Applications

Giacomo Boracchi, Francesco Trovò

June 10th, 2020

Politecnico di Milano, DEIB

francesco1.trovo@polimi.it

Lecture Overview

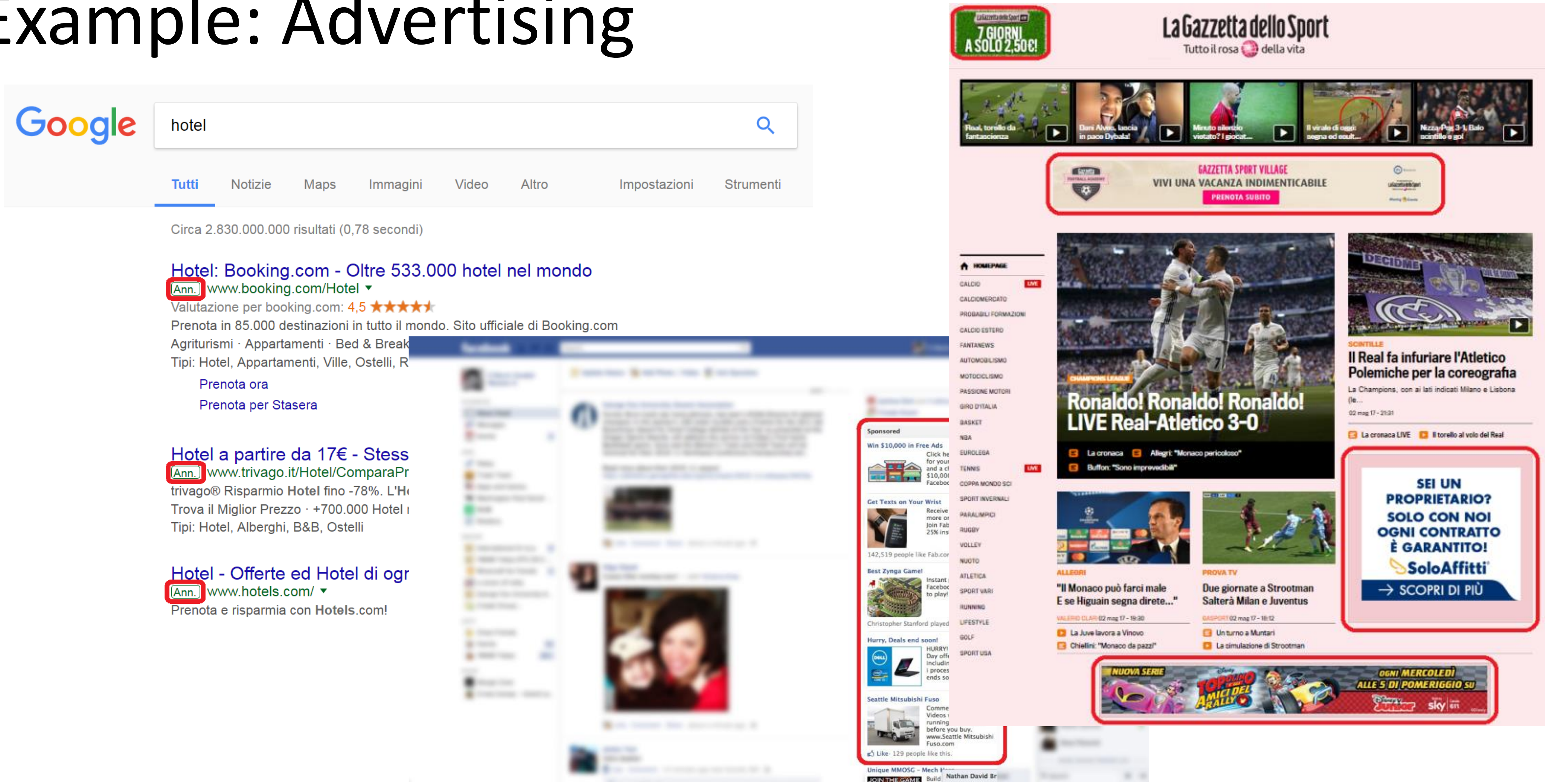
Non-Stationary MAB

Applications

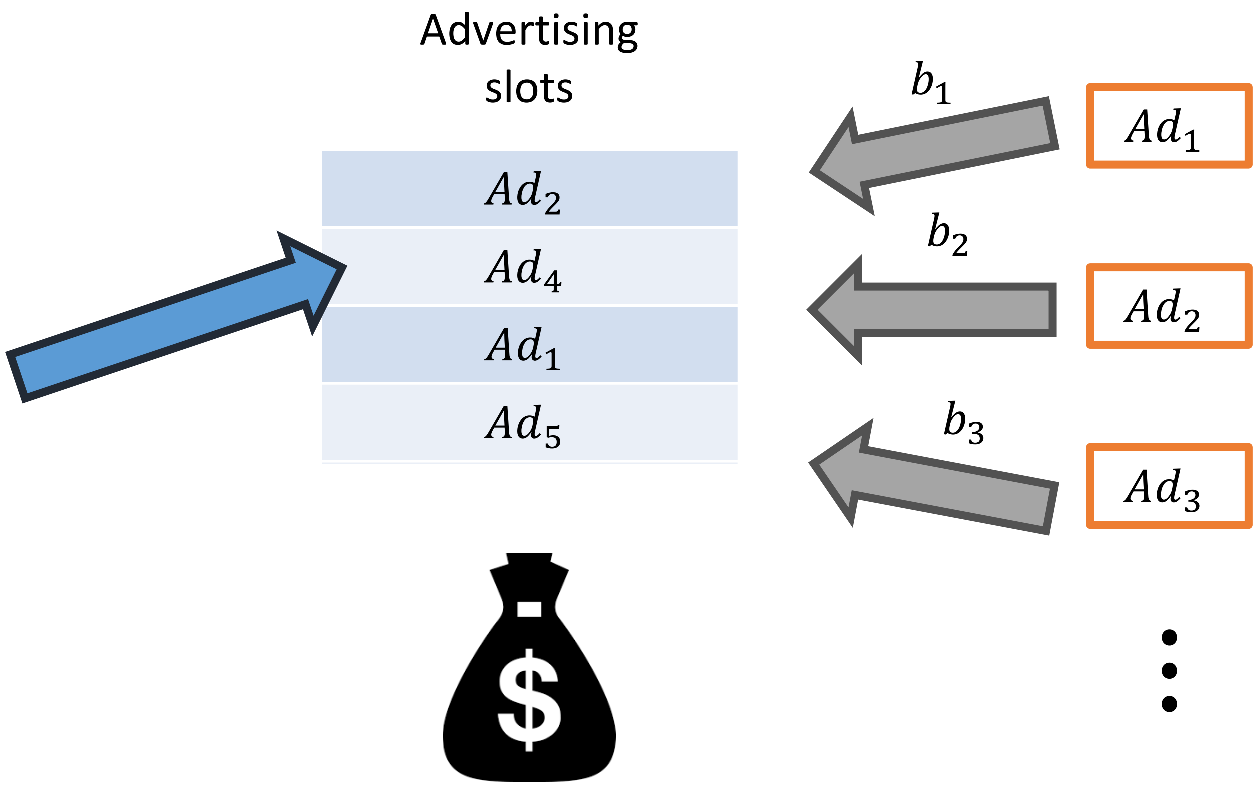
- ***Side Channel Attacks***
- ***Sparse Representation for Online Monitoring***
- ***Credit Card Fraud Detection***

Advertising Application

Example: Advertising



Advertising Process



Cost c_4
Value $p_4 \cdot \rho_4$

Real-World Application

We have an **advertisement** (ad) and we want to optimize its performance

To display the add we need to provide the platform with bid b_t, i.e., the maximum amount of money we would like to spend for a click on the add at round t

We will get on average as reward for that ad:

$$(p \cdot \rho - c_t) \cdot ctr_t$$

- p Price of the item
- ρ Conversion rate (probability that a user buys our product once he/she is on our website)
- c_t Cost of the click
- ctr_t Click trough rate

Problem Formulation

The click trough rate ctr_t is determined by the bid b_t we select, and the relationship between the two is unknown

First model:

- The user are malicious: we model the problem as an adversarial MAB
- The user are behaving stochastically: we model the problem as a stochastic MAB

Problem solved!

Phenomena to Include in the Model

- Depending on the market the behavior of the user might change suddenly, e.g., due to the entrance of a new player or new product in the market
    - The change can happen at any time point
    - The change is instantaneous
    - The change does not occur too often (otherwise we are in the adversarial setting)

- Sales on some product presents seasonal behaviors, their appeal to the users is different from one month of the year to the other, e.g., ice cream
    - Continuous change of the expected rewards
    - The information in the near past still provides some value to the learner

Non-Stationary MAB Problem

Set of arms $a_1, \dots, a_K$

Set of rewards whose expected value $\mu_{i,t}$ might change over time

Abrupt:

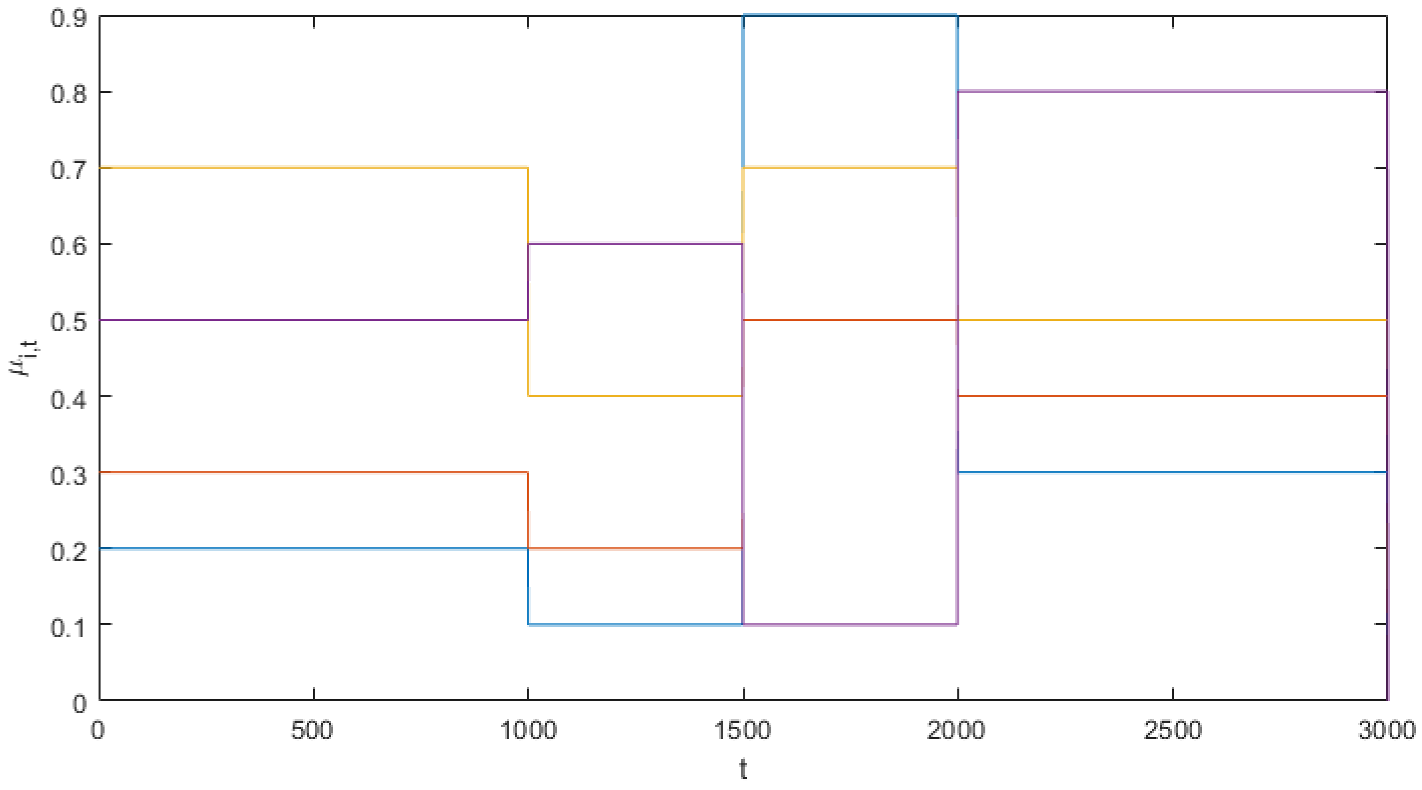
- There exists a set of phases $\phi = \{b_0, \dots, b_\Upsilon\}$ in the learning process identified by some breakpoints b_j
- The rewards are constant during each phase $\mu_{i,t} = \mu_{i,t+1}$ if $t, t+1 \in [b_j, b_{j+1})$

Smoothly changing:

- The expected reward of each arm cannot vary more than a limited amount between consecutive rounds $|\mu_{i,t} - \mu_{i,h}| \leq \sigma|t-h|$ for each $t, h \in \{1, \dots, T\}$

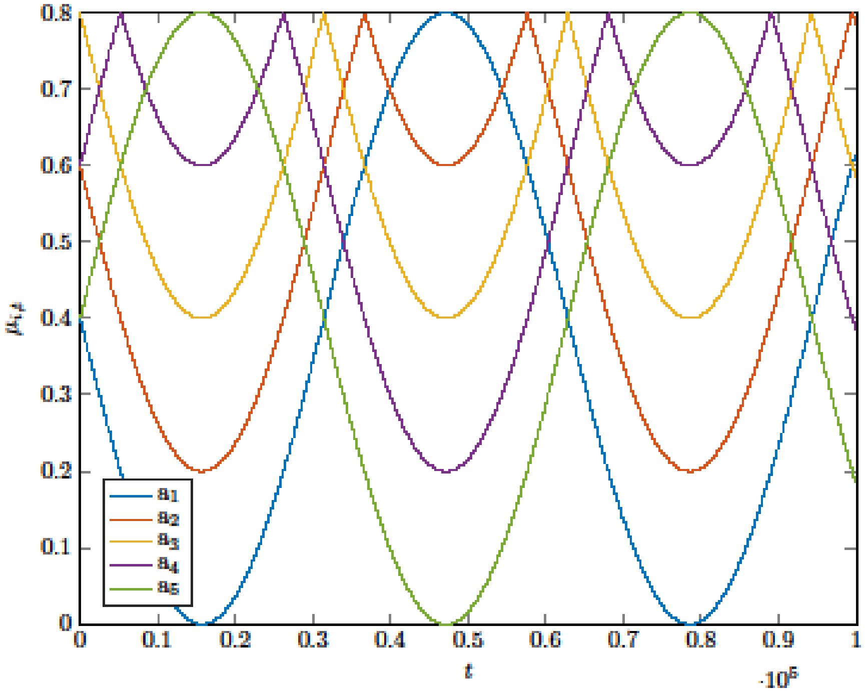
Example of Reward Evolution over Time

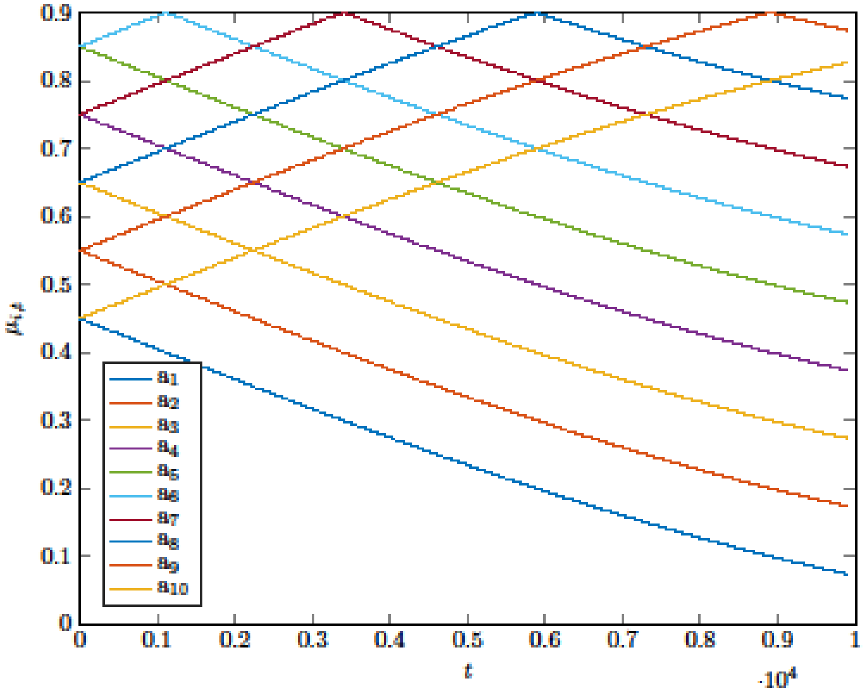
Abruptly Changing Arm Reward



Example of Reward Evolution over Time

Smoothly Changing Arm Reward





Dynamic Regret

Definition: Stochastic Dynamic Regret

Given an algorithm A, selecting an arm I_t at round t the Regret of A over a time horizon of T rounds is:

$$R_n(A) = \sum_{t=1}^{n} [\mu_t^* - E[\mu_{I_t,t}]]$$

where the expectation is w.r.t. the forecaster stochasticity

The best arm is defined as the one with $\mu_t^* = \max_i E[\mu_{i,t}]$

We are tracking the best arm, which may change over rounds

Deal with Abrupt Changes

Two different possible approaches:

- Passive approach: use more the information of the near past (e.g., over a predefined time window), not explicitly trying to analyse the time point in which the change occurred

- Active approach: use a CDT to decide when the reward expected value has changed

Passive Approach – SW-UCB

at each round t

play the arm I_t having the largest $u_{i,t} = \bar{g}_{i,t,\tau} + \sqrt{\frac{\xi \log \min\{t,\tau\}}{N_{i,t,\tau}}}$

get reward $g_{I_t,t}$

update the sample mean $\bar{g}_{I_t,t}$ and the bounds for all the arms

Where:

- $\bar{g}_{i,t,\tau}$ expected reward over the last τ rounds
- $N_{i,t,\tau}$ number of pulls of the arm a_i over the last τ rounds

Regret of the SW-UCB

Theorem

The SW-UCB algorithm applied to an abruptly changing stochastic MAB problem with $\xi > 0.5$ and $\tau = 2\sqrt{\frac{n \log n}{\Upsilon}}$ with K arms suffers a regret of:

$$R_n \leq O\left(K\sqrt{\Upsilon\, n \log n}\right)$$

It requires that:

- We know the number of breakpoints Υ in advance
- We know the time horizon n in advance
- It does not scale well computationally if n is large

Doubling Trick

If we do know the time horizon in advance:

- Starting from $k = 0$ Run the algorithm on a phase $[2^k, 2^{k+1}]$ as if the process would end at time 2^{k+1} (with a number of rounds equal to 2^k)
- At the beginning of each phase restart the algorithm from scratch

For a time horizon n we have a number of phases $k \; = \log_2 n$

The regret becomes:

$$R_n = \sum_{k=0}^{\log_2 n} R_{2^k} \leq \sum_{k=0}^{\log_2 n} \sqrt{\Upsilon_k 2^k \log 2^k} \leq \log_2 n \sqrt{\Upsilon n \log n}$$

Active Approach

Naïve implementation:

- We run a standard MAB algorithm for stochastic MAB (UCB1, TS)
- We run in parallel a CDT to identify if a change occurs

Issue: we receive samples from an arm according to its performance

- The optimal arm is pulled a number of times which is $O(t)$ at each round t
- Each suboptimal arm is pulled a number of times $O\left(\frac{\log t}{\Delta_{i,t}}\right)$

Therefore, the expected detection delay $E[D]$ becomes of the order of $E\left[D\,\frac{\Delta_{i,t}}{logt}\right]$

CDT-UCB

at each round t

if at least one of the CDT is not ready yet

play the arm required by the CDT

else

run a CDT on the reward of each arm

if the CDT is positive

reset the expected value, the number of pulls, and the CDT for that arm

play the arm I_t having the largest $u_{i,t} = \bar{g}_{i,t} + \sqrt{\frac{\log n_t}{N_{i,t}}}$ with probability $1 - \alpha$

an arm at random with probability α

get reward $g_{I_t,t}$

update the sample mean $\bar{g}_{I_t,t}$ and the bounds for all the arm

Liu, Fang, Joohyun Lee, and Ness Shroff. "A change-detection based framework for piecewise-stationary multi-armed bandit problem." ***AAAI*** **2018**

CDT-UCB Regret

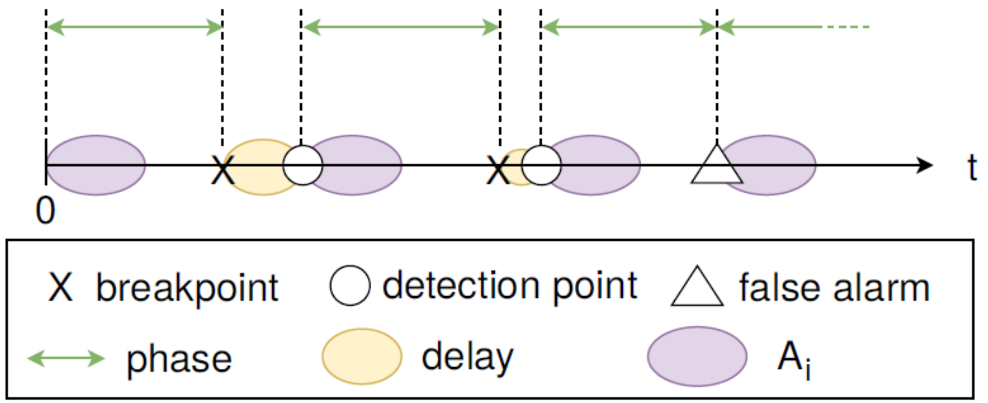
Theorem

The CDT-UCB algorithm applied to an abruptly changing stochastic MAB problem with suffers a regret of:

$$R_n \leq (\Upsilon + E[F_n]) \cdot \left(\frac{4 \log n}{\Delta_i^2} + \frac{\pi^2}{3} \right) + \frac{\pi^2}{3} + \Upsilon E[D] + \frac{\alpha n}{K}$$

where $\mathrm{E}[F_n]$ is the expected numbers of false positive up to time n of the CDT and $\mathrm{E}[D]$ is the expected detection delay of the CDT

Idea on the composition of the regret over an abruptly changing environment



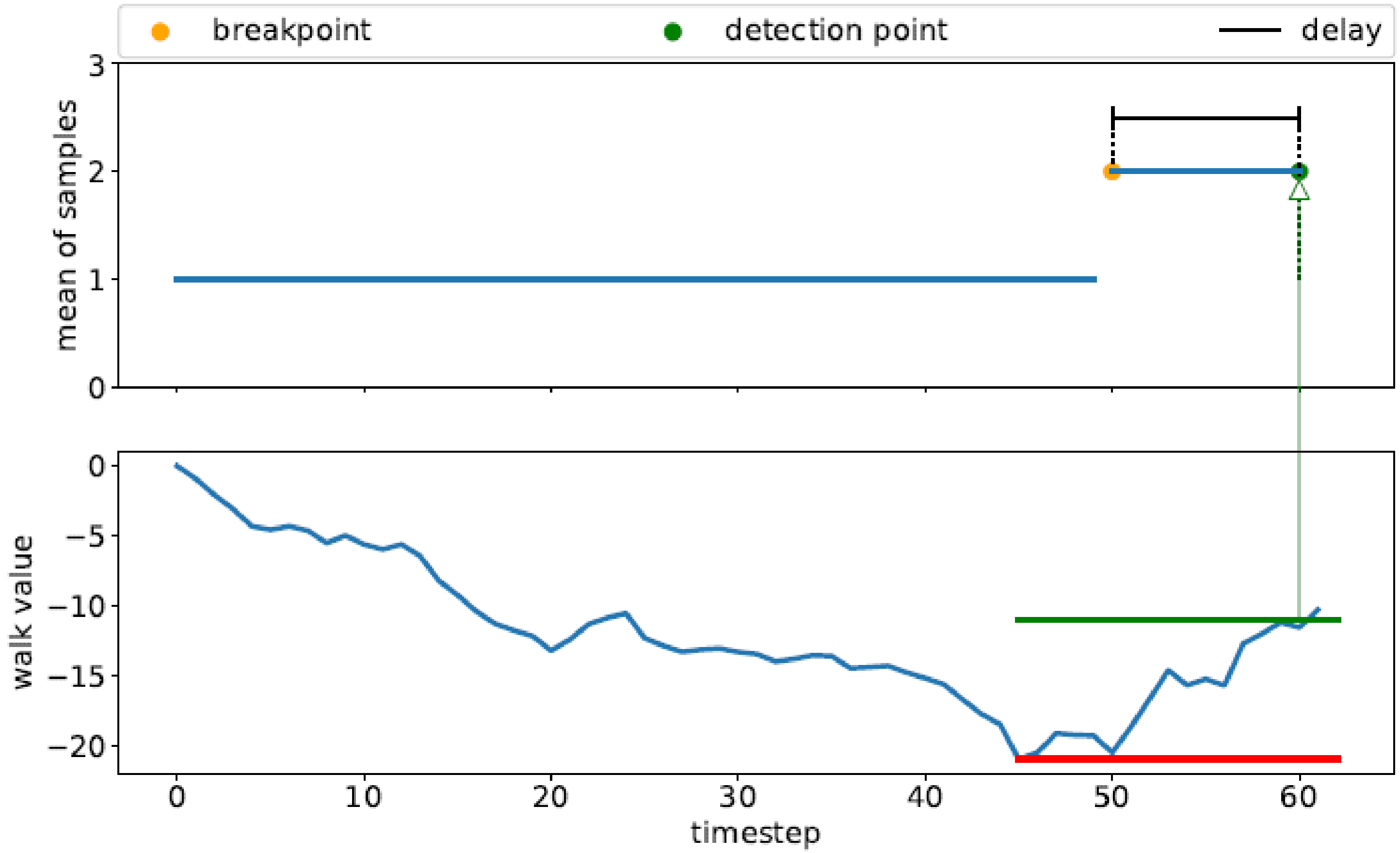
Boracchi, Trovò

A Possible CDT for the Task

Cusum test:

- We first identify a value for the expected reward for each arm $\bar{\mu_i}$ (which requires m samples from each arm)
- We set a threshold h defining when the test will trigger and a parameter ϵ s.t. the changes in the expected rewards are larger than 3ϵ
- We compute:
  - $g^+_{i,t} \leftarrow \max\{0, g^+_{i,t-1} + r_{i,t} - \bar{\mu}_i - \epsilon\}$
  - $g^-_{i,t} \leftarrow \max\{0, g^-_{i,t-1} + \bar{\mu}_i - r_{i,t} - \epsilon\}$
- If either one of the two indexes are exceeding the threshold h, we say a change occurred

Example of the Execution of the CUSUM



Boracchi, Trovò

CUSUM-UCB Regret

Theorem

The CUSUM-UCB algorithm applied to an abruptly changing stochastic MAB problem using $\xi = 1$, $\alpha = CK\sqrt{\frac{\Upsilon}{n}\log\frac{n}{\Upsilon}}$ suffers a regret of:

$$R_n = O\left(\frac{\Upsilon \log \mathrm{n}}{\Delta_i^2} + \sqrt{n\,\Upsilon \log\frac{n}{\Upsilon}}\right)$$

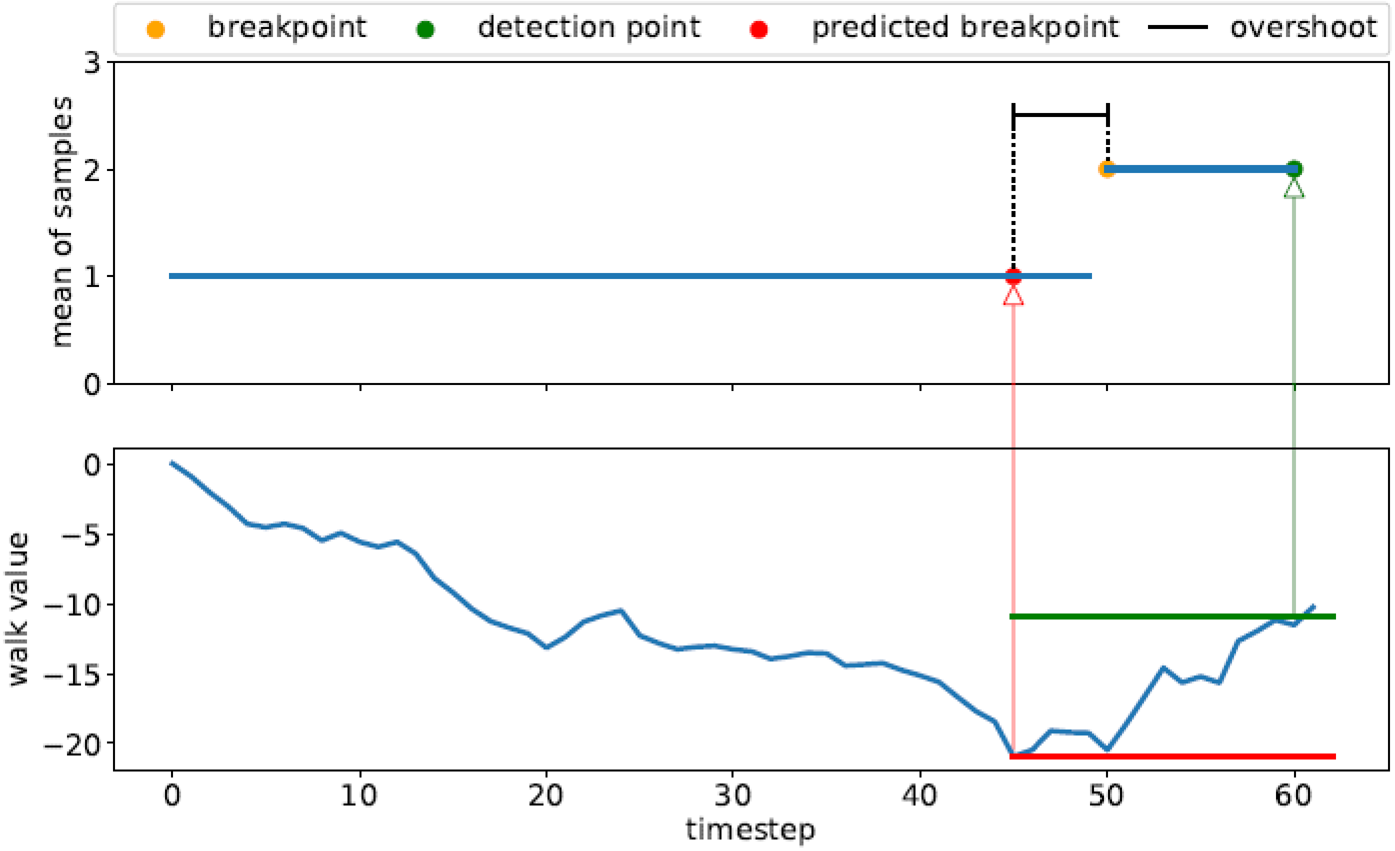
using a specific choice of $h \propto \log\frac{n}{\Upsilon}$

We have an improvement in terms of the number of breakpoints Υ present over the time horizon n

Liu, Fang, Joohyun Lee, and Ness Shroff. "A change-detection based framework for piecewise-stationary multi-armed bandit problem." *AAAI* 2018

Last Improvement

Change Point Estimation



Boracchi, Trovò

Critical Analysis of the Active Approaches

Pros:

- Able to directly identify the CDT (valuable information)
- In general they are more powerful than passive approaches

Cons:

- Requires the knowledge of the minimum gap 3ϵ provided by the change
- Requires that the changes does not occur less than before mK rounds from the previous one

Smooth Changes SW-KL-UCB

Under the assumptions that:

- The number of times the arms are closer than Δ over the time horizon is bounded by $H(\Delta, n) \leq F\,\Delta\,\mathrm{T}$
- The Lipchitz constant for the variation of the mean reward is σ

We can use:

at each round t

play the arm I_t having the largest

$q_{i,t,\tau} = \sup\{q \geq \bar{g}_{I_t,t,\tau}, N_{i,t,\tau} KL\left(\bar{g}_{I_t,t,\tau}, q\right) \leq \log n_{t,\tau} + c \log\log n_{t,\tau}\}$

get reward $g_{I_t,t}$

update the sample mean $\bar{g}_{I_t,t,\tau}$ and the bounds for all the arms

Regret Results

Theorem

The SW-KL-UCB algorithm applied to an abruptly changing stochastic MAB problem with $\tau = \frac{1}{8}\sigma^{-\frac{3}{4}} \log\frac{1}{\sigma}$ suffers a regret of:

$$\limsup_{n \to +\infty} \frac{R_n}{n} \leq C\sigma^{\frac{1}{4}} \log\frac{1}{\sigma}$$

In conclusion we do not have a vanishing regret, but a per-round regret which is limited by the Lipchitz constant σ

Hint: it is possible to derive a better result assuming that $H(\Delta, n) \propto n^{\beta}$ but the authors did not analysed this option in their work

Both Changes – SW-TS

set a prior $\pi_{i,1} = Beta(1,1)$ for each arm i

at each round t

select a sample θ_i from each distribution $\pi_{i,t,\tau}$

play the arm I_t having the largest θ_i

get reward $g_{I_t,t}$

update the vectors $\alpha_{i,t,\tau}$ and $\beta_{i,t,\tau}$ the current reward and excluding the rewards which are older than τ rounds

The idea is still to keep only the most recent samples to take a decision

It still requires to store a number of samples which is dependent on the sliding window τ

Trovò, Francesco, et al. "Sliding-Window Thompson Sampling for Non-Stationary Settings." ***JAIR 2020*****.**

Results for SW-TS

Theorem

The SW-TS algorithm applied to an abruptly changing stochastic MAB problem with $\tau = \sqrt{n}$ suffers a regret of:

$$R_n \leq K\,\Upsilon + \sum_{i=1}^{K} \left(52 \frac{log\sqrt{n}}{\Delta_i} + \log\sqrt{n} + 5 + \frac{19}{\log\sqrt{n}} \right) \sqrt{n}$$

Theorem

The SW-TS algorithm applied to a smoothly changing stochastic MAB problem using a sliding window of $\tau = n^{1-\beta}$ suffers a regret of:

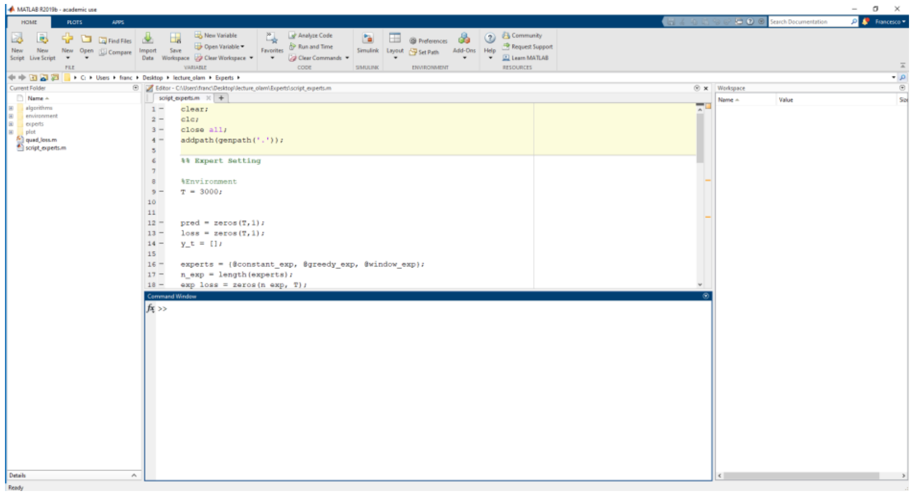
$$R_n = O\left(n^{\beta}\right)$$

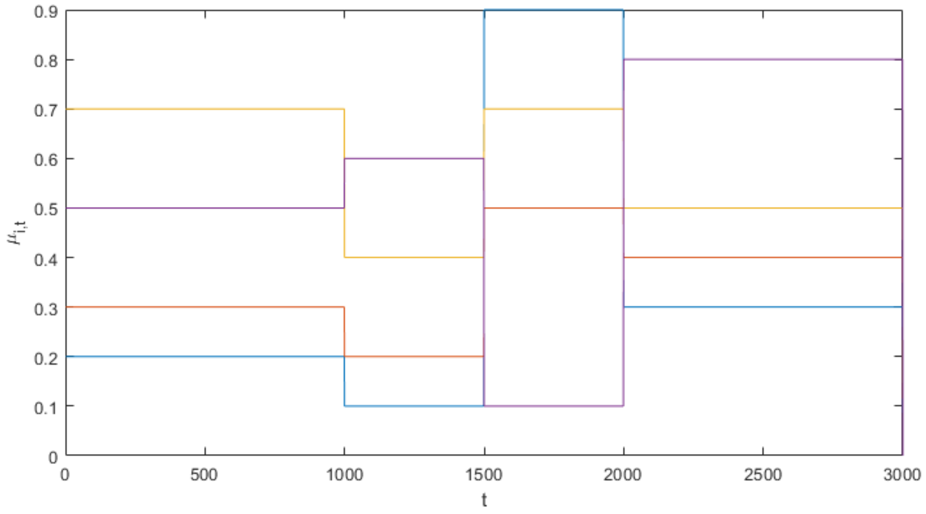
assuming the environment satisfies $H(\Delta, n) \propto n^{\beta}$

Regarding the per-round regret in the we have a dependence on $\sigma^{\frac{1}{2}}$

Matlab Exercise

- Evaluate the performance of the standard MAB in an NS-MAB environment
- Check if the standard algorithms provides sublinear regret in such a situation
- Implement a passive bandit method for the NS-MAB environment (required)
- Implement an active bandit method for the NS-MAB environment (optional)





Boracchi, Trovò

Strengthening Sequential Side-Channel Attacks Through Change Detection

Luca Frittoli, Matteo Bocchi, Silvia Mella, Diego Carrera, Beatrice Rossi, Pasqualina Fragneto, Ruggero Susella and Giacomo Boracchi

Accepted to Transactions on CHES 2020

Side Channel Attacks

Attacking Crypto Algorithms

Cryptanalysis is the art and science of analyzing information systems in order to study the hidden aspects of the systems

- Mathematical analysis of cryptographic algorithms
- Side Channel Attacks

What is a “Side Channel”?

Based on information gained from the **physical implementation** of a cryptosystem

- No theoretical weaknesses in the algorithm
- No brute force

Example

Example 2

HOME » NEWS » WORLD NEWS » EUROPE » NETHERLANDS

Dutch police catch cannabis growers after spotting snow-free roof

Police in the Netherlands have been identifying cannabis growers from the lack of snow on the roofs of their houses

INDEPENDENT

NEWS CORONAVIRUS ADVICE LOCKDOWN GUIDE US POLITICS VOICES SPORT CULTURE INDY/LIFE INDYBEST INDY100 VOUCHERS PREMIUM

News > World > Europe

Melting snow being used by police to find cannabis farms in the Netherlands

Snow-free roofs can indicate the high temperatures needed to grow the drug

Lizzie Dearden | @lizziedearden | Tuesday 10 February 2015 13:31 | 27 comments

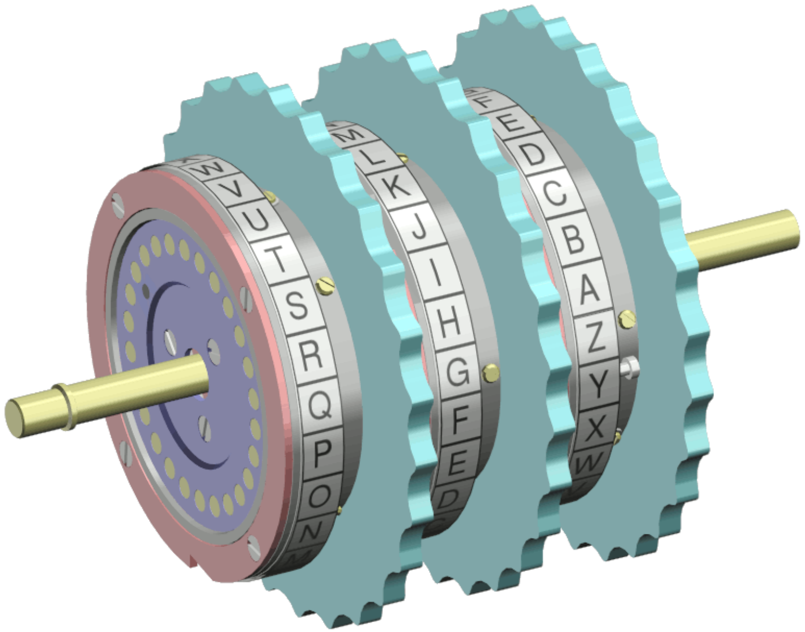
A little bit of history [1]

The first official information related to SCA attack dates back to the year 1965.

P. Wright (a scientist with GCHQ at that time) reported in [2] that MI5, the British intelligence agency, was trying to break a cipher used by the Egyptian Embassy in London, but their efforts were stymied by the limits of their computational power.

[1] YongBin Zhou, DengGuo Feng. *Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing*. IACR Eprint archive, 2005.

[2] P. Wright. *Spy Catcher: The Candid Autobiography of a Senior Intelligence Officer*. Viking Press, 1987.



A little bit of history [1]

Wright suggested placing a microphone near the rotor-cipher machine used by the Egyptian, to spy the click-sound the machine produced. By listening to the clicks of the rotors as cipher clerks reset them each morning, MI5 successfully deduced the core position of 2 or 3 of the machine’s rotors.

This additional information reduced the computation effort needed to break the cipher, and MI5 could spy on the embassy’s communication for years.

On the other hand, the original seminal works, as well as many subsequent pioneering ideas, on SCA attacks in public cryptography research community are all due to Paul Kocher, and start appearing from 1996 on.

Why “Side Channel”?

More effective against modern cryptosystems

In some applications the attacker does actually have physical access to the device

- Electronic passports, identity cards, driver licenses…
- IoT devices
- Point Of Sale
- Access Control/Badges
- Smartphone
- Car keys
- Pay TV

Use Case: Pay TV

The **key** that protects the content **is stored within the smartcard**

The **smartcard** is **provided** to the end user

- No more in the hands of the owner of the contents

Extracting one key from a single smartcard allows to program several new smartcards with the same key → **clones**

- One broken smartcard means broken system

How to do a “Side Channel”?

The attacker must have physical access to the device under attack

The attacker knows the algorithm under attack

- The only secret is the key

1st stage → Measurements

2nd stage → Analysis of the measurements

- Statistical analysis
- Application of cryptanalysis

Power Analysis

Instantaneous power consumption of a device depends on the data it processes and on the operation it performs

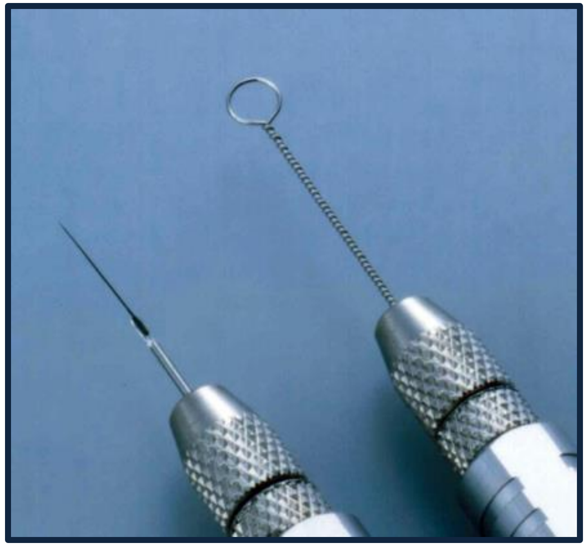
Timing Attacks

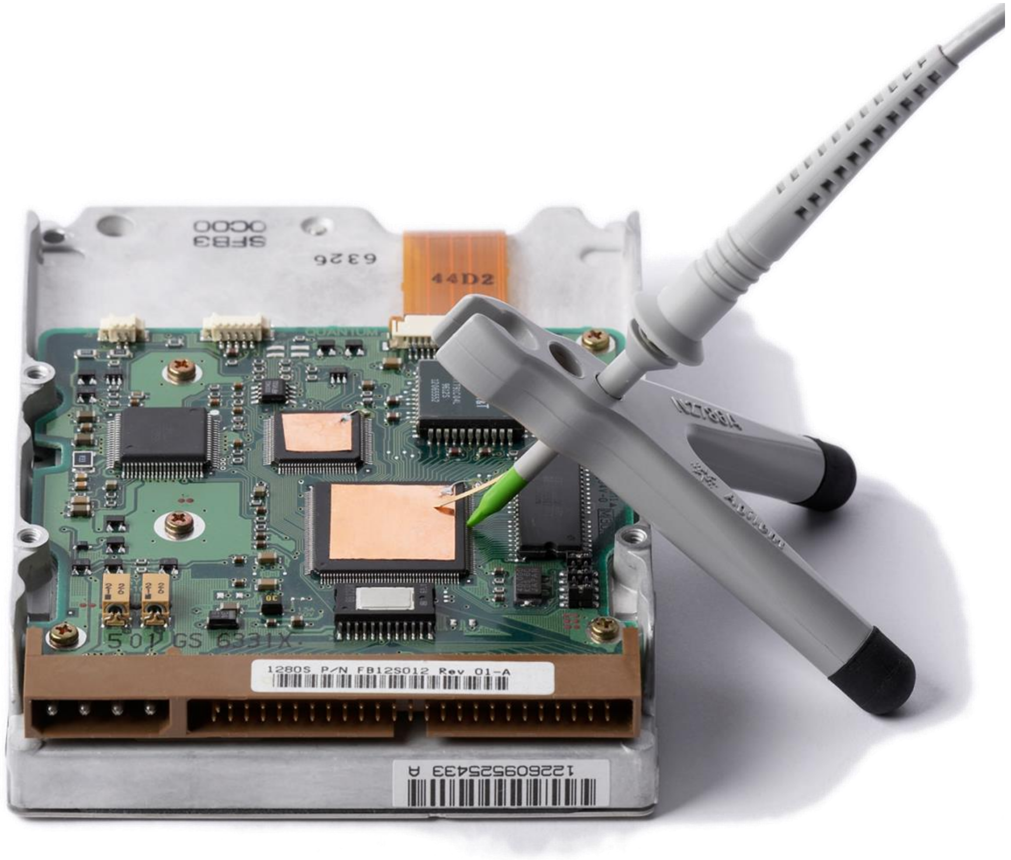
Cryptosystems often take slightly different amounts of time to process different inputs

Timing attacks can be launched against a workstation running a protocol such as SSL with RSA over a local network

Electromagnetic Analysis

The flow of current through a CMOS device induces electromagnetic emanations and causes electromagnetic leakage





Power Analysis

Basic Idea

There must be some **relationship between the device's power consumption and what it's doing**

Try to **exploit this relationship** to get the secret key

Introduced by P. Kocher, J. Jaffe, and B. Jun in 1999

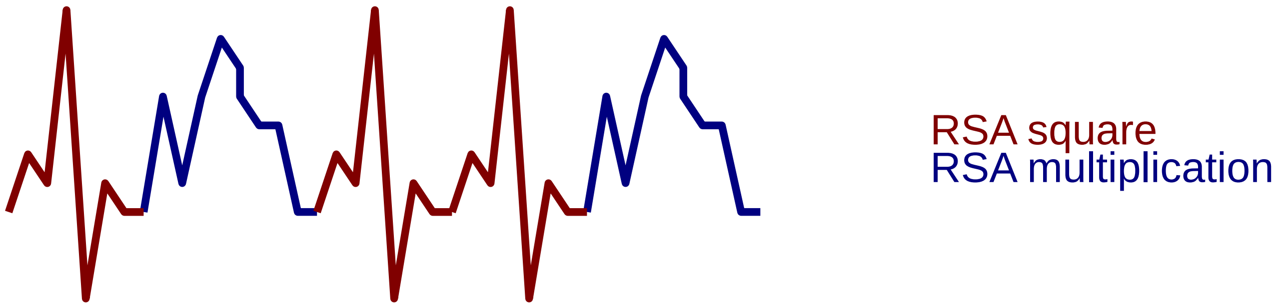
Simple Power Analysis

Observation on a single power trace during the computation of the crypto algorithm

Try to distinguish between different operations related to the value of the secret key (patterns)

Example: RSA algorithms scans the private key bit by bit

- Performs a Square if bit is 0, otherwise performs a Square and a Multiplication
- If attackers can distinguish operations, hey will get the key

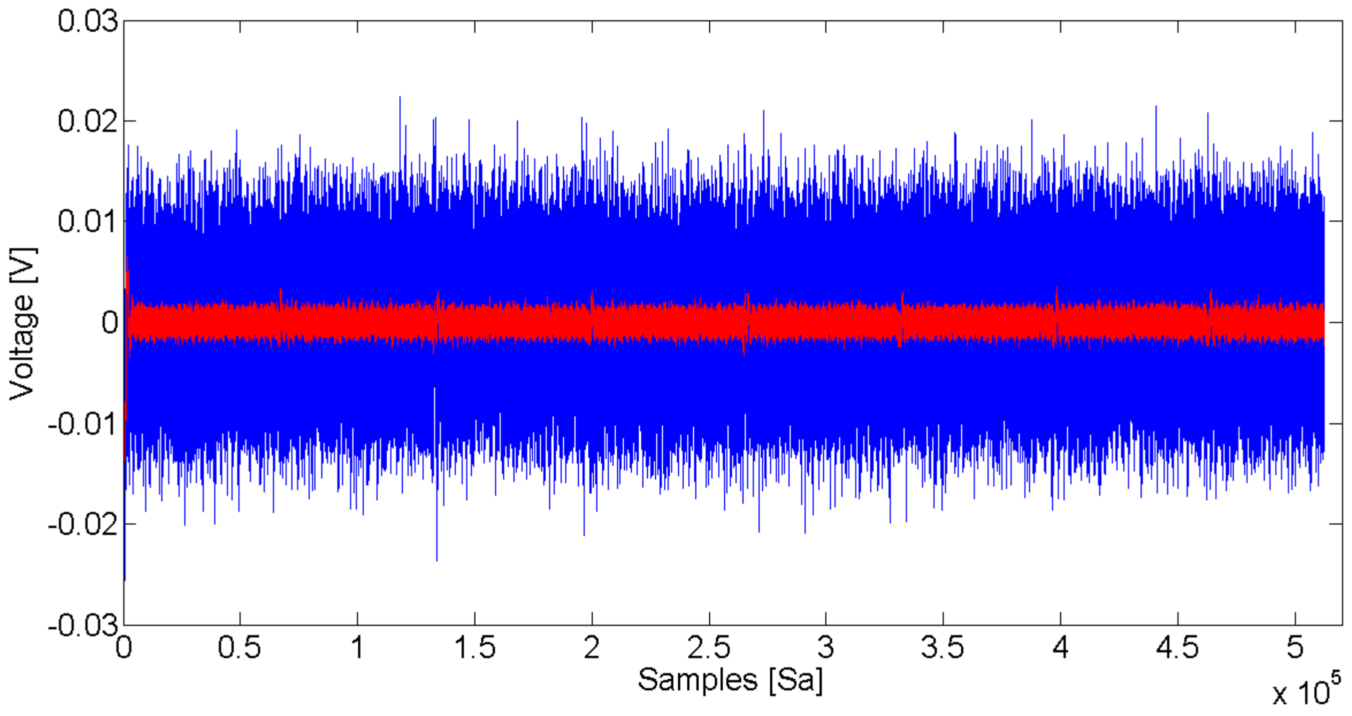


Limit of Simple Power Analysis

Requires to analyze a single power trace with very high accuracy

Usually noise is high and it is not possible to perform this kind of analysis

- Noise is due to several factors but mainly due to other activity linked to power consumption and measurement



Differential Power Analysis (DPA)

Requires a **large numbe of power traces**

- Each trace corresponds to a single execution

Each execution is done with a **different input** (plaintext)**, but same key**

Therefore we obtain different power traces corresponding to execution with **different input/plaintext** values but **same key**

Plaintext and/or ciphertext should be known by the attacker

- A common assumption which is also true in most real applications

No detailed knowledge of the cryptographic device is required

Can work even with noisy power traces

- The more the power traces the more the noise can be reduced

Consumption Model

Instantaneous power consumption in digital CMOS devices:

$$P(t) = P_{const}(t) + P_{instr}(t) + P_{data}(t) + P_{noise}(t)$$

- P_{const} (t) is unimportant for DPA
- P_{instr} (t) is fixed by the particular instruction executed
- P_{data}(t) is due to the currently processed data
- P_{noise} (t) has to be minimized
- DPA exploits the difference in multiple measurements $P(t)$ due to the $P_{data}(t)$

The basic idea is to associate the device power consumption with the values processed

Hamming Weight Model

Try to estimate $P_{data}(t)$

Based on the fact that a bit set to 1 consumes more than a bit set to 0

Very simple model, yet still in use today

Sometimes the Hamming Distance Model is preferable

- It measure the **transitions** (the bit which are changing their values) of a signal or register

DPA: (1/4)

Collect the side channel of the execution of the algorithm providing different inputs

- $Input_0$ → $Trace_0$ =  : 
- $Input_1$ → $Trace_1$ =  : 
- $Input_n$ → $Trace_n$ =  : 

Identify a sensitive variable in the algorithm

- E.g. SV is the result of the following operation Input[0] XOR Key[0]
- The target is Key[0]

DPA: (2/4)

For all recorded $\text{Input}_{0\ldots n}$, and for all possible $\text{Key}[0] = 1, \ldots, m$ compute

$$HW(Input_i[0]\ XOR\ j)$$

Create a table of guesses:

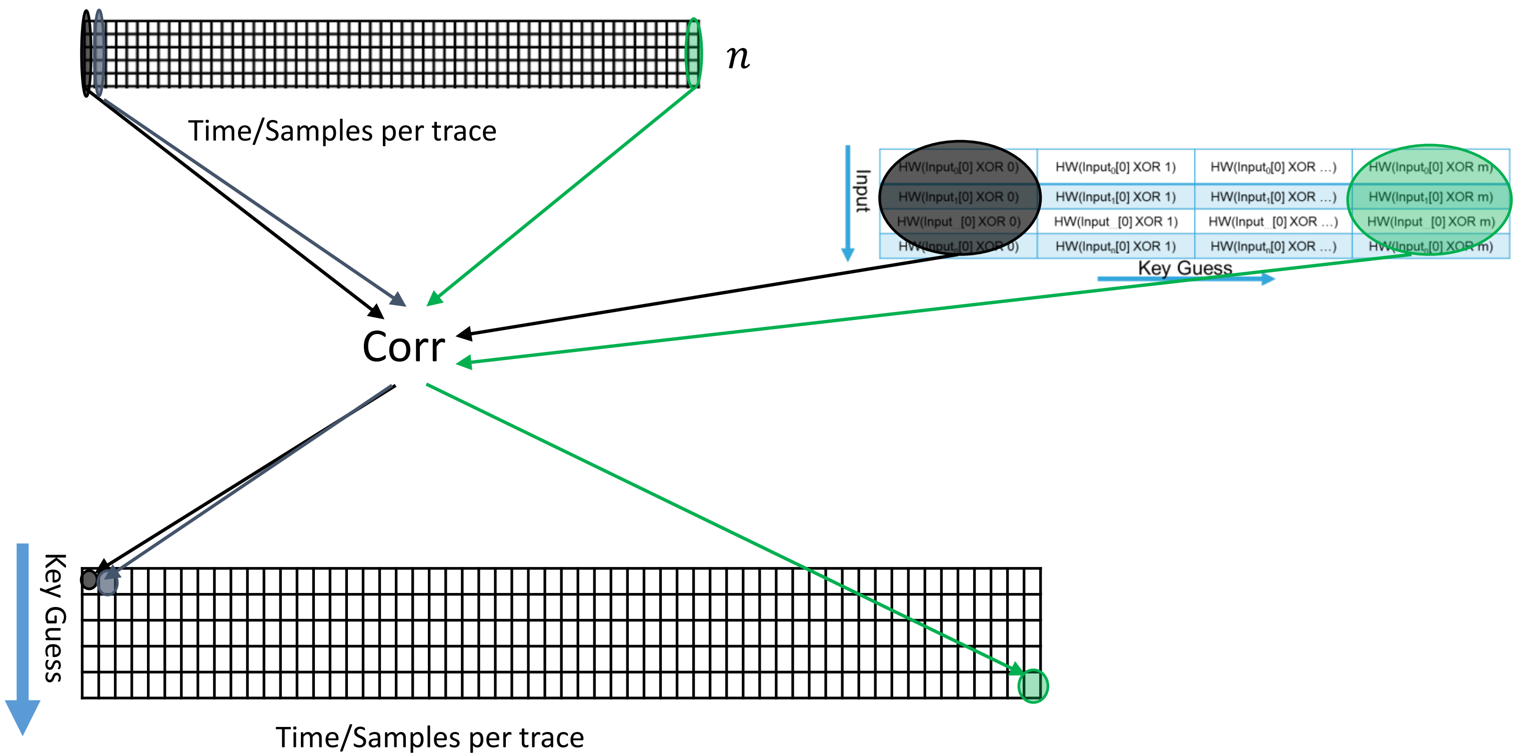
Input ↓

HW(Input_0[0] XOR 0)	HW(Input_0[0] XOR 1)	HW(Input_0[0] XOR …)	HW(Input_0[0] XOR m)
HW(Input_1[0] XOR 0)	HW(Input_1[0] XOR 1)	HW(Input_1[0] XOR …)	HW(Input_1[0] XOR m)
HW($\text{Input}_{\ldots}$[0] XOR 0)	HW($\text{Input}_{\ldots}$[0] XOR 1)	HW($\text{Input}_{\ldots}$[0] XOR …)	HW($\text{Input}_{\ldots}$[0] XOR m)
HW(Input_n[0] XOR 0)	HW(Input_n[0] XOR 1)	HW(Input_n[0] XOR …)	HW(Input_n[0] XOR m)

Key Guess →

DPA: Basic Idea (3/4)

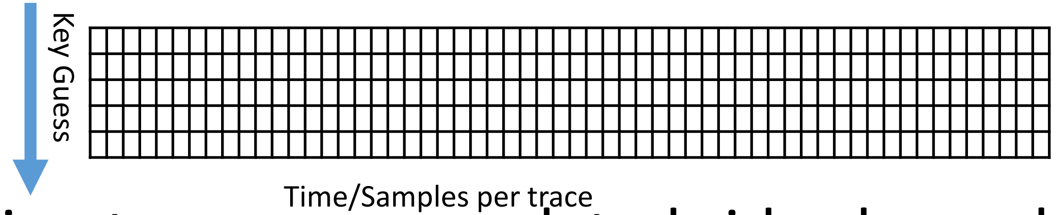
Create a matrix with the n recorded power traces



For each column (time sample) compute the correlation coefficient with every column in the guess table. Each row corresponds to a different key

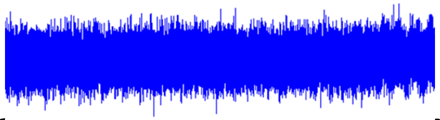
DPA: Basic Idea (4/4)

Result is a matrix of correlation traces (1 per each key guess)



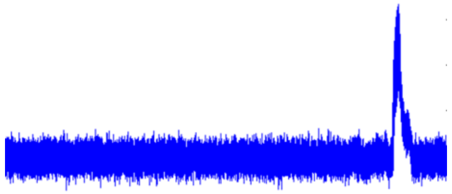
In (m-1) correlation traces we correlated side channel traces with intermediate variables which are never computed

- Because the key is wrong
- So it's like correlating with a random vector
  - Expected correlation is close to zero



But in 1 correlation traces we correlated side channel traces with intermediate variables that are actually computed

- At some point in time, when our sensitive variable is computed, we expect a peak towards 1



Timing Attacks

What is a Timing Attack

A side channel attack in which the attacker attempts to compromise a cryptosystem by analyzing the time taken to execute cryptographic algorithms

In some cases, exploitable from remote locations

Effective when computational timings depends on secret

Need to have encryption timings with high accuracy

- Noise and sensitivity must be lower than the timing difference we want to measure

Vulnerability comes from…

Sometimes is a matter of algorithm

- Often, algorithms leaks information through timings difference because computational steps depend on data values
- Choose a constant-time algorithm to avoid these attacks
- E.g. Modular exponentiation (we will see it later) can be done with Square&Multiply algorithm (variable-time) or with Square&Multiply Always (constant-time)

Otherwise, can be a matter of implementation

- Cache-Timing Attack takes advantage of data-dependent timing variations during accesses into the cache (greater computational time for cache miss)
- It exploits implementations in which secret data is used as an array index (e.g. AES Sbox)
- Almost every implementation can be made constant-time in order to avoid these attacks

So, what about change-detection?

Sequential Analysis

Recovering a single-key value is pointless, we need to recover the whole sequence of key values to break the message

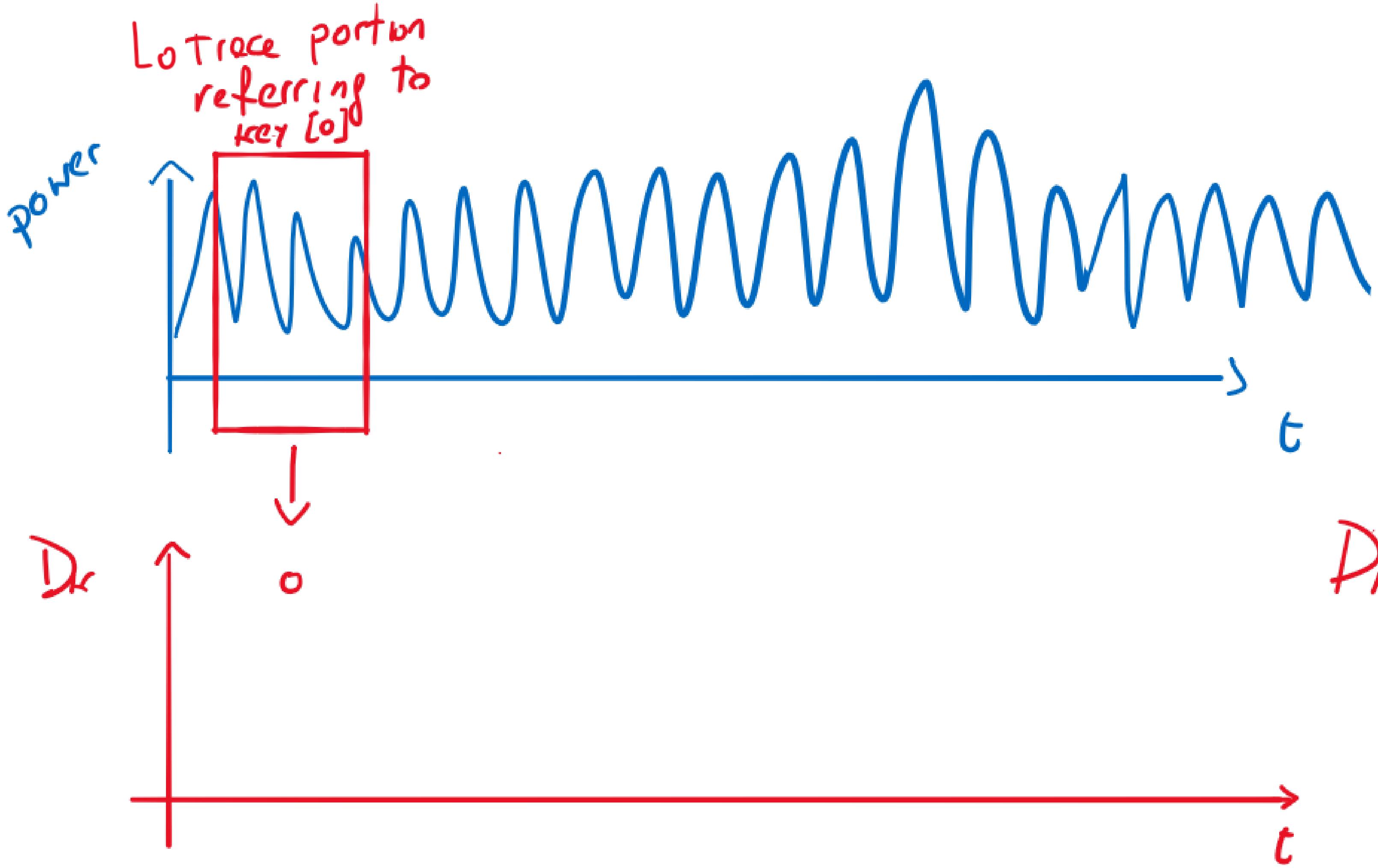
- When a single bit of the key is wrong, **the recovered message will be completely wrong**
- **You need to carry out the entire attack before realizing that it was unsuccessful**

We can repeat the analysis over multiple portions L_k of the power trace

- to compute $HW(Inputi[1]\ XOR\ j)$ we can estimate $Input_i[1]$ by leveraging the first key guess $key[0]$
- … and iterate until we get to $key[\text{end}]$

Sequential Analysis

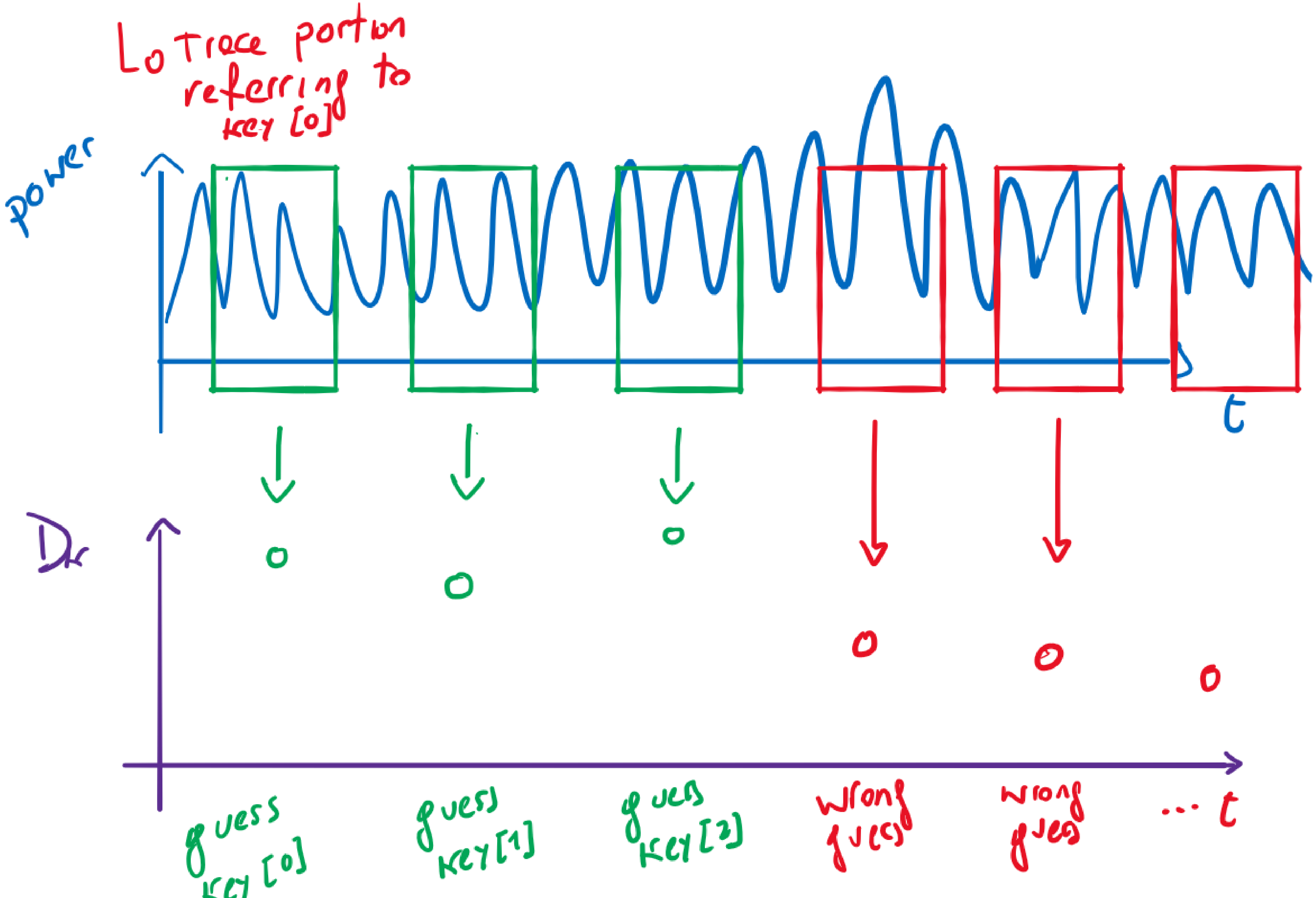
For the sake of simplicity, assume we can infer the key from a single trace



$$D_K = \max_{x \in X} \rho(L_K, H_K^x)$$

Boracchi, Trovò

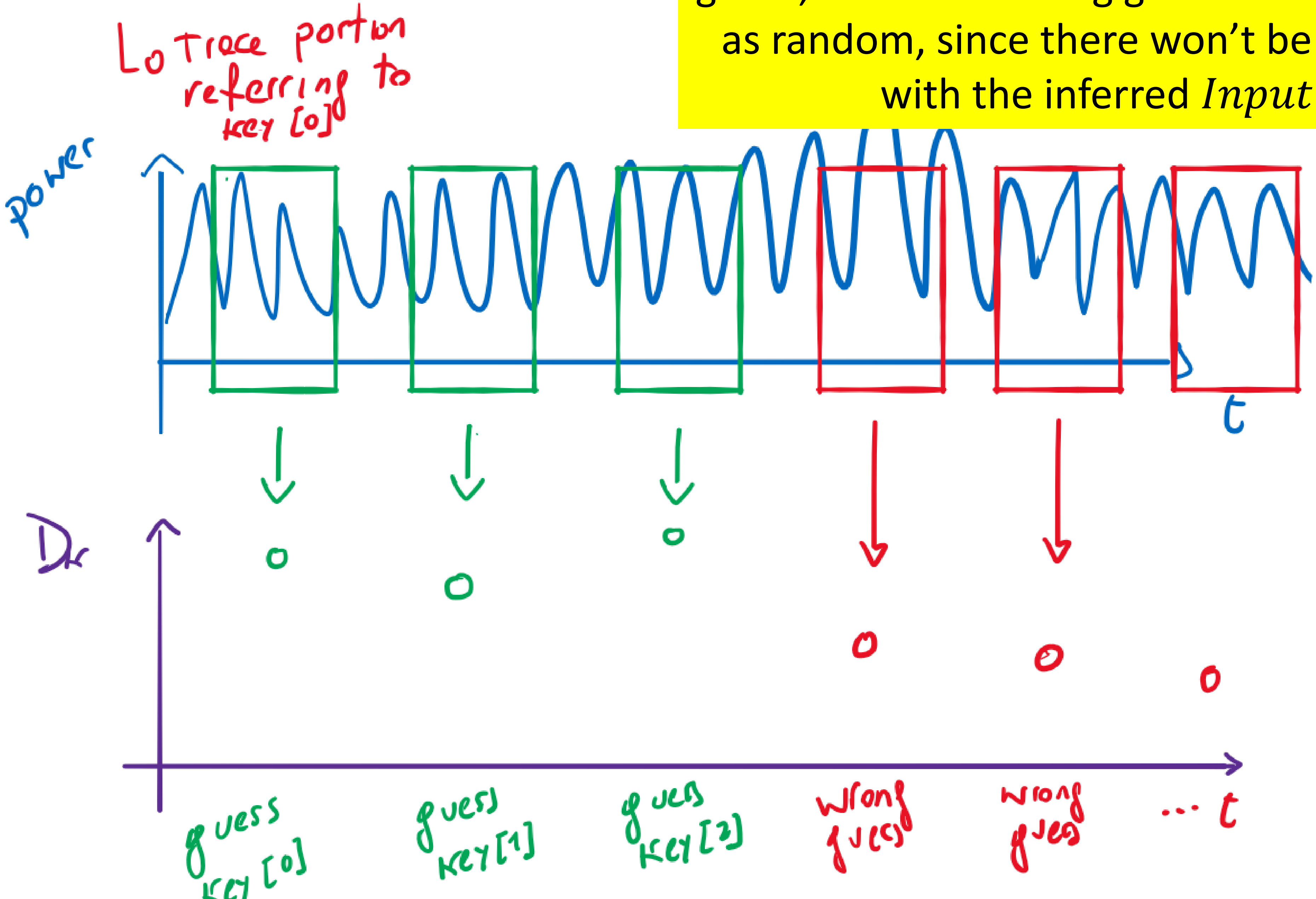
Sequential Analysis



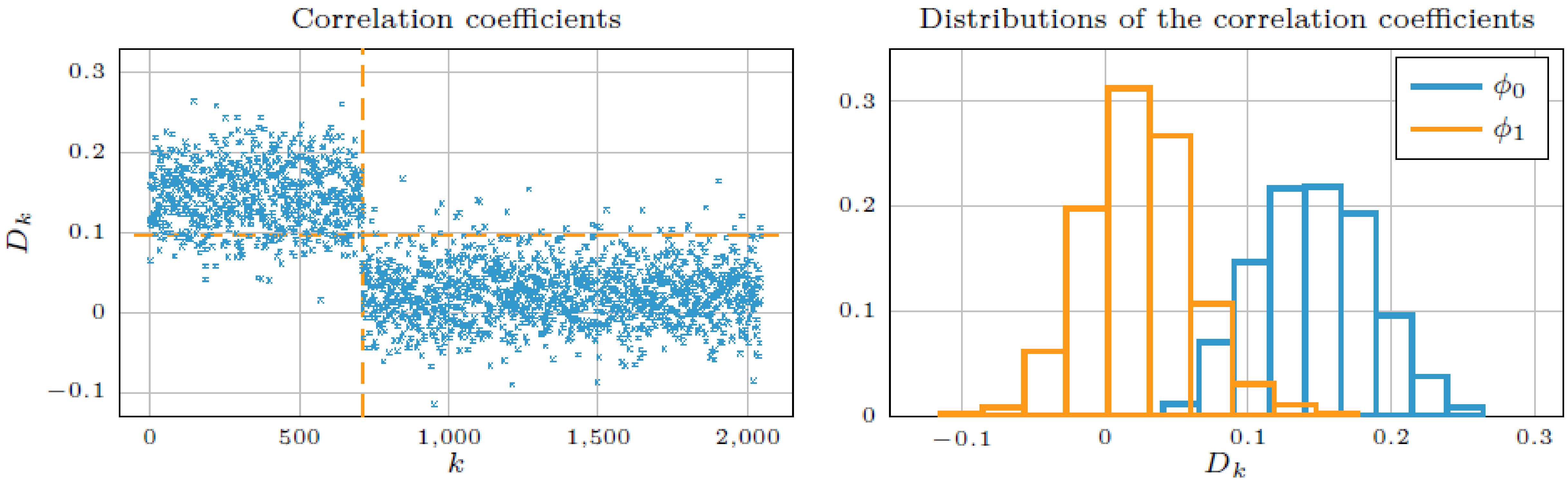
Boracchi, Trovò

Sequential Analysis

Once, for some reason, we make a wrong key guess, all the following guesses will be as good as random, since there won't be correlation with the inferred $Input[t]$

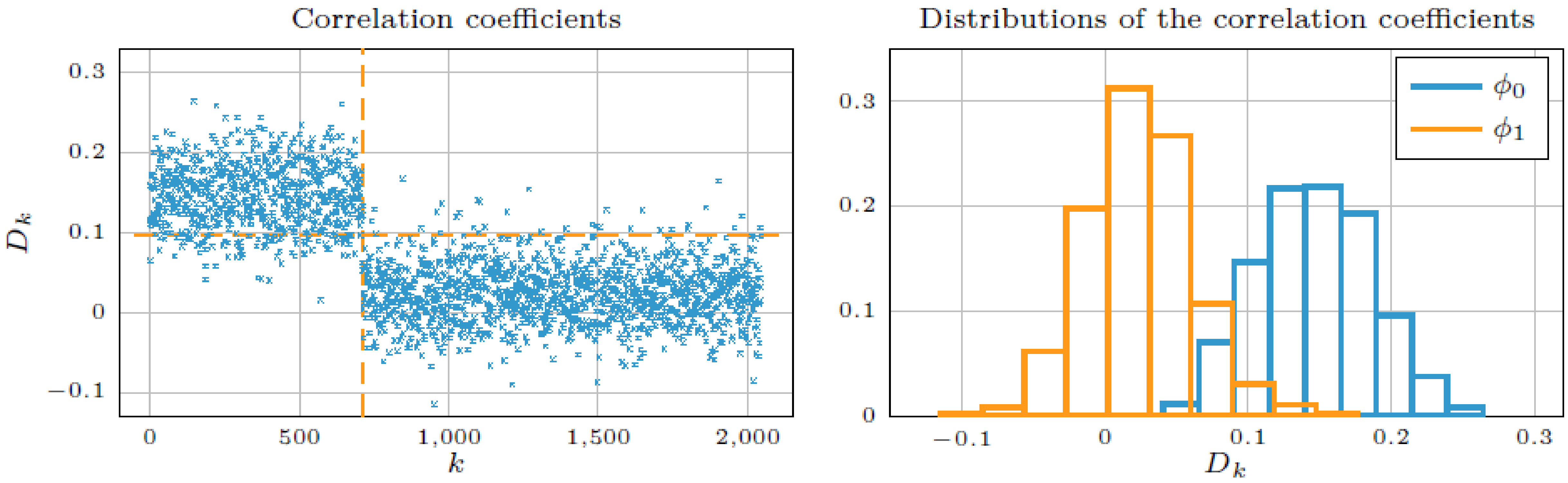


Wrong guesses introduce distribution changes!



Boracchi, Trovò

Wrong guesses introduce distribution changes!



Detecting distribution changes in the correlation coefficient means detecting wrong key guesses!

Boracchi, Trovò

Change Detection over Correlation Values

We detect distribution changes in the sequence of correlation values by means of a sequential monitoring scheme.

We adopted an online and nonparametric CDT: online CPM based on the Lepage statistics

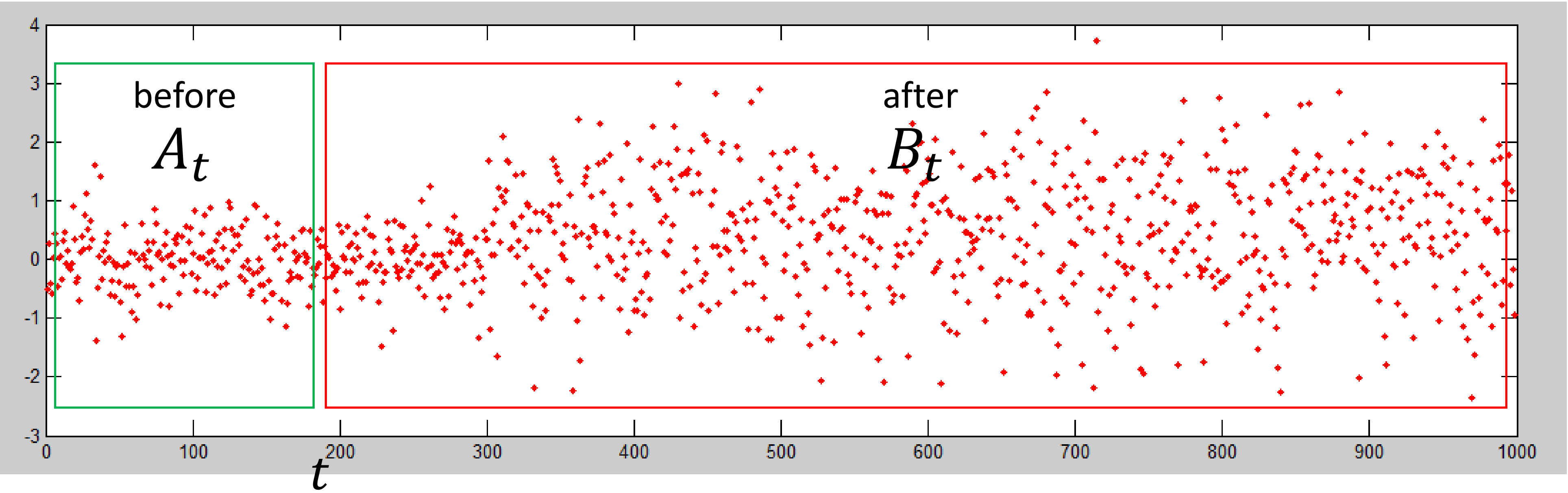
Nonparametric Monitoring of Data Streams for Changes in Location and Scale

Gordon J. Ross, Dimitris K. Tasoulis, and Niall M. Adams

Department of Mathematics
Imperial College London
London, SW7 2AZ, U.K.
(*gordon.ross03@imperial.ac.uk*; *d.tasoulis@imperial.ac.uk*;
n.adams@imperial.ac.uk)

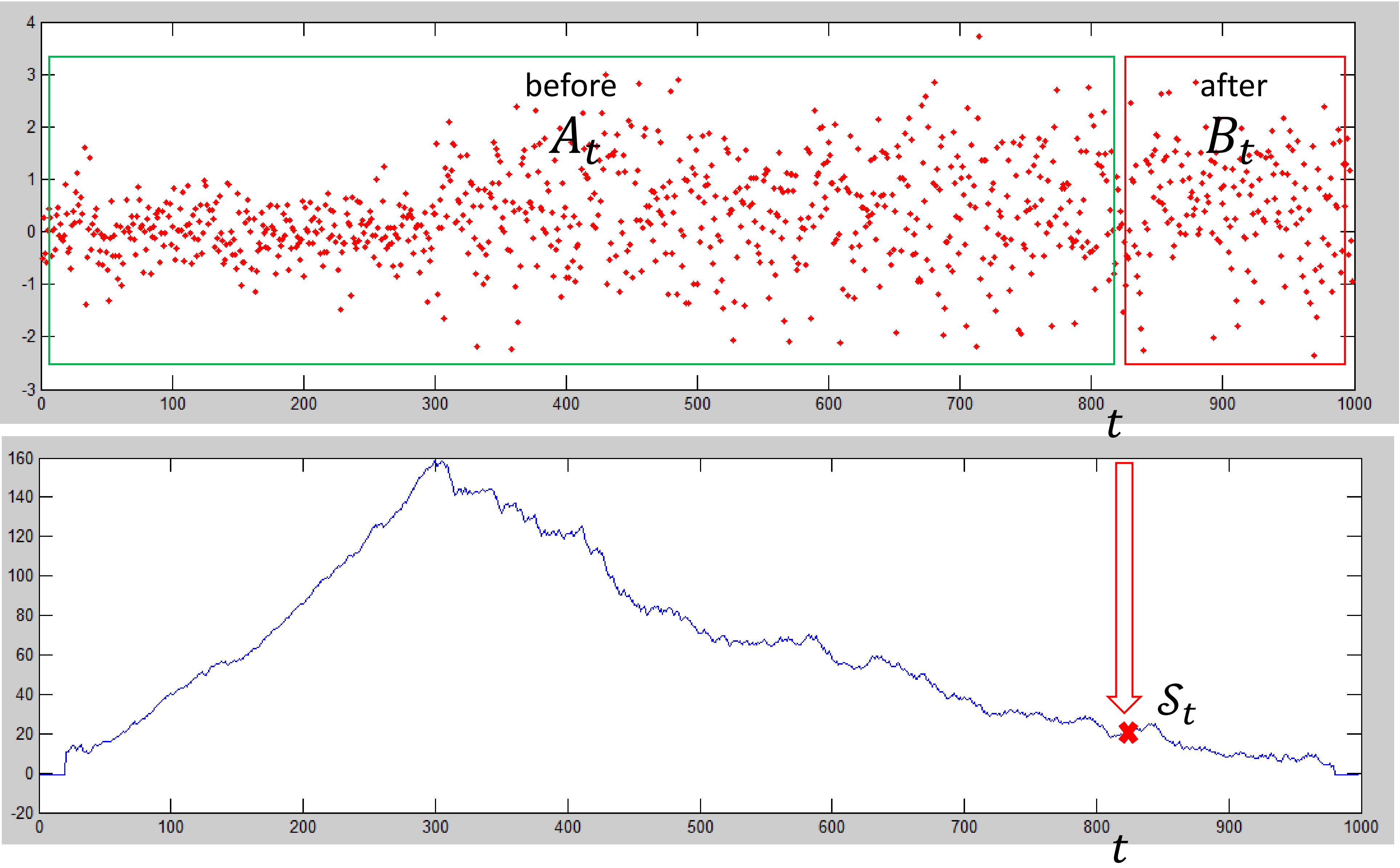
G. J. Ross, D. K. Tasoulis, and N. M. Adams, "Nonparametric monitoring of data streams for changes in location and scale," Technometrics, vol. 53, no. 4, pp. 379–389, 2011.

The Change Point Method (CPM)



- Test a single point t to be a change point
- Split the dataset in two sets $A_t, B_t \subset X$, namely samples «before» and «after» the putative change at t
- **Compute a test statistic** $\mathcal{S}_t$ 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

The Change Point Method (CPM)



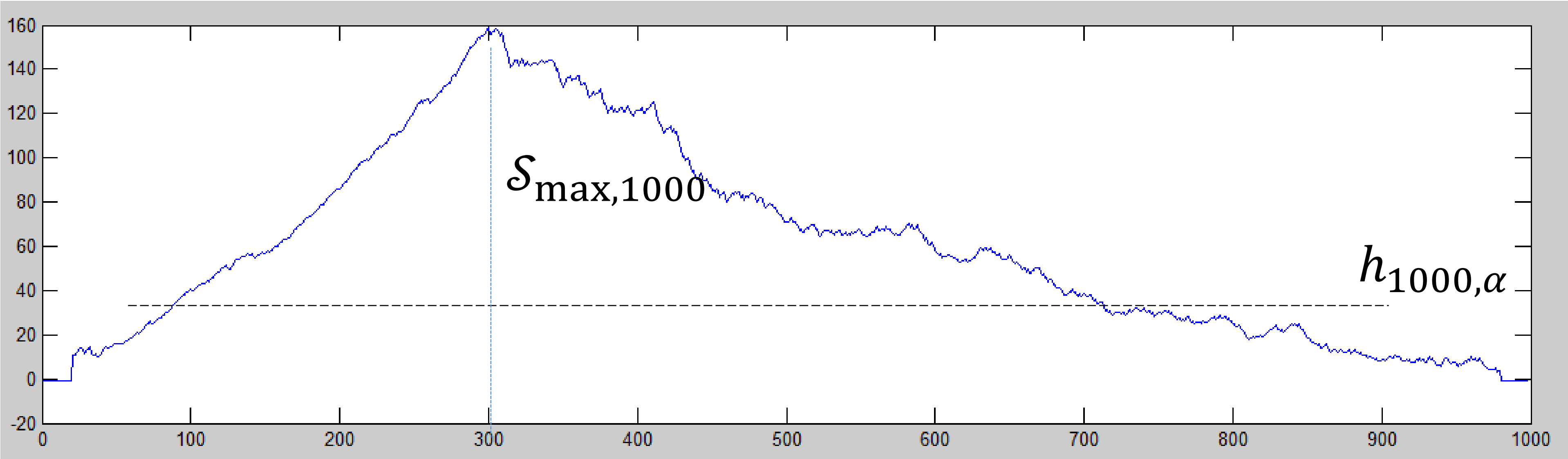
Boracchi, Trovò

The 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 $\mathcal{S}_{\max,1000}$ 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



Strenghtened Sequential Attack

Algorithm 3 Strengthened sequential attack

Input: target algorithm, ciphertext c, side channel $\{\mathbf{L}_k\}_{k=1}^{K}$, distinguisher $\mathcal{D}$, set of possible window lengths $\mathbf{S}$

Output: estimated secret key $\hat{d}$

$\widehat{O}_1$ is initialized as in Algorithm 1
for $k = 1, \ldots, K$ **do**
  $\hat{d}[k] \leftarrow \arg\max_{\mathbf{x} \in \mathbf{X}} \mathcal{D}(\mathbf{x}, \widehat{O}_k, \mathbf{L}_k)$ // Algorithm 2, line 3
  $\widehat{O}_{k+1} \leftarrow \texttt{operations}(\hat{d}[k], \widehat{O}_k, c)$ // Algorithm 2, line 4
  $D_k \leftarrow \max_{\mathbf{x} \in \mathbf{X}} \mathcal{D}(\mathbf{x}, \widehat{O}_k, \mathbf{L}_k)$ // save the distinguisher value D_k
  $\mathbf{D} \leftarrow (\mathbf{D}, D_k)$ // append D_k to the sequence $\mathbf{D}$
  change_detected, $\hat{\tau} \leftarrow \mathrm{CDT}(\mathbf{D})$ // error detection
  if change_detected // error correction **then**
    align $\hat{\tau}$ and k
    for each $w \in \mathbf{S}$ **do**
      succ_correction, $\mathbf{x}_{\text{best}} \leftarrow \texttt{correction}(\hat{\tau}, w, \mathbf{D})$ // Algorithm 4
      if succ_correction **then**
        break
      end if
    end for
    $(\hat{d}[\hat{\tau} - u], \ldots, \hat{d}[\hat{\tau} + u]) \leftarrow \mathbf{x}_{\text{best}}, \quad k \leftarrow \hat{\tau} + u + 1$
    $\mathbf{D} \leftarrow \mathbf{D} \setminus (D_{\hat{\tau}-u}, \ldots, D_{\hat{\tau}+u})$
  end if
end for
return $\hat{d} \leftarrow (\hat{d}[1], \ldots, \hat{d}[K])$

Luca Frittoli, Matteo Bocchi, Silvia Mella, Diego Carrera, Beatrice Rossi, Pasqualina Fragneto, Ruggero Susella and Giacomo Boracchi "Strengthening Sequential Side-Channel Attacks Through Change Detection", Accepted on TCHES 2020

Strenghtened Sequential Attack

Algorithm 3 Strengthened sequential attack

Input: target algorithm, ciphertext c, side channel $\{\mathbf{L}_k\}_{k=1}^{K}$, distinguisher $\mathcal{D}$, set of possible window lengths $\mathbf{S}$
Output: estimated secret key $\hat{d}$

$\widehat{O}_1$ is initialized as in Algorithm 1
for $k = 1, \dots, K$ **do**
  $\hat{d}[k] \leftarrow \arg\max_{\mathbf{x} \in \mathbf{X}} \mathcal{D}(\mathbf{x}, \widehat{O}_k, \mathbf{L}_k)$ // Algorithm 2, line 3
  $\widehat{O}_{k+1} \leftarrow \texttt{operations}(\hat{d}[k], \widehat{O}_k, c)$ // Algorithm 2, line 4
  $D_k \leftarrow \max_{\mathbf{x} \in \mathbf{X}} \mathcal{D}(\mathbf{x}, \widehat{O}_k, \mathbf{L}_k)$ // save the distinguisher value D_k
  $\mathbf{D} \leftarrow (\mathbf{D}, D_k)$ // append D_k to the sequence $\mathbf{D}$
  $\text{change\_detected}, \hat{\tau} \leftarrow \text{CDT}(\mathbf{D})$ // error detection
  if change_detected // error correction **then**
    align $\hat{\tau}$ and k
    for each $w \in \mathbf{S}$ **do**
      $\text{succ\_correction}, \mathbf{x}_{\text{best}} \leftarrow \texttt{correction}(\hat{\tau}, w, \mathbf{D})$ // Algorithm 4
      if succ_correction **then**
        break
      end if
    end for
    $(\hat{d}[\hat{\tau} - u], \dots, \hat{d}[\hat{\tau} + u]) \leftarrow \mathbf{x}_{\text{best}}, \quad k \leftarrow \hat{\tau} + u + 1$
    $\mathbf{D} \leftarrow \mathbf{D} \setminus (D_{\hat{\tau}-u}, \dots, D_{\hat{\tau}+u})$
  end if
end for
return $\hat{d} \leftarrow (\hat{d}[1], \dots, \hat{d}[K])$

Luca Frittoli, Matteo Bocchi, Silvia Mella, Diego Carrera, Beatrice Rossi, Pasqualina Fragneto, Ruggero Susella and Giacomo Boracchi "Strengthening Sequential Side-Channel Attacks Through Change Detection", Accepted on TCHES 2020

Strenghtened Sequential Attack

Algorithm 3 Strengthened sequential attack

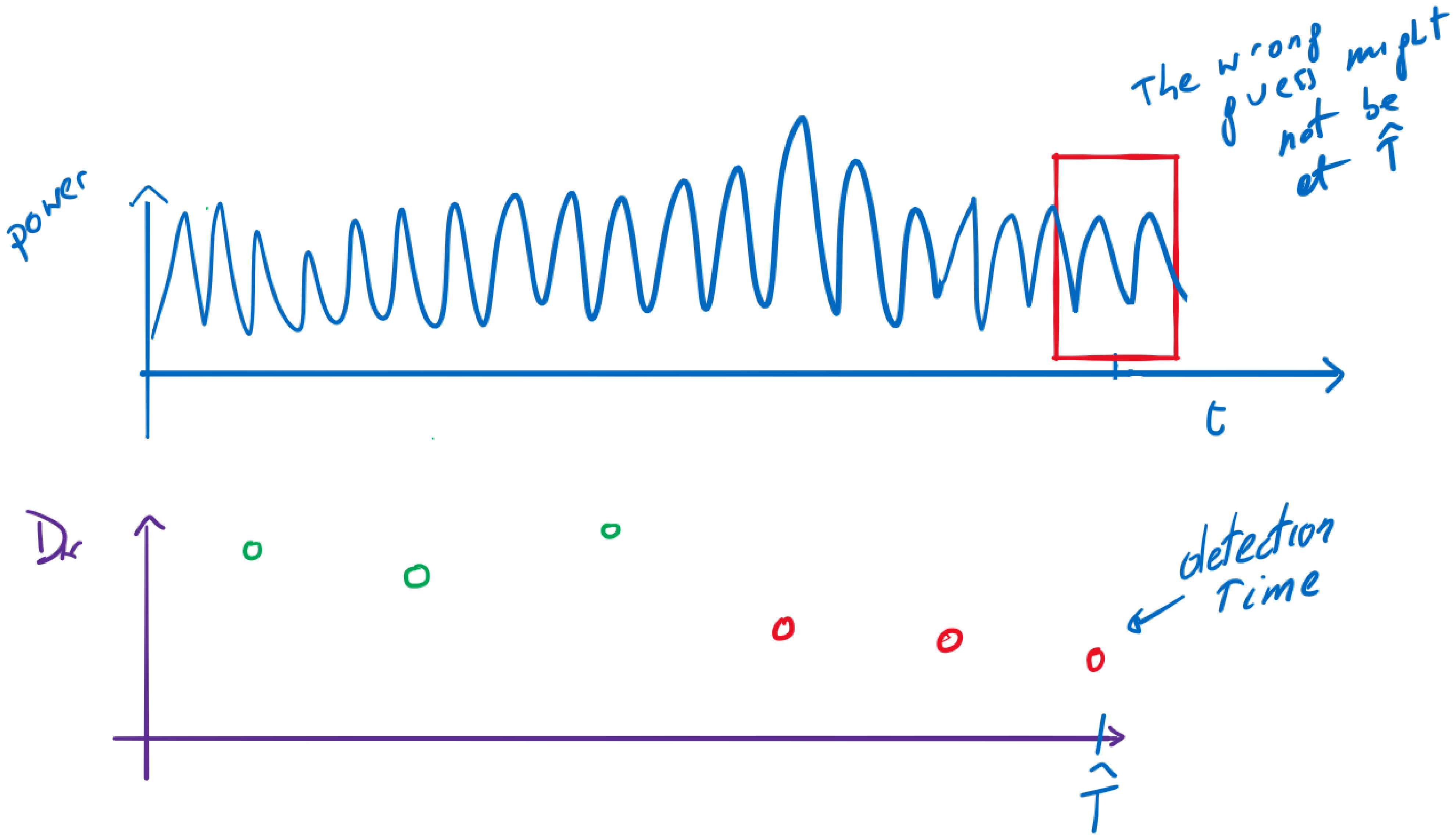
Input: target algorithm, ciphertext c, side channel $\{\mathbf{L}_k\}_{k=1}^{K}$, distinguisher $\mathcal{D}$, set of possible window lengths $\mathbf{S}$

Output: estimated secret key $\hat{d}$

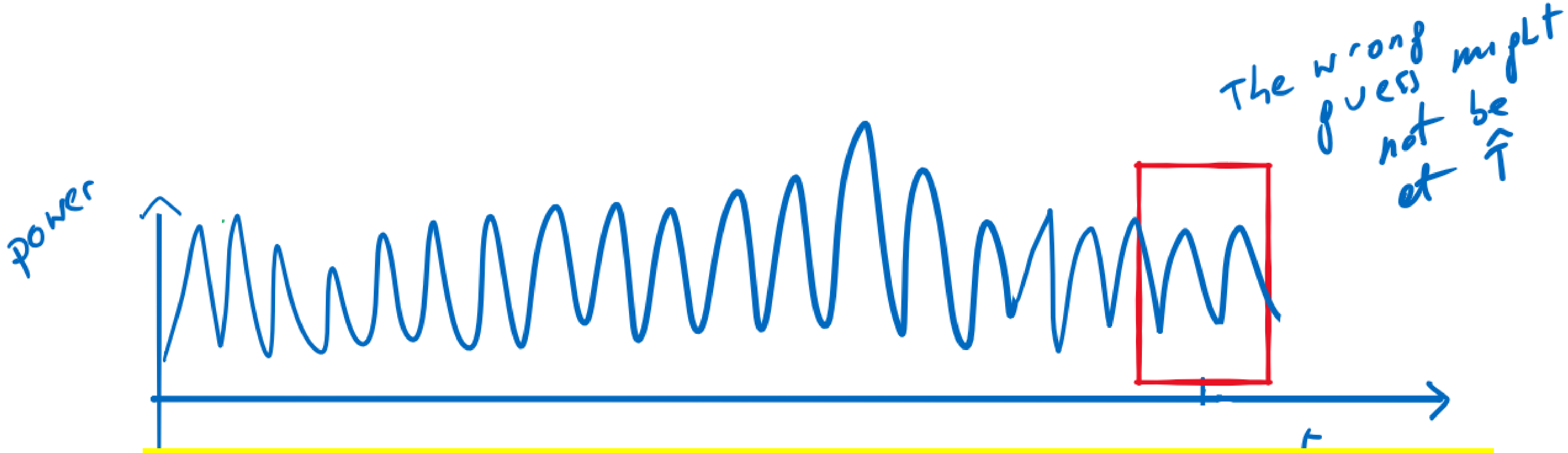
$\widehat{O}_1$ is initialized as in Algorithm 1
for $k = 1, \dots, K$ **do**
    $\hat{d}[k] \leftarrow \arg\max_{\mathbf{x} \in \mathbf{X}} \mathcal{D}(\mathbf{x}, \widehat{O}_k, \mathbf{L}_k)$ // Algorithm 2, line 3
    $\widehat{O}_{k+1} \leftarrow \texttt{operations}(\hat{d}[k], \widehat{O}_k, c)$ // Algorithm 2, line 4
    $D_k \leftarrow \max_{\mathbf{x} \in \mathbf{X}} \mathcal{D}(\mathbf{x}, \widehat{O}_k, \mathbf{L}_k)$ // save the distinguisher value D_k
    $\mathbf{D} \leftarrow (\mathbf{D}, D_k)$ // append D_k to the sequence $\mathbf{D}$
    change_detected, $\hat{\tau} \leftarrow \mathrm{CDT}(\mathbf{D})$ // error detection
    if change_detected // error correction **then**
        align $\hat{\tau}$ and k
        for each $w \in \mathbf{S}$ **do**
            succ_correction, $\mathbf{x}_{\text{best}} \leftarrow \texttt{correction}(\hat{\tau}, w, \mathbf{D})$ // Algorithm 4
            if succ_correction **then**
                break
            end if
        end for
        $(\hat{d}[\hat{\tau} - u], \dots, \hat{d}[\hat{\tau} + u]) \leftarrow \mathbf{x}_{\text{best}}, \quad k \leftarrow \hat{\tau} + u + 1$
        $\mathbf{D} \leftarrow \mathbf{D} \setminus (D_{\hat{\tau}-u}, \dots, D_{\hat{\tau}+u})$
    end if
end for
return $\hat{d} \leftarrow (\hat{d}[1], \dots, \hat{d}[K])$

Luca Frittoli, Matteo Bocchi, Silvia Mella, Diego Carrera, Beatrice Rossi, Pasqualina Fragneto, Ruggero Susella and Giacomo Boracchi "Strengthening Sequential Side-Channel Attacks Through Change Detection", Accepted on TCHES 2020

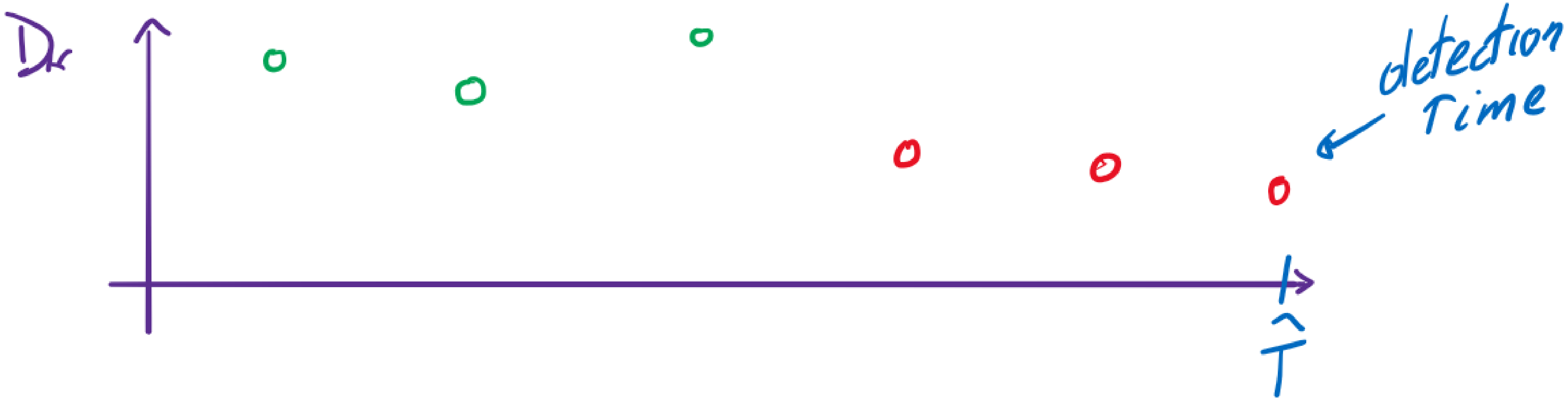
Correction Procedure



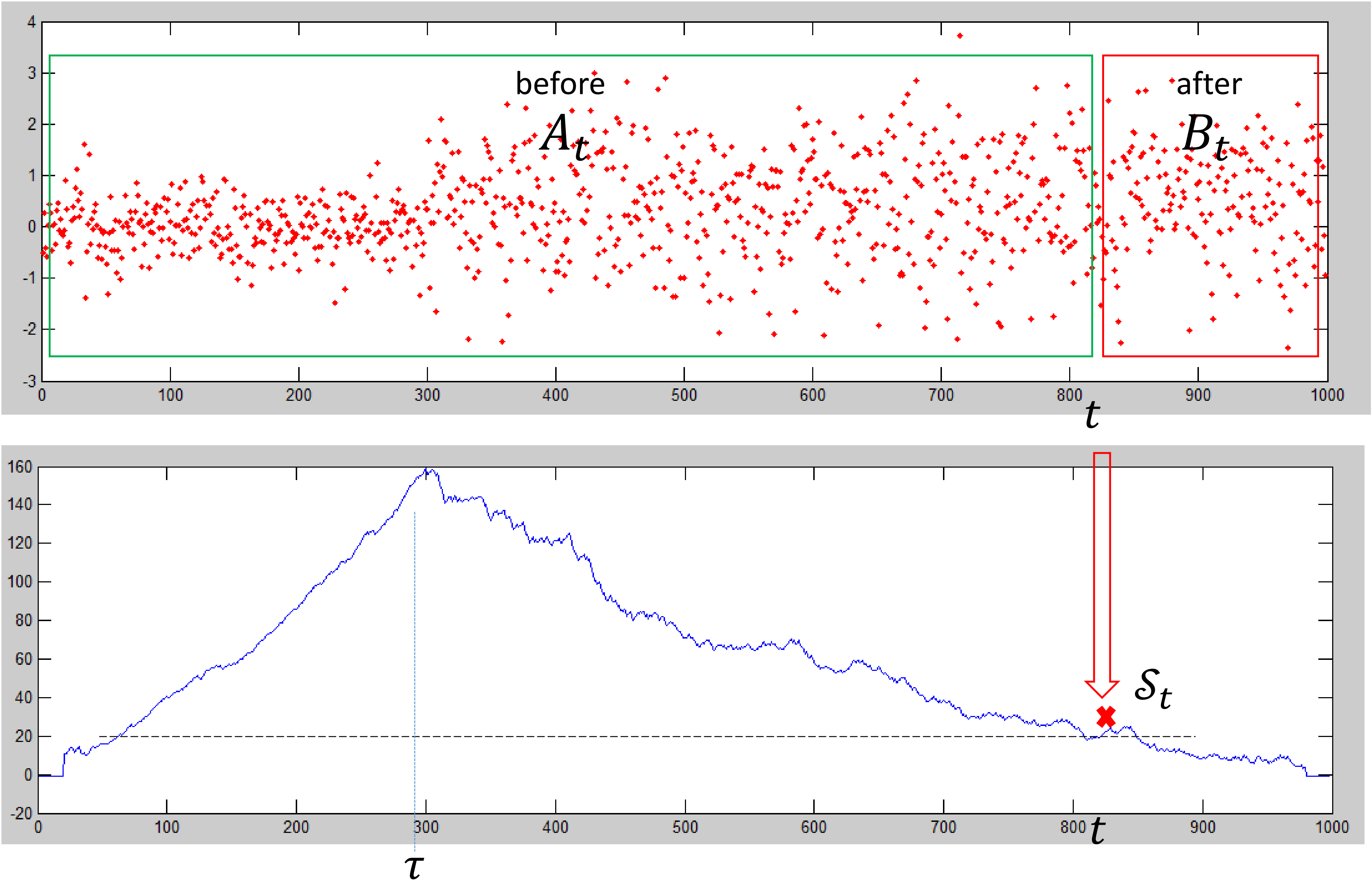
Correction Procedure



The wrong guess rarely occured at the detection time $\hat{T}$



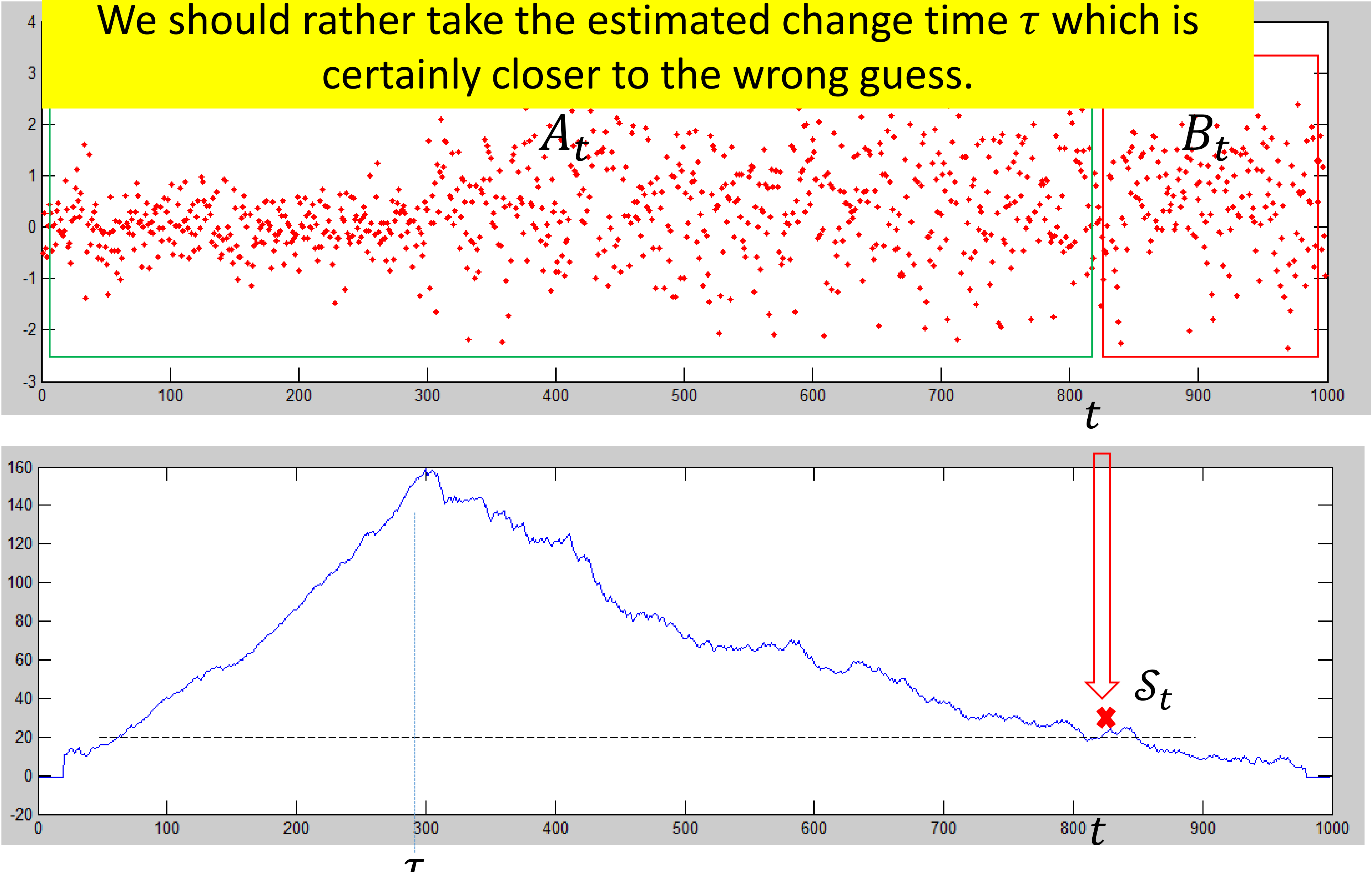
The Change Point Method (CPM)



Boracchi, Trovò

The Change Point Method (CPM)

We should rather take the estimated change time τ which is certainly closer to the wrong guess.



Boracchi, Trovò

Correction Procedure

It is however not enough to simply flip the key value at τ, because:

- The wrong guess might just be nearby τ
- Detection might be a false positive! Namely, a detection was fired but no change has occurred in the sequence.

We should rather crop a sequence W_τ around τ and perform a brute-force search for the most likely key value over there.

- We can increase the sizes of W_τ until we find to a key portion yielding enough statistical evidence that the sequence is stationary.

Correction Procedure

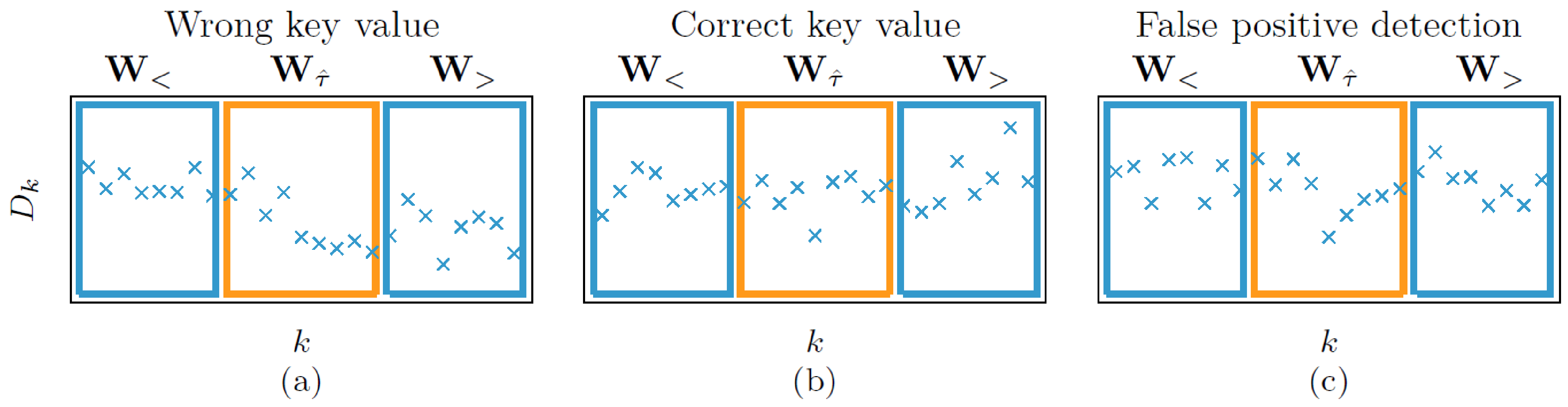


Figure 2: Examples of distinguisher values contained in the windows $\mathbf{W}_{\hat{\tau}}$, $\mathbf{W}_<$, $\mathbf{W}_>$ and the rationale behind our correction procedure. In (a) the key value tested in $\mathbf{W}_{\hat{\tau}}$ is wrong, thus $\mathbf{W}_<$ and $\mathbf{W}_>$ have different distributions. In (b), the tested key value is correct, thus $\mathbf{W}_<$ and $\mathbf{W}_>$ follow the same distribution. In (c), $\hat{\tau}$ is a false positive detection, thus the distributions of $\mathbf{W}_<$ and $\mathbf{W}_>$ coincide, but this is different around the detection, i.e. in $\mathbf{W}_{\hat{\tau}}$. This latter illustration indicates why it is necessary not to include $\mathbf{W}_{\hat{\tau}}$ in the error correction, and also to remove this window from $\mathbf{D}$ when the monitoring restarts.

Luca Frittoli, Matteo Bocchi, Silvia Mella, Diego Carrera, Beatrice Rossi, Pasqualina Fragneto, Ruggero Susella and Giacomo Boracchi "Strengthening Sequential Side-Channel Attacks Through Change Detection", Accepted on TCHES 2020

Strenghtened Sequential Attacks

Algorithm 4 Correction procedure

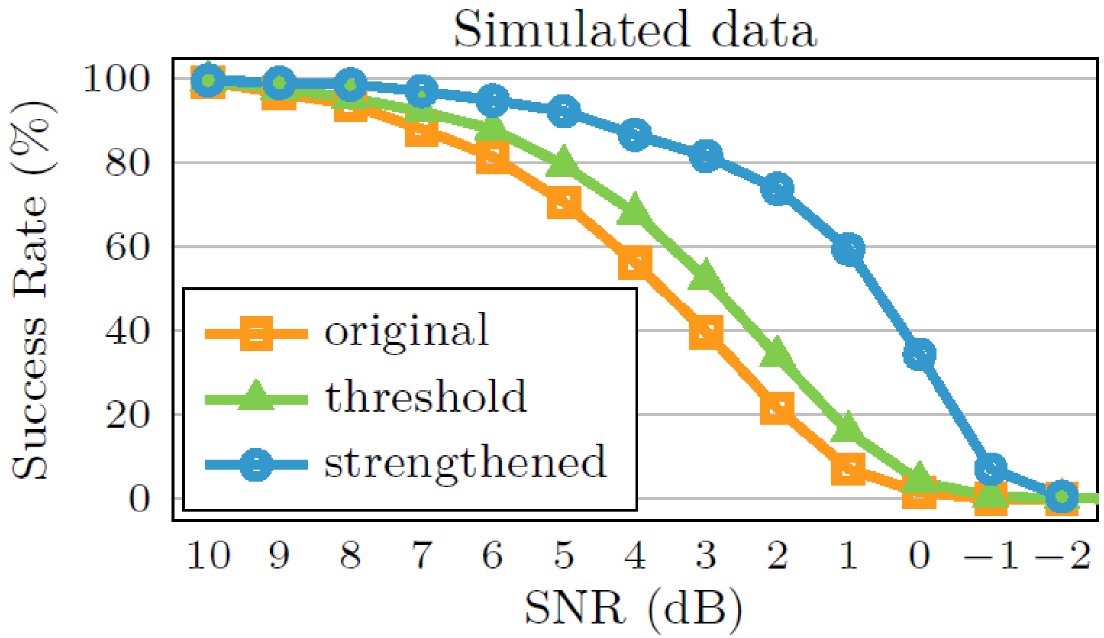
Input: target algorithm, ciphertext c, side channel $\{\mathbf{L}_k\}_{k=1}^{K}$, distinguisher $\mathcal{D}$, change point $\hat{\tau}$, $\mathbf{W}_{\hat{\tau}}$ with size $w = 2u + 1$, distinguisher sequence $\mathbf{D}$, predicted output $\widehat{O}_{\hat{\tau}-u}$

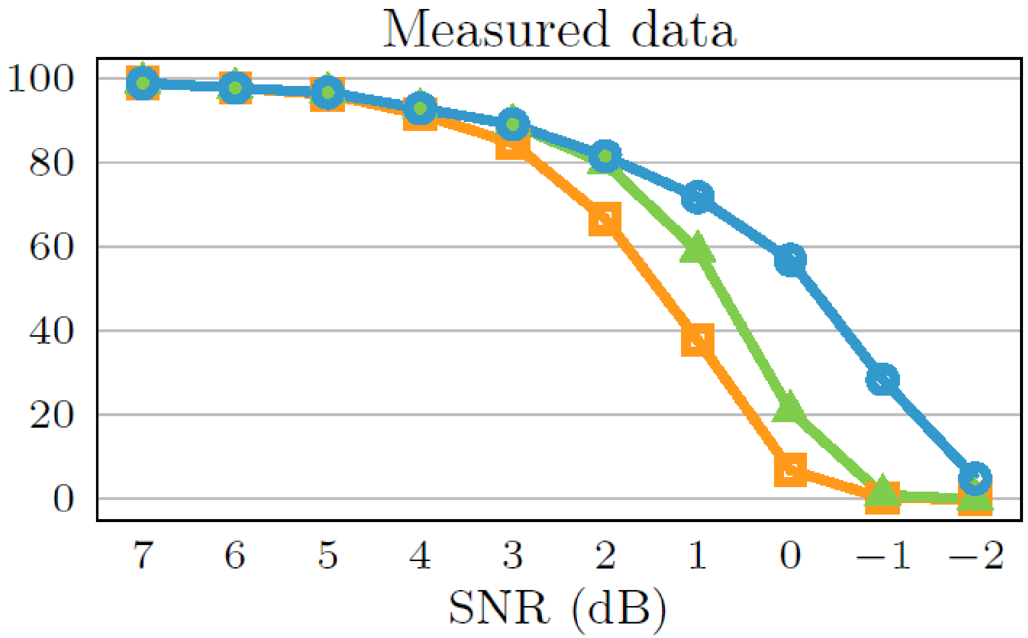
Output: correction goodness (succ_correction), best estimated key $\mathbf{x}_{\text{best}}$ over $\mathbf{W}_{\hat{\tau}}$

1: **for** $\mathbf{x} \in \mathbf{X}^w$ **do**
2: set $(\hat{d}^{\mathbf{x}}[\hat{\tau} - u], \dots, \hat{d}^{\mathbf{x}}[\hat{\tau} + u]) = \mathbf{x}$ // initialization
3: compute $\widehat{O}^{\mathbf{x}}_{\hat{\tau}-u+1}, \dots, \widehat{O}^{\mathbf{x}}_{\hat{\tau}+u+1}$ using `operations` // as in Algorithm 2, line 4
4: restart the attack from step $k = \hat{\tau} + u + 1$
5: select the two windows $\mathbf{W}_{<} \leftarrow \{D_k\}_{k<\hat{\tau}-u}$, $\mathbf{W}_{>} \leftarrow \{D^{\mathbf{x}}_k\}_{k>\hat{\tau}+u}$
6: run the statistical test $\mathcal{S}(\mathbf{W}_{<}, \mathbf{W}_{>})$
7: **if** the test yields enough statistical evidence **then**
8: **return** true, $\mathbf{x}$
9: **end if**
10: **end for**
11: **return** false, the $\mathbf{x}$ maximizing the statistic in line 6

Luca Frittoli, Matteo Bocchi, Silvia Mella, Diego Carrera, Beatrice Rossi, Pasqualina Fragneto, Ruggero Susella and Giacomo Boracchi "Strengthening Sequential Side-Channel Attacks Through Change Detection", Accepted on TCHES 2020

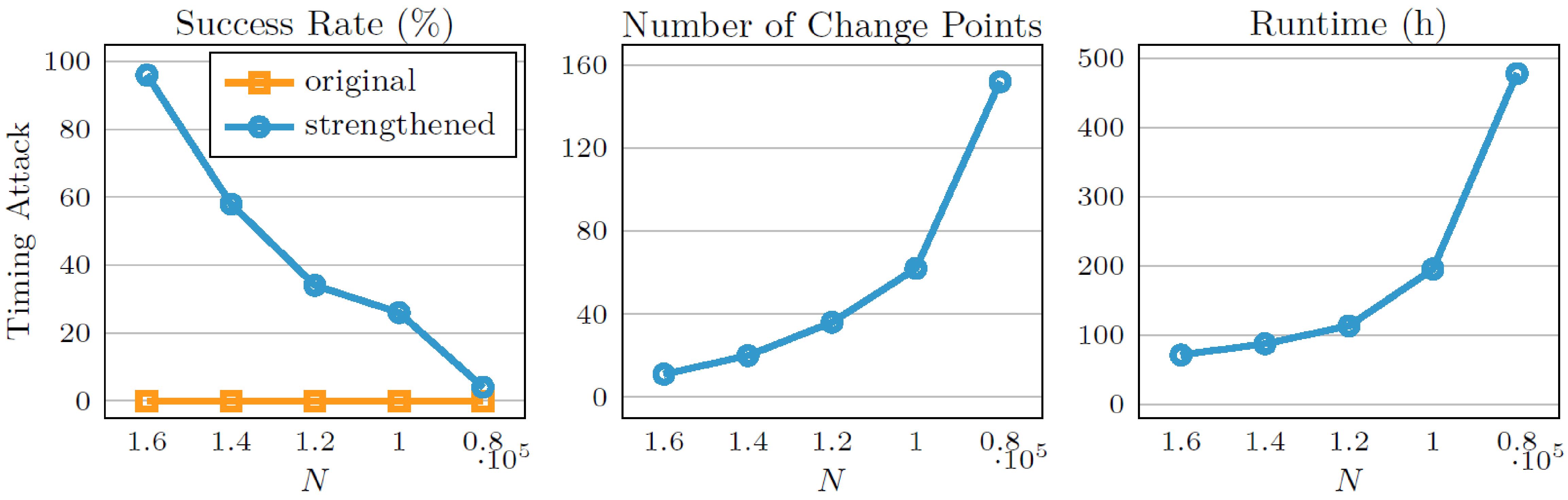
Strenghtened Sequential DPA Attack





Luca Frittoli, Matteo Bocchi, Silvia Mella, Diego Carrera, Beatrice Rossi, Pasqualina Fragneto, Ruggero Susella and Giacomo Boracchi "Strengthening Sequential Side-Channel Attacks Through Change Detection", Accepted on TCHES 2020

Strenghtened Sequential Timing Attack



Luca Frittoli, Matteo Bocchi, Silvia Mella, Diego Carrera, Beatrice Rossi, Pasqualina Fragneto, Ruggero Susella and Giacomo Boracchi "Strengthening Sequential Side-Channel Attacks Through Change Detection", Accepted on TCHES 2020

Sparse Representations For Online Monitoring

Diego Carrera, Marco Longoni, Beatrice Rossi,
Pasqualina Fragneto, Giacomo Boracchi

Data-Driven Models and Online Monitoring

Data-driven models are ubiquitous in monitoring problems

Online ECG monitoring

Collaboration with

Bora life.augmented

Data-Driven Models and Online Monitoring

Data-driven models are ubiquitous in monitoring problems

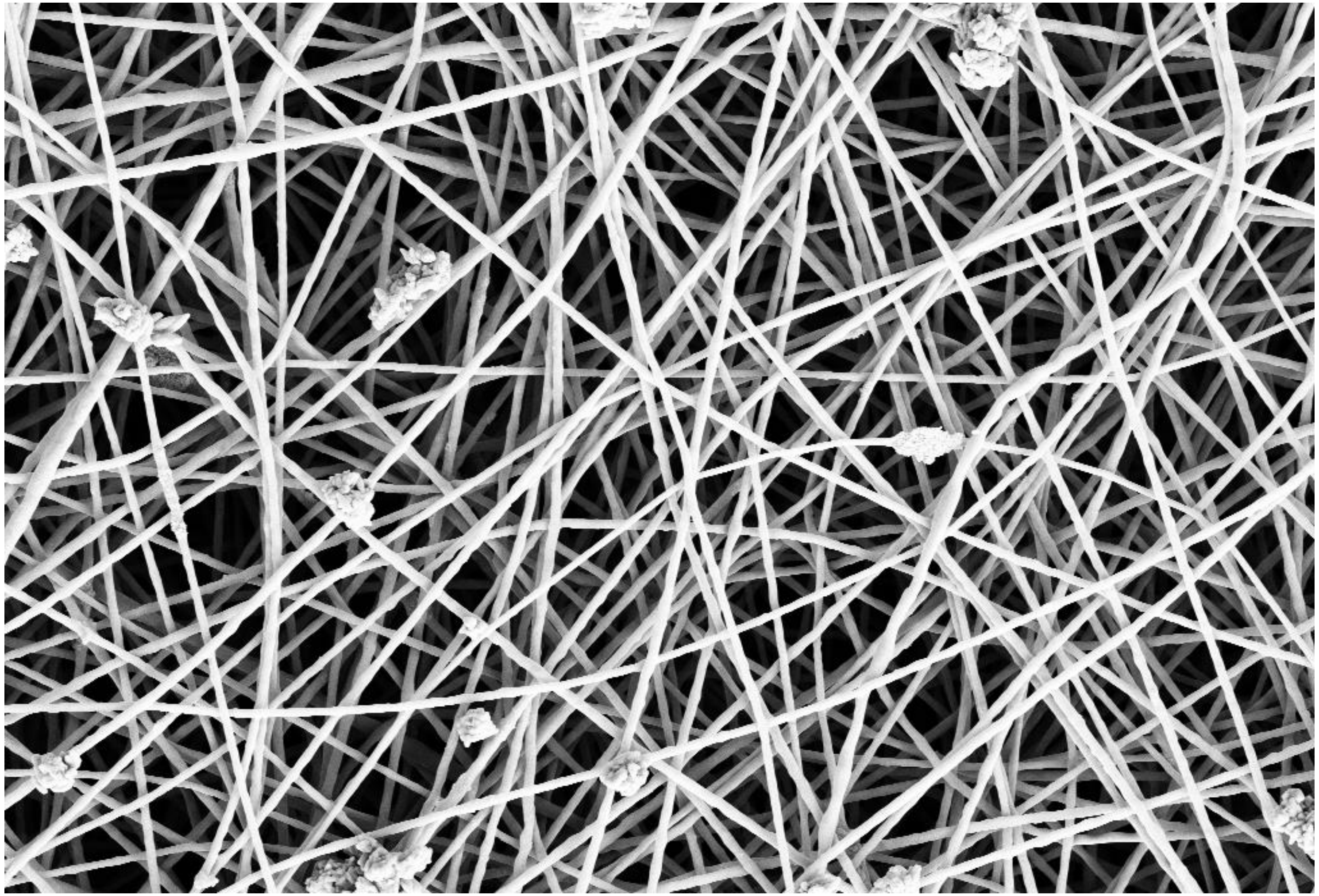
Automatic detection of anomalous heartbeats

Collaboration with

Bora life.augmented

Data-Driven Models and Online Monitoring

Data-driven models are ubiquitous in monitoring problems

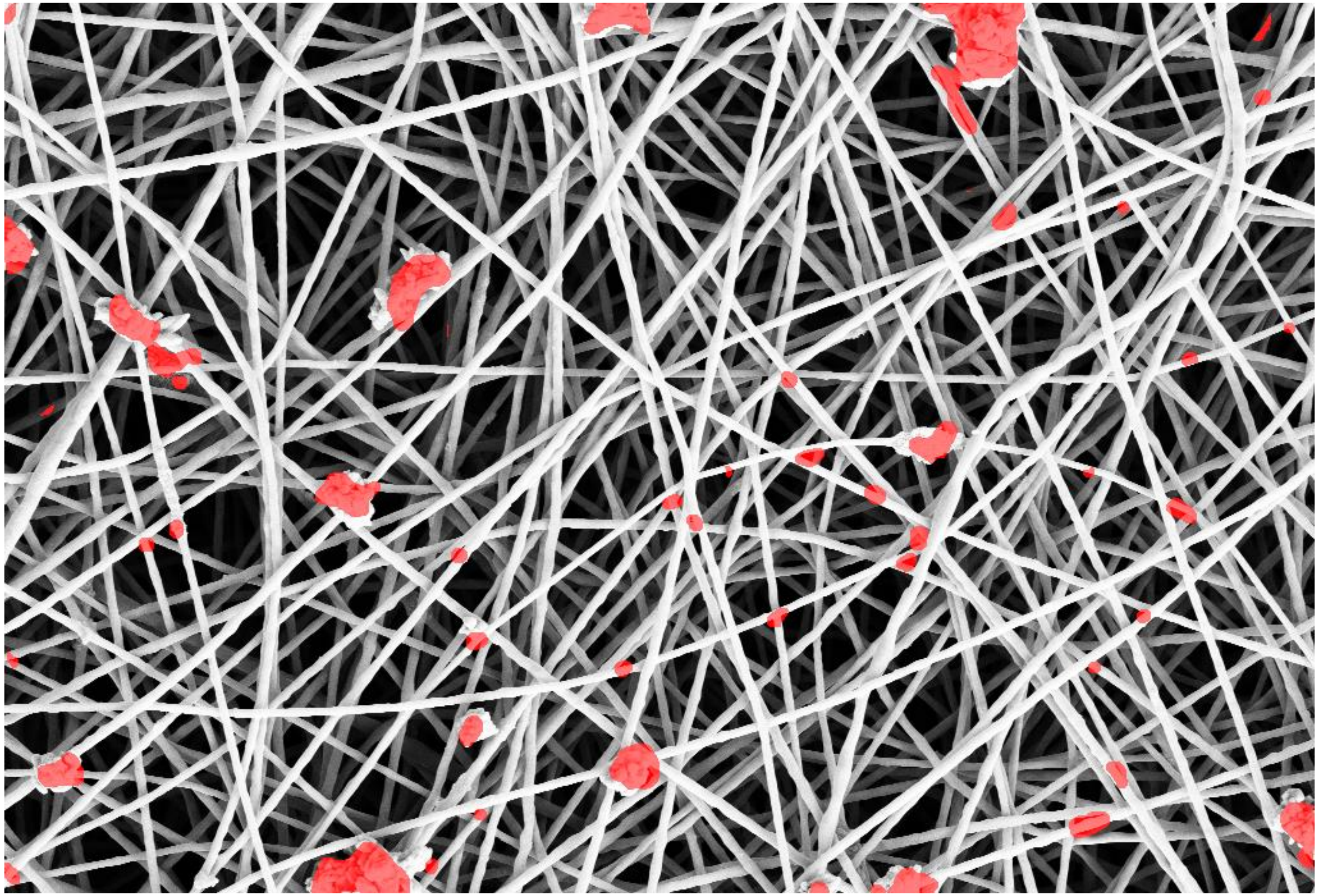


Quality inspection of nanofibers through SEM

Collaboration with

Data-Driven Models and Online Monitoring

Data-driven models are ubiquitous in monitoring problems



Automatically measure defect area

Collaboration with

imati

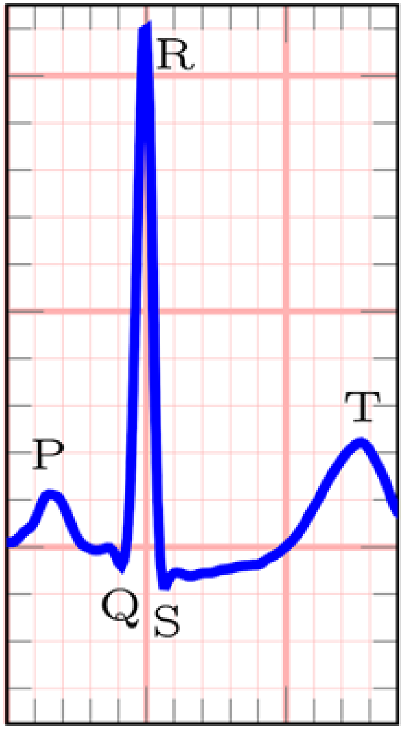
Data-Driven Models and Online Monitoring: The Addressed Challenges

Data-Driven Models and Online Monitoring

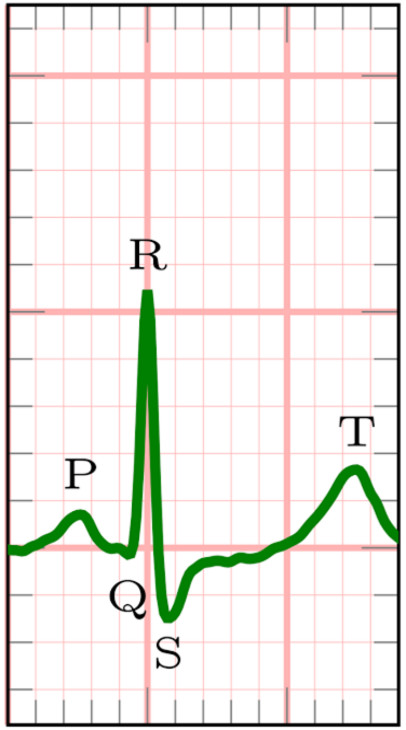
Addressed Challenges

- No / not enough supervised samples: unsupervised learning

User 1



User 2

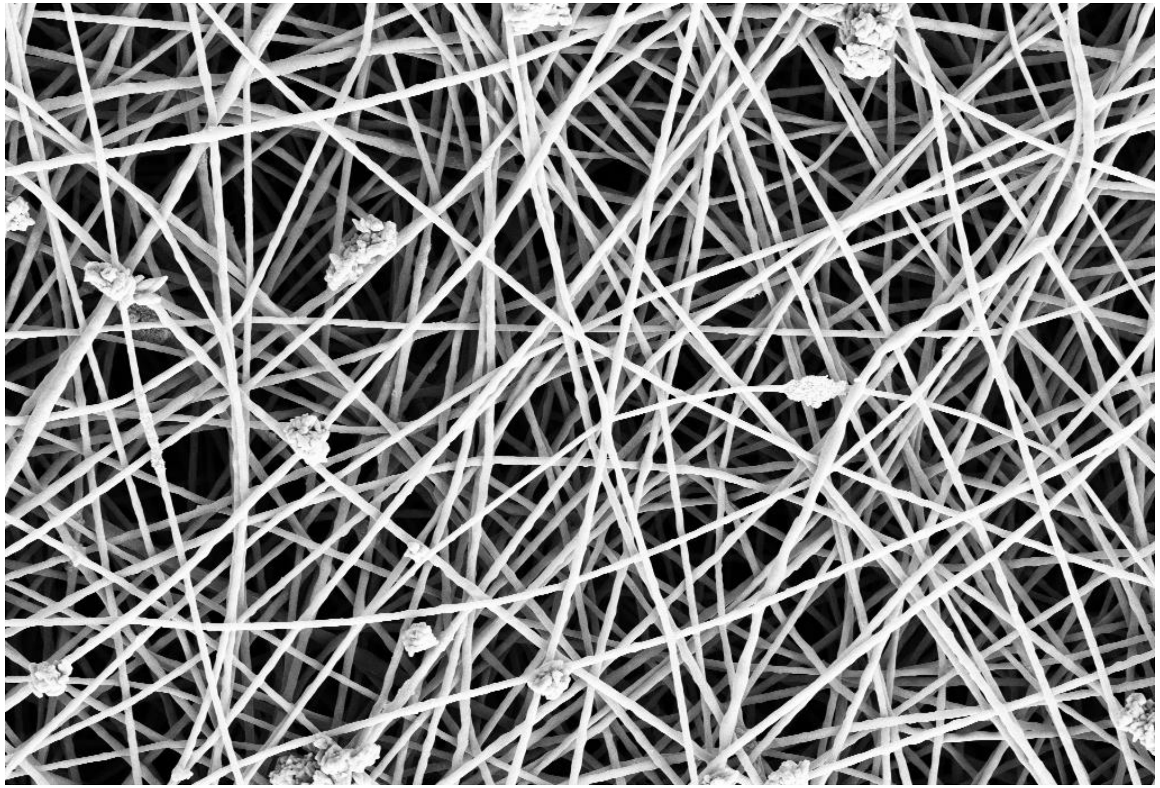


Different users feature different heartbeat morphology

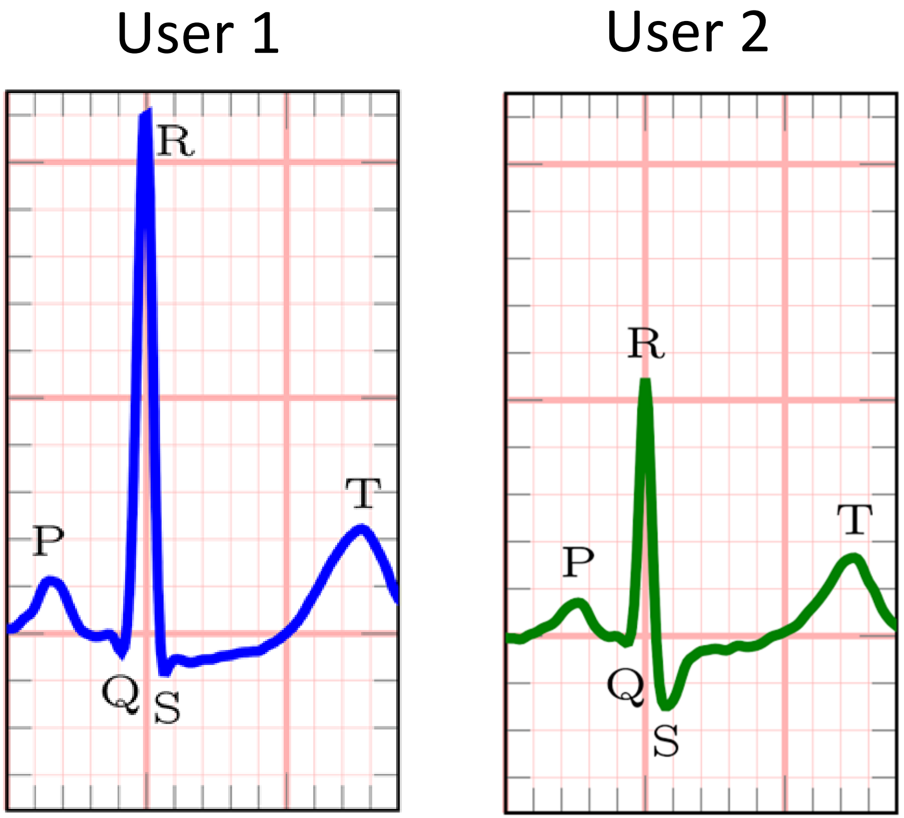
Data-Driven Models and Online Monitoring

Addressed Challenges

- No / not enough supervised samples: unsupervised learning



Better not to assume any specific defect shape

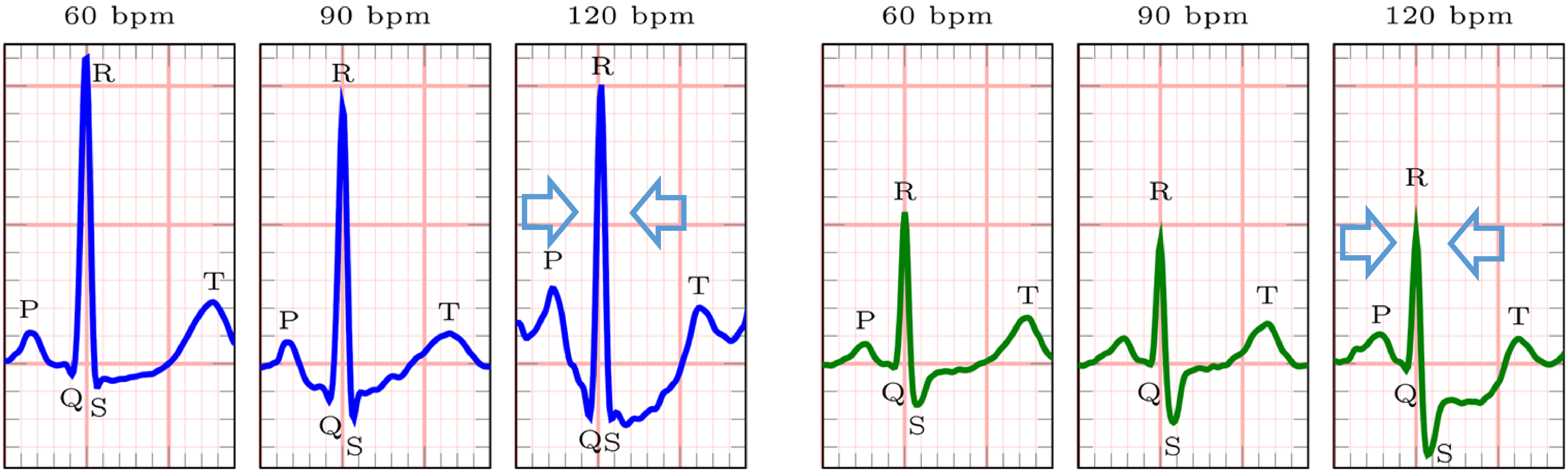


Different users feature different heartbeat morphology

Data-Driven Models and Online Monitoring

Addressed Challenges

- No / not enough supervised samples: unsupervised learning
- Test data might differ from training data: need of adaptation



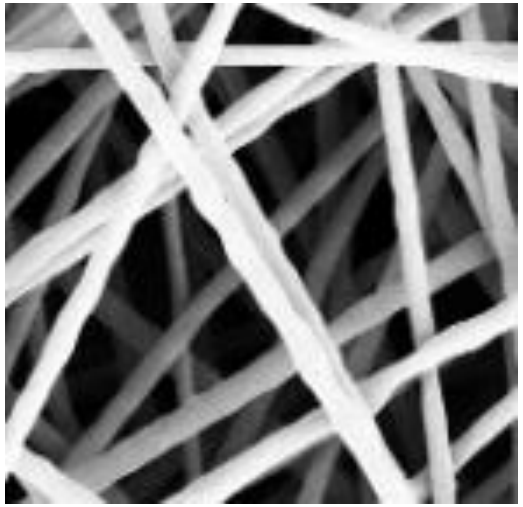
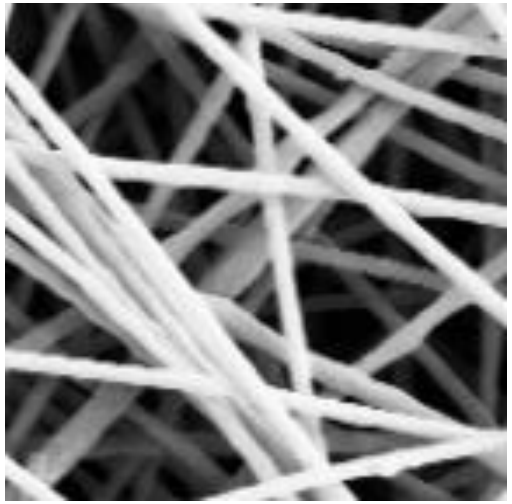
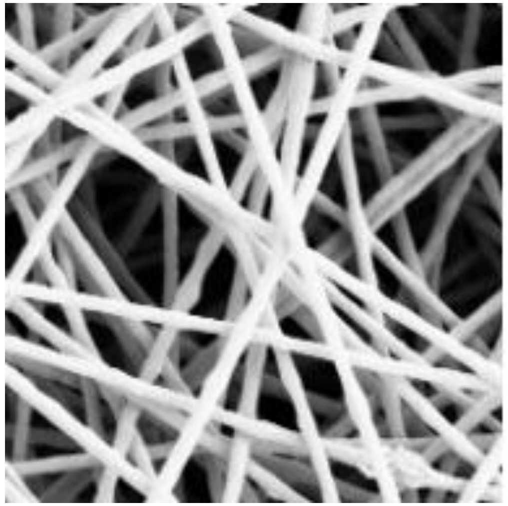
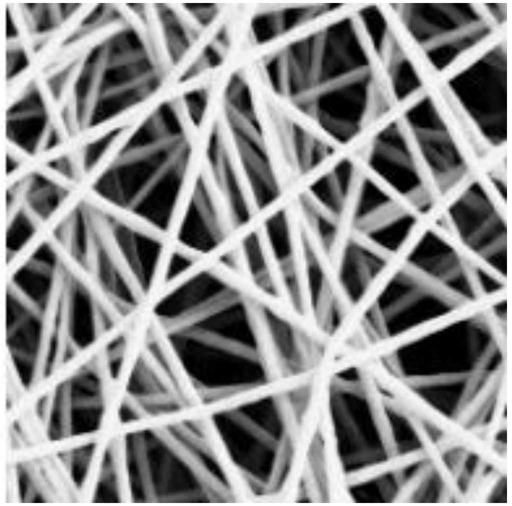
The heartbeat morphology changes when the heart rate increases

Boracchi, Trovò

Data-Driven Models and Online Monitoring

Addressed Challenges

- No / not enough supervised samples: unsupervised learning
- Test data might differ from training data: need of adaptation

Defects have to be detected at different zooming levels

Dictionaries Yielding Sparse Representations

Dictionaries Yielding Sparse Representations

A viable solution for online monitoring

- Unsupervised models
- Easy to plug in a **change/anomaly detection** framework
- Easy to **adapt**
- Simple and interpretable models

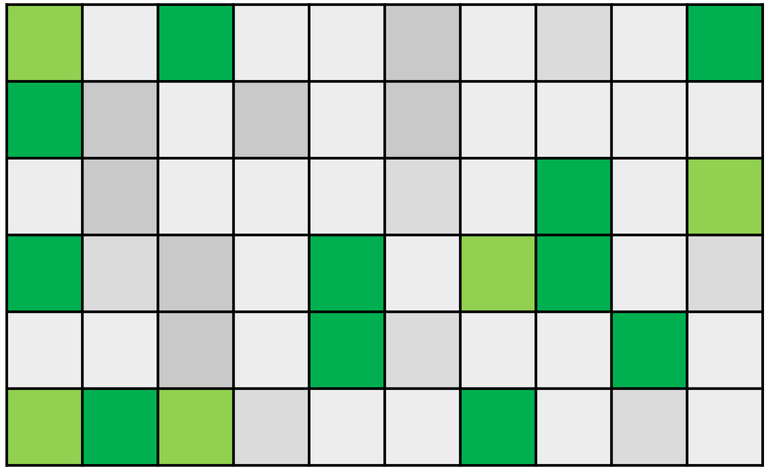
Dictionaries Yielding Sparse Representations

A viable solution for online monitoring

- Unsupervised models
- Easy to plug in a **change/anomaly detection** framework
- Easy to **adapt**
- Simple and interpretable models

Dictionaries are just matrices! $D \in \mathbb{R}^{n \times m}$

$D =$



Dictionaries Yielding Sparse Representations

A viable solution for online monitoring

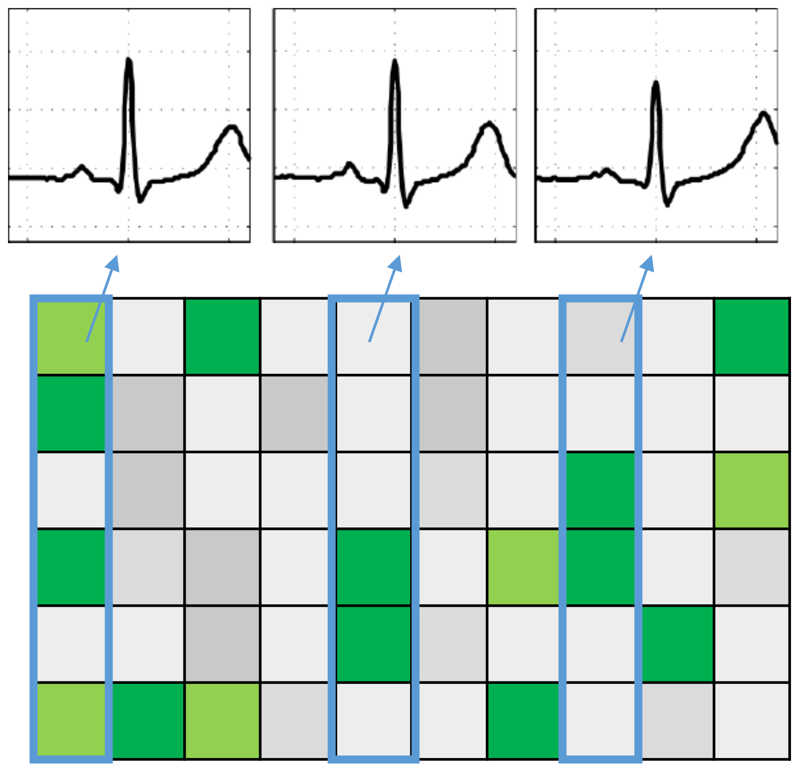
- Unsupervised models
- Easy to plug in a **change/anomaly detection** framework
- Easy to **adapt**
- Simple and interpretable models

Dictionaries are just matrices! $D \in \mathbb{R}^{n \times m}$

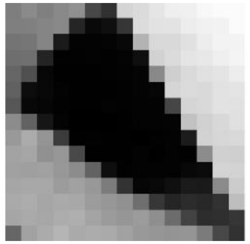
Each column is called *an atom:*

- lives in the input space
- it is one of the learned building blocks to reconstruct the input signal

$D =$



Dictionaries Yielding Sparse Representations



$\boldsymbol{s}$

A viable solution for online monitoring

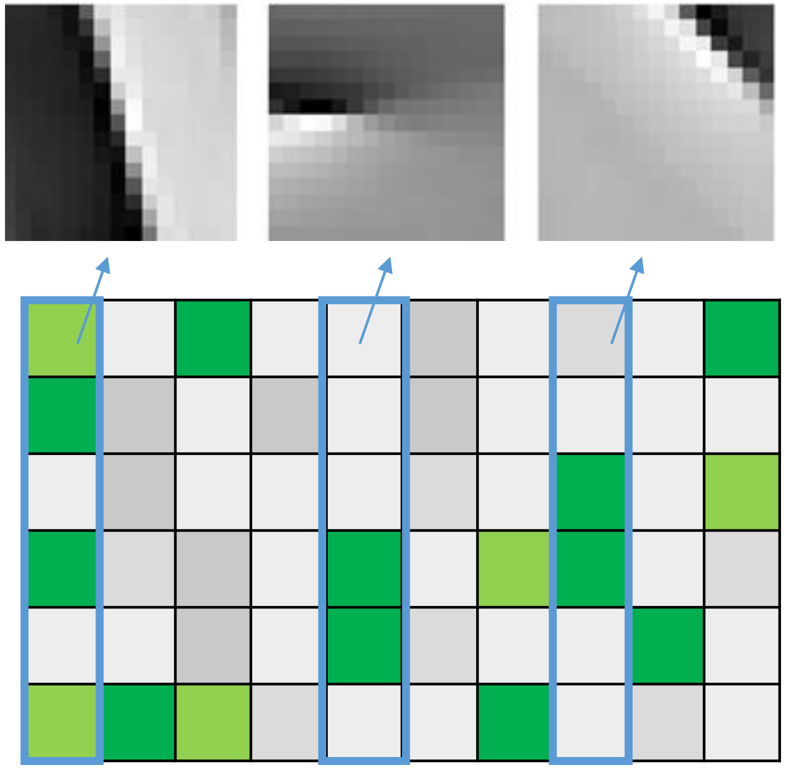
- Unsupervised models
- Easy to plug in a **change/anomaly detection** framework
- Easy to **adapt**
- Simple and interpretable models

Dictionaries are just matrices! $D \in \mathbb{R}^{n \times m}$

Each column is called *an atom:*

- lives in the input space
- it is one of the learned building blocks to reconstruct the input signal

$D =$

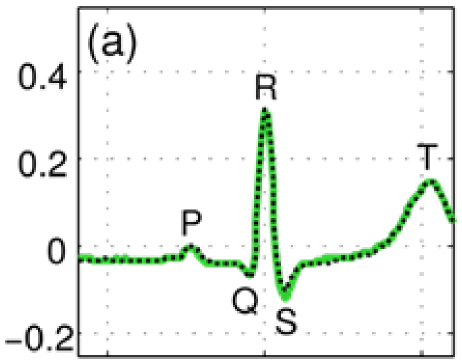


Sparse Representations

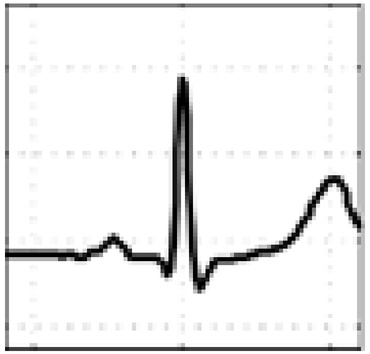
Let $\boldsymbol{s} \in \mathbb{R}^n$ be the input signal, a **sparse representation** is

$$\boldsymbol{s} = \sum_{i=1}^{M} x_i \, \boldsymbol{d_i},$$

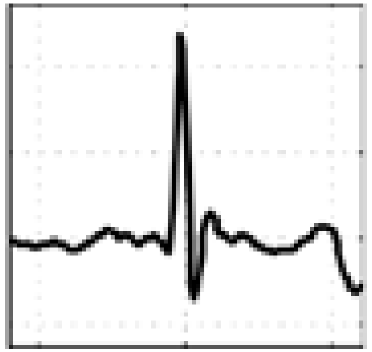
A sparse representation is a **linear combination** of **few dictionary atoms** $\{\boldsymbol{d_i}\}$



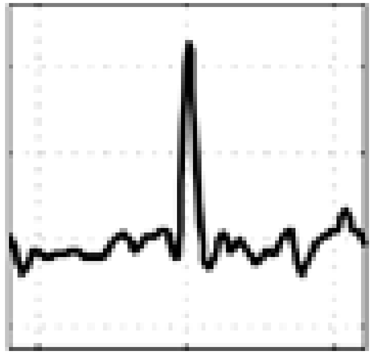
$= 0.7 \ *$



$+0.1 \ *$



$-0.2 \ *$

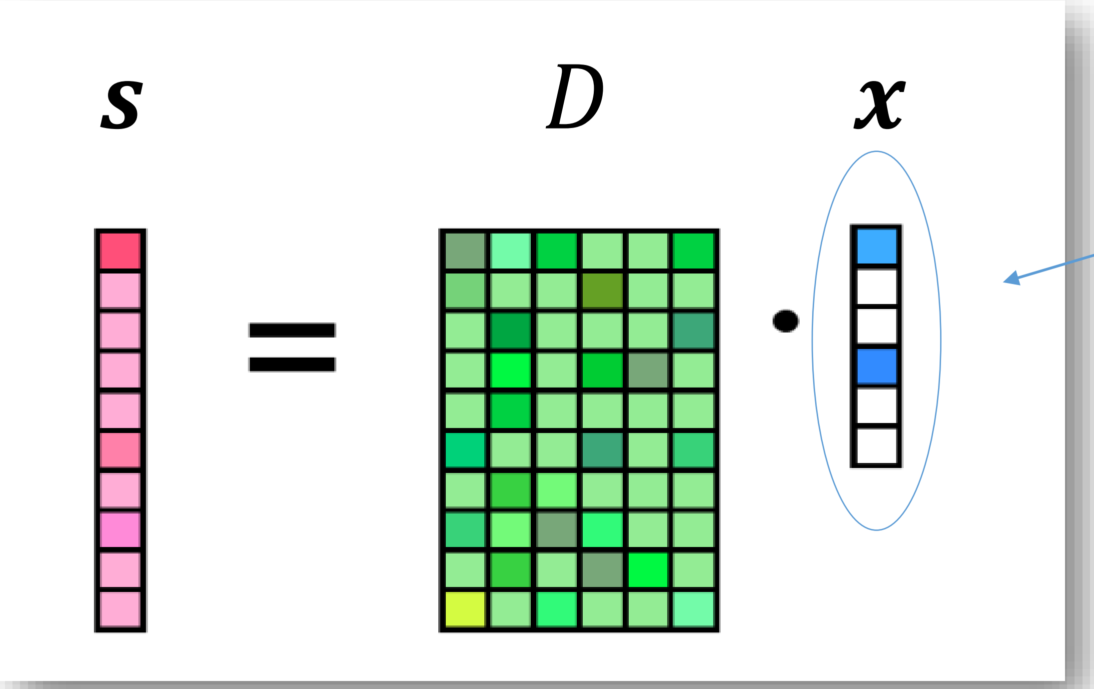


Sparse Representations

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

$$\boldsymbol{s} = \sum_{i=1}^{M} x_i \, \boldsymbol{d_i} = D\boldsymbol{x}$$

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



This vector $\boldsymbol{x} = [x_1, \quad \dots, \quad x_M]$ is **sparse**

Sparse Coding…

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

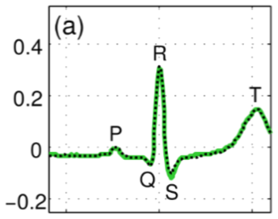
$$\boldsymbol{s} \in \mathbb{R}^n \quad \longrightarrow \quad \boldsymbol{x} \in \mathbb{R}^m$$

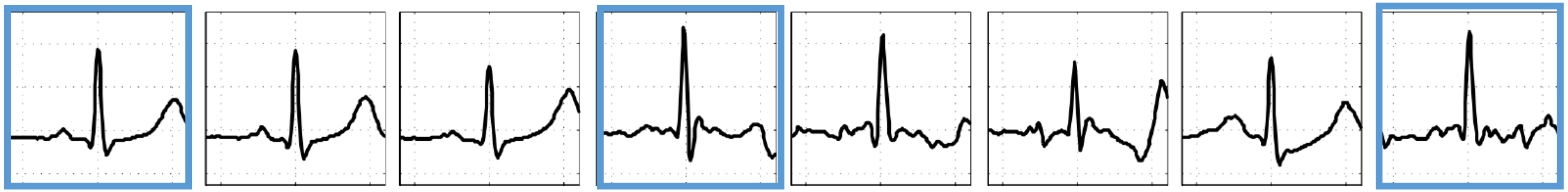
It is solved as the following optimization problem (OMP or BPDN in case of $||\boldsymbol{x}||_1$ reg.)

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

In the previous illustration $\boldsymbol{x} = [0.7, 0, 0, 0.1, 0, 0, 0, -0.2]$

$\boldsymbol{s}$





$\boldsymbol{x} =$ 0.7   0   0   0.1   0   0   0 -0.2

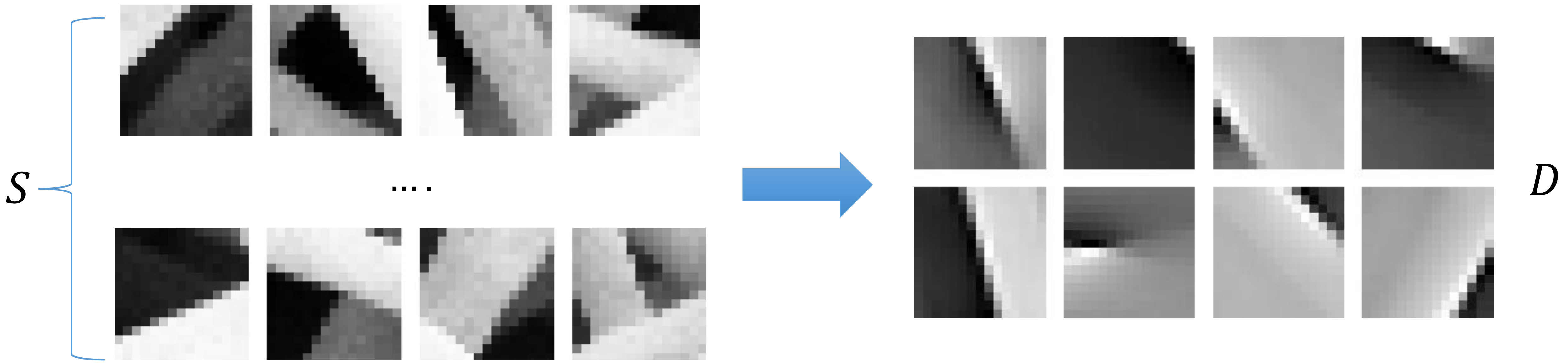
… and Dictionary Learning

Dictionary Learning: estimate D from a training set of M signals $S \in \mathbb{R}^{n \times M}$

$$S = \{\boldsymbol{s_1}, \dots \boldsymbol{s_M}\} \quad \longrightarrow \quad D \in \mathbb{R}^{n \times m}$$

It is solved as the following optimization problem (OMP or ADMM in case of $||\boldsymbol{x}||_1$)

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



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

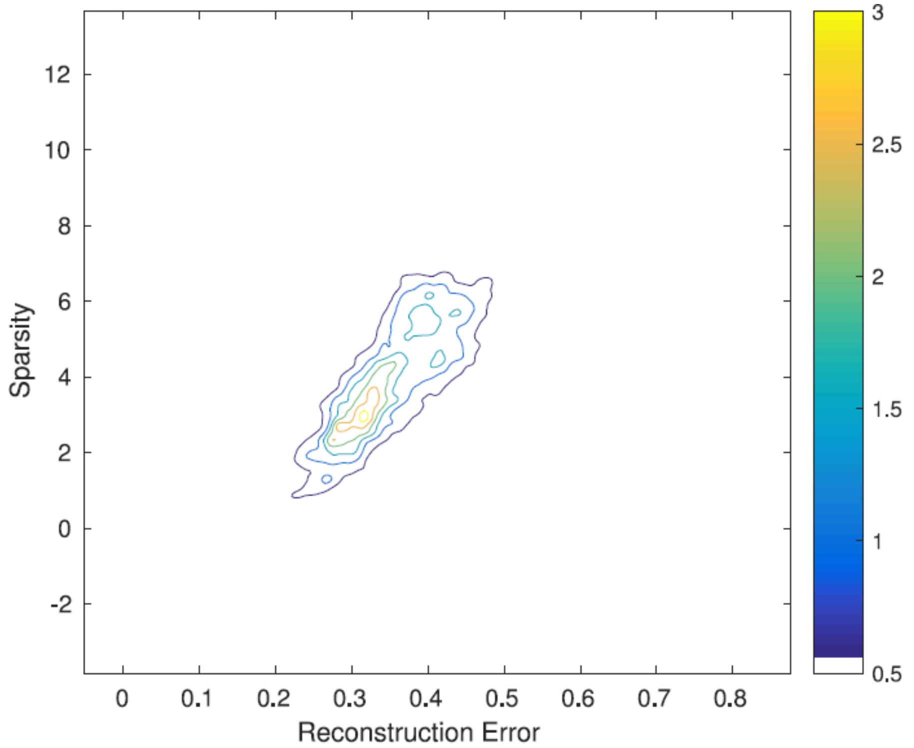
Anomaly Detection Through Dictionaries

Online Monitoring Sparse Representations

Training:

Learn a dictionary D from a training set S containing **normal instances**

Learn how normal data are reconstructed by D



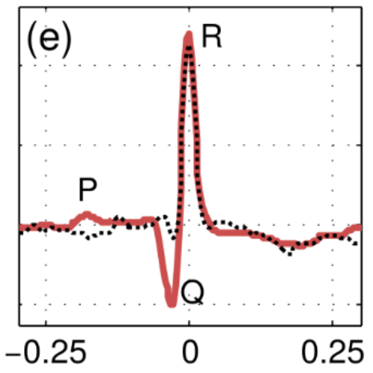
Online Monitoring Sparse Representations

Training:

Learn a dictionary D from a training set S containing **normal instances**

Learn how normal data are reconstructed by D

Anomalies



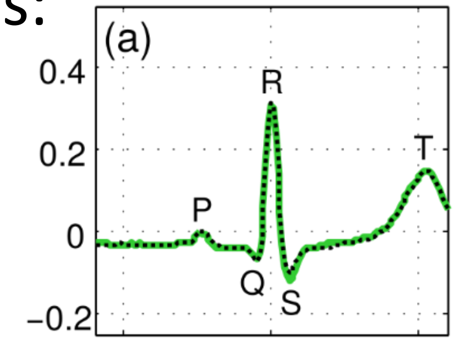
Anomaly Detection:

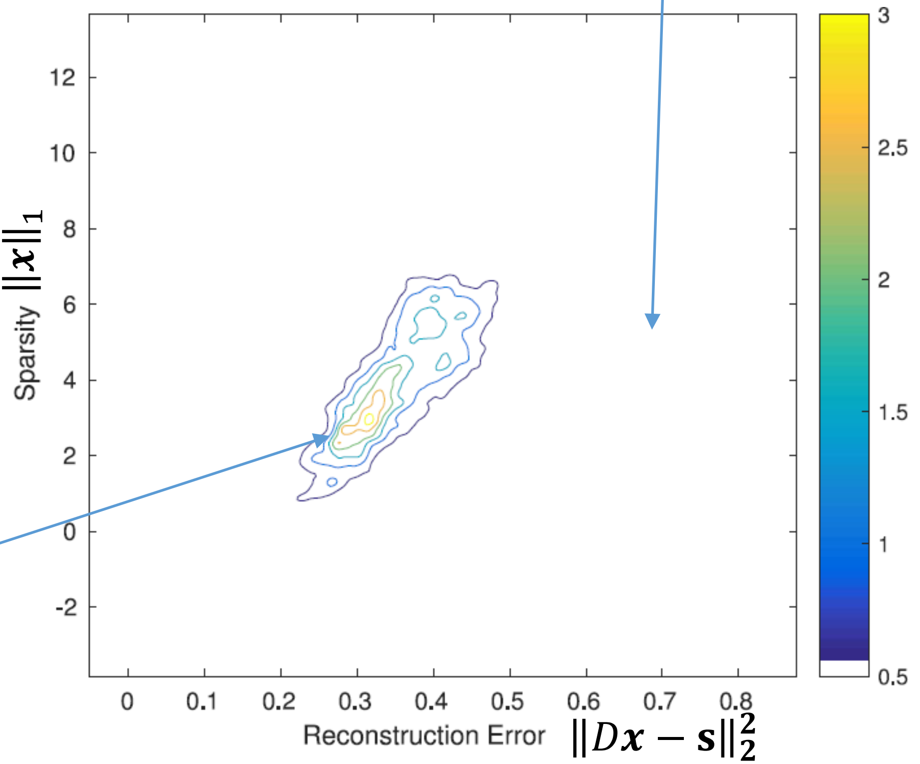
Sparse Coding: encode each test signal $\boldsymbol{s}$ w.r.t. D, and assess its conformance with D.

Check whether the representation is:

- Sparse $\|\boldsymbol{x}\|_1$
- Accurate $\|D\boldsymbol{x} - \mathbf{s}\|_2^2$

Normal data:





Online Monitoring Sparse Representations

Training:

Learn a dictionary D from a training set S containing **normal instances**

Learn how normal data are reconstructed by D

Anomaly Detection:

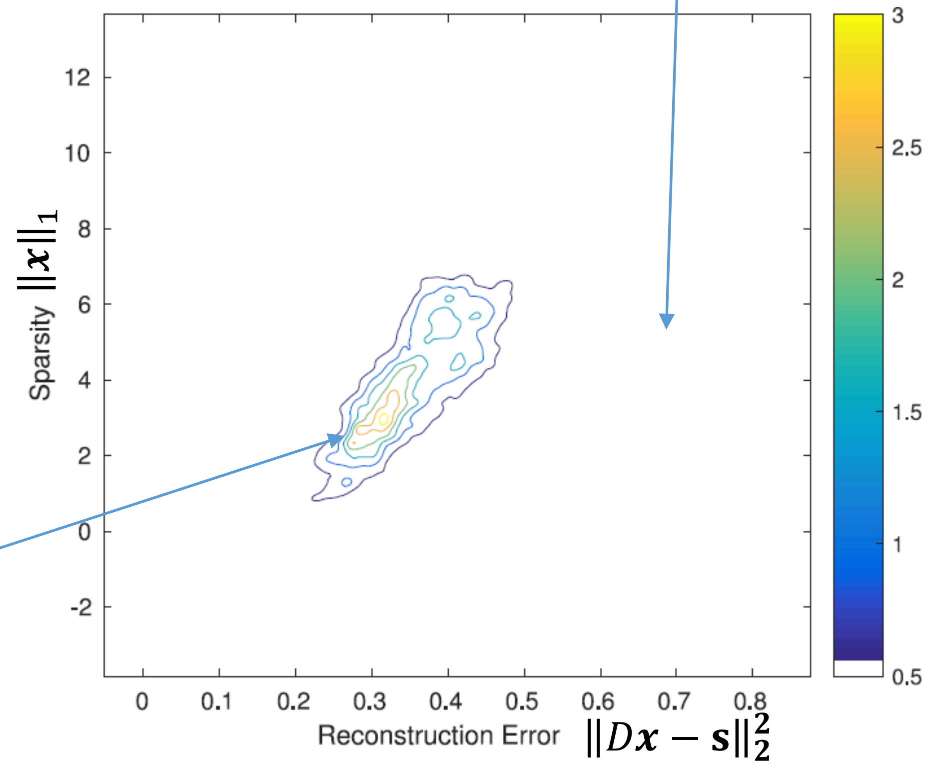
Sparse Coding: encode each test signal $\boldsymbol{s}$ w.r.t. D, and assess its conformance with D.

Check whether the representation is:

- Sparse $\|\boldsymbol{x}\|_1$
- Accurate $\|D\boldsymbol{x} - \mathbf{s}\|_2^2$

Anomalies

Normal data

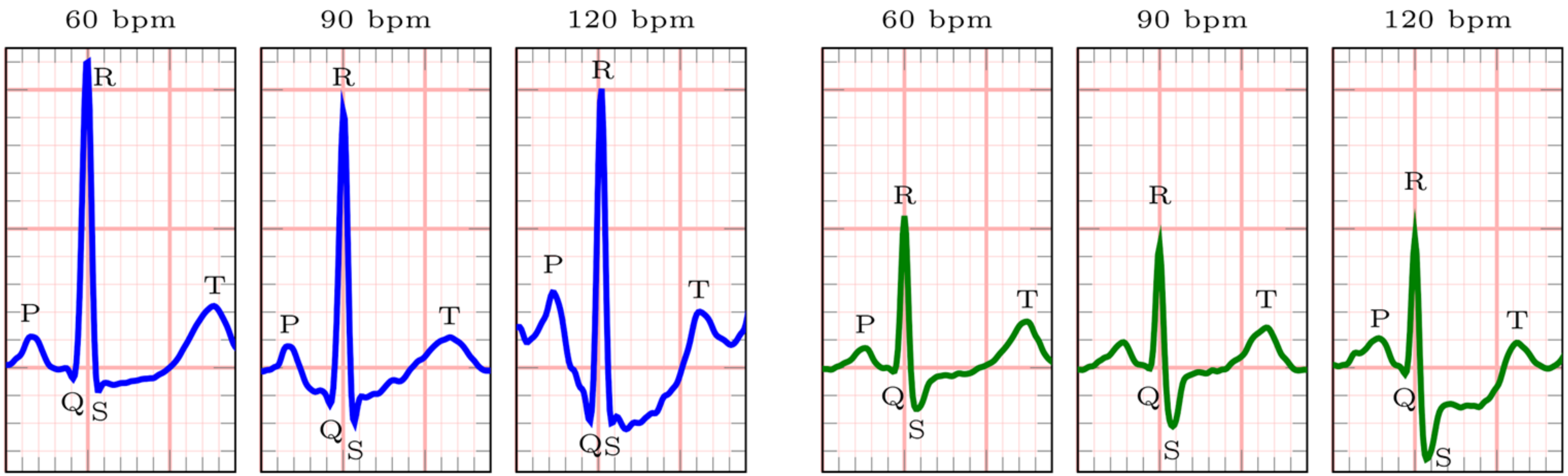


Dictionary Adaptation

Domain Adaptation for Online ECG Monitoring

The issue:

- Dictionary has to be learned from each user
- ECG tracings for training can be only acquired in resting conditions
- During daily activities heart-rate changes and do not match the learned dictionary



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

Boracchi, Trovò

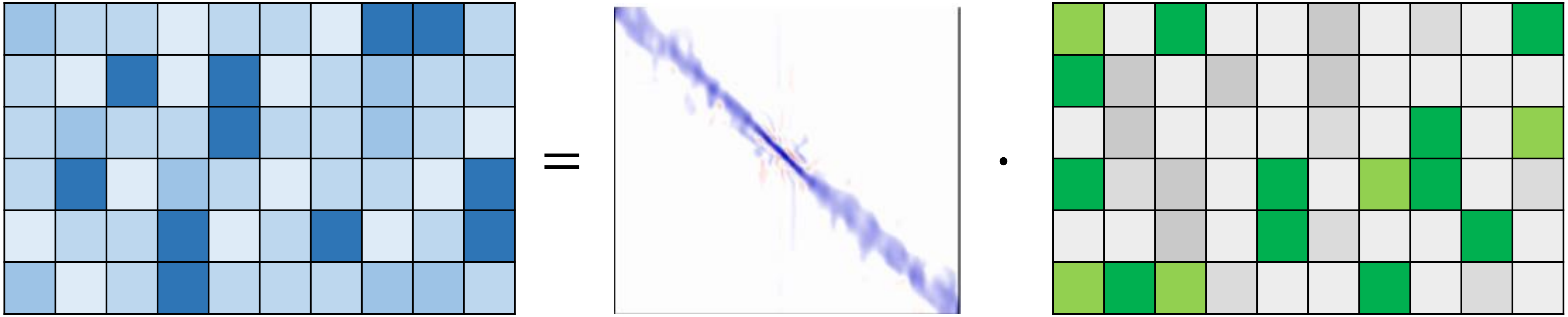
Domain Adaptation for Online ECG Monitoring

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

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



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

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

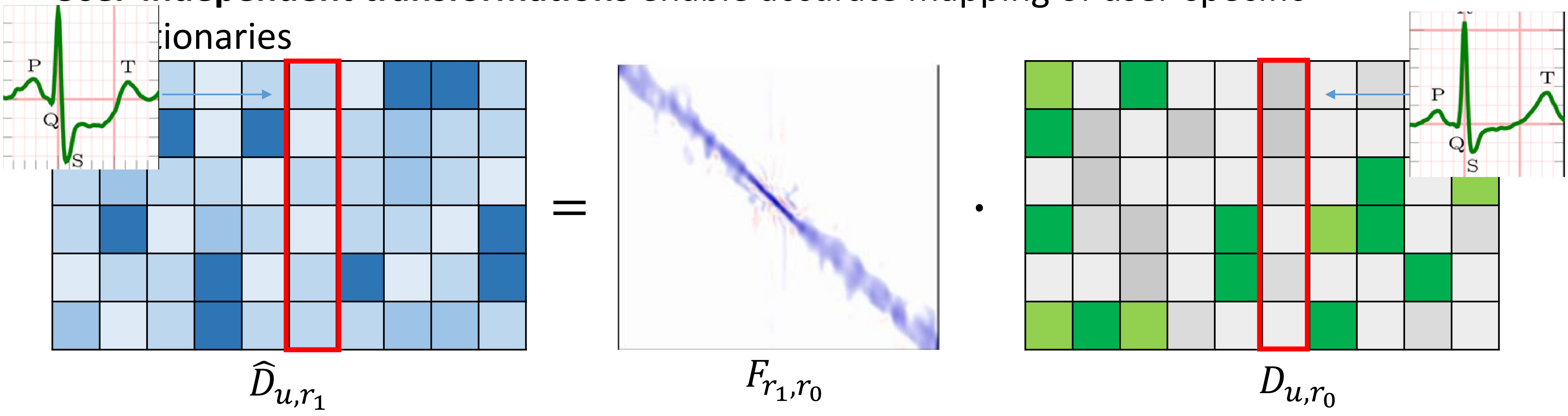
Domain Adaptation for Online ECG Monitoring

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

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



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

Domain Adaptation for Online ECG Monitoring

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

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

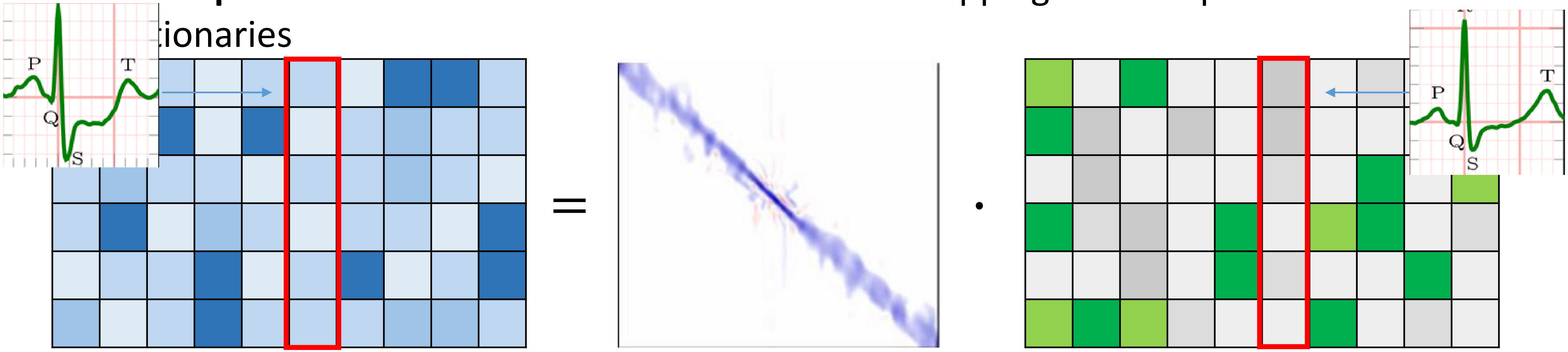
Surp

cont

A similar form of adaptation can be implemented to adapt the anomaly detection threshold

User-independent transformations enable accurate mapping of user-specific

tionaries



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

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

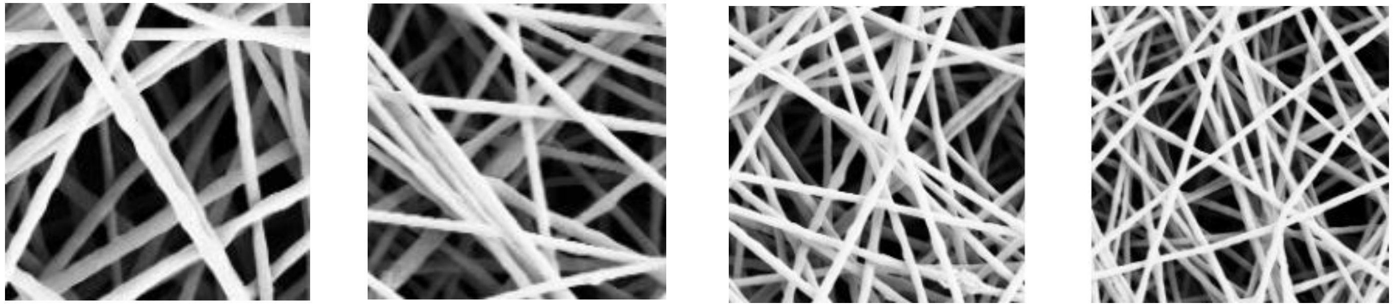
Domain Adaptation on Quality Inspection

The Issue:

- SEM images can be acquired at different zooming levels

Solution:

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



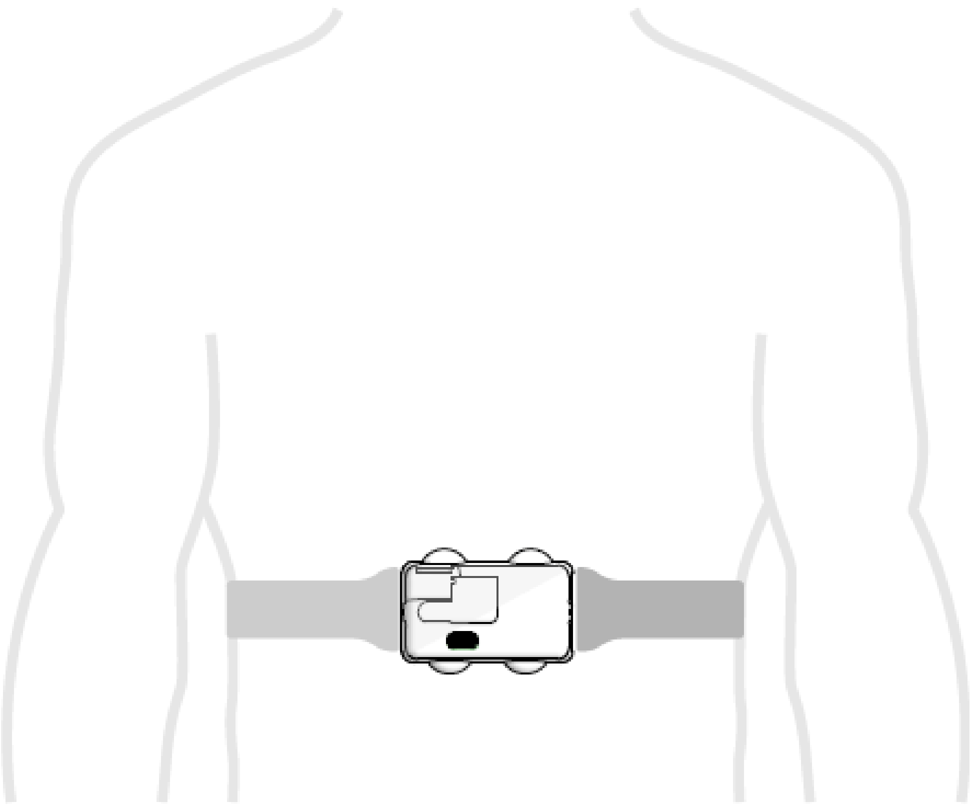
$$D = [\quad D_1 \quad D_2 \quad D_3 \quad D_4 \quad]$$

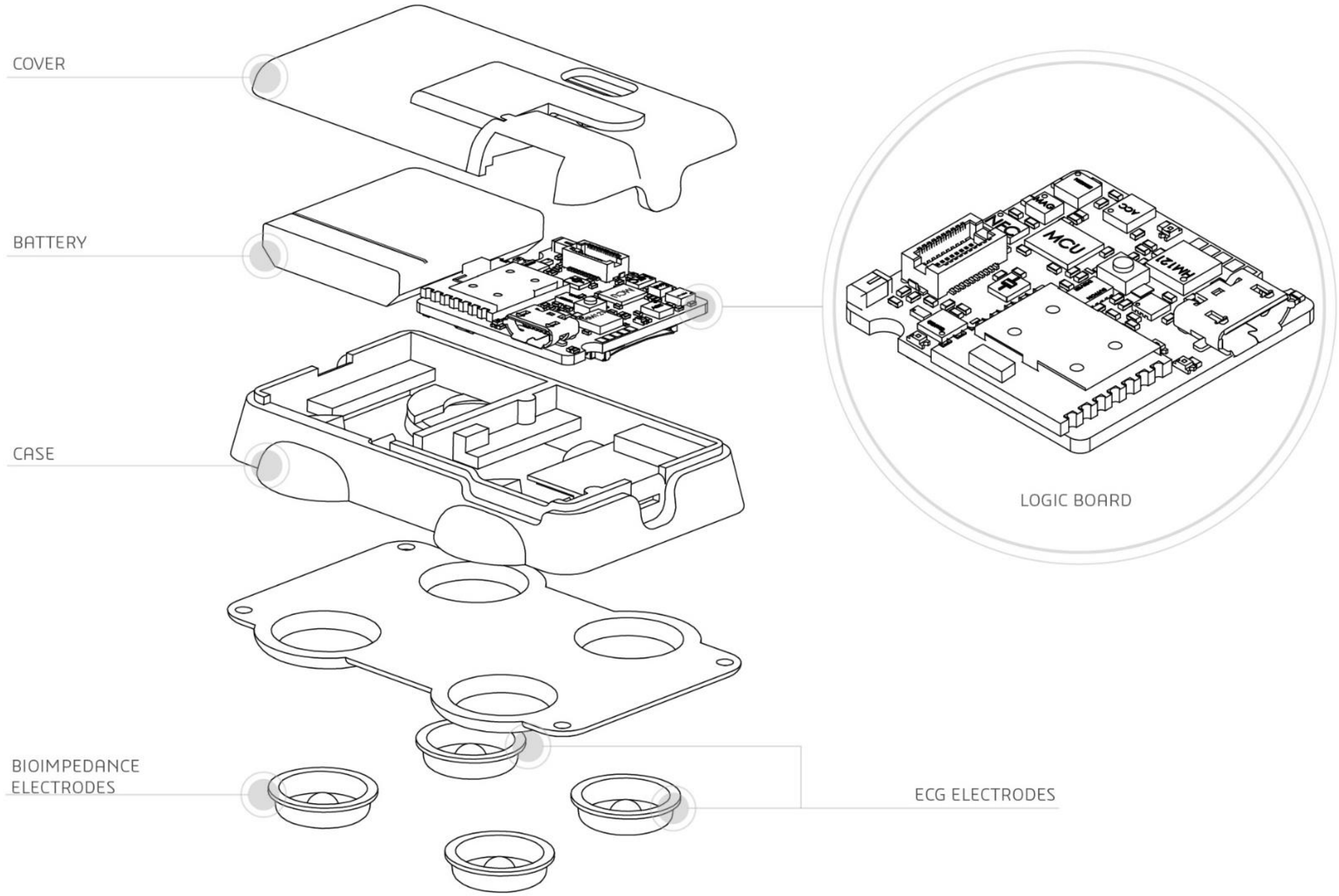
Carrera D., Boracchi G., Foi A. and Wohlberg Brendt "Scale-invariant Anomaly Detection With multiscale Group-sparse Models" IEEE ICIP 2016

Online ECG Monitoring by Wearable Devices

The BIO2BIT device

ECG signals are recorded by the BIO2BIT device

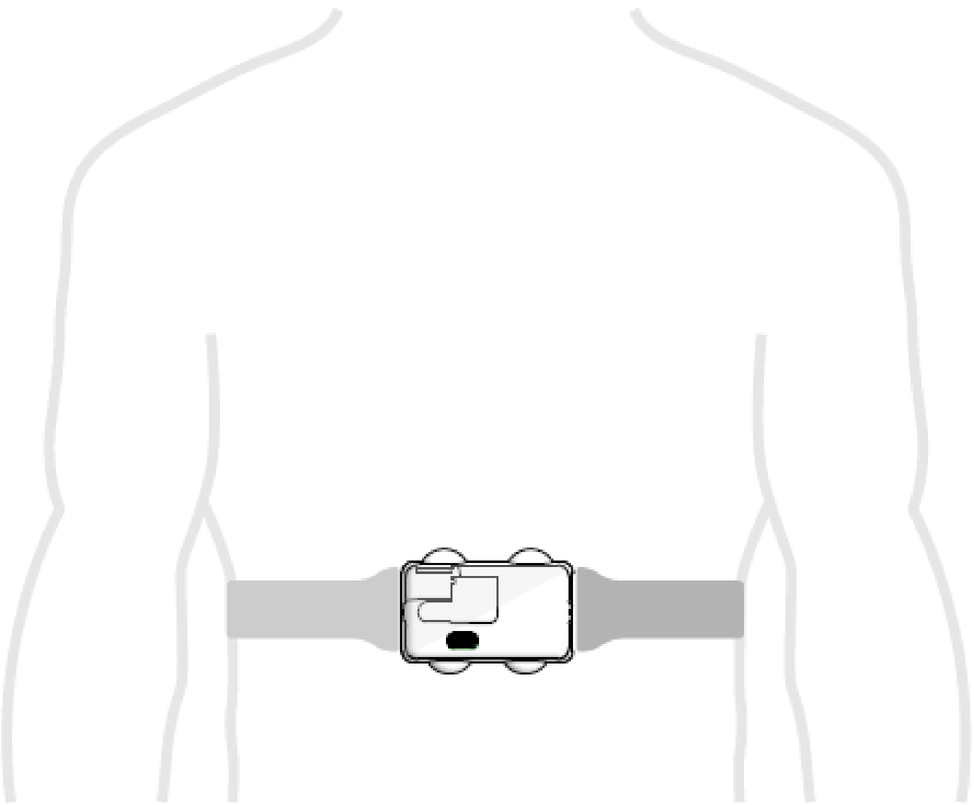


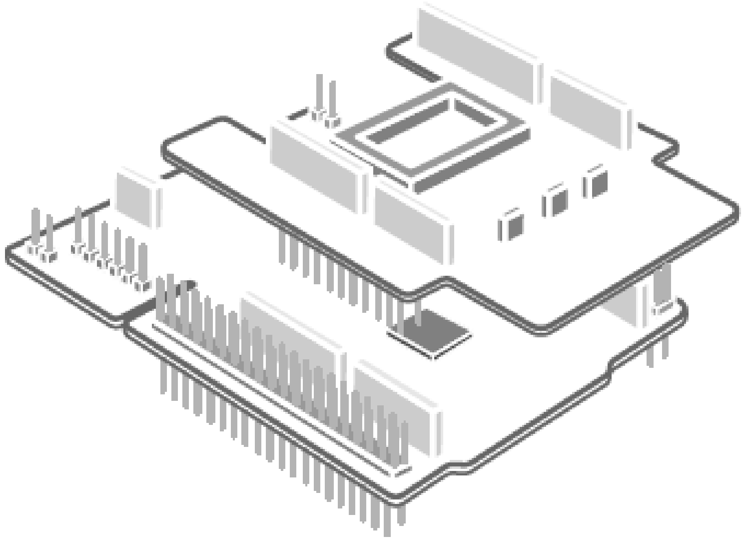


Online ECG Signals by Wearable Devices

ECG signals are recorded by the BIO2BIT device

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



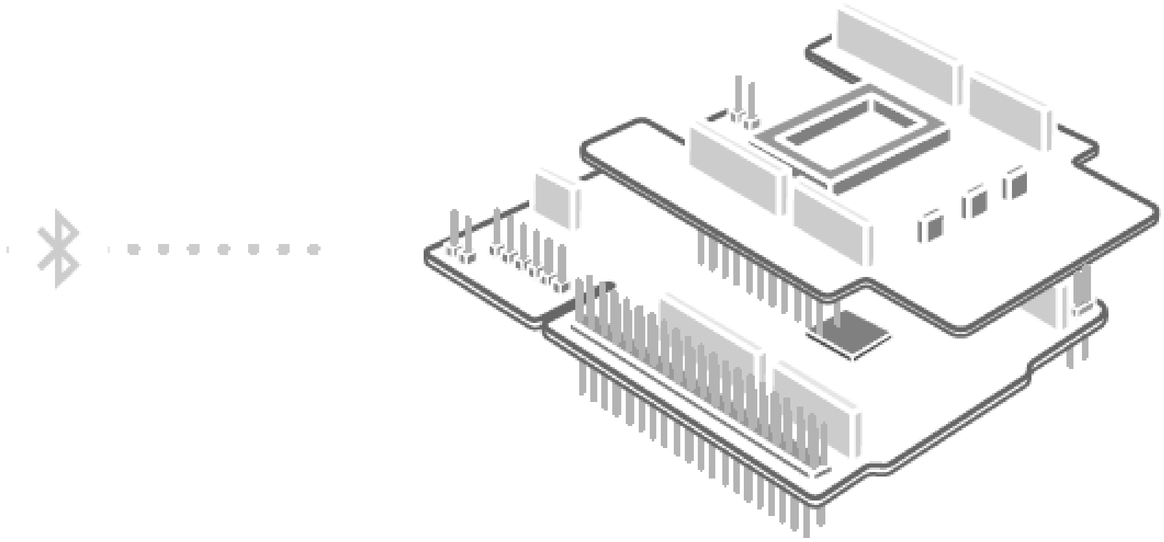


Nucleo
STM32L476RG

Online ECG Signals by Wearable Devices

Sparse Coding

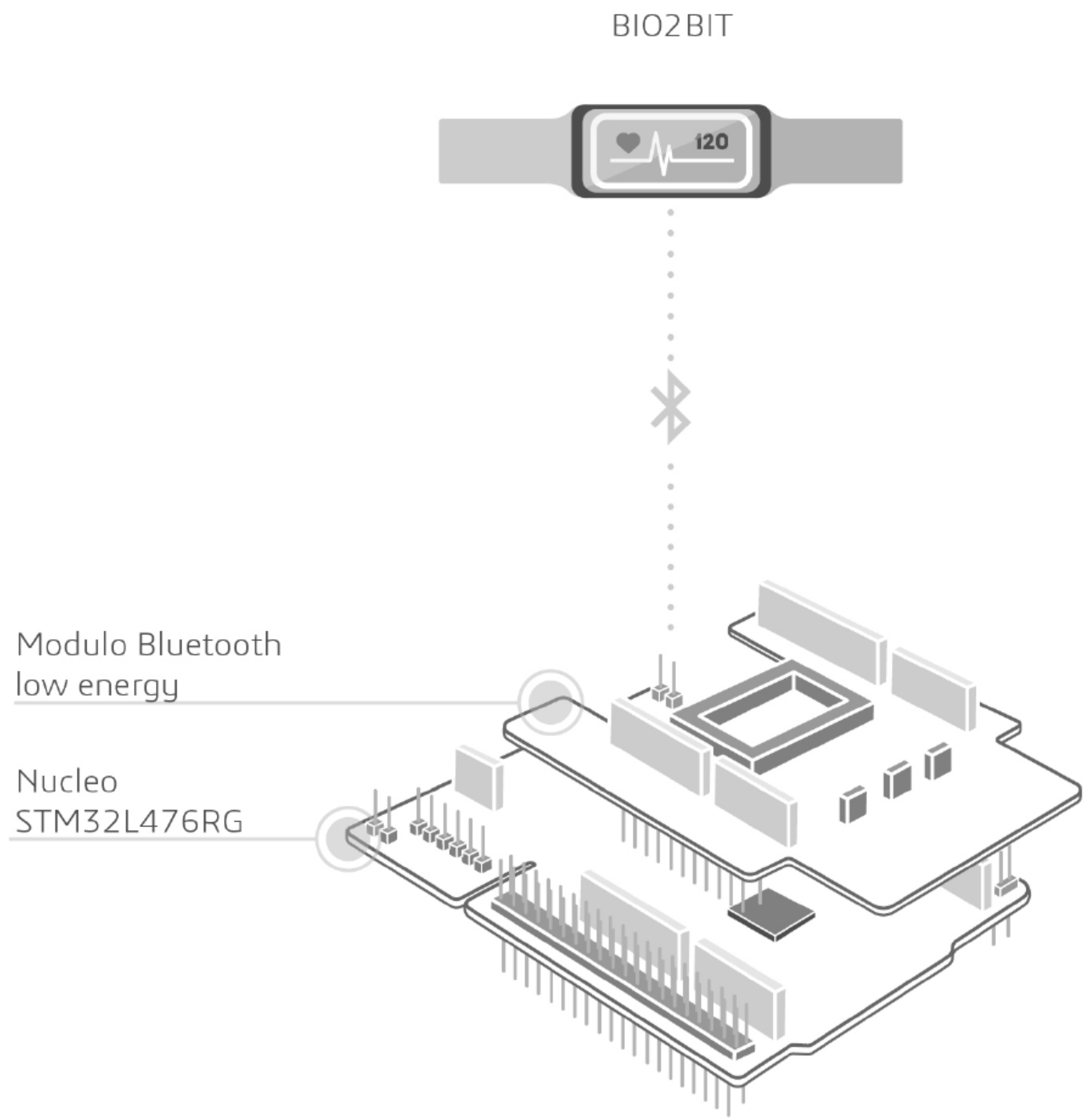
- Optimized OMP for underdetermined dictionaries
- Performed in real-time on such a low-power wearable device



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

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

Online ECG Signals by Wearable Devices



Dictionary Learning

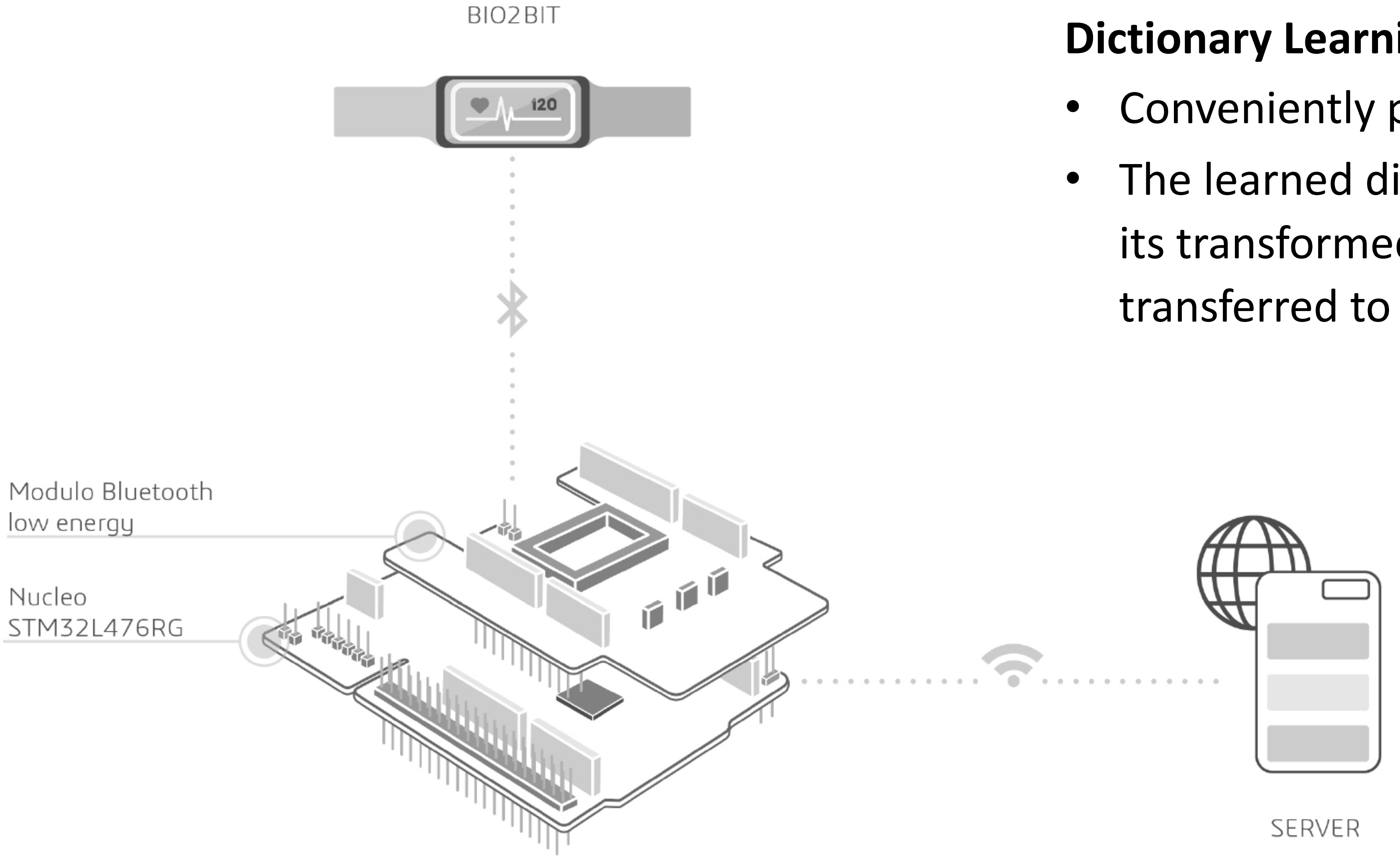
5 minutes of ECG signals are enough to learn a dictionary D_{u,r_0} that is:

- User-specific
- Position-specific

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

Carrera D., Rossi B., Zambon D., Fragneto P., and Boracchi G. "ECG Monitoring in Wearable Devices by Sparse Models", ECML-PKDD 2016
Longoni M., Carrera D., Rossi B., Fragneto P., Pessione M., Boracchi G A Wearable Device for Online and Long-Term ECG Monitoring , *International Joint Conference on Artificial Intelligence (IJCAI) 2018 - Demo Track*

Online ECG Signals by Wearable Devices



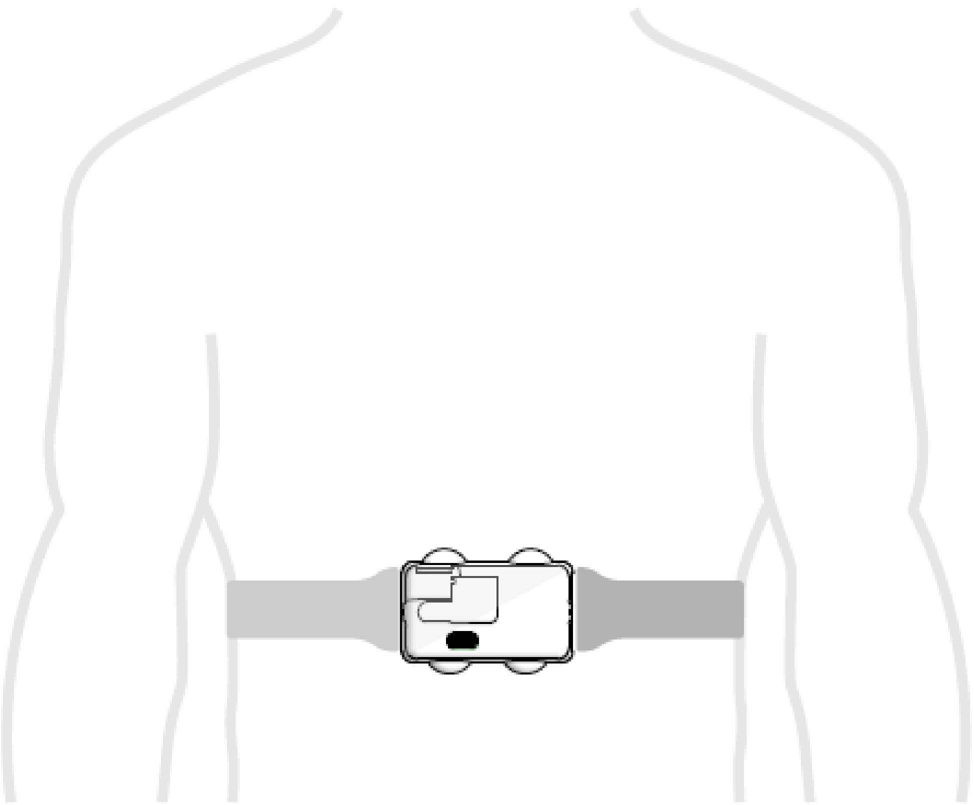
Dictionary Learning

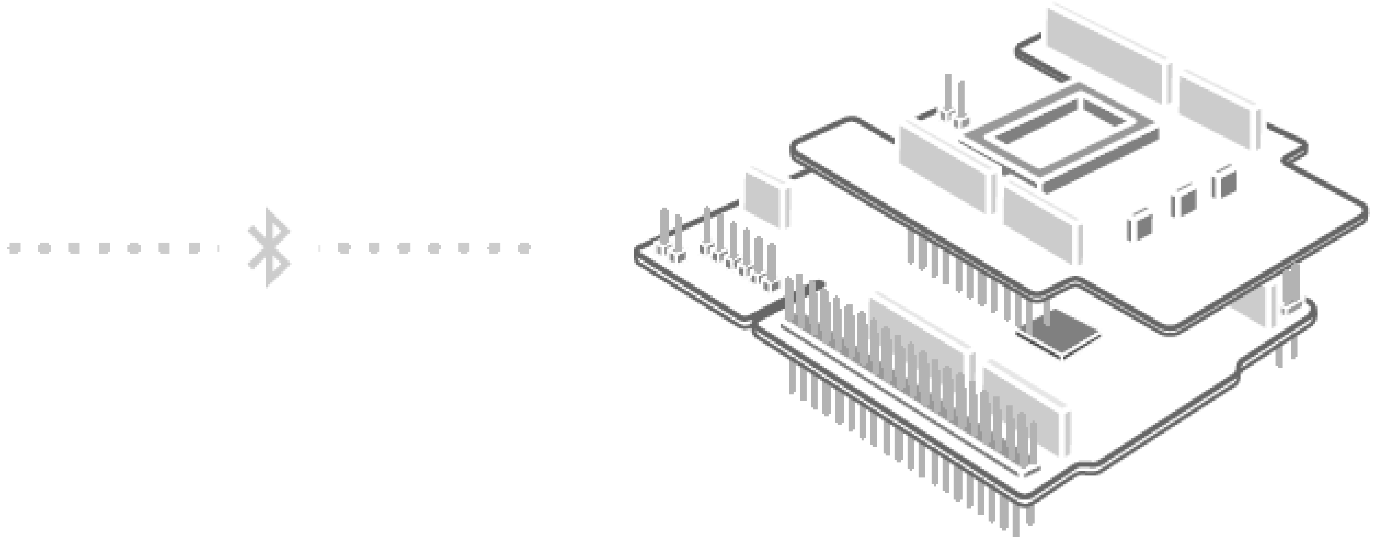
- Conveniently performed on an host
- The learned dictionary D_{u,r_0} and all its transformed versions D_{u,r_i} are transferred to the dongle

Carrera D., Rossi B., Zambon D., Fragneto P., and Boracchi G. "ECG Monitoring in Wearable Devices by Sparse Models", ECML-PKDD 2016
Longoni M., Carrera D., Rossi B., Fragneto P., Pessione M., Boracchi G A Wearable Device for Online and Long-Term ECG Monitoring , *International Joint Conference on Artificial Intelligence (IJCAI) 2018 - Demo Track*

Online ECG Signals by Wearable Devices

Online Monitoring:
The dongle computes the heart rate and selects the correct dictionary for performing the sparse coding





Nucleo
STM32L476RG

Results: MIT-BIH dataset

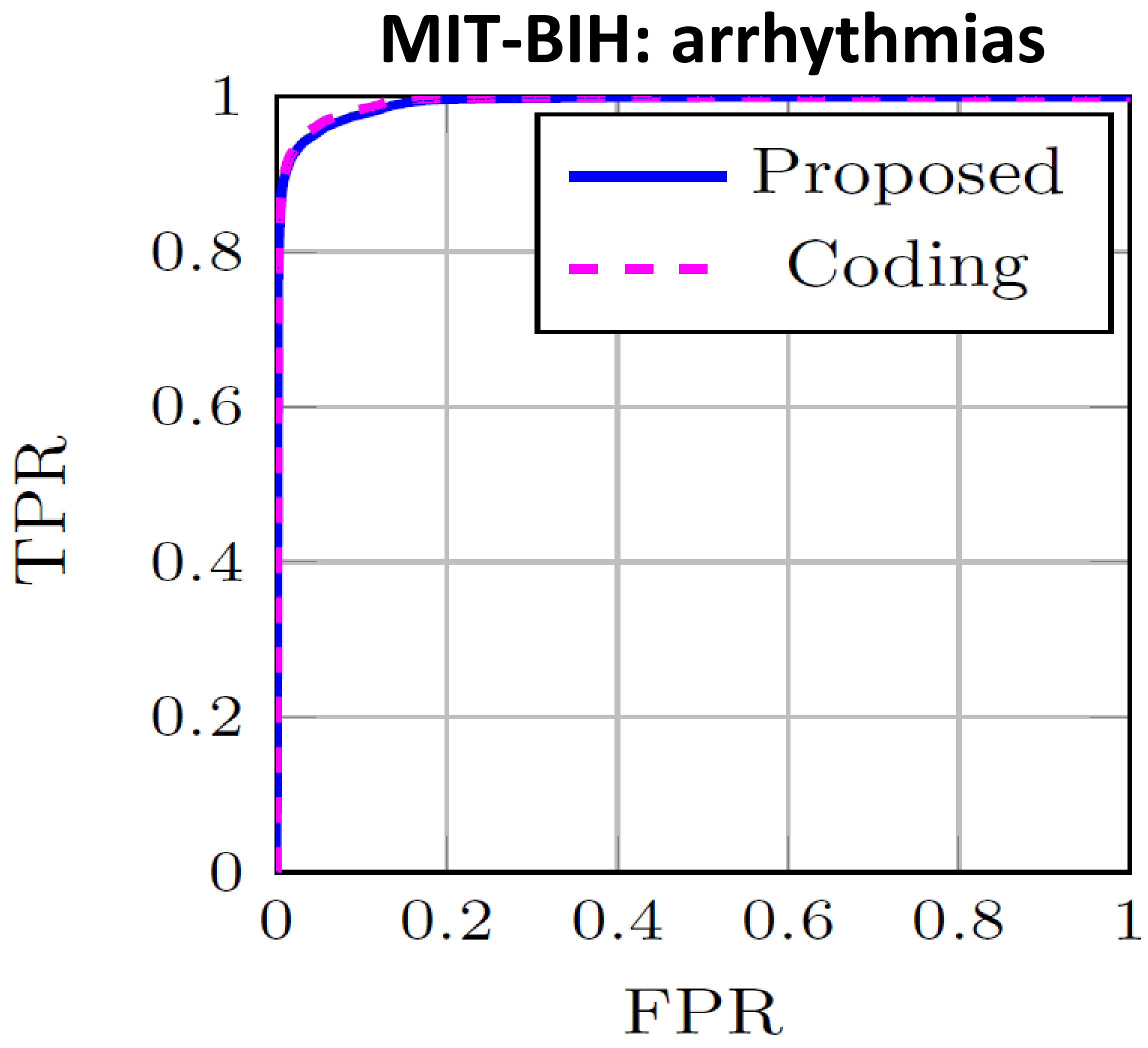
Performance measures

- FPR: false positive rate
- TPR: true positive rate
- ROC curve: reports TPR against FPR at different detection sensitivity

Remarks:

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

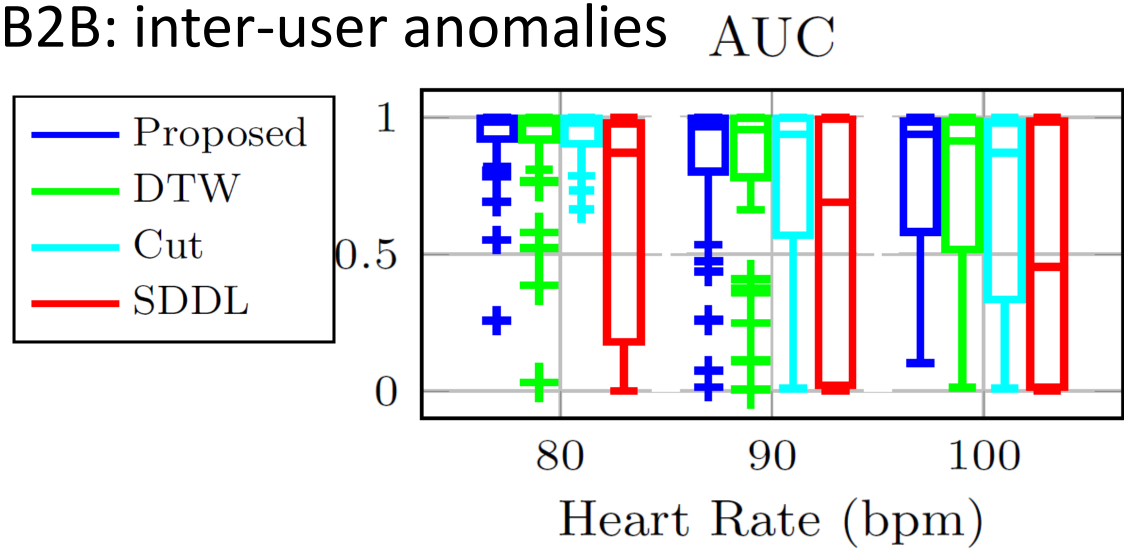
However, our detector is much less computationally demanding



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

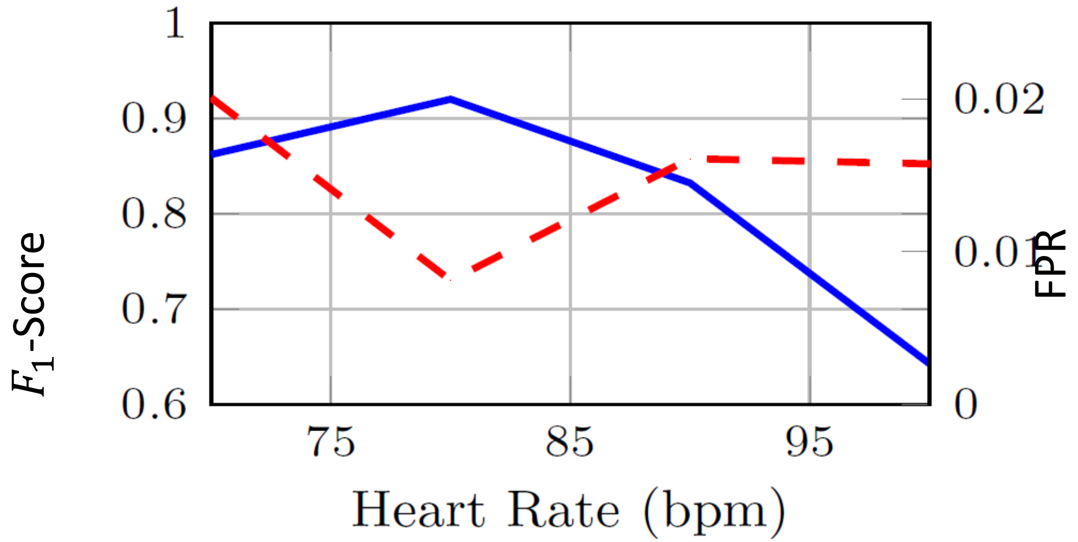
B2B dataset (in-house dataset with arrhythmias)

B2B: inter-user anomalies



Performance measures

- **AUC**: area under the ROC curve. The closer to 1, the better
- $\boldsymbol{F_1}$-score: combines both FPR and TPR in a global indicator. The larger thebetter



Remarks

Both the AUC and the F_1-score are large when the heart rate increases.

The FPR is maintained almost constant

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

Credit Cart Fraud Detection

Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi and Gianluca Bontempi, "**Credit Card Fraud Detection: a Realistic Modeling and a Novel Learning Strategy**", *IEEE Transactions on Neural Networks and Learning Systems, 2017*

A collaboration with…

Since November 2014 we started a collaboration with

- The Machine Learning Group at Université Libre de Bruxelles, Belgium (Prof. Gianluca Bontempi).
- Atos Wordline, a Belgium company that analyses about 600K credit card transactions everyday

The Fraud Detection System (FDS)

The Fraud Detection System (FDS):

- Performs security controls to prevent frauds
- Automatically analyzes all the authorized transactions and **alerts** the most suspicious ones
- Involves investigators that check the alerts and possibly block fraudulent cards

Fraud detection is challenging because:

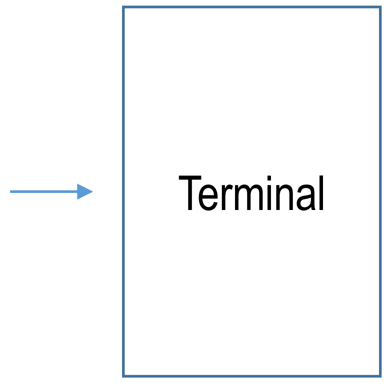
- New fraudulent strategies appear and genuine transactions might also change over time
- Genuine transactions far outnumber frauds (< 0.2%)
- Investigators that can actually check only few alerts

The **goal** of the project is to **improve** the precision of alerts automatically generated by the FDS

A Closer view on the FDS

The levels of control in the Atos Worldline FDS

The Terminal



The Terminal

Acceptance checks like:

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

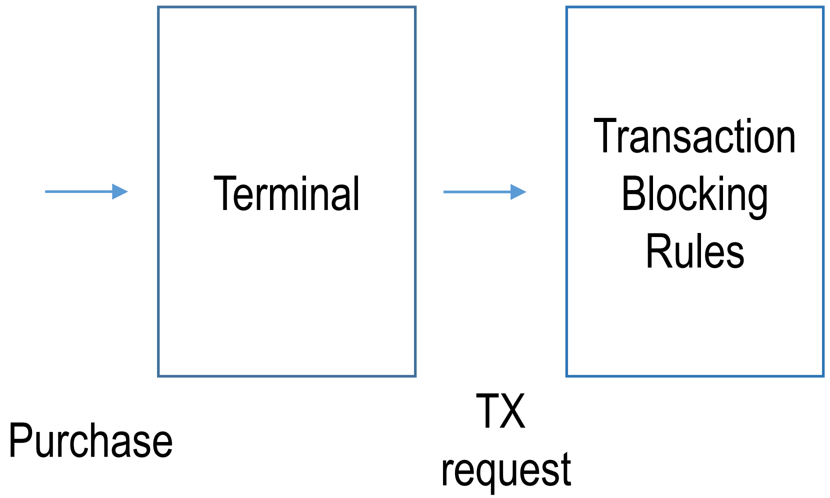
are immediately performed.

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

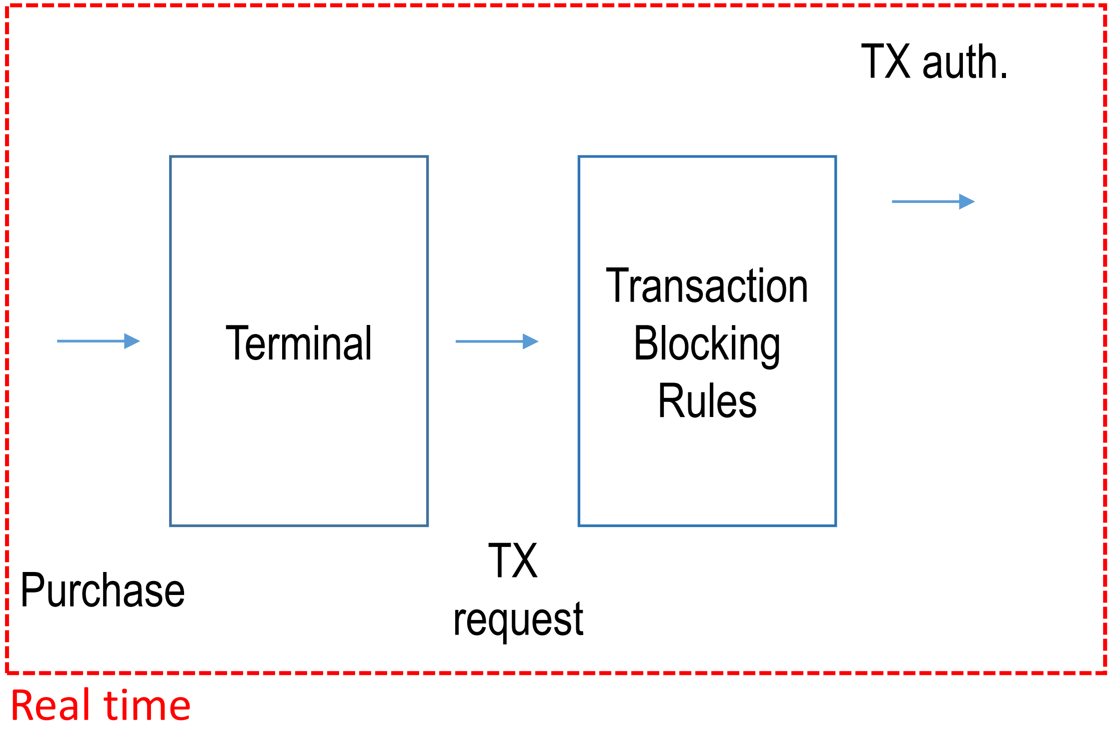
These rules:

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

All the transaction RX passing these rules becomes **authorized transactions** and further analysed by the FDS

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

Real Time Processing



Feature Augmentation

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

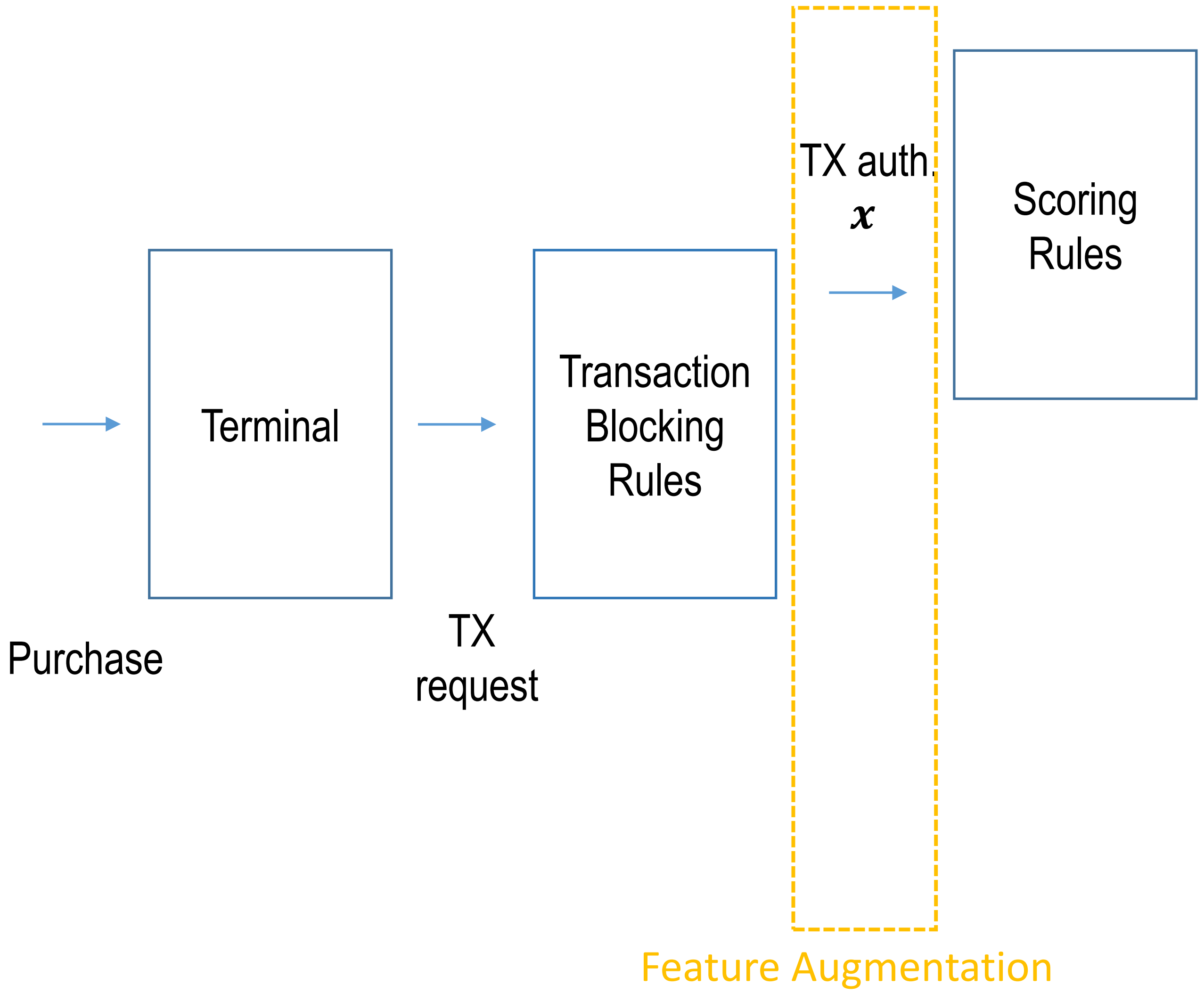
The components of $\boldsymbol{x}$ include data about the current transaction and customary shopping habits of the cardholder, e.g.:

- 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



Scoring Rules

Scoring rules are if-then-else statement that:

- Are **expert-driven**, designed by investigators.
- Operate on augmented features (components of $\boldsymbol{x}$)
- Assign a **score**: the larger the score the more risky the transaction (an estimate of the probability for $\boldsymbol{x}$ 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 operate in near-real time

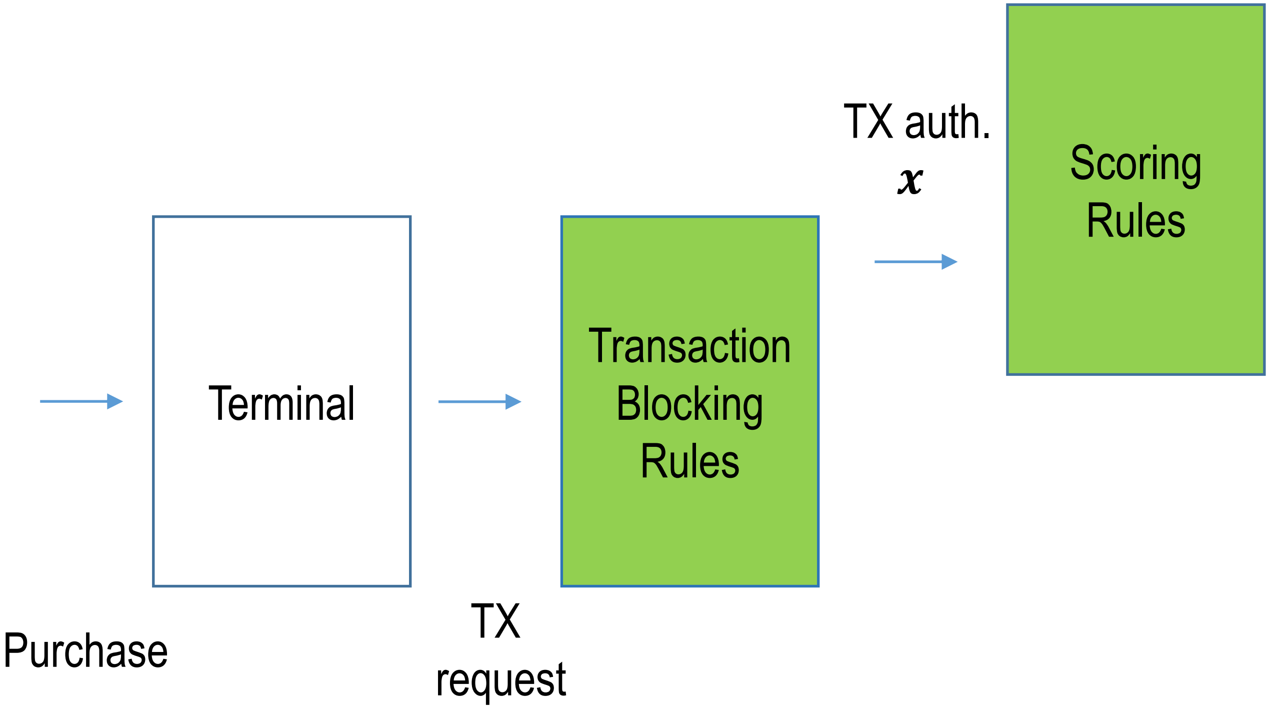
Scoring Rules

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* $+ 3\sigma$ *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 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:

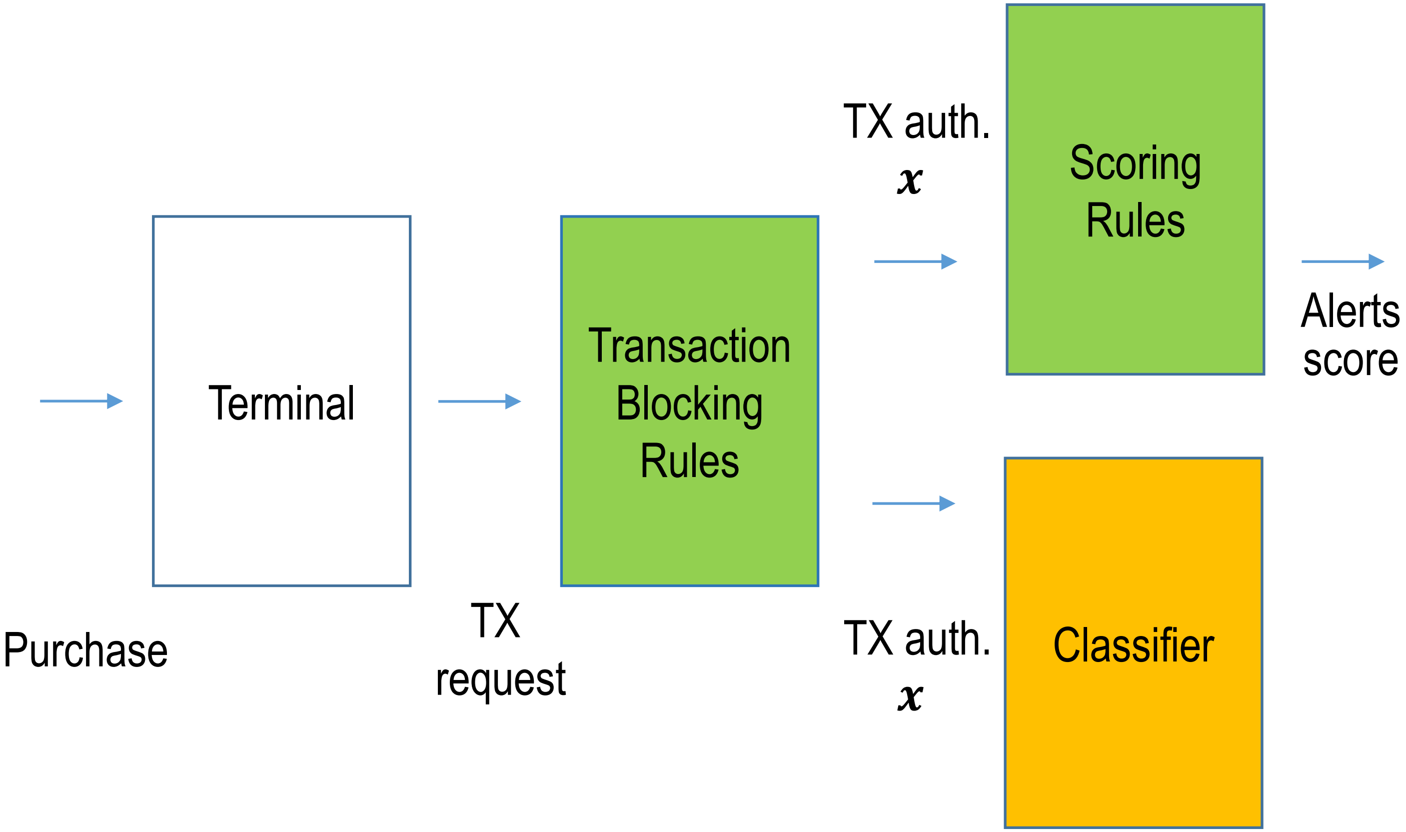
- Can detect **well-known / reasonable** frauds
- Involve **few components** of the feature vector
- **Difficult** to **exploit correlation** among features

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

.. as far as these are meaningful for separating frauds from genuine transactions

Alerts Generation



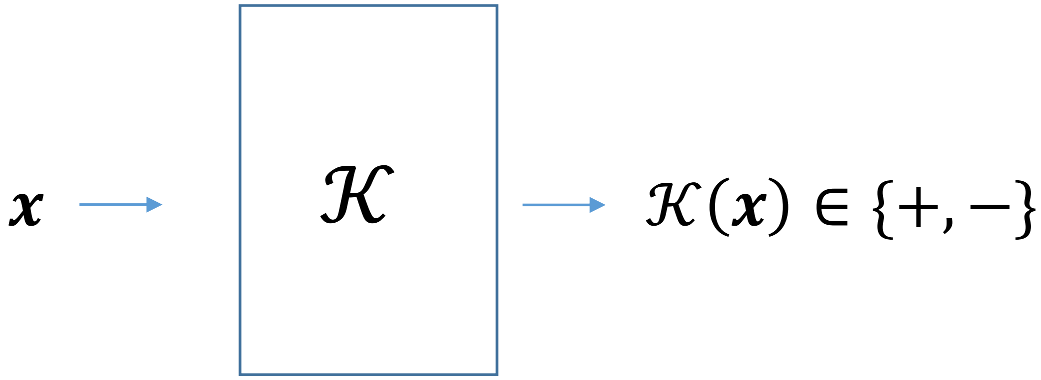
The Classifier in the FDS

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

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

where the label $y = \{+, -\}$, i.e., $\{«fraud», «genuine»\}$

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



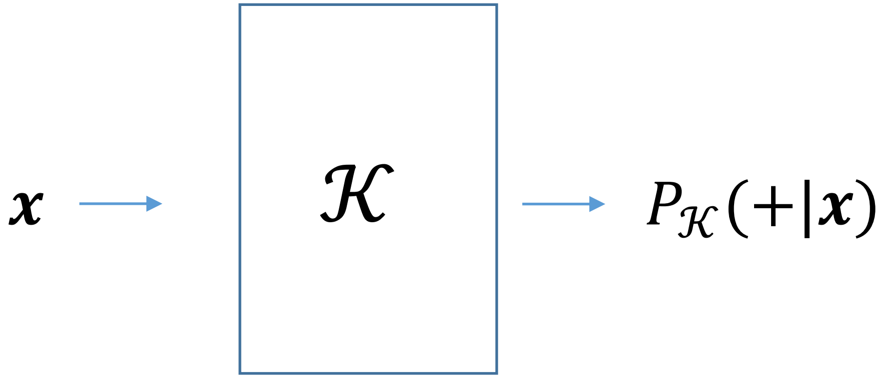
$\mathcal{K}$ considers transactions labeled as '+' as frauds

Alerts Reported to Investigators

It is not feasible to alert all transactions labeled as frauds

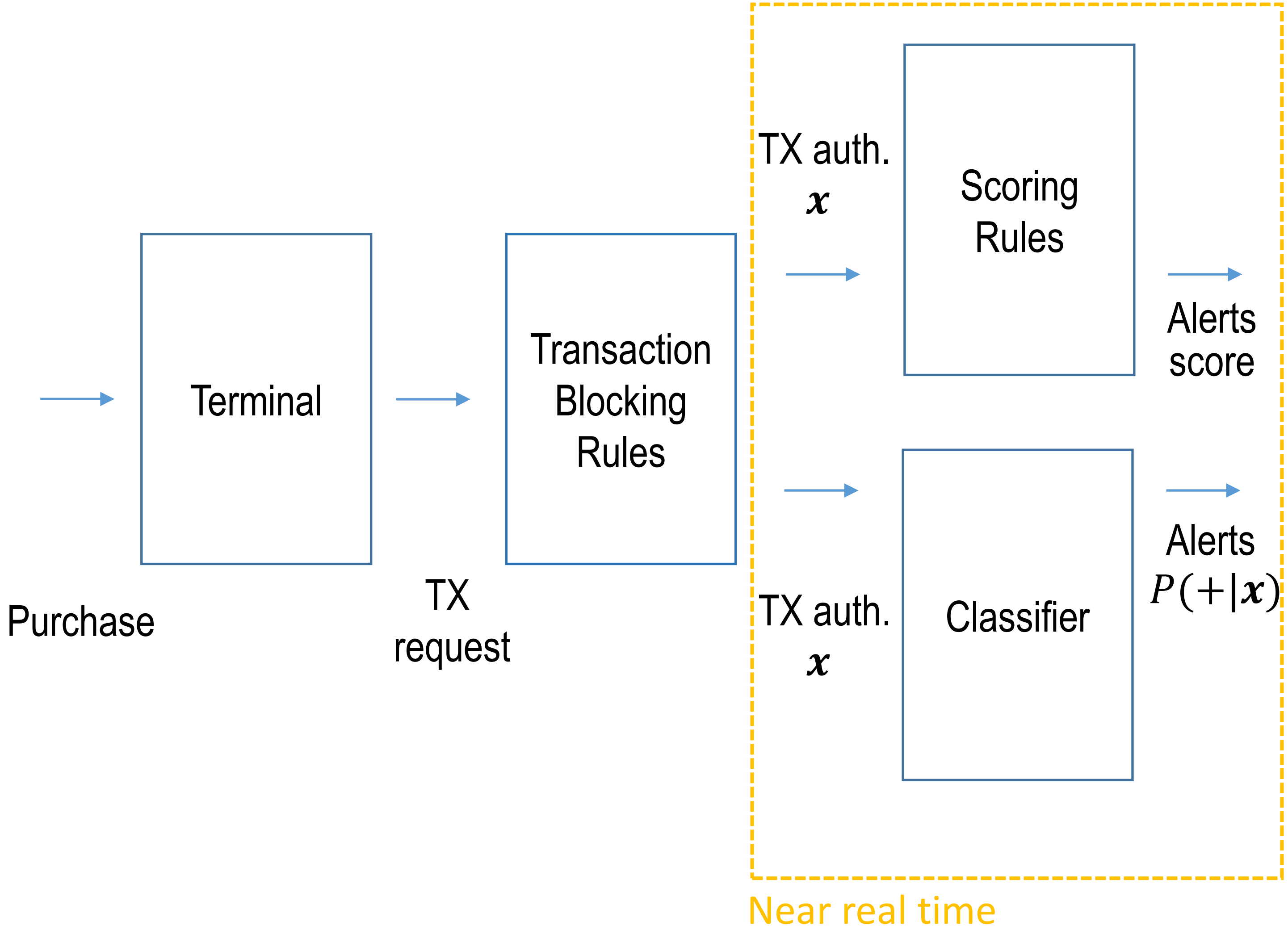
Only few transactions that are **very likely** to be frauds can be alerted.

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

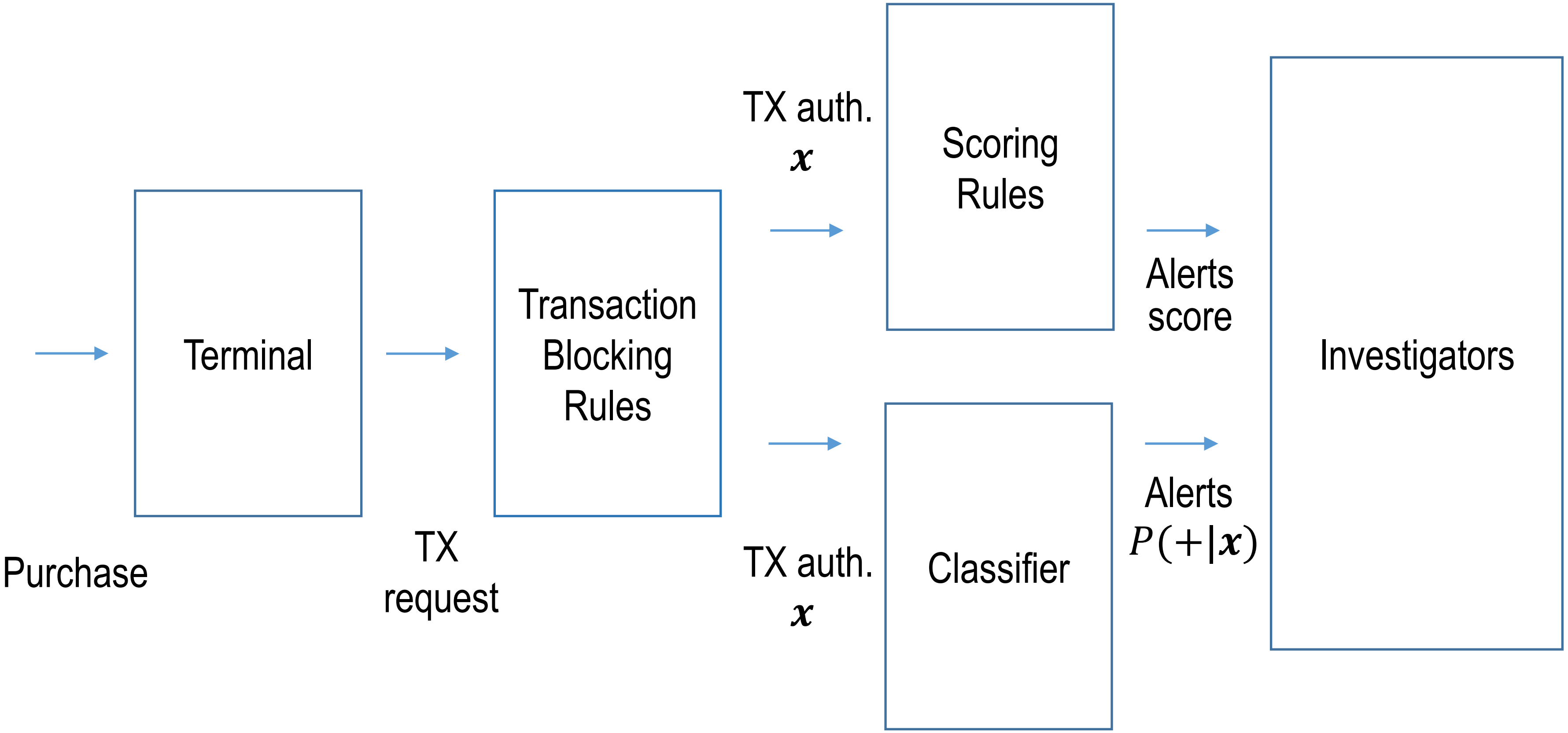


and only transactions yielding $P_{\mathcal{K}}(+|\boldsymbol{x}) \approx 1$ raise an alert

Near real time processing



Investigators



Investigators

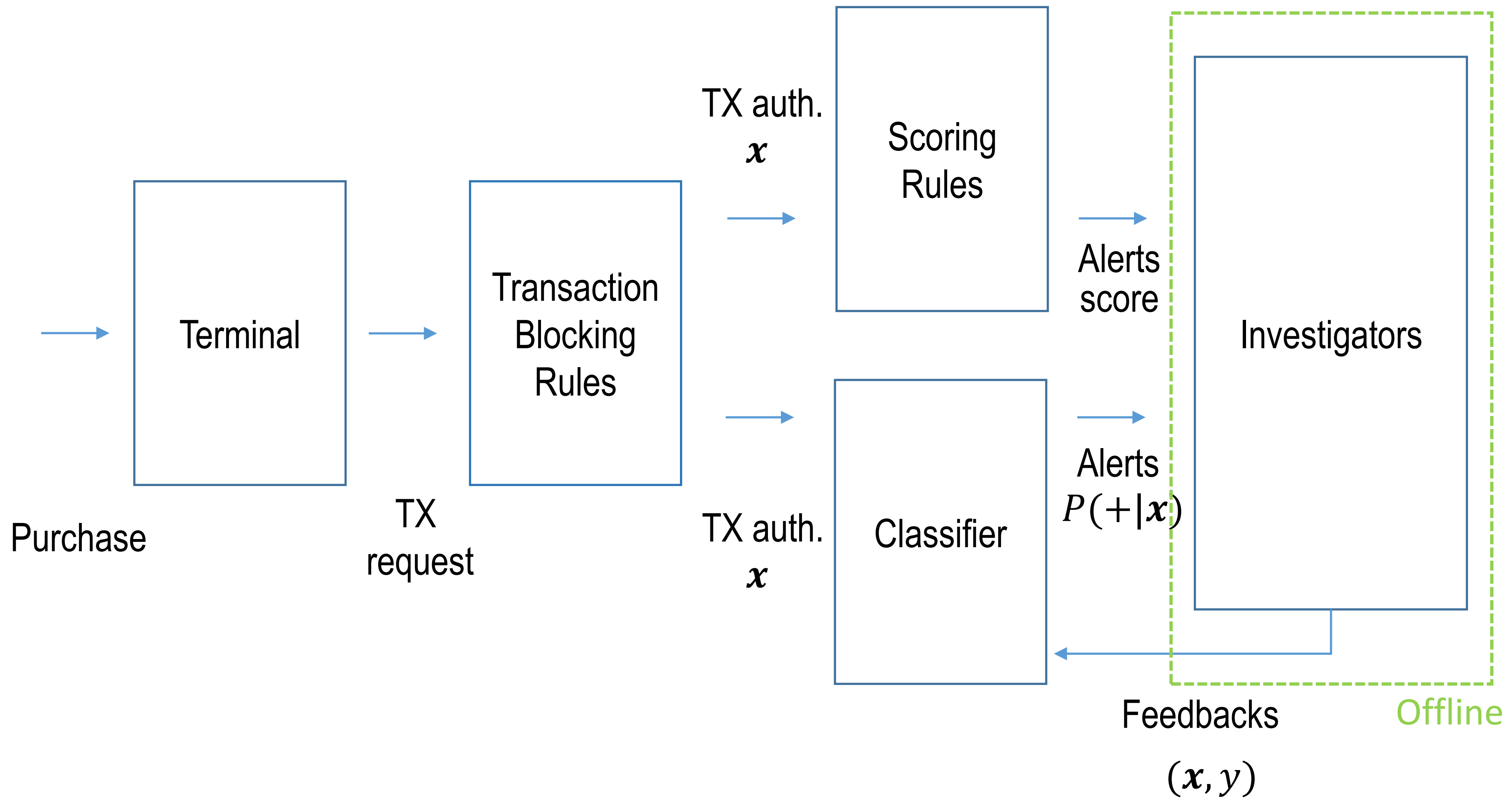
Investigators are **professionals** that are experienced in analyzing credit card transactions:

- 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



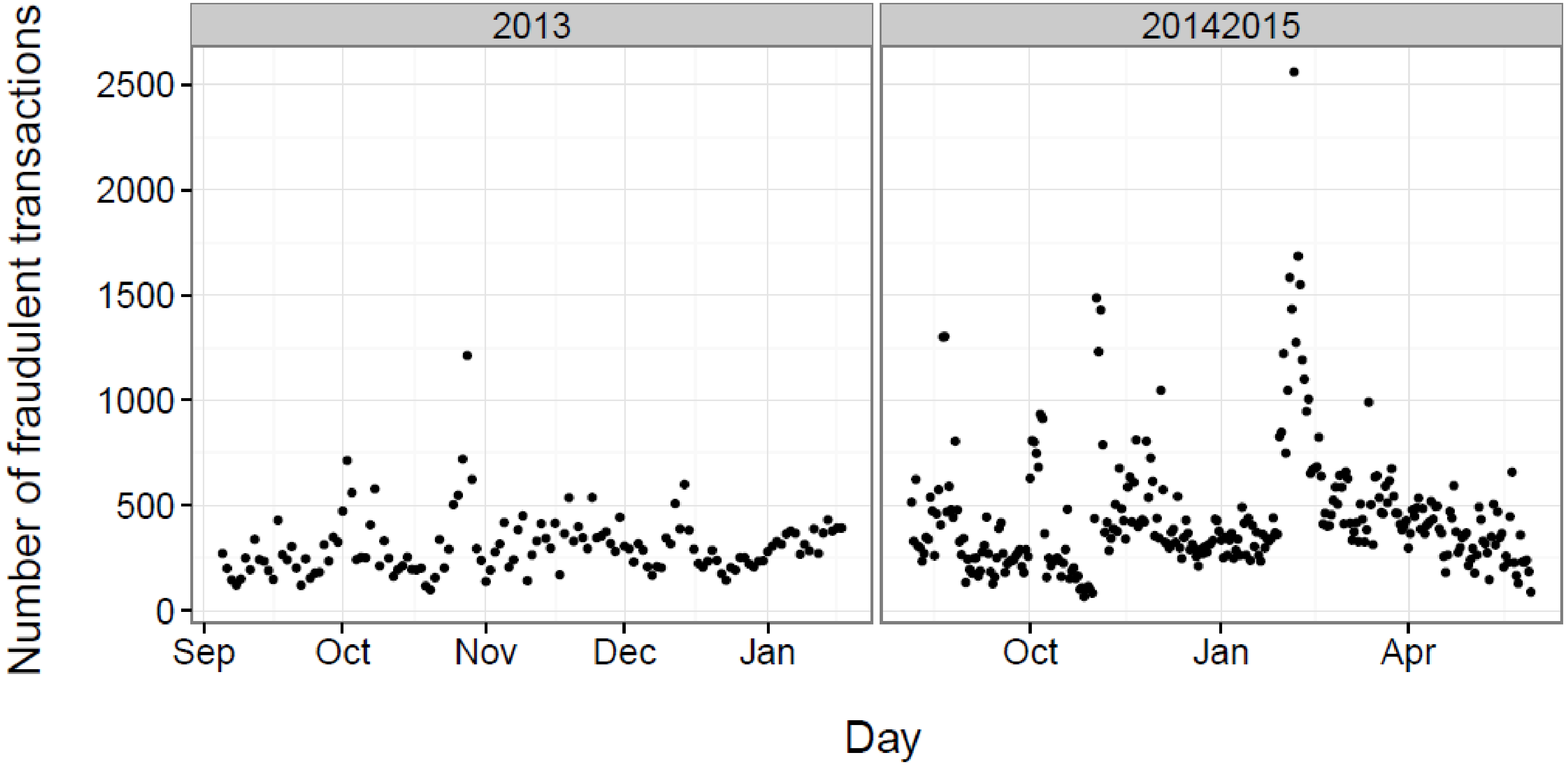
Reseach Challenges When using a Data-Driven Model in a FDS

First Research Challenges: Class Imbalance

In the real world, genuine transactions far outnumber the fraudulent ones.

The overall number of fraudulent transactions is less 0.2%

These are the statistics for our datasets



First Research Challenges: Class Imbalance

An unbalanced training set could lead $\mathcal{K}$ to consider all transactions as "genuine", as this solution yield the smaller number of misclassified samples.

Main solutions:

- Resampling to balancing the proportion of classes
- Reweighting of training samples or cost-sensitive learning to assign different misclassification penalties

The best one also depends on the specific classifier in use.

In our experiments we used Random Forest (that are particularly effective in FDS) and can be easily combined with resampling methods.

Second Research Challenge: Concept Drift

In practice:

- Fraudsters constantly prepare new attacks
- Genuine purchases follow seasonality
- Everybody changes his own shopping habit over time

⇒ the process generating $\boldsymbol{x}$ is **nonstationary**

$$\boldsymbol{x} \sim \mathcal{X}_t$$

Concept Drift: a change in the data-generating process

The FDS become **obsolete** soon since:

- Expert-driven rules become inadequate and could not detect frauds or report to many false alerts
- $\mathcal{K}$ assumes $\boldsymbol{x}$ follow the same distribution of training data: when $\mathcal{X}$ changes, $\mathcal{K}$ becomes unfit

Adaptation in a FDS

Expert-driven rules are added/updated/removed by investigators according to the most recent trends

This is very important to timely :

- Include prior information in the FDS
- Detect specific (known) fraudulent patterns

In contrast, investigators **cannot manipulate** $\mathcal{K}$, which requires to be updated/retrained from data

Learning in Nonstationary Environment (NSE)

Learning methods for NSE is an important research topic in computational intelligence community

Two strategies for learning/adapting a DDM in a NSE

Learning in Nonstationary Environment (NSE)

Learning methods for NSE is an important research topic in computational intelligence community

Two strategies for learning/adapting a DDM in a NSE

- Active approach («Detect and React»)

Active approach **(our expertise)**:

- Monitor by a **change-detection test** the performance of $\mathcal{K}$ or distribution of $\boldsymbol{x}$ to **detect concept drift**
- After each detection **automatically** identifies suitable training data coherent with the current state of $\mathcal{X}$
- **Reconfigure** $\mathcal{K}$ only when a change is detected
- Provide information **when the change has occurred**

Learning in Nonstationary Environment (NSE)

Learning methods for NSE is an important research topic in computational intelligence community

Two strategies for learning/adapting a DDM in a NSE

- Active approach («Detect and React»)
- Passive approach (Continuous adaptation)

Passive approach:

- $\mathcal{K}$ is **steadily updated** on recent supervised samples
- No **change-detection information**

Learning in Nonstationary Environment (NSE)

Learning methods for NSE is an important research topic in computational intelligence community

Two strategies for learning/adapting a DDM in a NSE

- Active approach («Detect and React»)
- Passive approach (Continuous adaptation)

Which is the best depends on:

- Availability of supervised information,
- System resources
- Expected change rate/type
- Interest of having information about the change

Adaptation in a FDS

In a FDS, the passive approach is the most suited.

In practice, $\mathcal{K}$ is **updated**, as soon as **enough** supervised samples are gathered.

Then, $\mathcal{K}$ becomes $\mathcal{K}_t$ and is updated (say) everyday

Sliding Window Approach: use supervised information from the last δ days

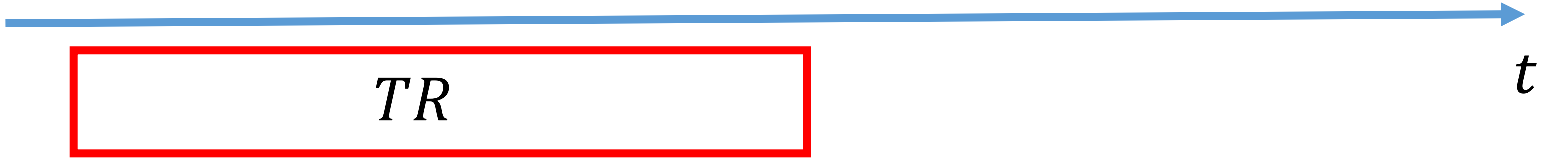
Adaptation in a FDS

In a FDS, the passive approach is the most suited.

In practice, $\mathcal{K}$ is **updated**, as soon as **enough** supervised samples are gathered.

Then, $\mathcal{K}$ becomes $\mathcal{K}_t$ and is updated (say) everyday

Sliding Window Approach: use supervised information from the last δ days



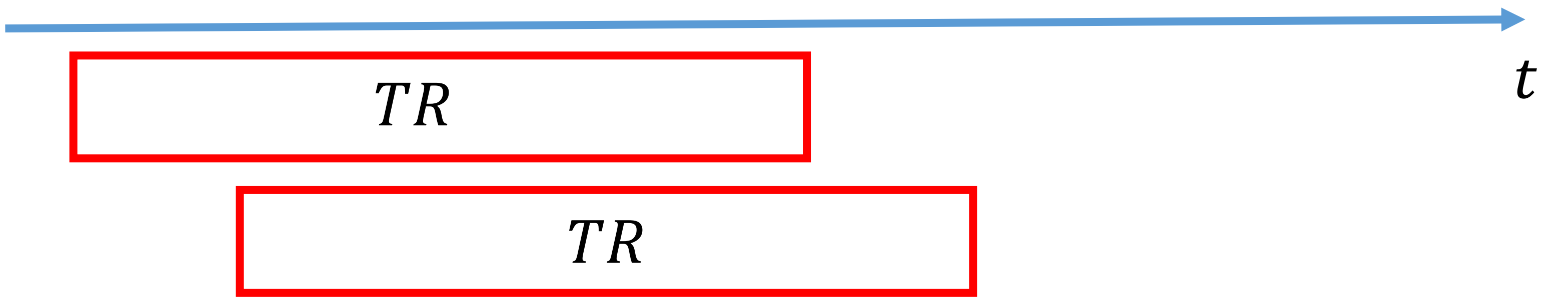
Adaptation in a FDS

In a FDS, the passive approach is the most suited.

In practice, $\mathcal{K}$ is **updated**, as soon as **enough** supervised samples are gathered.

Then, $\mathcal{K}$ becomes $\mathcal{K}_t$ and is updated (say) everyday

Sliding Window Approach: use supervised information from the last δ days



Adaptation in a FDS

In a FDS, the passive approach is the most suited.

In practice, $\mathcal{K}$ is **updated**, as soon as **enough** supervised samples are gathered.

Then, $\mathcal{K}$ becomes $\mathcal{K}_t$ and is updated (say) everyday

Sliding Window Approach: use supervised information from the last δ days

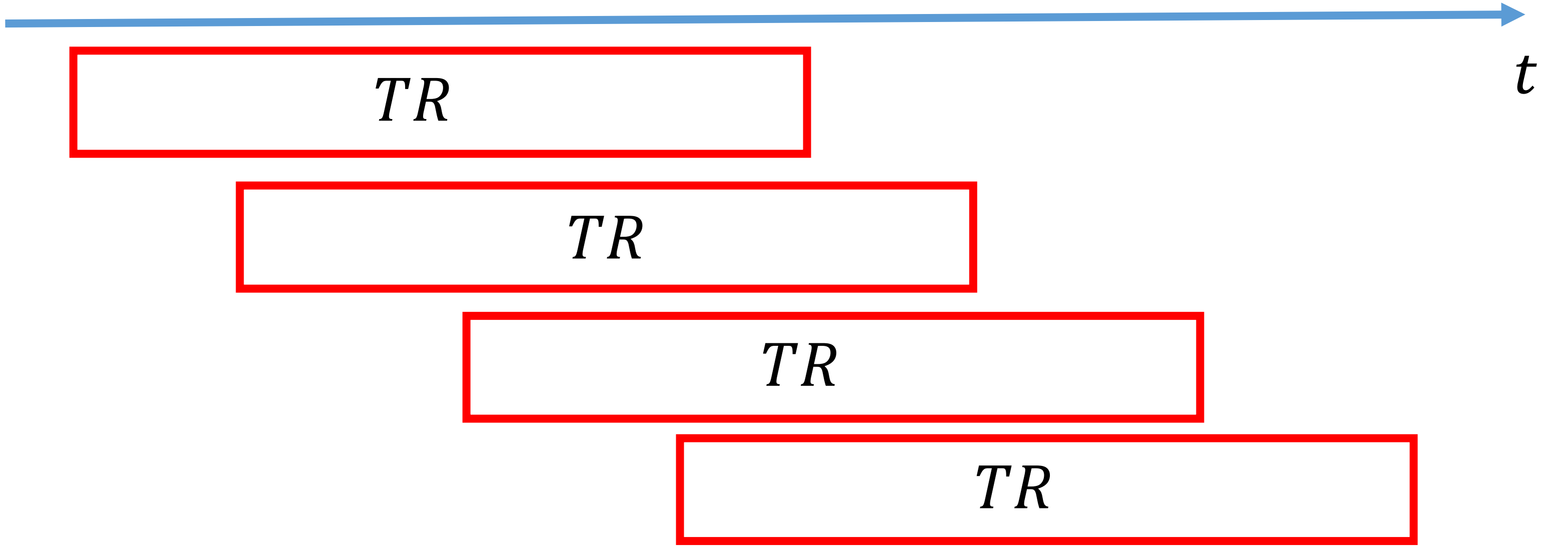
Adaptation in a FDS

In a FDS, the passive approach is the most suited.

In practice, $\mathcal{K}$ is **updated**, as soon as **enough** supervised samples are gathered.

Then, $\mathcal{K}$ becomes $\mathcal{K}_t$ and is updated (say) everyday

Sliding Window Approach: use supervised information from the last δ days



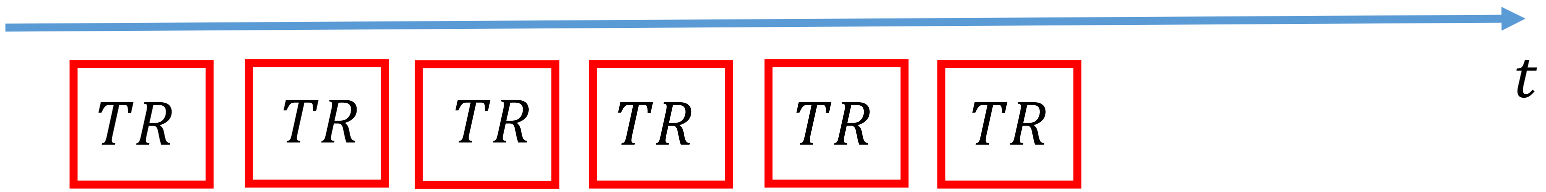
Adaptation in a FDS

In a FDS, the passive approach is the most suited.

In practice, $\mathcal{K}$ is **updated**, as soon as **enough** supervised samples are gathered.

Then, $\mathcal{K}$ becomes $\mathcal{K}_t$ and is updated (say) everyday

Ensemble Approach: train a different classifier on each day and then aggregate their outputs



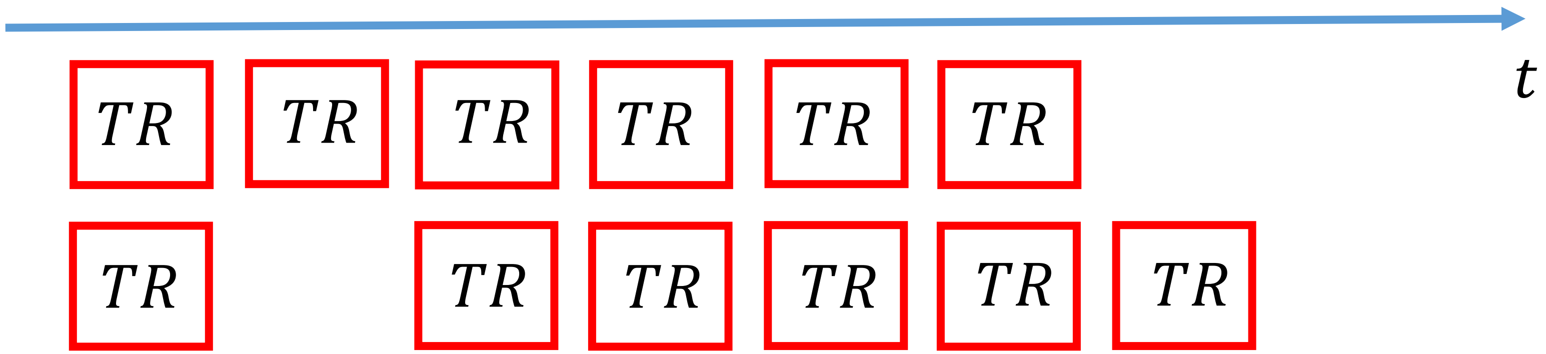
Adaptation in a FDS

In a FDS, the passive approach is the most suited.

In practice, $\mathcal{K}$ is **updated**, as soon as **enough** supervised samples are gathered.

Then, $\mathcal{K}$ becomes $\mathcal{K}_t$ and is updated (say) everyday

Ensemble Approach: train a different classifier on each day and then aggregate their outputs



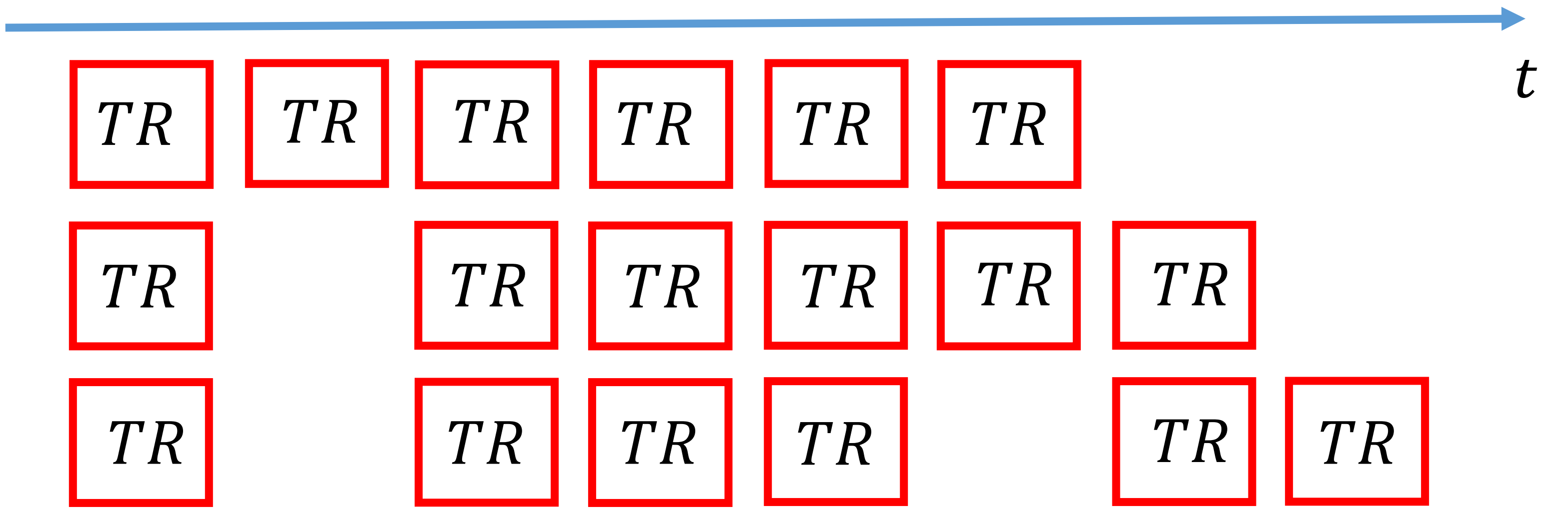
Adaptation in a FDS

In a FDS, the passive approach is the most suited.

In practice, $\mathcal{K}$ is **updated**, as soon as **enough** supervised samples are gathered.

Then, $\mathcal{K}$ becomes $\mathcal{K}_t$ and is updated (say) everyday

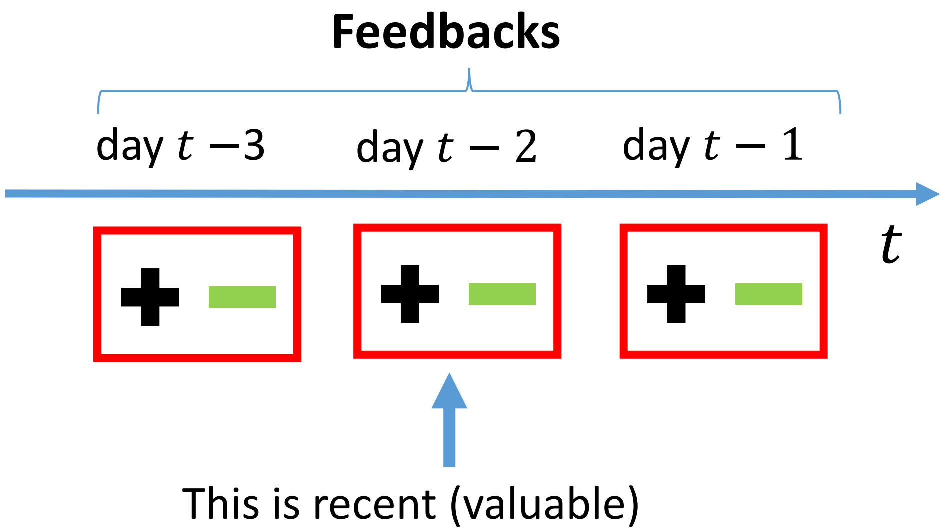
Ensemble Approach: train a different classifier on each day and then aggregate their outputs



Supervised Information in a FDS

The supervised information available in the FDS is:

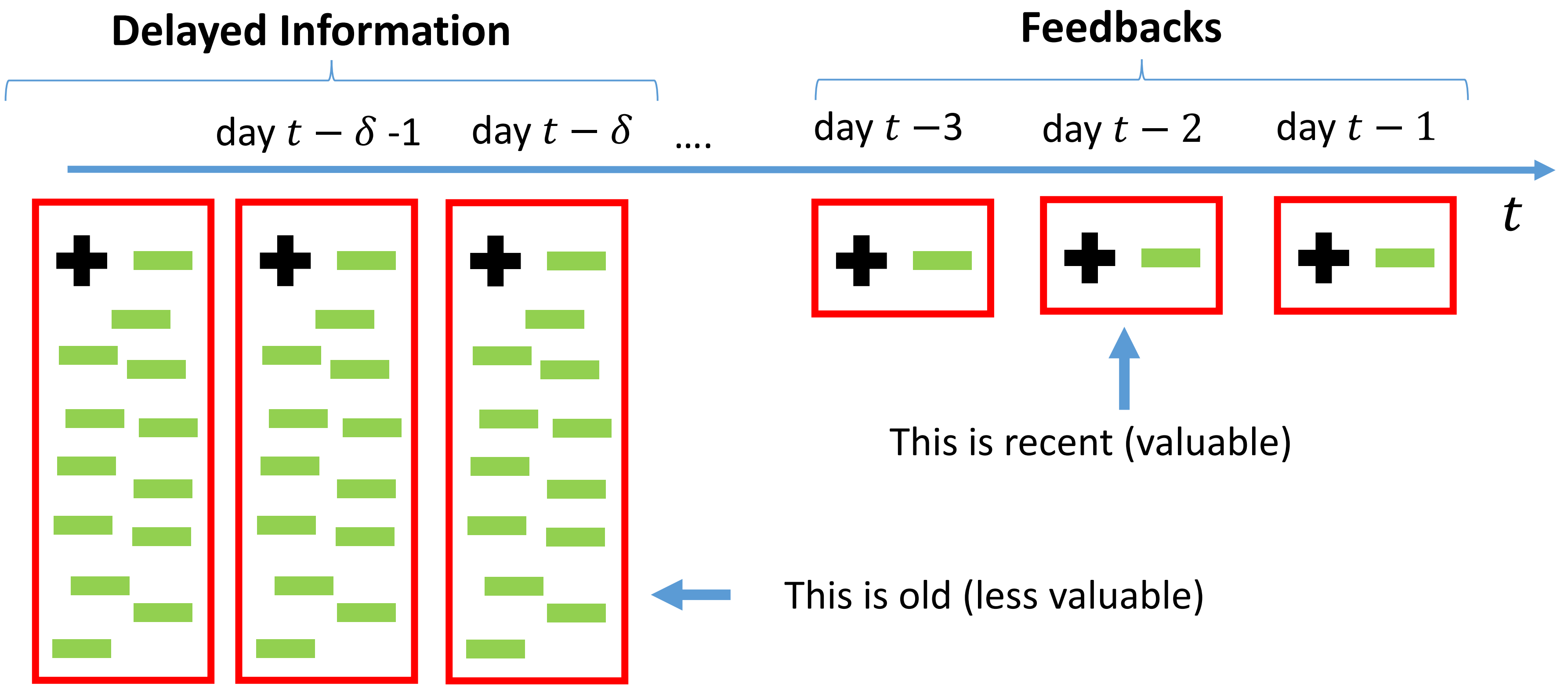
- **Few, very recent feedbacks** of yesterday’s alert



Supervised Information in a FDS

The supervised information available in the FDS is:

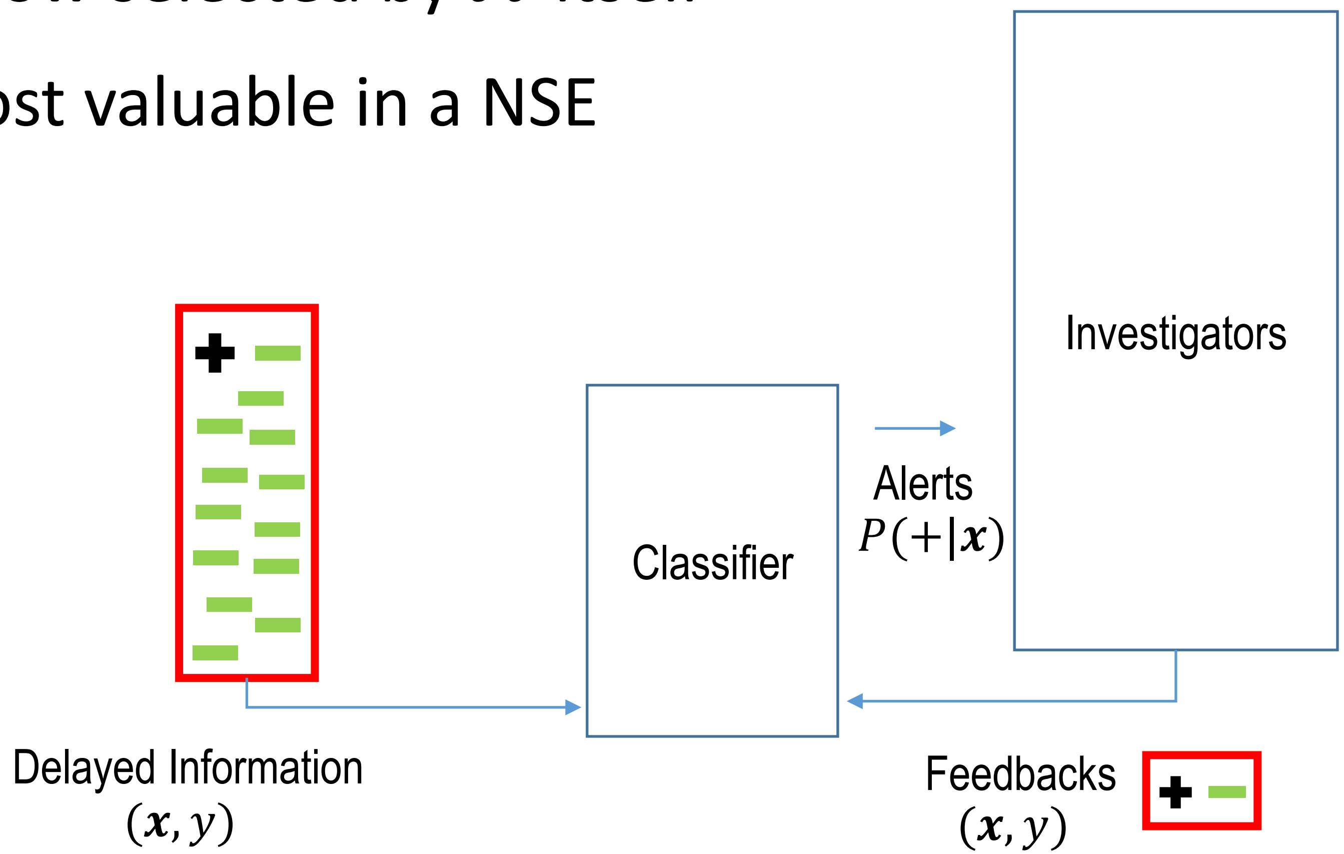
- **Few, very recent feedbacks** of yesterday's alert
- **All the transactions** authorized **several days before** (verification latency)



Third Research Challenge: Sample Selection Bias

The only recent supervised information is provided by the «alert-feedback» interaction

- Feedbacks are somehow selected by $\mathcal{K}$ itself
- Feedbacks are the most valuable in a NSE



Third Research Challenge: Sample Selection Bias

Why are feedbacks and delayed samples different?

- They have different class proportions
- Feedbacks are only highly suspicious transactions
- Feedbacks are more recent than the others

When training and testing distributions are different there is a **sample selection bias**

Main solutions in to correct sample selection bias

- Importance weighting
- Ensemble methods using unsupervised samples

An Effective Solution to Fraud Detection

"Feedback and delayed samples are different in nature and should be exploited differently"

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 TNNLS 2017

An Effective Solution to Fraud Detection

“Feedback and delayed samples are different in nature and should be exploited differently”

Learn two separate classifiers from:

- Feedback (get a classifier $\mathcal{F}$)
- Delayed Samples (get a classifier $\mathcal{D}$)

Aggregate the outputs

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 TNNLS 2017

An Effective Solution to Fraud Detection

"Feedback and delayed samples are different in nature and should be exploited differently"

Learn two separate classifiers from:

- Feedback (get a classifier $\mathcal{F}$)
- Delayed Samples (get a classifier $\mathcal{D}$)

Aggregate the outputs

$$P_{\mathcal{K}}\left(+|\boldsymbol{x}\right) = \alpha P_{\mathcal{F}}\left(+|\boldsymbol{x}\right) + (1-\alpha)\, P_{\mathcal{D}}\left(+|\boldsymbol{x}\right)$$

Sliding Window

	Dataset 2013		Dataset 2014	
classifier	mean	sd	mean	sd
$\mathcal{F}$	0.609	0.250	0.596	0.249
$\mathcal{W}^D$	0.540	0.227	0.549	0.253
$\mathcal{W}$	0.563	0.233	0.559	0.256
$\mathcal{A}^W$	0.697	0.212	0.657	0.236

Ensembles

	Dataset 2013		Dataset 2014	
classifier	mean	sd	mean	sd
$\mathcal{F}$	0.603	0.258	0.596	0.271
$\mathcal{E}^D$	0.459	0.237	0.443	0.242
$\mathcal{E}$	0.555	0.239	0.516	0.252
$\mathcal{A}^E$	0.683	0.220	0.634	0.239

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 TNNLS 2017