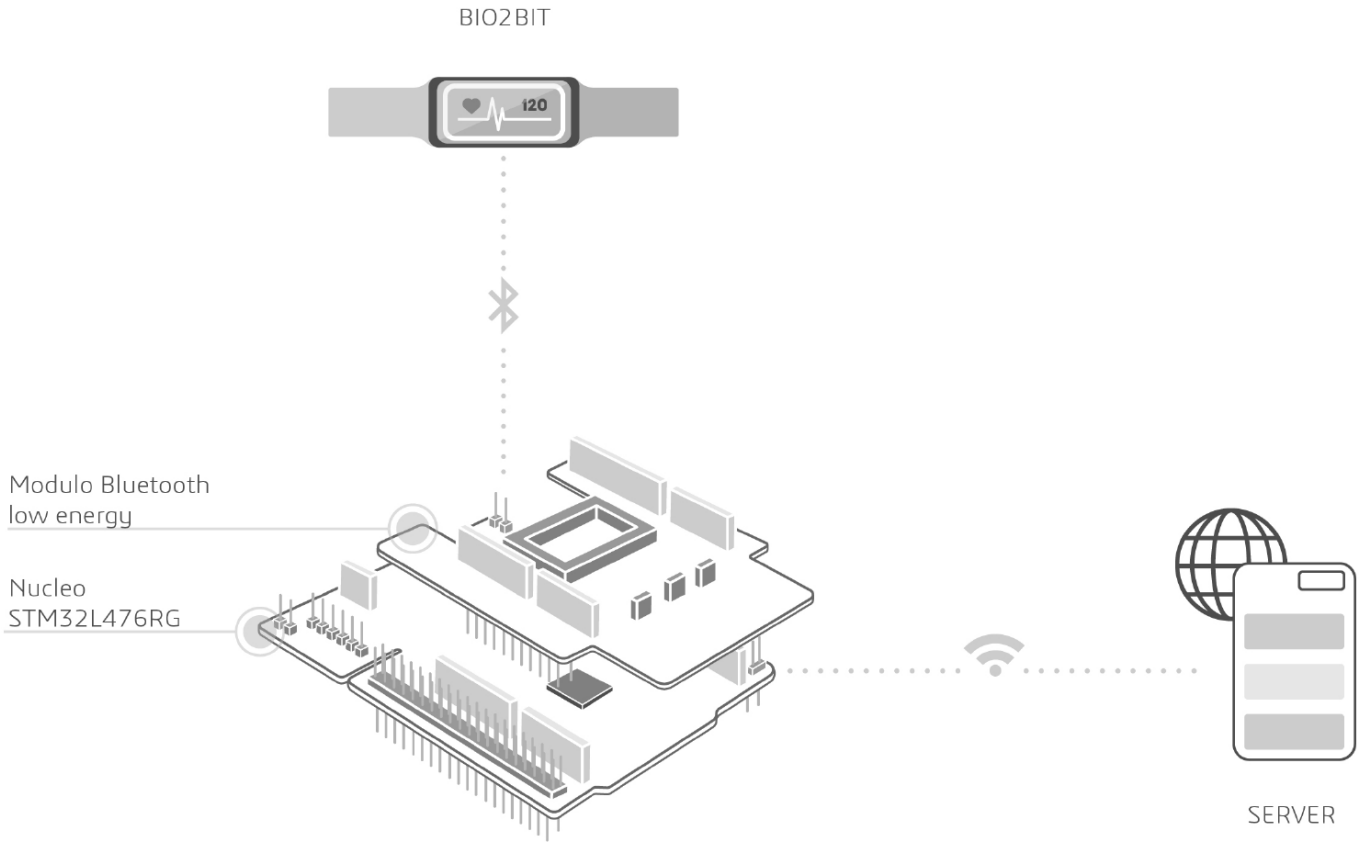
- Optimized OMP for underdetermined dictionaries
- Performed in real-time on such a low-power wearable device

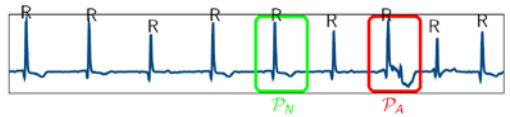


M. Longoni, D. Carrera, B. Rossi, P. Fragneto, M. Pessione and G. Boracchi “A Wearable Device for Online and Long-Term ECG Monitoring”, *International Joint Conference on Artificial Intelligence (IJCAI) 2018 - Demo Track*

D. Carrera, B. Rossi, P. Fragneto and G. Boracchi "*Online Anomaly Detection for Long-Term ECG Monitoring using Wearable Devices*", Pattern Recognition 2019

ONLINE ECG SIGNALS BY WEARABLE DEVICES





Dictionary Learning

5 minutes of ECG signals are enough to learn a dictionary D_{u,r_0} that is:

- User-specific
- Position-specific

Describing the morphology of the heartbeats of that specific user in resting conditions

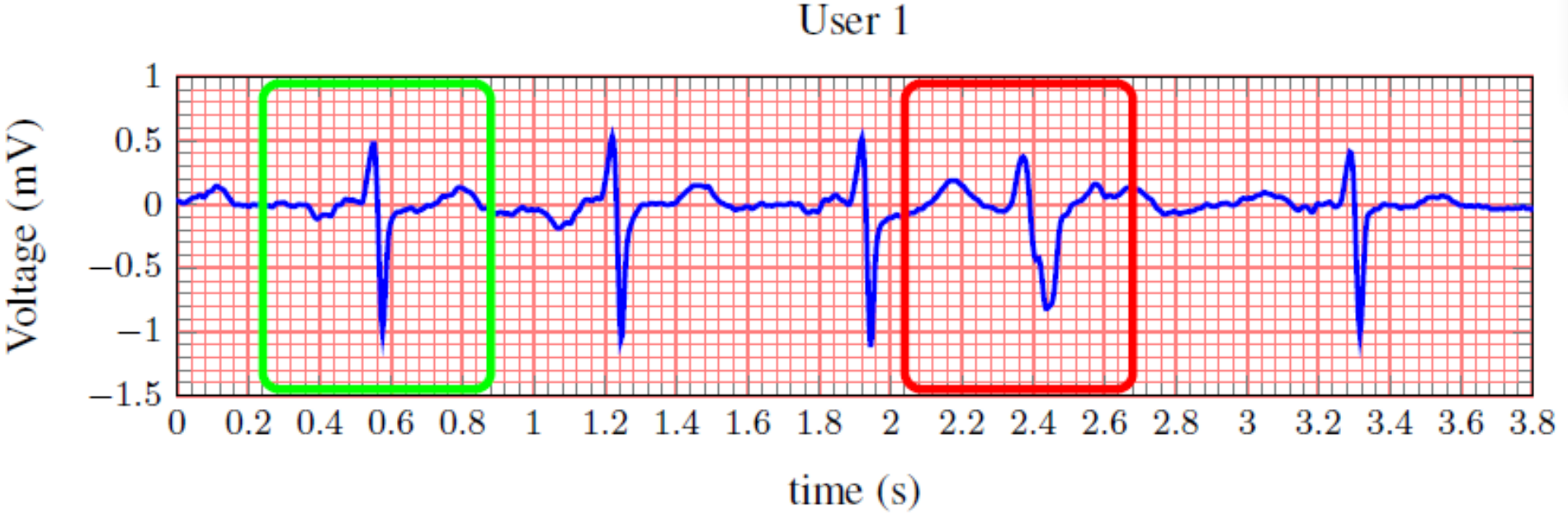
Dictionary Learning

- Conveniently performed on an host
- The learned dictionary D_{u,r_0} and all its transformed versions D_{u,r_i} are transferred to the dongle

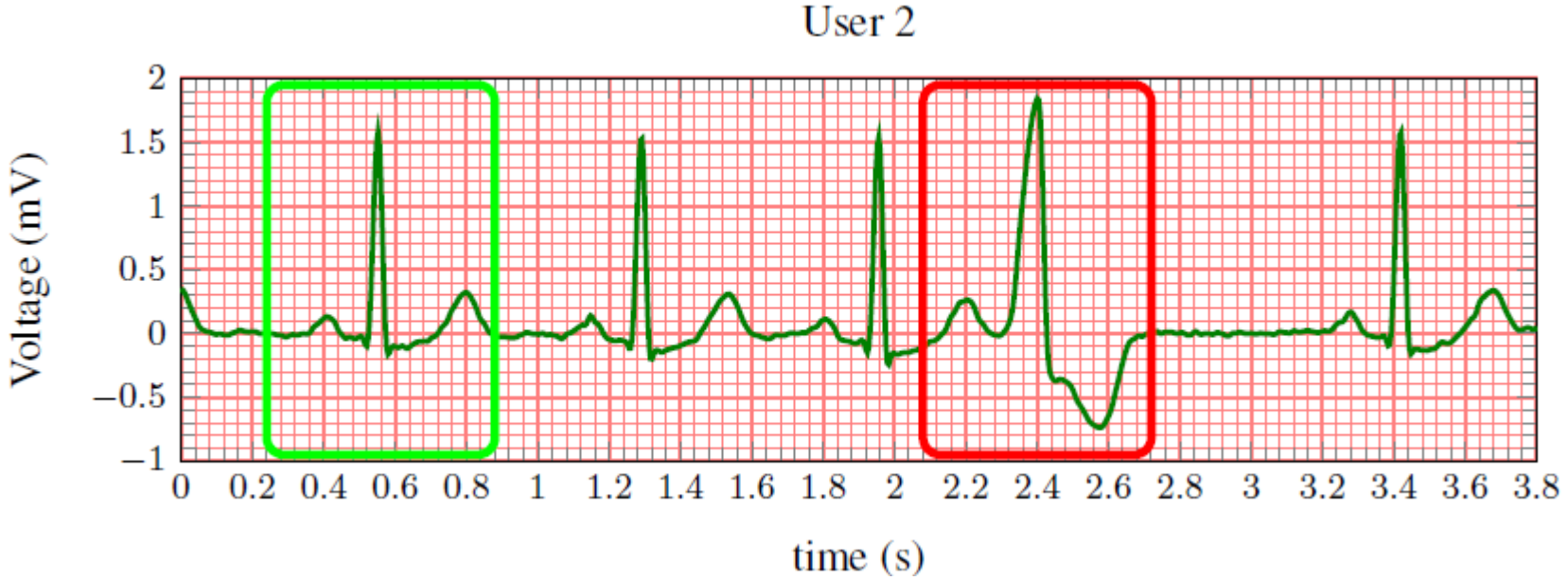
M. Longoni, D. Carrera, B. Rossi, P. Fragneto, M. Pessione and G. Boracchi “A Wearable Device for Online and Long-Term ECG Monitoring”, *International Joint Conference on Artificial Intelligence (IJCAI) 2018 - Demo Track*

D. Carrera, B. Rossi, P. Fragneto and G. Boracchi "*Online Anomaly Detection for Long-Term ECG Monitoring using Wearable Devices*", Pattern Recognition 2019

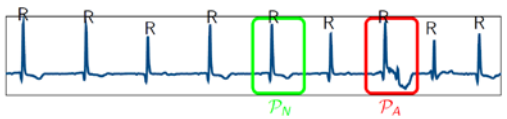
DIFFERENT USERS FEATURE DIFFERENT HEARTBEAT MORPHOLOGY



Dictionary has to be learned from each user / each device placement



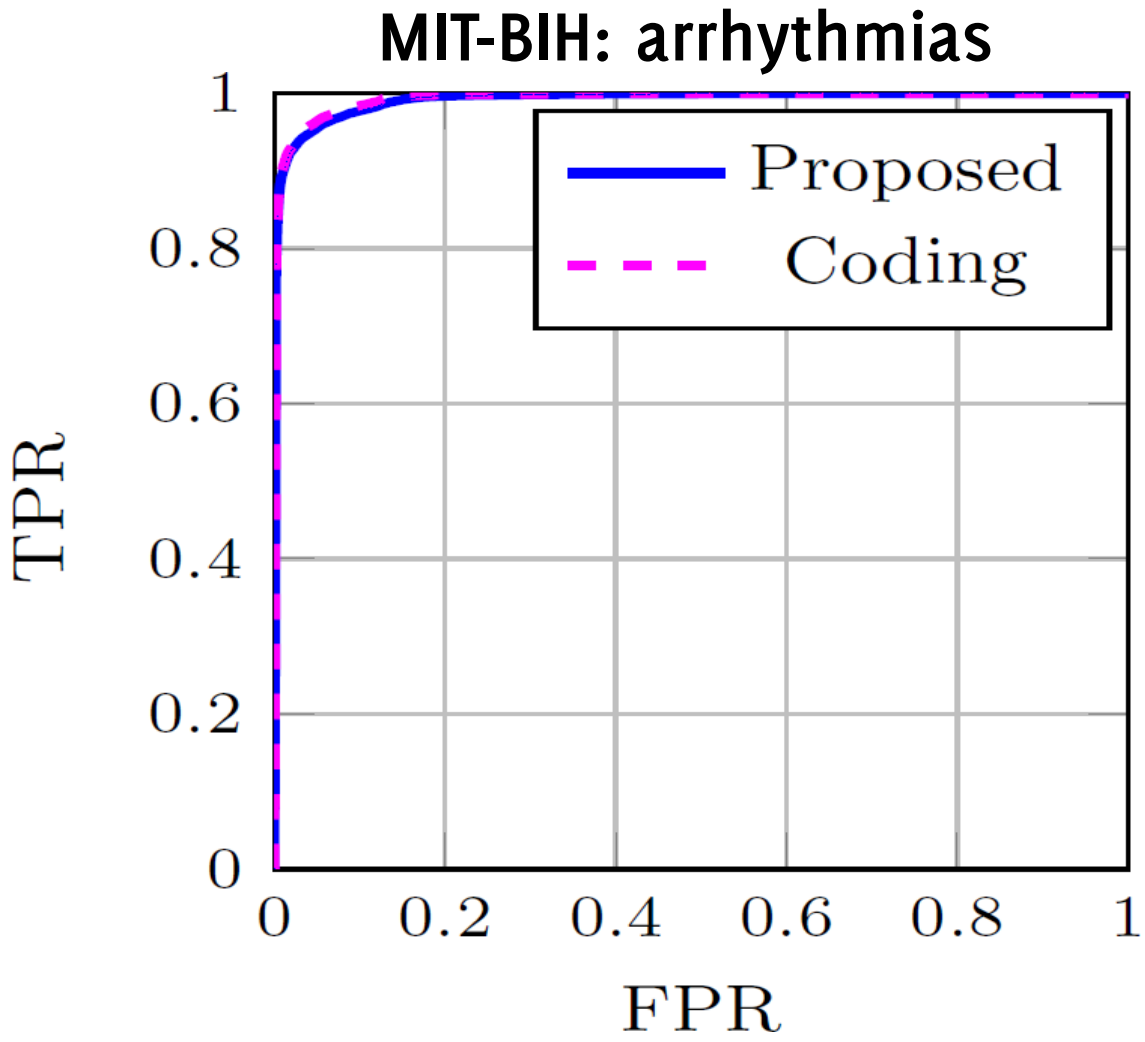
Boracchi, Carrera, ICPR 2020



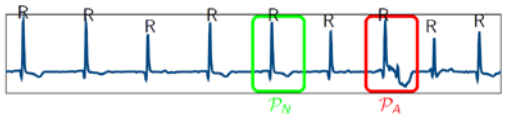
RESULTS: MIT-BIH DATASET

Our solution achieves competitive performance against a state-of-the-art anomaly detector on the MIT-BIH dataset.

However, our detector is much less computationally demanding



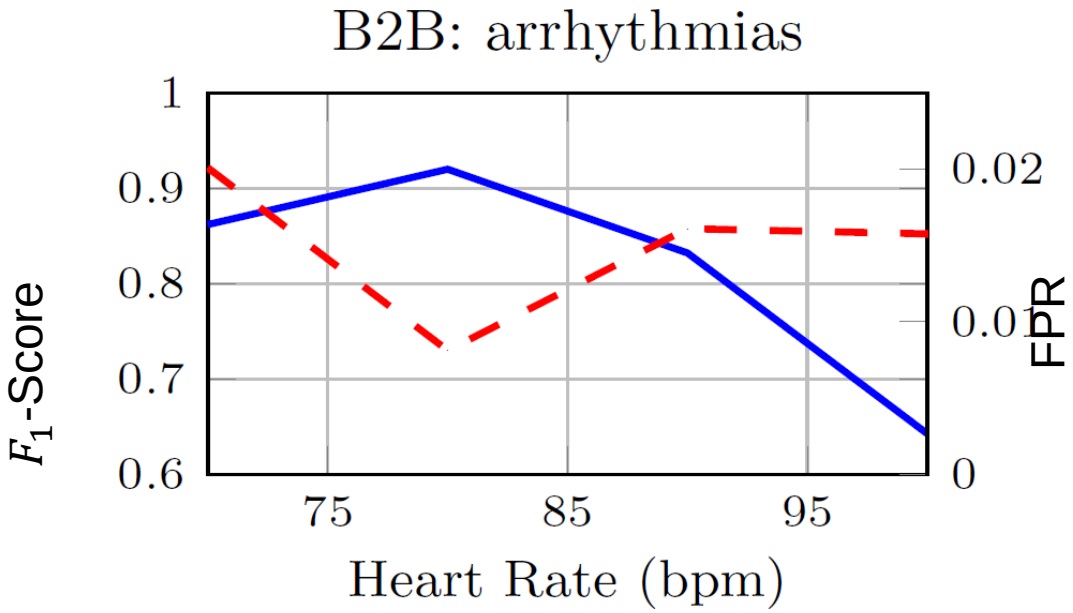
D. Carrera, B. Rossi, P. Fragneto and G. Boracchi "*Online Anomaly Detection for Long-Term ECG Monitoring using Wearable Devices*", Pattern Recognition 2019

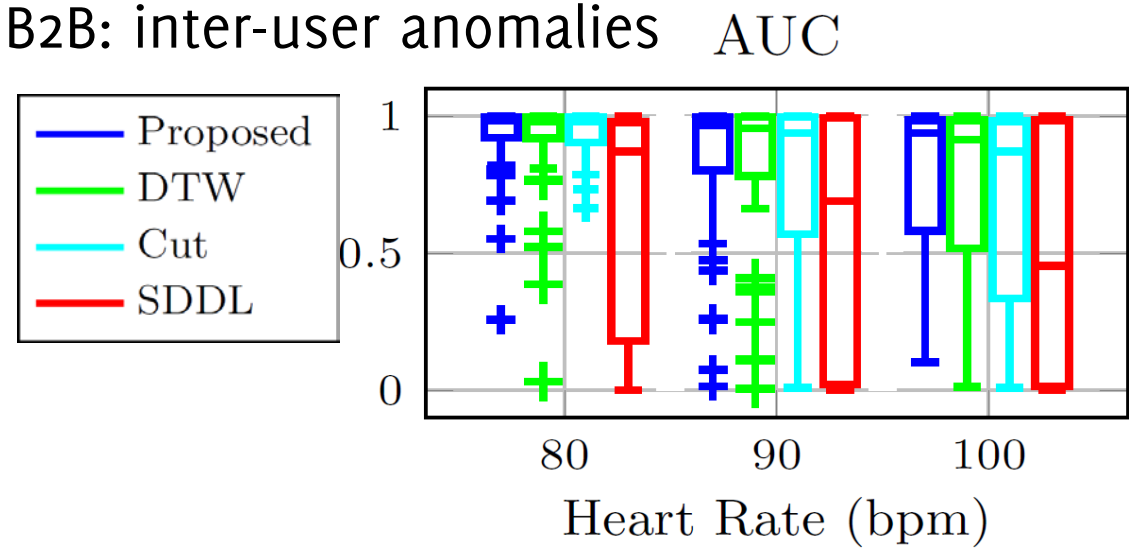


B2B DATASET (IN-HOUSE DATASET WITH ARRHYTHMIAS)

Both the AUC and the F_1-score are large when the heart rate increases.

The FPR is maintained almost constant





D. Carrera, B. Rossi, P. Fragneto and G. Boracchi "Online Anomaly Detection for Long-Term ECG Monitoring using Wearable Devices", Pattern Recognition 2019

Any Question?

DEEP LEARNING FOR ANOMALY DETECTION

IMAGE CLASSIFICATION

The problem: assigning to an *input image* $\boldsymbol{s}$ one *label* l from a *fixed set of* L *categories* Λ

$\boldsymbol{s}$ ⇨ "wheel" 65%, "tyre" 30%..

$\boldsymbol{s}$ ⇨ "castle" 55%, "tower" 43%..

Λ = {"wheel", "cars" ...."castle", "baboon", ... }

DEEP LEARNING AND IMAGE CLASSIFICATION

Since 2010 ImageNet organizes **ILSVRC** (ImageNet Large Scale Visual Recognition Challenge)

Classification error rate (top 5 accuracy):

- In 2011: 25%

Deep learning

- In 2012: 16% (achieved by a CNN)
- In 2017: < 5% (for 29 of 38 competing teams, deep learning)

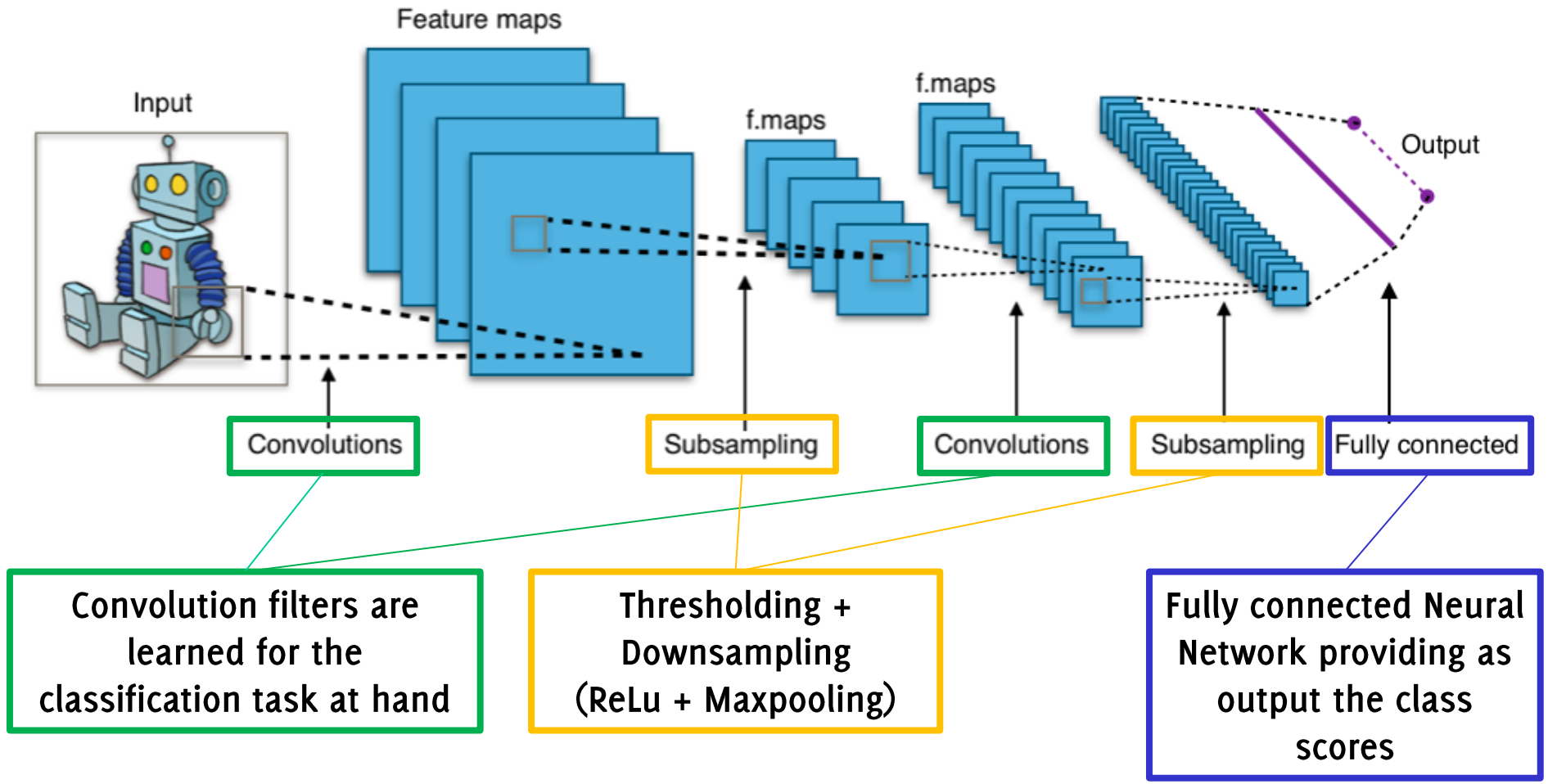
DEEP LEARNING AND IMAGE CLASSIFICATION

Deep Learning boasted image classification performance, thanks to

- Advances in parallel hardware (e.g. GPU)
- Availability of large annotated dataset (e.g. the ImageNet project is a large database visual recognition over 14M hand-annotated images in more than 20K categories)

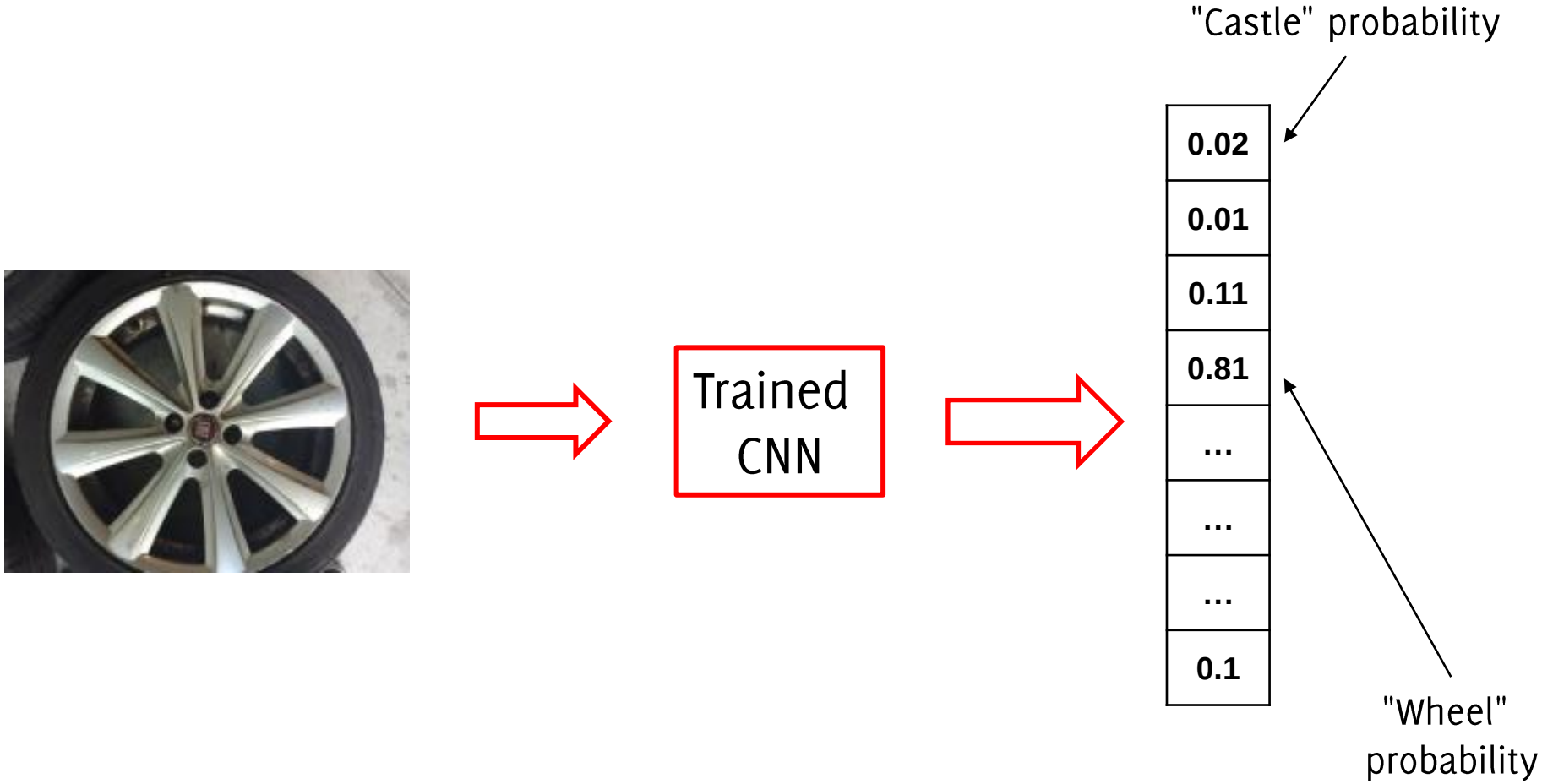
CONVOLUTIONAL NEURAL NETWORKS (CNN)

The typical architecture of a convolutional neural network



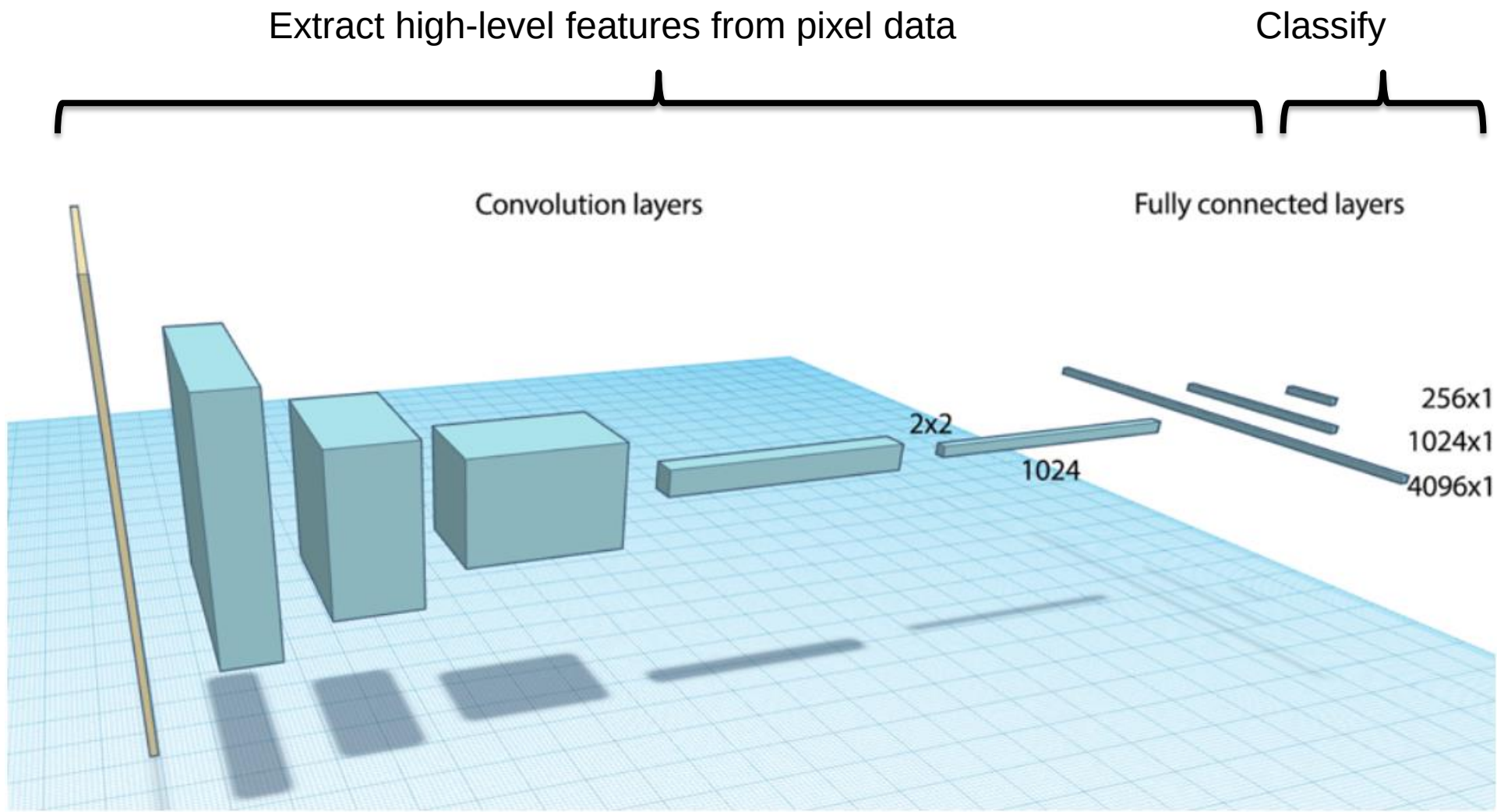
Boracchi, Carrera, ICPR 2020

THE OUTPUT OF A CNN



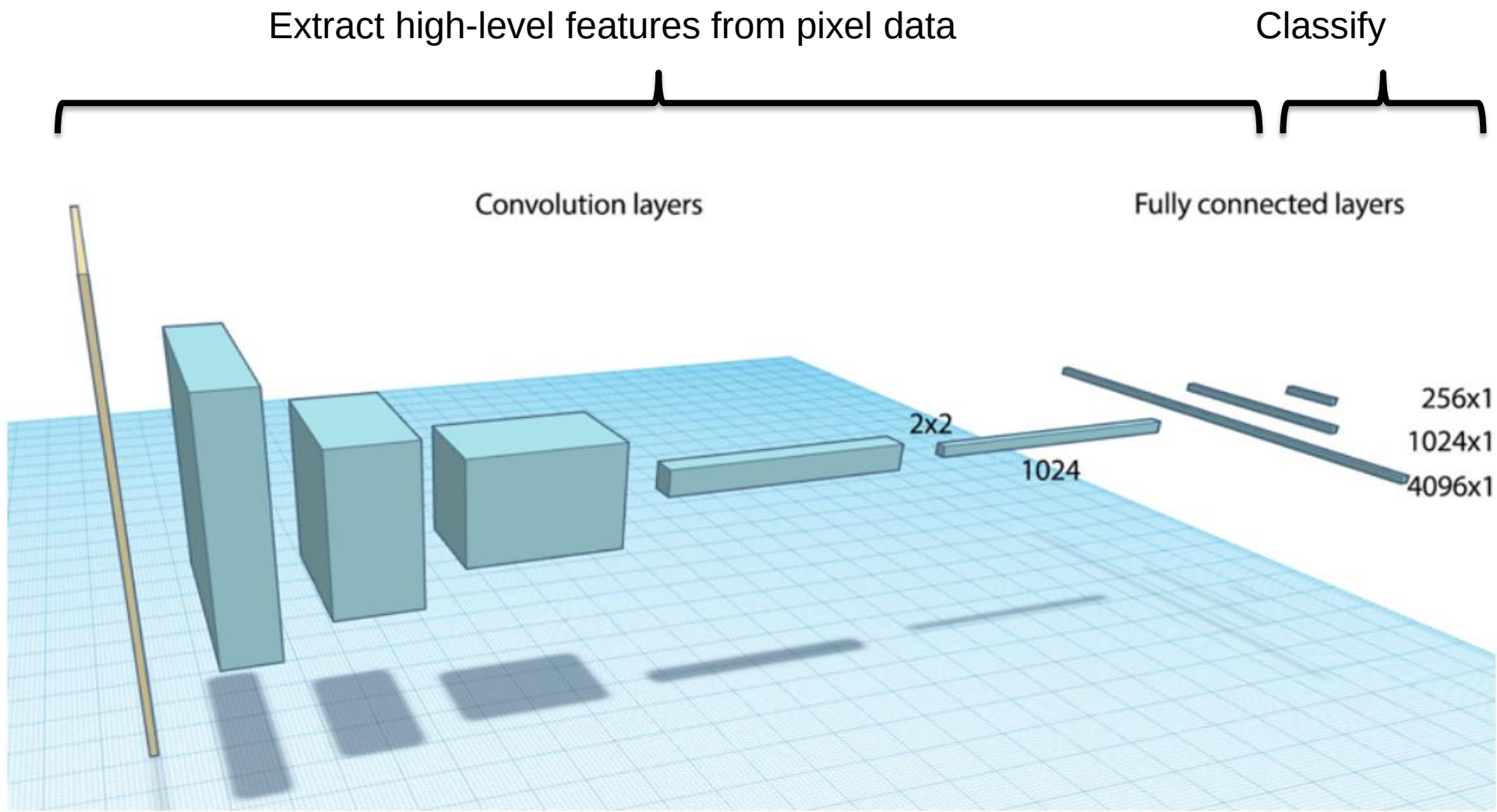
Boracchi, Carrera, ICPR 2020

CNN AS DATA-DRIVEN FEATURE EXTRACTOR



CNN AS DATA-DRIVEN FEATURE EXTRACTOR

The feature extractor and the classifier are jointly learned in a end-to-end fashion



SEMI-SUPERVISED APPROACHES

- CNN as data-driven feature extractor
  - Transfer learning
  - Self-supervised learning
  - Autoencoders
  - Domain-based
- Generative models

SEMI-SUPERVISED APPROACHES

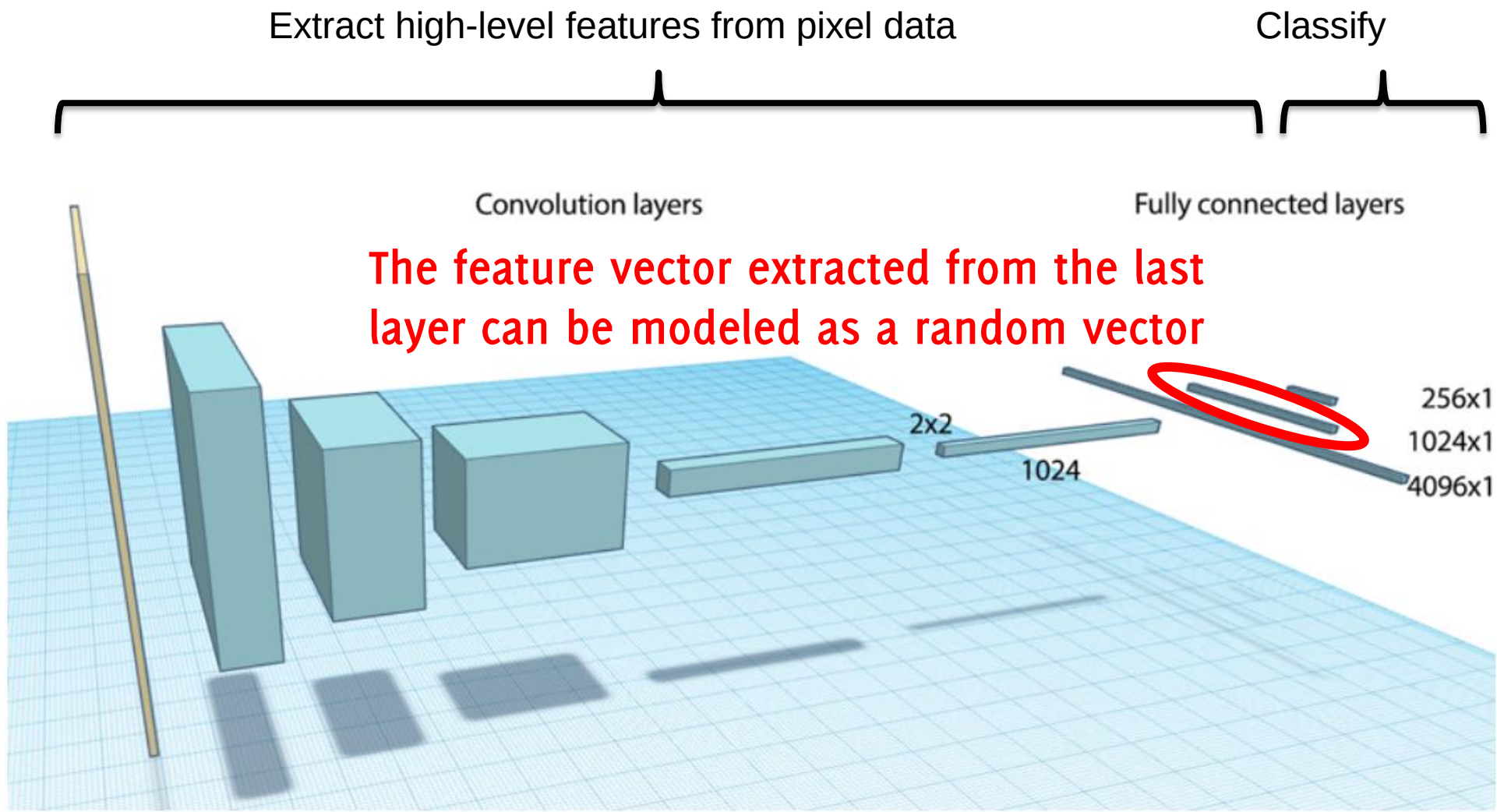
- CNN as data-driven feature extractor
  - Transfer learning
  - Self-supervised learning
  - Autoencoders
  - Domain-based
- Generative models

THE THREE STEP IN ANOMALY DETECTION IN IMAGES

Recall the three major ingredients

- Feature extraction
- Anomaly score
- Decision rule

CNN AS DATA-DRIVEN FEATURE EXTRACTOR



TRANSFER LEARNING

Idea:

- Use a pretrained network CNN (e.g. AlexNet), that was trained for a different task and on a different dataset
- Throw away the last layer(s)
- Use the reduced CNN ψ to build a new dataset TR' from TR:

$$TR' = \{\psi(\boldsymbol{s_i}),\ \boldsymbol{s_i} \in TR\}$$

- Train your favorite anomaly detector on TR' to define the **anomaly score** and the **decision rule**

Boracchi, Carrera, ICPR 2020

TRANSFER LEARNING

- Features extracted from a CNN, i.e., $\psi(\boldsymbol{s})$ is typically very large for deep networks (e.g. ResNET). **Reduce data-dimensionality** by PCA defined on a set of normal features
- **Anomalies** can be **detected by measuring distance w.r.t. normal features**, possibly using clustering to speed up performance.

Napoletano P., Piccoli F., Schettini R., *"Anomaly Detection in Nanofibrous Materials by CNN-Based Self-Similarity"*, Sensors 2018

TRANSFER LEARNING

Pros: pretrained networks are very powerful models, since they usually trained on datasets with million of images

Cons:

- the network is **not trained on normal** data. Meaningful structures in normal images might not be successfully captured by network trained on images from a different domain (e.g. medical vs natural images)
- The anomaly score and the CNN are not jointly learned, while the end-to-end learning strategy is the key to achieve impressive results in supervised tasks.

Boracchi, Carrera, ICPR 2020

SEMI-SUPERVISED APPROACHES

- CNN as data-driven feature extractor
  - Transfer learning
  - Self-supervised learning
  - Autoencoders
  - Domain-based
- Generative models

SELF-SUPERVISED LEARNING

With transfer learning we use a model trained on a **different dataset** (e.g., Imagenet) to address a **different task** (e.g., classification).

The idea of self-supervised learning is to train a model on the **target dataset** but solving a **different task,** for which we can easily obtain the labels.

SELF-SUPERVISED LEARNING

We can **build** a **labeled dataset** for multiclass **classification** from normal data

- Consider a set of T transformation $\mathcal{T} = \{\tau_1, \dots, \tau_T\}$
- Apply each transformation τ_i to every $\mathbf{s} \in TR$:

$$TR_{new} = \{(\tau_i(\boldsymbol{s}), i) \mid \boldsymbol{s} \in TR, i = 1, \dots, T\}$$

- Train a CNN on TR_{new}
- The output of the last layer of the CNN is used as feature vector

SELF-SUPERVISED LEARNING

We can **build** a **labeled dataset** for multiclass **classification** from normal data

- Consider a set of T transformation $\mathcal{T} = \{\tau_1, \dots, \tau_T\}$
- Apply each transformation τ_i to every $\mathbf{s} \in TR$:

$$TR_{new} = \{(\tau_i(\boldsymbol{s}), i) \mid \boldsymbol{s} \in TR, i = 1, \dots, T\}$$

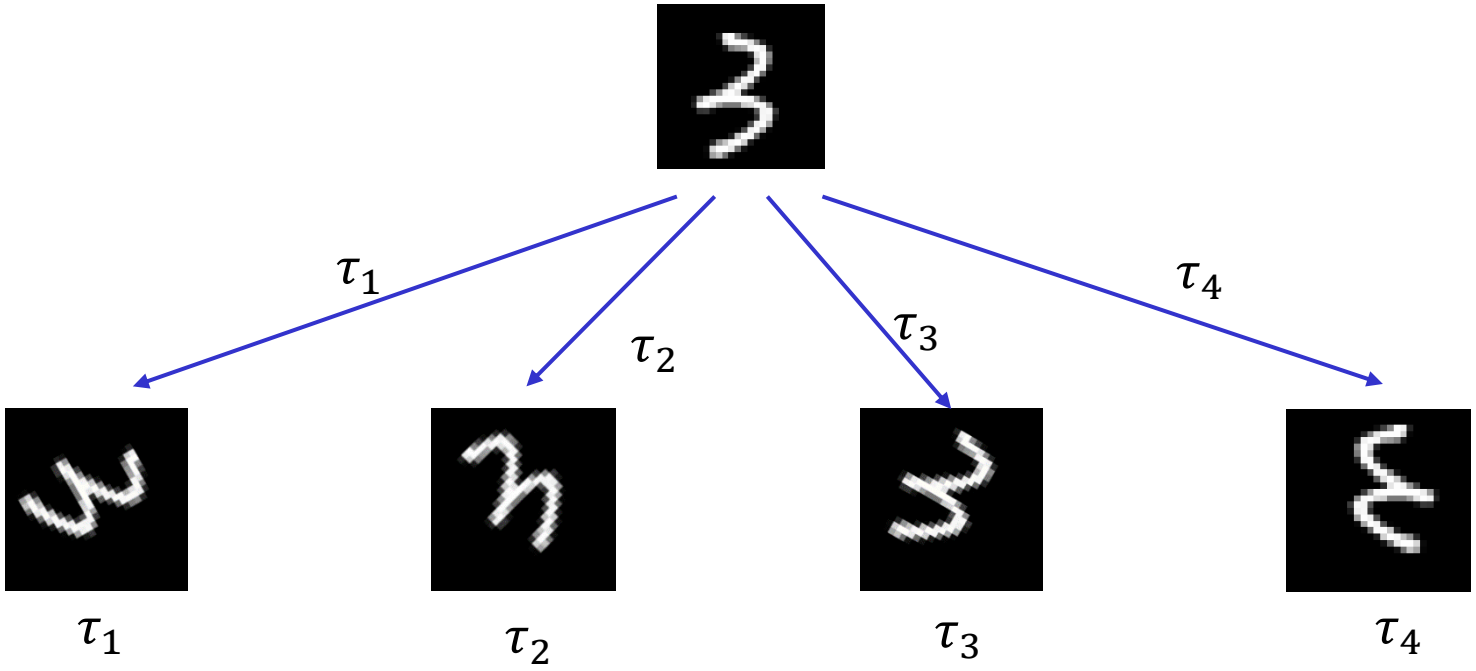
- Train a CNN on TR_{new}
- The output of the last layer of the CNN is used as feature vector

We can avoid using a pre-trained model, and train a CNN directly on our dataset

SELF-SUPERVISED LEARNING

Example:

- TR contains only images representing digit 3
- $\mathcal{T}$ contains rotations and horizontal/vertical flips

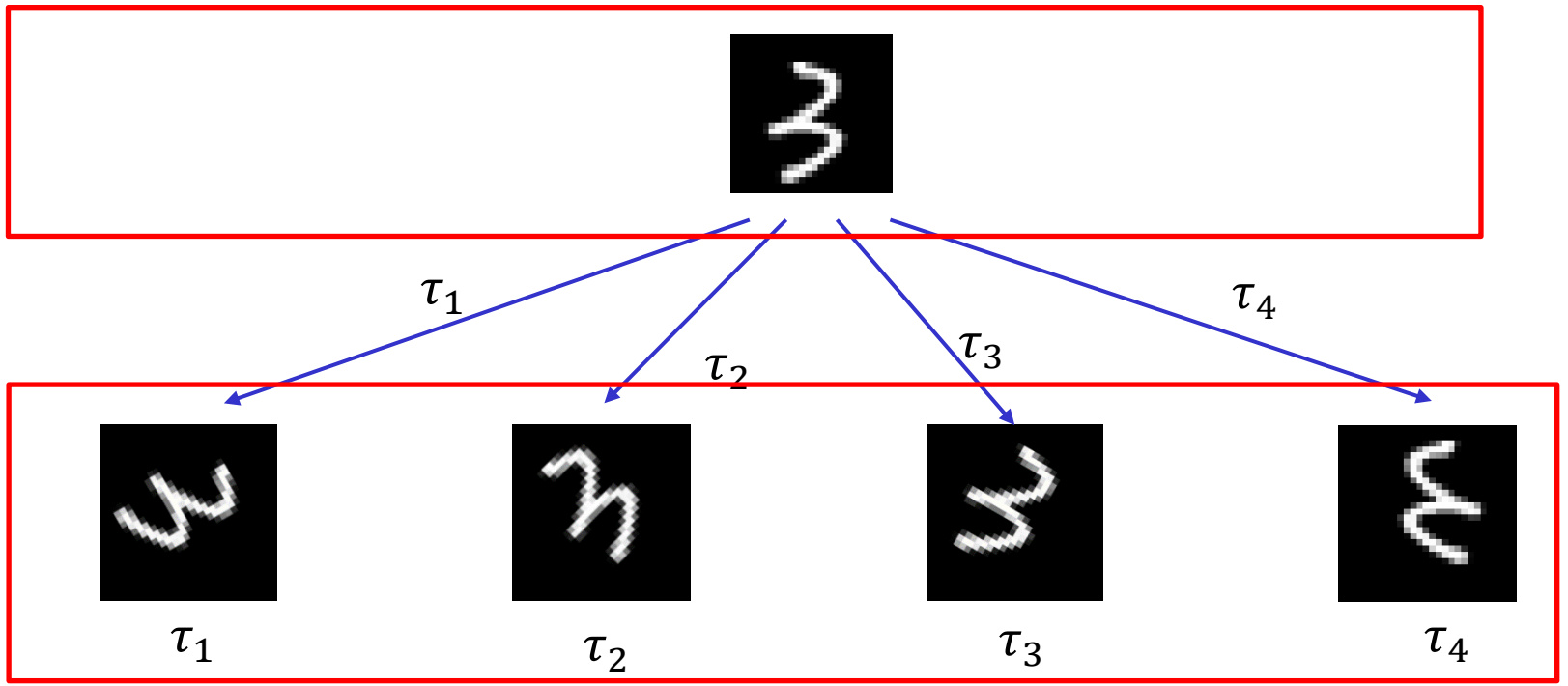


Boracchi, Carrera, ICPR 2020

SELF-SUPERVISED LEARNING

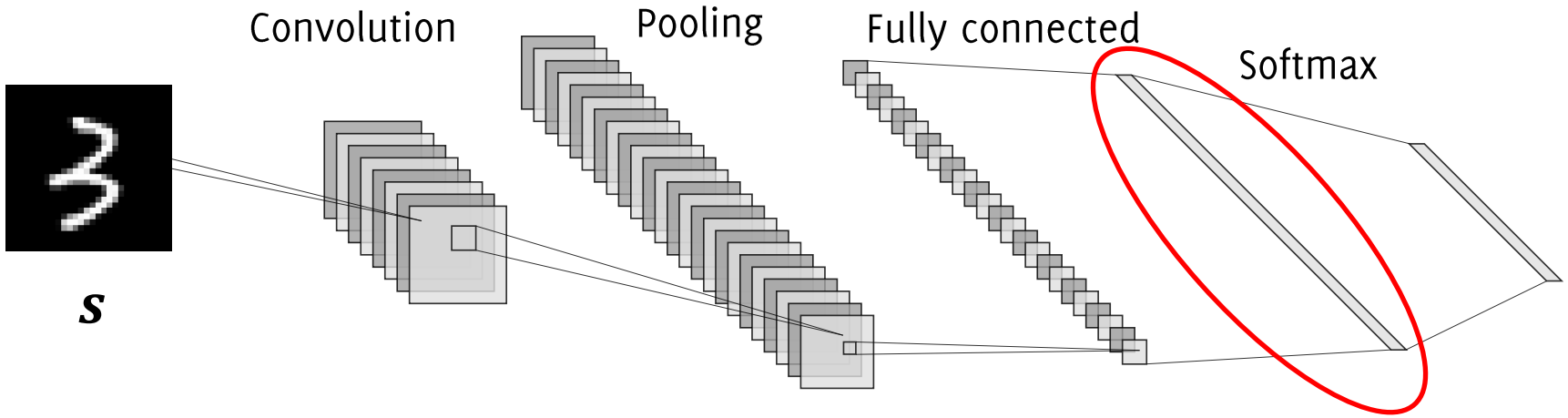
Example:

- TR contains only images representing digit 3
- $\mathcal{T}$ contains rotations and horizontal/vertical flips



Boracchi, Carrera, ICPR 2020

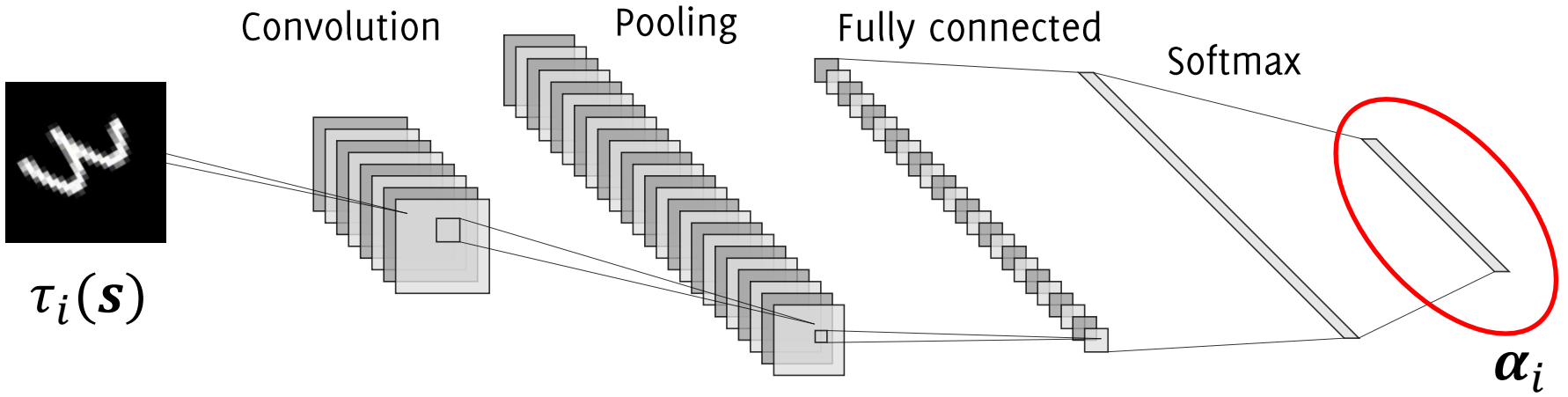
SELF-SUPERVISED LEARNING



We can use the second last layer as **feature vector** and train an anomaly detector

Problem: we have to split our training set in two sets. The first set is used to train the classifier, the second one to train the anomaly detector

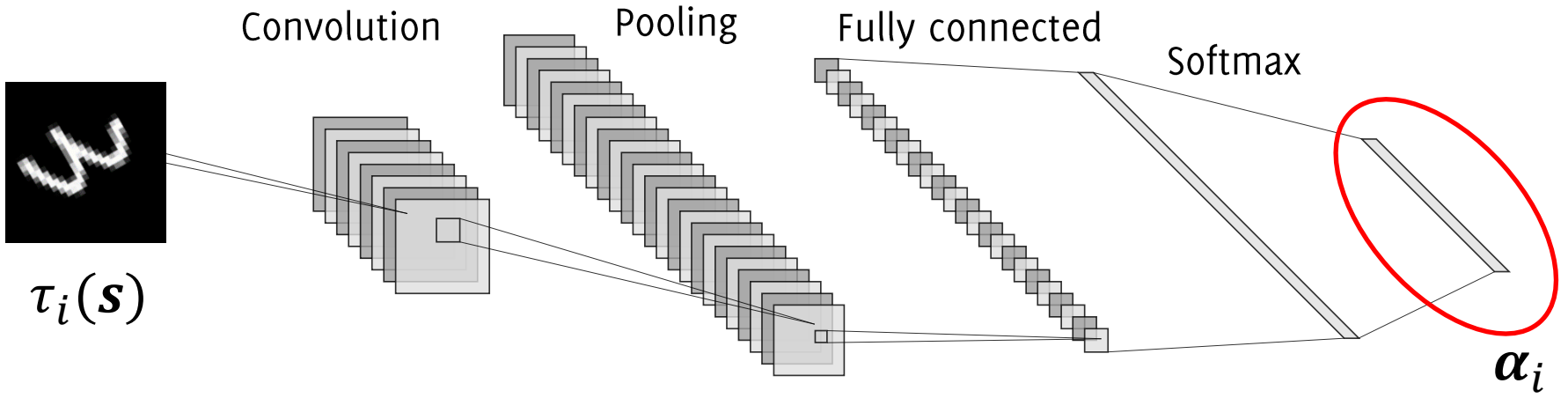
SELF-SUPERVISED LEARNING



Another approach is to compute feed the network with the all the trasformed versions $\tau_i(\boldsymbol{s})$ of the test sample $\boldsymbol{s}$ to obtain $\{\boldsymbol{\alpha}_i\}_{i=1}^{T}$

Golan, El-Yaniv, *"Deep Anomaly Detection Using Geometric Transformations"*, NeurIPS 2018

SELF-SUPERVISED LEARNING



Another approach is to compute feed the network with the all the trasformed versions $\tau_i(\boldsymbol{s})$ of the test sample $\boldsymbol{s}$ to obtain $\{\boldsymbol{\alpha}_i\}_{i=1}^{T}$

$$\mathcal{A}(\boldsymbol{s}) = 1 - \frac{1}{T}\sum_{i=1}^{T}[\boldsymbol{\alpha}_i]_i$$

Where $[\boldsymbol{\alpha}_i]_i$ is the posterior probability of label i given $\tau_i(\boldsymbol{s})$

Golan, El-Yaniv, *"Deep Anomaly Detection Using Geometric Transformations"*, NeurIPS 2018

SELF-SUPERVISED LEARNING

The set of transformation has to be properly chosen:

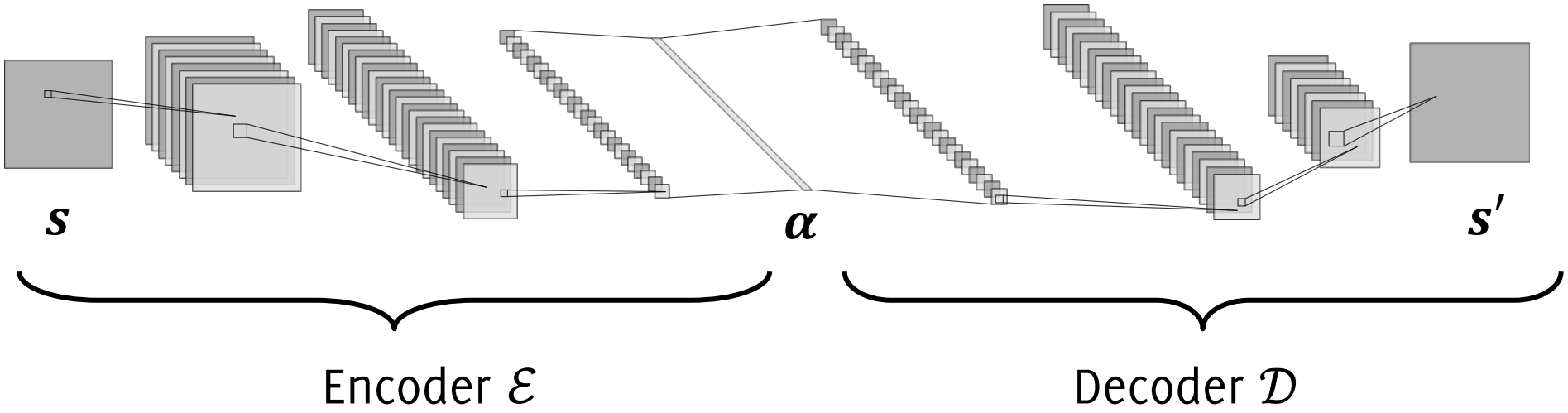
- if during training the trained **classifier cannot discriminate** the transformed samples, it **does not extract meaningful feature** for anomaly detection
- **Non-geometric transformations** (Gaussian blur, gamma correction, sharpening) might eliminate important features and **are less performing** than geometric ones

Golan, El-Yaniv, *"Deep Anomaly Detection Using Geometric Transformations"*, NeurIPS 2018

SEMI-SUPERVISED APPROACHES

- CNN as data-driven feature extractor
  - Transfer learning
  - Self-supervised learning
  - Autoencoders
  - Domain-based
- Generative models

AUTOENCODERS (REVISITED)



Autoencoders can be trained directly on normal data by minimizing the reconstruction loss:

$$\sum_{\boldsymbol{s} \in TR} \left\| \boldsymbol{s} - \mathcal{D}\left(\mathcal{E}(\boldsymbol{s})\right) \right\|_2$$

Boracchi, Carrera, ICPR 2020

AUTOENCODERS (REVISITED)

We expect that autoencoders trained only on normal data do not provide good reconstruction to anomalous data. Thus we can use the reconstruction error an anomaly score:

$$\mathrm{err}(\boldsymbol{s}) = \left\| \boldsymbol{s} - \mathcal{D}\left(\mathcal{E}(\boldsymbol{s})\right) \right\|_2$$

The **gradients** w.r.t. the autoencoder weights $\frac{\partial \mathrm{err}}{\partial \theta}$ turn to be very **discriminative** between normal and anomalous sample. This can be exploited in a more complex loss

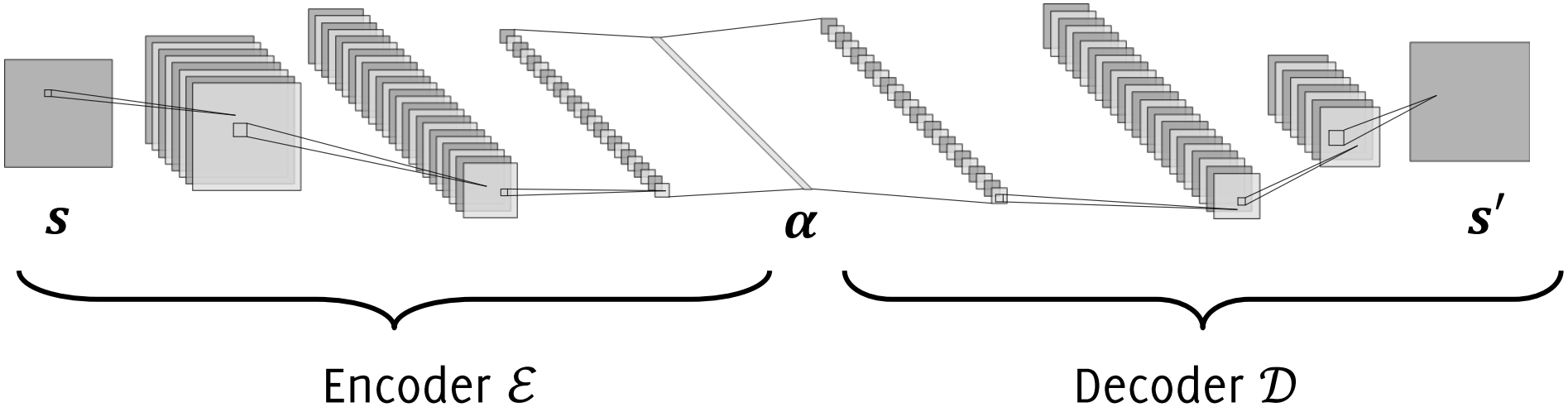
$$\mathcal{L}(\boldsymbol{s}) = \mathrm{err}(\boldsymbol{s}) + \mathcal{L}_{grad}(\boldsymbol{s})$$

Where $\mathcal{L}_{grad}$ is designed to enforce gradients computed on data in training set to be **aligned to each other**.

Then $\mathcal{L}(\boldsymbol{s})$ can be used as anomaly score

G. Kwon, M. Prabhushankar, D. Temel, and G. AIRegib, “Backpropagated Gradient Representations for Anomaly Detection”, ECCV 2020

AUTOENCODERS (REVISITED)



We can fit a **density model** (e.g. Gaussian Mixture) on $\boldsymbol{\alpha} = \mathcal{E}(\boldsymbol{s})$**:**

$$\boldsymbol{\alpha} \sim \sum_i \pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i},$$

Where $\varphi_{\boldsymbol{\mu}_i, \Sigma_i}$ is the pdf of $\mathcal{N}(\boldsymbol{\mu_i}, \Sigma_i)$

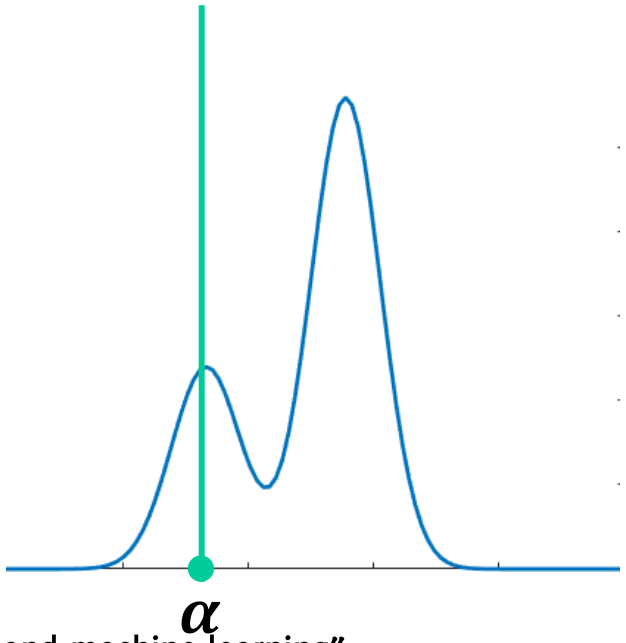
Boracchi, Carrera, ICPR 2020

EM-ALGORITHM FOR GAUSSIAN MIXTURES

Estimation of Gaussian Mixture parameters $\{\pi_i, \boldsymbol{\mu}_i, \Sigma_i\}$ from a training set $\{\boldsymbol{\alpha}_n\}_n$ is typically performed via EM-algorithm, that iterates the E and M steps

- **E-step:** compute the membership weights $\gamma_{n,i}$ for each training sample $\boldsymbol{\alpha}_n$

$$\gamma_{n,i} = \frac{\pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i}(\boldsymbol{\alpha}_n)}{\sum_k \pi_k \varphi_{\boldsymbol{\mu}_k, \Sigma_k}(\boldsymbol{\alpha}_n)}$$



$$\gamma_1 \sim 1$$
$$\gamma_2 \sim 0$$

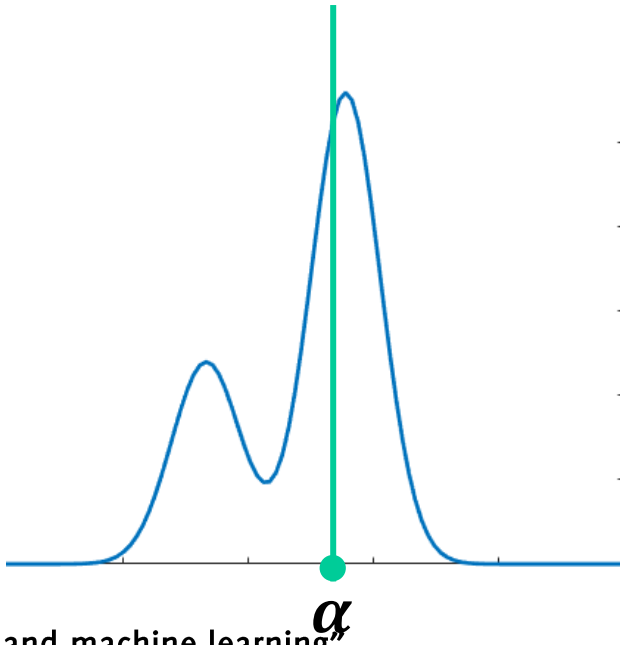
Bishop, "Pattern recognition and machine learning"

EM-ALGORITHM FOR GAUSSIAN MIXTURES

Estimation of Gaussian Mixture parameters $\{\pi_i, \boldsymbol{\mu}_i, \Sigma_i\}$ from a training set $\{\boldsymbol{\alpha}_n\}_n$ is typically performed via EM-algorithm, that iterates the E and M steps

- **E-step:** compute the membership weights $\gamma_{n,i}$ for each training sample $\boldsymbol{\alpha}_n$

$$\gamma_{n,i} = \frac{\pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i}(\boldsymbol{\alpha}_n)}{\sum_k \pi_k \varphi_{\boldsymbol{\mu}_k, \Sigma_k}(\boldsymbol{\alpha}_n)}$$



$$\gamma_1 \sim 0$$
$$\gamma_2 \sim 1$$

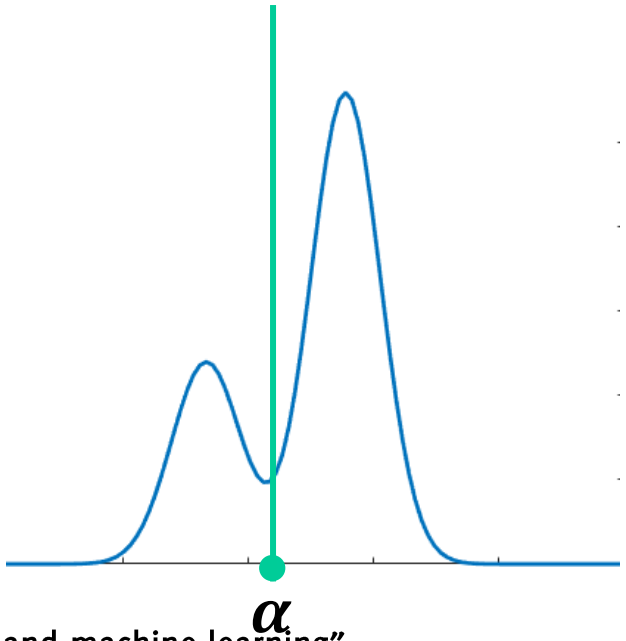
Bishop, "Pattern recognition and machine learning"

EM-ALGORITHM FOR GAUSSIAN MIXTURES

Estimation of Gaussian Mixture parameters $\{\pi_i, \boldsymbol{\mu}_i, \Sigma_i\}$ from a training set $\{\boldsymbol{\alpha}_n\}_n$ is typically performed via EM-algorithm, that iterates the E and M steps

- **E-step:** compute the membership weights $\gamma_{n,i}$ for each training sample $\boldsymbol{\alpha}_n$

$$\gamma_{n,i} = \frac{\pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i}(\boldsymbol{\alpha}_n)}{\sum_k \pi_k \varphi_{\boldsymbol{\mu}_k, \Sigma_k}(\boldsymbol{\alpha}_n)}$$



$$\gamma_1 \sim \frac{1}{2}$$
$$\gamma_2 \sim \frac{1}{2}$$

Bishop, "Pattern recognition and machine learning"

EM-ALGORITHM FOR GAUSSIAN MIXTURES

Estimation of Gaussian Mixture parameters $\{\pi_i, \boldsymbol{\mu}_i, \Sigma_i\}$ from a training set $\{\boldsymbol{\alpha}_n\}_n$is typically performed via EM-algorithm, that iterates the E and M steps

- **E-step:** compute the membership weights $\gamma_{n,i}$ for each training sample $\boldsymbol{\alpha}_n$

$$\gamma_{n,i} = \frac{\pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i}(\boldsymbol{\alpha}_n)}{\sum_k \pi_k \varphi_{\boldsymbol{\mu}_k, \Sigma_k}(\boldsymbol{\alpha}_n)}$$

- **M-step:** update the parameters of the Gaussian Mixture

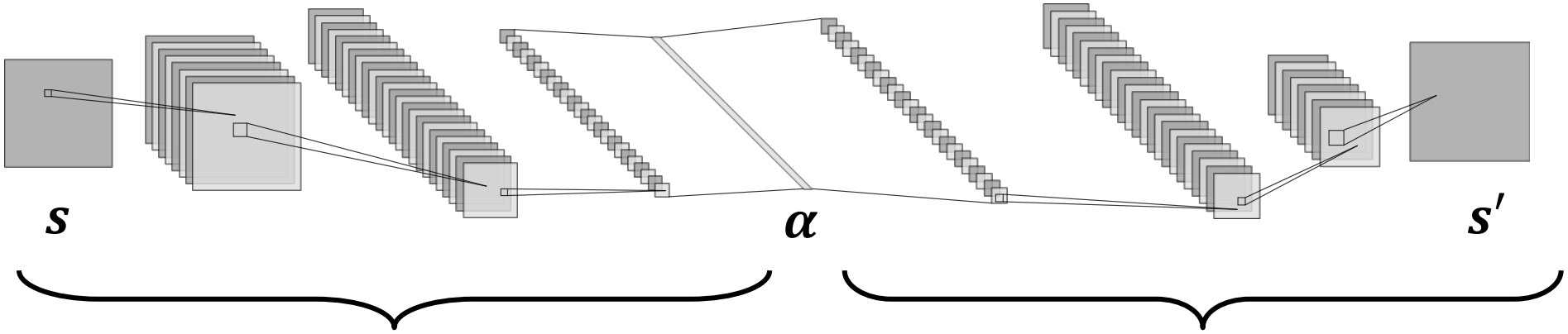
$$\pi_i = \frac{1}{N} \sum_n \gamma_{n,i}$$

$$\mu_i = \frac{\sum_n \gamma_{n,i} \boldsymbol{\alpha}_n}{\Sigma_n \gamma_{n,i}}$$

$$\Sigma_i = \frac{\sum_n \gamma_{n,i} (\boldsymbol{\alpha}_n - \boldsymbol{\mu}_i)(\boldsymbol{\alpha}_n - \boldsymbol{\mu}_i)^T}{\Sigma_n \gamma_{n,i}}$$

Bishop, “Pattern recognition and machine learning”

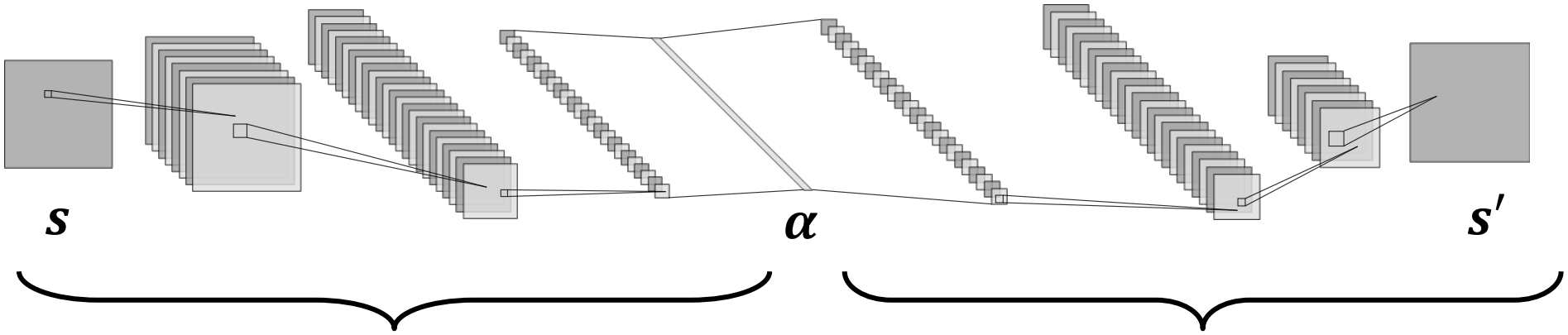
AUTOENCODERS (REVISITED)



We can compute the likelihood of a test sample $\boldsymbol{s}$ as:

$$\mathcal{L}(\boldsymbol{s}) = \sum_i \pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i}(\mathcal{E}(\boldsymbol{s}))\,,$$

AUTOENCODERS (REVISITED)



We can compute the likelihood of a test sample $\boldsymbol{s}$ as:

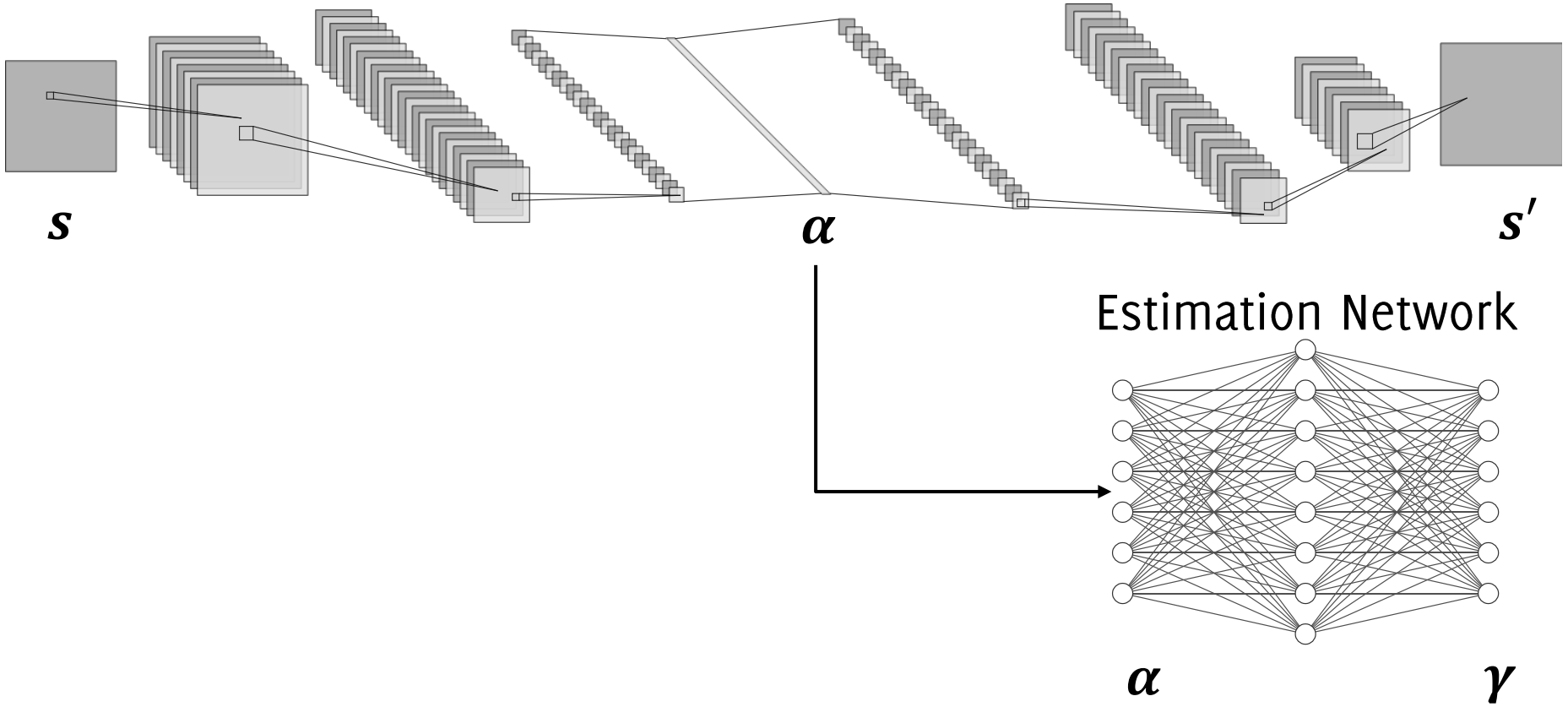
$$\mathcal{L}(\boldsymbol{s}) = \sum_i \pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i}(\mathcal{E}(\boldsymbol{s}))\,,$$

The autoencoder and the Gaussian Mixture are not jointly learned!

Boracchi, Carrera, ICPR 2020

JOINT LEARNING OF AUTOENCODER AND DENSITY MODEL

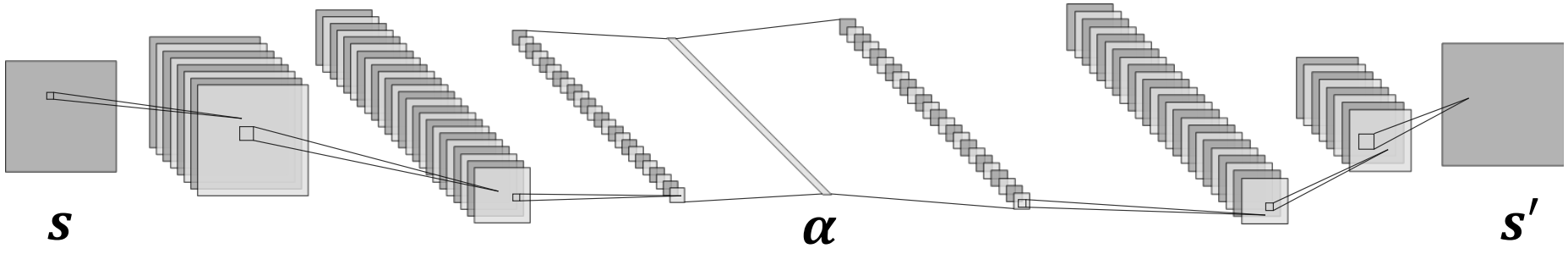
Idea: given a training set of N samples use a NN to predict the membership weights of each sample



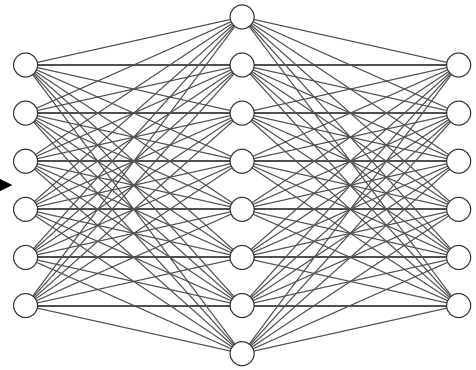
Zong et al, "Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection", *ICLR 2018*

JOINT LEARNING OF AUTOENCODER AND DENSITY MODEL

Idea: given a training set of N samples use a NN to predict the membership weights of each sample



Estimation Network



E-step

Estimate the GM parameters as:

$$\pi_i = \frac{1}{N}\sum_n \gamma_{n,i}$$

$$\mu_i = \frac{\sum_n \gamma_{n,i}\alpha_n}{\sum_n \gamma_{n,i}}$$

$$\Sigma_i = \frac{\sum_n \gamma_{n,i}(\alpha_n - \mu_i)(\alpha_n - \mu_i)^T}{\sum_n \gamma_{n,i}}$$

M-step

Zong et al, "Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection", *ICLR 2018*

DEEP AUTOENCODING GAUSSIAN MIXTURE MODEL

Minimize the loss:

$$\min \sum_{\boldsymbol{s}} \left|\left|\boldsymbol{s} - \mathcal{D}\big(\mathcal{E}(\boldsymbol{s})\big)\right|\right|_2^2 + \lambda \mathcal{R}(\mathcal{E}(\boldsymbol{s}))$$

Where

$$\mathcal{R}(\boldsymbol{\alpha}) = -\log \sum_i \pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i}(\boldsymbol{\alpha})$$

Additional regularizations has to be imposed on Σ_i to avoid trivial solution

$\mathcal{R}(\mathcal{E}(\boldsymbol{s}))$ can be used an anomaly score for a sample $\boldsymbol{s}$

Zong et al, "Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection", *ICLR 2018*

SOME REMARKS

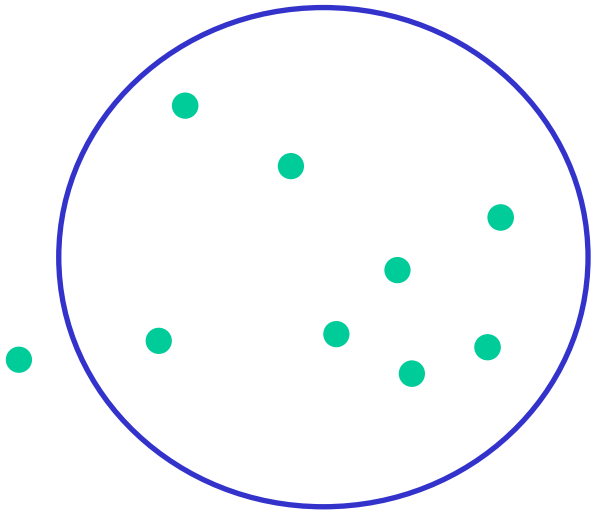
- The **estimation network** introduces a **regularization** that helps to avoid local optima of the recontruction error
- The autoencoder is then able to extract meaningful feature from normal data
- Density estimation enables anomaly detection, but it is a **more complicated** task

SEMI-SUPERVISED APPROACHES

- CNN as data-driven feature extractor
  - Transfer learning
  - Self-supervised learning
  - Autoencoders
  - Domain-based
- Generative models

SUPPORT VECTOR DATA DESCRIPTION (SVDD) REVISITED

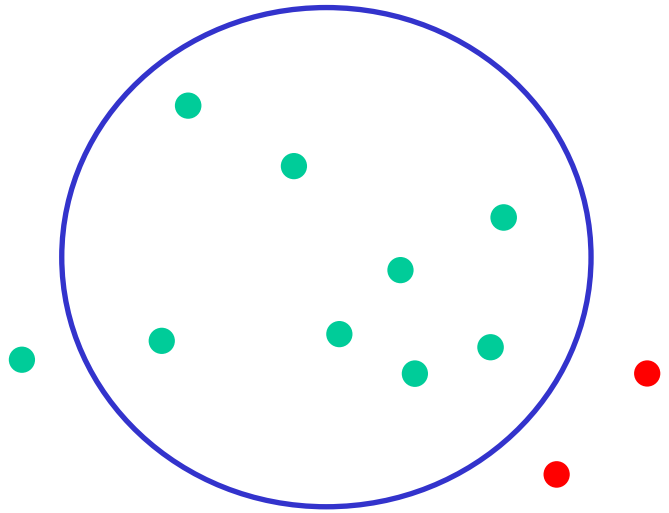
We want to find an **hypersphere** that, in the feature space, **encloses most of the normal** data



Tax, D. M., Duin, R. P. "*Support vector domain description*". Pattern recognition letters, 20(11), 1191-1199 (1999)

SUPPORT VECTOR DATA DESCRIPTION (SVDD) REVISITED

We want to find an **hypersphere** that, in the feature space, **encloses most of the normal** data

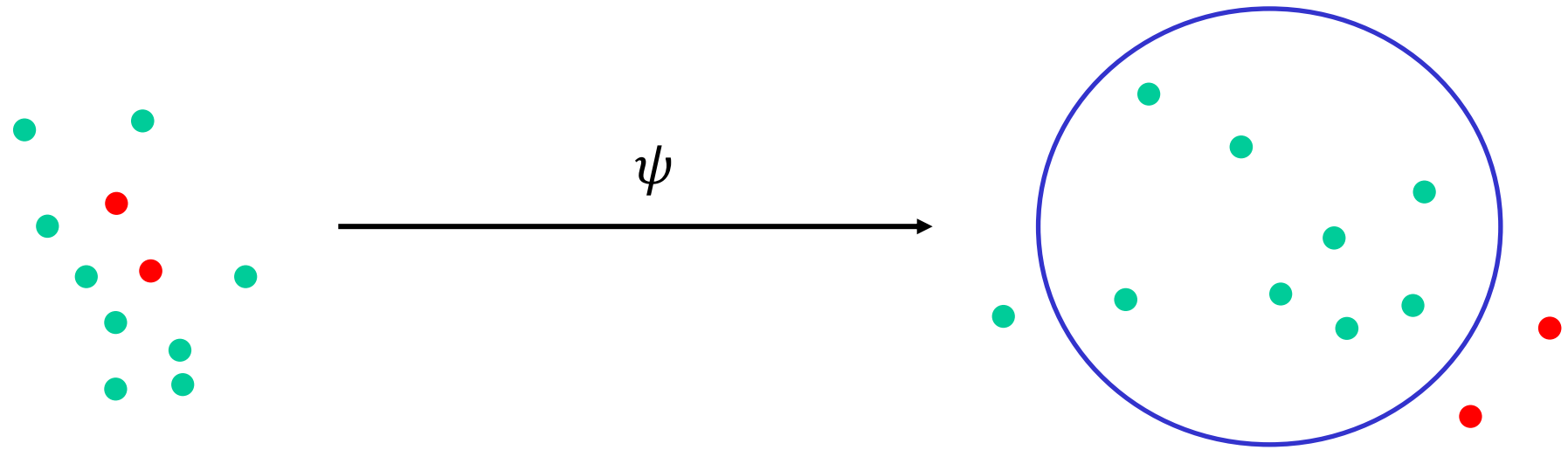


We expect that anomalous data lie outside the sphere

Tax, D. M., Duin, R. P. "*Support vector domain description*". Pattern recognition letters, 20(11), 1191-1199 (1999)

SUPPORT VECTOR DATA DESCRIPTION (SVDD) REVISITED

Typically the sphere is computed in a **high** (possibly infinite) **dimensional** feature **space**

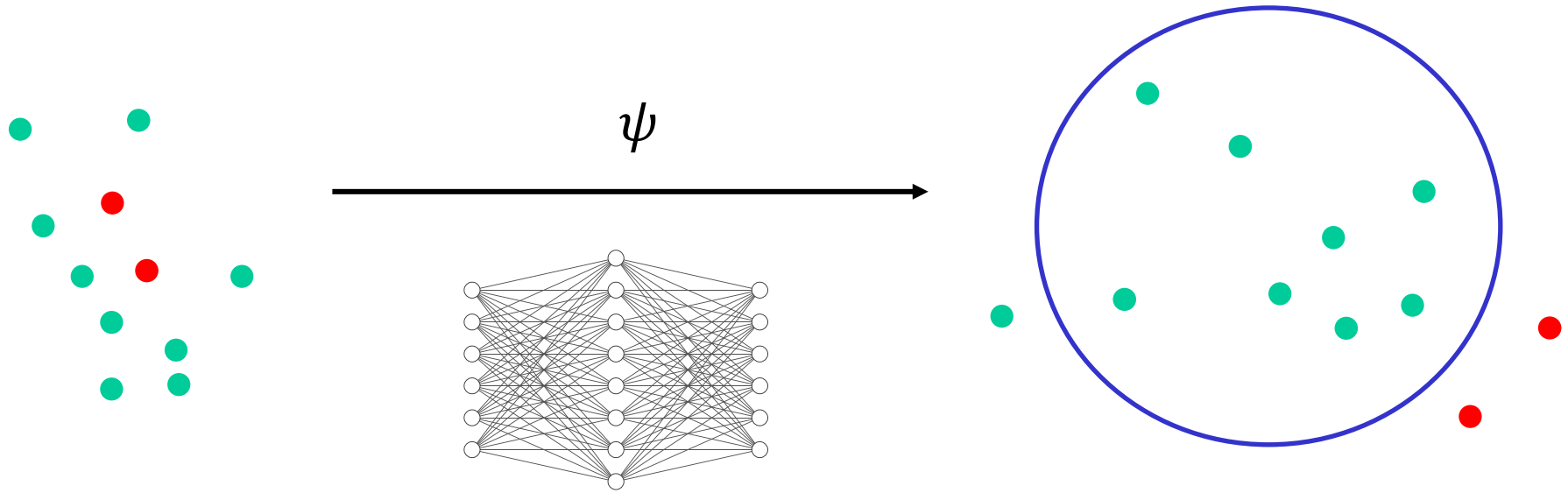


Feature are defined using **kernels**

- Polinomial kernel
- Gaussian kernel

SUPPORT VECTOR DATA DESCRIPTION (SVDD) REVISITED

Idea: can we learn the feature from normal data using a neural network?



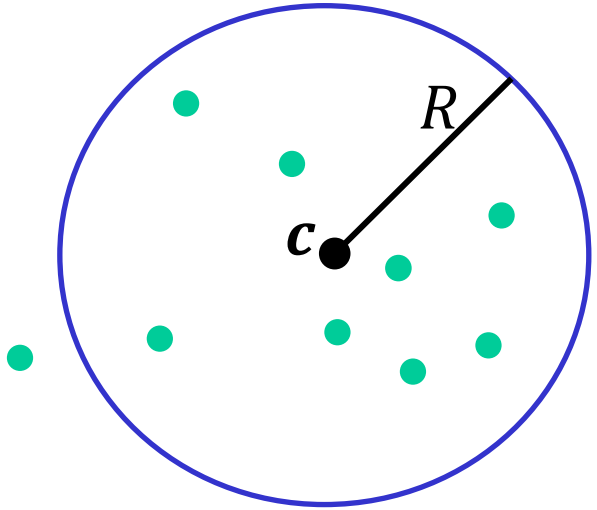
Ruff et al, *"Deep One-Class Classification", ICML 2018*

SOFT-BOUNDARY DEEP SVDD

Minimize the loss:

$$\min_{R,\boldsymbol{\theta}} R^2 + \frac{1}{\nu N}\sum_{n=1}^{N} \max\{0, \left|\left|\psi_{\boldsymbol{\theta}}(\boldsymbol{s}_n) - \boldsymbol{c}\right|\right|^2 - R^2\} + \lambda\left|\left|\boldsymbol{\theta}\right|\right|^2$$

- The samples $\boldsymbol{s}_n$ such that $\psi_{\boldsymbol{\theta}}(\boldsymbol{s}_n)$ is inside the sphere do not contribute to the loss
- ν provides a bound on the False Positive Rate
- $\lambda\left|\left|\boldsymbol{\theta}\right|\right|^2$ is a regularization term



A test sample $\boldsymbol{s}$ is anomalous if $\psi_{\boldsymbol{\theta}}(\boldsymbol{s}) - \boldsymbol{c} > R$

Ruff et al, *"Deep One-Class Classification", ICML 2018*

SOFT-BOUNDARY DEEP SVDD

Minimize the loss:

$$\min_{R,\boldsymbol{\theta}} R^2 + \frac{1}{\nu N}\sum_{n=1}^{N} \max\{0, \left|\left|\psi_{\boldsymbol{\theta}}(\boldsymbol{s}_n) - \boldsymbol{c}\right|\right|^2 - R^2\} + \lambda\left|\left|\boldsymbol{\theta}\right|\right|^2$$

Remarks:

- Some contraints must be imposed on the network $\psi_{\boldsymbol{\theta}}$ to avoid trivial solutions:
  - **No bias terms**
  - **Unbounded** activations
- $\boldsymbol{c}$ is not optimized but has to be precomputed from data
  - $\boldsymbol{c}$ must be different from $\boldsymbol{c}_0 = \psi_0(\boldsymbol{s})$

Ruff et al, *"Deep One-Class Classification", ICML 2018*

A SIMPLER FORMULATION: DEEP SVDD

$$\min_{\boldsymbol{\theta}} + \frac{1}{N}\sum_{n=1}^{N}\left|\left|\psi_{\boldsymbol{\theta}}(\boldsymbol{s}_n) - \boldsymbol{c}\right|\right|^2 + \lambda\left|\left|\boldsymbol{\theta}\right|\right|^2$$

Cons:

- No bound on the FPR provided by ν
- A threshold has to be chosen for the anomaly score:

$$\mathcal{A}(\boldsymbol{s}) = \left|\left|\psi_{\boldsymbol{\theta}}(\boldsymbol{s}) - \boldsymbol{c}\right|\right|^2$$

Ruff et al, *"Deep One-Class Classification", ICML 2018*

SEMI-SUPERVISED APPROACHES

- CNN as data-driven feature extractor
  - Transfer learning
  - Self-supervised learning
  - Autoencoders
  - Domain-based
- Generative models

GENERATIVE MODELS

Goal:

generative models generate, given a training set of images (data) S, other images (data) that are similar to those in S

WHAT FOR GENERATIVE MODELS?

- Generative models can be used for data augmentation, simulation and planning
- Realistic samples for artwork, super-resolution, colorization, etc.
- You are getting close to the “holy grail” of modeling the distribution of natural images

Radford, et al “Unsupervised representation learning with deep convolutional generative adversarial networks. ICLR 2016

GENERATIVE ADVERSARIAL NETWORKS (GAN)

The GAN approach:

Do not look for an **explicit density model** ϕ_S describing the manifold of natural images.

Just find out a **model** able to **generate samples** that looks like training samples $S \subset \mathbb{R}^n$

Instead of sampling from ϕ_S, just use:

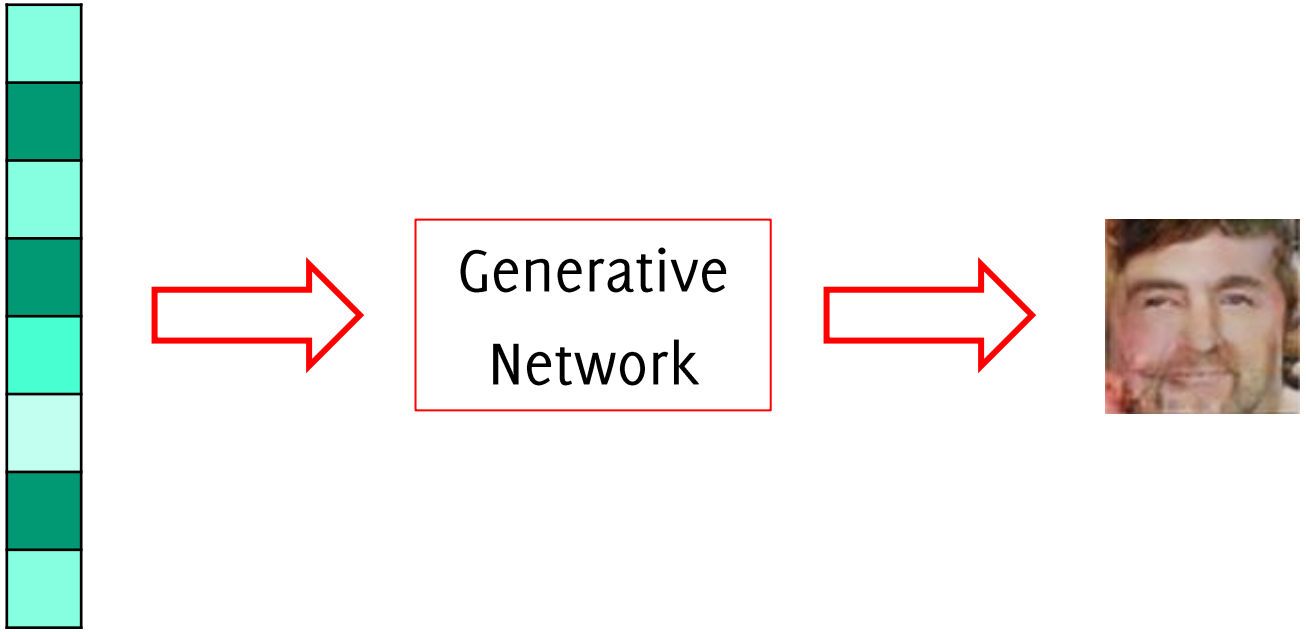
- Sample a seed from a known distribution ϕ_z
- Feed this seed to a learned transformation that generates realistic samples, as if they were drawn from ϕ_S

Use a **neural network to learn this transformation**

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

GENERATIVE ADVERSARIAL NETWORKS (GAN)

The GAN approach:



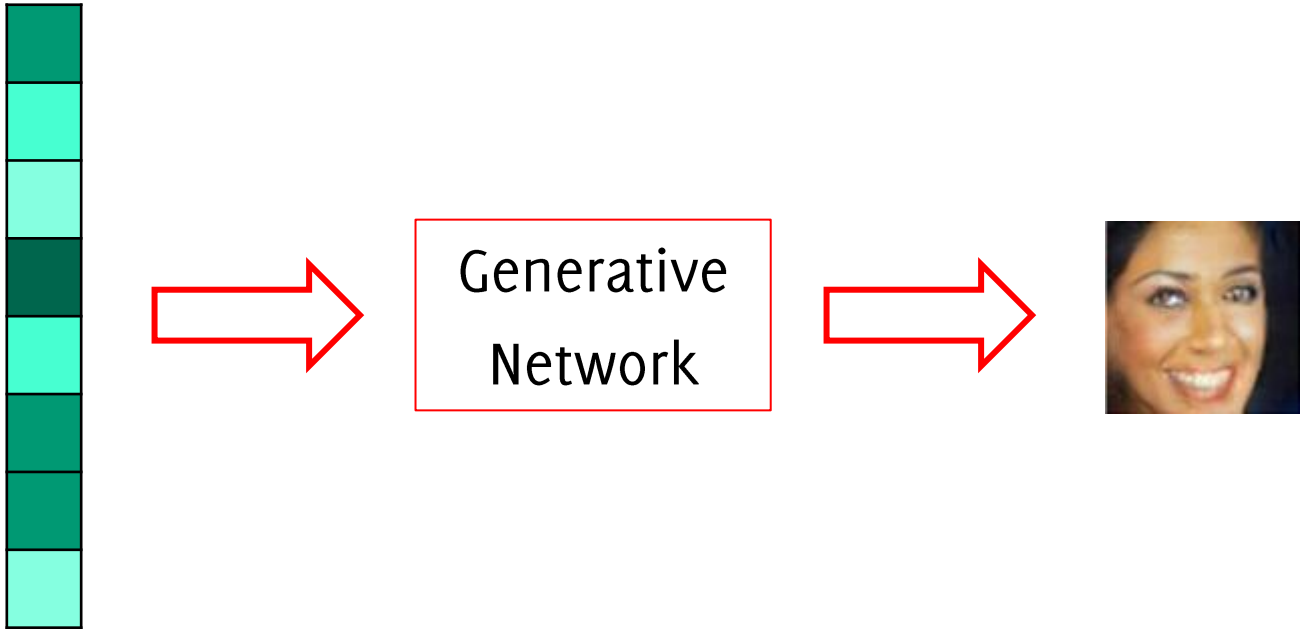
$z \sim \phi_z$

Draw a sample from
the noise distribution

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

GENERATIVE ADVERSARIAL NETWORKS (GAN)

The GAN approach:



$z \sim \phi_z$

Draw a sample from
the noise distribution

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

GENERATIVE ADVERSARIAL NETWORKS (GAN)

The GAN solution: Train a pair of neural networks with different tasks that compete in a sort of **two player game**.

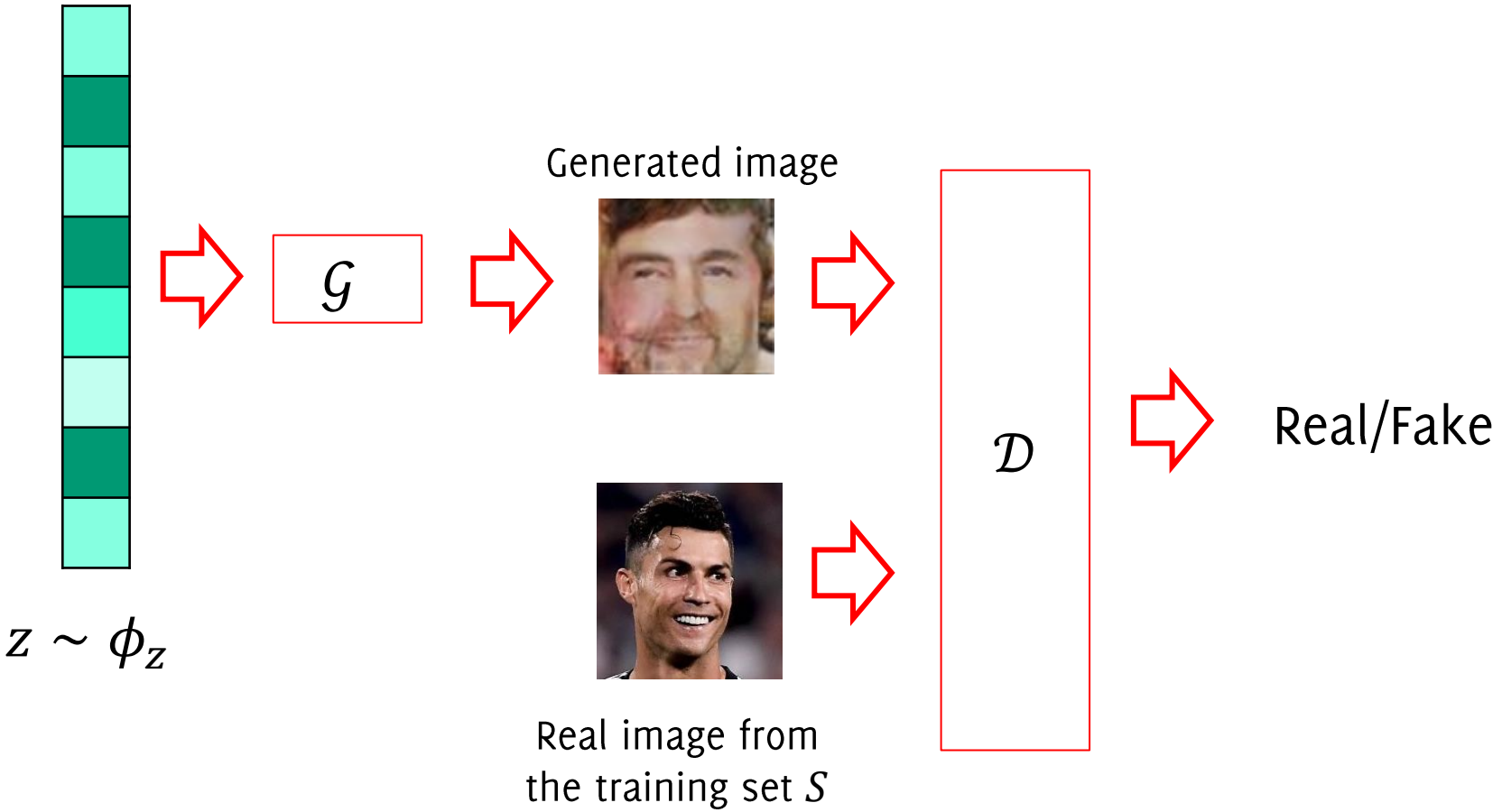
These models are:

- Generator $\mathcal{G}$ that produces realistic samples e.g. taking as input some random noise. $\mathcal{G}$ tries to fool the discriminator
- Discriminator $\mathcal{D}$ that takes as input an image and assess whether it is real or generated by $\mathcal{G}$

Train the two and at the end, keep only $\mathcal{G}$

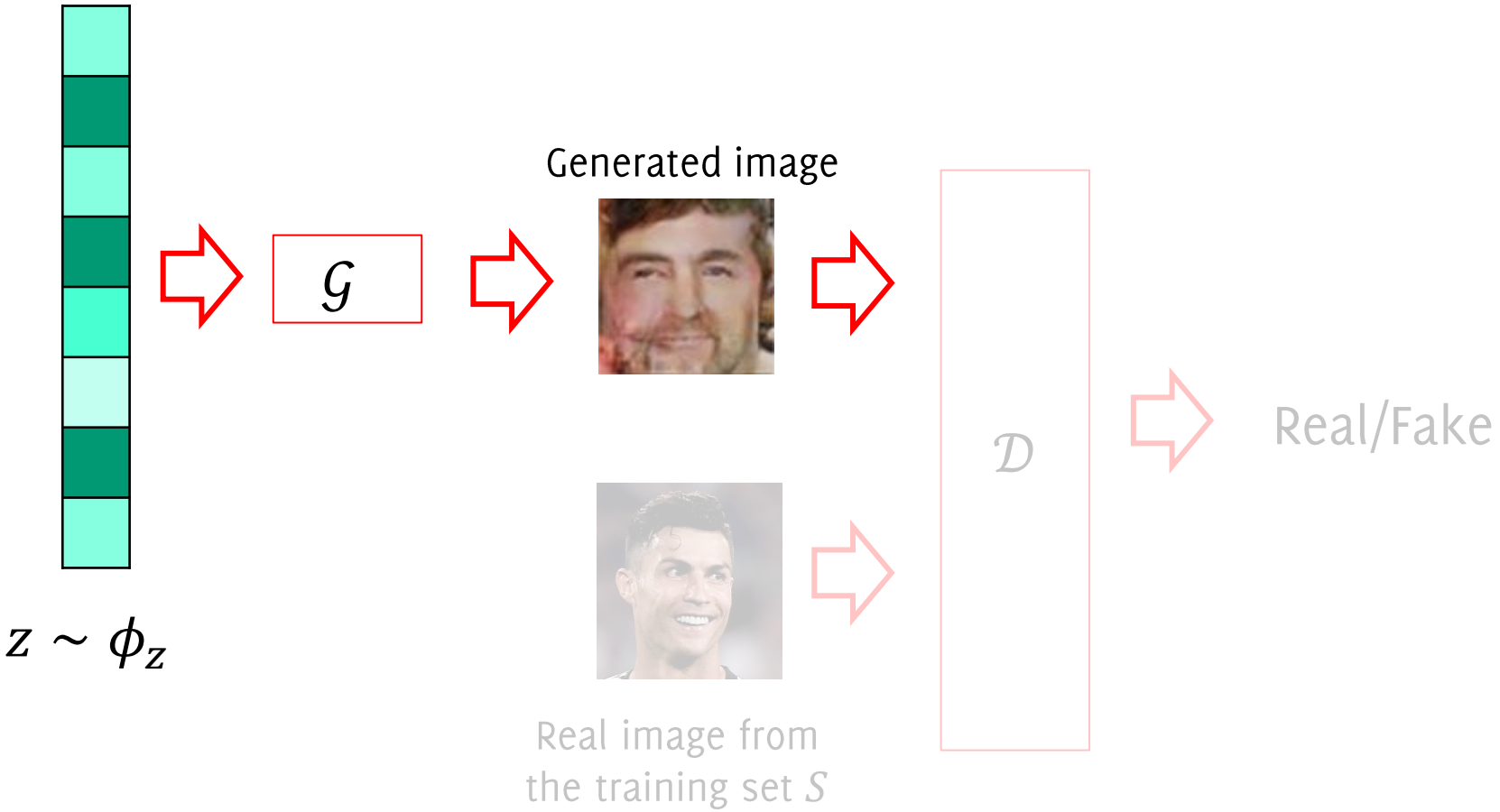
Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

GAN ARCHITECTURE



Boracchi, Carrera, ICPR 2020

USING GAN



Discriminator $\mathcal{D}$ is completely useless and as such dropped. After a successful GAN training, $\mathcal{D}$ is not able to distinguish the real/fake

GAN

Both $\mathcal{D}$ and $\mathcal{G}$ are conveniently chosen as Neural Networks

Setting up the stage

Our networks take as input:

- $\mathcal{D} = \mathcal{D}(\boldsymbol{s})$
- $\mathcal{G} = \mathcal{G}(\boldsymbol{z})$

$\boldsymbol{s} \in \mathbb{R}^n$ is an input image (either real or generated by $\mathcal{G}$) and $\boldsymbol{z} \in \mathbb{R}^d$ is some random noise to be fed to the generator.

Our network give as output:

$$\mathcal{D}(\cdot): \mathbb{R}^n \to [0,1]$$

the posteriori for an input to be a true image (1)

$$\mathcal{G}(\cdot): \mathbb{R}^d \to \mathbb{R}^n$$

the generated image

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

GAN TRAINING

A good discriminator is such:

- $\mathcal{D}(\boldsymbol{s})$ is maximum when $s \in S$
- $1 - \mathcal{D}(\boldsymbol{s})$ is maximum when s was generated from $\mathcal{G}$
- $1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big)$ is maximum when $\boldsymbol{z} \sim \phi_Z$

Training $\mathcal{D}$ consists in maximizing the binary cross-entroy

$$\max_{\mathcal{D}} \big(\mathrm{E}_{s \sim \phi_S} [\log \mathcal{D}(\boldsymbol{s})] + \mathrm{E}_{z \sim \phi_Z} \big[\log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big)) \big] \big)$$

GAN TRAINING

A good discriminator is such:

- $\mathcal{D}(\boldsymbol{s})$ is maximum when $s \in S$
- $1 - \mathcal{D}(\boldsymbol{s})$ is maximum when s was generated from $\mathcal{G}$
- $1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big)$ is maximum when $\boldsymbol{z} \sim \phi_Z$

Training $\mathcal{D}$ consists in maximizing the binary cross-entroy

$$\max_{\mathcal{D}}\big(\mathrm{E}_{s\sim\phi_S}[\log \mathcal{D}(\boldsymbol{s})] + \mathrm{E}_{z\sim\phi_Z}\big[\log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))\big]\big)$$

This has to be 1 since $s \sim \phi_S$, thus images are real

This has to be 0 since $\mathcal{G}(\boldsymbol{z})$ is a generated (fake) image

GAN TRAINING

A good discriminator is such:

- $\mathcal{D}(\boldsymbol{s})$ is maximum when $s \in S$
- $1 - \mathcal{D}(\boldsymbol{s})$ is maximum when s was generated from $\mathcal{G}$
- $1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big)$ is maximum when $\boldsymbol{z} \sim \phi_Z$

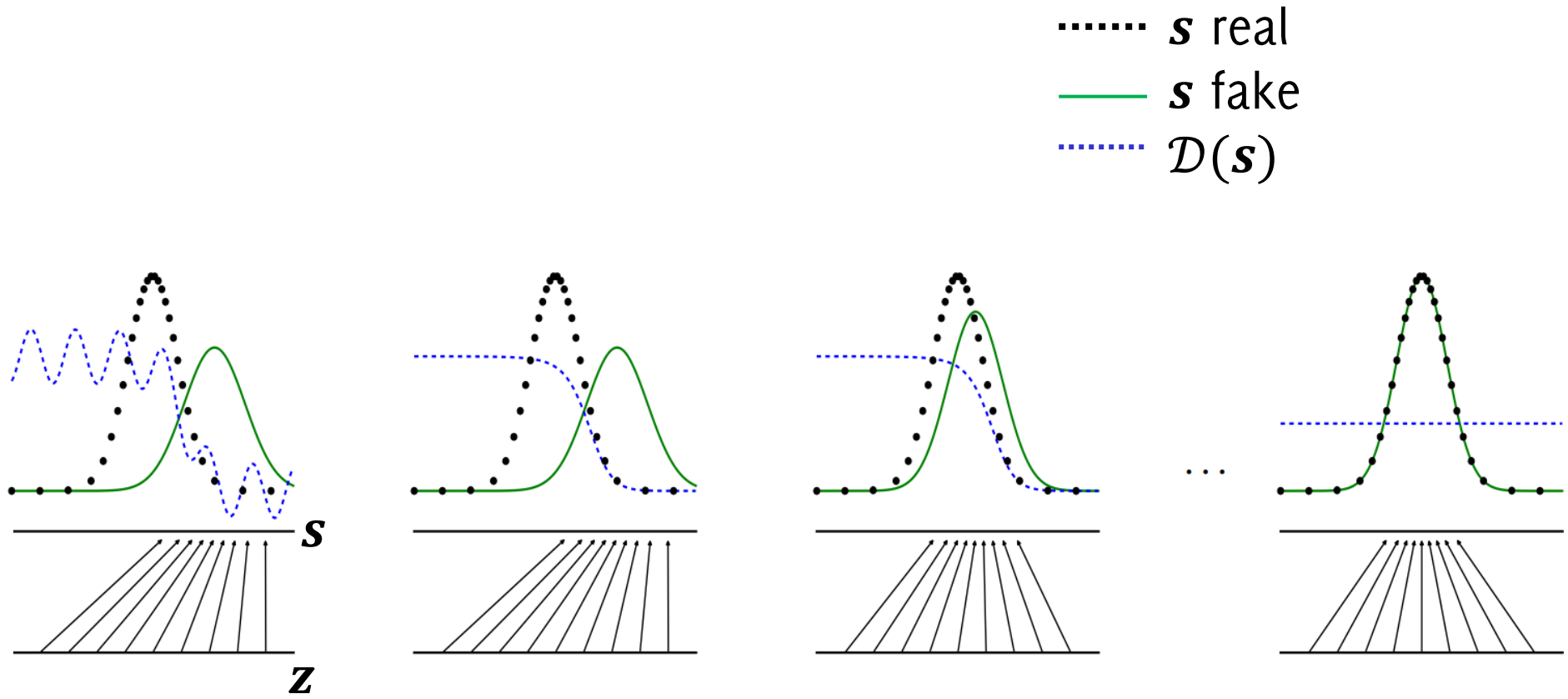
Training $\mathcal{D}$ consists in maximizing the binary cross-entroy

$$\max_{\mathcal{D}}\big(\mathrm{E}_{s\sim\phi_S}[\log \mathcal{D}(\boldsymbol{s})] + \mathrm{E}_{z\sim\phi_Z}\big[\log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))\big]\big)$$

A good generator $\mathcal{G}$ is the one which makes $\mathcal{D}$ to fail

$$\min_{\mathcal{G}}\max_{\mathcal{D}}\big(\mathrm{E}_{s\sim\phi_S}[\log \mathcal{D}(\boldsymbol{s})] + \mathrm{E}_{z\sim\phi_Z}\big[\log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))\big]\big)$$

ILLUSTRATION



Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

MNIST

Generated samples

nearest sample to the second-last column

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

INTERPOLATION IN THE LATENT SPACE

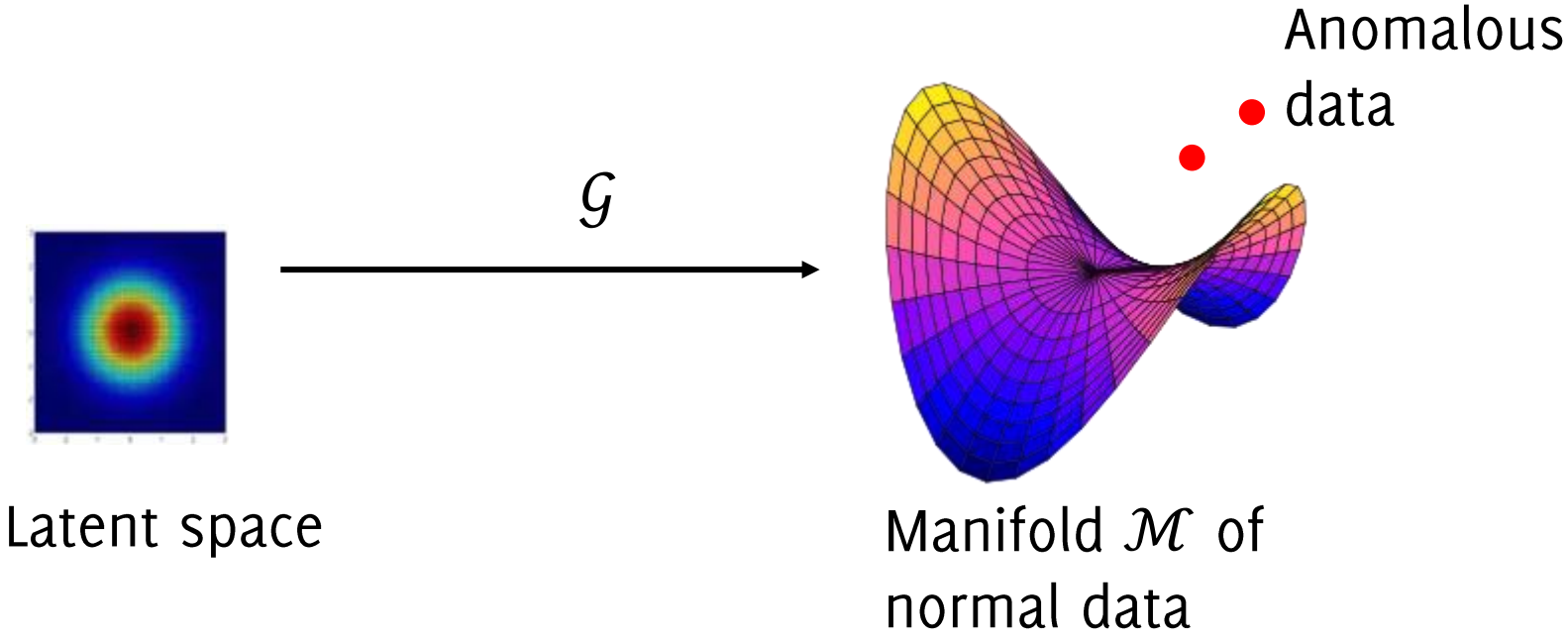
We can interpolate between two points in the latent space and obtain smooth transitions from a digit to another one

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

GAN FOR ANOMALY DETECTION

Idea: let us train a GAN on normal data. We expect that the generator $\mathcal{G}$ cannot generate any anomalous sample $\boldsymbol{s}$.

Problem: Given a test sample $\boldsymbol{s}$ how can we determine if it could be generated by $\mathcal{G}$?



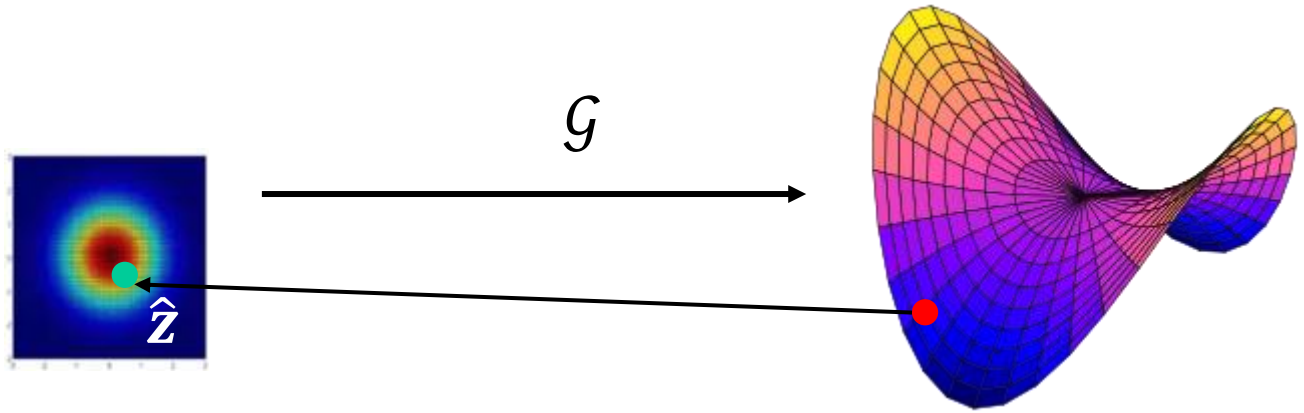
Boracchi, Carrera, ICPR 2020

ANOGAN

Project the test sample $\boldsymbol{s}$ on the manifold $\mathcal{M}$ by solving the optimization problem:

$$\hat{\boldsymbol{z}} = \min_{\boldsymbol{z}} \left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right| + \lambda \log(1 - \mathcal{D}(\mathcal{G}(\boldsymbol{z})))$$

- $\left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right|$ ensures that $\boldsymbol{s}$ is well approximated by the generator
- $\log(1 - \mathcal{D}(\mathcal{G}(\boldsymbol{z})))$ ensures that the projection $\mathcal{G}(\hat{\boldsymbol{z}})$ is similar to a real (normal) sample



Boracchi, Carrera, ICPR 2020

ANOGAN

Project the test sample $\boldsymbol{s}$ on the manifold $\mathcal{M}$ by solving the optimization problem:

$$\hat{\boldsymbol{z}} = \min_{\boldsymbol{z}} \left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right| + \lambda \log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))$$

- $\left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right|$ ensures that $\boldsymbol{s}$ is well approximated by the generator
- $\log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))$ ensures that the projection $\mathcal{G}(\hat{\boldsymbol{z}})$ is similar to a real (normal) sample (since $\mathcal{G}$ fools $\mathcal{D}$)

Anomaly score:

$$\mathcal{A}(\boldsymbol{s}) = \left|\left|\mathcal{G}(\hat{\boldsymbol{z}}) - \boldsymbol{s}\right|\right| + \lambda \log(1 - \mathcal{D}\big(\mathcal{G}(\hat{\boldsymbol{z}})\big))$$

Schlegl et al, "*Unsupervised anomaly detection with generative adversarial networks to guide marker discovery*", IPMI 2017

ANOGAN

Project the test sample $\boldsymbol{s}$ on the manifold $\mathcal{M}$ by solving the optimization problem:

$$\hat{\boldsymbol{z}} = \min_{\boldsymbol{z}} \left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right| + \lambda \log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))$$

- $\left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right|$ ensures that $\boldsymbol{s}$ is well approximated by the generator
- $\log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))$ ensures that the projection $\mathcal{G}(\hat{\boldsymbol{z}})$ is similar to a real (normal) sample (since $\mathcal{G}$ fools $\mathcal{D}$)

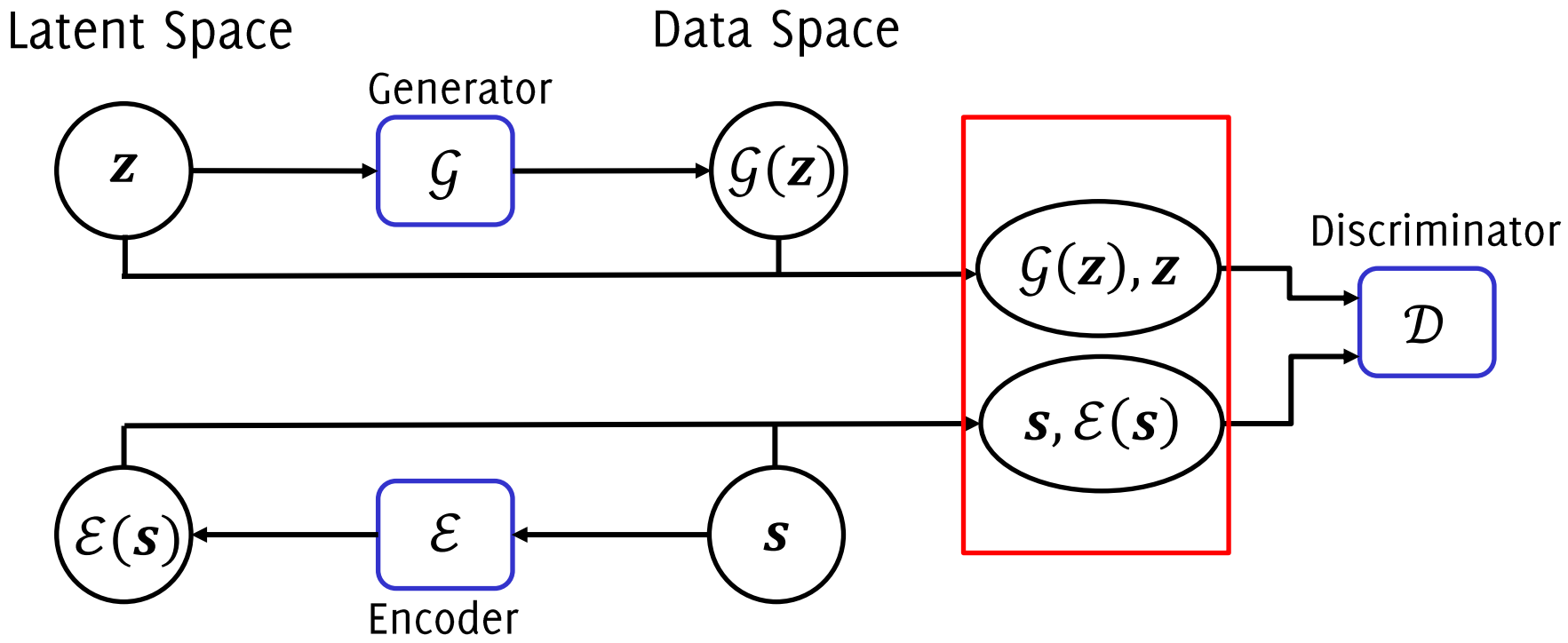
Anomaly score:

$$\mathcal{A}(\boldsymbol{s}) = \left|\left|\mathcal{G}(\hat{\boldsymbol{z}}) - \boldsymbol{s}\right|\right| + \lambda \log(1 - \mathcal{D}\big(\mathcal{G}(\hat{\boldsymbol{z}})\big))$$

We need to solve an optimization problem for each test sample!

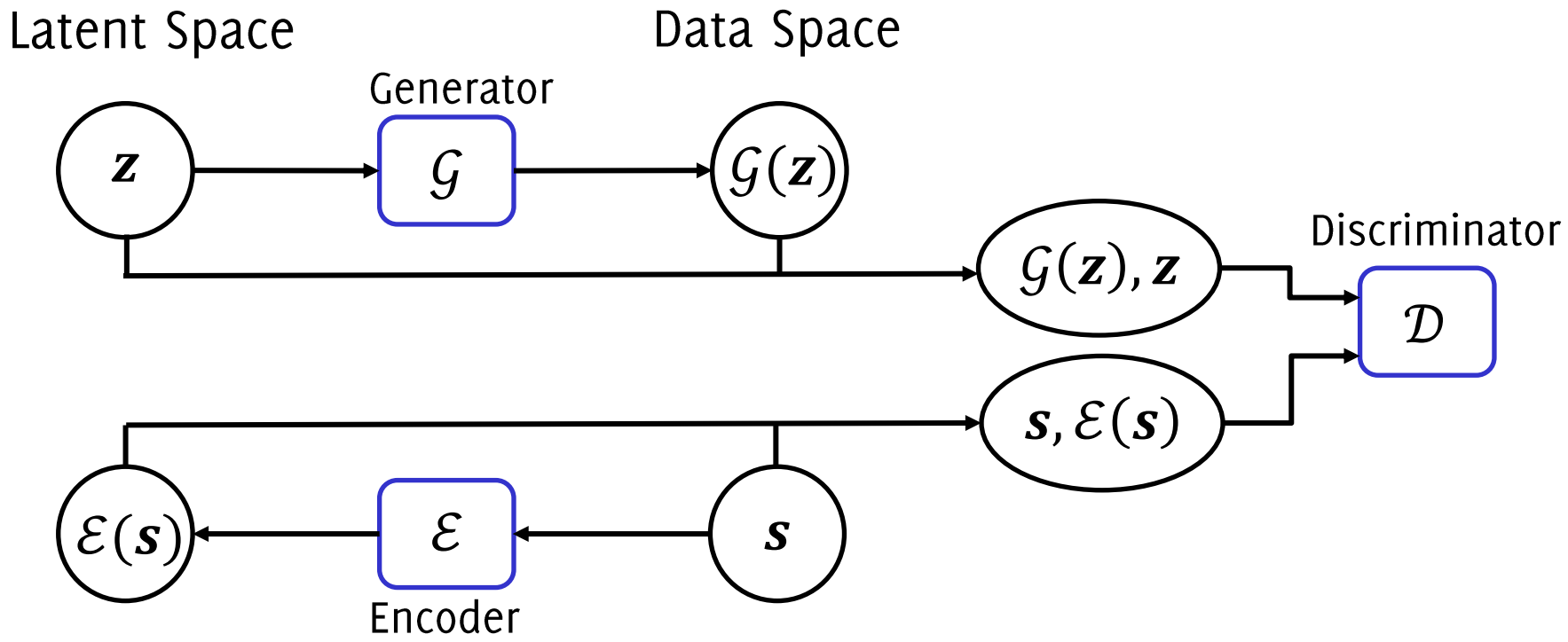
Schlegl et al, "*Unsupervised anomaly detection with generative adversarial networks to guide marker discovery*", IPMI 2017

BIDIRECTIONAL GAN



Donahue et al, "*Adversarial feature learning*", ICLR 2017

BIDIRECTIONAL GAN

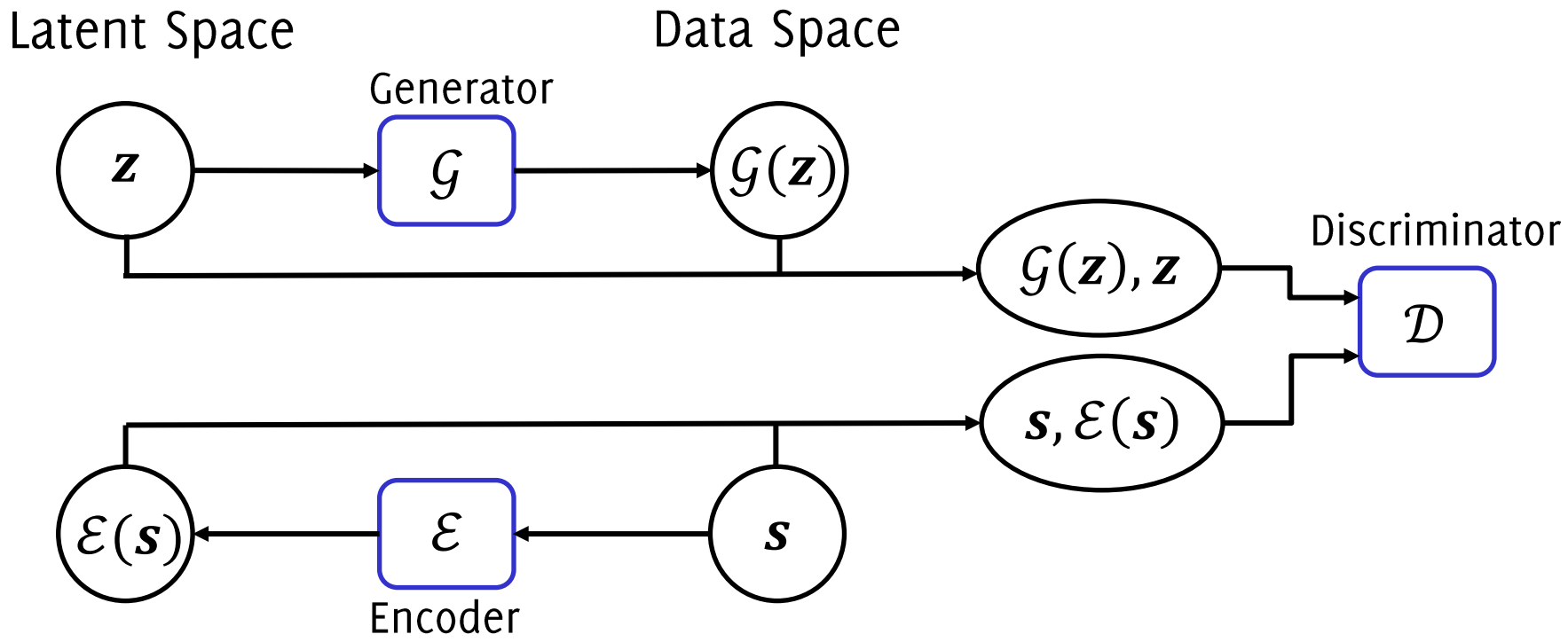


$$\min_{\mathcal{G},\mathcal{E}} \max_{\mathcal{D}} \mathcal{L}(\mathcal{D}, \mathcal{E}, \mathcal{G})$$

$$\mathcal{L}(\mathcal{D}, \mathcal{E}, \mathcal{G}) = \mathrm{E}_{s\sim\phi_S}[\log \mathcal{D}(\boldsymbol{s}, \mathcal{E}(\boldsymbol{s})] + \mathrm{E}_{z\sim\phi_Z}[\log(1 - \mathcal{D}(\mathcal{G}(\boldsymbol{z}), \boldsymbol{z}))]$$

Donahue et al, "*Adversarial feature learning*", ICLR 2017

BIDIRECTIONAL GAN



It can be proved that on the manifold $\mathcal{M}$ (i.e. on normal data):

$$\mathcal{E} = \mathcal{G}^{-1}$$

Donahue et al, "*Adversarial feature learning*", ICLR 2017

ANOGAN IMPROVED

We can use BiGAN to efficiently invert the generator in AnoGAN:

~~$$\hat{\boldsymbol{z}} = \min_{\boldsymbol{z}} \left|\left|G(\boldsymbol{z}) - \boldsymbol{s}\right|\right| + \lambda \log(1 - D(G(\boldsymbol{z})))$$~~

$$\hat{\boldsymbol{z}} = \mathcal{E}(\boldsymbol{s})$$

Zenati et al, "*Efficient GAN-based anomaly detection*", Workshop ICLR 2018

ANOGAN IMPROVED

We can use BiGAN to efficiently invert the generator in AnoGAN:

$$\hat{\boldsymbol{z}} = \min_{\boldsymbol{z}} \left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right| + \lambda \log(1 - \mathcal{D}(\mathcal{G}(\boldsymbol{z})))$$

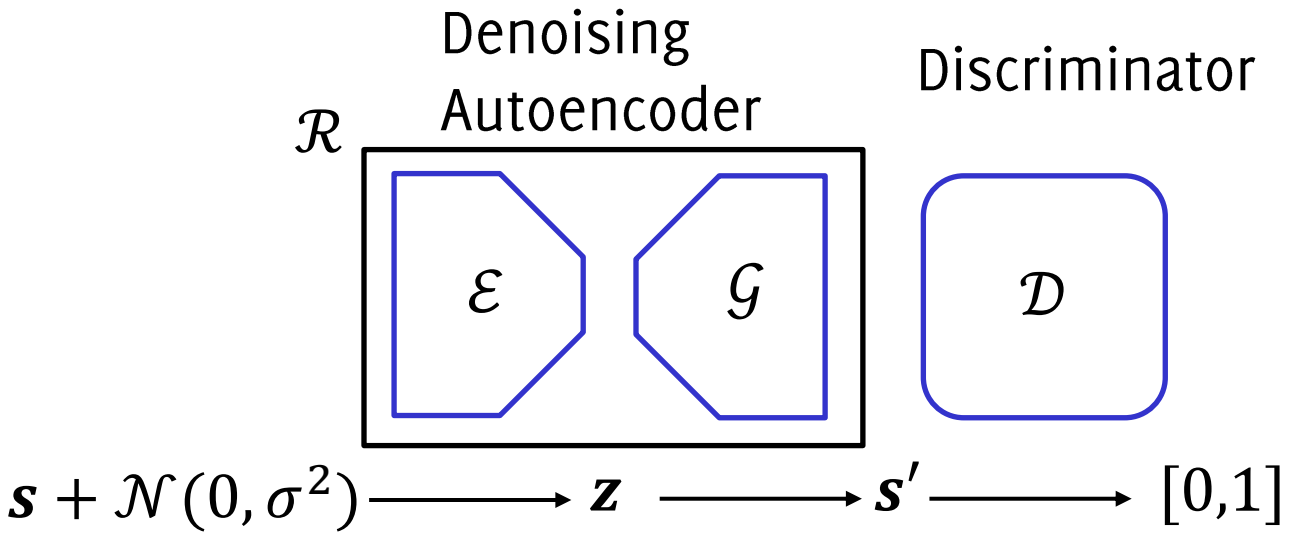
$$\hat{\boldsymbol{z}} = \mathcal{E}(\boldsymbol{s})$$

$$\mathcal{A}(\boldsymbol{s}) = \left|\left|\mathcal{G}(\hat{\boldsymbol{z}}) - \boldsymbol{s}\right|\right| + \lambda \log\left(1 - \mathcal{D}\left(\mathcal{G}(\hat{\boldsymbol{z}})\right)\right) =$$

$$= \left|\left|\mathcal{G}(\mathcal{E}(\boldsymbol{s})) - \boldsymbol{s}\right|\right| + \lambda \log\left(1 - \mathcal{D}\left(\mathcal{G}(\mathcal{E}(\boldsymbol{s}))\right)\right)$$

Zenati et al, "*Efficient GAN-based anomaly detection*", Workshop ICLR 2018

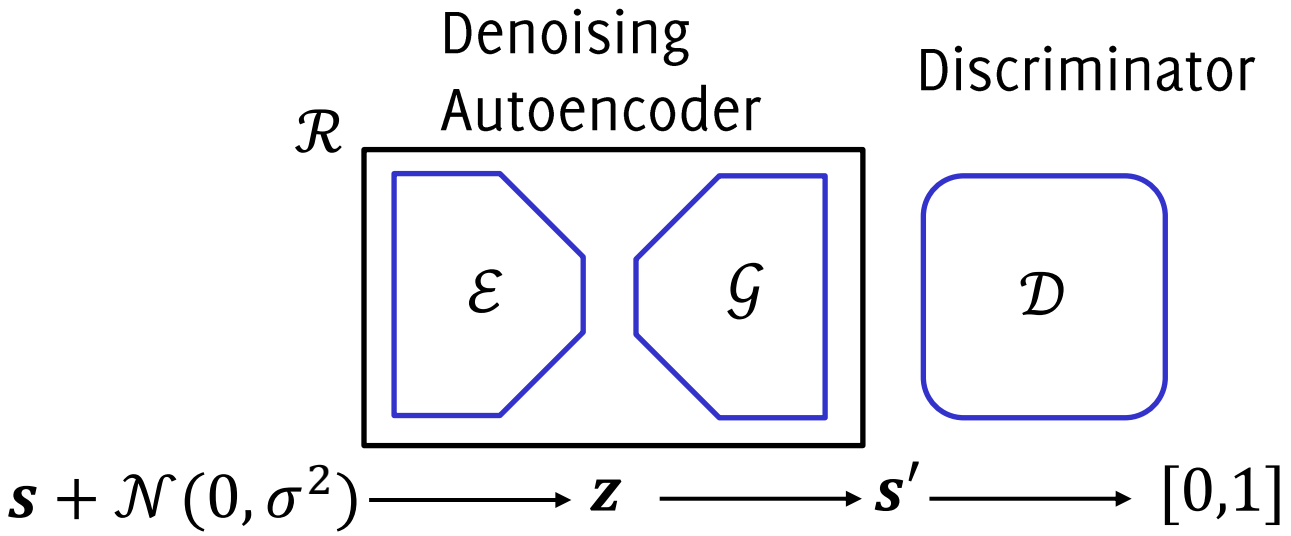
AUTOENCODER AS GENERATIVE MODELS



- $\mathcal{R}$ tries to reconstruct sample $\boldsymbol{s}$ from its noisy version $\tilde{\boldsymbol{s}}=\boldsymbol{s} + \mathcal{N}(0, \sigma^2)$
- $\mathcal{D}$ tries to discriminate between noise-free and reconstructed samples

Sabokrou et al, "*Adversarially learned one-class classifier for novelty detection*", CVPR 2018

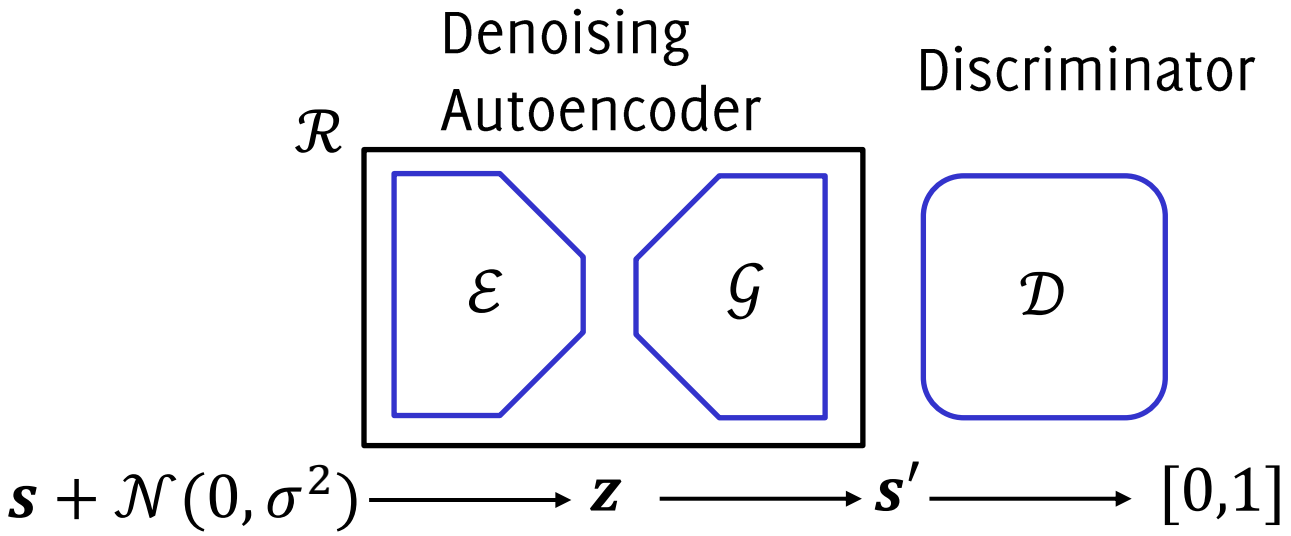
AUTOENCODER AS GENERATIVE MODELS



- $\mathcal{R}$ tries to reconstruct sample $\boldsymbol{s}$ from its noisy version $\tilde{\boldsymbol{s}}=\boldsymbol{s} + \mathcal{N}(0, \sigma^2)$
- $\mathcal{D}$ tries to discriminate between noise-free and reconstructed samples

Sabokrou et al, "*Adversarially learned one-class classifier for novelty detection*", CVPR 2018

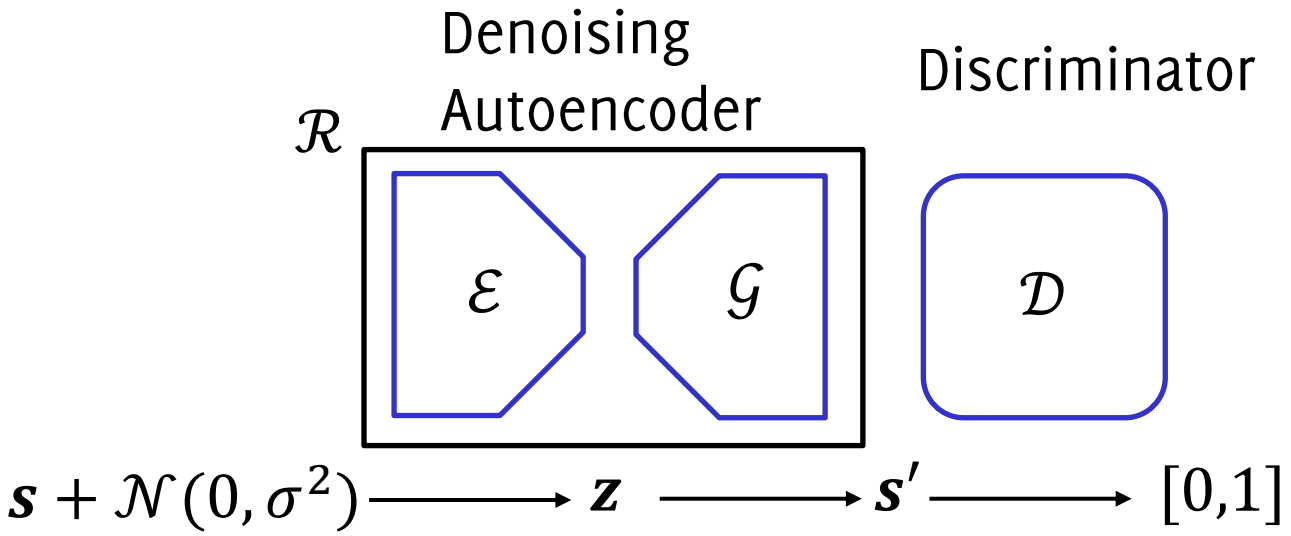
AUTOENCODER AS GENERATIVE MODELS



$$\min_{\mathcal{R}} \max_{\mathcal{D}} \left(\mathrm{E}_{\boldsymbol{s} \sim \phi_S} [\log \mathcal{D}(\boldsymbol{s})] + \mathrm{E}_{\tilde{\boldsymbol{s}} \sim \phi_S + \mathcal{N}(0,\sigma^2)} \left[\log \left(1 - \mathcal{D}\left(\mathcal{R}(\tilde{\boldsymbol{s}}) \right) \right) \right] \right)$$

Sabokrou et al, "*Adversarially learned one-class classifier for novelty detection*", CVPR 2018

AUTOENCODER AS GENERATIVE MODELS



$$\min_{\mathcal{R}} \max_{\mathcal{D}} \left(\mathrm{E}_{\boldsymbol{s} \sim \phi_S} [\log \mathcal{D}(\boldsymbol{s})] + \mathrm{E}_{\tilde{\boldsymbol{s}} \sim \phi_S + \mathcal{N}(0,\sigma^2)} \left[\log(1 - \mathcal{D}(\mathcal{R}(\tilde{\boldsymbol{s}}))) \right] \right)$$

We expect that $\mathcal{R}$ can successfully reconstruct (thus, fool $\mathcal{D}$) only normal sample:

$$\mathcal{A}(\boldsymbol{s}) = 1 - \mathcal{D}(\mathcal{R}(\boldsymbol{s}))$$

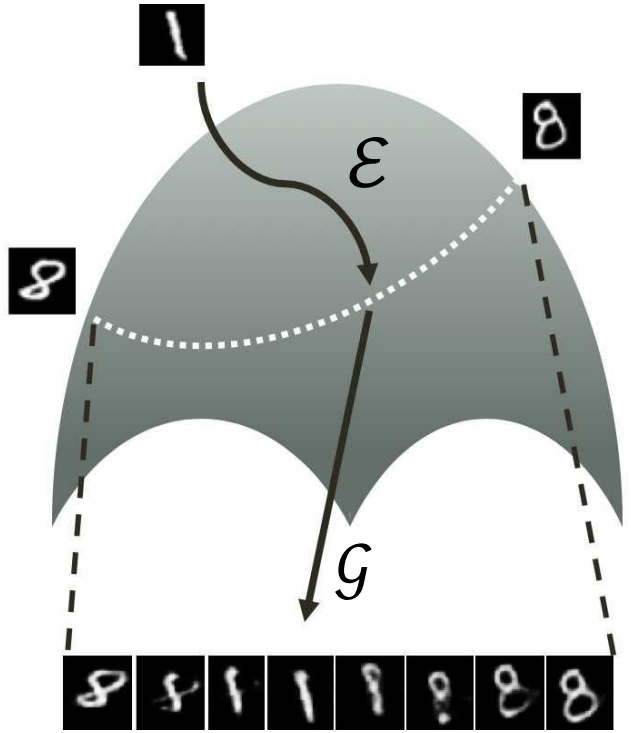
Sabokrou et al, "*Adversarially learned one-class classifier for novelty detection*", CVPR 2018

AUTOENCODER AS GENERATIVE MODELS

The generator $\mathcal{G}$ may be able to generate samples also of **anomalous class**

In this case it would be **impossible** to use $\mathcal{G}$ to **discriminate** between normal and anomalous samples

This may happen if the $\mathcal{E}$ maps anomalous samples in region of the latent space that were not explored during training



Perera et al, "*OCGAN: One-class novelty detection using GANs with constrained latent representations*", CVPR 2019

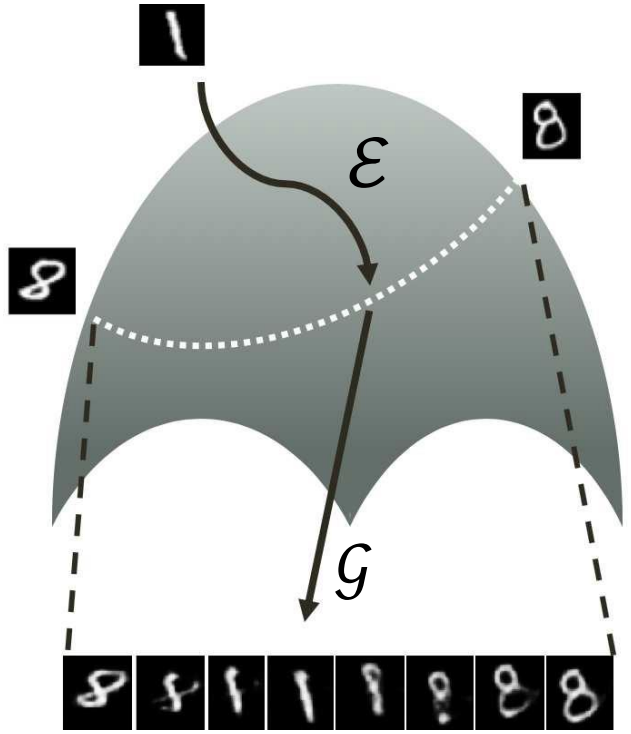
AUTOENCODER AS GENERATIVE MODELS

The generator $\mathcal{G}$ may be able to generate samples also of **anomalous class**

In this case it would be **impossible** to use $\mathcal{G}$ to **discriminate** between normal and anomalous samples

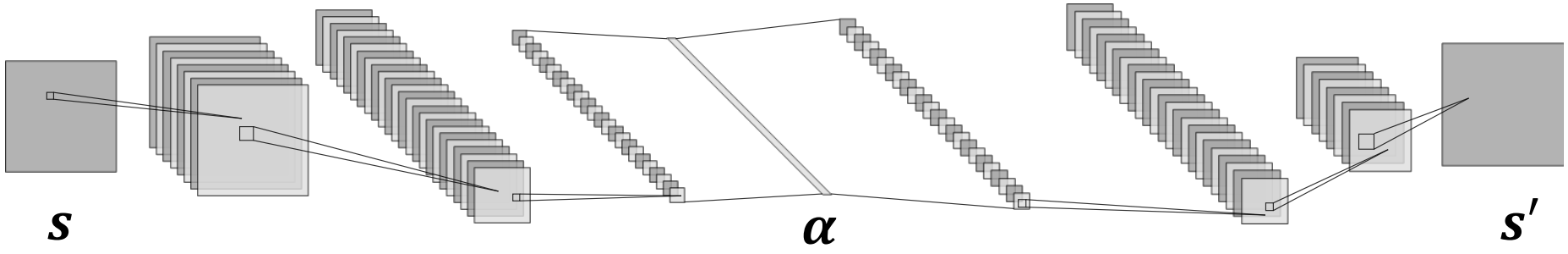
This may happen if the $\mathcal{E}$ maps anomalous samples in region of the latent space that were not explored during training

Idea: we can **enforce a known distribution on the latent space**

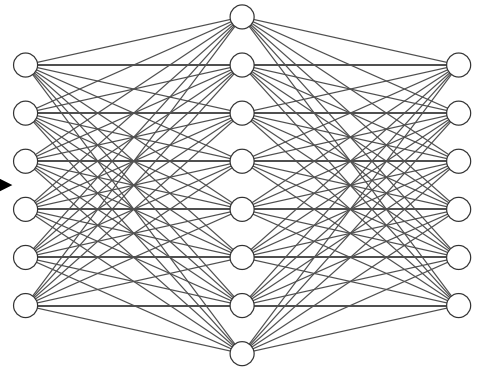


Perera et al, "*OCGAN: One-class novelty detection using GANs with constrained latent representations*", CVPR 2019

AUTOENCODER AS GENERATIVE MODELS



Estimation Network



Estimate the GM parameters as:

$$\pi_i = \frac{1}{N}\sum_n \gamma_{n,i}$$

$$\mu_i = \frac{\sum_n \gamma_{n,i}\boldsymbol{\alpha}_n}{\sum_n \gamma_{n,i}}$$

$$\Sigma_i = \frac{\sum_n \gamma_{n,i}(\boldsymbol{\alpha}_n - \boldsymbol{\mu}_i)(\boldsymbol{\alpha}_n - \boldsymbol{\mu}_i)^T}{\sum_n \gamma_{n,i}}$$

Zong et al, *"Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection", ICLR 2018*

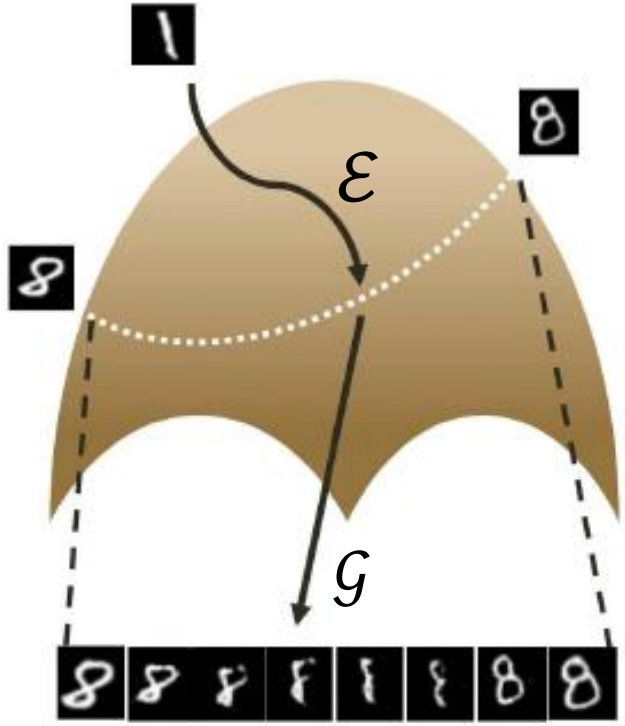
AUTOENCODER AS GENERATIVE MODELS

The generator $\mathcal{G}$ may be able to generate samples also of **anomalous class**

In this case it would be **impossible** to use $\mathcal{G}$ to **discriminate** between normal and anomalous samples

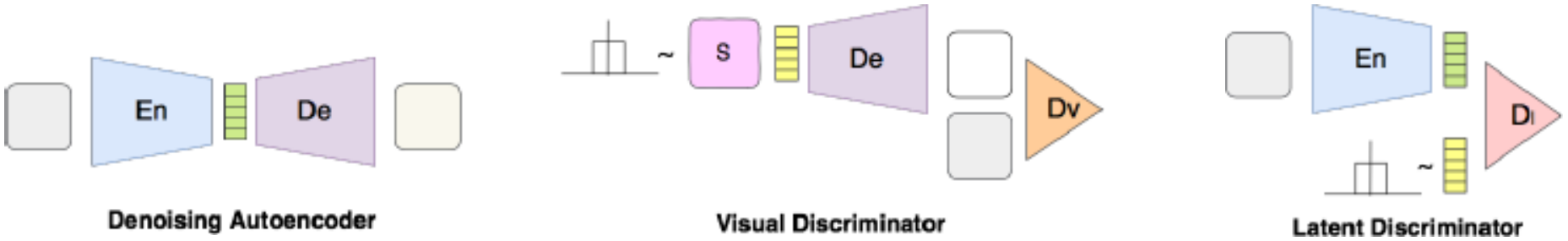
Idea: we can enforce a known distribution on the latent space

This can be done using a **latent discriminator** on the latent space



Perera et al, "*OCGAN: One-class novelty detection using GANs with constrained latent representations*", CVPR 2019

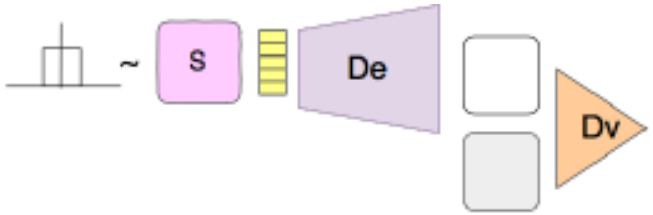
MUCH MORE COMPLICATED ARCHITECTURE



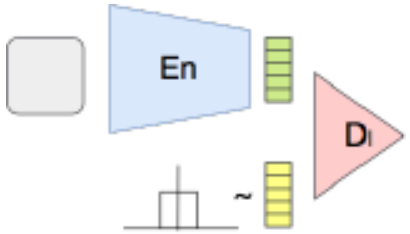
Perera et al, "*OCGAN: One-class novelty detection using GANs with constrained latent representations*", CVPR 2019

MUCH MORE COMPLICATED ARCHITECTURE

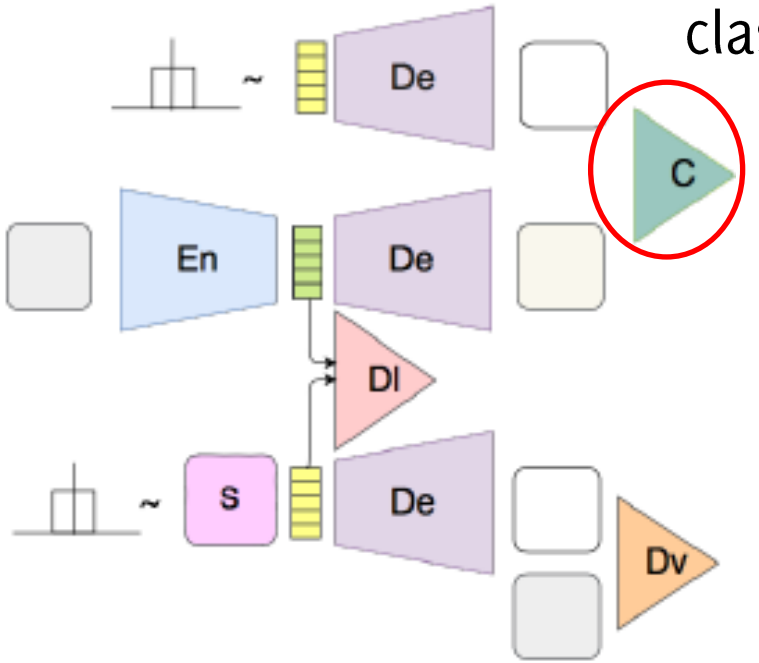
Denoising Autoencoder



Visual Discriminator

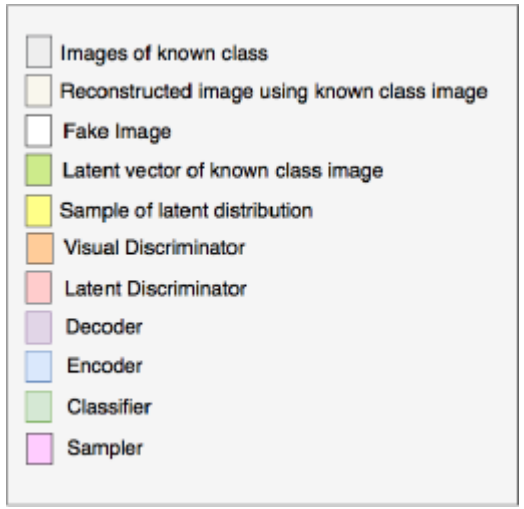


Latent Discriminator



OCGAN

The posterior probability of the classifier is used as anomaly score



Perera et al, "*OCGAN: One-class novelty detection using GANs with constrained latent representations*", CVPR 2019

CONCLUDING REMARKS

A FEW CONCLUDING REMARKS

Nowadays, anomaly detection problems are **ubiquitous** in **engineering and applied sciences**.

The presented general **framework encompasses most of algorithms** in the literature, which often **boil down to**

- **Feature extraction** (background model).
- **Definition of suitable statistics** (anomaly score).
- **Applying decision rules** to a set of **random variables**.

A FEW CONCLUDING REMARKS

When **data** are characterized by **complex structures**, as in case of images and signals, the feature extraction phase is the most critical one.

Data-driven models provide **meaningful representations** of signal and images, and these can be used to find good feature for anomaly detection.

Detectability loss:

- when d increases, changes become more difficult to detect, at least when monitoring the log-likelihood.
- This should be accounted when designing/learning features: **irrelevant components are harmful**!

A FEW CONCLUDING REMARKS

Nowadays the most powerful algorithms for feature extraction are based on deep learning, and in particular **Convolutional Neural Networks**

The key of the success of CNNs is the **end-to-end learning**, and in our case the **feature extractor** and the **anomaly score** are jointly learned.

Still, anomaly-detection methods based on deep learning rely on **decision rule**, which has to be defined according to some **statistical criteria**, e.g. to guarantee a fixed false positive rate.

THANK YOU VERY MUCH!

For any questions, feel free to contact us

giacomo.boracchi@polimi.it

diego.carrera@st.com

https://boracchi.faculty.polimi.it/tutorials.html