



Change and Anomaly Detection in Signal, Images, and General Data Streams

Giacomo Boracchi
DEIB, Politecnico di Milano,
giacomo.boracchi@polimi.it

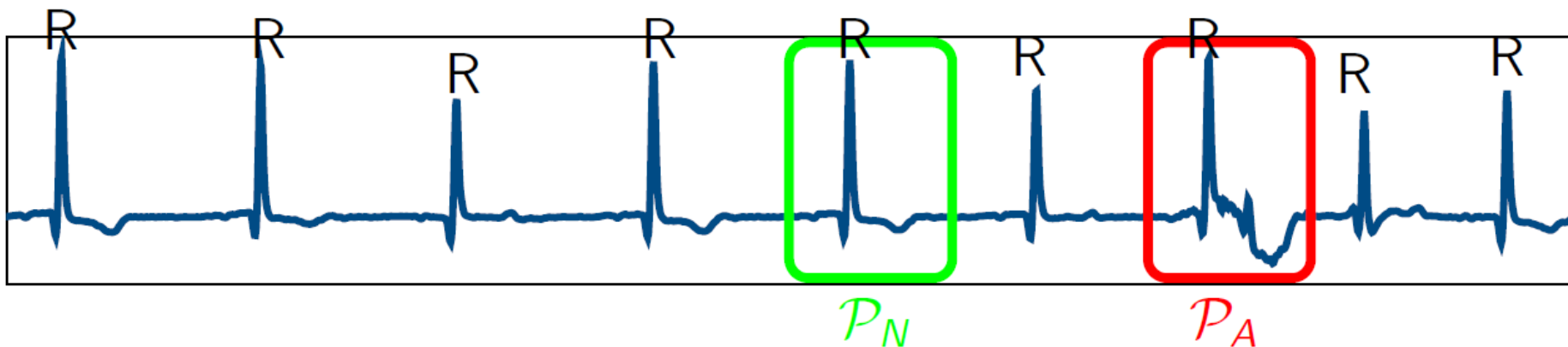
April 15, 2018
ICASSP 2018,
Calgary, Canada



... AN ANOMALY-DETECTION PROBLEM

Health monitoring / wearable devices:

Automatically analyze ECG tracings to detect arrhythmias or incorrect device positioning

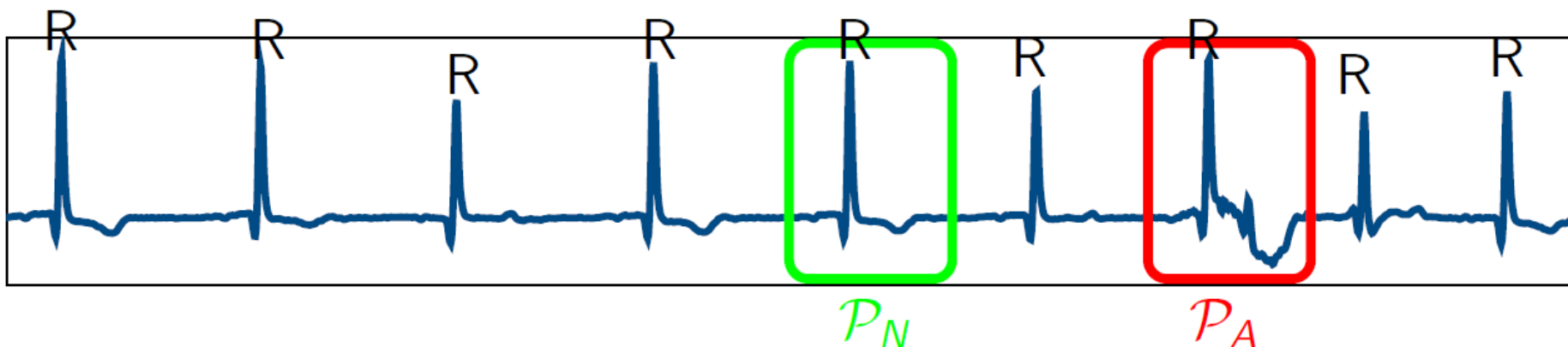
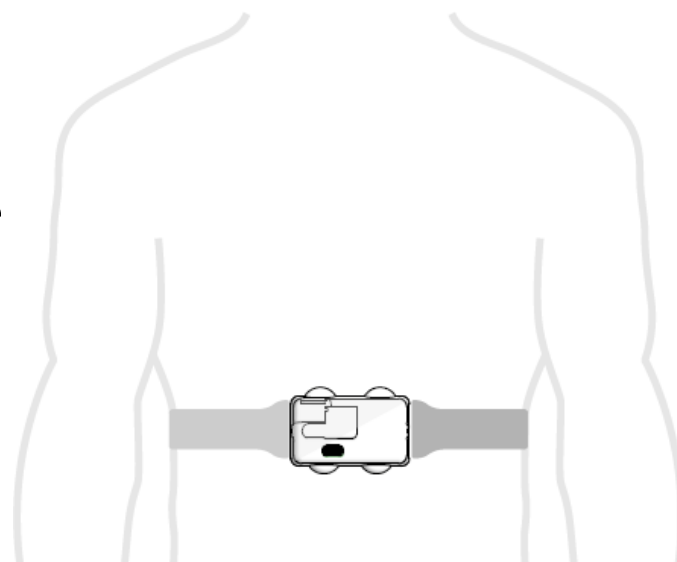




... AN ANOMALY-DETECTION PROBLEM

Health monitoring / wearable devices:

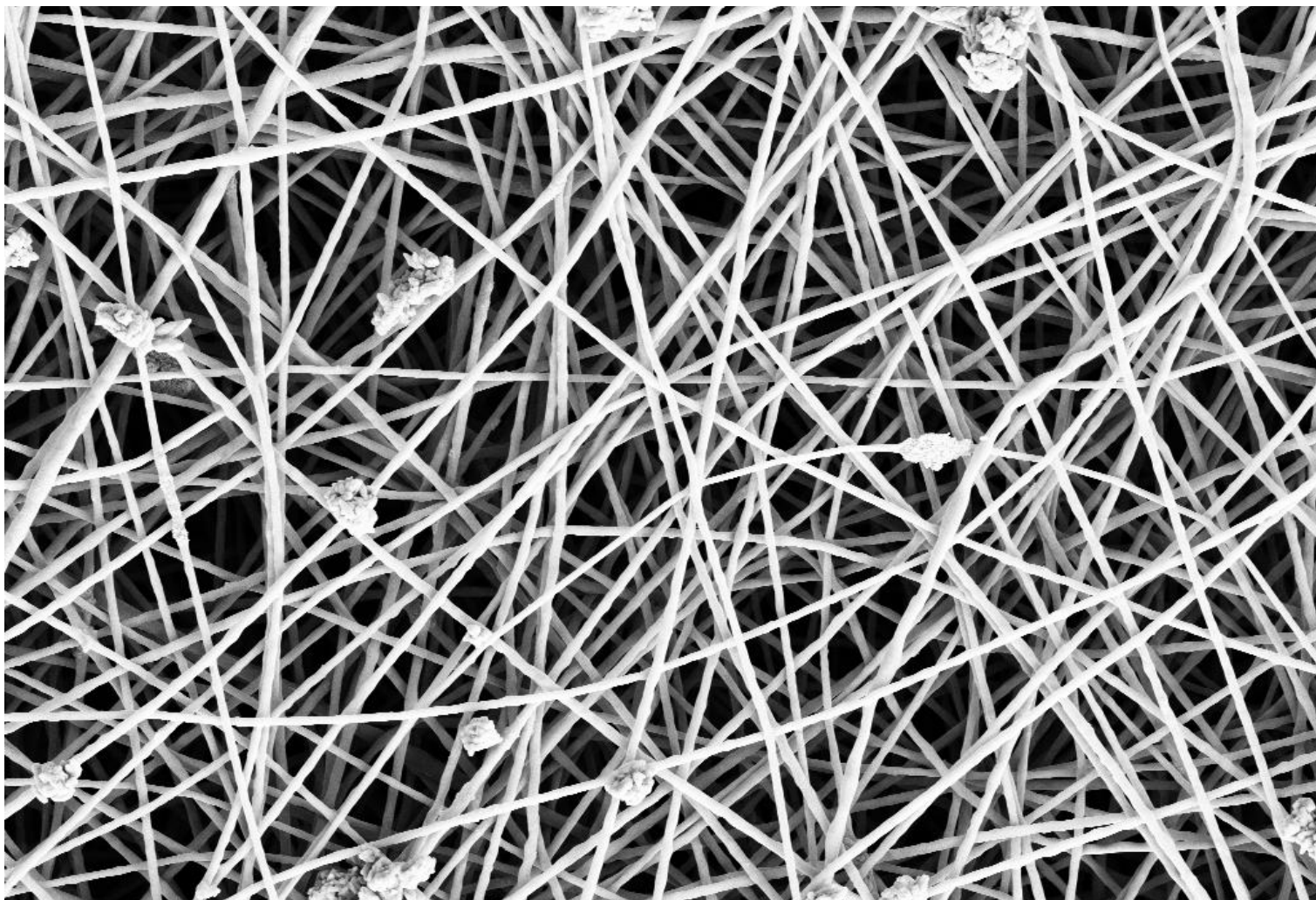
Automatically analyze ECG tracings to detect arrhythmias or incorrect device positioning





... AN ANOMALY-DETECTION PROBLEM

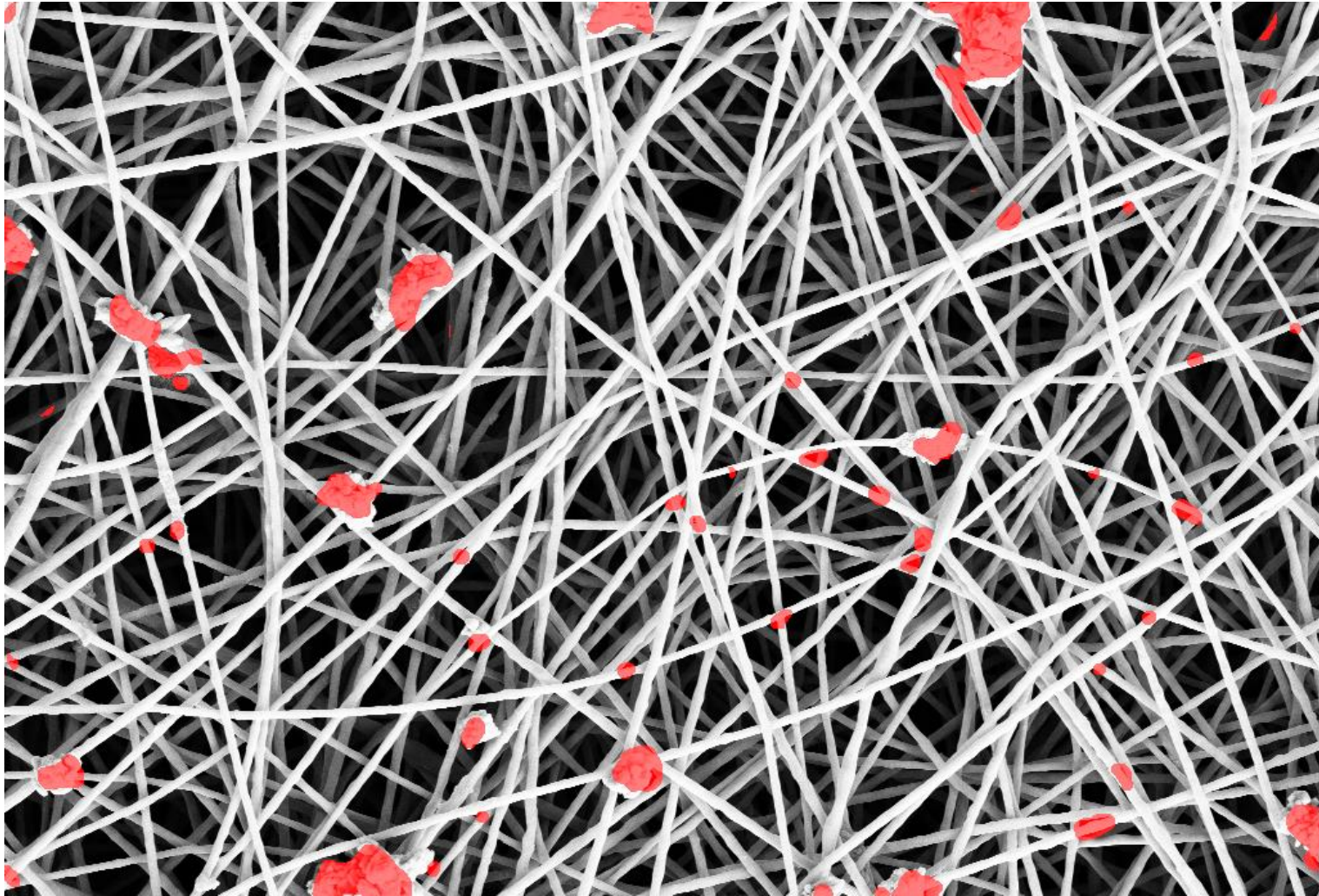
Quality Inspection Systems: monitoring the nanofiber production





... AN ANOMALY-DETECTION PROBLEM

Quality Inspection Systems: monitoring the nanofiber production

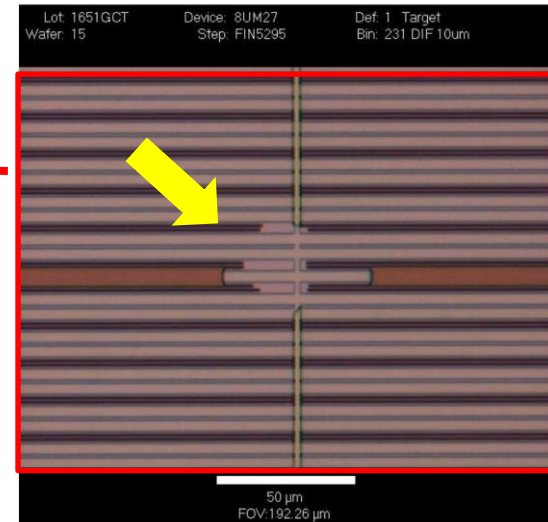
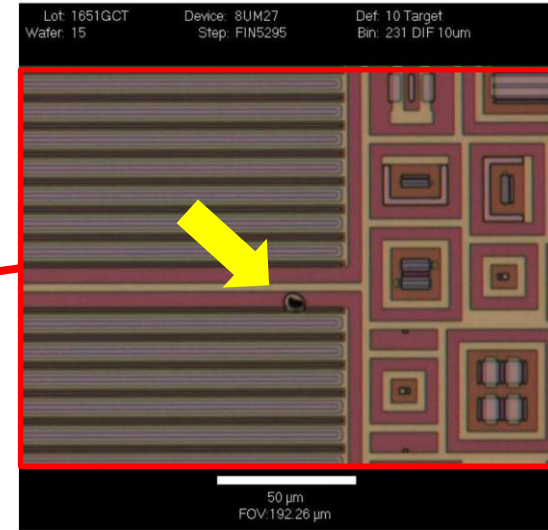
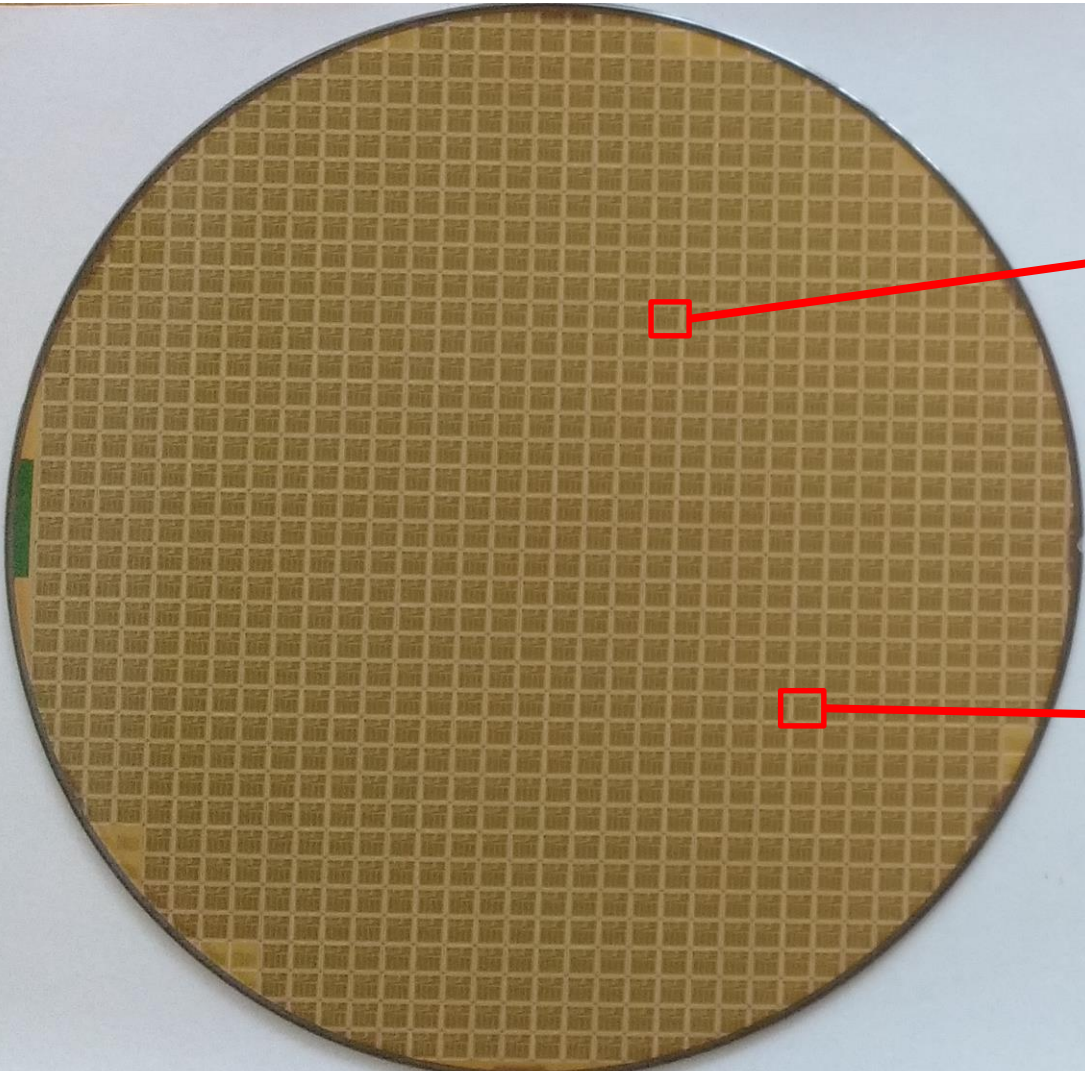


Carrera D., Manganini F., Boracchi G., Lanzarone E. "Defect Detection in SEM Images of Nanofibrous Materials", IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472



... AN ANOMALY-DETECTION PROBLEM

Detection of anomalies in chip production

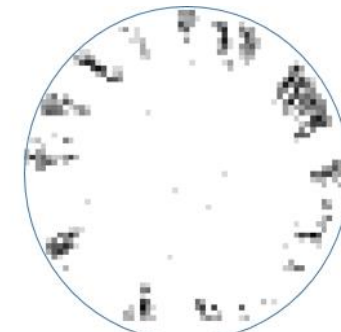
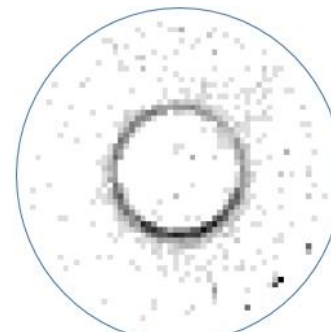
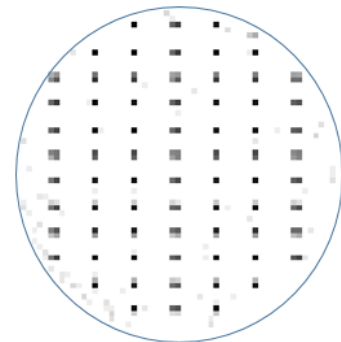
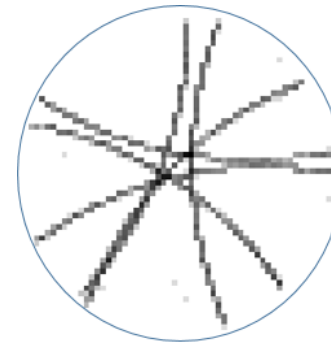
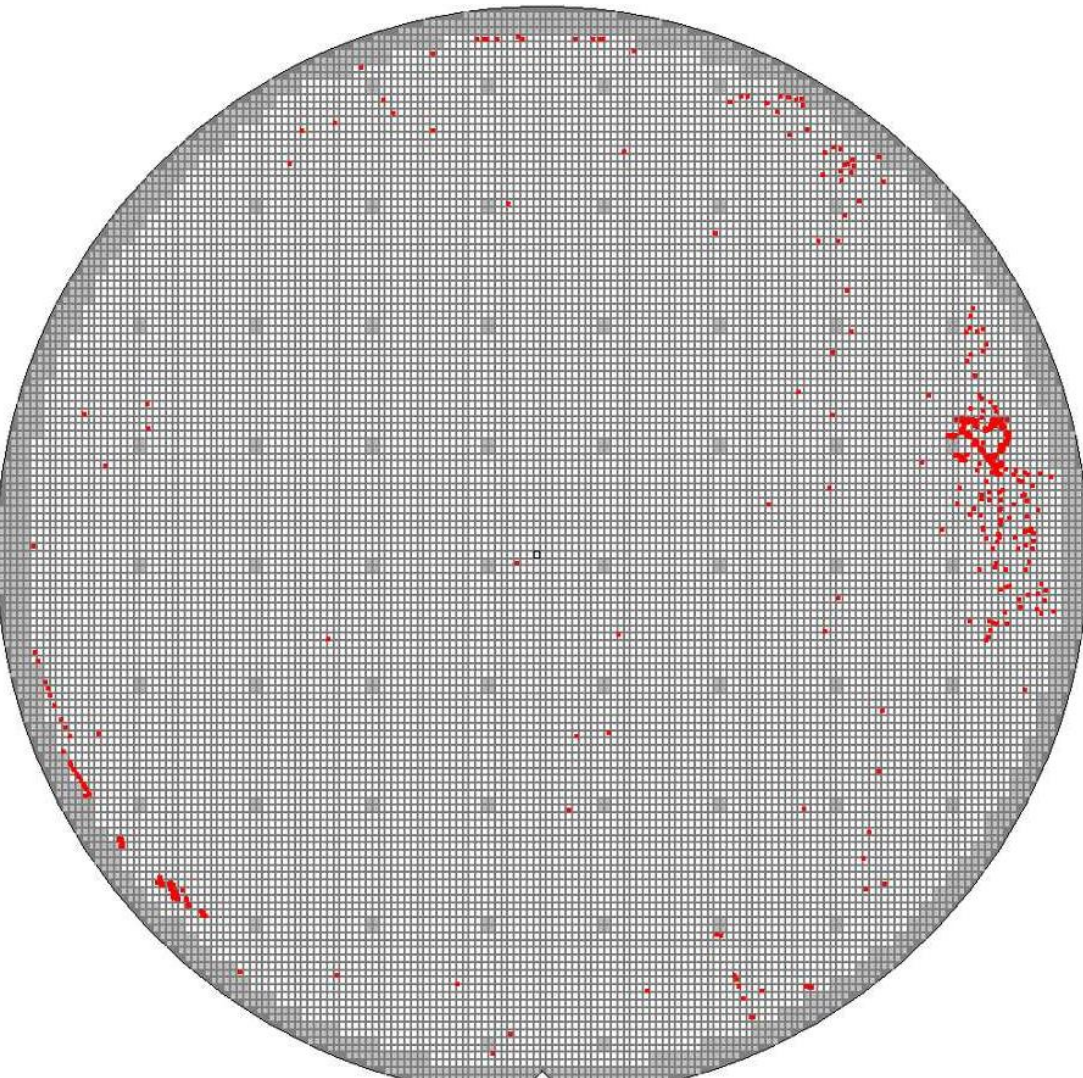




... AN ANOMALY-DETECTION PROBLEM

Detect anomalous patterns in the layout of defective chips.

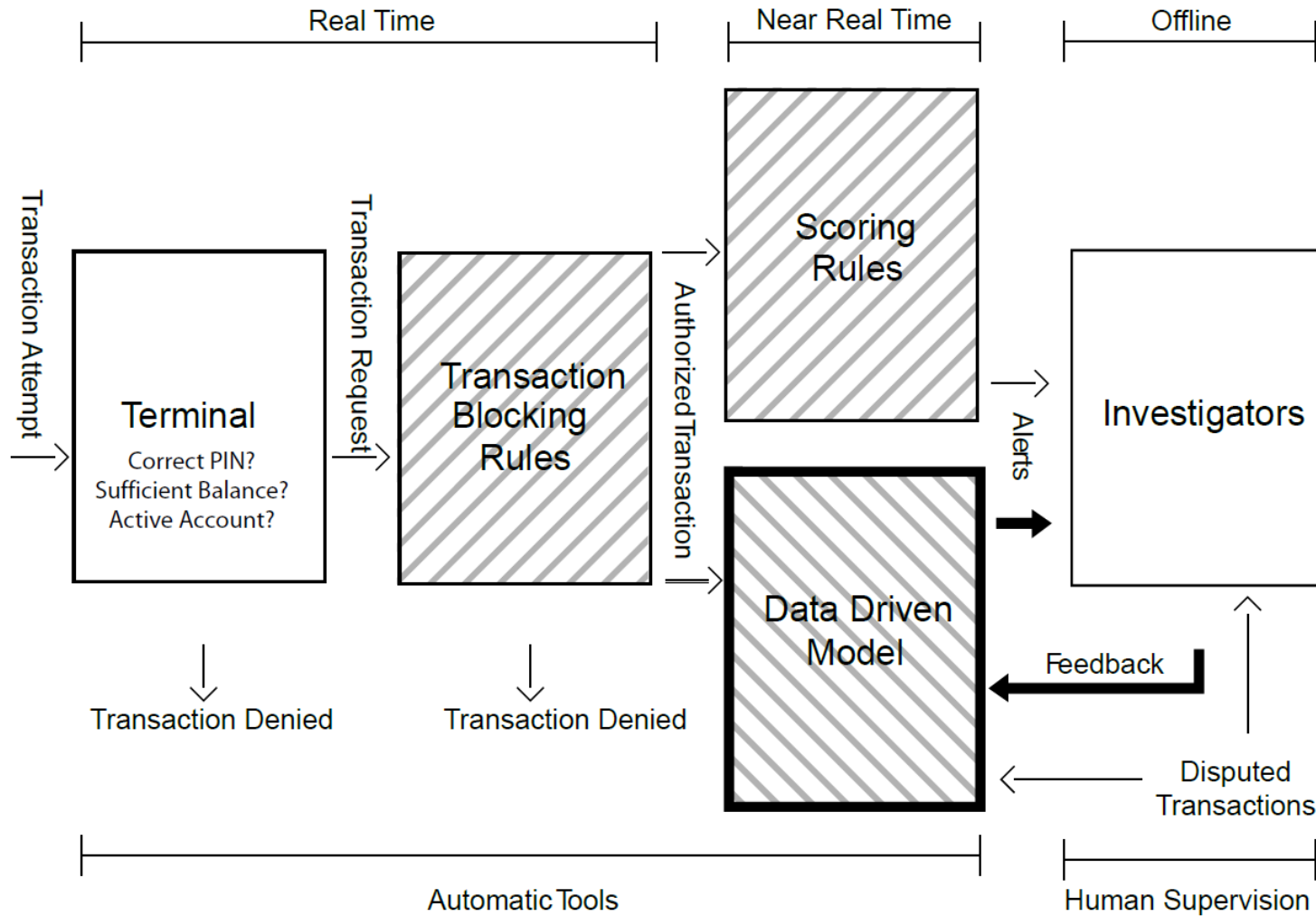
These might indicate issues and malfunctioning in the production process.





... AN ANOMALY-DETECTION PROBLEM

Fraud detection in streams of credit card transactions





OBJECT DETECTION

man

kid

glove

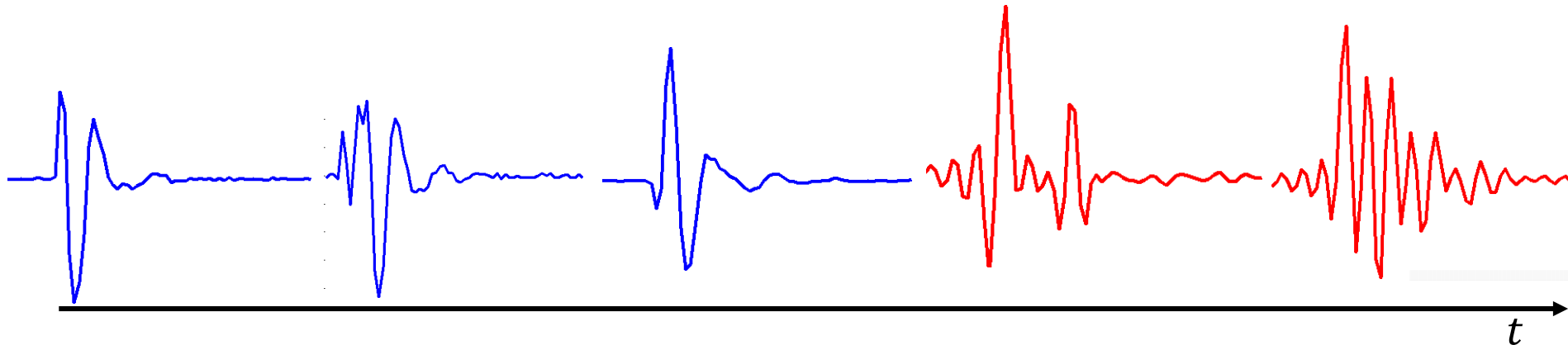




... A CHANGE-DETECTION PROBLEM

Environmental Monitoring

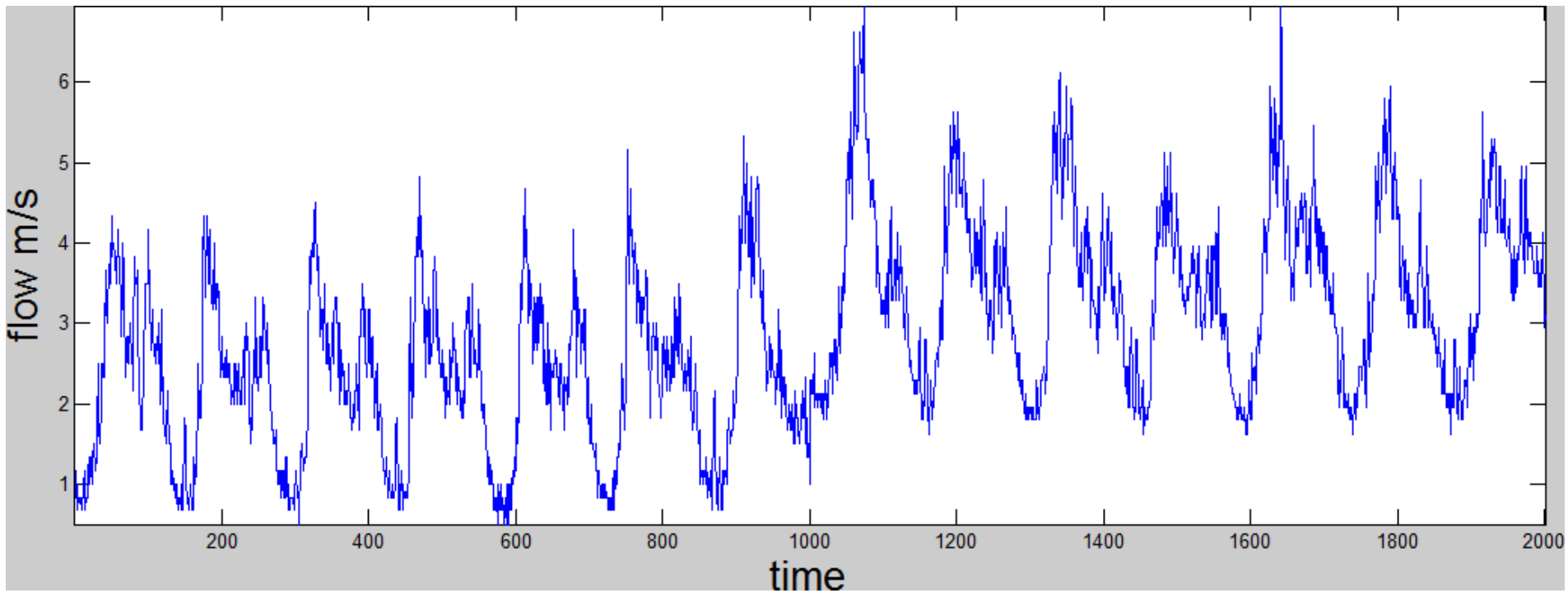
A sensor network monitoring rock faces: detecting changes in the waveforms that are recorded by MEMS sensors in network units.





... A CHANGE-DETECTION PROBLEM

Leak detection in Water Distribution Networks

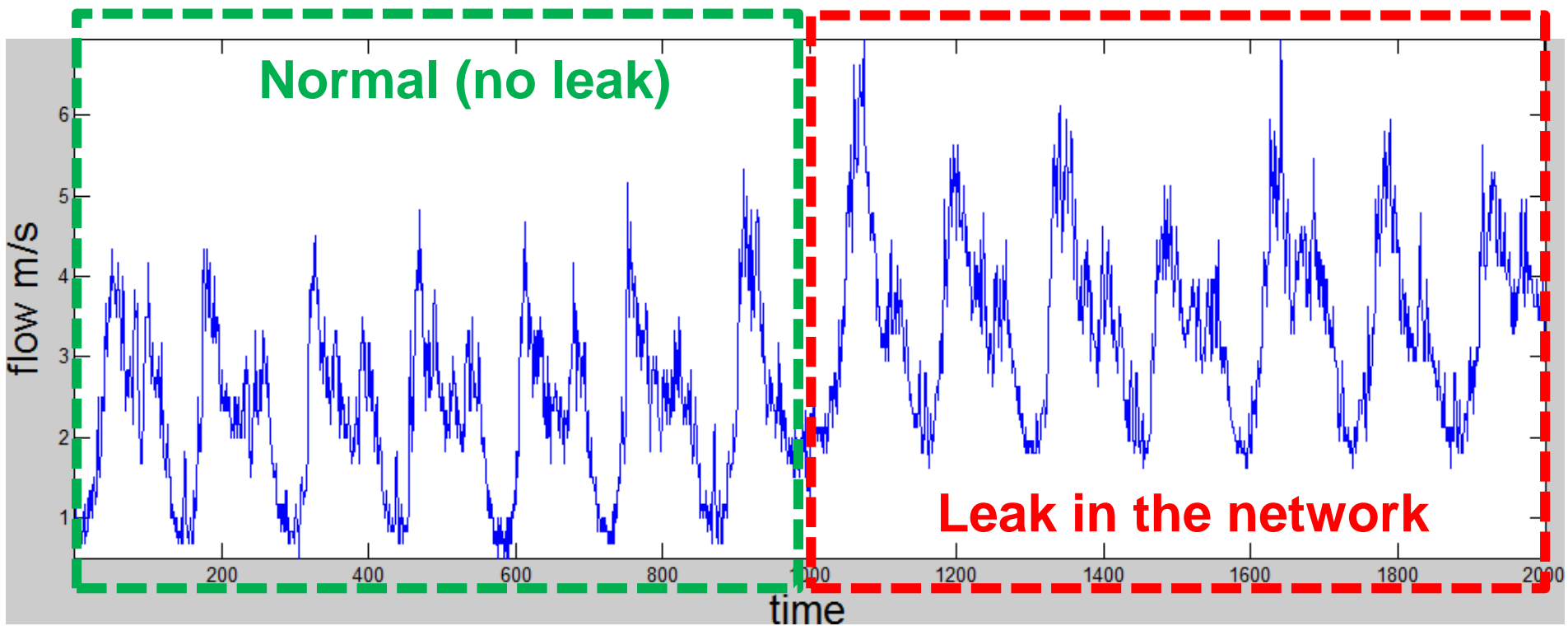




... A CHANGE-DETECTION PROBLEM

Leak detection in Water Distribution Networks

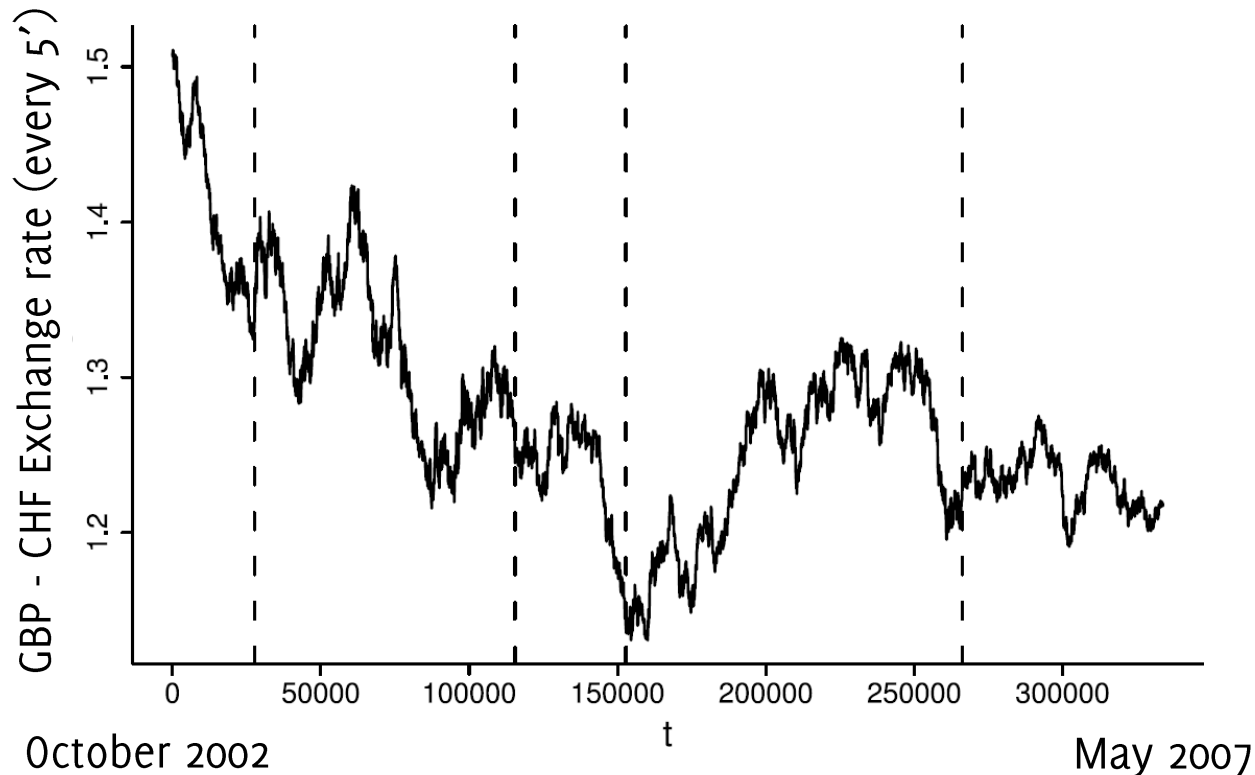
Similar problems arise in other critical infrastructure monitoring scenarios





... A CHANGE-DETECTION PROBLEM

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





... 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

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

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A. *"A survey on concept drift adaptation"*. ACM Computing Surveys (CSUR), 46(4), 44. (2014)



Part1, in the “Random Variable” world:

- Problem formulation and performance measures
- Anomaly/Change Detection in the ideal conditions
- Anomaly Detection in realistic conditions
- Change Detection in realistic conditions
- Monitoring high-dimensional data: detectability loss
- Best Experimental Practices

Part2, out of the “Random Variable” world:

- Anomaly/Change Detection for Signals and Images
- Anomaly/Change detection using learned models



DISCLAIMERS

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

I am **mainly** considering **numerical data**. In some cases, extensions apply to categorical or ordinal ones.

I refer to either changes/anomalies according to **my personal experience** in the applications I have been addressing.

For a **complete overview** on change/anomaly algorithms 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.



In The "Random Variable" World

Observations are i.i.d. realizations of a random variable



The Problem Formulation

Anomaly / Change Detection Problems
in a Statistical Framework



“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
- Image regions that do not conform a reference pattern

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



ANOMALY-DETECTION IN A STATISTICAL FRAMEWORK

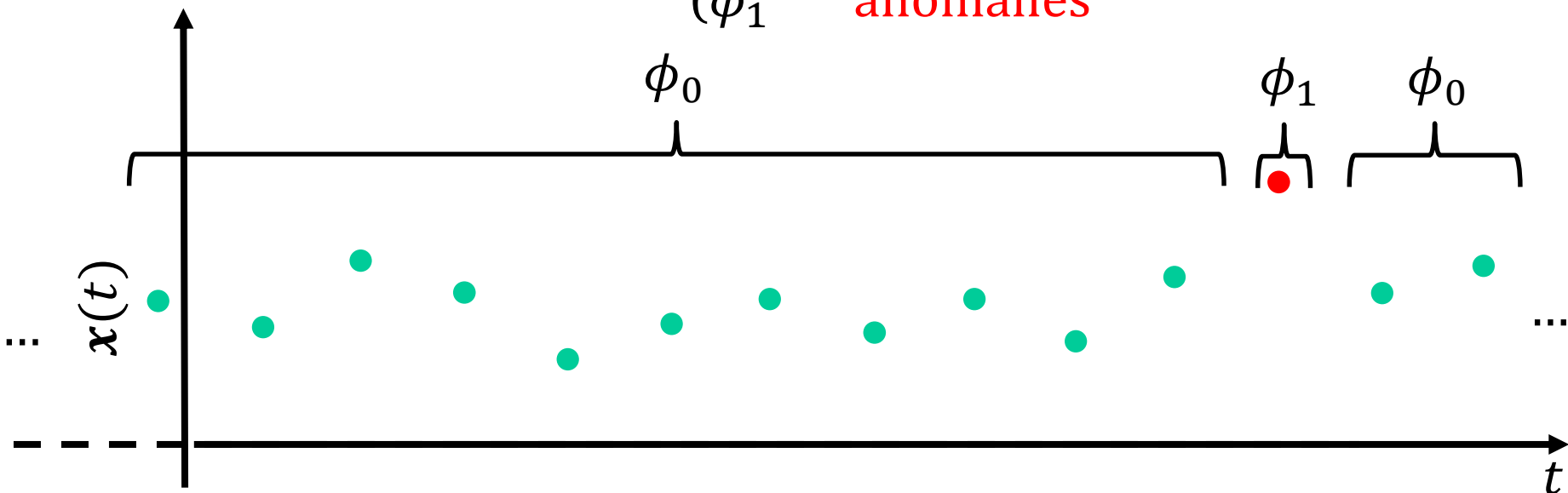
Often, the anomaly-detection problem boils down to:

Monitor a datastream

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

where $x(t)$ are realizations of a random variable having pdf ϕ_0 , and detect those points that are outliers i.e.,

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





ANOMALY-DETECTION IN A STATISTICAL FRAMEWORK

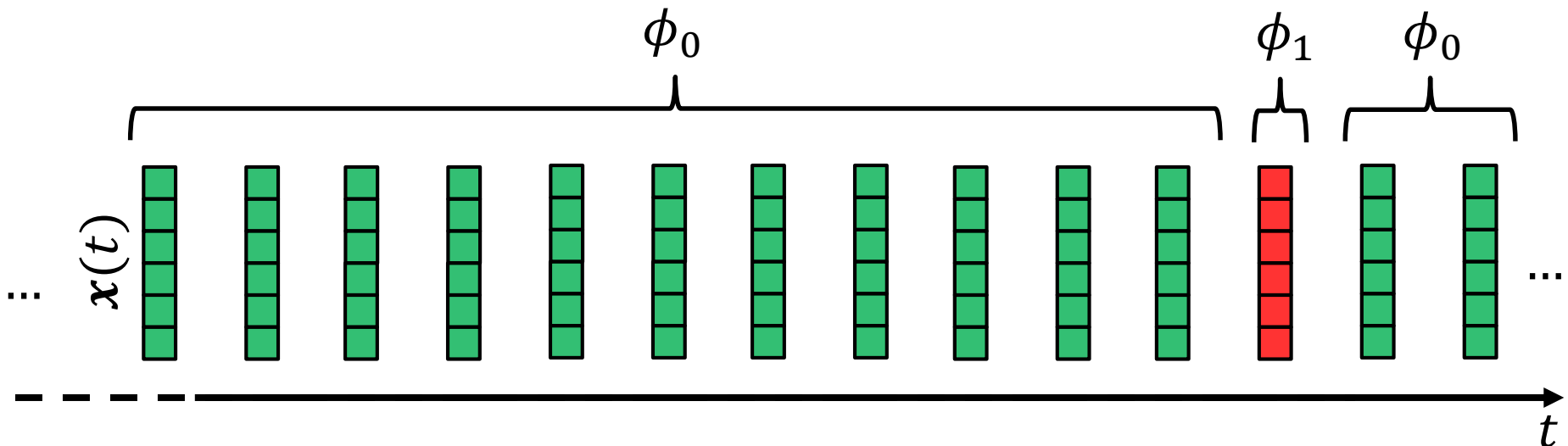
Often, the anomaly-detection problem boils down to:

Monitor a datastream

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

where $x(t)$ are realizations of a random variable having pdf ϕ_0 , and detect those points that are outliers i.e.,

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





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.



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.

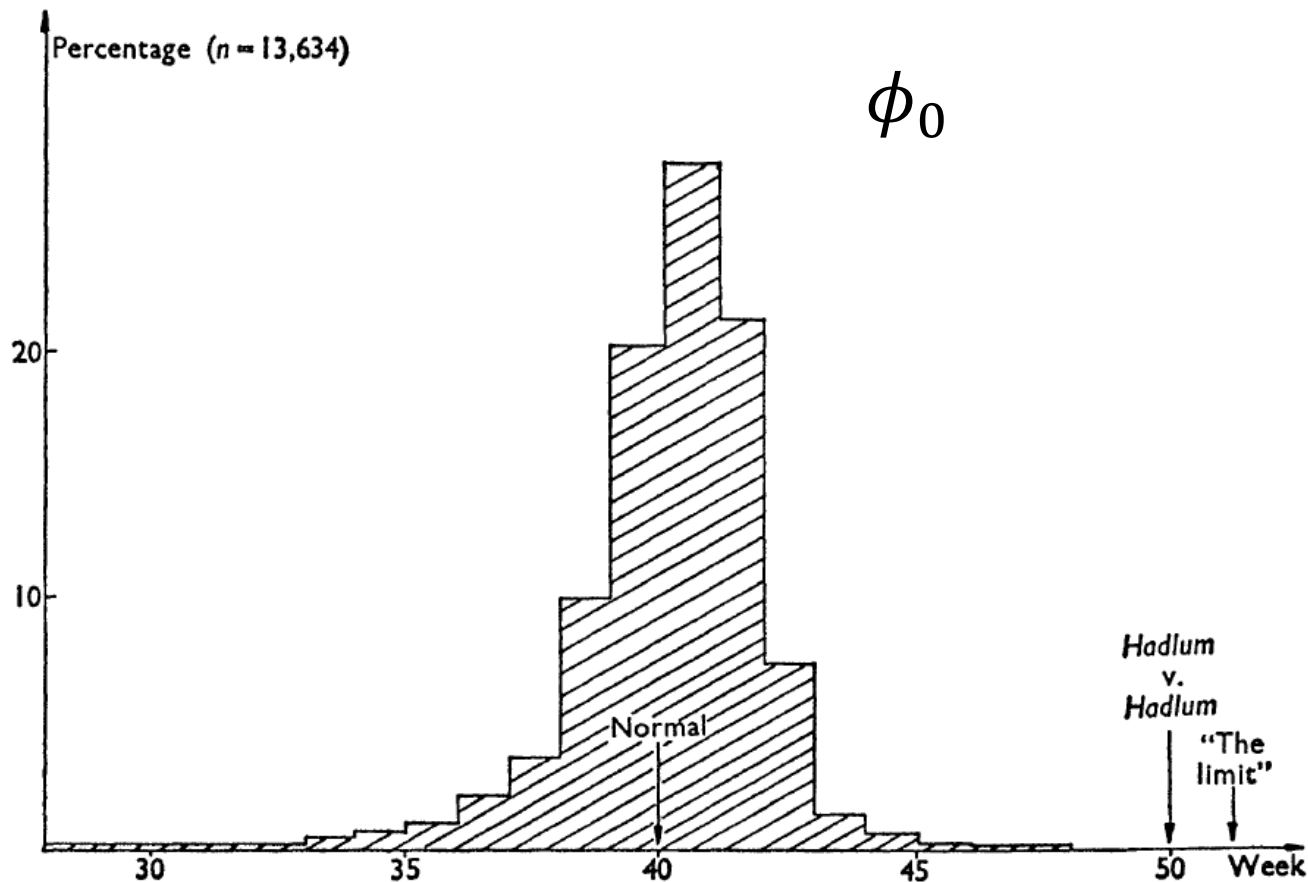
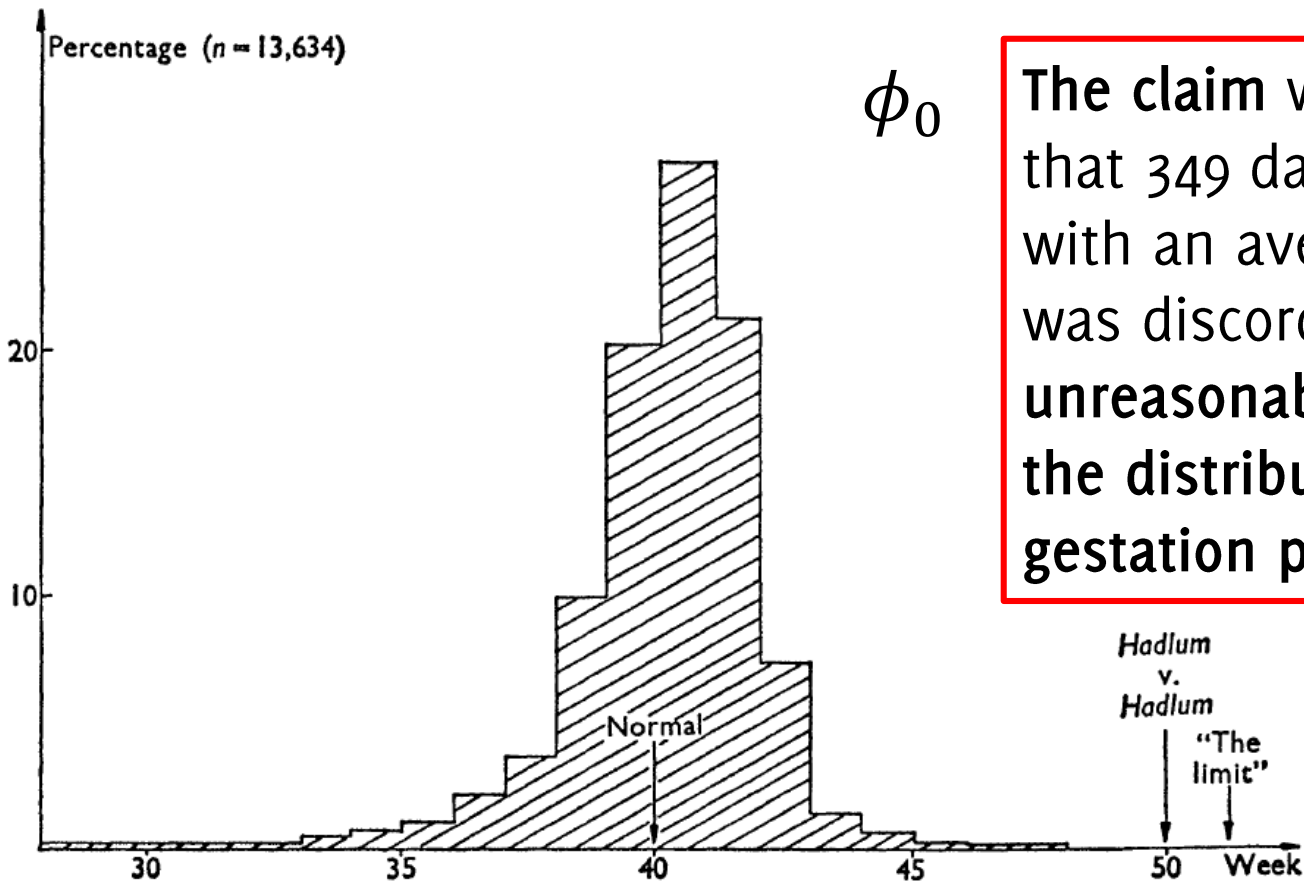


FIG. 1. Distribution of human gestation periods.



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.



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

FIG. 1. Distribution of human gestation periods.



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.

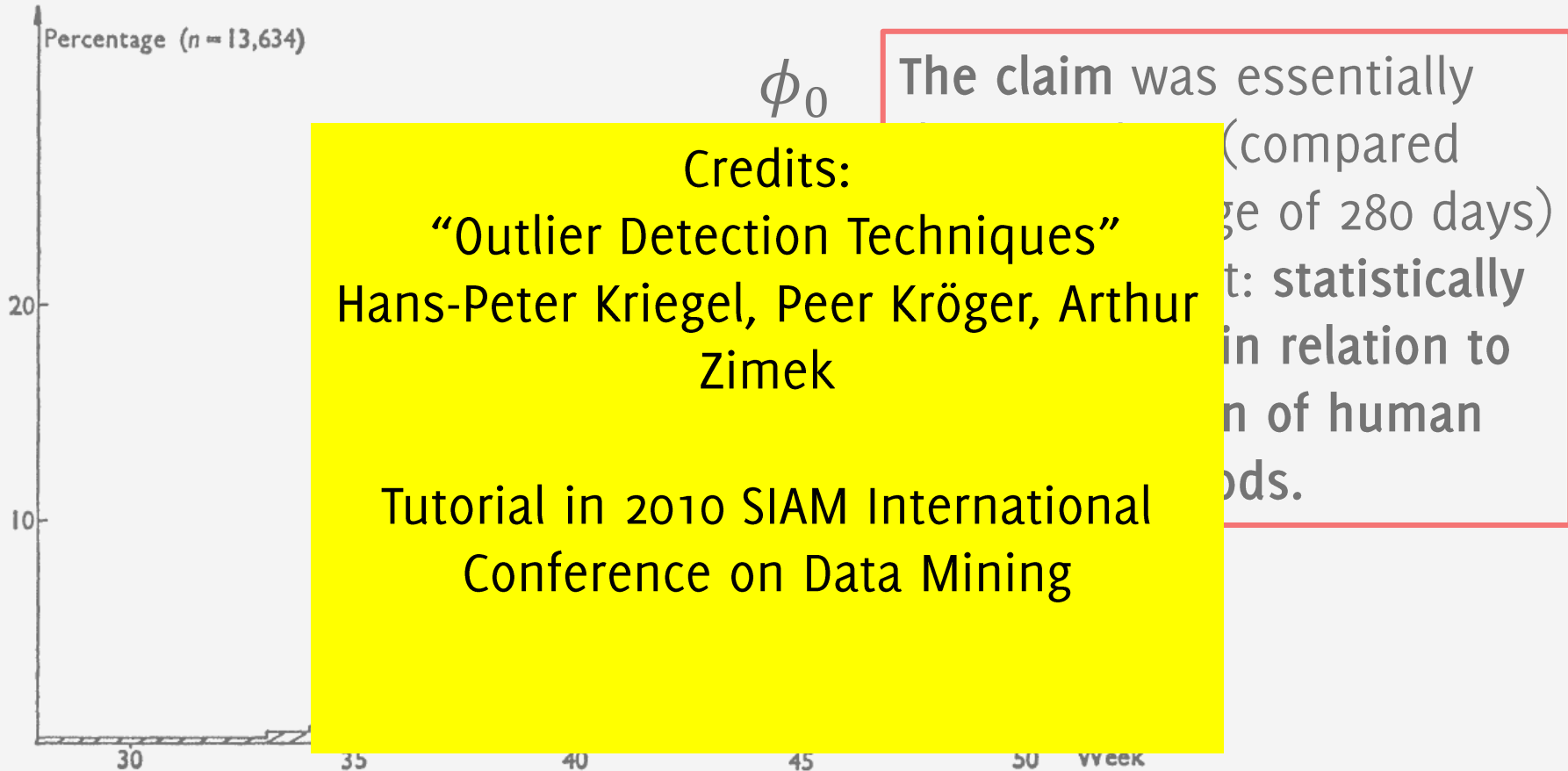


FIG. 1. Distribution of human gestation periods.



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-DETECTION IN A STATISTICAL FRAMEWORK

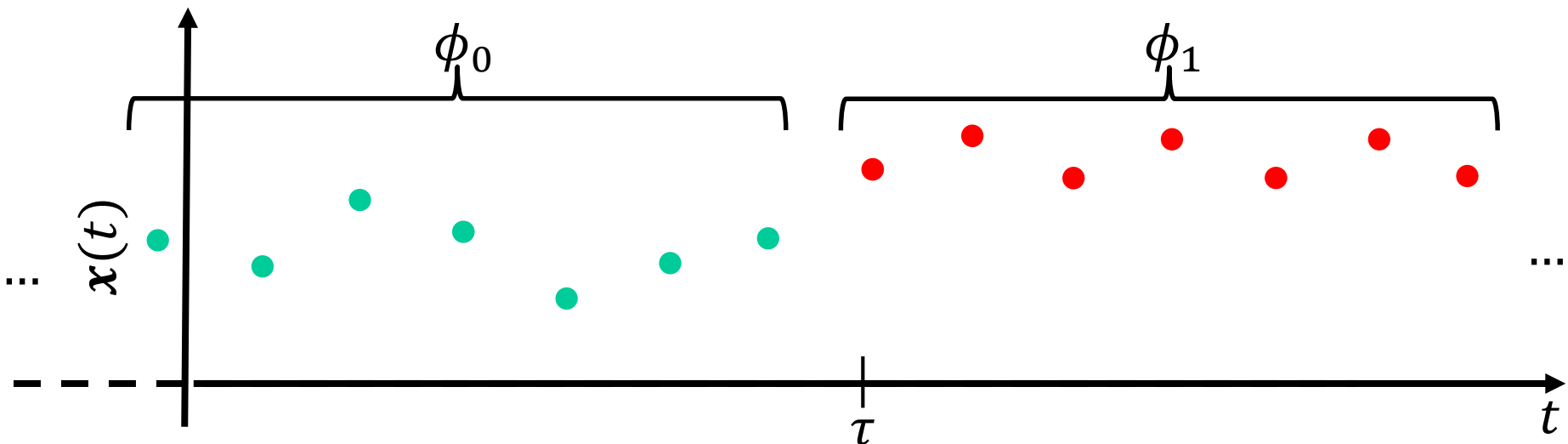
Often, the change-detection problem boils down to:

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

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

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

We denote such change as: $\phi_0 \rightarrow \phi_1$





CHANGE-DETECTION IN A STATISTICAL FRAMEWORK

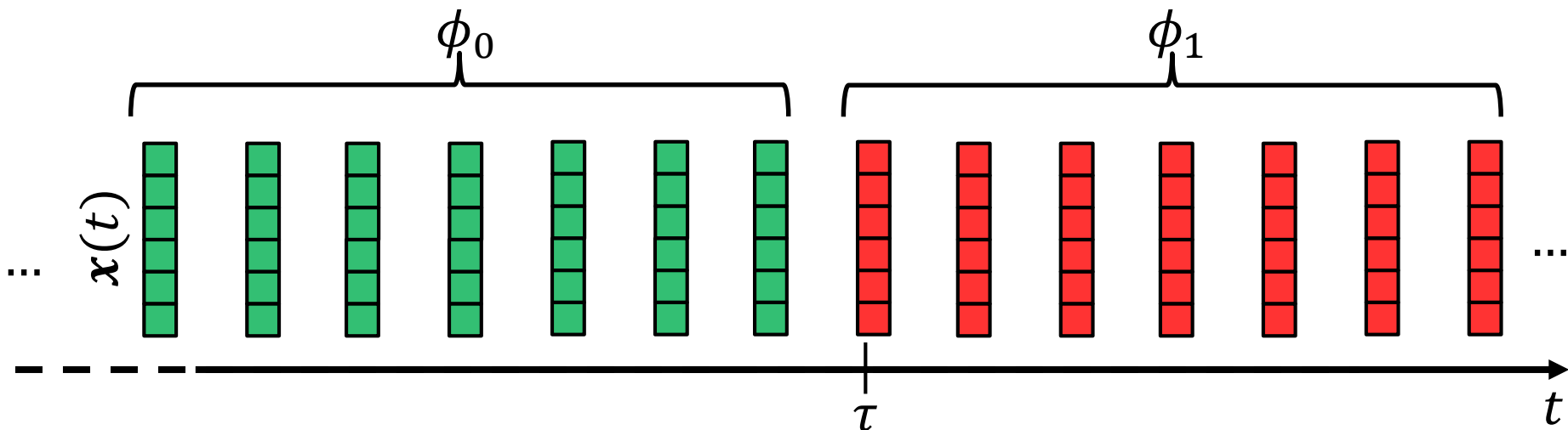
Often, the change-detection problem boils down to:

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

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

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

We denote such change as: $\phi_0 \rightarrow \phi_1$

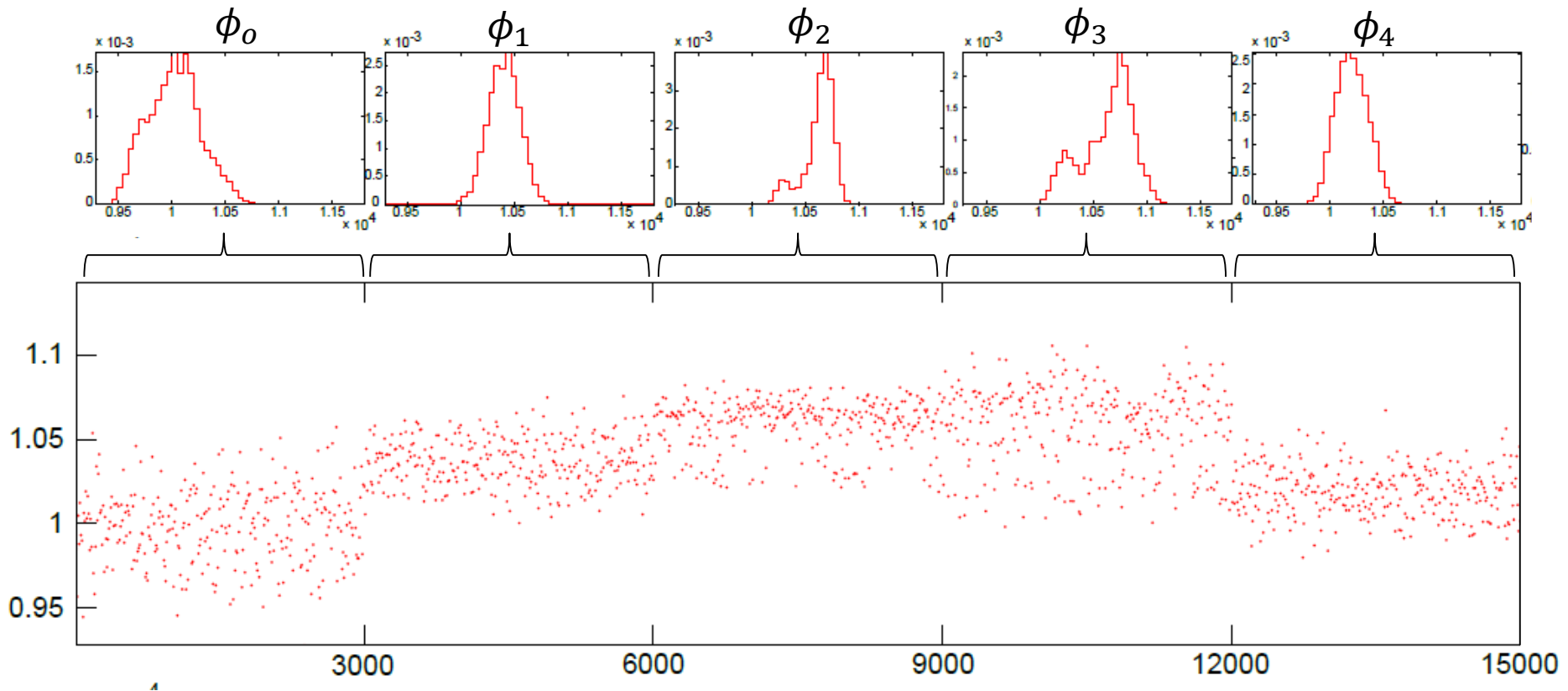




CHANGE-DETECTION IN A STATISTICAL FRAMEWORK

Here are data from an X-ray monitoring apparatus.

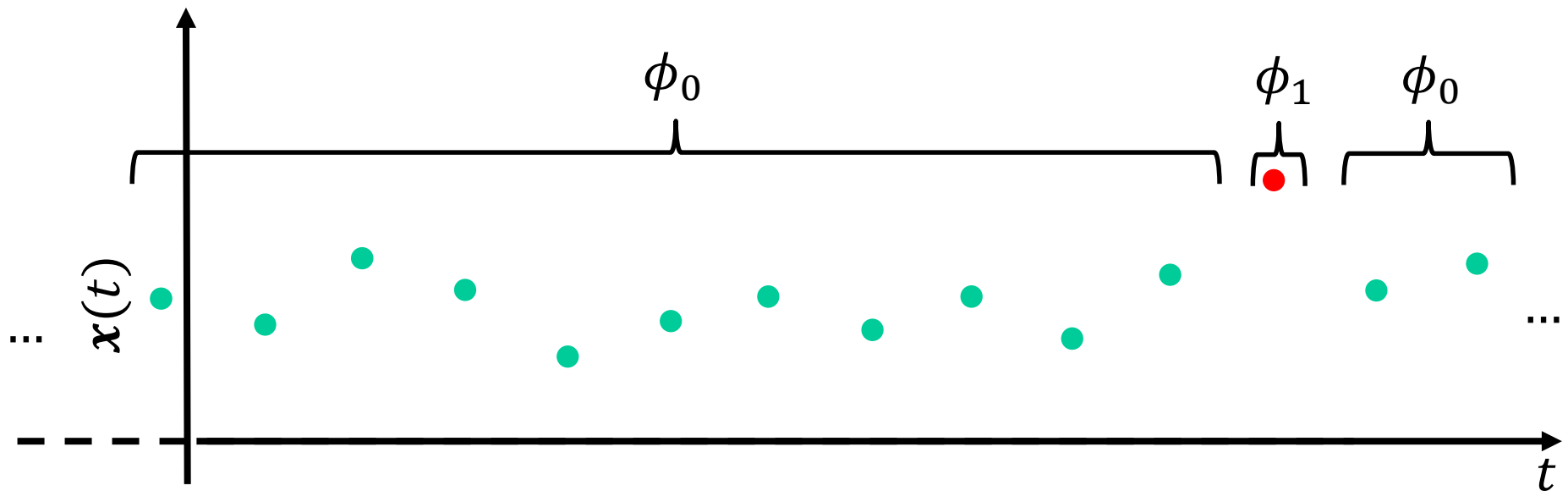
There are 4 changes $\phi_0 \rightarrow \phi_1 \rightarrow \phi_2 \rightarrow \phi_3 \rightarrow \phi_4$ corresponding to different monitoring conditions and/or analyzed materials





PROCESS CHANGES VS ANOMALIES

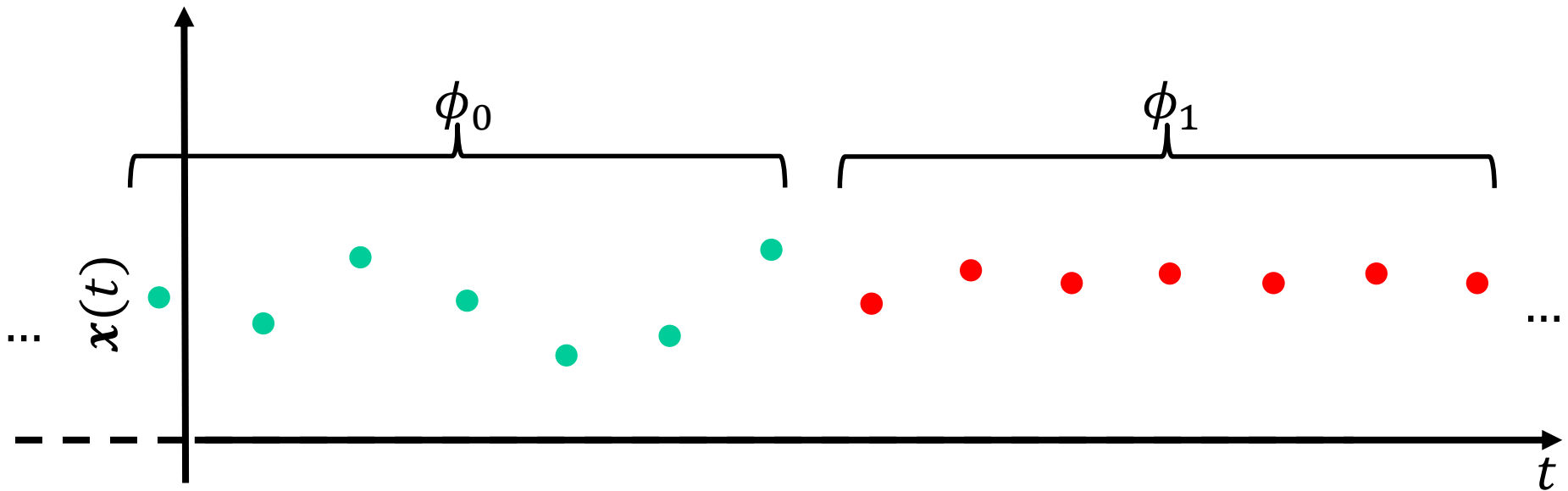
Not all anomalies are due to process changes





PROCESS CHANGES VS ANOMALIES

Not all process changes result in anomalies





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



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

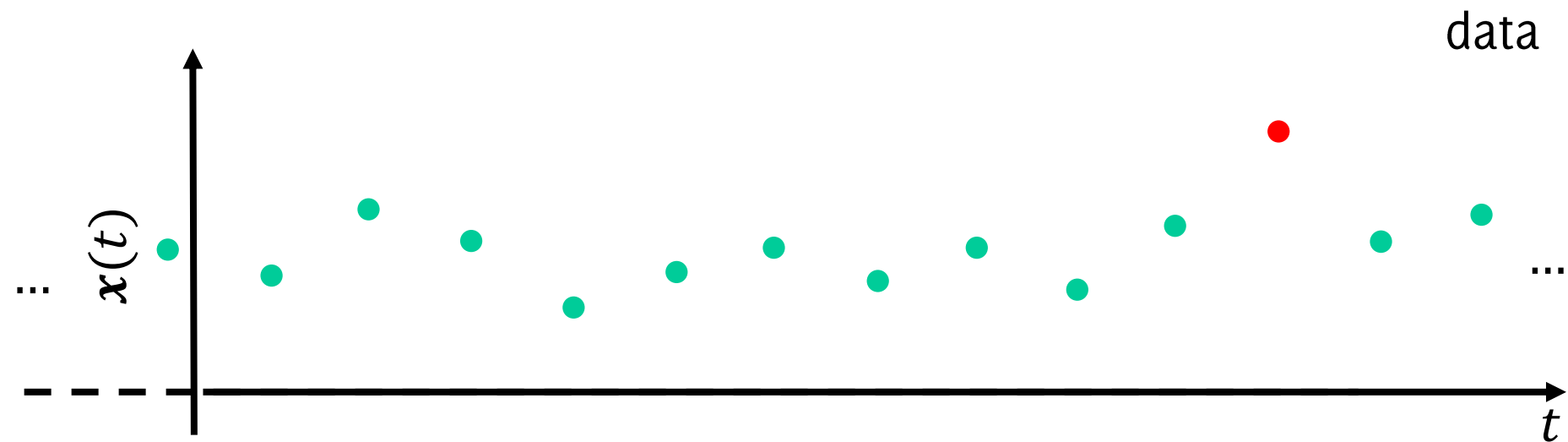


Most algorithms are composed of:

- 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)

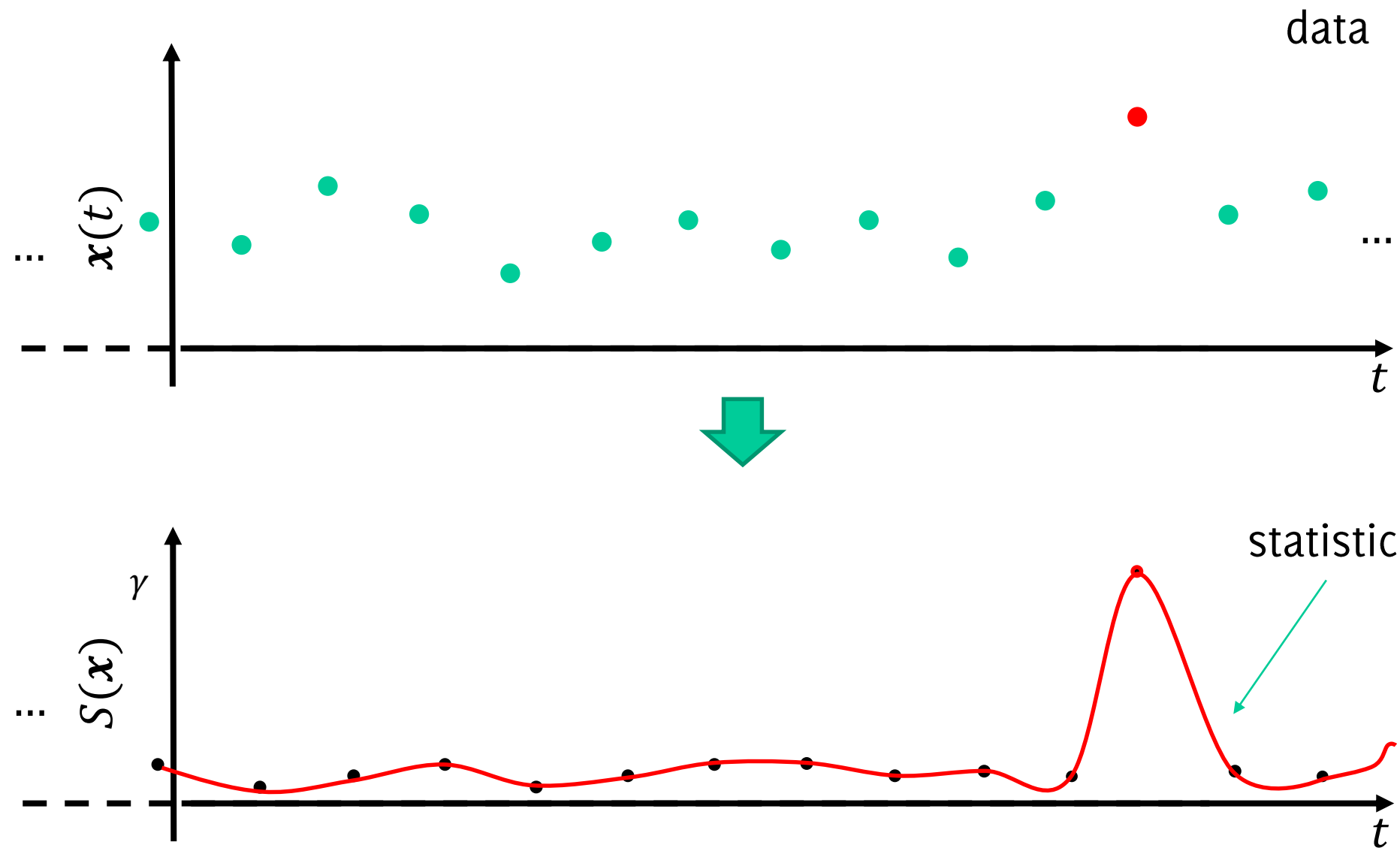


THE TYPICAL SOLUTIONS



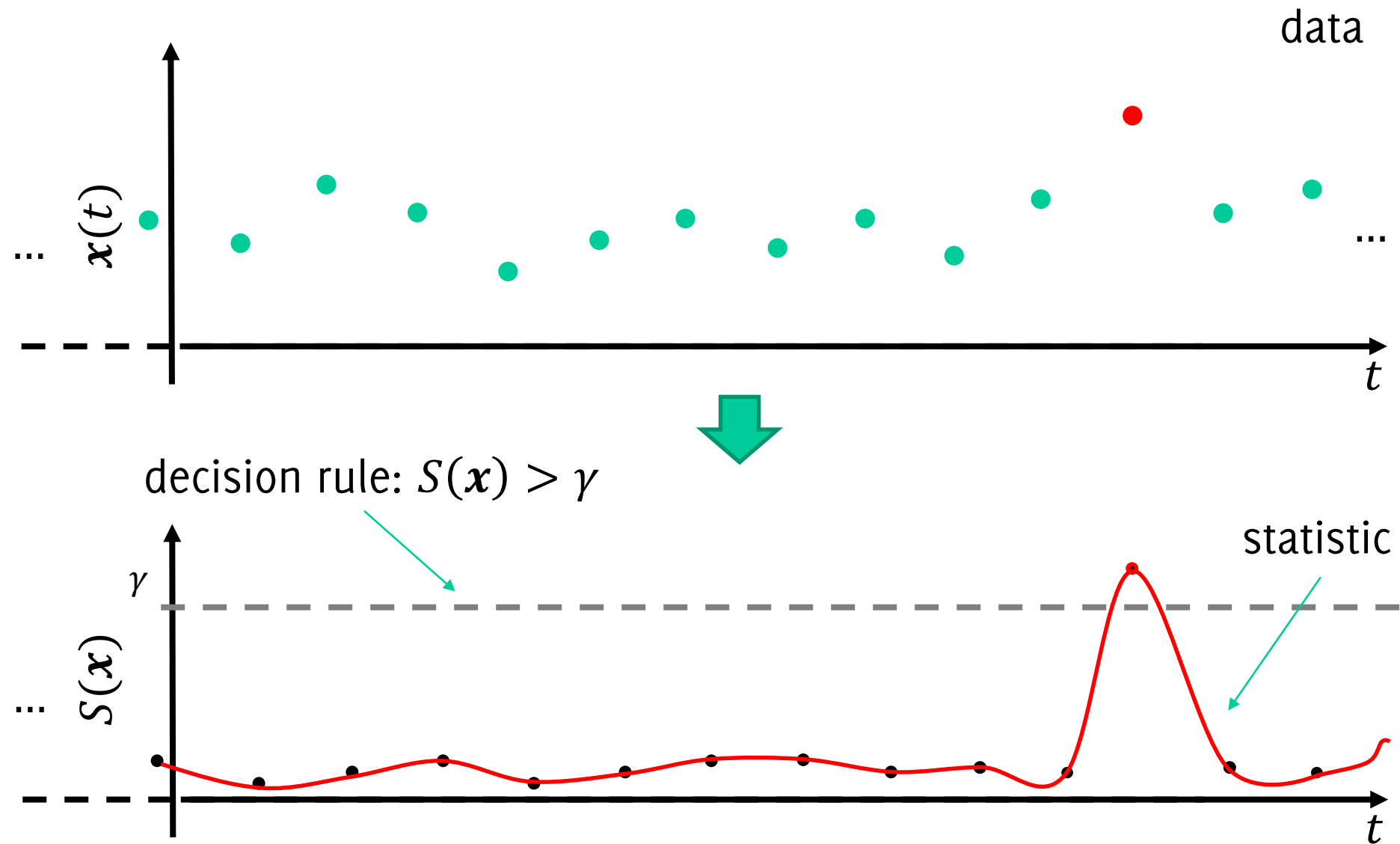


THE TYPICAL SOLUTIONS





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



Performance Measures

How to assess performance of
change/anomaly detection algorithms



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



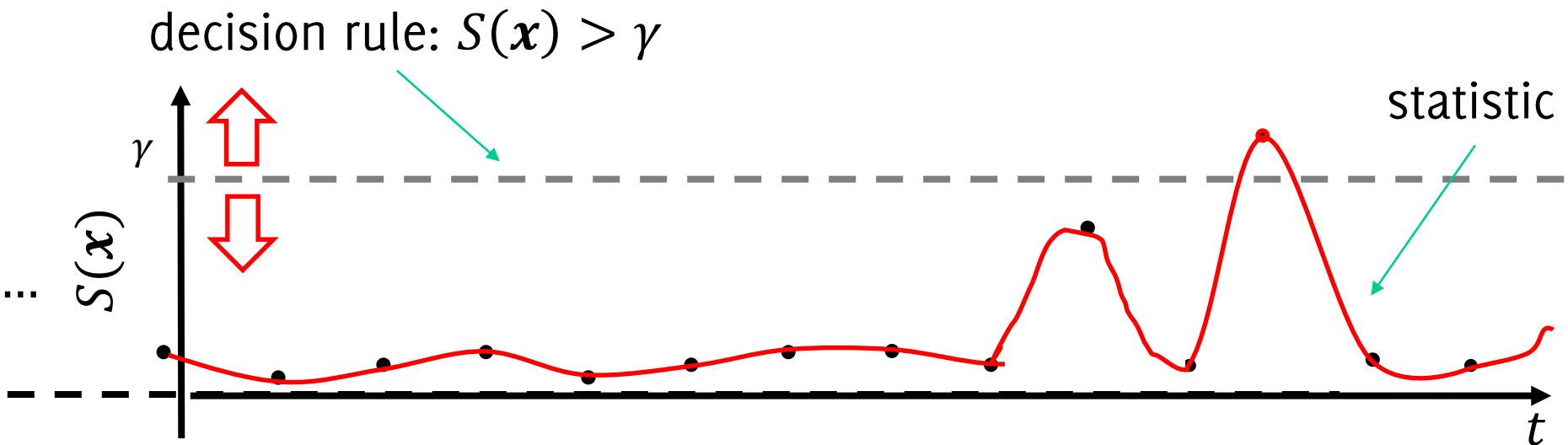
ANOMALY-DETECTION PERFORMANCE

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



THE TYPICAL SOLUTIONS

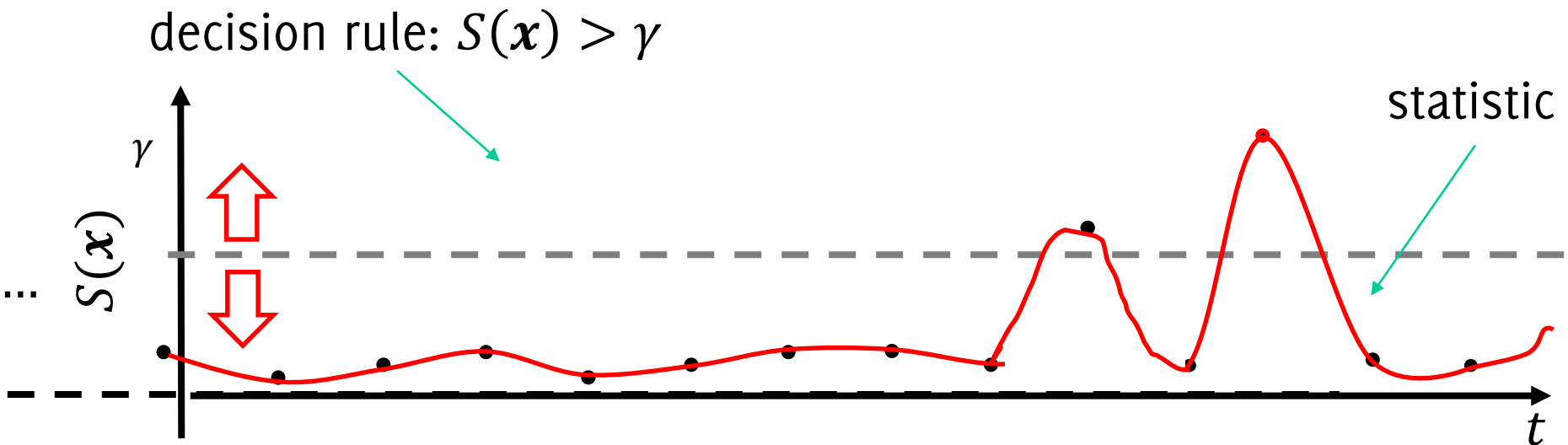
By changing γ it is possible to achieve different detection performance (e.g. more true positive, more false positives)





THE TYPICAL SOLUTIONS

By changing γ it is possible to achieve different detection performance (e.g. more true positive, more false positives)





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**

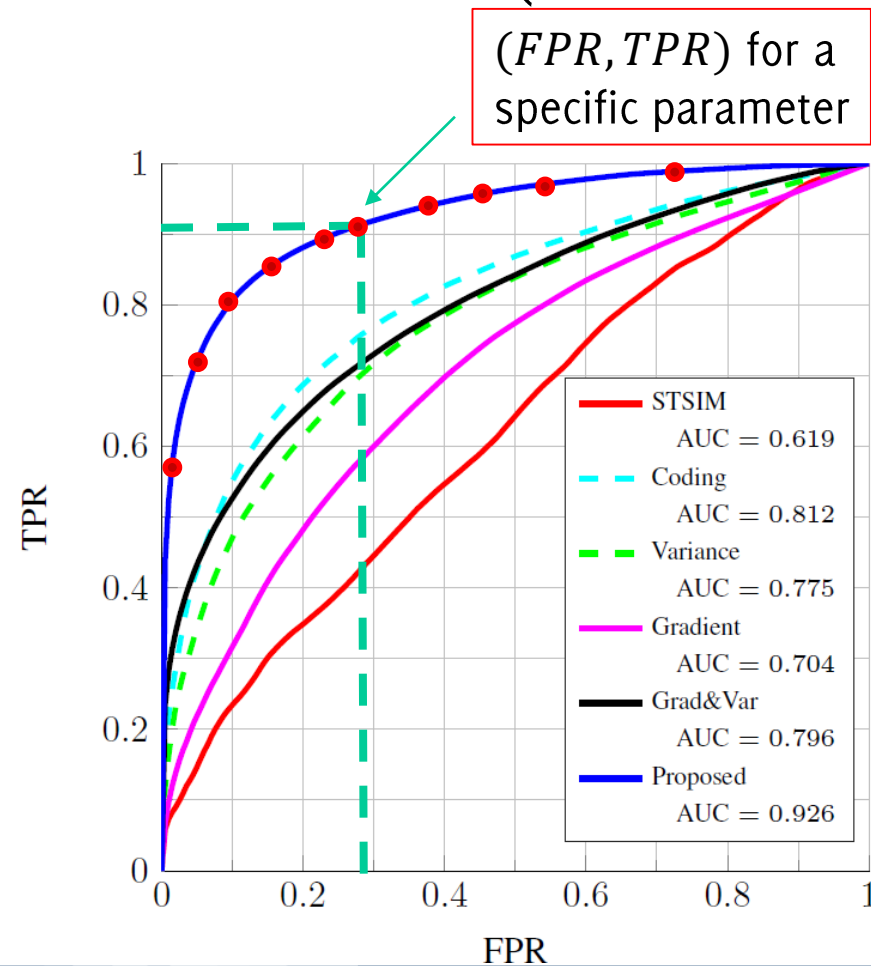
The ideal detector would achieve:

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

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

$$ARL_0 = \mathbb{E}_x[\hat{T} | \phi_0]$$

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

$$ARL_1 = \mathbb{E}_x[\hat{T} | \phi_1]$$

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



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**,

$$DD = \mathbb{E}_x[\hat{T} - \tau \mid \hat{T} \geq \tau, \phi_1]$$

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, and here depend on the sequence length

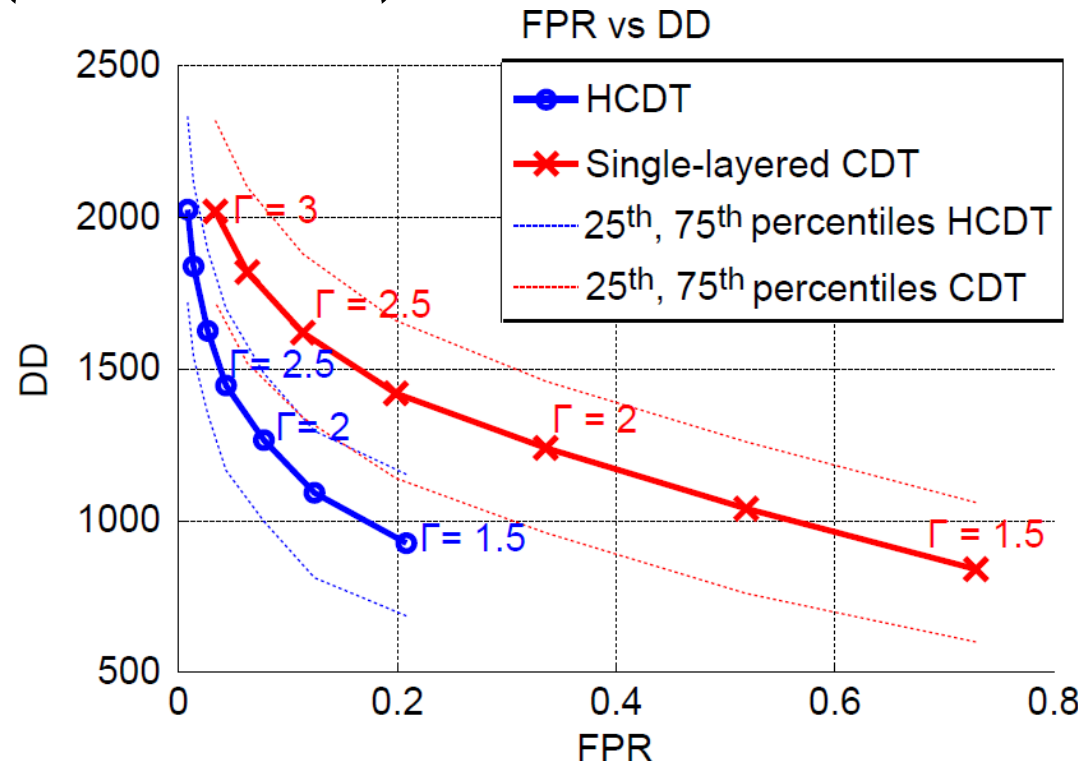



CHANGE-DETECTION PERFORMANCE

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





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

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

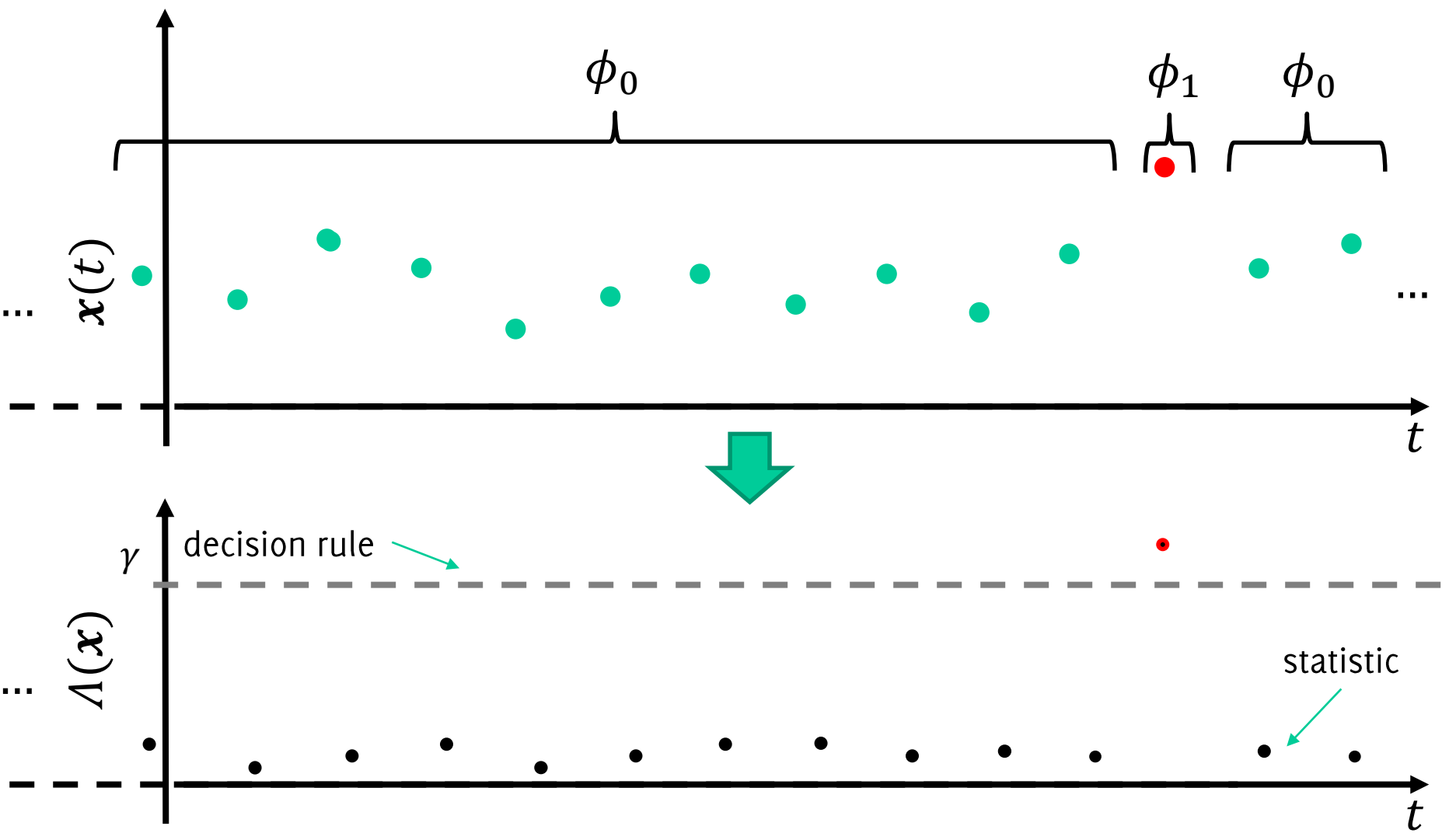
According to the Neumann Pearson lemma, the most powerful **statistic** to detect changes is the **likelihood ratio**

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

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 $\Lambda(\mathbf{x})$





THE CUSUM TEST ON THE LIKELIHOOD RATIO

CUSUM involves the calculation of a **C**umulative **S**UM, which makes it a sequential monitoring scheme.

It can be applied to the log-likelihood ratio:

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

The CUSUM statistic is:

$$S(t) = \max\left(0, S(t-1) + \log(\Lambda(x(t)))\right)$$

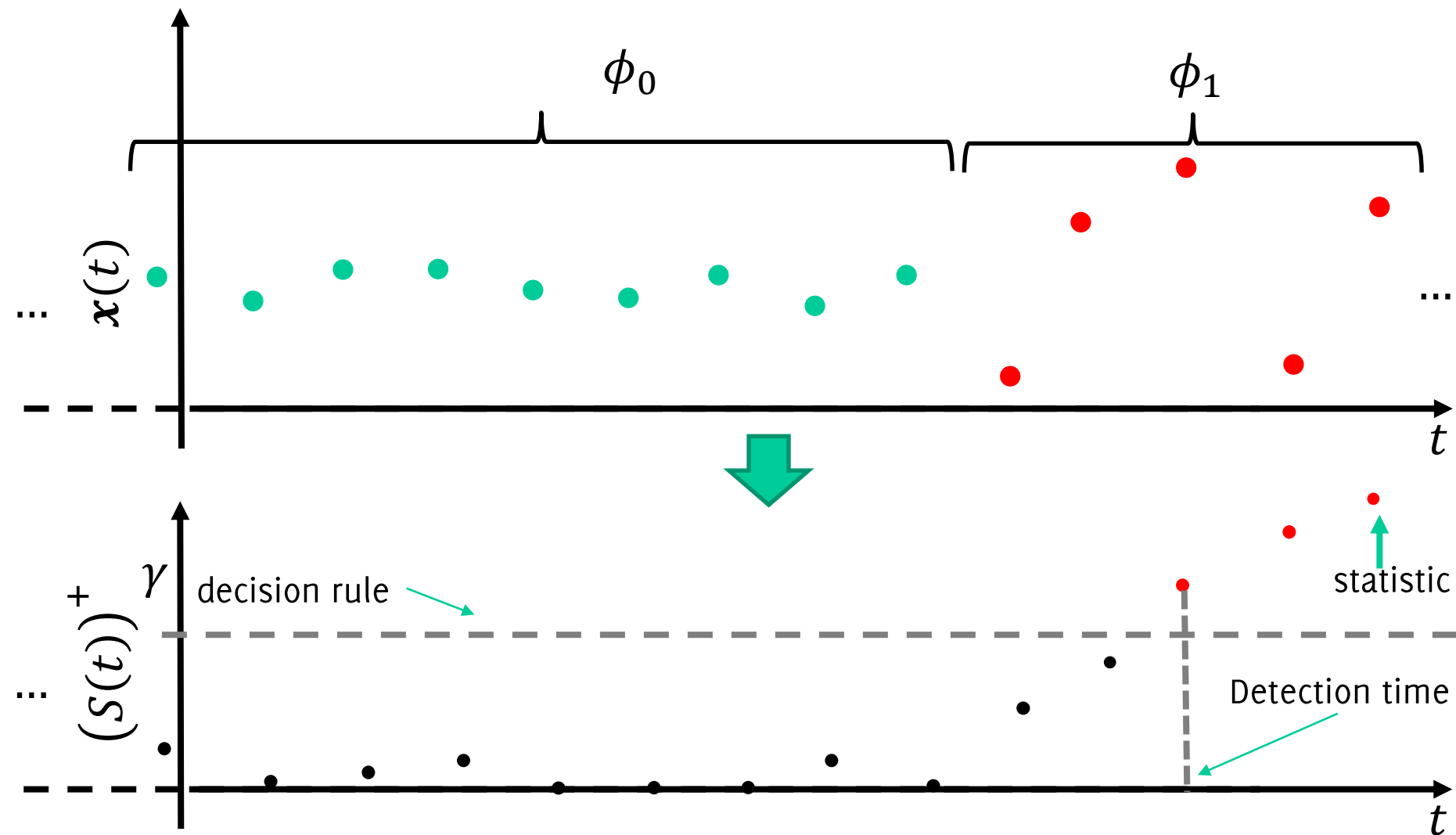
And the decision rule is

$$S(t) > \gamma$$



CUSUM TEST

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





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



Anomaly Detection: More Realistic Settings

...anomaly detection when ϕ_0 and ϕ_1 are unknown



DATA DISTRIBUTION IS 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.



DATA DISTRIBUTION IS 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.



SUPERVISED ANOMALY DETECTION - SOLUTIONS

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

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

Solution:

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

During training:

- Learn a classifier \mathcal{K} from TR .

During testing:

- compute the classifier output $\mathcal{K}(\mathbf{x})$ / set a threshold on the posterior $p_{\mathcal{K}}(-|\mathbf{x})$ / select the k −most likely anomalies

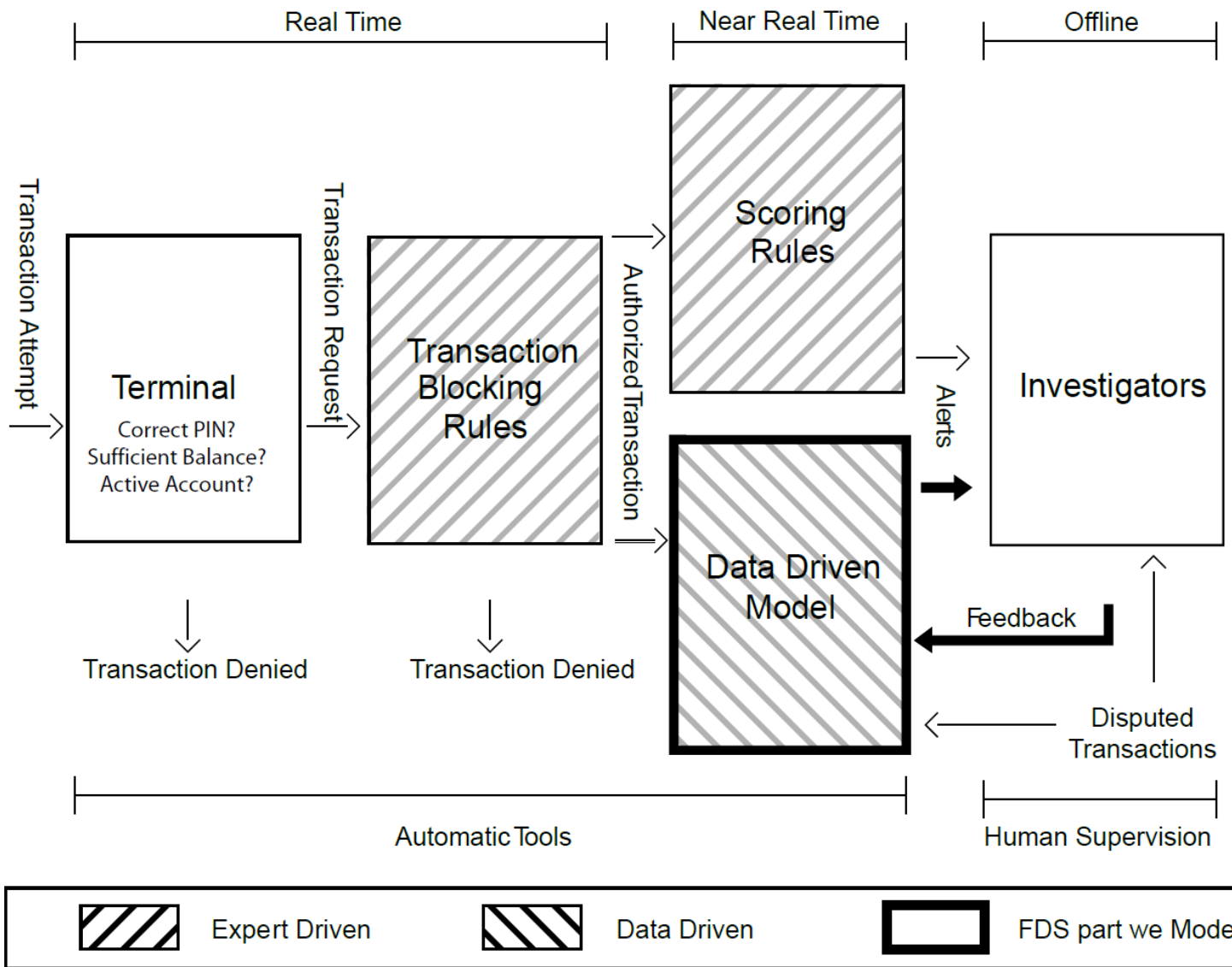


These **classification problems are challenging** because these anomaly-detection settings typically imply:

- **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 annotation is performed by controlling the detected anomalies, and this biases the selection of training samples.

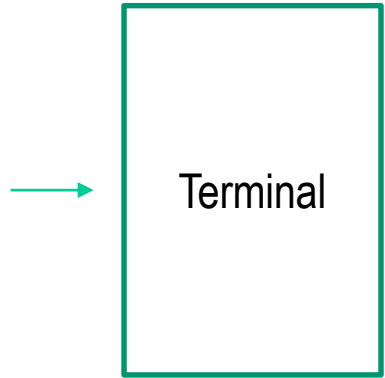


SUPERVISED ANOMALY DETECTION – AN EXAMPLE





THE TERMINAL



Purchase



Acceptance checks like:

- Correct PIN
- Number of attempts
- Card status (active, blocked)
- Card balance / availability

are immediately performed.

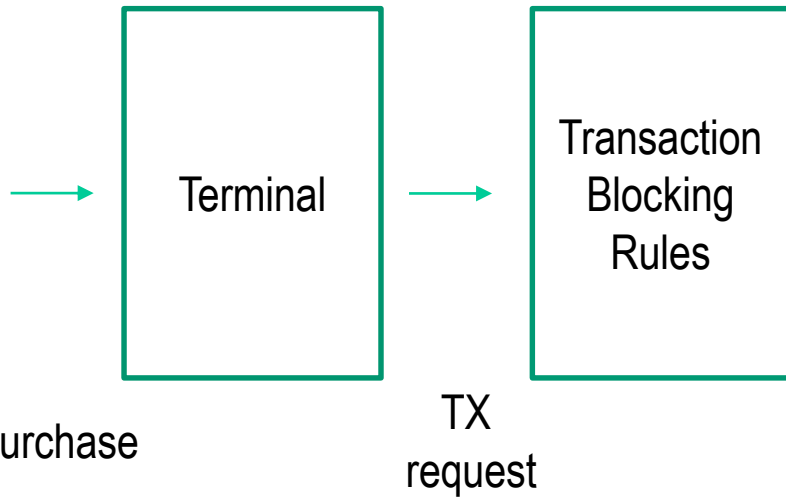
These checks are done in **real time**, and **preliminary filter** our purchases: when these checks are not satisfied, the card/transaction can be blocked.

Otherwise, a **transaction request** is entered in the system that include information of the actual purchase:

- *transaction amount, merchant id, location, transaction type, date time, ...*



BLOCKING RULES





TRANSACTION BLOCKING RULES

Association rules (if-then-else statements) like*

IF Internet transactions AND compromised website THEN deny the transaction

These rules:

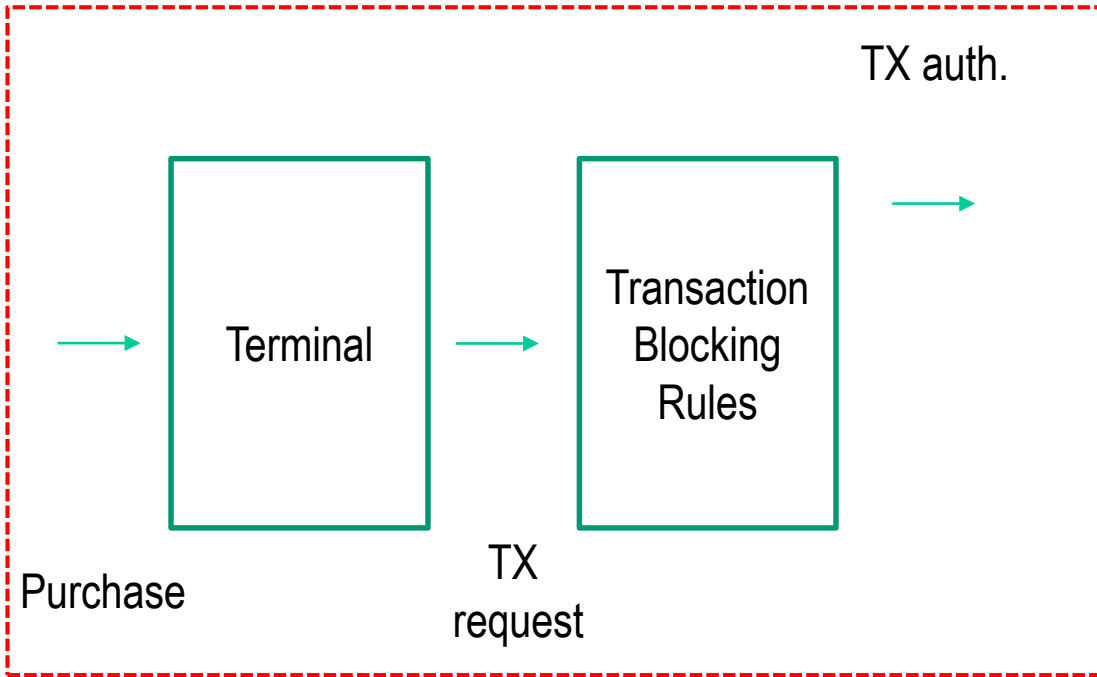
- are **expert-driven**, designed by investigators
- involves quite simple expressions with a few data
- are easy to interpret
- have always «deny the transaction» as statement
- are executed in real time

All the transaction RX passing these rules are **authorized transactions** and further analyzed by the FDS

(*) Transaction blocking rules are confidential and this is just a reasonable example



REAL TIME PROCESSING



Real time



FEATURE AUGMENTATION

A feature vector \mathbf{x} is associated to each authorized transaction.

The components of \mathbf{x} include data about the current transaction and customary shopping habits of the cardholder, e.g.:

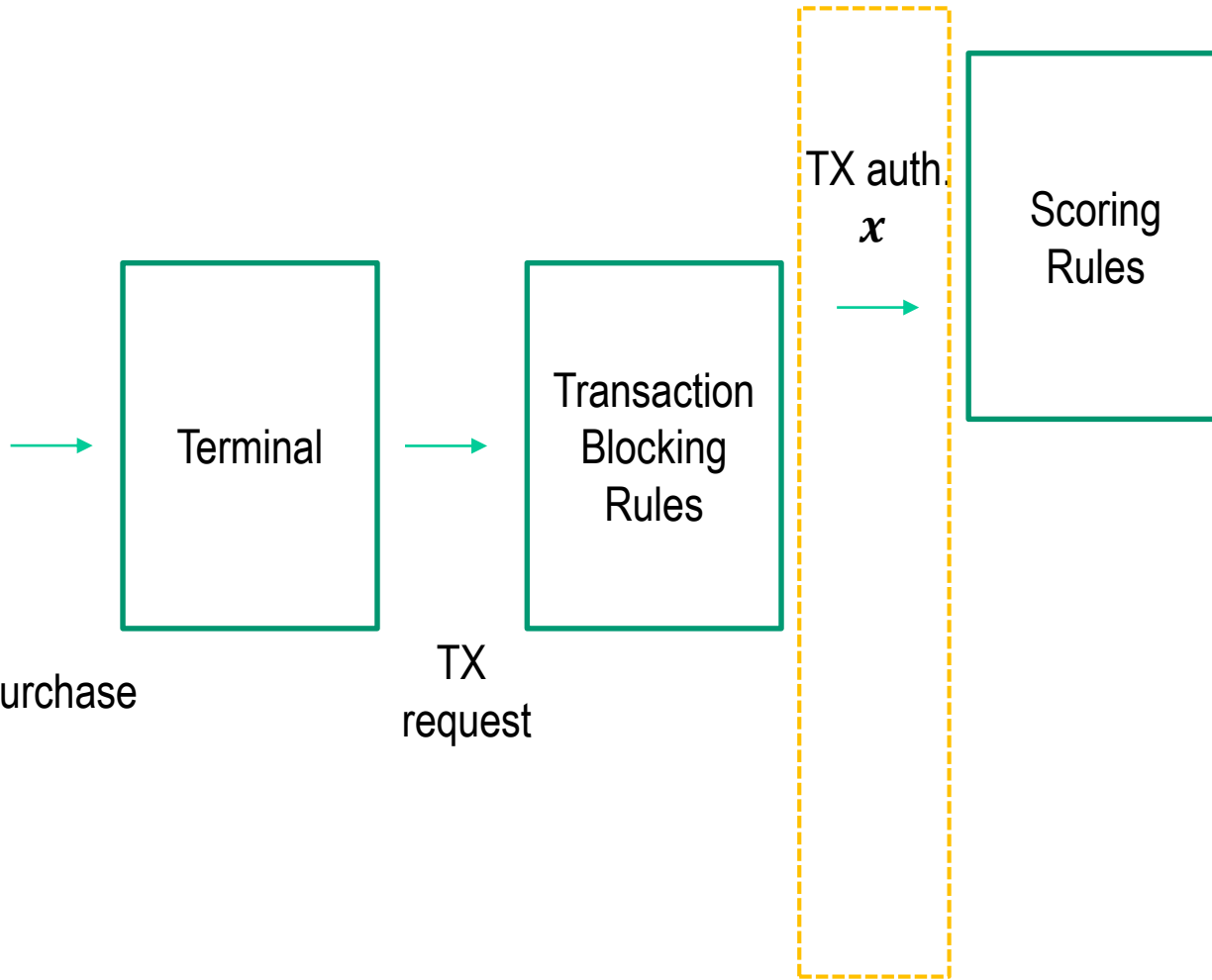
- the average expenditure
- the average number of transactions per day
- the cardholder age
- the location of the last purchases
- ...

and are very informative for fraud-detection purposes

Overall, about 40 features are extracted in near-real time.



SCORING RULES



Feature Augmentation



SCORING RULES

Scoring rules are if-then-else statements that:

- are being processed in near-real time
- are **expert-driven**, designed by investigators.
- Operate on augmented features (components of \mathbf{x})
- Assign a **score**: the larger the score the more risky the transaction (an estimate of the probability for \mathbf{x} to be a fraud, according to investigator expertise)
- Feature vector receiving large scores are alerted
- Are easy to interpret and are designed by investigators



SCORING RULES

Examples* of scoring rules might be:

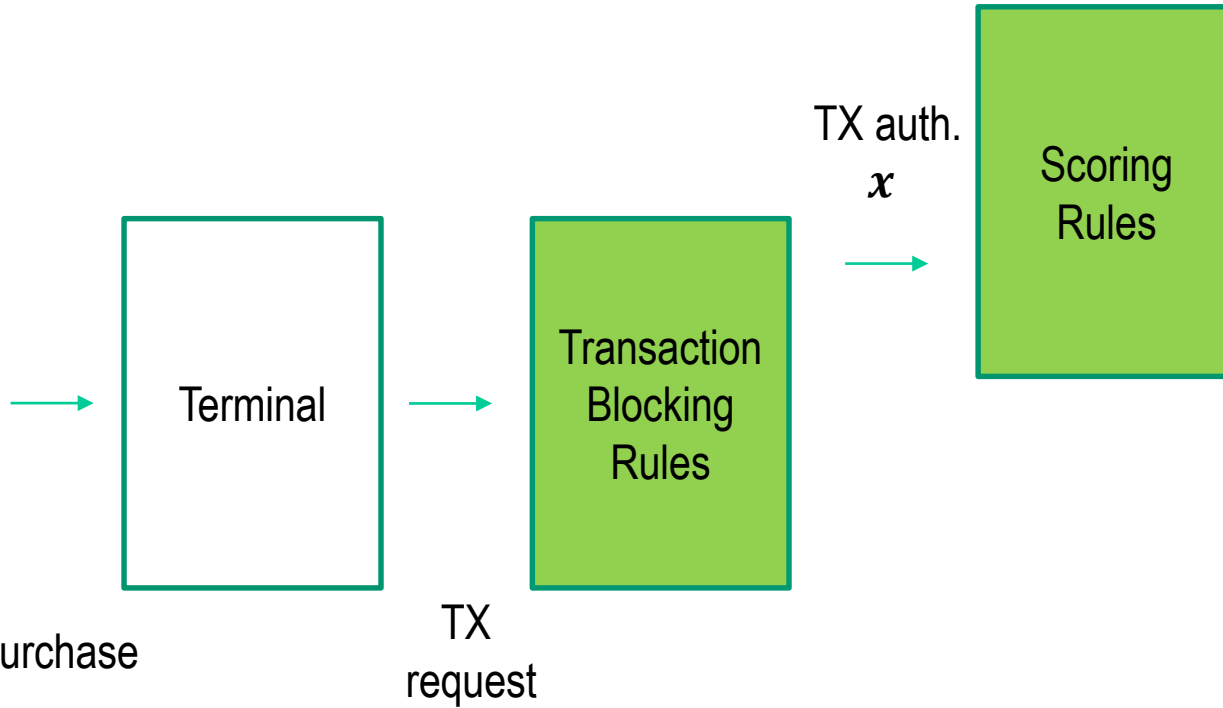
- *IF previous transaction in a different country AND less than 2 hours since the previous transaction, AND operation using PIN THEN fraud score = 0.95*
- *IF amount > average of transactions + 3σ AND country is a fiscal paradise AND customer travelling habits low THEN fraud score = 0.75*

(*) Scoring rules are confidential and these are just a reasonable examples



EXPERT DRIVEN MODELS IN THE FDS

 Expert-driven





EXPERT-DRIVEN VS DATA-DRIVEN MODELS

Scoring rules are an **expert-driven model**, thus:

- Can detect **well-known / reasonable** frauds
- Involve **few components** of the feature vector
- **Difficult** to **exploit correlation** among features



EXPERT-DRIVEN VS DATA-DRIVEN MODELS

Scoring rules are an **expert-driven model**, thus:

- Can detect **well-known / reasonable** frauds
- Involve **few components** of the feature vector
- **Difficult** to **exploit correlation** among features

Fraudulent patterns can be directly **learned from data**, by means of a **data-driven model** (DDM). This should:

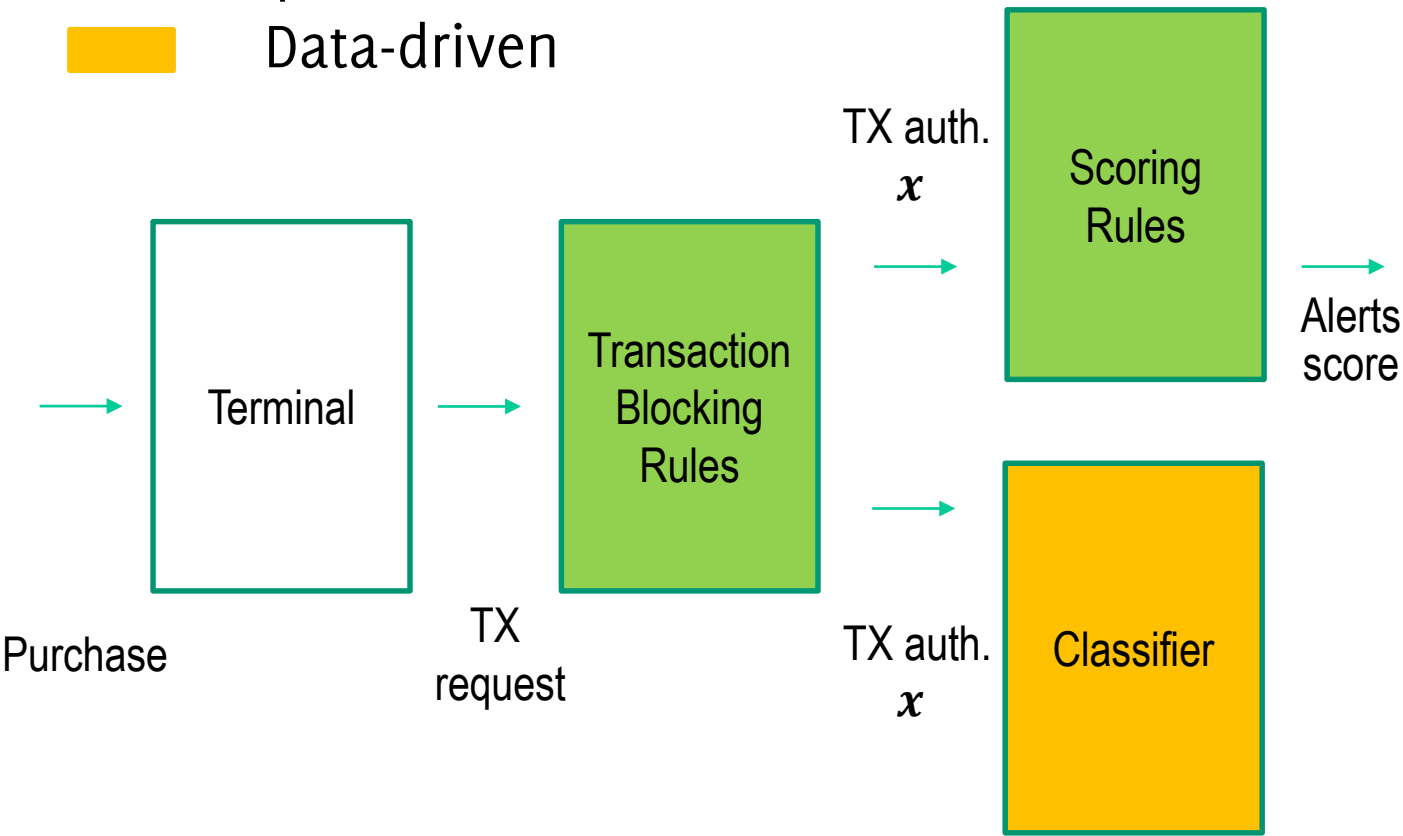
- Simultaneously analyze **several components** of the feature vector
- Uncover **complex relations among features** that cannot be identified by investigator

These relations can be meaningful for separating frauds from genuine transactions



ALERTS GENERATION

- Expert-driven
- Data-driven





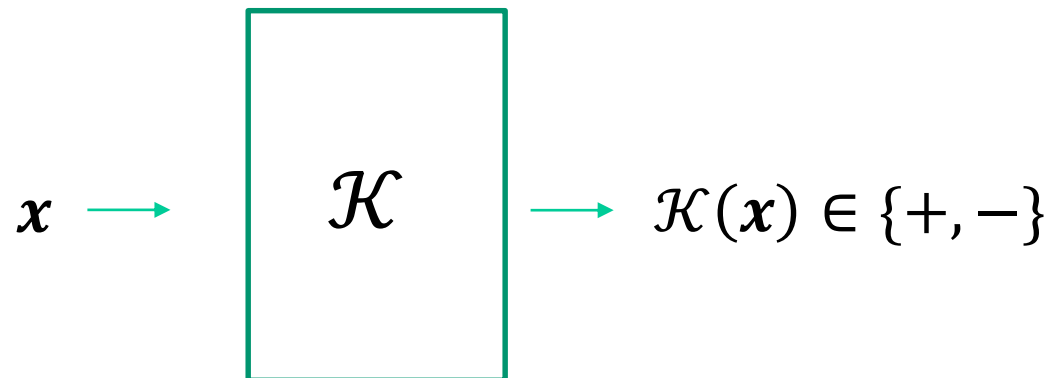
THE CLASSIFIER IN THE FDS

A classifier \mathcal{K} is learned from a **training set** that contains:
labeled feature vectors

$$TR = \{(\mathbf{x}, y)_i, i = 1, \dots, N\}$$

where the label $y = \{+, -\}$, i.e., $\{\langle\textit{fraud}\rangle, \langle\textit{genuine}\rangle\}$

In practice, the classifier \mathcal{K} then can assign a label, $+$ or $-$ to each incoming feature vector \mathbf{x}



\mathcal{K} considers transactions labeled as '+' as frauds

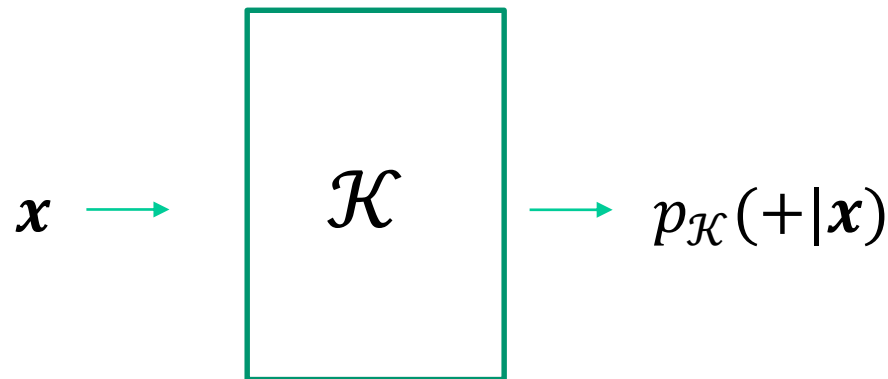


ALERTS REPORTED TO INVESTIGATORS

It is not feasible to alert all transactions labeled as frauds

Only few transactions that are **very likely** to be frauds can be alerted.

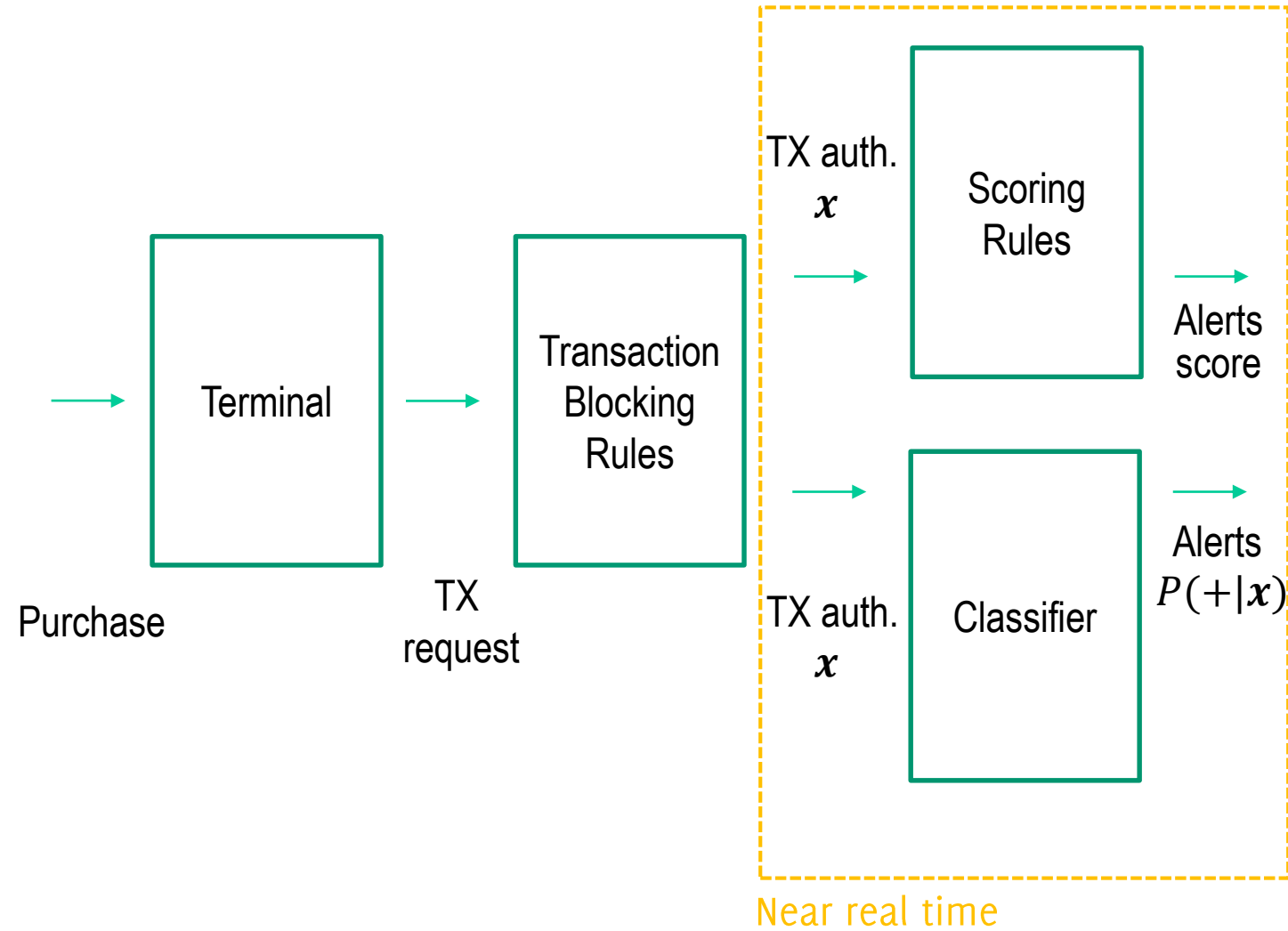
Thus, the FDS typically consider $p_{\mathcal{K}}(+|\mathbf{x})$, an **estimate of the probability** for \mathbf{x} to be a fraud according to \mathcal{K}



and only transactions yielding $p_{\mathcal{K}}(+|\mathbf{x}) \approx 1$ raise an alert

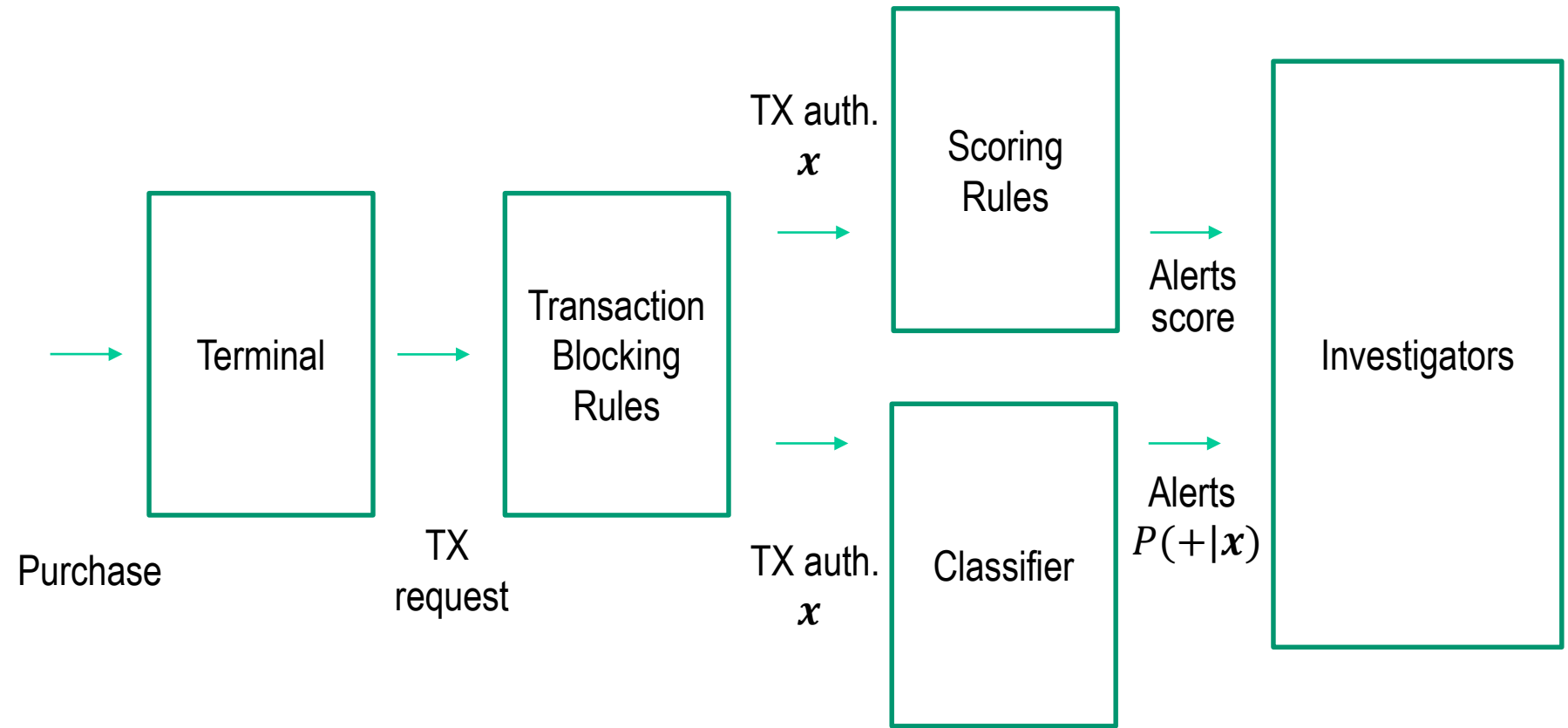


NEAR REAL TIME PROCESSING





INVESTIGATORS





INVESTIGATORS

Investigators are **professionals** that are experienced in analyzing credit card transactions:

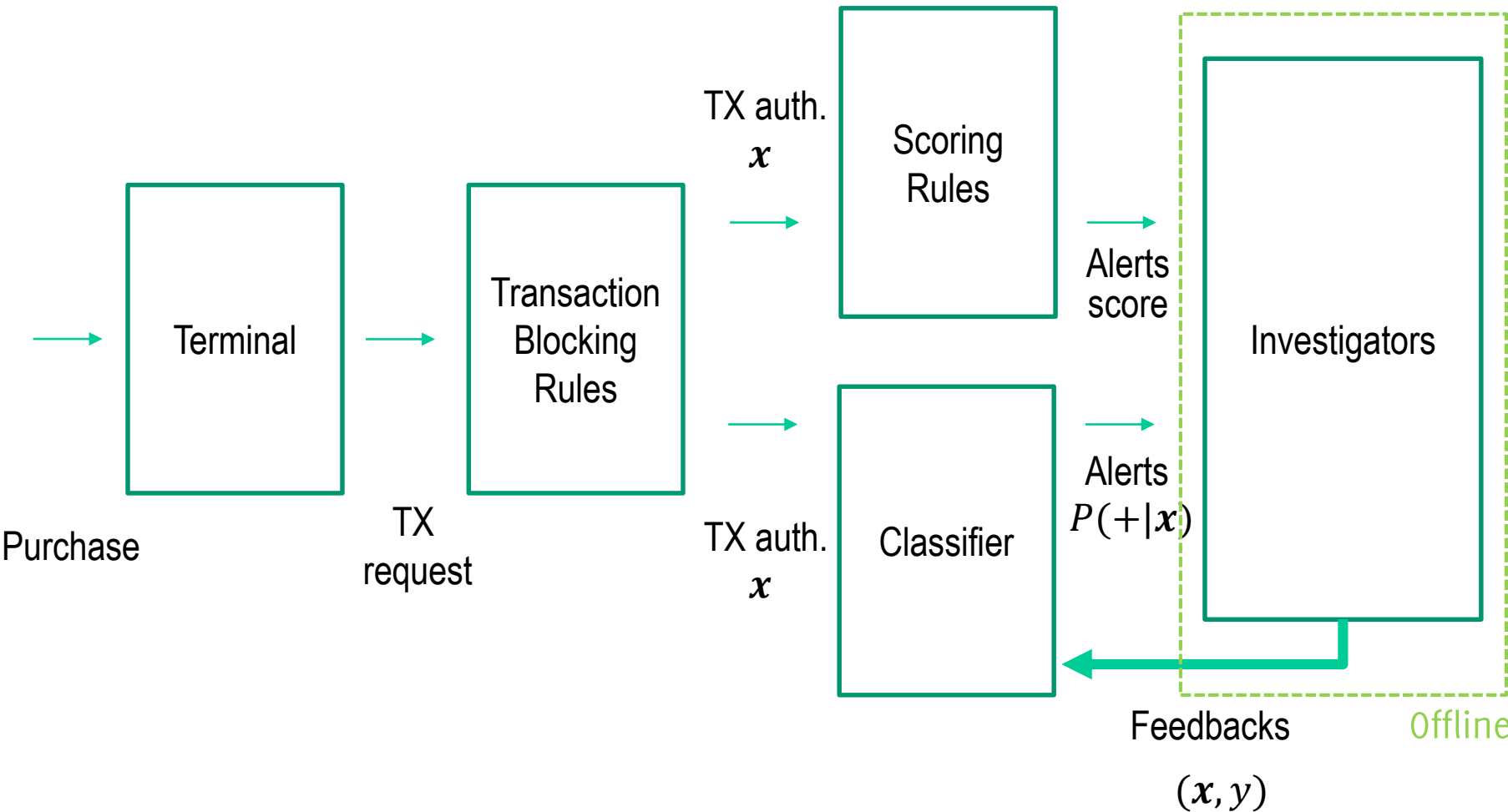
- they **design blocking/scoring rules**
- they **call cardholders** to check whether alerts correspond to frauds
- as soon as they detect a fraud, they block the card
- they annotate the **true label** of checked transactions

The labels associated to transactions comes in the form of **feedbacks** and can be used to re-train/update \mathcal{K}

Given the limited number of investigators, the large number of transactions, the multiple sources of alerts, etc ... it is important to provide **very precise alerts**



OFFLINE PROCESSING





SUPERVISED ANOMALY DETECTION – AN EXAMPLE

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

Sampling Selection Bias:

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

Class Imbalance:

- Frauds are typically less than 1% of genuine transactions

Concept Drift:

- Fraudster always implement new strategies



SUPERVISED ANOMALY DETECTION – AN EXAMPLE

Since feedbacks and delayed samples are *very different*, a better solution for training the classifier \mathcal{K} consists in:

- Training a classifier \mathcal{F} on feedbacks
- Training a classifier \mathcal{D} on delayed samples
- Detect frauds by aggregating their posteriors

$$p_{\mathcal{K}}(-|\mathbf{x}) = \alpha p_{\mathcal{F}}(-|\mathbf{x}) + (1 - \alpha)p_{\mathcal{D}}(-|\mathbf{x})$$

Caveat: When testing a fraud-detection algorithms the alert-feedback interaction should be considered.

Few dataset provided (see below).



DATA DISTRIBUTION IS 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.



SEMI-SUPERVISED ANOMALY DETECTION

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

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

Very practical assumptions:

- **Normal data** are often **easy to gather**
- **Anomalous data** are **difficult/costly to collect/select** and it would be **difficult to gather a representative training set**
- **New anomalies** might appear than those ones in TR

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**



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

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

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

During testing:

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



Advantages:

- $\hat{\phi}_0(\mathbf{x})$ yields a **confidence associated to each decision**
- If the density estimation process is robust to outliers, it is possible to tolerate few anomalous samples in TR
- **Histograms are simple to compute** in relatively small dimensions

Challenges:

- **Fitting complex models in high-dimensional data** might be challenging
- Histograms traditionally suffer from **curse of dimensionality** when d increases
- Often the **1D histograms** of the marginals are monitored, **ignoring the correlations** among components



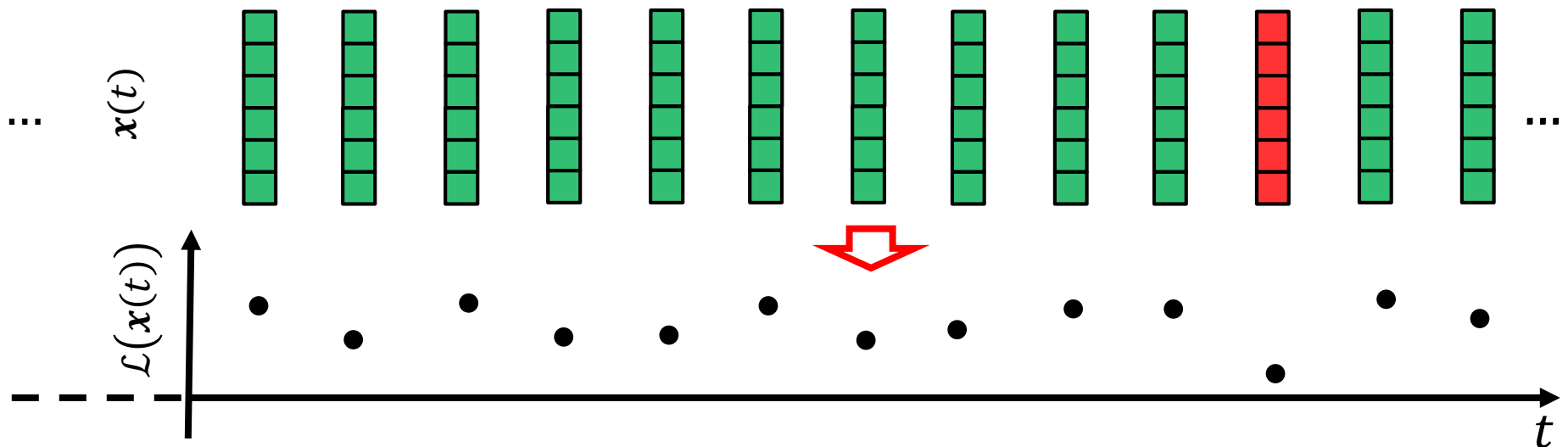
DENSITY-BASED METHODS: MONITORING THE LOG-LIKELIHOOD

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

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

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

3. Monitor $\{\mathcal{L}(\mathbf{x}(t)), t = 1, \dots\}$





DENSITY-BASED METHODS: MONITORING THE LOG-LIKELIHOOD

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

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

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

3. Monitor $\{\mathcal{L}(\mathbf{x}(t)), t = 1, \dots\}$

This is quite a popular approach in either anomaly-detection and sequential monitoring 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



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)

Pimentel, M. A., Clifton, D. A., Clifton, L., Tarassenko, L. "*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 data-density is large.**

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

$$f(\mathbf{x}) = \text{sign}(\langle w, \psi(\mathbf{x}) \rangle - \rho)$$

Where the hyperplane w and the margin $\rho > 0$ is estimated to separate normal data from the origin

A linear separation in the feature space corresponds to a variety of nonlinear boundaries in the input space.



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.**

Similar detection formulas hold, measuring the distance in the feature space between the sphere center and $\psi(\mathbf{x})$ for $\mathbf{x} \in TR$.

The function is always defined by a few 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.



DATA DISTRIBUTION IS 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.



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

One in principle could use:

- 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



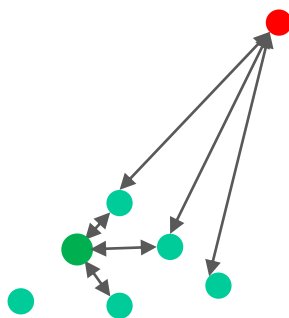
DISTANCE-BASED METHODS

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**



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

Zhao, M., Saligrama, V. "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



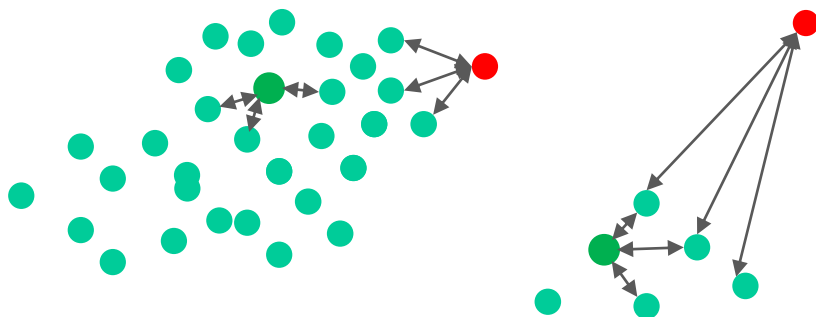
DISTANCE-BASED METHODS

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**





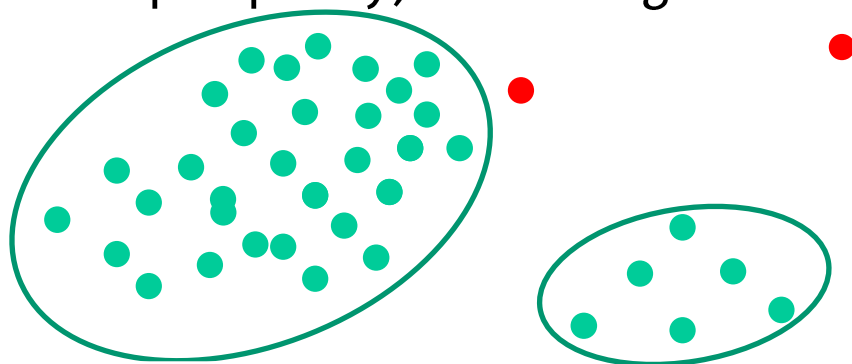
DISTANCE-BASED METHODS

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





Change Detection: More Realistic Settings

...change detection when ϕ_0 and ϕ_1 are unknown



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)

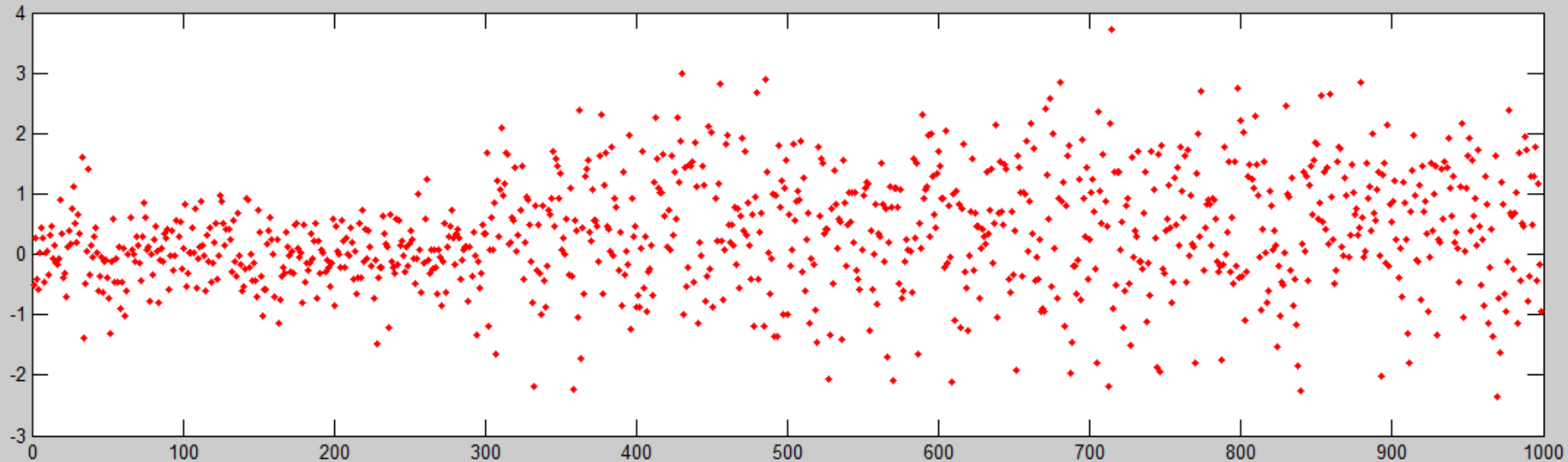
The basic functioning of CPM is illustrated for offline monitoring, but CPM can be **iterated to perform to sequential monitoring**.

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



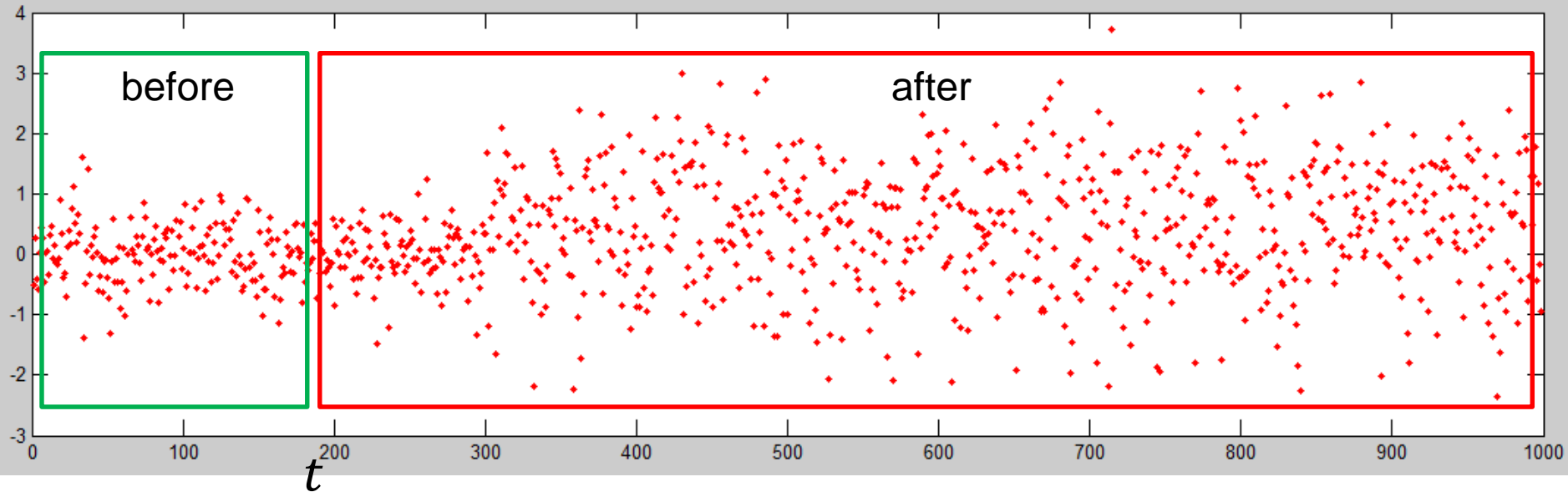
ILLUSTRATION OF CHANGE POINT METHOD (CPM)



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



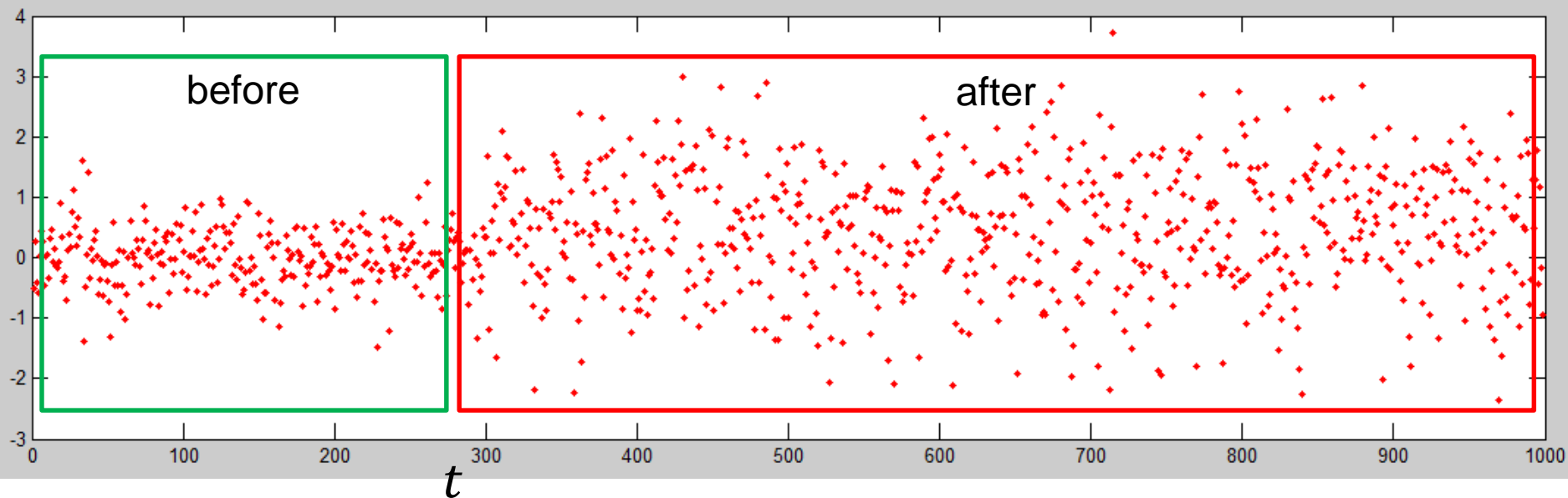
ILLUSTRATION OF CHANGE POINT METHOD (CPM)



- 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



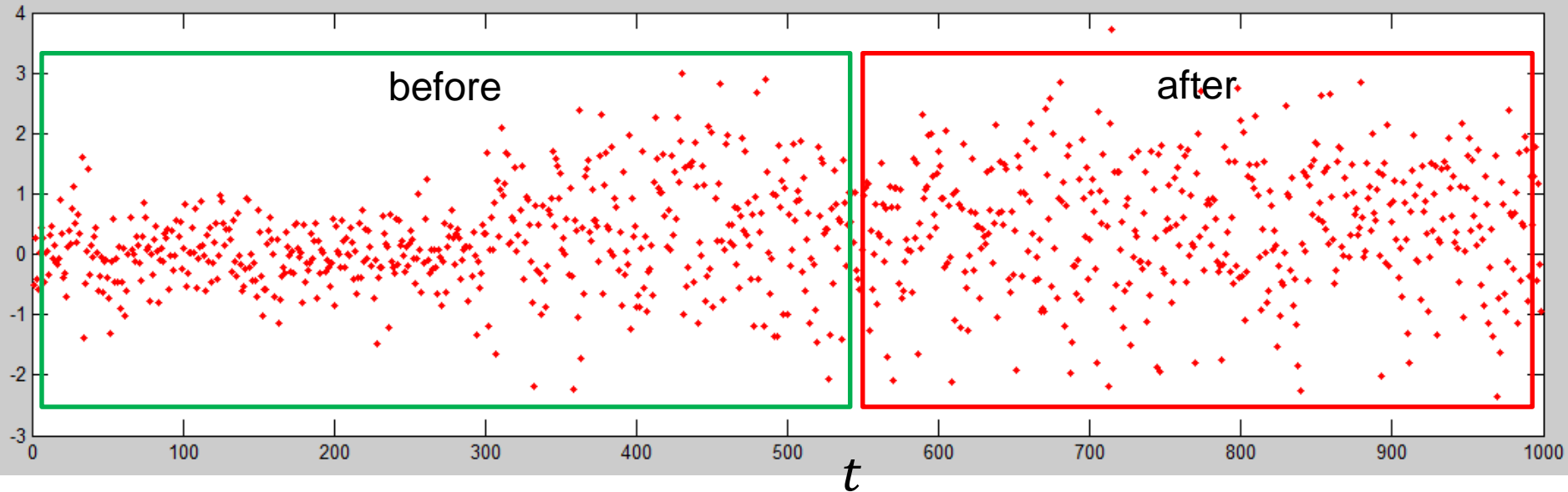
ILLUSTRATION OF CHANGE POINT METHOD (CPM)



- 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



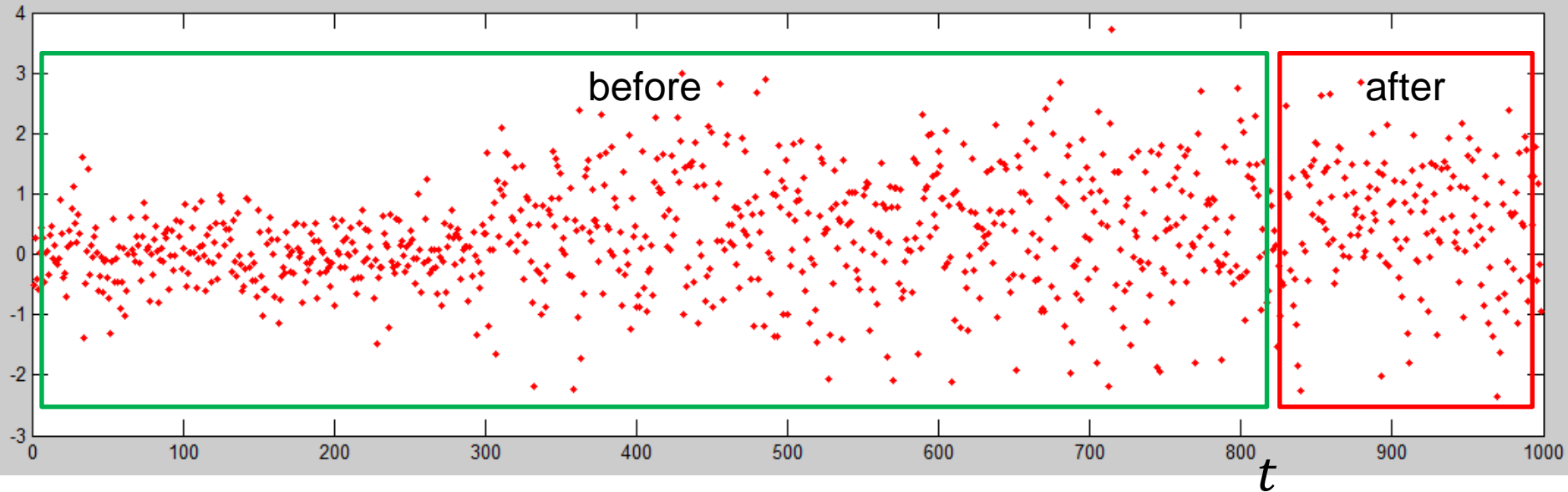
ILLUSTRATION OF CHANGE POINT METHOD (CPM)



- 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



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



ILLUSTRATION OF CHANGE POINT METHOD (CPM)

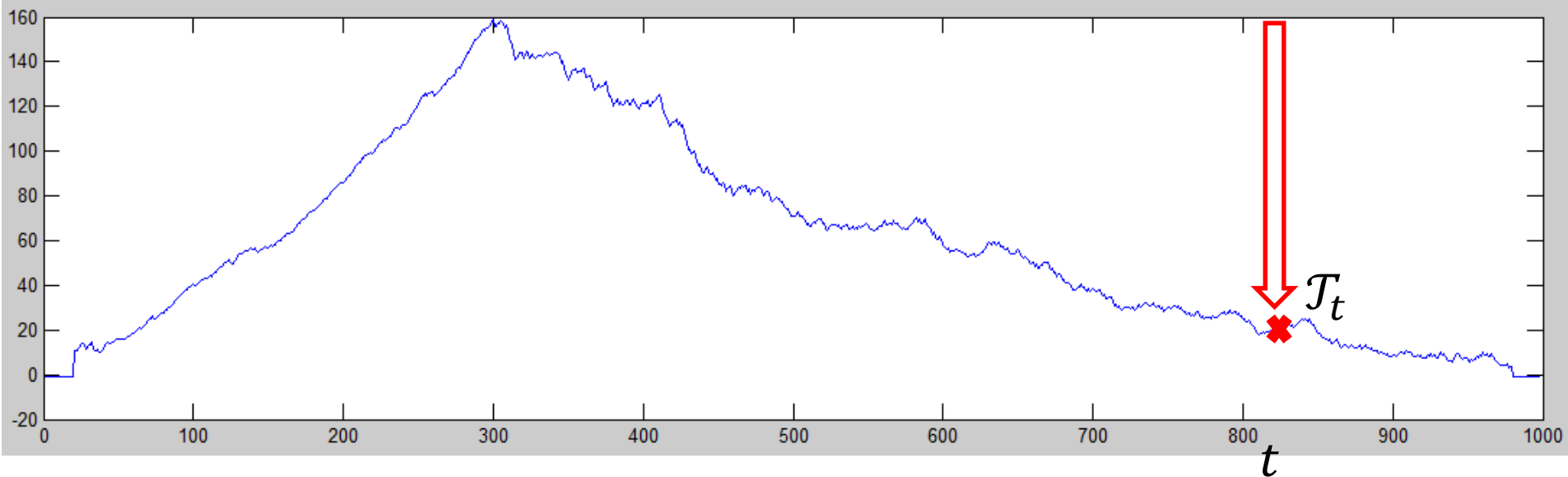
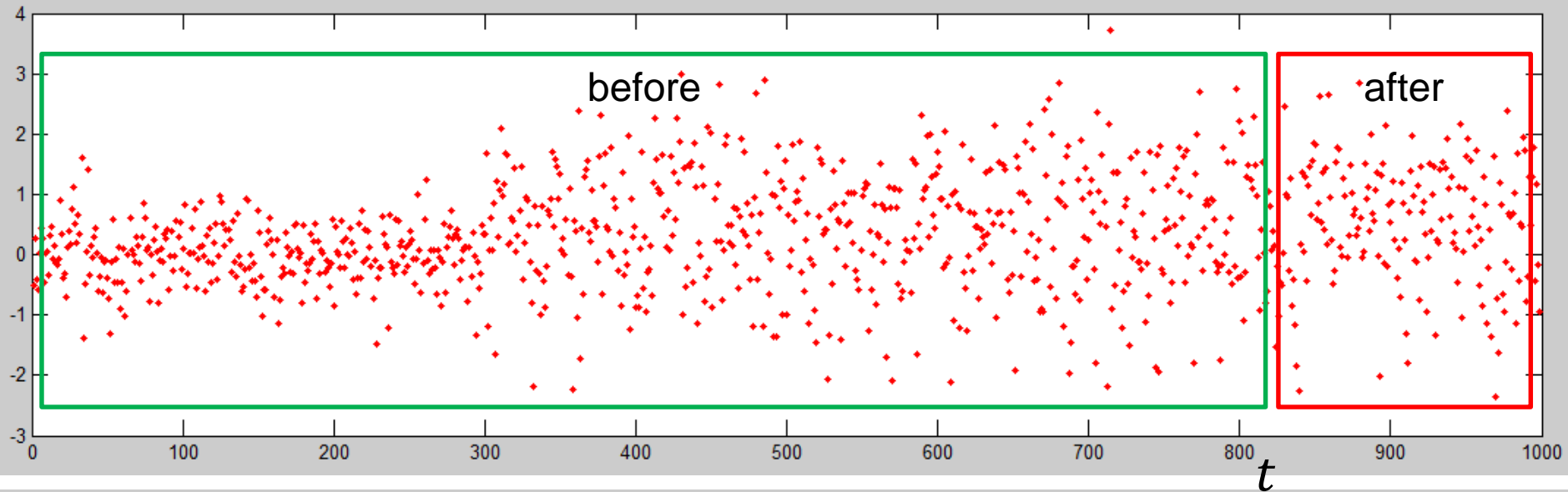


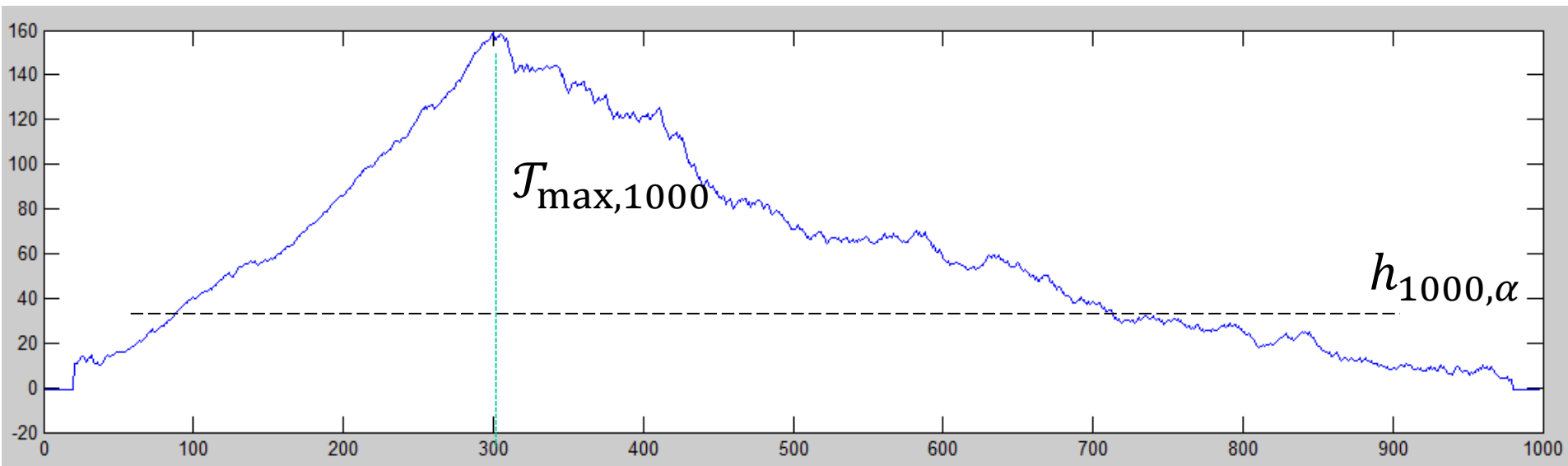


ILLUSTRATION OF CHANGE POINT METHOD (CPM)

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

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

The CPM framework can be extended to online monitoring, and in this case it is possible to control the ARL_0





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 $\phi_0 \rightarrow \phi_1$ 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 .

Pro: CPMs do not require training samples

Cons: None of these statistic can be used on multivariate data.



CHANGE DETECTION APPROACHES

Parametric Settings:

- The Change-Point Formulation

Non-parametric Settings:

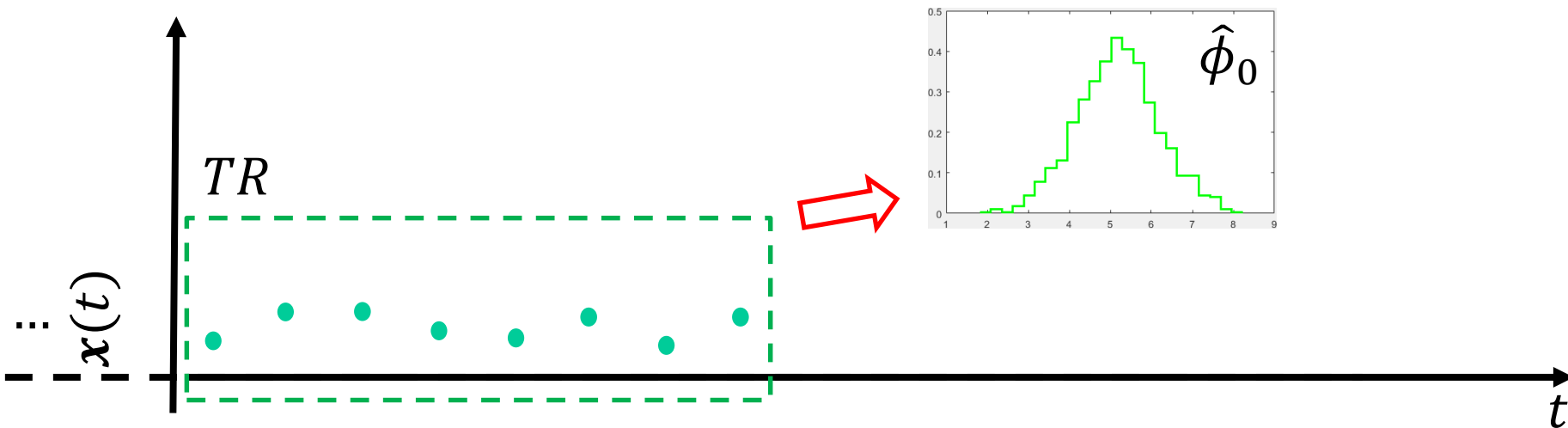
- The Change-Point Formulation
- Change-Detection by Histograms
- Change-Detection by Monitoring Features
- Hierarchical Change-Detection Tests



CHANGE DETECTION BY MEANS OF HISTOGRAMS

Most 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"



HISTOGRAMS

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

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

Where $\{S_k\}_k$ is a disjoint covering 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 an 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 $N = \#X$

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.

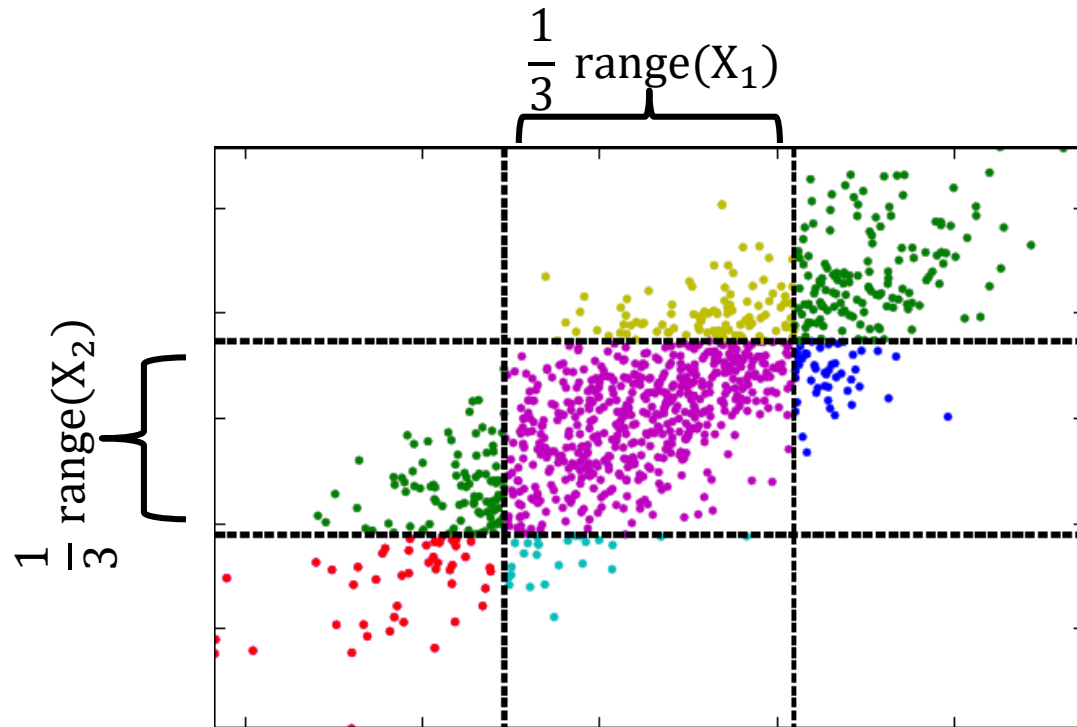
Build a tessellation of $\text{supp}(X)$ by splitting each component in q equally sized parts.

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 $\text{supp}(X)$

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

being $K = q^d + 1$



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



HISTOGRAMS YIELDING UNIFORM DENSITY

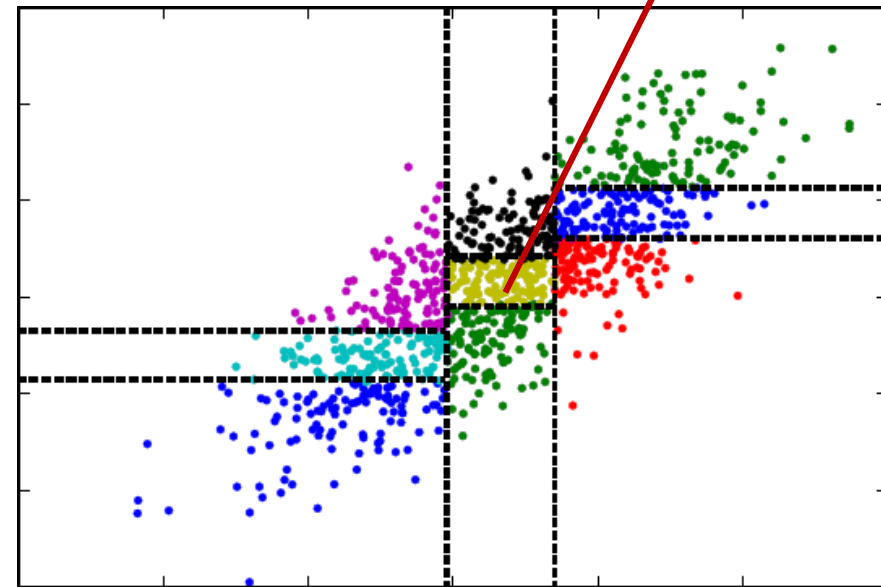
Define the partition $\{S_k\}_k$ in such a way that all the subsets have the **uniform density**, i.e.,

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

Such that each of the q^d hyper-rectangles contains the same number of points $\frac{N}{9}$ 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 $q = 1/3$



CHANGE DETECTION BY HISTOGRAMS: MONITORING SCHEME

Two major monitoring schemes using histograms:

- Likelihood-based methods
- Distance-based methods

whose applicability depends on the type of histogram

Pros: often histograms can be scanned very efficiently as binary trees with splits over a single component

Cons: when d increases, some partitioning schemes are not viable as they require q^d bins. Need to move to a k-d tree or other partitioning schemes



LOG-LIKELIHOOD – BASED MONITORING SCHEME

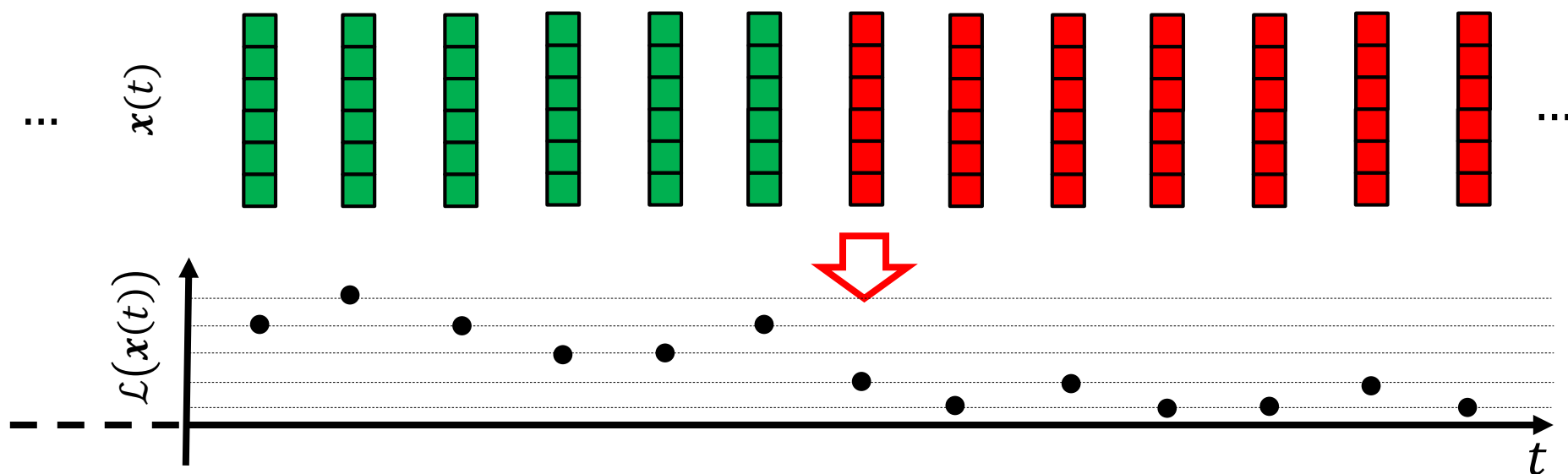
As in density-based methods, $\hat{\phi}_0$ can be used to compute the log-likelihood, which can be then monitored by univariate CDT

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

2. During testing, compute

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

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

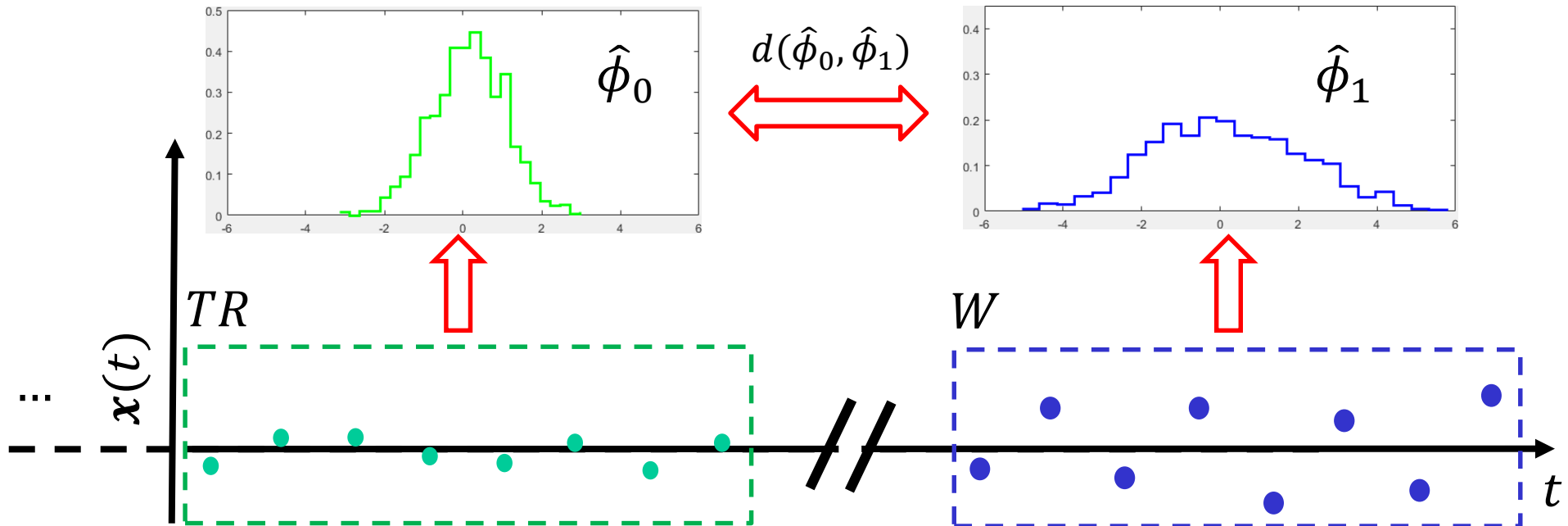




DISTANCE - BASED MONITORING SCHEME

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

- Crop a window W over the most recent data
- Estimate $\hat{\phi}_1 = \{(S_k, p_k^1)\}_{k=1, \dots, K}$ from W
- 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

Example of distances d between distributions are:

- Kullback-Leibler divergence
- Total variation distance, Pearson chi-square test
- Kolmogorov-Smirnov, Cramer-Von-Mises distance
- Kernel methods



DISTANCE – BASED MONITORING USING HISTOGRAMS

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

$$p_k^W = \frac{\#\{x_i \in S_k \cap W\}}{v}$$

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

$$d_{PS}(h^0, h^W) = v \sum_k \frac{|p_k^0 - p_k^W|}{p_k^0} \quad (\text{Pearson})$$

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



DISTANCE – BASED MONITORING SCHEME

Thresholding the distance is the typical stopping rule. Thresholds:

- are defined from the empirical distribution of $d(\hat{\phi}_0, \hat{\phi}_1)$, which is computed through a Boosting procedure.
- are analytically provided, as in case the of Pearson statistic

Similar approaches rules can be used to compare features extracted from $\hat{\phi}_0$ and $\hat{\phi}_1$ 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" TNLS 2016

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



CHANGE DETECTION APPROACHES

Parametric Settings:

- The Change-Point Formulation

Non-parametric Settings:

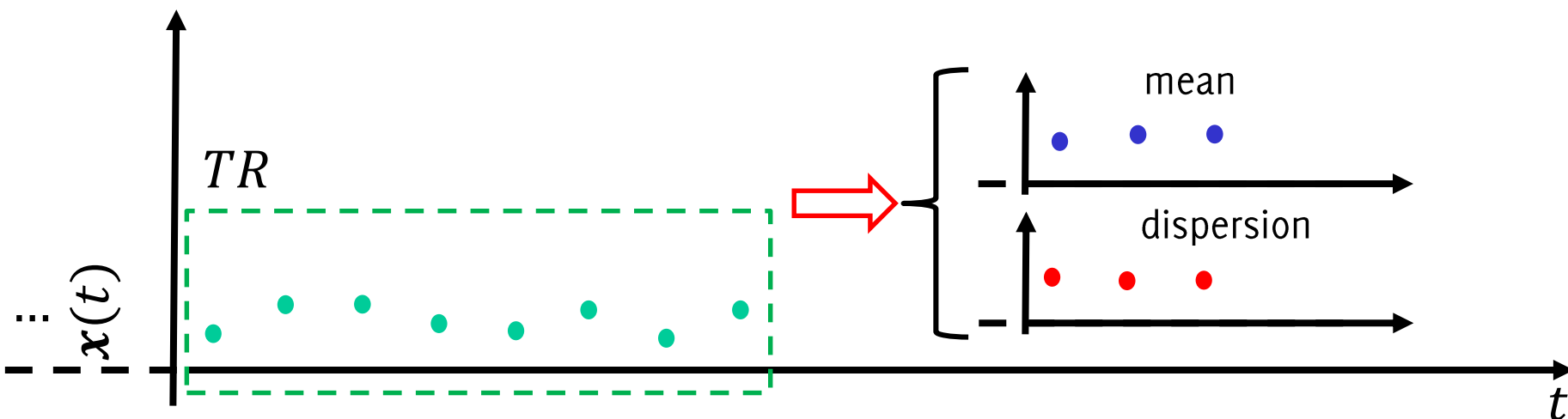
- The Change-Point Formulation
- Change-Detection by Histograms
- Change-Detection by Monitoring Features
- Hierarchical Change-Detection Tests



CHANGE DETECTION BY MONITORING FEATURES

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

Extract indicators (features), which are **expected to change** when $\phi_0 \rightarrow \phi_1$ and which **distribution is known** under ϕ_0





NONPARAMETRIC SETTINGS: SEQUENTIAL MONITORING

Examples of **decision rules** for features

- **CPM**, which can control the ARL_0
- **NP-CUSUM**, to detect changes in the data expectation
- **ICI rule**, to detect changes in the data expectation

Unfortunately **most** nonparametric statistics and the decision rules **do not apply to 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.

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

Tartakovsky, A. G., Veeravalli, V. V. "*Change-point detection in multichannel and distributed systems*". *Applied Sequential Methodologies: Real-World Examples with Data Analysis*, 173, 339-370, 2004

Alippi C., Boracchi G. and Roveri M. "Ensembles of Change-Point Methods to Estimate the Change Point in Residual Sequences" *Soft Computing*, Springer, Volume 17, Issue 11 (2013)



OUTLINE

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.



HIERARCHICAL CHANGE-DETECTION TESTS

In nonparametric sequential monitoring it is convenient to

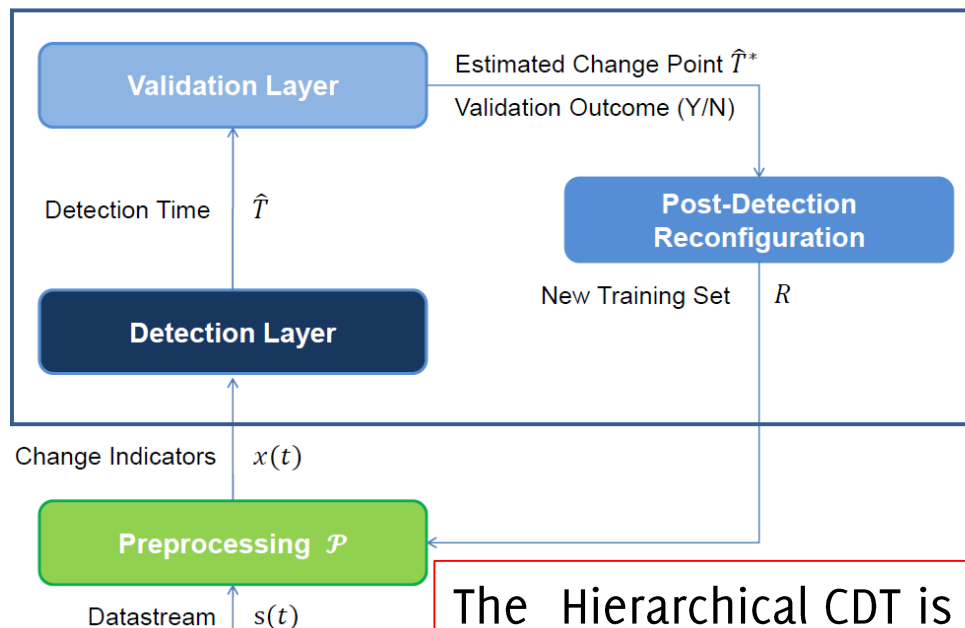
- **online sequential CDTs** for detection purposes
- **offline hypothesis tests** for validation purposes.

This results in two-layered (hierarchical) CDTs

Offline HT is activated to validate any detection

Online CDT detects process changes in the input datastream

Hierarchical Change-Detection Test

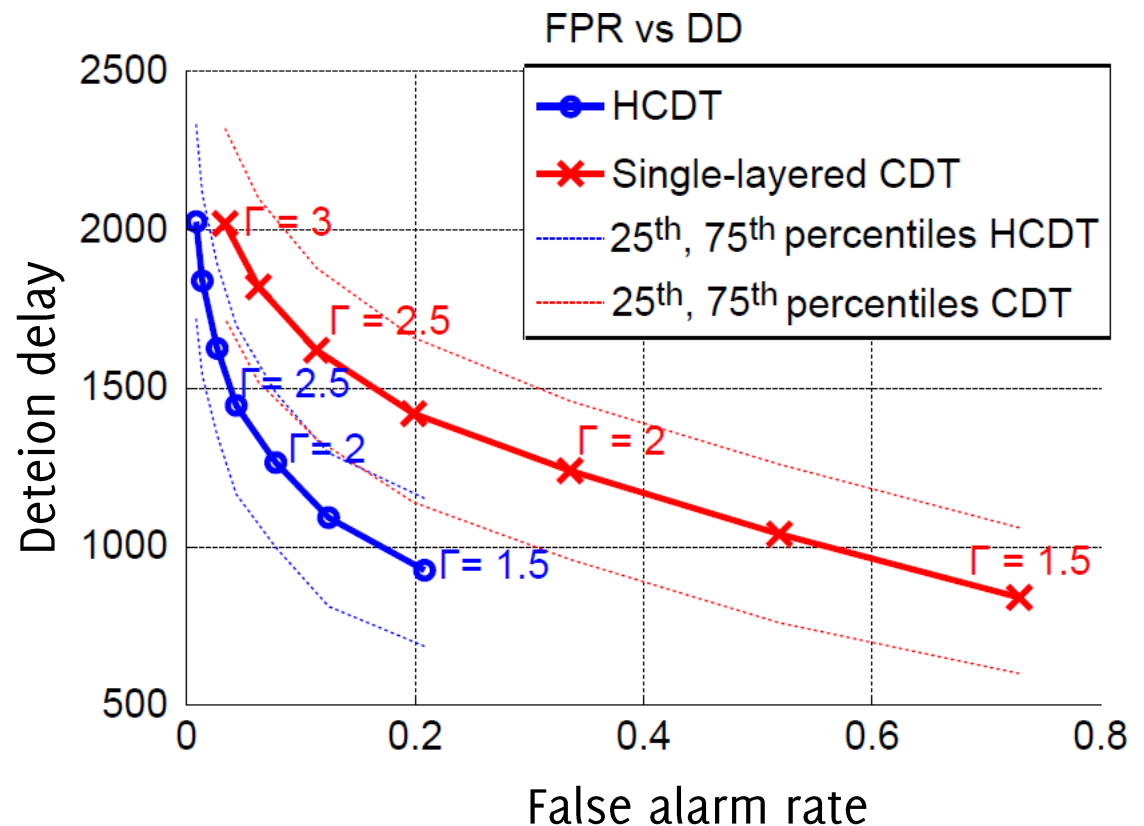


The Hierarchical CDT is automatically reconfigured



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.





Monitoring High-Dimensional Data

Change/anomaly in streams of high-dimensional
random vectors



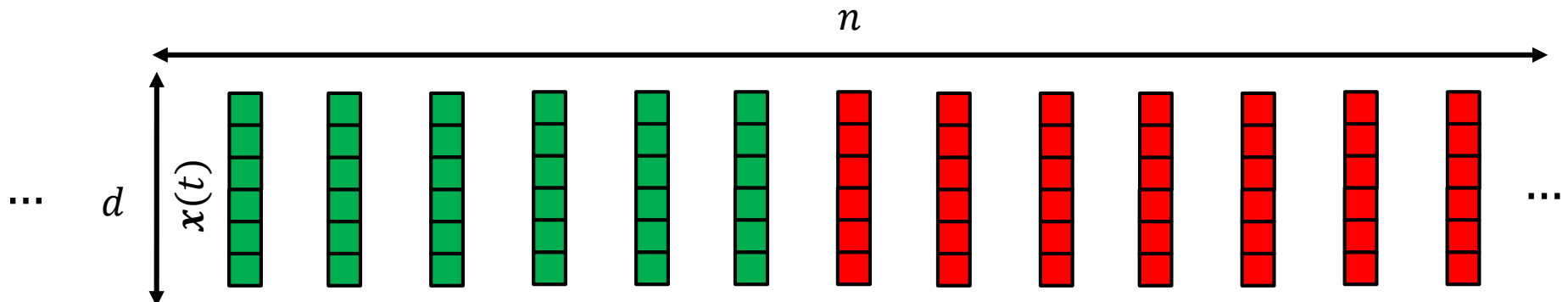
... WHEN DATA “ARE MANY”

When n (or the data throughput) grows:

- **Memory issues:** not feasible to store all the data in memory
- **Computational issues:** algorithms should be $\mathcal{O}(1)$, and single-pass
- **Having a lot of training samples is good!**

Thus, there is need for

- approximated statistics
- Incremental formulas, dataset pruning/summarization

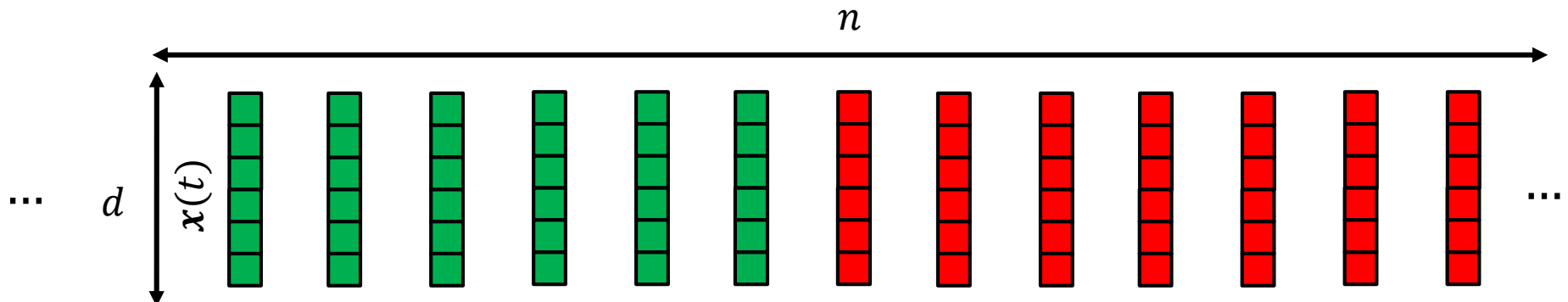




... WHEN DATA “ARE LARGE”

When d grows:

- **Memory issues:** not feasible to store many data in memory
- Difficult to find a model $\hat{\phi}_0$, many training samples needed
- Number of irrelevant component might increase
- Distance-based methods are difficult to tune
- Combinatorial growth of the number of subspaces
- Data-visualization issues
- **Detectability loss**





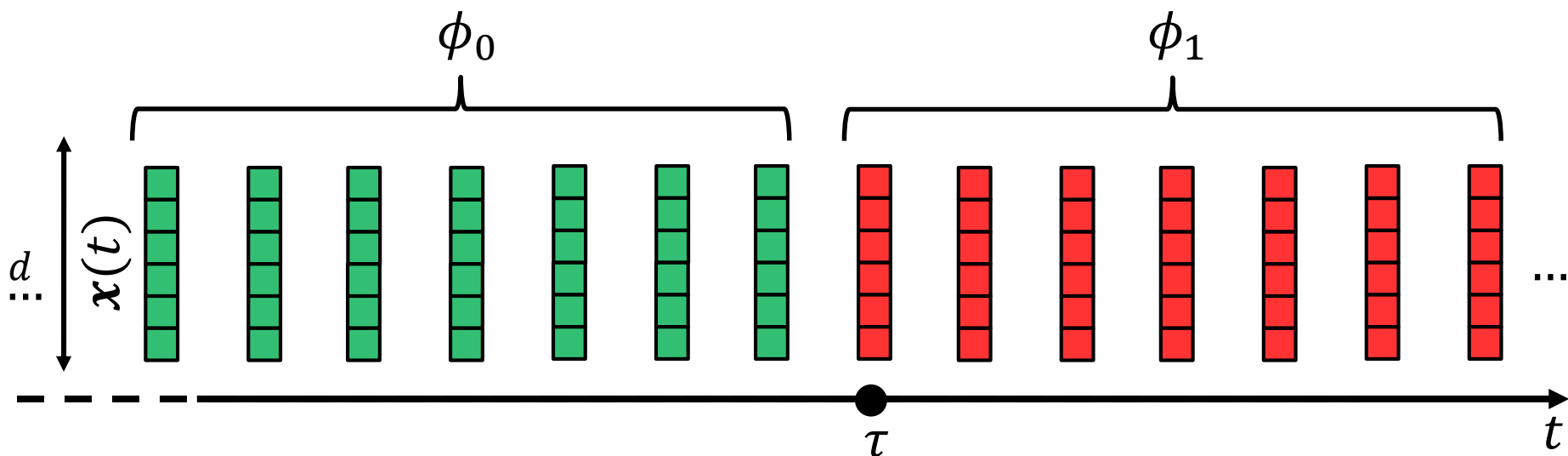
Detectability loss in high-dimensional data

How data dimension affects monitoring the Log-likelihood



OUR GOAL

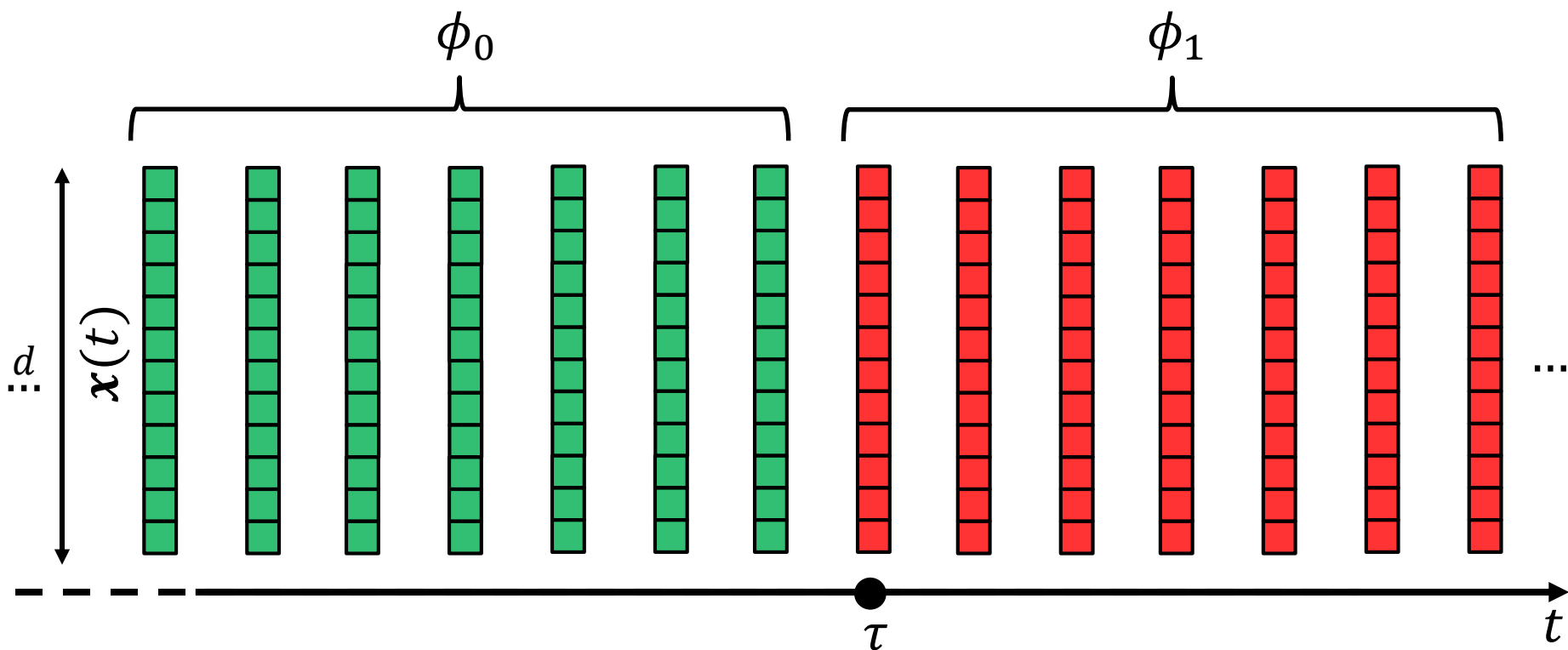
*Study how the **data dimension d** influences the **change detectability**, i.e., how difficult is to solve change/anomaly detection problems*





OUR GOAL

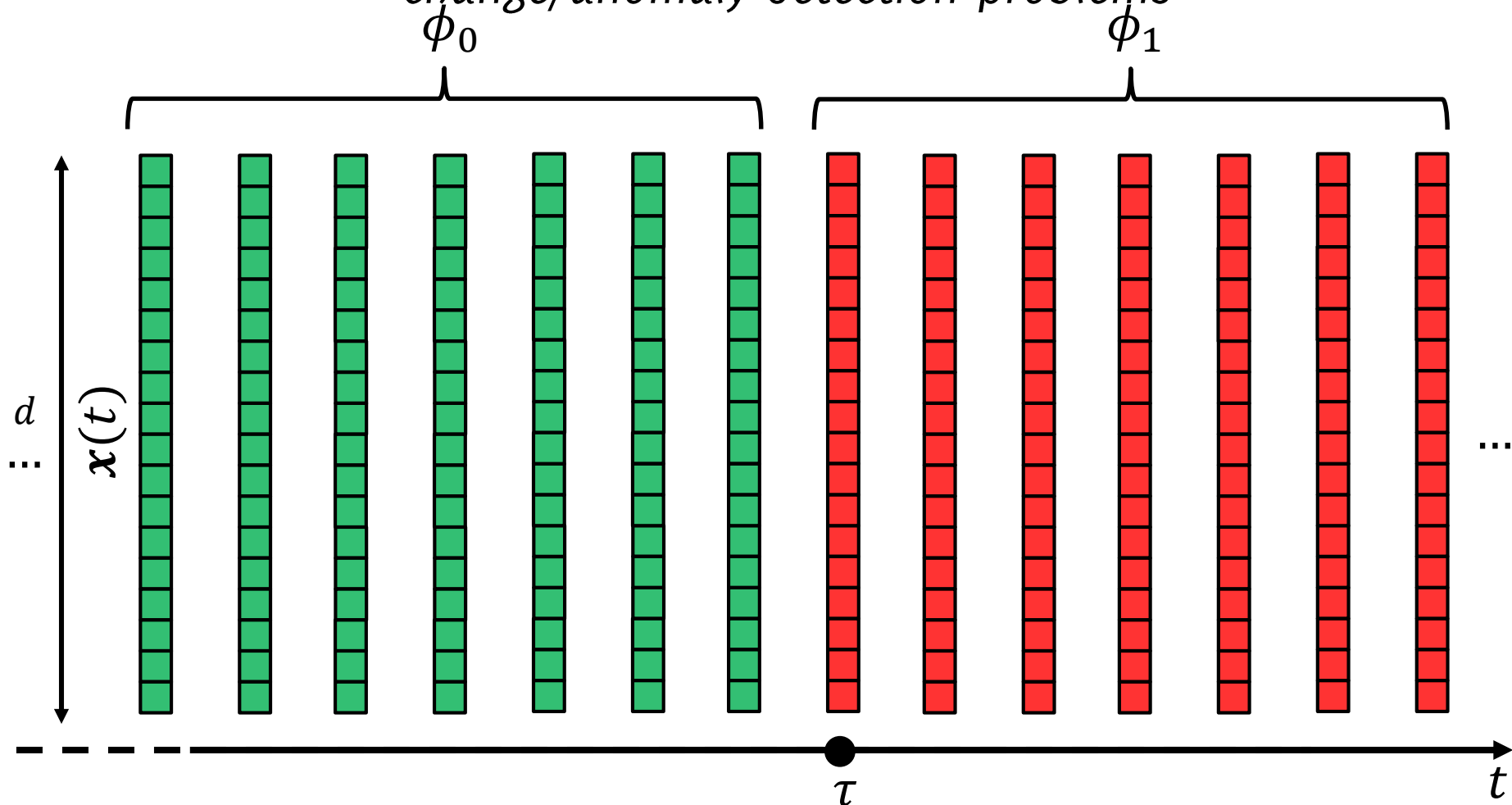
*Study how the **data dimension d** influences the **change detectability**, i.e., how difficult is to solve change/anomaly detection problems*





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** 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



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**



Preliminaries:

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

The *detectability loss*

- Analytical results
- Empirical analysis



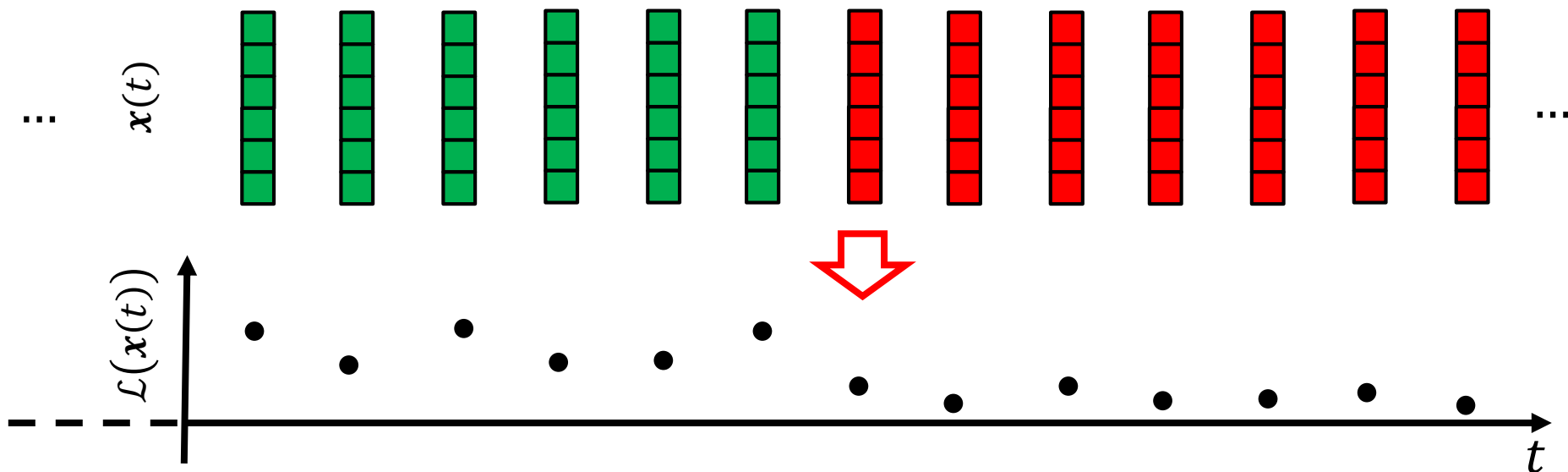
HOW? MONITORING THE LOG-LIKELIHOOD

A typical approach to monitor the log-likelihood

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

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

3. Monitor $\{\mathcal{L}(\mathbf{x}(t)), t = 1, \dots\}$





Preliminaries:

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

The *detectability loss*

- Analytical results
- Empirical analysis



The *Signal to Noise Ratio of the change*

$$\text{SNR}(\phi_0 \rightarrow \phi_1) = \frac{\left(\mathbb{E}_{x \sim \phi_0} [\mathcal{L}(\mathbf{x})] - \mathbb{E}_{x \sim \phi_1} [\mathcal{L}(\mathbf{x})] \right)^2}{\text{var}_{x \sim \phi_0} [\mathcal{L}(\mathbf{x})] + \text{var}_{x \sim \phi_1} [\mathcal{L}(\mathbf{x})]}$$

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



Preliminaries:

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

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*

$$\begin{aligned} \text{sKL}(\phi_0, \phi_1) &= \text{KL}(\phi_0, \phi_1) + \text{KL}(\phi_1, \phi_0) = \\ &= \int \log \left(\frac{\phi_0(\mathbf{x})}{\phi_1(\mathbf{x})} \right) \phi_0(\mathbf{x}) d\mathbf{x} + \int \log \left(\frac{\phi_1(\mathbf{x})}{\phi_0(\mathbf{x})} \right) \phi_1(\mathbf{x}) d\mathbf{x} \end{aligned}$$

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



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** $\text{SNR}(\phi_0 \rightarrow \phi_1)$ **varies** in **change-detection problems** that have

- **different data dimensions d**
- **constant change magnitude $s\text{KL}(\phi_0, \phi_1)$**



Preliminaries:

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

The *detectability loss*

- Analytical results
- Empirical analysis



THE DETECTABILITY LOSS

Theorem

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

$$\text{SNR}(\phi_0 \rightarrow \phi_1) < \frac{C}{d}$$

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



THE DETECTABILITY LOSS: REMARKS

Theorem

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

$$\text{SNR}(\phi_0 \rightarrow \phi_1) < \frac{C}{d}$$

where C is a constant that depends only on $\text{sKL}(\phi_0, \phi_1)$

Remarks:

- Changes of a given magnitude, $\text{sKL}(\phi_0, \phi_1)$, become more difficult to detect when d increases
- DL does not depend on how ϕ_0 changes
- 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

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

$$\text{SNR}(\phi_0 \rightarrow \phi_1) < \frac{C}{d}$$

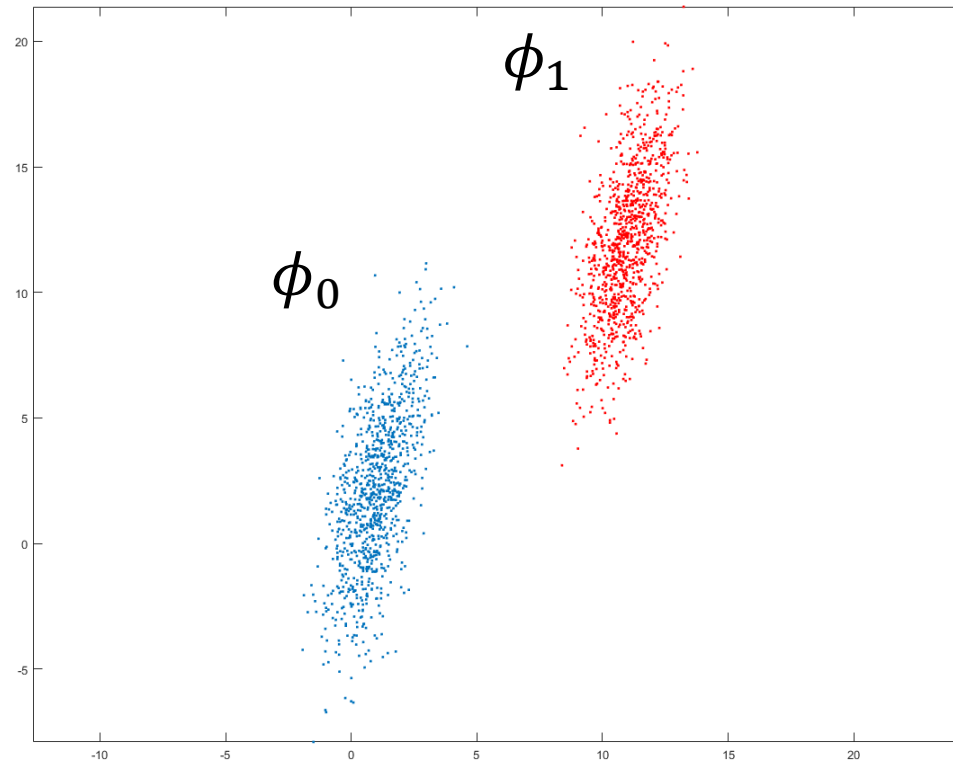
where C is a constant that depends only on $\text{sKL}(\phi_0, \phi_1)$



THE DETECTABILITY LOSS: THE CHANGE MODEL

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

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

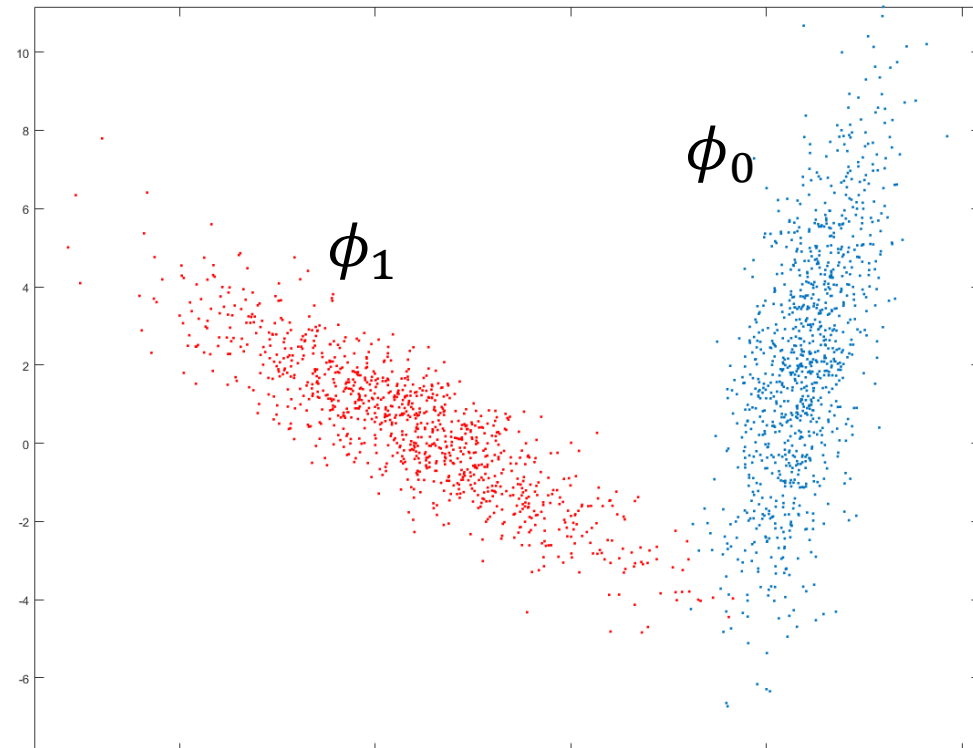




THE DETECTABILITY LOSS: THE CHANGE MODEL

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

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



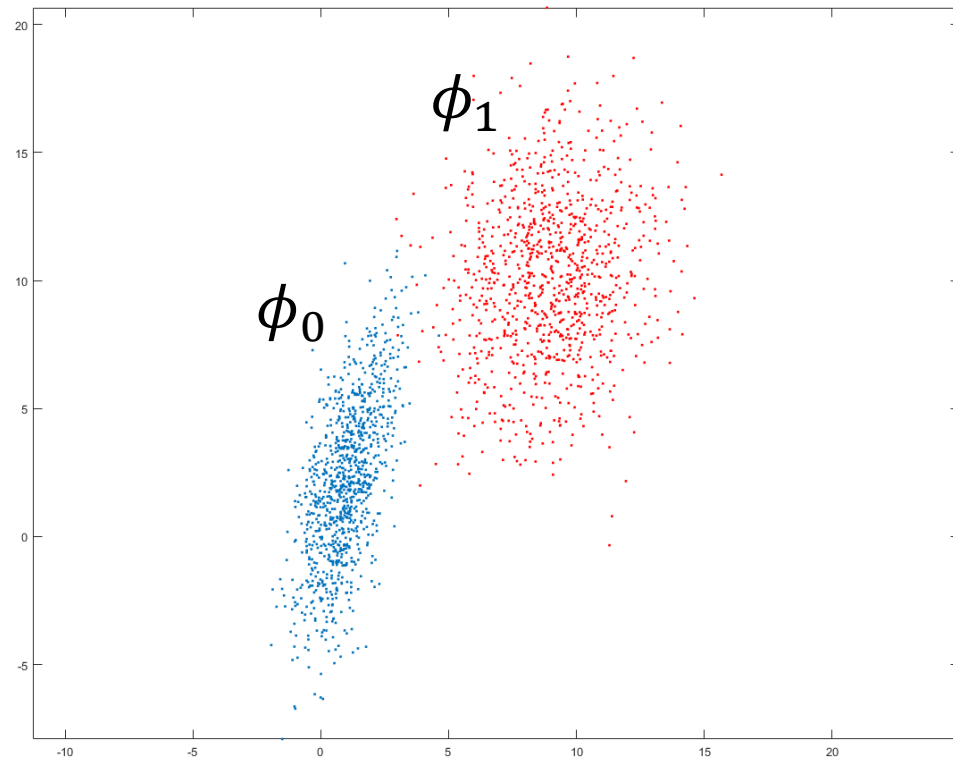


THE DETECTABILITY LOSS: THE CHANGE MODEL

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

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

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





THE DETECTABILITY LOSS: THE GAUSSIAN ASSUMPTION

Theorem

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

$$\text{SNR}(\phi_0 \rightarrow \phi_1) < \frac{C}{d}$$

where C is a constant that depends only on $\text{sKL}(\phi_0, \phi_1)$



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 variables having independent components
- The result have been empirically confirmed in case of approximations of $\mathcal{L}(\cdot)$ typically used for Gaussian mixtures



Preliminaries:

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

The *detectability loss*

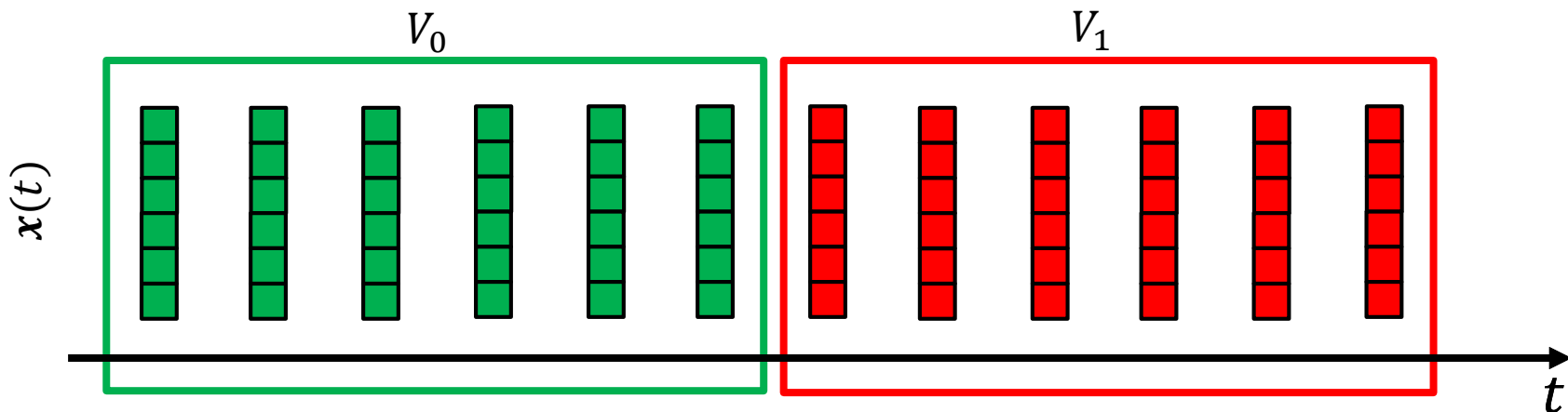
- Analytical results
- Empirical analysis



THE DETECTABILITY LOSS: EMPIRICAL ANALYSIS

The data

- Synthetically generate streams with different dimensions d
- Estimate $\hat{\phi}_0$ by GM from a **stationary training set**
- In each stream we introduce $\phi_0 \rightarrow \phi_1$ such that
$$\phi_1(\mathbf{x}) = \phi_0(Q\mathbf{x} + \mathbf{v}) \text{ and } \text{sKL}(\phi_0, \phi_1) = 1$$
- **Test data: two windows** V_0 and V_1 (500 samples each) selected before and after the change.





THE DETECTABILITY LOSS: EMPIRICAL ANALYSIS

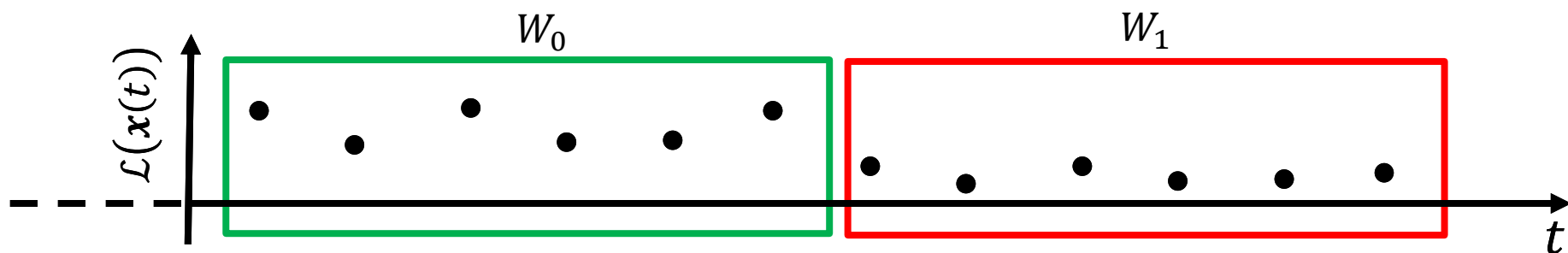
We measure the change-detectability as:

- Compute $\mathcal{L}(\hat{\phi}_0(\mathbf{x}))$ 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

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

where h controls the amount of false positives

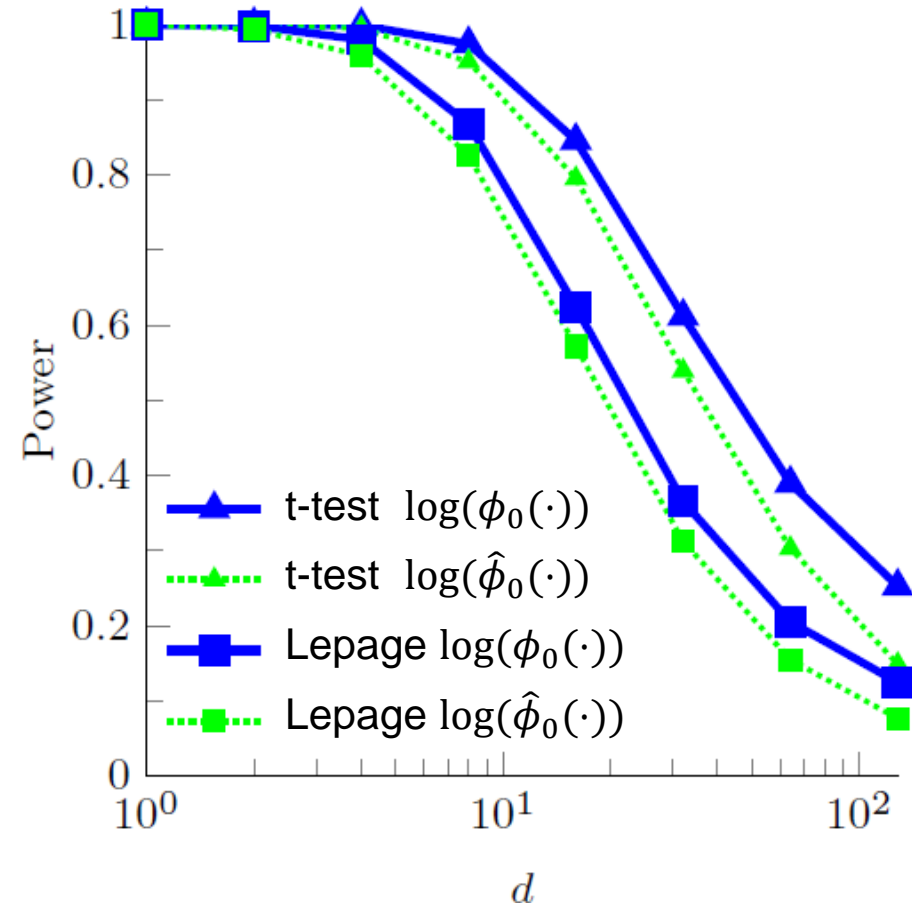
- Use the **power** of this test to assess change detectability





THE HYPOTHESIS TESTS POWER ON GAUSSIAN STREAMS

Gaussians



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 $\log(\phi_0(\cdot))$

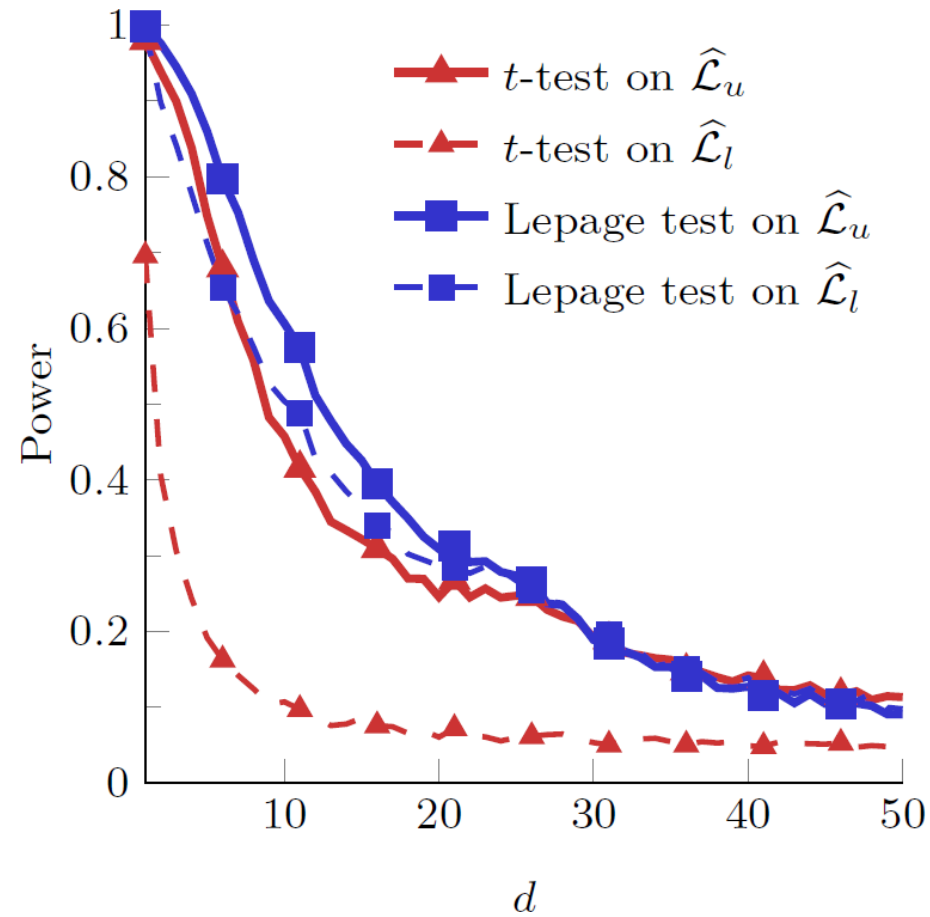
Results

- 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

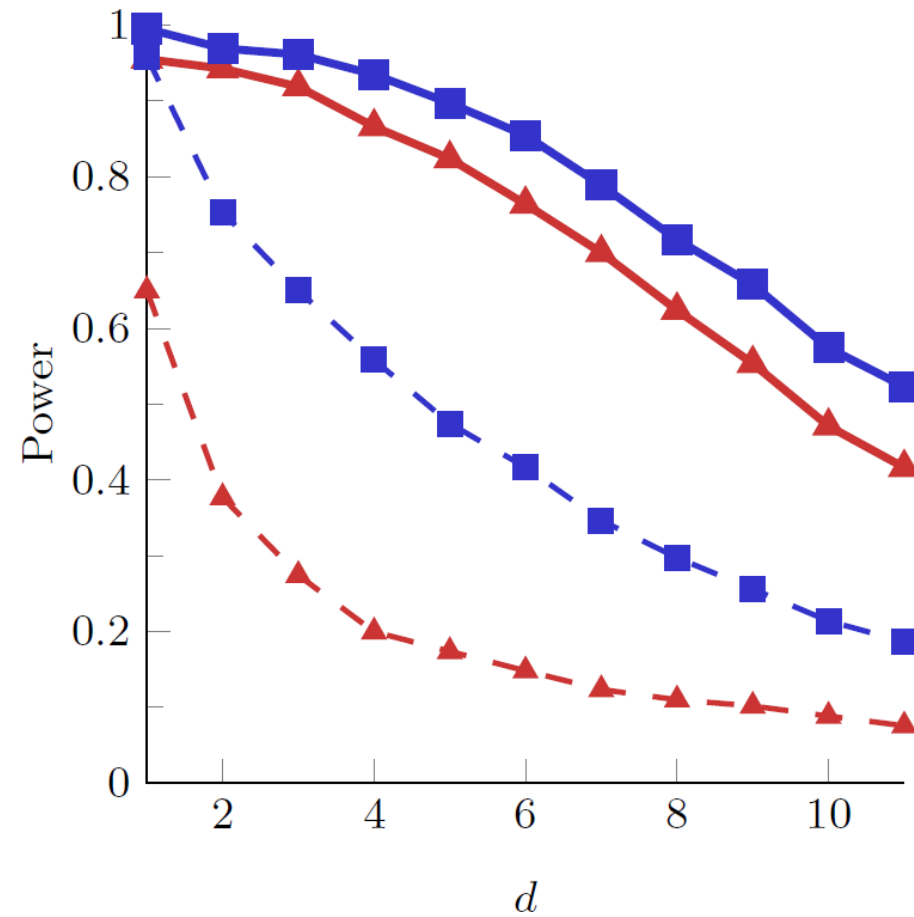


THE HYPOTHESIS TESTS POWER ON UCI DATASETS

Particle

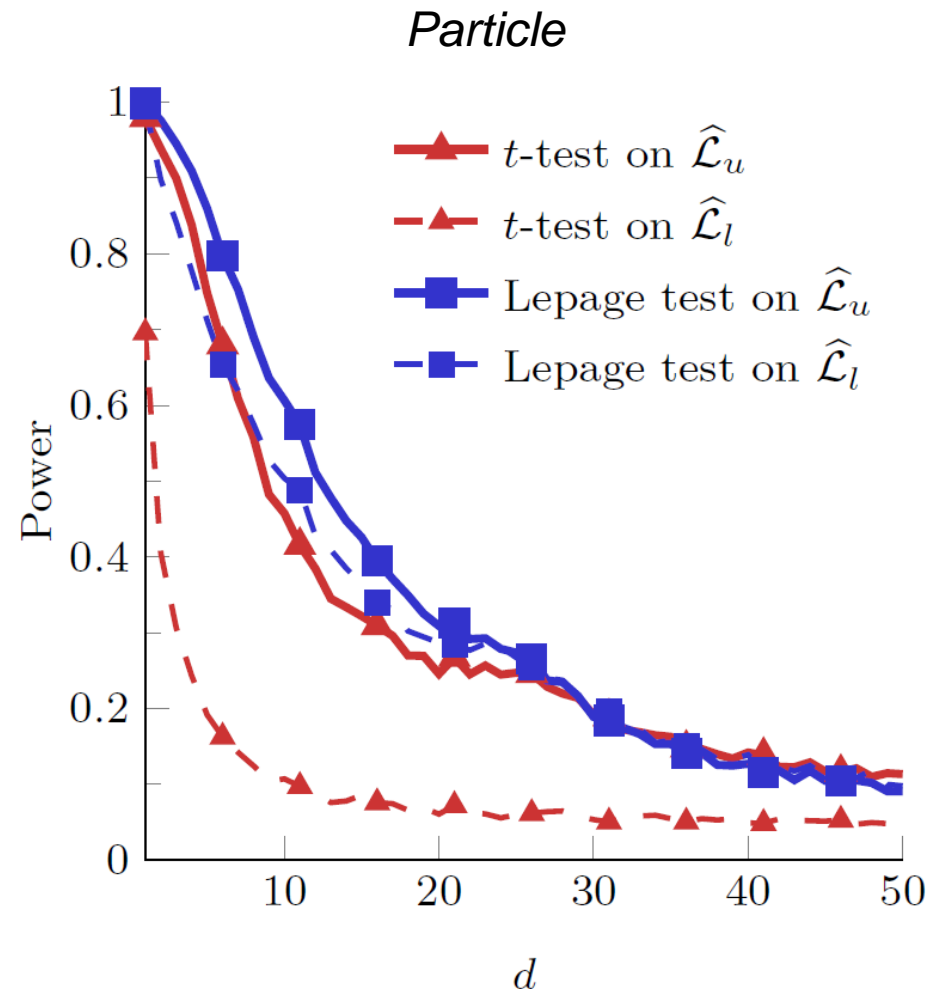


Wine





THE HYPOTHESIS TESTS POWER ON PARTICLE DATASET



Remarks:

- ϕ_1 is defined through CCM a framework to control the change magnitude and yield $s\text{KL}(\phi_0, \phi_1) \approx 1$
- $\hat{\phi}_0$ 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



A NOTE ON EXPERIMENTAL PRACTICES

To correctly assess the **change-detection performance**, in particular when changing the input size, it is **necessary to control the magnitude of injected changes**

Controlling the change magnitude provides results that are better interpretable and reproducible.



CCM: CONTROLLING THE CHANGE MAGNITUDE

Watch-out: Most of experimental practices to manipulate datastreams for change-detection purposes, lead to changes that steadily increase with d

