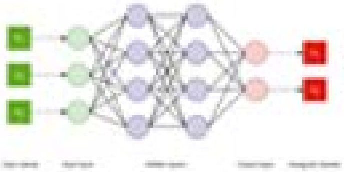
DragStream : An Interpretable Anomaly and Concept Drift Detector In Univariate Data-Streams

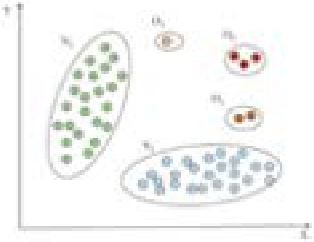
An overview of Anomaly Detection for Data Streams

Journée GT IA – Prog. DATA, 19 octobre 2023

Engelbert Mephu Nguifo



El Cheikh, 2023



Kim et al., ICML 2018

LIMOS - UMR CNRS

Laboratoire d'Informatique, de Modelisation et d'optimisation des systemes

www.limos.fr

- IDENTITY :

  Models and **Tools** for

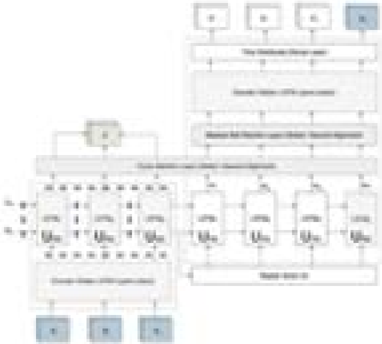
    - Design, Representation, Evaluation, Forecasting, Control, and Optimization

  of **Complex Organizational Systems**

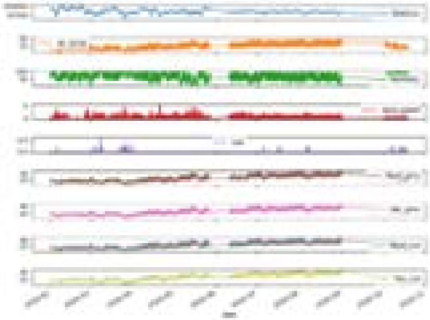
    - Transport, Telecoms, Manufacturing, Ecosystems, Biosystems

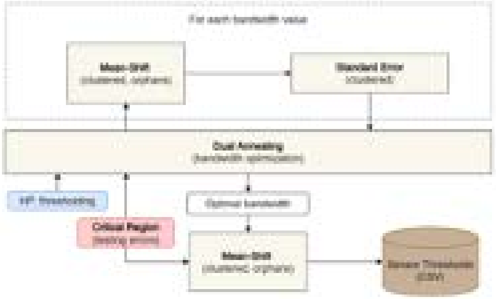
- TOPICS :
  - Models and Algorithms to support Decision making (MAAD)
    - AGC : Algorithmique, Graphes Complexité
    - MOCA : Métamodélisation, Optimisation Continue et Applications
    - …
  - Information and Communication Systems (SIC)
    - DSI : Data, Services, and Intelligence
    - …
  - Decision-support tools for Production and Services (ODPS)

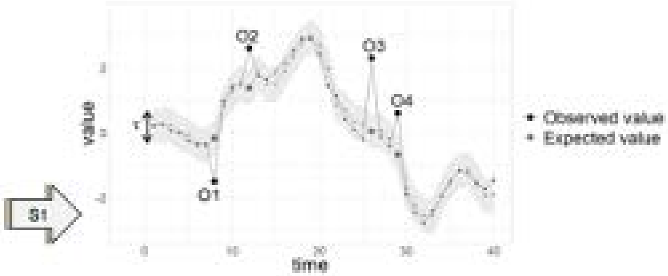
Détection d’anomalies dans des séries temporelles multivariées



- **Contexte** : ANR DIRE – Thèse M Giannoulis
- **Objectif** : prévention/prévision de risques liés à l’activité des écosystèmes hydrothermaux
- **Verrous** : données multivariées non stationnaires, haute fréquence d’acquisition,
- **Méthodes** : développement d’un algorithme ad hoc (LSTMs+mécanismes d’attention+optimisation bayésienne) (1)
- Développement d’un système expert pour l’applicatif

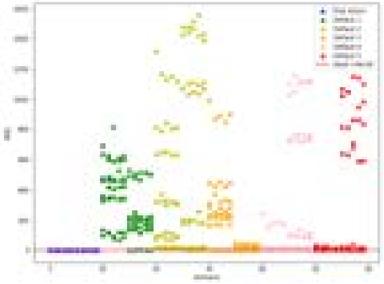




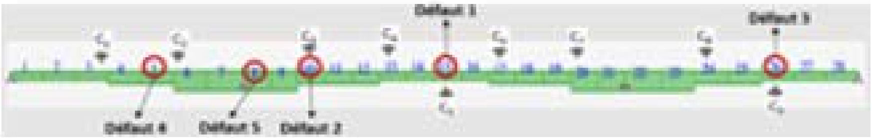


(1) M Giannoulis, A Harris, V Barra, DITAN: A deep-learning domain agnostic framework for Detection and Interpretation of Temporally-based multivariate Anomalies, **Pattern Recognition 143 (2023) 109814**

Identification et localisation de défauts structuraux par apprentissage profond



- **Contexte** : CIFRE / CIDECO – Thèse D Benhaddouche
- **Objectif** : prévention/prévision de risques liés à l'activité des écosystèmes hydrothermaux
- **Verrous** : problème inverse, incertitude dans l'estimation des paramètres, problème en grande dimension, détection de signaux faibles, remontée au défaut structural
- **Méthodes** : Modèles d'apprentissage profond (AE) (1)




```
Capteurs sur défauts:
 ['c5', 'c3', 'c9', 'c2', 'c3']
```

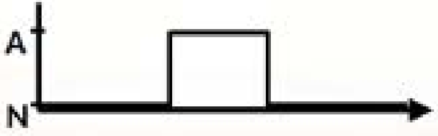
	Mode1	Mode2	Mode3	Mode4	Mode5	Mode6	Mode7	Mode8	Mode9	Mode10	Réel
D1	c5	c6	c5	c6	c5	c6	c5	c6	c5	c6	C5
D2	c3	c3	c3	c4	c4	c3	c5	c3	c3	c1	C3
D3	c9	c8	c6	c8	c8	c7	c9	c9	c6	c8	C9
D4	c2	c2	c1	c2	c2	c3	c3	c1	c2	c1	C2
D5	c3	c1	c8	c3	c1	c3	c3	c1	c3	c3	C3

(1) D Benhaddouche, A Chateauneuf , V Barra , Identification et localisation des défauts structuraux sur les ponts par auto-encodeur profond, JFMS 2023 (accepté)

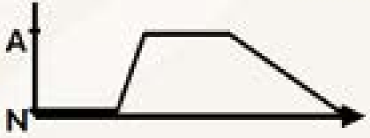
Gestion d'étiquettes incertaines

Application à l'élevage de pointe (thèse N. Wagner)

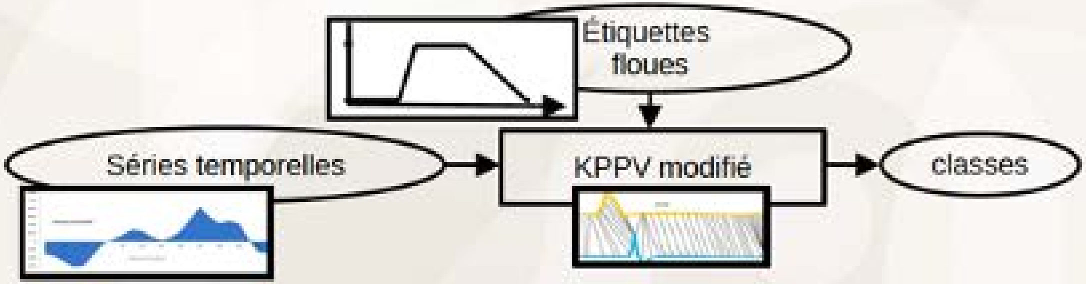
- Étiquettes Normal / Anomalie (Maladie 1, 2, …)
  - Problématique : étape binaire



- Connaissances a priori pour la création d'étiquettes possibilistes

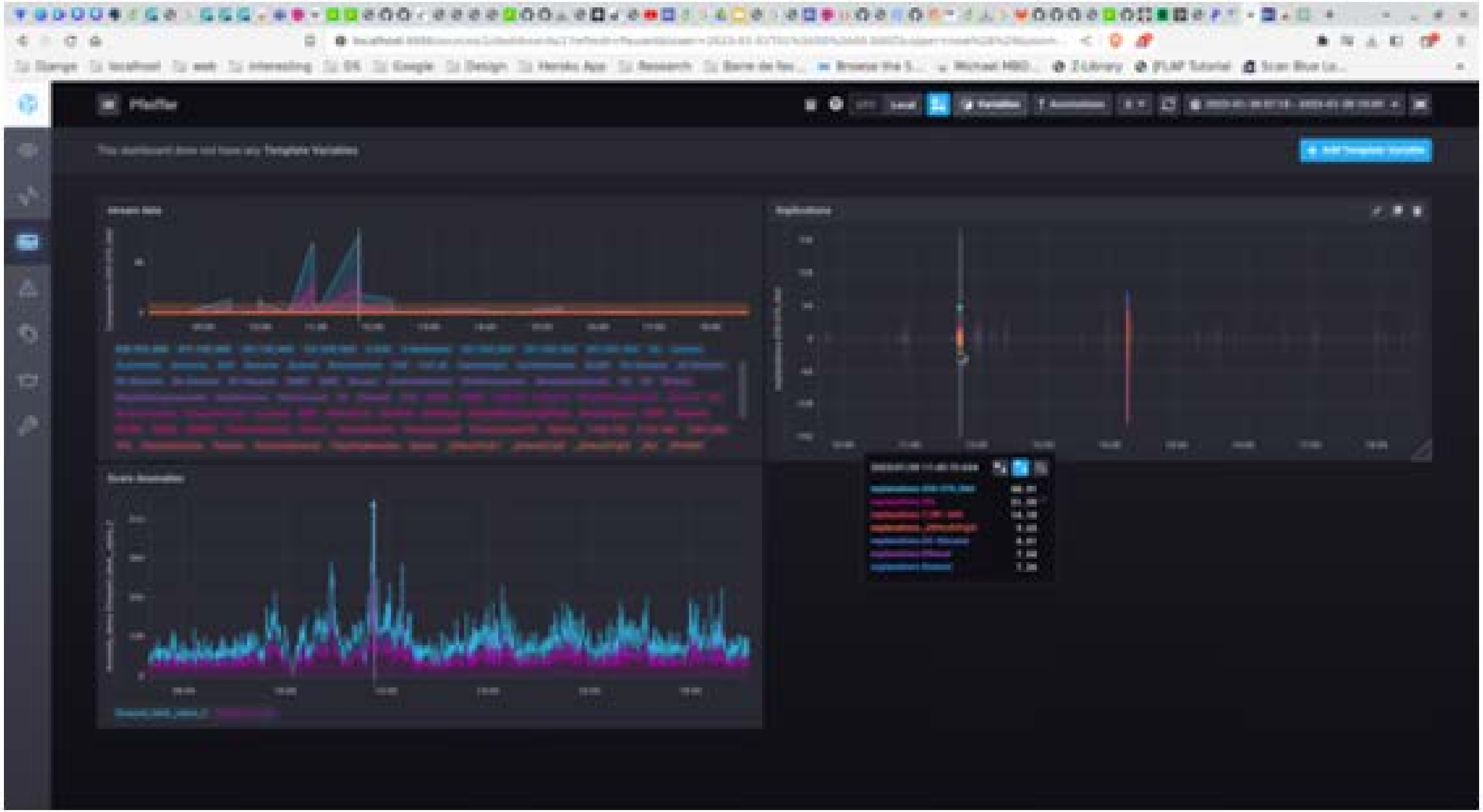


- Classification supervisée



https://theses.hal.science/tel-03121939/document

Projet DASMA : dasma.limos.fr



Content

- Basic notions
- Anomaly Detection on data stream
- Dragstream
- Challenges

Basic notions

- Anomaly / Discord / Outlier / Novelty
  - Change detection / Concept drift

- Data : where, which, what, who, why ? Data stream (Time series)
  - https://hpi-information-systems.github.io/timeeval-evaluation-paper/notebooks/Datasets.html

- Data representation

- {Supervised / Semi-supervised / Unsupervised} Learning

- Evaluation : empirical vs theoretical

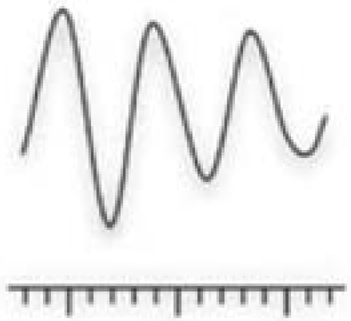
TIME SERIES :

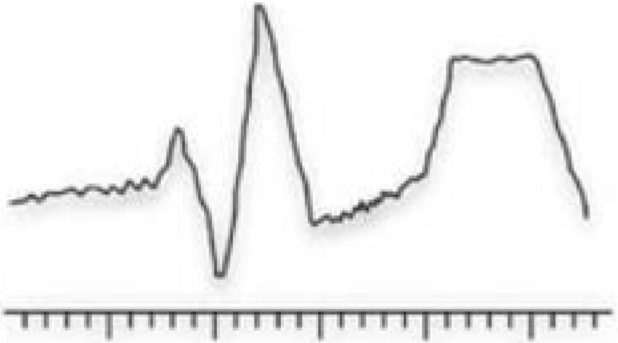
A time series (TS) represents a **collection of values** obtained from **sequential** measurements **over time**.

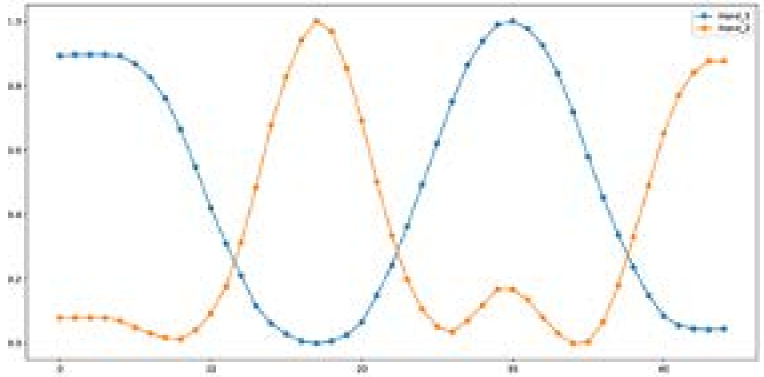
i.e a time-series T is an d-*ordered* sequence of *n real-valued* variables

$$T = [(t_1,x_1), \ldots, (t_n,x_n)],\ x_i \in R^d.$$

$$\text{If } d=1 \text{ then } T = (t_1, \ldots, t_n),\ t_i \in R.$$





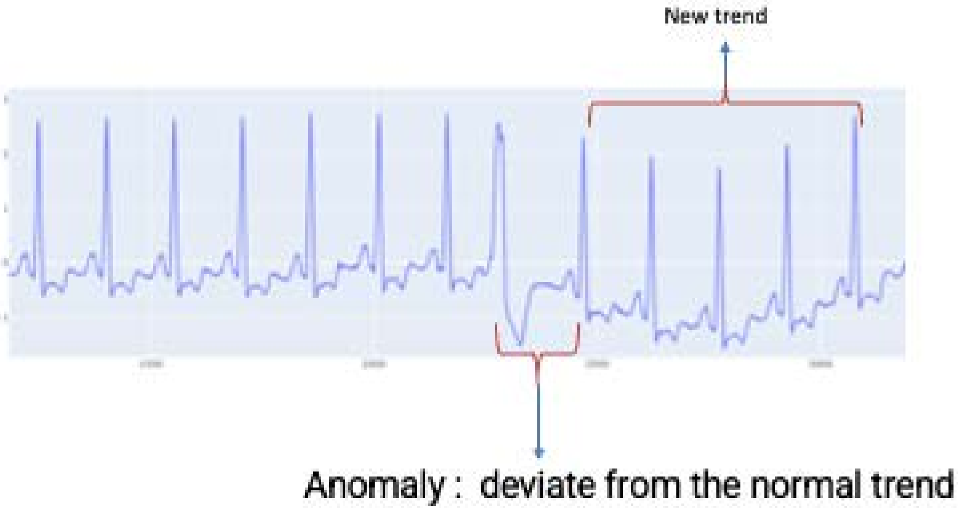


Univariate or Multivariate ?

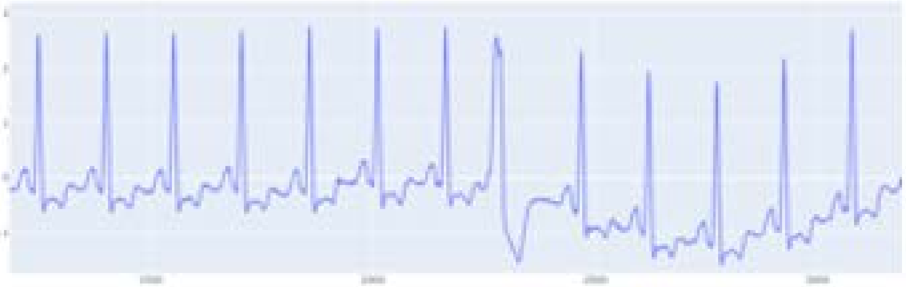
Trend, Cycle, Seasonal, Irregular ?

Time series vs Stream data ?

- Data stream
  - Data continuously arriving
  - Infinite data
  - Concept drift

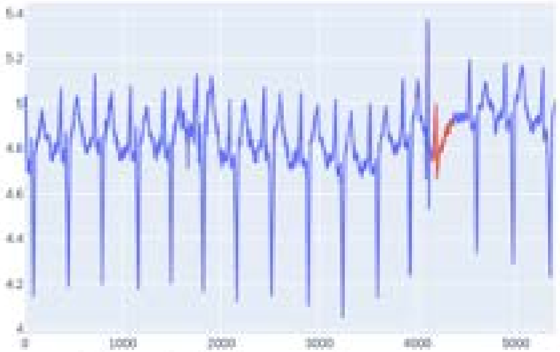


- Anomaly detection
  - Airbone contamination monitoring
  - ECG anomaly detection
  - Network intrusion detection
  - etc.

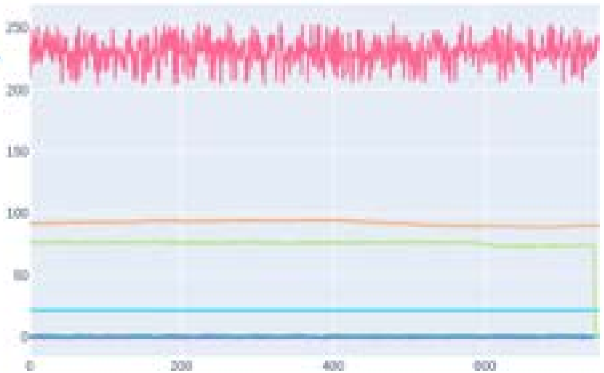


Concept drift

Abnormal point

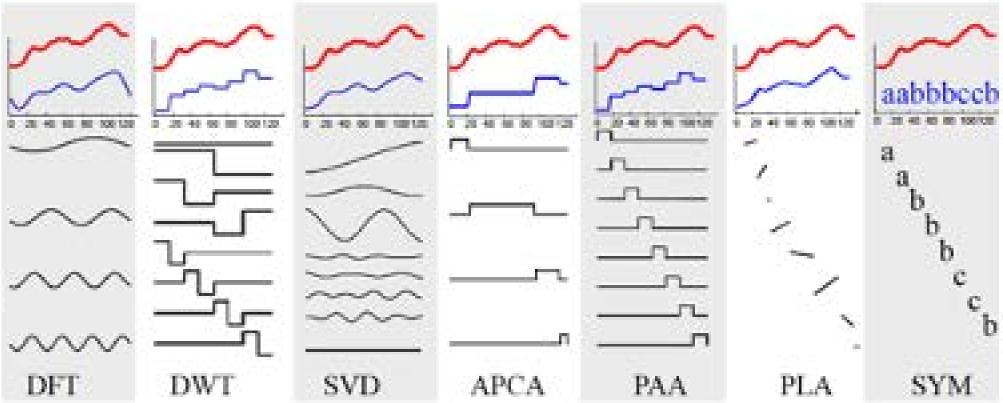


Discord / Abnormal sequence



Abnormal serie

TIME SERIES :



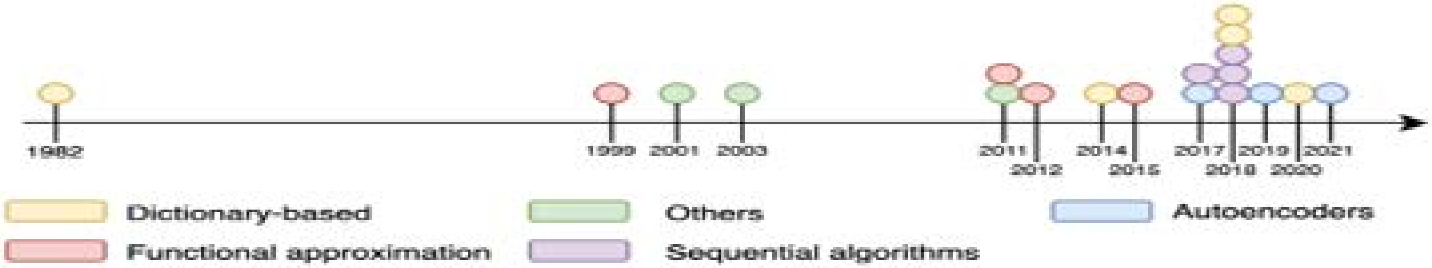
Cornuéjols, 2014

Definitions : Subsequence, DB, rep

- Subsequence :
  - Given a time series $T = (t_1, \ldots, t_n)$ of length n, a *subsequence* S of T is a series of length $m \leq n$ consisting of *contiguous time instants* from T,

$$S = (t_k, t_{k+1}, \ldots, t_{k+m-1}) \qquad \text{with } 1 \leq k \leq n-m+1$$

- Database :
  - A time series database DB is an **unordered** set of time series.
- Representation :
  - A representation of T is a model T' of reduced dimensionality n' (n' << n) such that T' closely approximates T.



Time Series Compression Survey

GIACOMO CHIAROT and CLAUDIO SILVESTRI, Department of Environmental Sciences, Informatics, and Statistics of Ca' Foscari University of Venice

Smart objects are increasingly widespread and their ecosystem, also known as the Internet of Things (IoT), is relevant in many application scenarios. The huge amount of temporally annotated data produced by these smart devices demands efficient techniques for the transfer and storage of time series data. Compression techniques play an important role toward this goal and, even though standard compression methods could be used with some benefit, there exist several ones that specifically address the case of time series by exploiting their peculiarities to achieve more effective compression and more accurate decompression in the case of lossy compression techniques. This article provides a state-of-the-art survey of the principal time series compression techniques, proposing a taxonomy to classify them considering their overall approach and their characteristics. Furthermore, we analyze the performances of the selected algorithms by discussing and comparing the experimental results that were provided in the original articles.

The goal of this article is to provide a comprehensive and homogeneous reconstruction of the state-of-the-art, which is currently fragmented across many articles that use different notations and where the proposed methods are not organized according to a classification.

CCS Concepts: • **Theory of computation → Data compression**; • **Information systems →** *Data streaming*;

Additional Key Words and Phrases: Time series, compression, streams

ACM Reference format:
Giacomo Chiarot and Claudio Silvestri. 2023. Time Series Compression Survey. *ACM Comput. Surv.* 55, 10, Article 198 (February 2023), 32 pages.
https://doi.org/10.1145/3560814

Another representation:

Matrix Profile

https://www.cs.ucr.edu/~eamonn/MatrixProfile.html

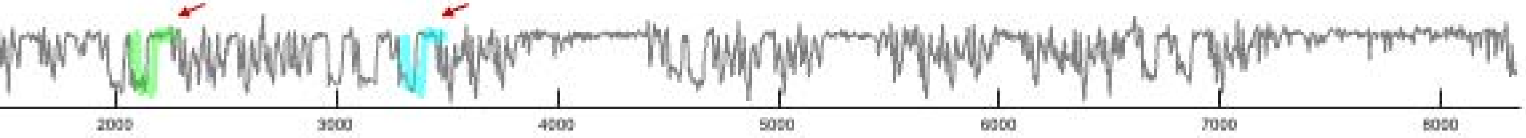
TIME SERIES : A COMPLEX DATA

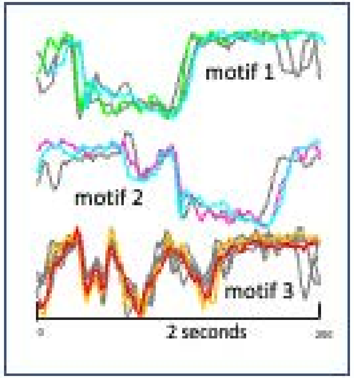
Definitions : Subsequence, DB, representation, similarity

- (Dis)Similarity :
  - The **(dis)similarity measure** Sim(T, U) between time series T and U is a function taking two time series as inputs and returning the distance between these series.
  - Example : Euclidian distance, DTW, …
  - Properties : nonnegative, symmetry, subadditivity (triangle inequality)
  - The *subsequence (dis)similarity measure* $\mathrm{Sim}_{subseq}(T, S)$ represents the distance between T and its <u>best matching location</u> in S, i.e. :

$$\mathrm{Sim}_{subseq}(T, S) = \min\left(\mathrm{Sim}(T, S')\right) \text{ for } S' \text{ a subsequence of size } |T| \text{ in } S$$

Fundamental Assumption: *Conservation is Key*





From Keog's slides : https://www.cs.ucr.edu/~eamonn/MatrixProfile.html

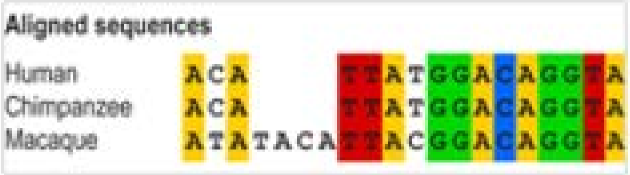
If a pattern is *conserved*, there must be some mechanism that conserves it. This is true in linguistics, music, genetics, literature, religions….

Much of Keog's work asks *what* is conserved in time series, *when* is it conserved, and *why* was an expected conservation not observed…

For discrete strings, *conserved* is easy to define, for example $papa = *a*a$. For *time series* it requires a distance function, e.g. : Euclidean Distance.

* Bengali: Bābā
* Mandarin : baba
* Polish : tata
* Swahili : baba
* Turkish : baba
* Xhosa: -tata
* Norwegian : papa
* Spanish : papá
* Swahili : baba
* English : papa
* Hindi : papa
* Indonesian : bapa

en.wikipedia.org/wiki/Mama_and_papa



What is the Matrix Profile?

- The Matrix Profile (MP) is a data structure that annotates a time series.
- **Key Claim:** Given the MP, most time series data mining problems are trivial or easy!
- Example : motif discovery, density estimation, anomaly detection, rule discovery, joins, segmentation, clustering etc. However, *you* can use the MP to solve *your* problems, or to solve a problem listed above, but in a different way, tailored to your interests/domain.
- **Key Insight**: The MP profile has many highly desirable properties, and any algorithm you build on top of it, will inherit those properties.
  - Say you use the MP to create: *An Algorithm to Segment Sleep States*
  - Then, *for free*, you have also created: *An Anytime Algorithm to Segment Sleep States*

    An Online Algorithm to Segment Sleep States

    A Parallelizable Algorithm to Segment Sleep States

    A GPU Accelerated Algorithm to Segment Sleep States

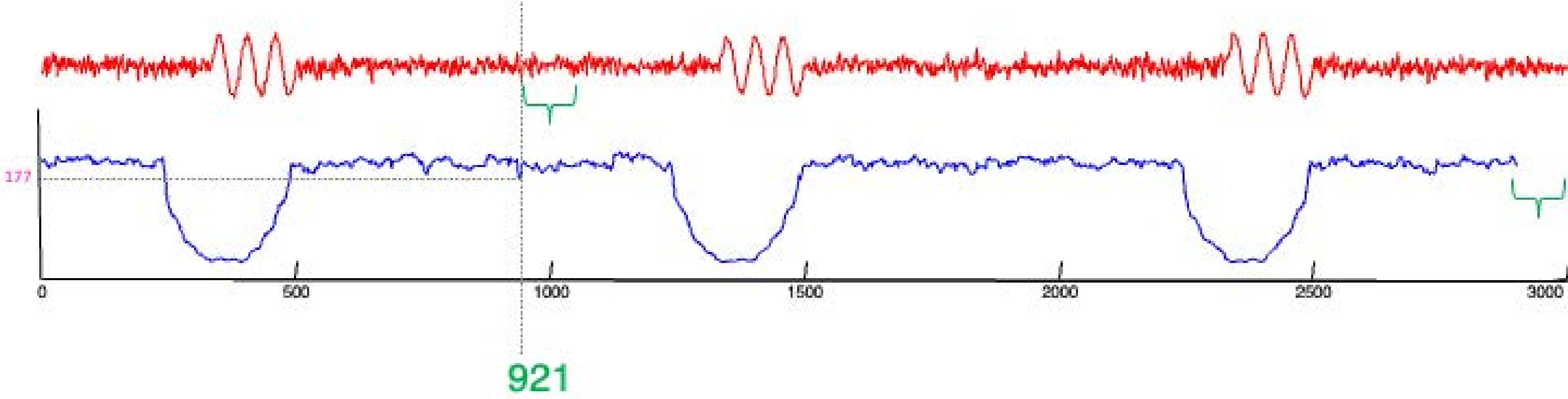
    An Algorithm to Segment Sleep States with Missing Data

    etc.

We can created a companion "time series", called a Matrix Profile or MP.

The matrix profile at the i^{th} location records the distance of the subsequence in T, at the i^{th} location, to its nearest neighbor under z-normalized Euclidean Distance.

For example, in the below, the subsequence starting at 921 happens to have a distance of 177.0 to its nearest neighbor (wherever it is).

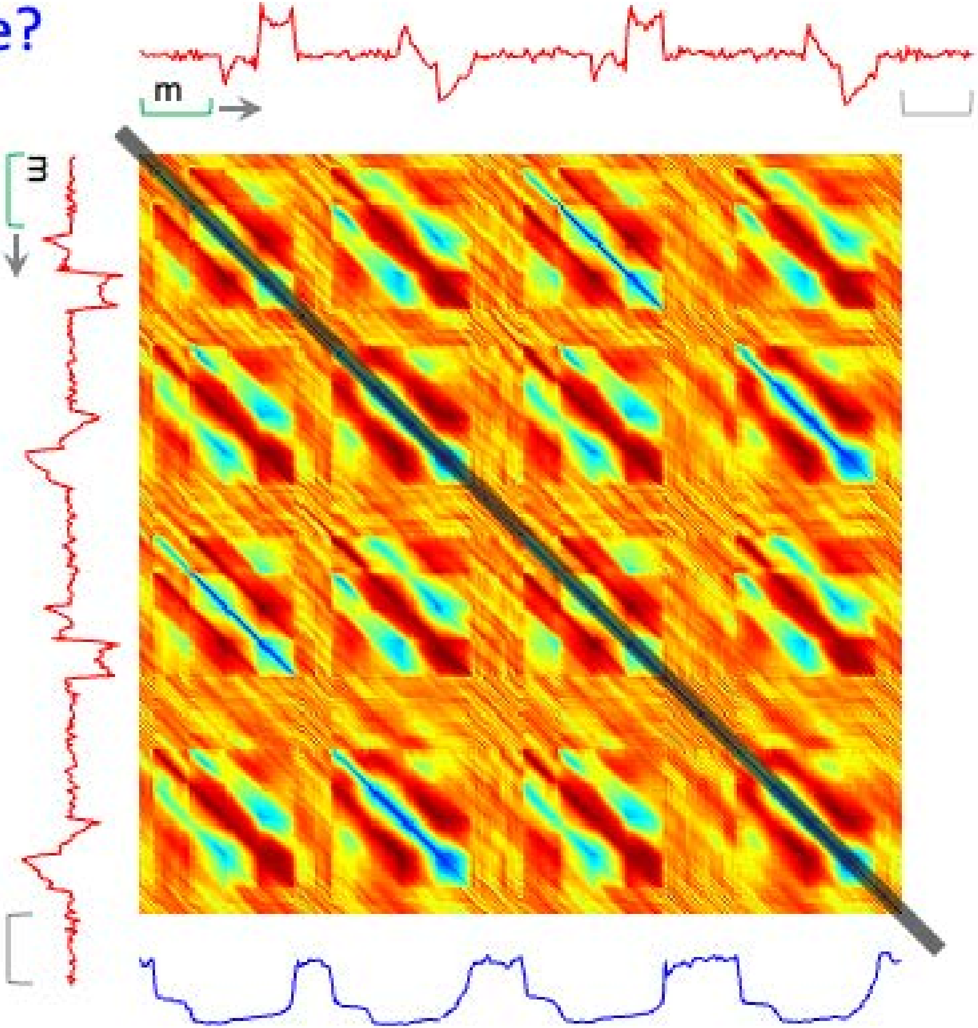


Why is it called the Matrix Profile?

One naïve way to compute it would be to construct a distance matrix of all pairs of subsequences of length *m*.

For each column, “project” down the smallest (*non diagonal*) value to a vector : Matrix Profile.

While in general we could never afford the memory to do this (4TB for just |T|= one million), for most applications the Matrix Profile is the *only* thing we need from the full matrix, and we can compute and store it very efficiently.

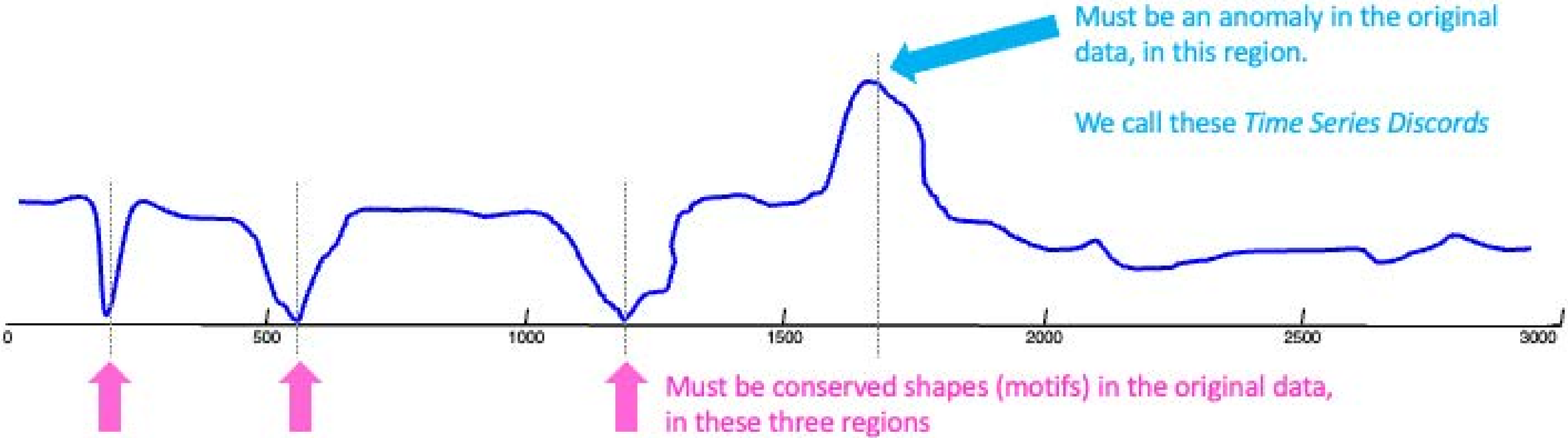


Key:
Small distances are blue
Large distances are red
Dark stripe is excluded

How to "read" a Matrix Profile

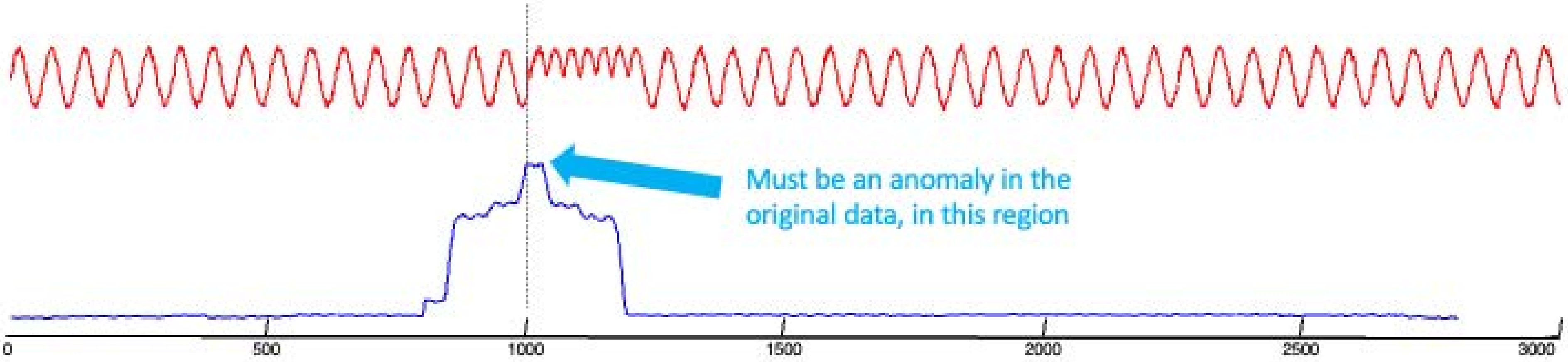
Where you see relatively low values, you know that the subsequence in the original time series must have (at least one) relatively similar subsequence elsewhere in the data (such regions are "motifs" or reoccurring patterns)

Where you see relatively high values, you know that the subsequence in the original time series must be unique in its shape (such areas are "discords" or anomalies).



How to "read" a Matrix Profile: Synthetic Anomaly Example

Where you see relatively high values, you know that the subsequence in the original time series must be unique in its shape. In fact, the highest point is *exactly* the definition of Time Series Discord, perhaps the best anomaly detector for time series*

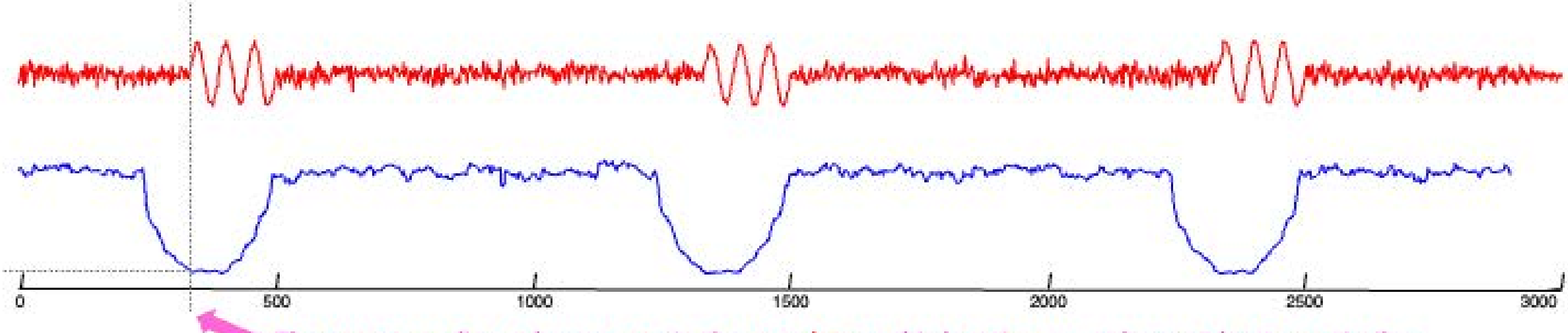


* Vipin Kumar performed an extensive empirical evaluation and noted that "..on 19 different publicly available data sets, comparing 9 different techniques (time series discords) is the best overall technique.". V. Chandola, D. Cheboli, V. Kumar. Detecting Anomalies in a Time Series Database. UMN TR09-004

How to "read" a Matrix Profile: Synthetic Motif Example

Where you see relatively low values, you know that the subsequence in the original time series must have (at least one) relatively similar subsequence elsewhere in the data.

In fact, the lowest points must be a tieing pair, and correspond exactly to the classic definition of *time series motifs*.



The corresponding subsequence in the raw data at this location, must have at *least one* similar subsequence somewhere

Content

- Basic notions
- Anomaly Detection on data stream
- Dragstream
- Challenges

Anomaly Detection on data stream

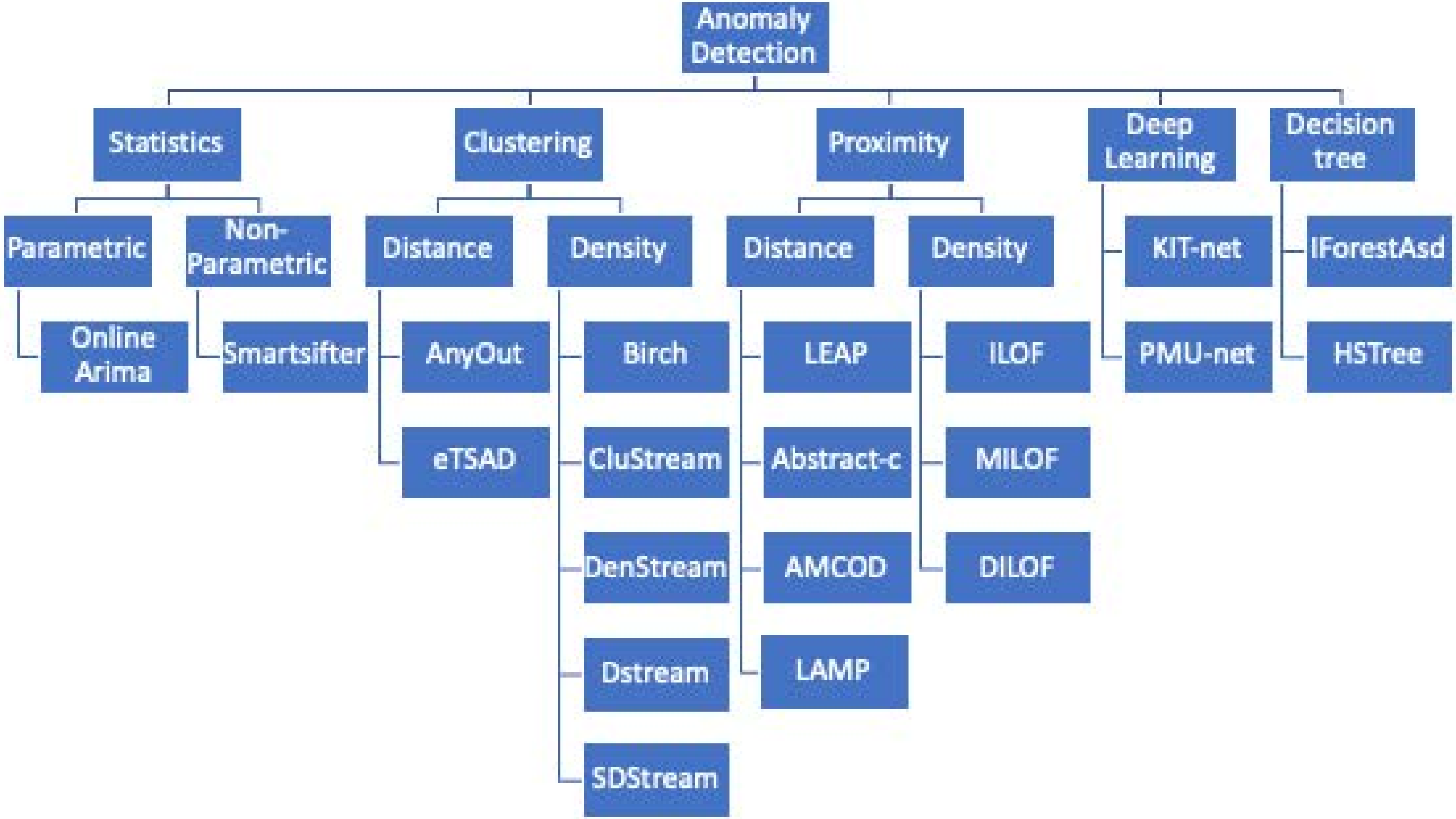
Models must be capable of :
- incorporating new information at the *speed data arrives*;
- *detecting changes* and adapting the models to the *most recent* information.
- *forgetting outdated* information;

Well-established literature on sub-sequence anomaly detection in **time series** but only few works for **data streams**.

Examples of data stream anomaly detection approaches :

- LAMP (Zimmerman et al, 2019): inspired from Matrix Profile; high complexity and retraining needed when concept drifts occur.
- HS-Squeezer-stream (Chau et al, 2018) inspired from Hotsax (Keogh et al, 2007)
- SAND (Boniol et al, 2021) inspired from DTW
- ....

Methods (non exhaustive)



Methods

Category	Process and assumption		
→ Deep learning	train the model →	predict an instance value →	anomaly = instance with a great error
→ Statistics	find normal values domain →		anomaly = instance out of the domain
→ Proximity	identify nearest neighbors →		anomaly = instance far from its neighbors
→ Tree	train tree based models →		anomaly = instances early isolated

The VLDB Journal
https://doi.org/10.1007/s00778-022-00773-x

REGULAR PAPER

A meta-level analysis of online anomaly detectors

Antonios Ntroumpogiannis[1] · Michail Giannoulis[2] · Nikolaos Myrtakis[1,3] · Vassilis Christophides[3] · Eric Simon[4] · Ioannis Tsamardinos[1]

Received: 31 December 2021 / Revised: 29 November 2022 / Accepted: 4 December 2022

Abstract

Real-time detection of anomalies in streaming data is receiving increasing attention as it allows us to raise alerts, predict faults, and detect intrusions or threats across industries. Yet, little attention has been given to compare the effectiveness and efficiency of anomaly detectors for streaming data (*i.e.*, of online algorithms). In this paper, we present a qualitative, synthetic overview of major online detectors from different algorithmic families (*i.e.*, distance, density, tree or projection based) and highlight their main ideas for constructing, updating and testing detection models. Then, we provide a thorough analysis of the results of a quantitative experimental evaluation of online detection algorithms along with their offline counterparts. The behavior of the detectors is correlated with the characteristics of different datasets (*i.e.*, meta-features), thereby providing a meta-level analysis of their performance. Our study addresses several missing insights from the literature such as (a) how reliable are detectors against a random classifier and what dataset characteristics make them perform randomly; (b) to what extent online detectors approximate the performance of offline counterparts; (c) which sketch strategy and update primitives of detectors are best to detect anomalies visible only within a feature subspace of a dataset; (d) what are the trade-offs between the effectiveness and the efficiency of detectors belonging to different algorithmic families; (*e*) which specific characteristics of datasets yield an online algorithm to outperform all others.

Keywords Anomaly detection · Online algorithms · Performance evaluation · Meta-learning

Multivariate

Comparison 9 methods for multivariate data : **distance-based** (MCOD, CPOD), **KNN-based** (LEAP, KNN_W), **density-based** detectors (STARE, RS-Hash, LOF), **tree-based** (HST/F, RRCF, IF, OCRF) and **projection-based** detectors (XSTREAM, LODA).

- Assess the reliability of detectors' effectiveness against a random classifier, and highlight the *dataset characteristics*

- Indicate when online detectors can approximate the effectiveness of offline detectors and under which conditions

- Indicate which is the best sketch strategy and update primitives of detectors

- analyze the trade-offs between the effectiveness and the efficiency of detectors belonging to different algorithmic families

- highlight the characteristics of datasets that make an online algorithm capable of outperforming all others

Content

- Basic notions
- Anomaly Detection on data stream
- DragStream
- Challenges

DragStream

Drag :

- Time series approach
- Inspired variable length discord detection in time series : MERLIN (Nakamura et al., 2020)
- Linear time and space complexity, at worst O(n*n)

➔ Discord set initialized with the first subsequence

➔ Two subsequences are close if their distance is lower than a predefined threshold **r**

➔ Two steps : Discord selection and Discord refinement

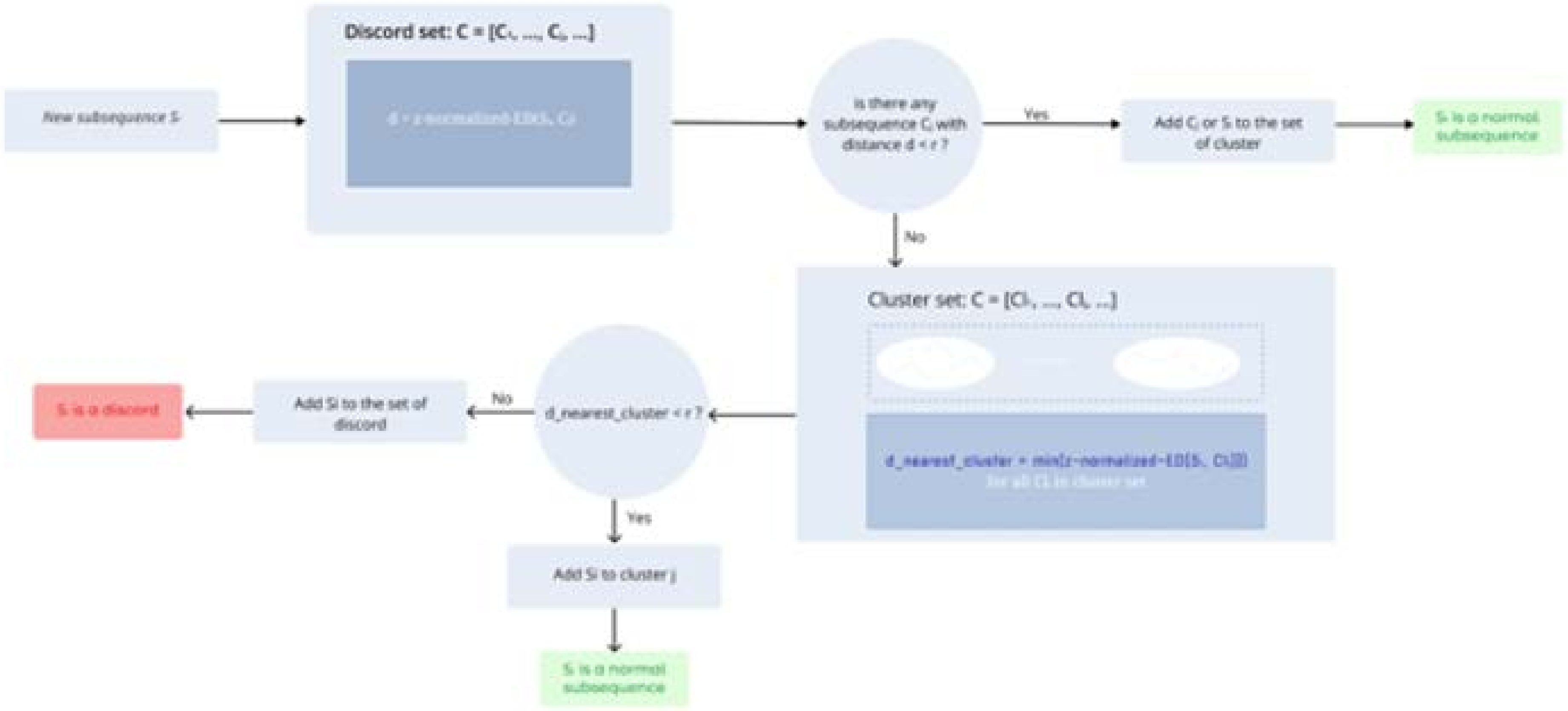
Adaptation :

- Summarize data on **limited memory**
- Be aware of **concept drift**

➔ Cluster subsequences as some abnormal point detection do for their past points

- example : Memory Efficient Local Outlier Factor (MILOF) (Salehi et al., 2016)
- Delete Inactive clusters which could represent past trend

DragStream



New subsequence S
Discord set: C = [C1, ..., Cj, ...]
Is there any subsequence Cj with distance d < r ?
Yes
Add Cj or Si to the set of cluster
Si is a normal subsequence
No
Cluster set: C = [Cl1, ..., Clj, ...]
d_nearest_cluster = min(z-normalized-ED(Si, Clj))
d_nearest_cluster < r ?
No
Add Si to the set of discord
Si is a discord
Yes
Add Si to cluster j
Si is a normal subsequence

DragStream

- We compared DragStream to:
  - Matrix Profile (Yeh et al., 2016): a SOTA discord detection approach for time series.
  - LAMP (Zimmerman et al, 2019) : A stream approach proposed for adapting Matrix Profile to streaming data

- **Datasets :**
  - (12) ECG, GPS, energy consumption datasets for discord detection from

(Bonio et al., ICDE,2020), (Keogh et al., ICDM, 2005), (Chi et al. SOICT, 2018), (Senin et al., EDBT, 2015)

- **Scores :**
  - Discord detection: F1-score over the **overlapping rate**=|Discord∩GroundTruth|/|Discord|

(Li et al., SIAM,2020), (Senin et al.,EDBT, 2015)

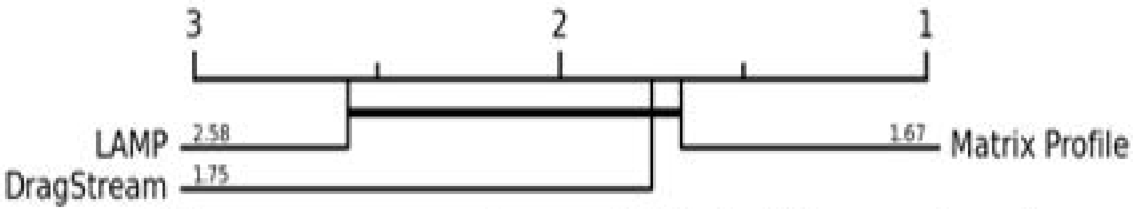
- For each Method
  - For each dataset
    - Optimize hyperparameters of the method over 30 iterations **(Bayesian optimization)**
    - Record the score of the method and its time computation

DragStream

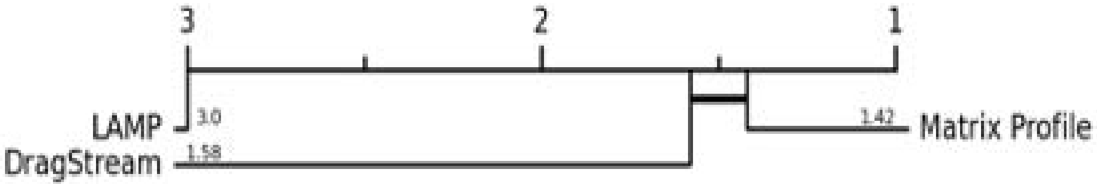
Dataset	Drag-stream	LAMP
stdb_308_1	0.19	**0.22**
xmitdb_x108_1	**0.24**	0
mitdb__100_180_1	**0.5**	0
chfdb_chf01_275_1	**0.5**	0.09
ltstdb_20221_43_1	**0.4**	0.1
mitdbx_108	**0.48**	0.285
qtdbsele0606	0.01	**0.55**
chfdbchf15	**0.5**	0.067
ann-gun	**0.36**	0.26
patient respiration	**0.67**	0.24
dutch power demand	**0.56**	0.1639
gps trajectory	**0.286**	0

Matrix Profile (Time series)
0.069
0.554
0.5468
0.63
0.415
0.821
0.005
0.81
0.026
0.46
0.75
0.08

DragStream



F1 scores comparison at 5% significance level



Time execution comparison at 5% significance level

DragStream

- DragStream, a new sub-sequence anomaly detection in data-streams for univariate data
- It extends Drag and borrows ideas from MILOF and Matrix Profile
- Experimental comparisons show that DragStream can be a competitive method compared to existing methods in the literature.

Advantages:

- Simple and easily interpretable
- Possibility to detect similar anomalies with clusters having few instances

Limitations:

- Setting optimal values for **r**
- More comparisons needed with existing methods.

Perspectives:

- Multivariate
- Variable length discord detection on data stream

Content

- Basic notions
- Anomaly Detection on data stream
- Dragstream
- Challenges

Open challenges in Anomaly detection

- Explainability on the fly
- Semi-supervised
- Multi-target, multi-task and transfer learning
- Distributed Streams
- Representation learning
- Concept drift
- Visualization
- Ease of use

ORIENTATIONS

KD of Complex Data
driven by Domain-Knowledge

1926

Metro of London

1933

ACM Communications, April 2007
"Is abstraction the key to computing"
Jeff Kramer (Imperial College L.)

Some references @ LIMOS:

- A. Ntroumpogiannis, M. Giannoulis, N. Myrtakis, V. Christophides, E. Simon, I. Tsamardinos. **A meta-level analysis of online anomaly detectors**. *The VLDB Journal* (2023). https://doi.org/10.1007/s00778-022-00773-x, Springer
- Nicolas Wagner, Violaine Antoine, Jonas Koko, Marie-Madeleine Mialon, Romain Lardy, Isabelle Veissier: **Comparison of Machine Learning Methods to Detect Anomalies in the Activity of Dairy Cows**. ISMIS 2020: 342-351
- A. M. S. N. Bibinbe, A. J. Djiberou Mahamadou, M. F. Mbouopda and E. Mephu Nguifo, "**DragStream: An Anomaly And Concept Drift Detector In Univariate Data Streams**" 2022 IEEE ICDM Workshops (ICDMW), Orlando, FL, USA, 2022, pp. 842-851, doi: 10.1109/ICDMW58026.2022.00113.
- A. M. S. Ngo Bibinbe, M. F. Mbouopda, G. R. Mbiadou Saleu and E. Mephu Nguifo, "**A survey on unsupervised learning algorithms for detecting abnormal points in streaming data**" 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 2022, pp. 1-8, doi: 10.1109/IJCNN55064.2022.9892195.

Thanks

- Issam Falih, Violaine Antoine
- Raissa Mbiadou, Florentin Jiechieu
- Abdoul J. Djiberou Mahamadou, Michael F. Mbouopda, Rim El Cheikh, Durande Kamga

- Anne M. S. Ngo Bibinbe, Kevin Nguetche, Rayane Bakari, Anthony Bertrand

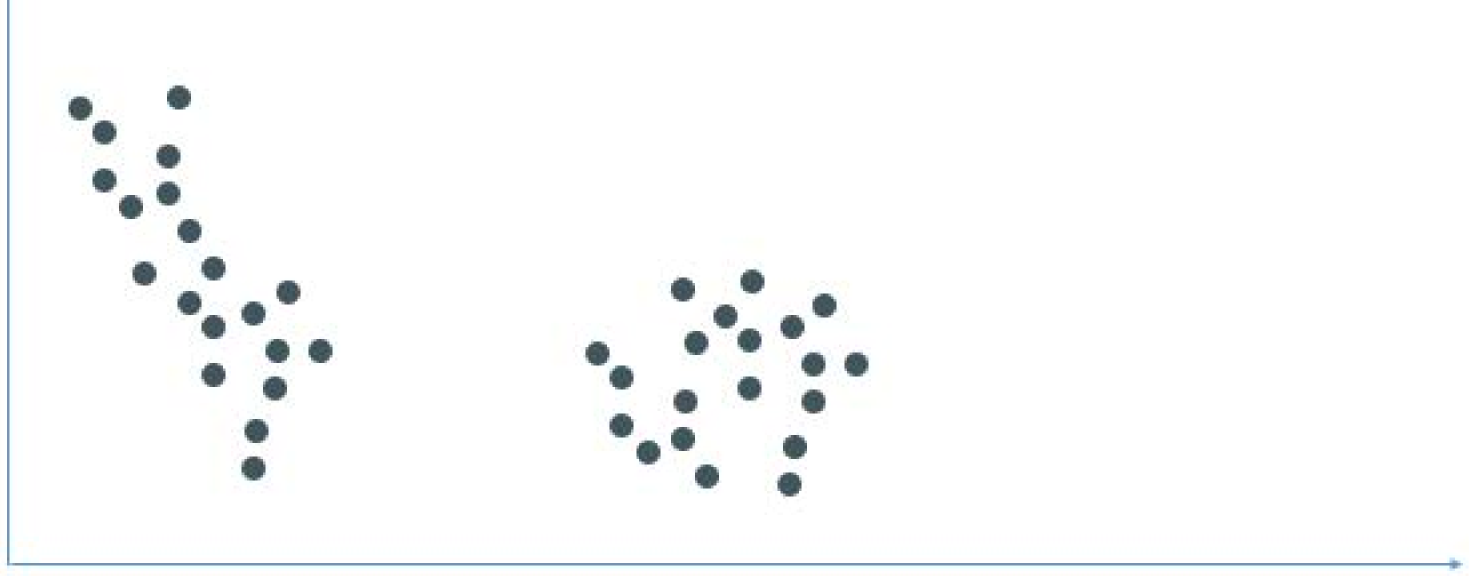
- Miners group : http://miners.limos.fr

- Industrial partner : Pfeiffer Vacuum, BPI

Questions

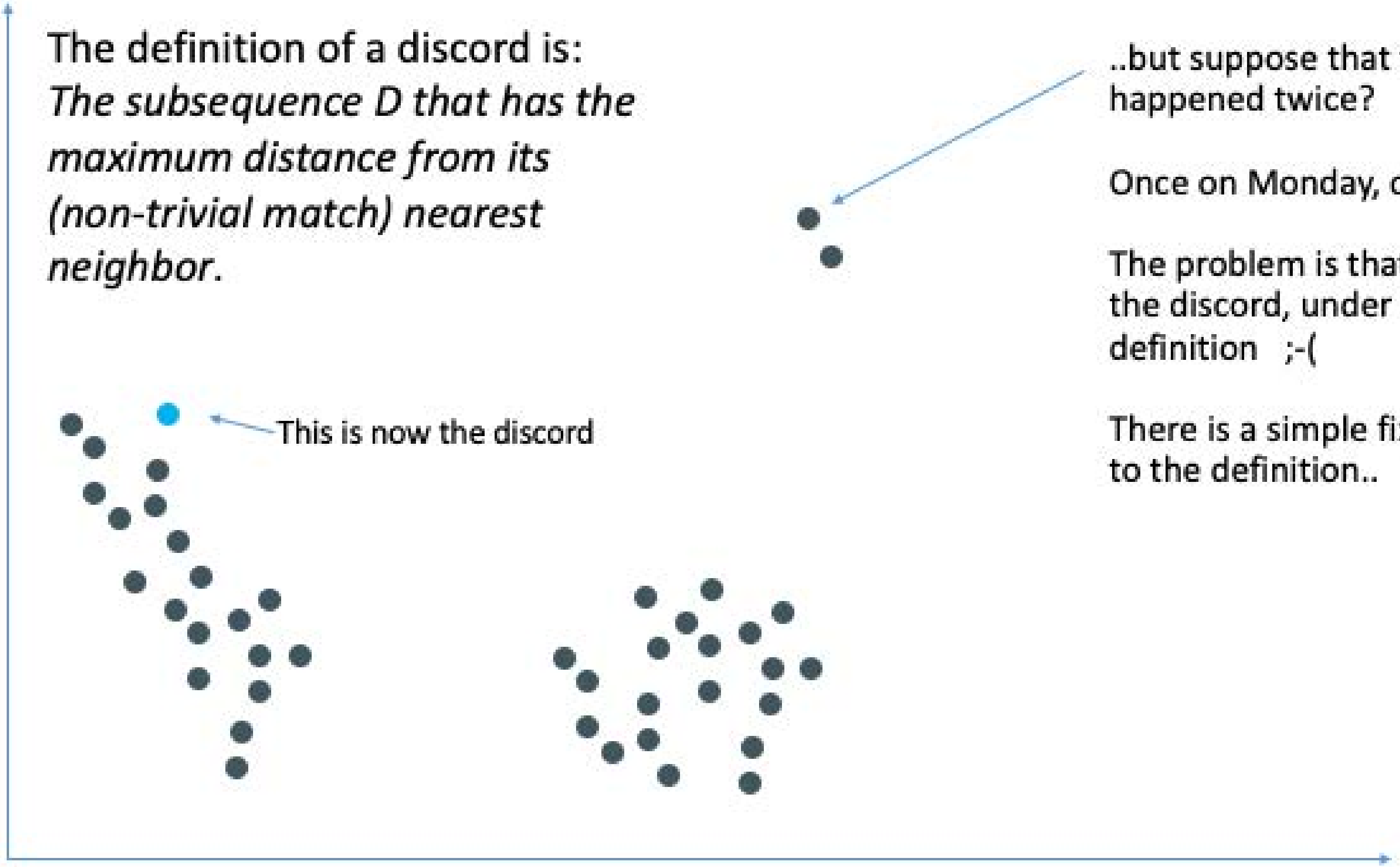
The twin freak problem (see next slide)

The definition of a discord is:
The subsequence D that has the maximum distance from its (non-trivial match) nearest neighbor.



The twin freak problem

The definition of a discord is:
The subsequence D that has the maximum distance from its (non-trivial match) nearest neighbor.



..but suppose that the anomaly happened twice?

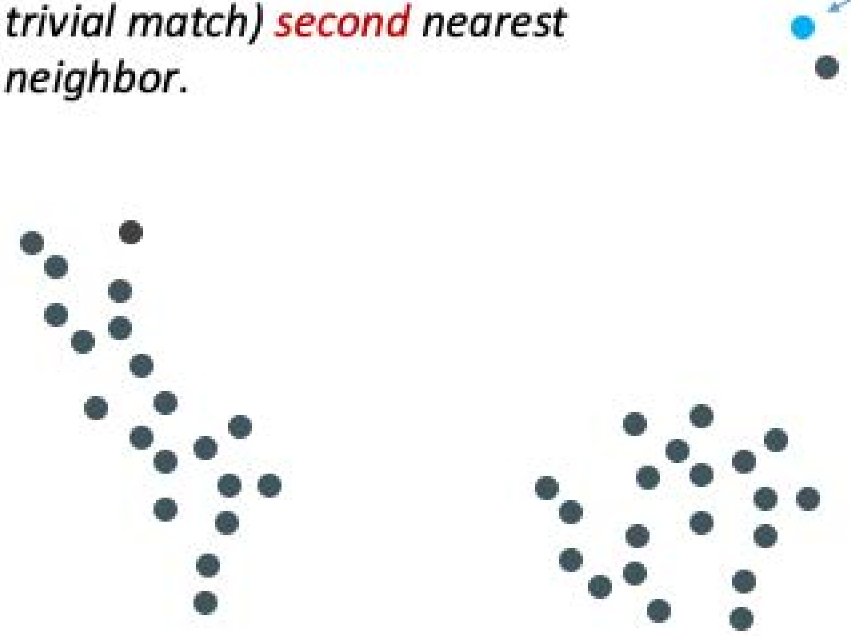
Once on Monday, once on Friday…

The problem is that it is no longer the discord, under our classic definition ;-(

There is a simple fix, a minor change to the definition..

The twin freak problem

The new definition of a discord is:
The subsequence D that has the maximum distance from its (non-trivial match) second nearest neighbor.



The new definition solves the problem.

However, what about the *triple* freak, or *quadruple* freak problem etc….

If an “anomaly” happens many times, it is probably not an anomaly, and we probably know about it anyway.

Nevertheless, it can be useful to generalize to the Kth nearest neighbor, for a small K, say 3

The subsequence D that has the maximum distance from its (non-trivial match) K nearest neighbor.

This is a trivial change/addition to the MP