



Time Series Anomaly Detection: *Foundations and Practices*

Paul Boniol
Inria, ENS, PSL University
Paul.boniol@inria.fr

Inria



Paul Boniol

*I am a researcher at Inria, a member of the VALDA project team, a joint team between **Inria** Paris, **École Normale Supérieure**, and **CNRS**.*

My research interest lies in the intersections between:

- **Massive time series** analytics and management systems.
- Unsupervised and supervised **anomaly detection** methods for large time series.
 - **Machine learning** for time series analytics.



paul.boniol@inria.fr

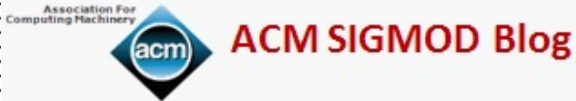


<https://boniopl.github.io/>

Inria



Paul Boniol



Paul Boniol -
Themis Palpanas

JULY 16, 2024

TIME SERIES ANOMALY DETECTION

≡ Time Series

What it is, how it works, where we are, and where we are heading

Anomaly detection is an important problem in data analytics with applications in many domains. In recent years, there has been an increasing interest in anomaly detection tasks applied to time series. In this post, we take a holistic view on anomaly detection in time series, starting from the core definitions and taxonomies related to time series and anomaly types, to a brief description of the anomaly detection methods proposed by different communities in the literature. We conclude with the recent benchmark and evaluation effort proposed recently in the data management community, and new perspectives and research direction.

Inria



Introduction: *Time series are Everywhere*

Energy Production



Edf.fr: tinyurl.com/yc7x5xje

Astrophysics



Virgo: <https://www.virgo-gw.eu/>

Medicine



tinyurl.com/39dx2us4

Volcanology

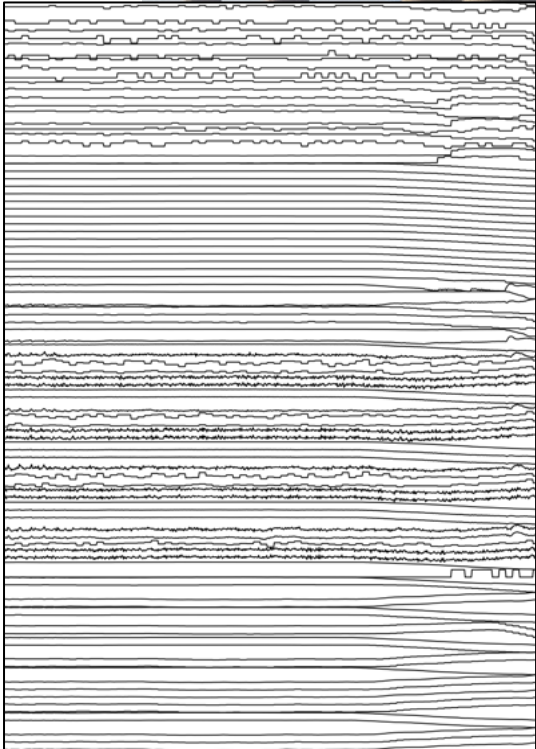


tinyurl.com/ybcttmfz

Introduction: *Time series are Everywhere*

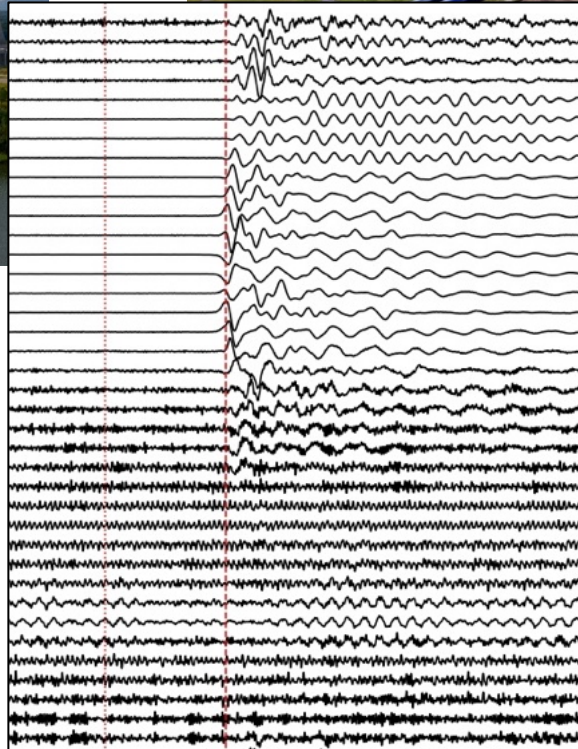
Energy Production

Secondary circuit sensor measurements



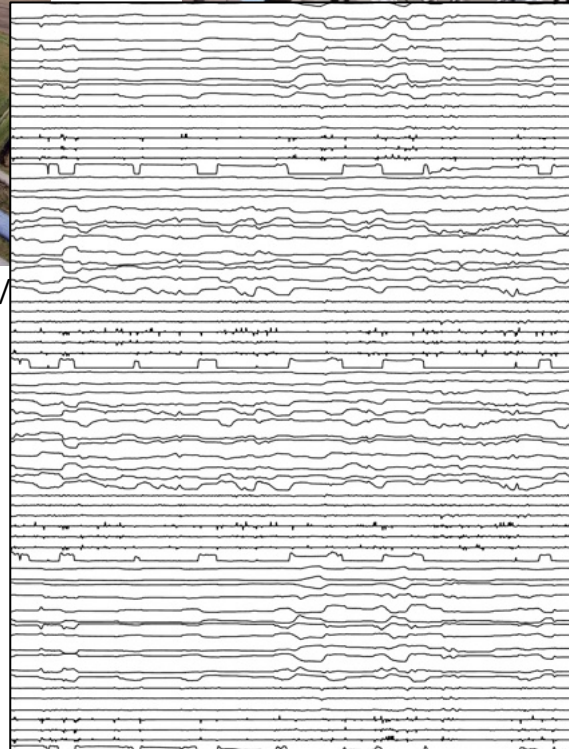
Astrophysics

Fiber-acoustic sensors in the VIRGO north building



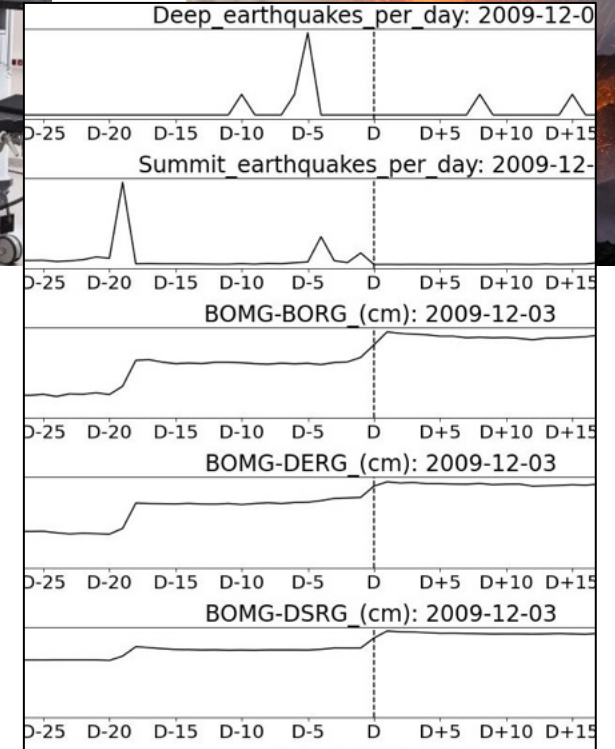
Medicine

Sensor measurements of the Da Vinci surgery robot



Volcanology

Sensor measurements on le Piton de la Fournaise

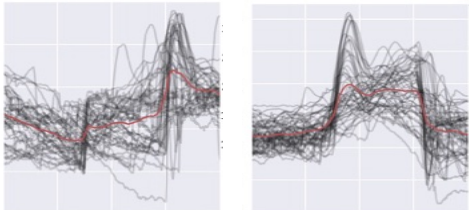


Introduction: *with Important Challenges*

Energy Production

Secondary circuit sensor measurements

Identification of precursors of feed-water pumps vibrations



Astrophysics

Fiber-acoustic sensors in the VIRGO north building

Noise detection in VIRGO interferometer north building



Medicine

Sensor measurements of the Da Vinci surgery robot

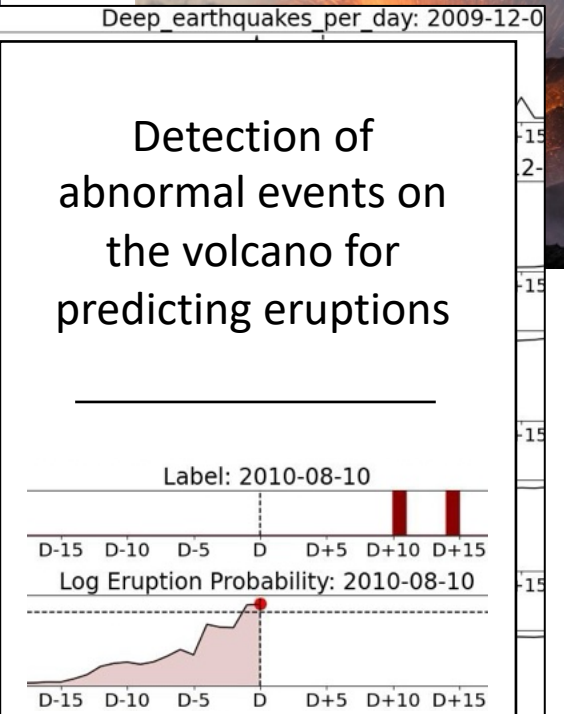
Unusual surgeons gestures detection



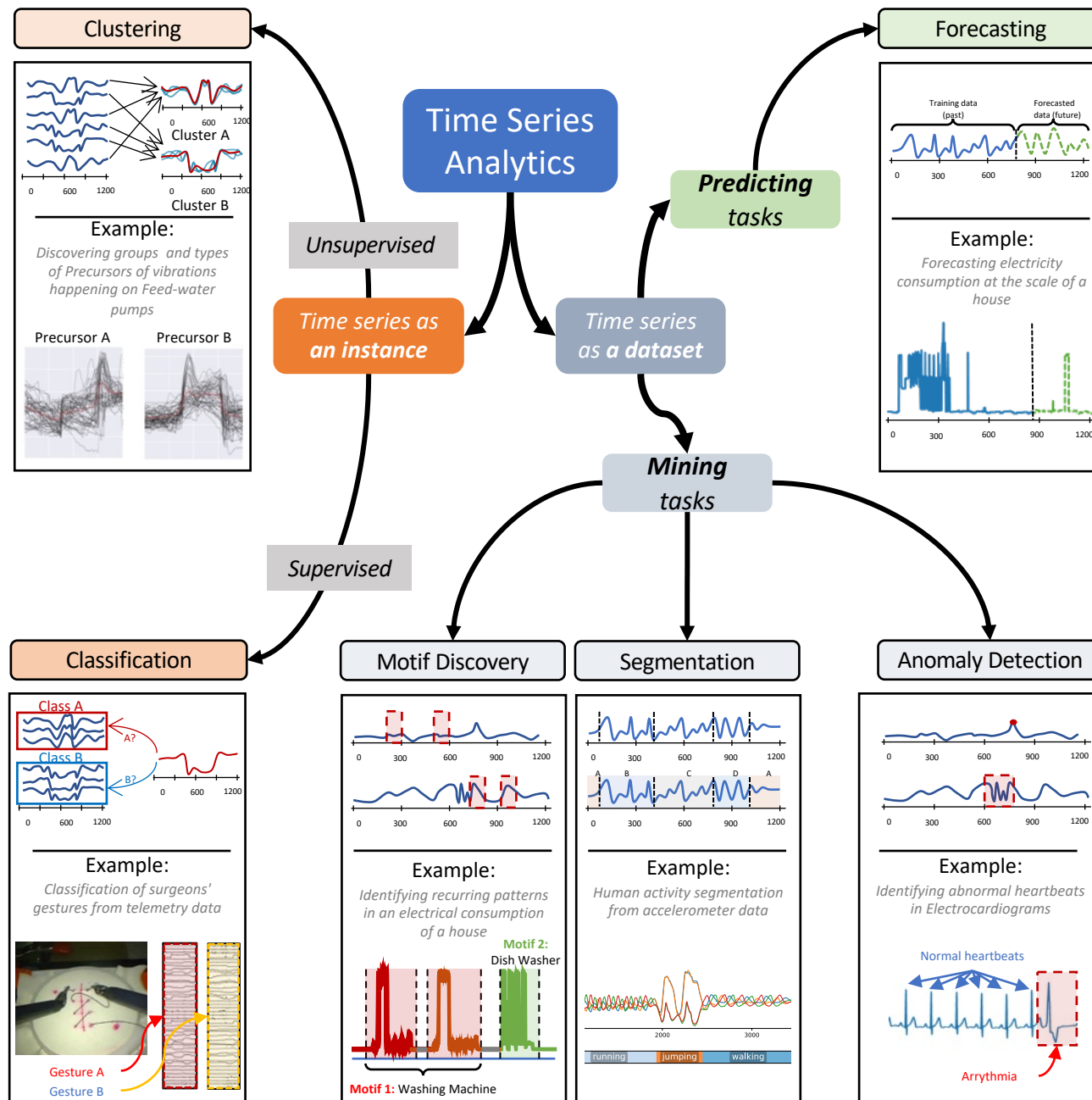
Volcanology

Sensor measurements on le Piton de la Fournaise

Detection of abnormal events on the volcano for predicting eruptions

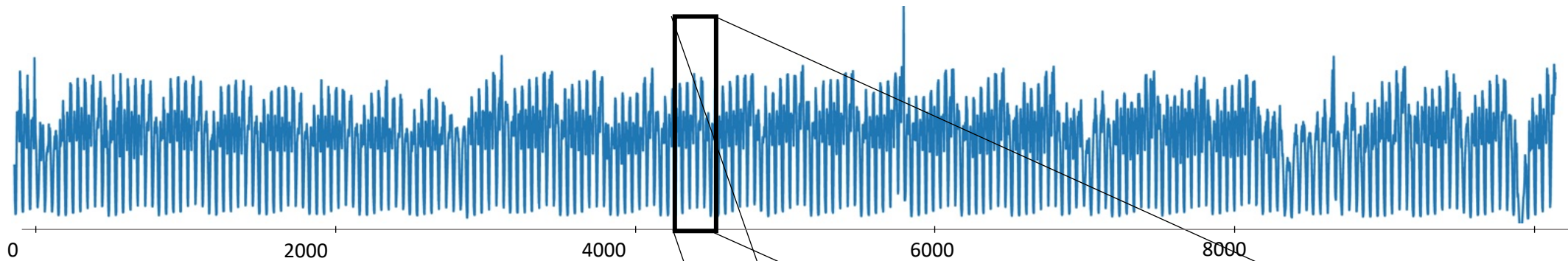


Introduction: Time Series Analytics

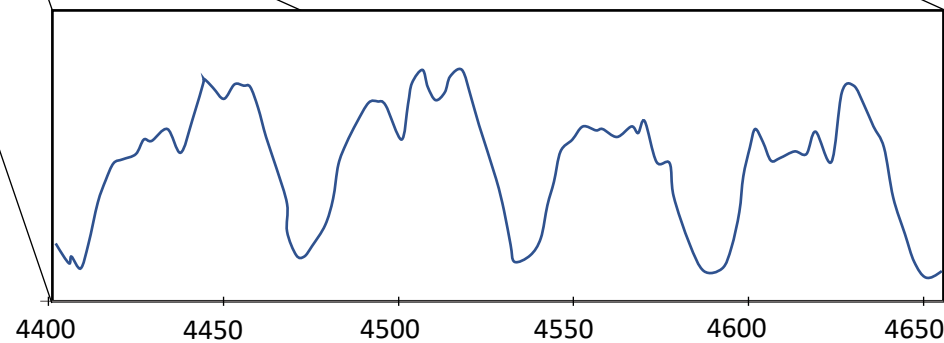


Introduction: *Anomaly Detection in Time Series*

- Time series T (example : number of taxi passengers in New York City)

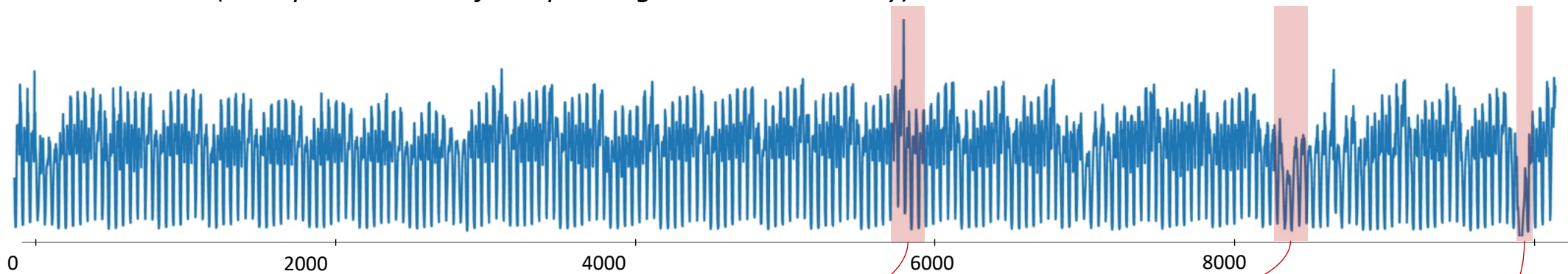


- Subsequence $T_{i,\ell}$
with $i = 4400, \ell = 250$

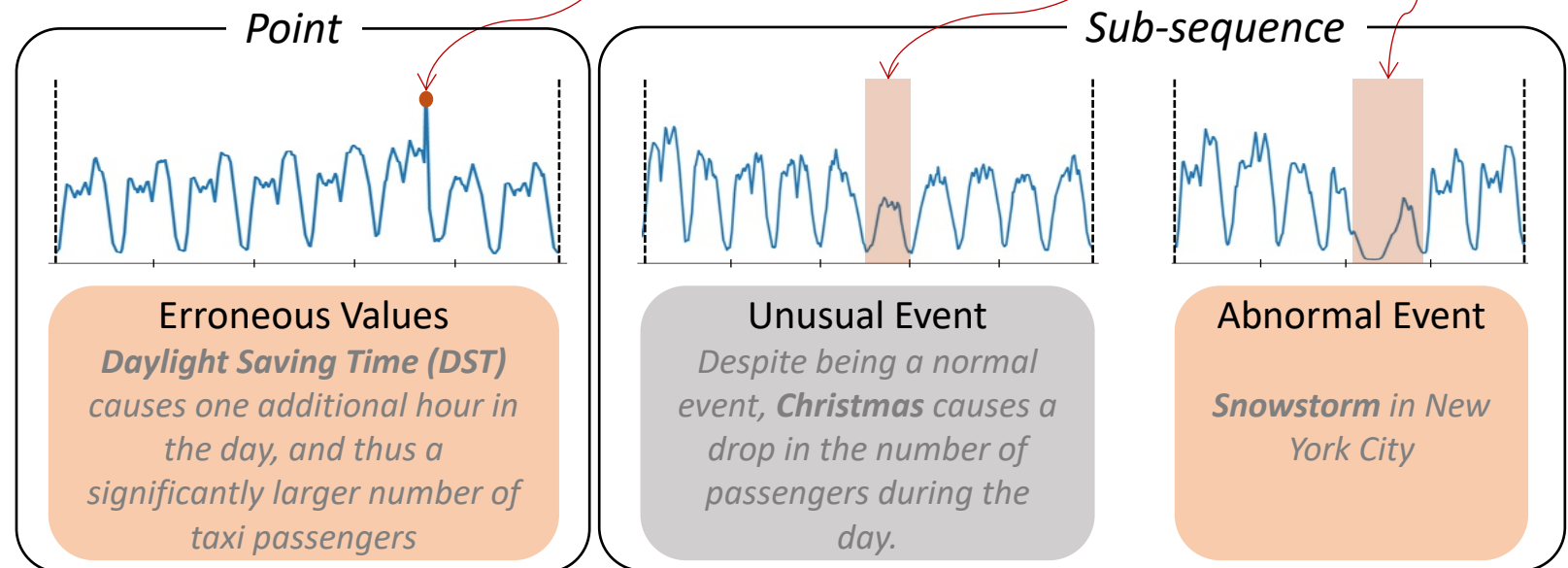


Introduction: *Anomaly Detection in Time Series*

- Time series T (example : number of taxi passengers in New York City)

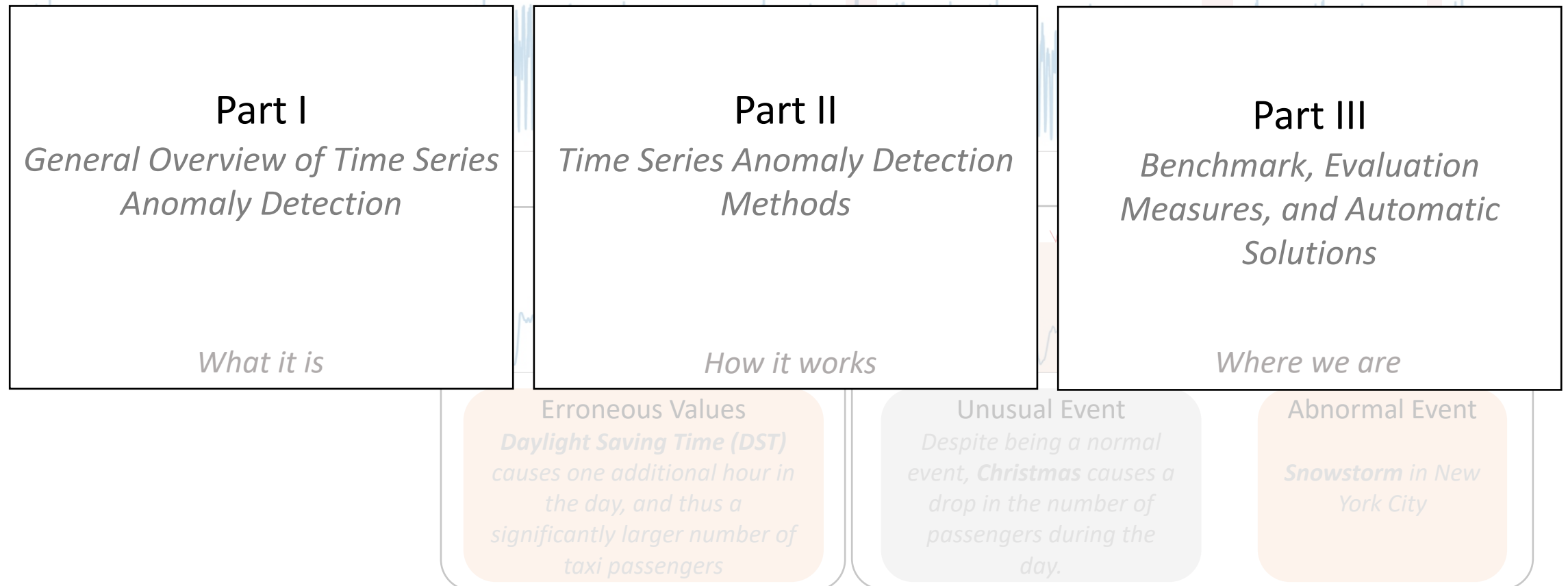


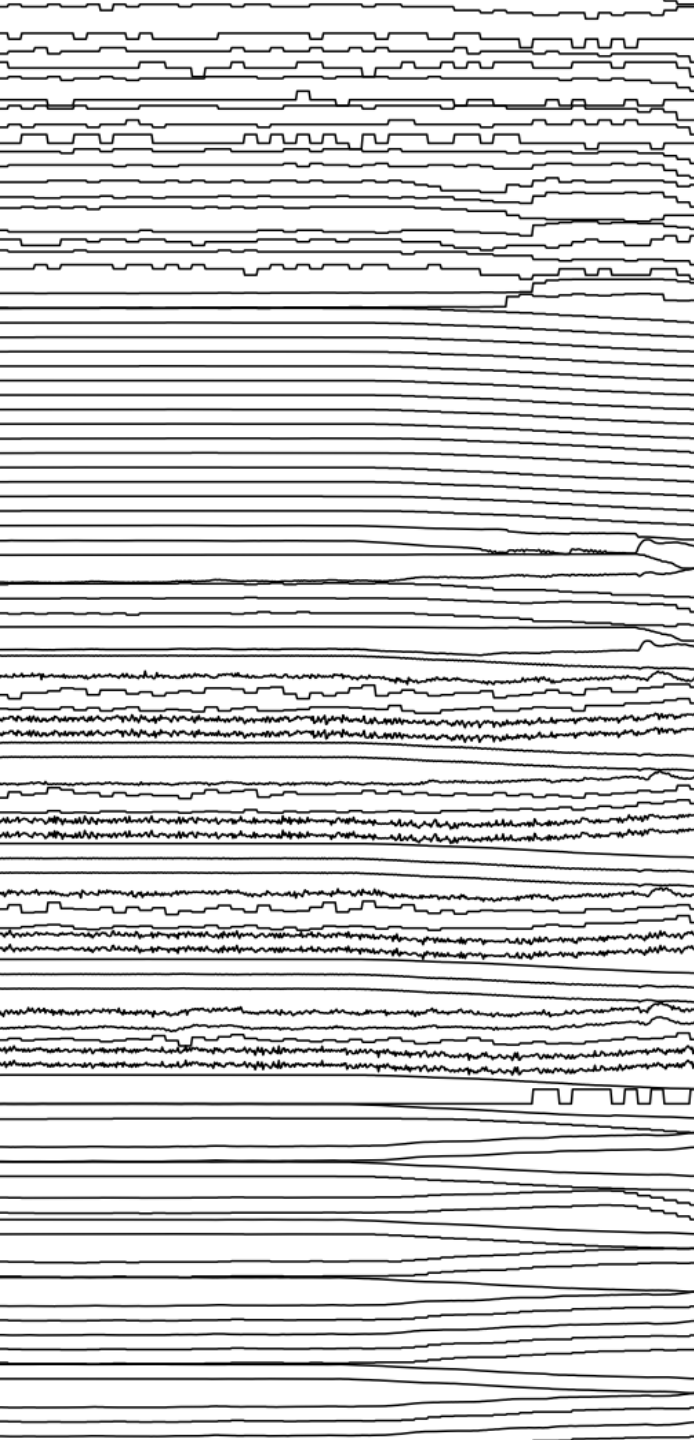
- Anomaly: *rare* point or sequence (of a given length) potentially *non-desired*



Introduction: *Anomaly Detection in Time Series*

- Time series T (example : number of taxi passengers in New York City)

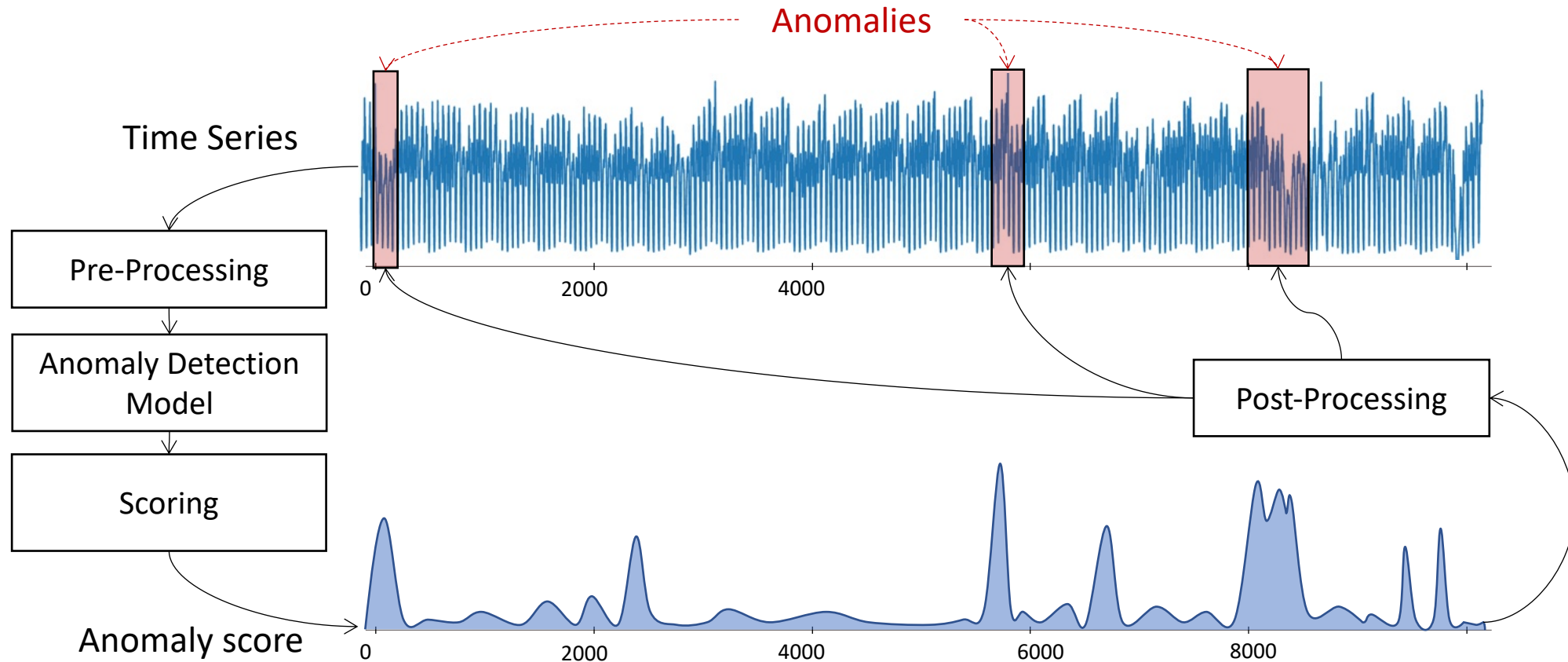




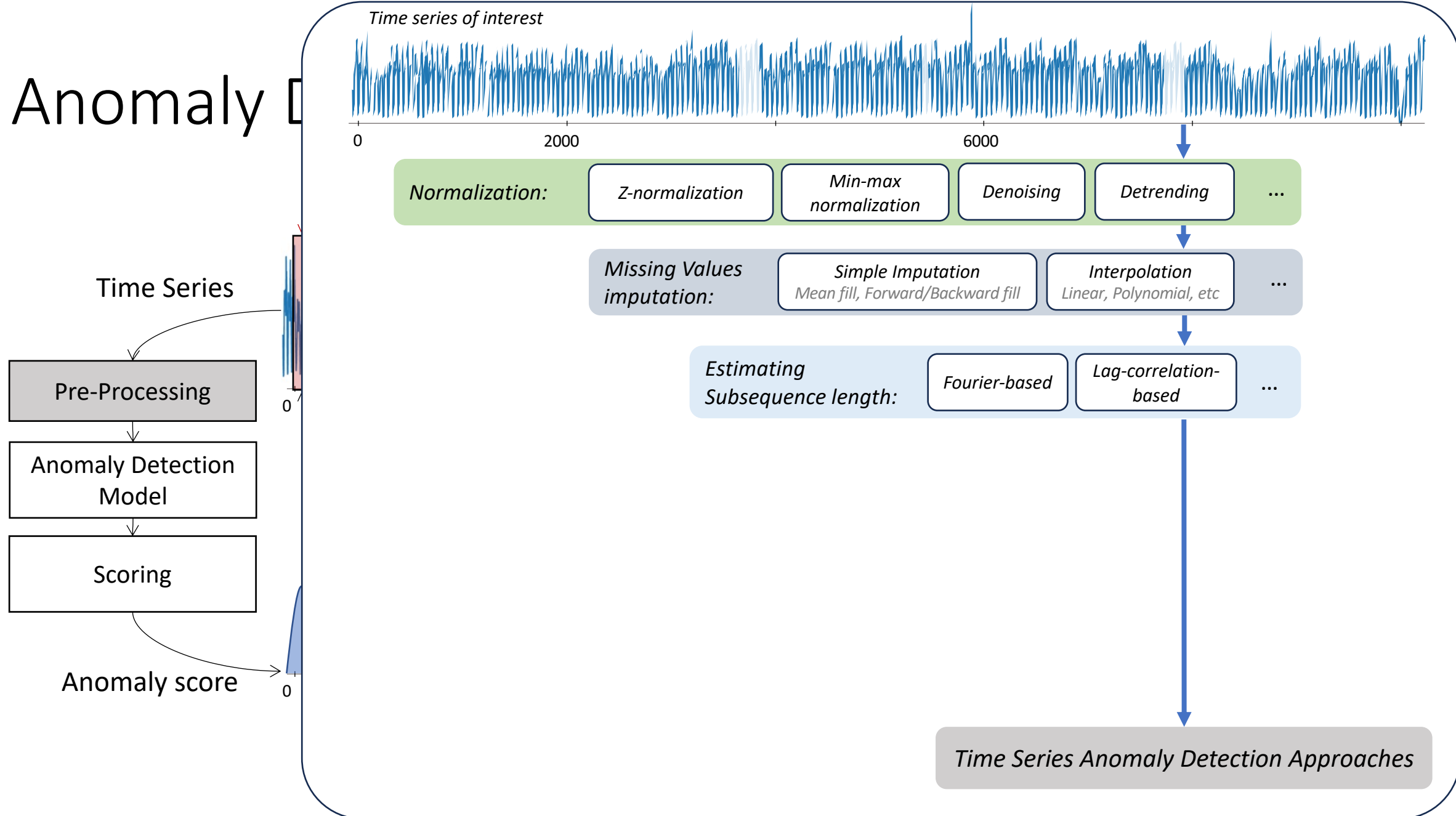
Part I.

Time Series Anomaly Detection

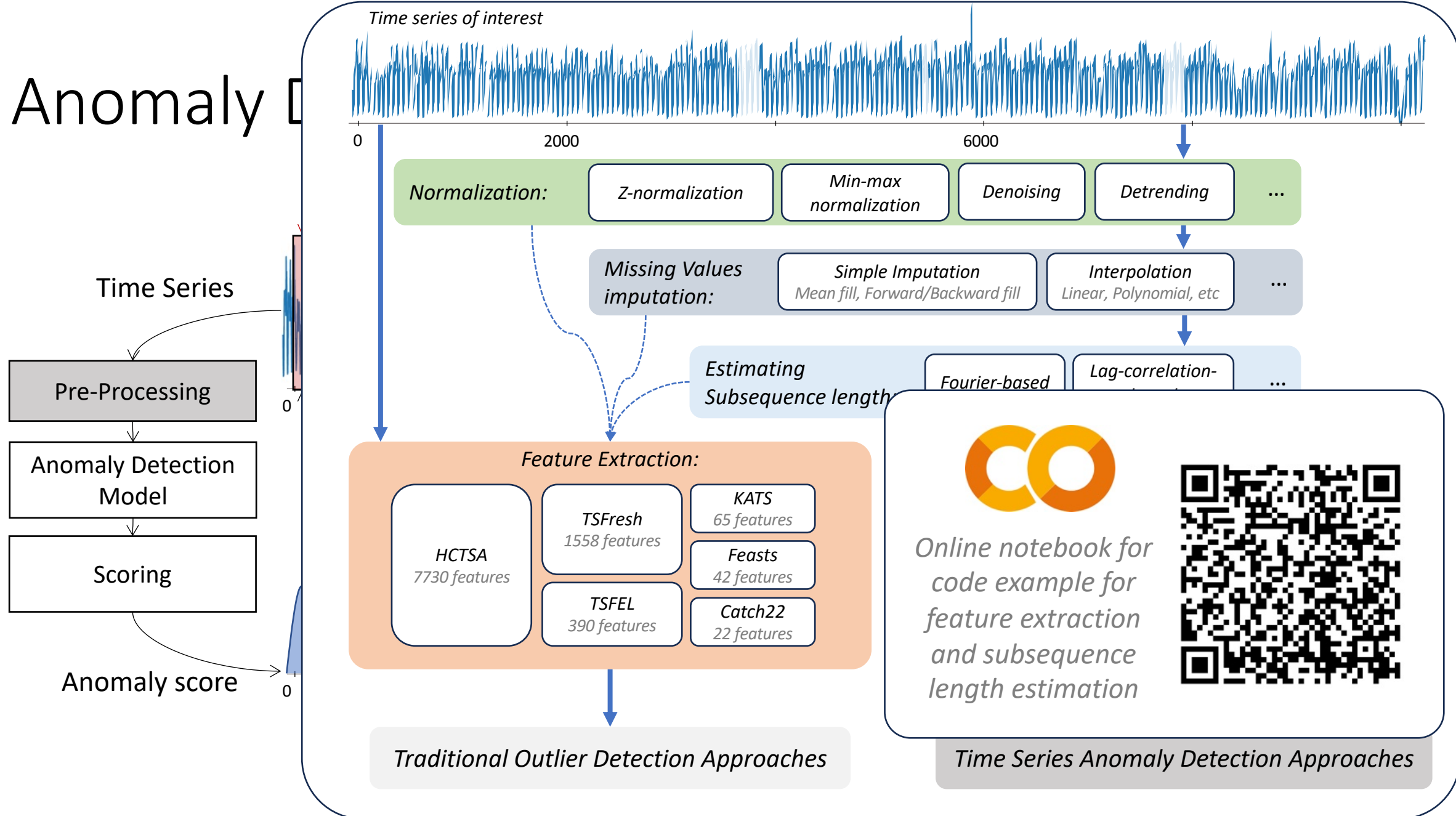
Anomaly Detection methods: *A taxonomy*



Anomaly

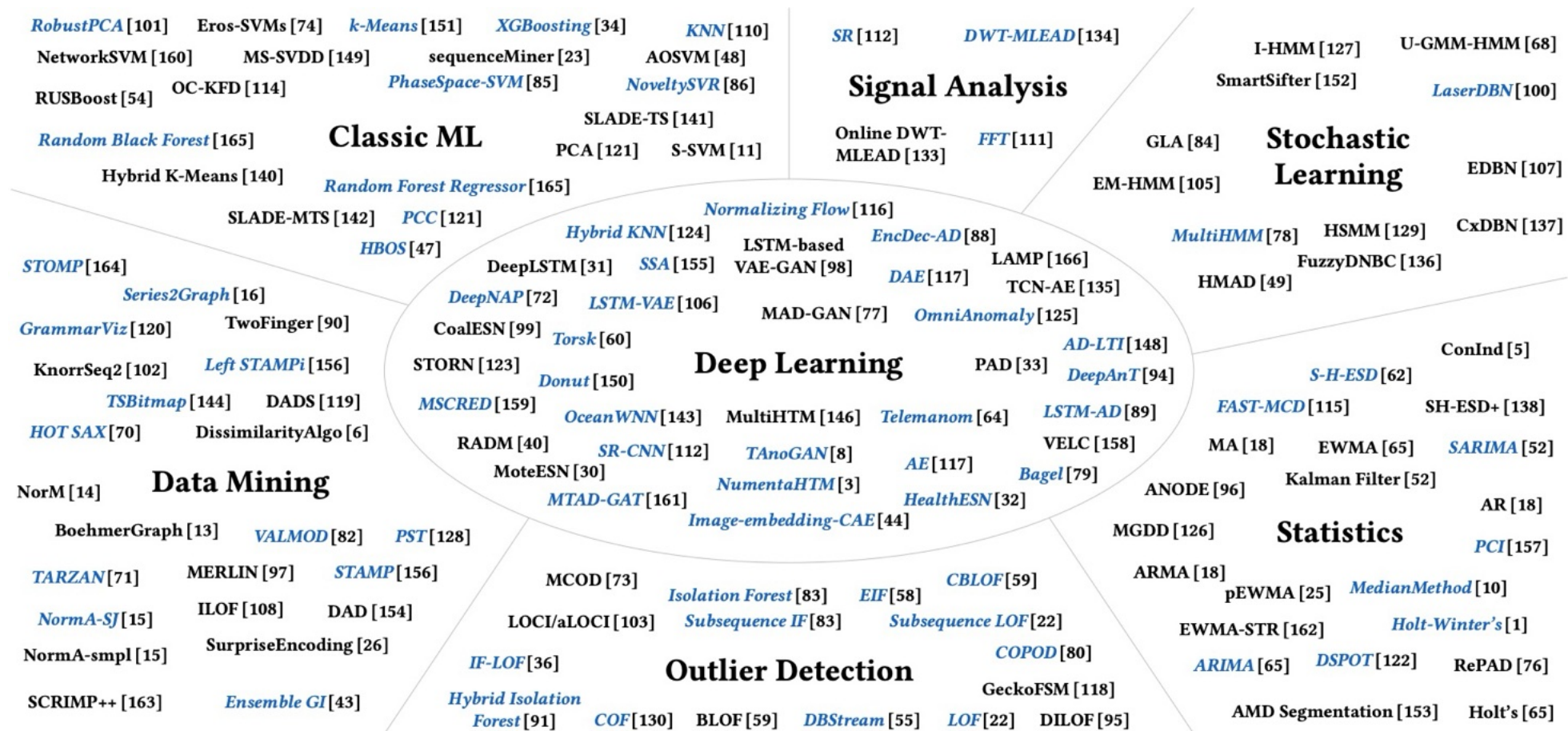


Anomaly



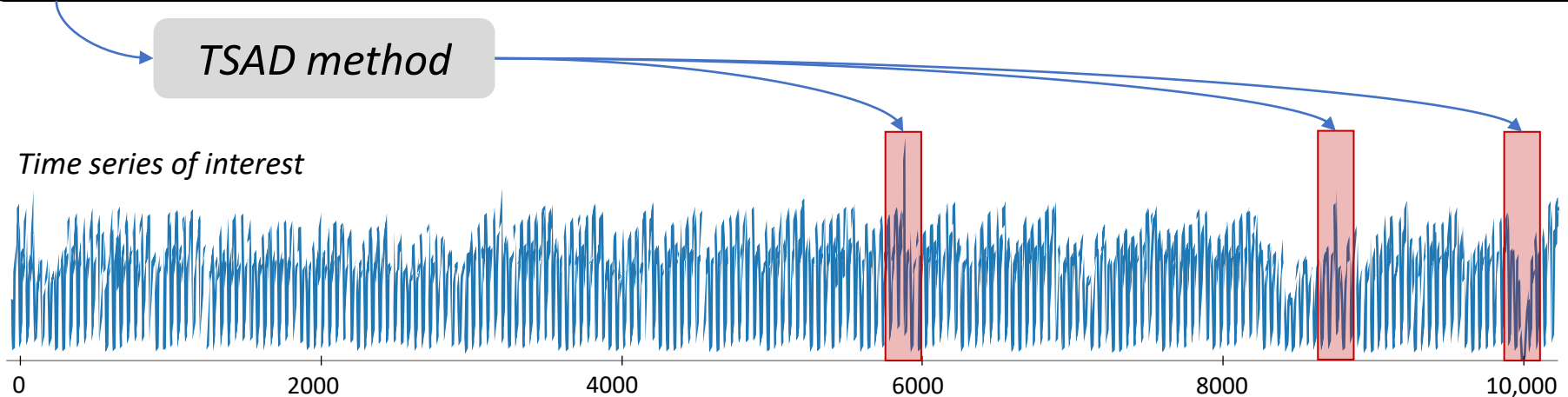
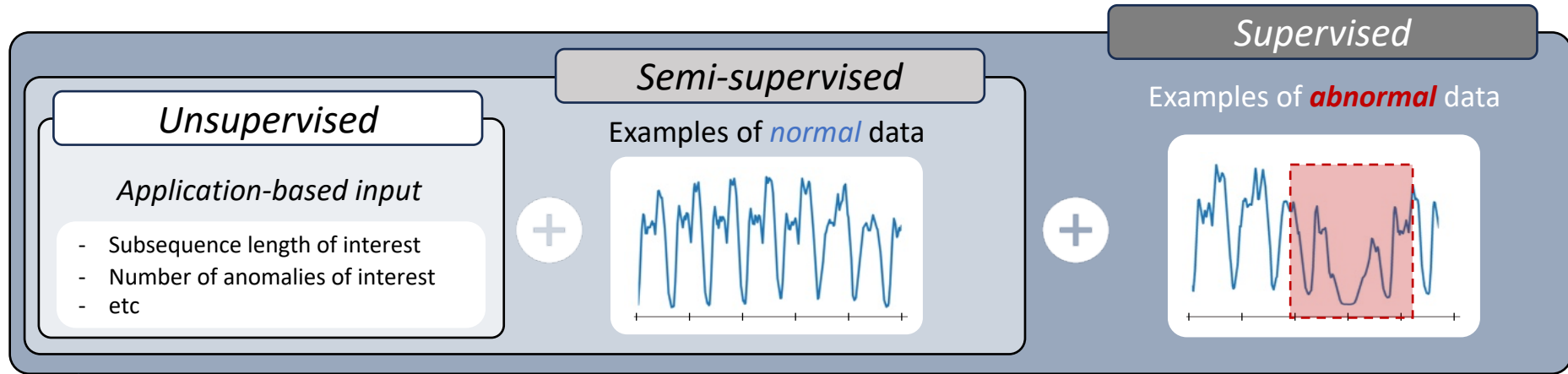
Anomaly Detection methods: *A taxonomy*

By domains [5] ...



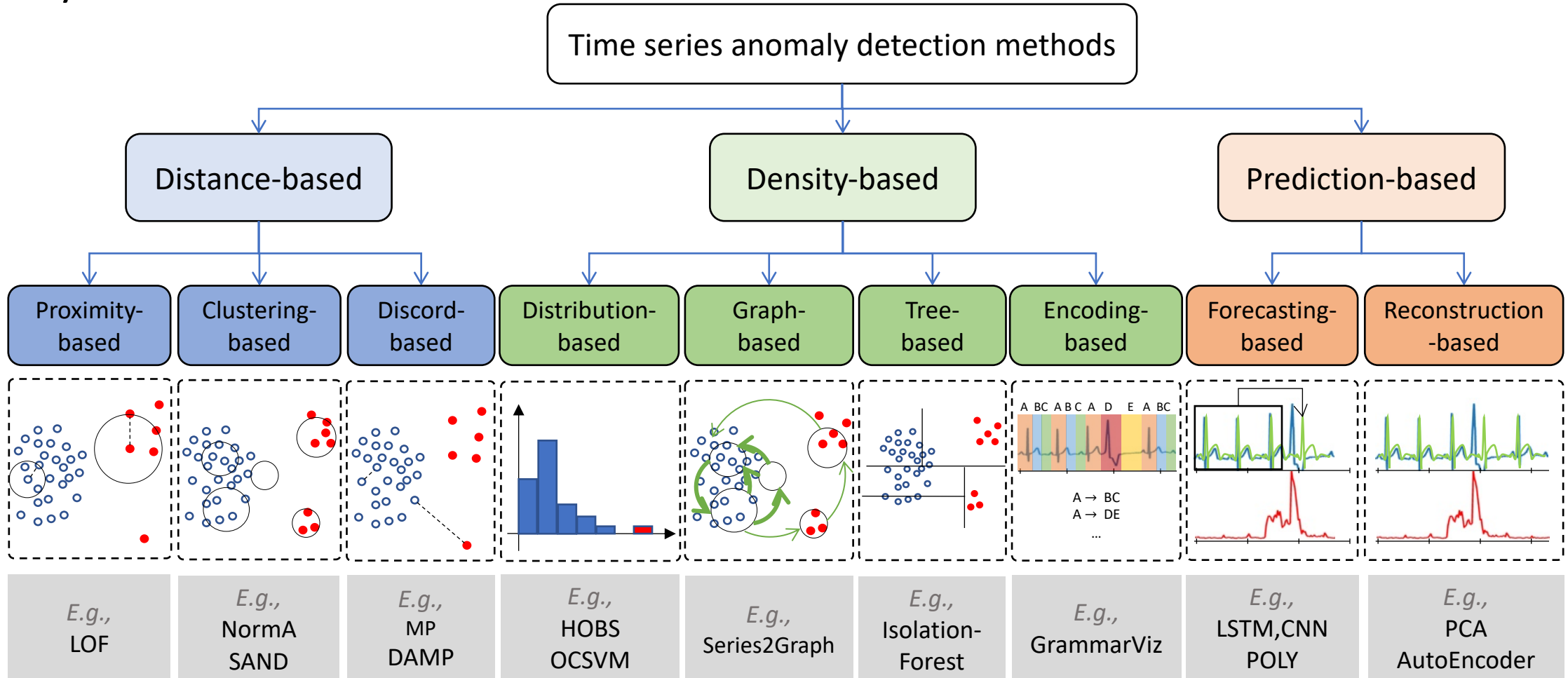
Anomaly Detection methods: *A taxonomy*

By inputs...

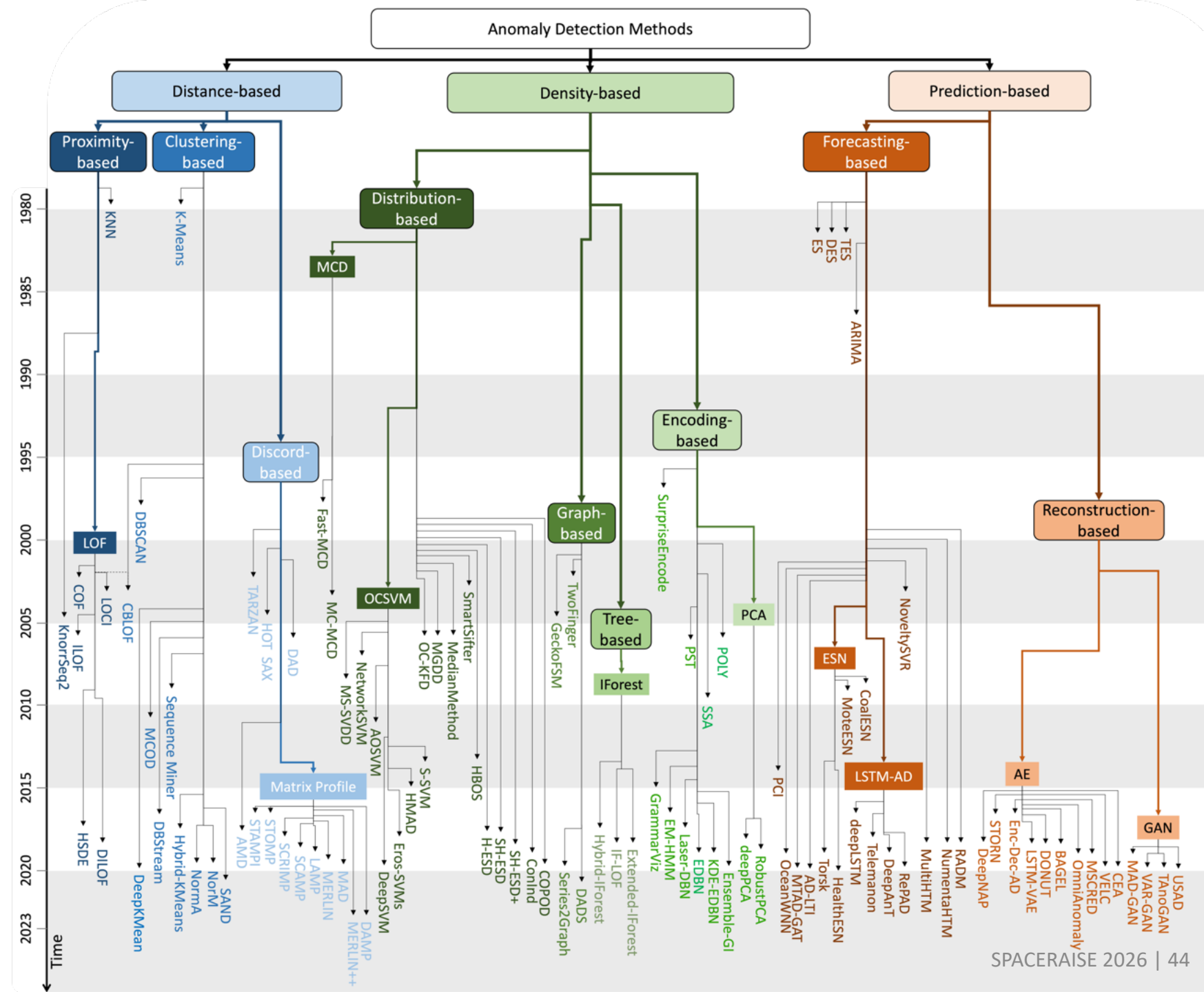


Anomaly Detection methods: *A taxonomy*

By methods...



Anomaly Detection methods: *A taxonomy*



Anomaly Detection methods: A taxonomy

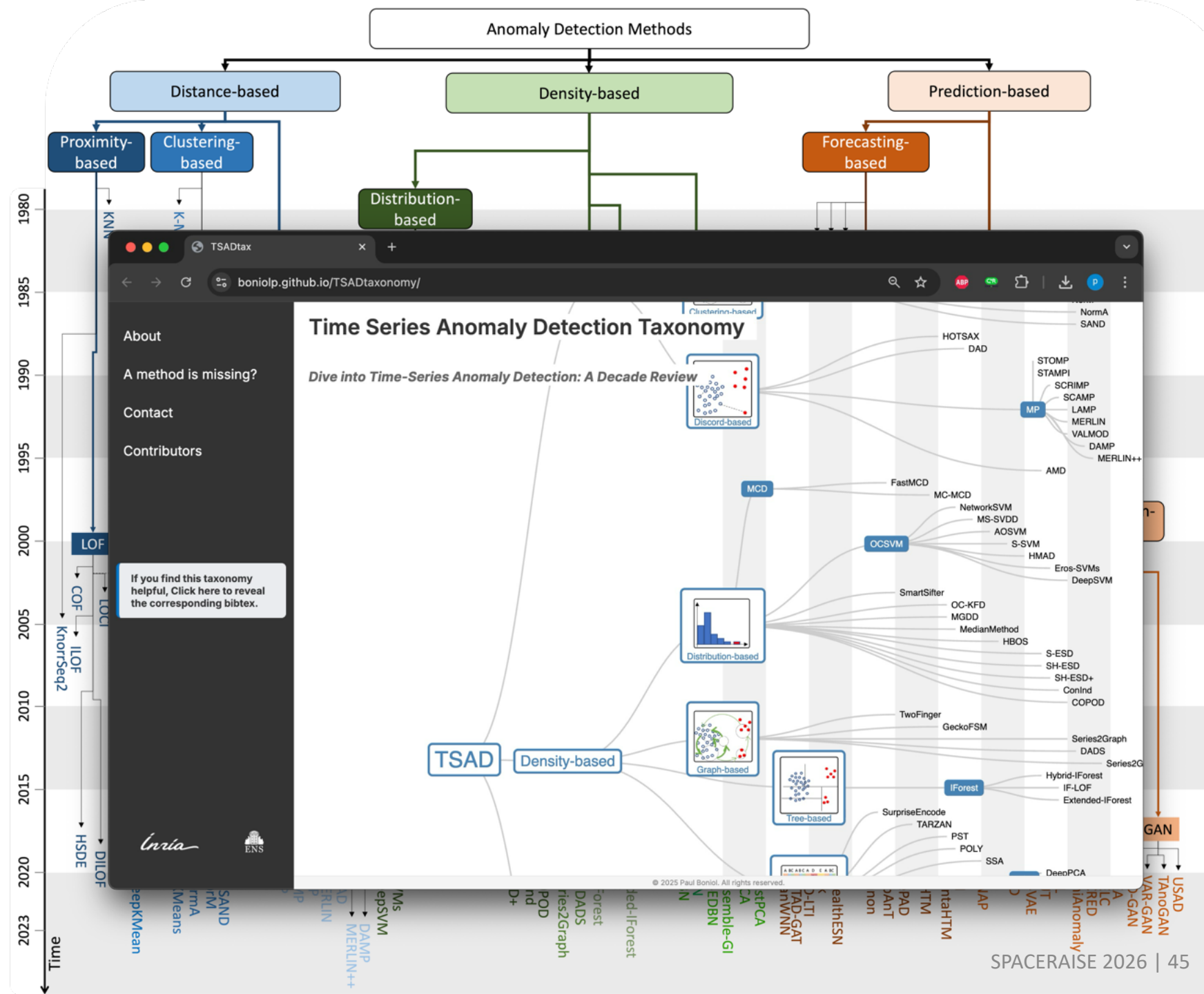
Online interactive
Taxonomy

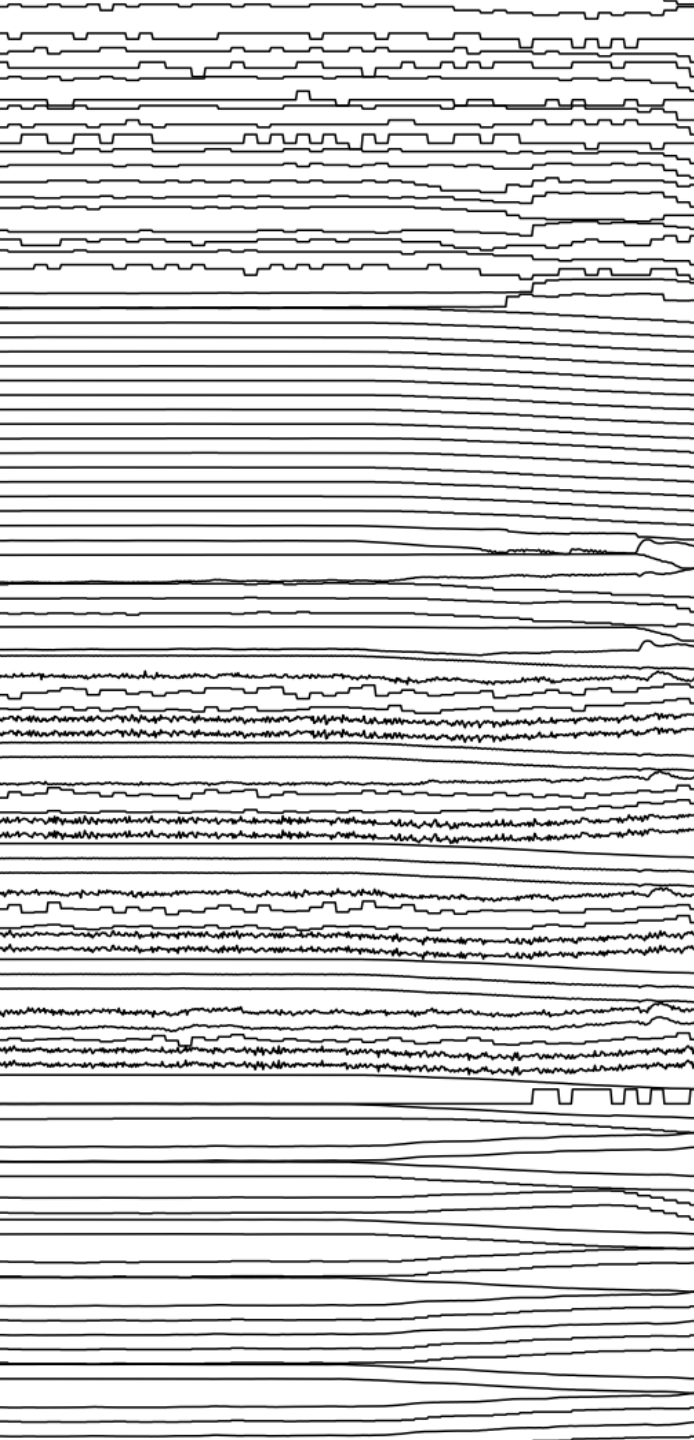


<https://boniold.github.io/TSADtaxonomy/>



<https://github.com/boniold/TSADtaxonomy>



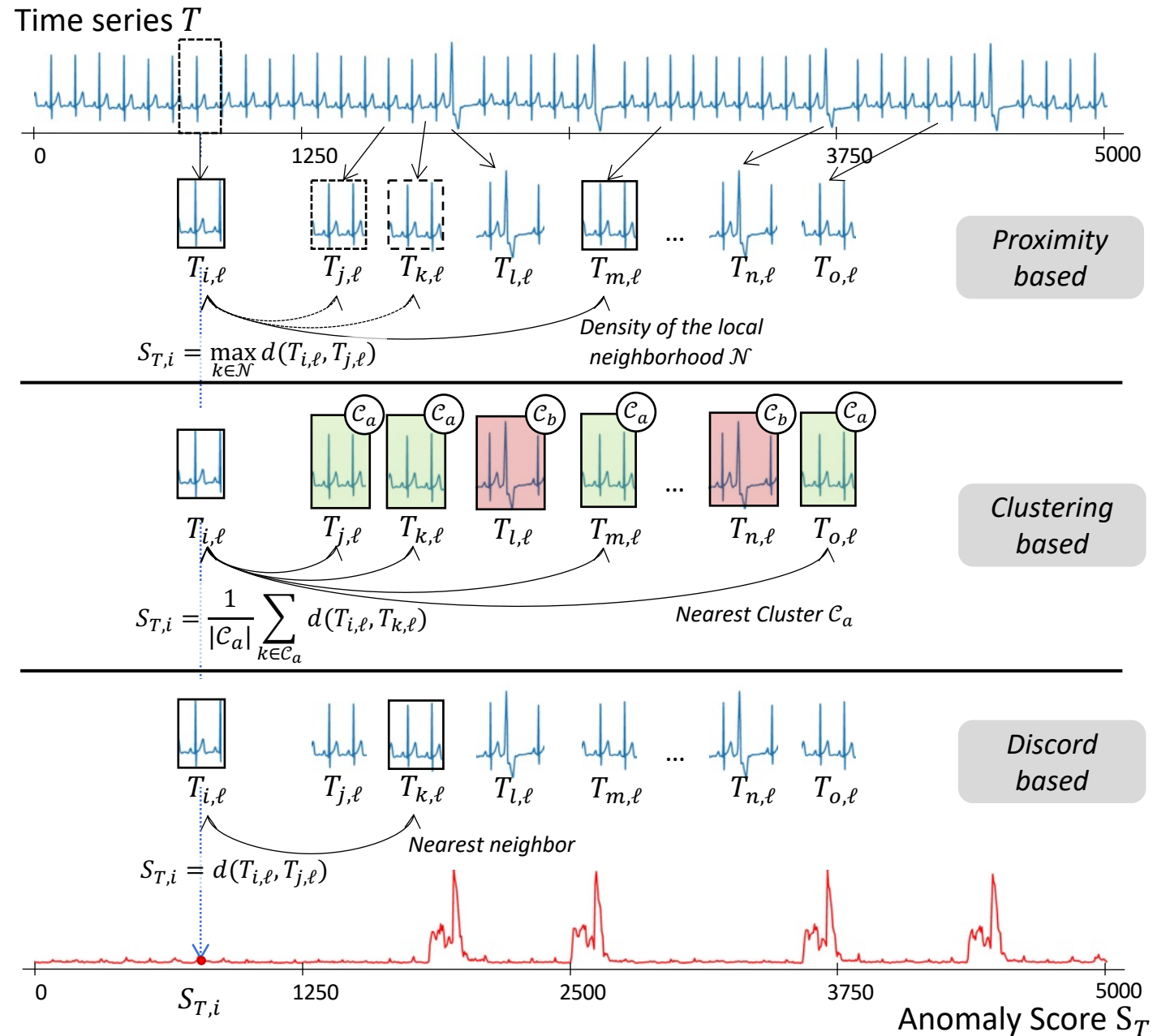
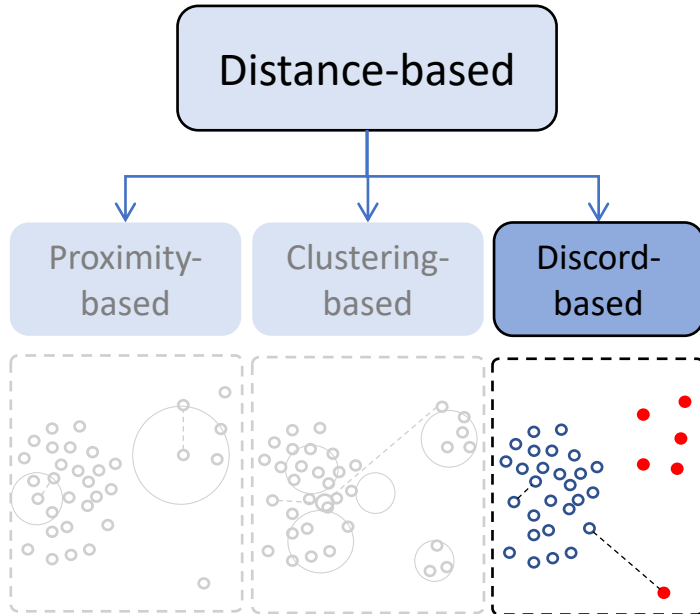


Part II.

Time Series Anomaly Detection Methods

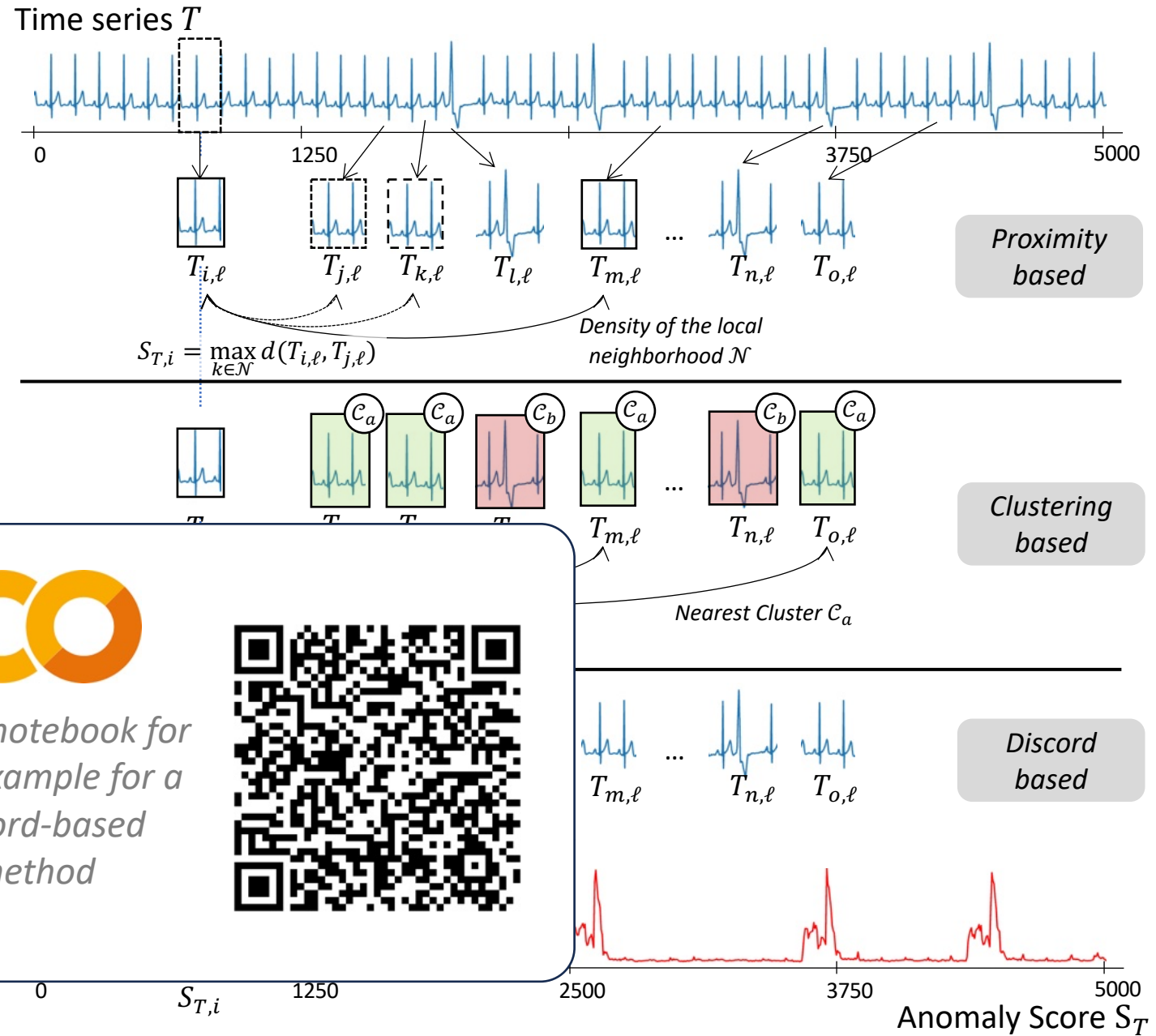
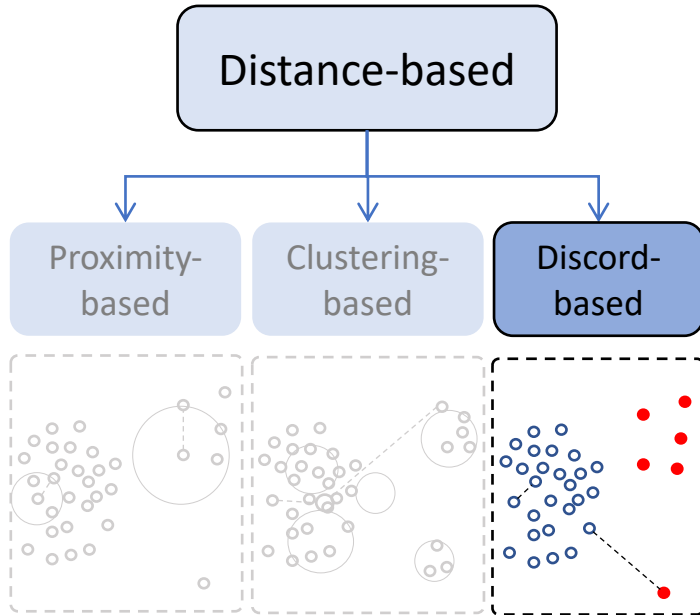
Anomaly Detection methods:

Distance-based



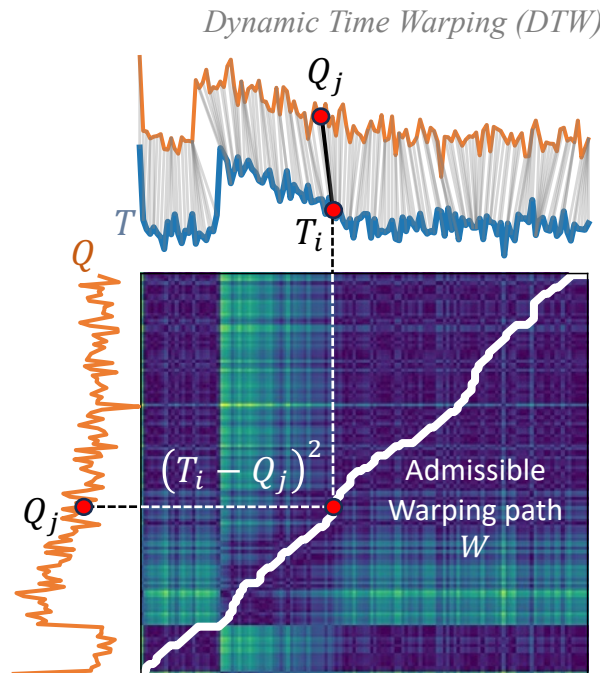
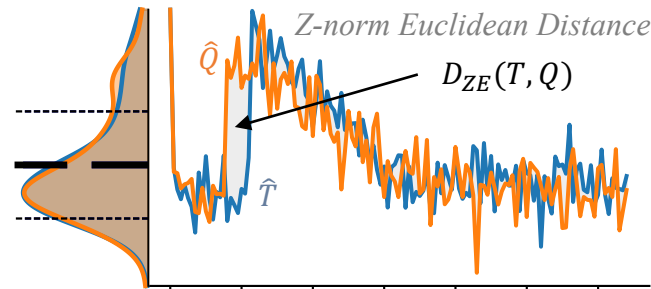
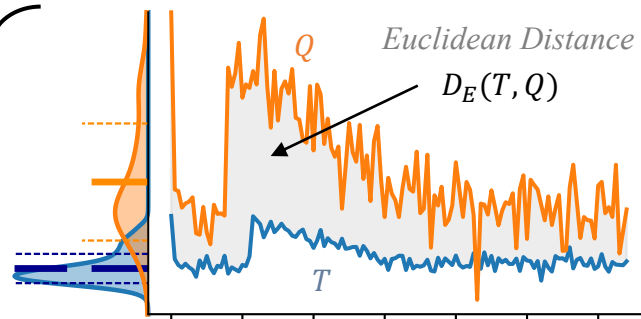
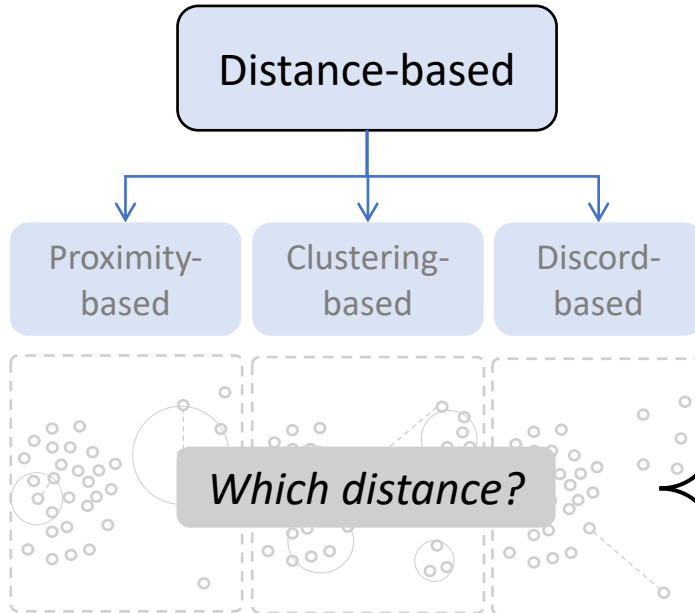
Anomaly Detection methods:

Distance-based



Anomaly Detection methods:

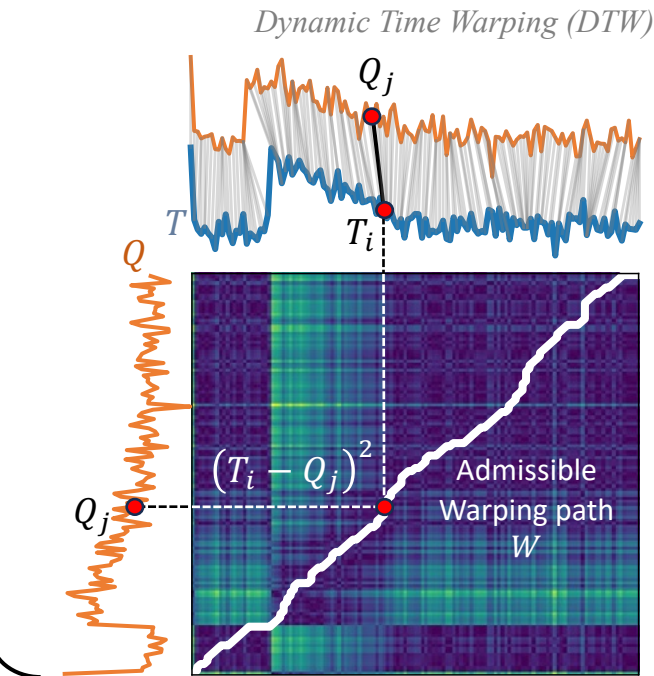
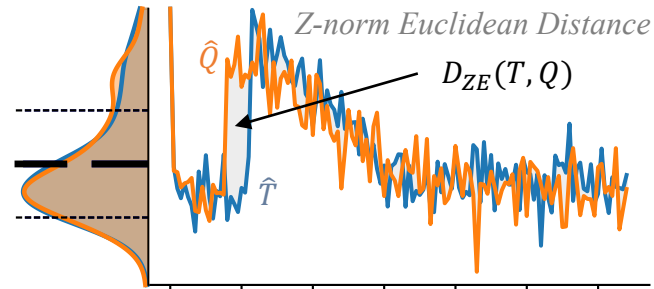
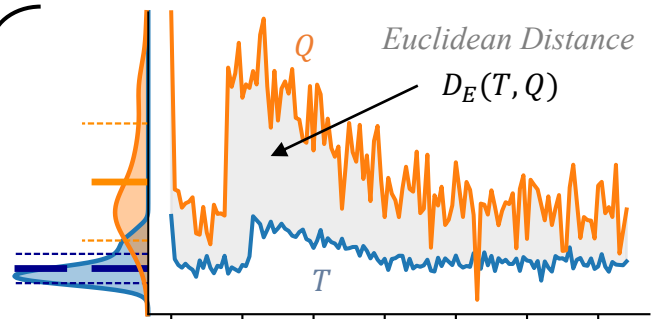
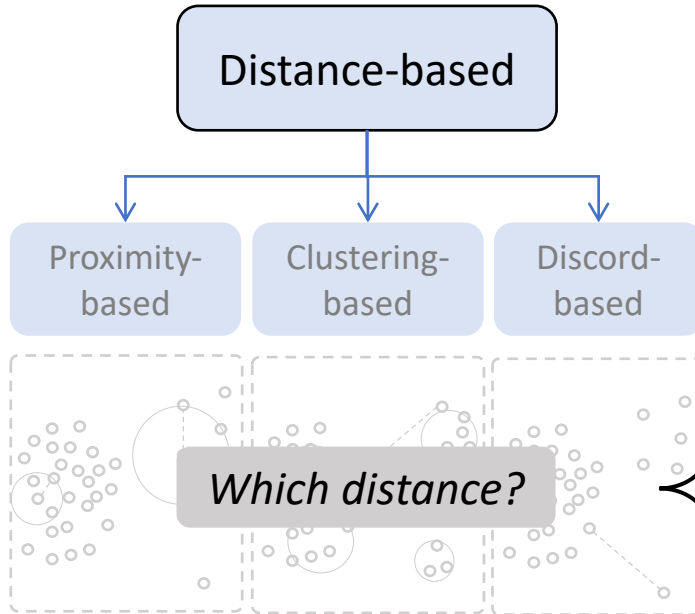
Distance-based



Distance	Formula	Time Complexity
Linear and Superlinear methods		
ED	$D(T, Q) = \sqrt{\sum_{i=0}^{n-1} (T_i - Q_i)^2}$	$O(n)$
z-norm ED	$D(T, Q) = \sqrt{\sum_{i=0}^{n-1} \left(\frac{T_i - \mu_T}{\sigma_T} - \frac{Q_i - \mu_Q}{\sigma_Q} \right)^2}$	$O(n)$
Pearson	$D(T, Q) = \sum_{i=0}^{n-1} \frac{(T_i - Q_i)^2}{Q_i}$	$O(n)$
Manhattan	$D(T, Q) = \sum_{i=0}^{n-1} T_i - Q_i $	$O(n)$
Chebyshev	$D(T, Q) = \max_{0 \leq i < n} T_i - Q_i $	$O(n)$
Cosine	$D(T, Q) = 1 - \frac{\sum_i T_i Q_i}{\sqrt{\sum_i T_i^2} \sqrt{\sum_i Q_i^2}}$	$O(n)$
SBD [99]	$D(T, Q) = 1 - \max_{\tau} \frac{\sum_{i=0}^{n-1} (T_i - \bar{T})(Q_{i+\tau} - \bar{Q})}{\sqrt{\sum_{i=0}^{n-1} (T_i - \bar{T})^2} \sqrt{\sum_{i=0}^{n-1} (Q_{i+\tau} - \bar{Q})^2}}$	$O(n \log n)$
Quadratic methods ($D(T, Q) = \sqrt{D(T_{n-1}, Q_{n-1})}$)		
DTW	$D(T_i, Q_j) = (T_i - Q_j)^2 + \min \begin{cases} D(T_{i-1}, Q_j), \\ D(T_i, Q_{j-1}), \\ D(T_{i-1}, Q_{j-1}) \end{cases}$	$O(n^2)$
LCSS [132]	$D(T_i, Q_j) = \begin{cases} 0 & i = 0 \text{ or } j = 0, \\ D(T_{i-1}, Q_{j-1}) + 1 & \text{if } T_i - Q_j \leq \epsilon, \\ \max \begin{cases} D(T_{i-1}, Q_j), \\ D(T_i, Q_{j-1}) \end{cases} & \text{otherwise.} \end{cases}$	$O(n^2)$
ERP [30]	$D(T_i, Q_j) = \begin{cases} \sum_{k=0}^i T_k - g & j = 0, \\ \sum_{k=0}^j Q_k - g & i = 0, \\ \min \begin{cases} D(T_{i-1}, Q_{j-1}) + T_i - Q_j , \\ D(T_{i-1}, Q_j) + T_i - g , \\ D(T_i, Q_{j-1}) + Q_j - g \end{cases} & \text{otherwise.} \end{cases}$	$O(n^2)$
MSM [121]	$D(T_i, Q_j) = \min \begin{cases} D(T_{i-1}, Q_{j-1}) + T_i - Q_j , \\ D(T_{i-1}, Q_j) + C(T_i, T_{i-1}, Q_j), \\ D(T_i, Q_{j-1}) + C(Q_j, Q_{j-1}, T_i) \end{cases}$ with $C(x, y, z) = \begin{cases} c & \text{if } x \in [y, z] \text{ or } x \in [z, y], \\ c + \min\{ x - y , x - z \} & \text{otherwise.} \end{cases}$	$O(n^2)$
TWED [83]	$D(T_i, Q_j) = \begin{cases} \sum_{k=1}^i T_k - T_{k-1} + \lambda i & j = 0, \\ \sum_{k=1}^j Q_k - Q_{k-1} + \lambda j & i = 0, \\ \min \begin{cases} D(T_{i-1}, Q_{j-1}) + T_i - Q_j + \nu i - j , \\ D(T_{i-1}, Q_j) + T_i - T_{i-1} + \lambda + \nu, \\ D(T_i, Q_{j-1}) + Q_j - Q_{j-1} + \lambda + \nu \end{cases} & \text{otherwise.} \end{cases}$	$O(n^2)$

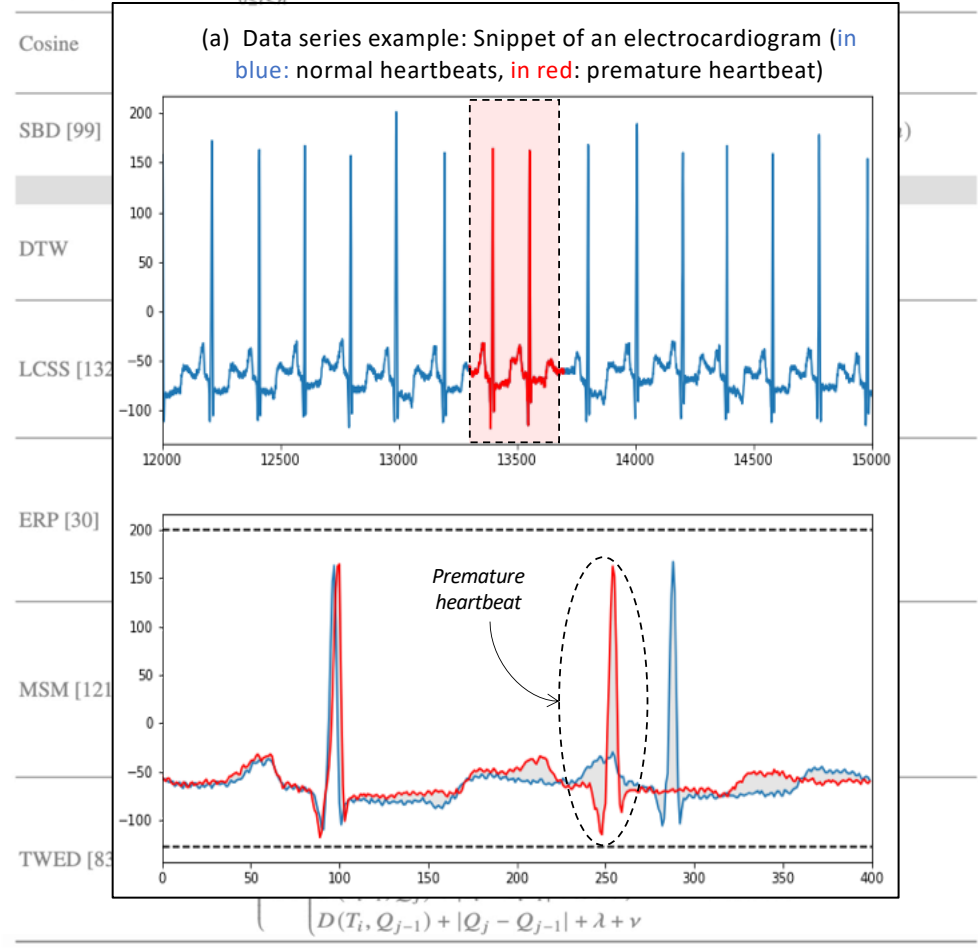
Anomaly Detection methods:

Distance-based



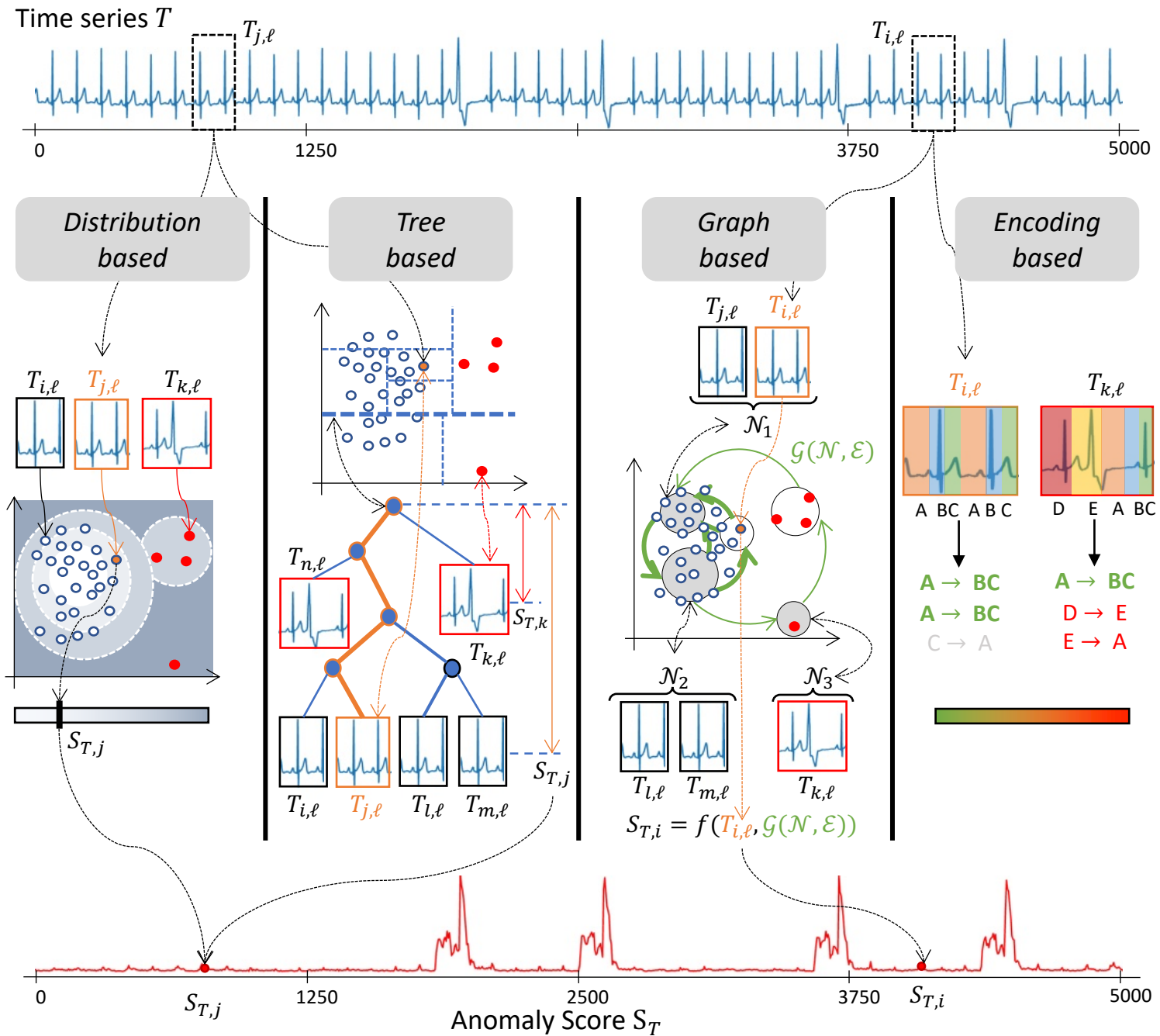
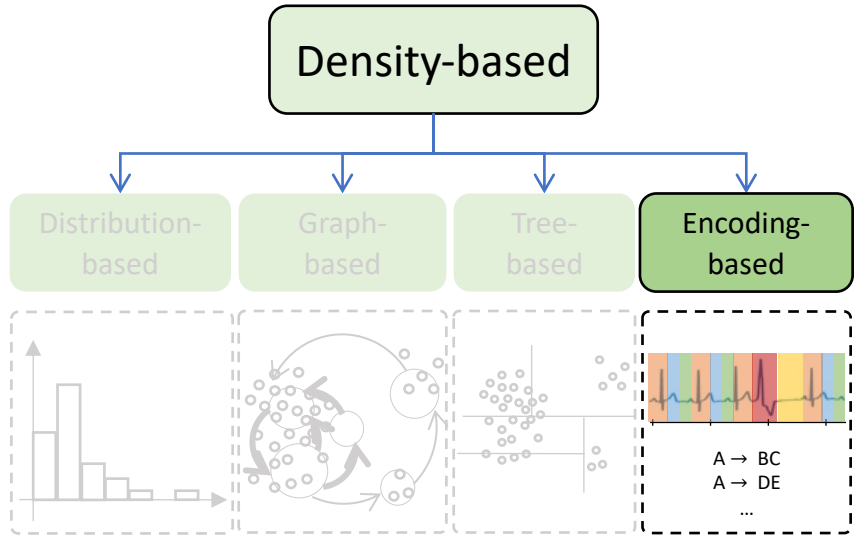
Distance	Formula	Time Complexity
Linear and Superlinear methods		
ED	$D(T, Q) = \sqrt{\sum_{i=0}^{n-1} (T_i - Q_i)^2}$	$O(n)$
z-norm ED	$D(T, Q)$	$O(n)$
Pearson	$D(T, Q)$	$O(n)$
Manhattan	$D(T, Q)$	$O(n)$
Chebyshev	$D(T, Q) = \max_{0 \leq i < n} T_i - Q_i $	$O(n)$

Which distance is the most appropriate for time series anomaly detection?



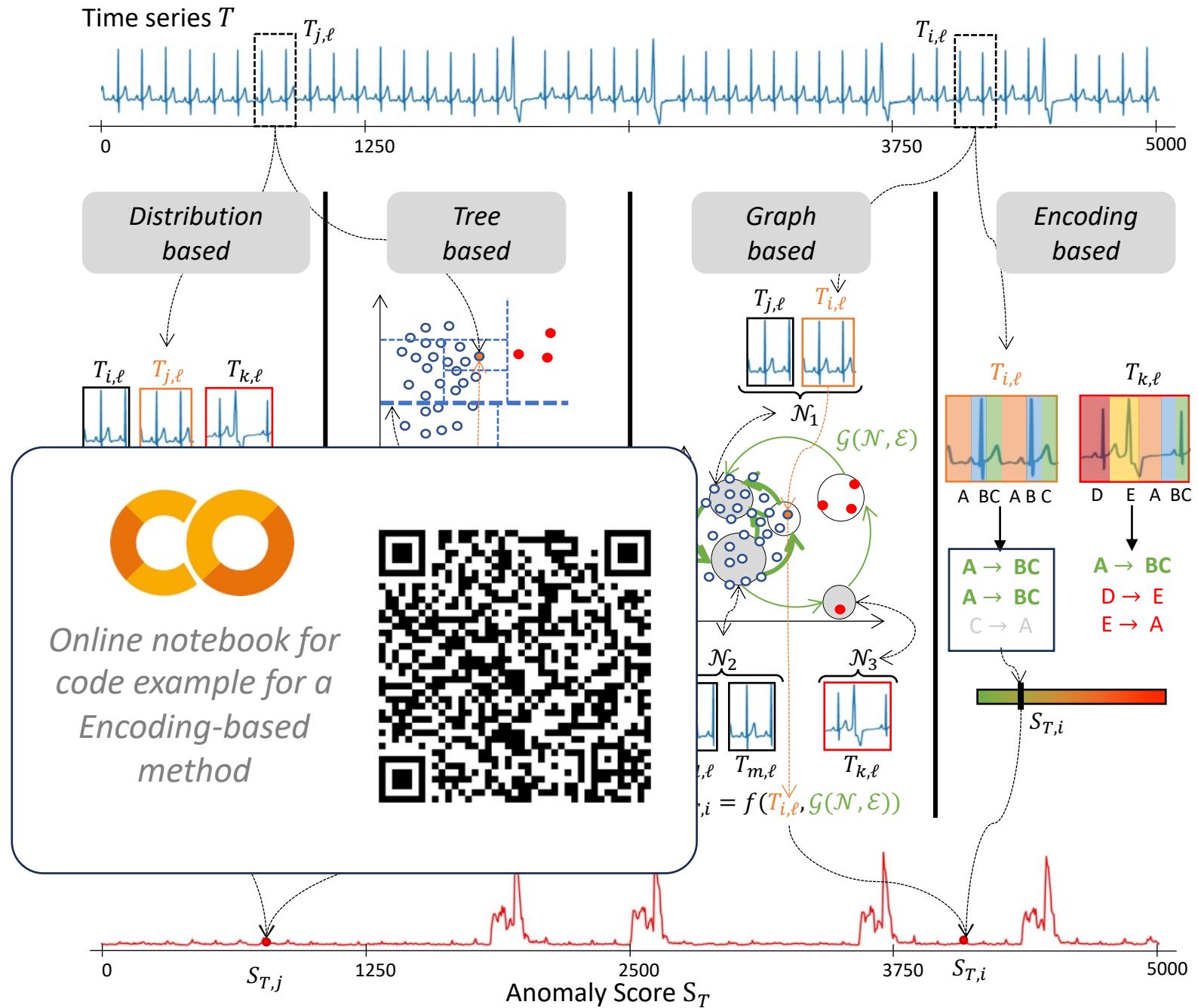
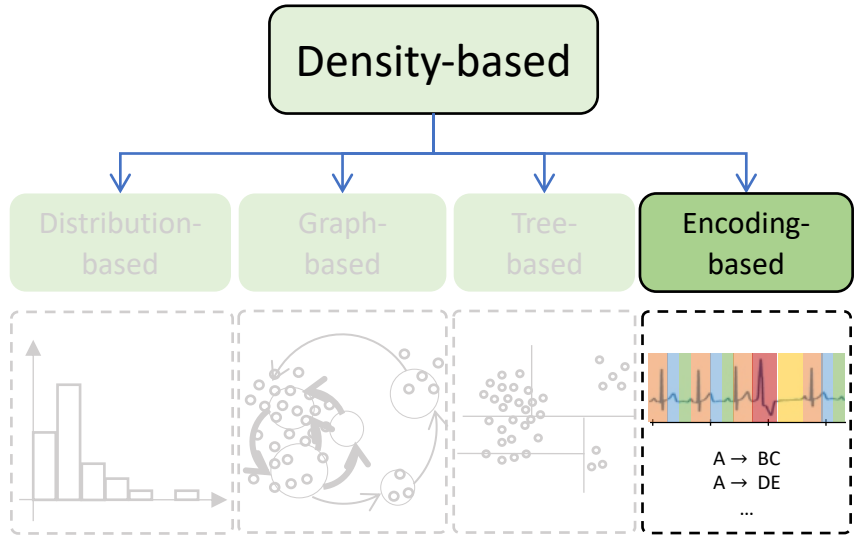
Anomaly Detection methods:

Density-based



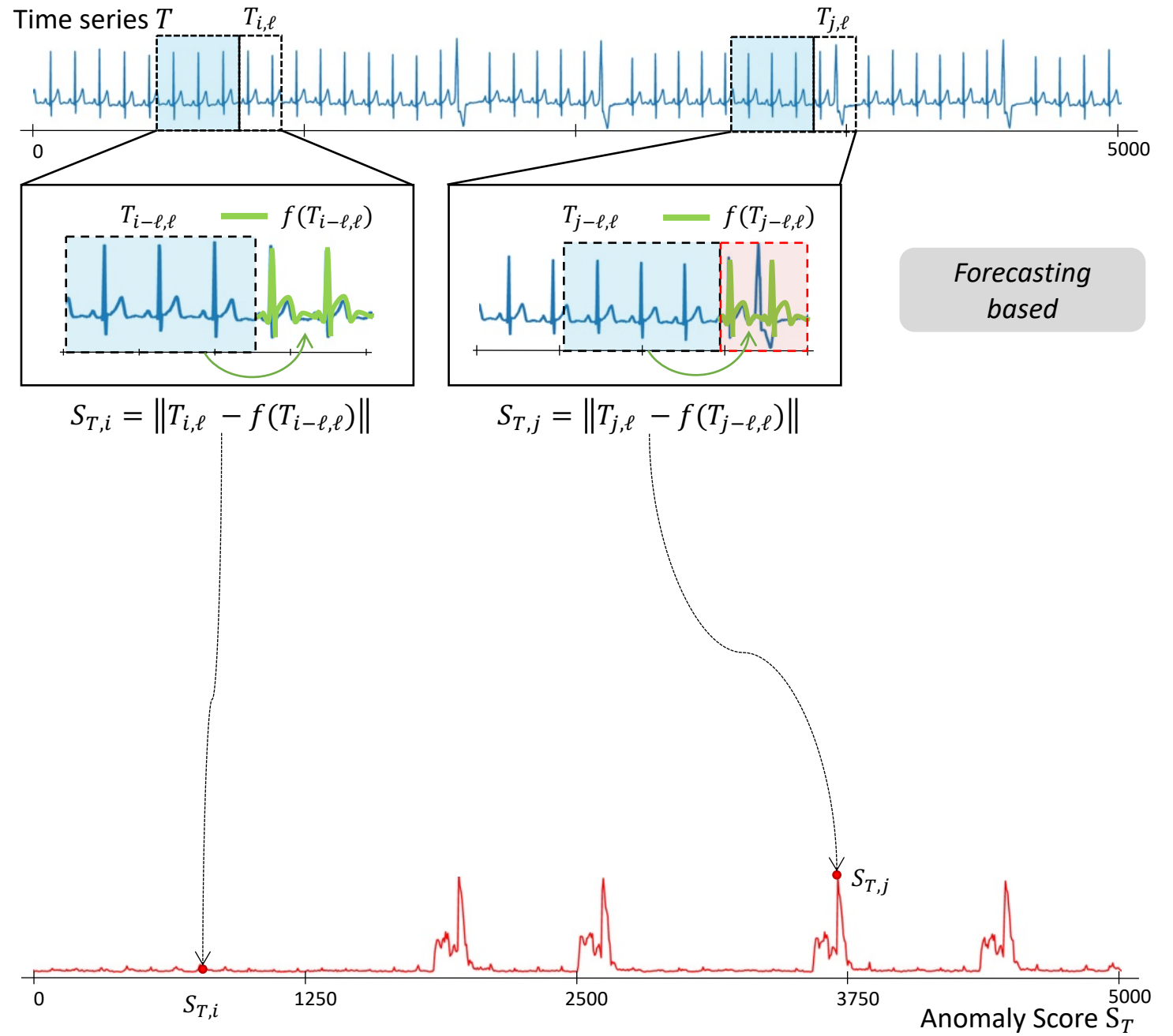
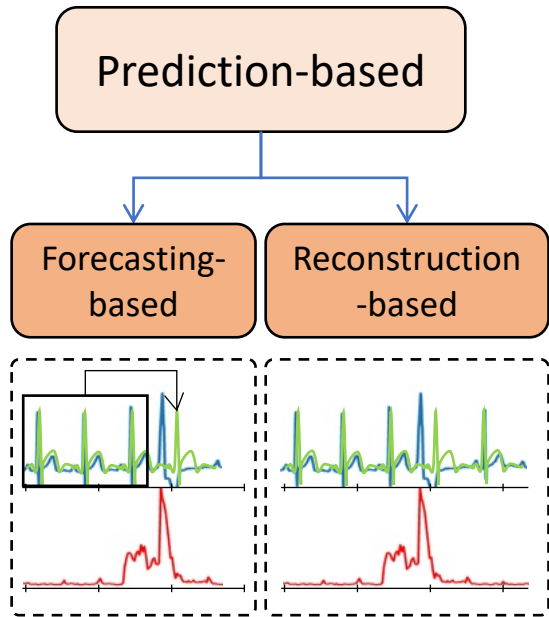
Anomaly Detection methods:

Density-based



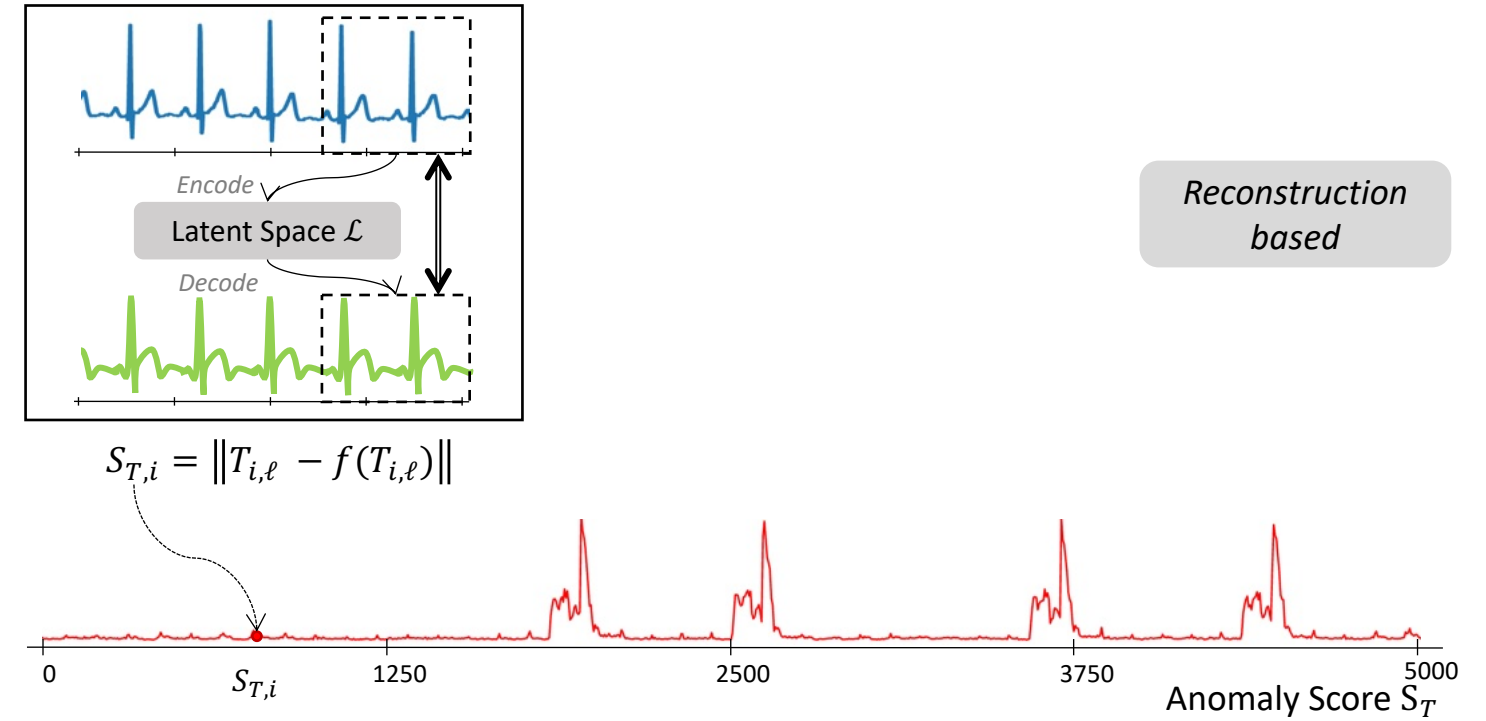
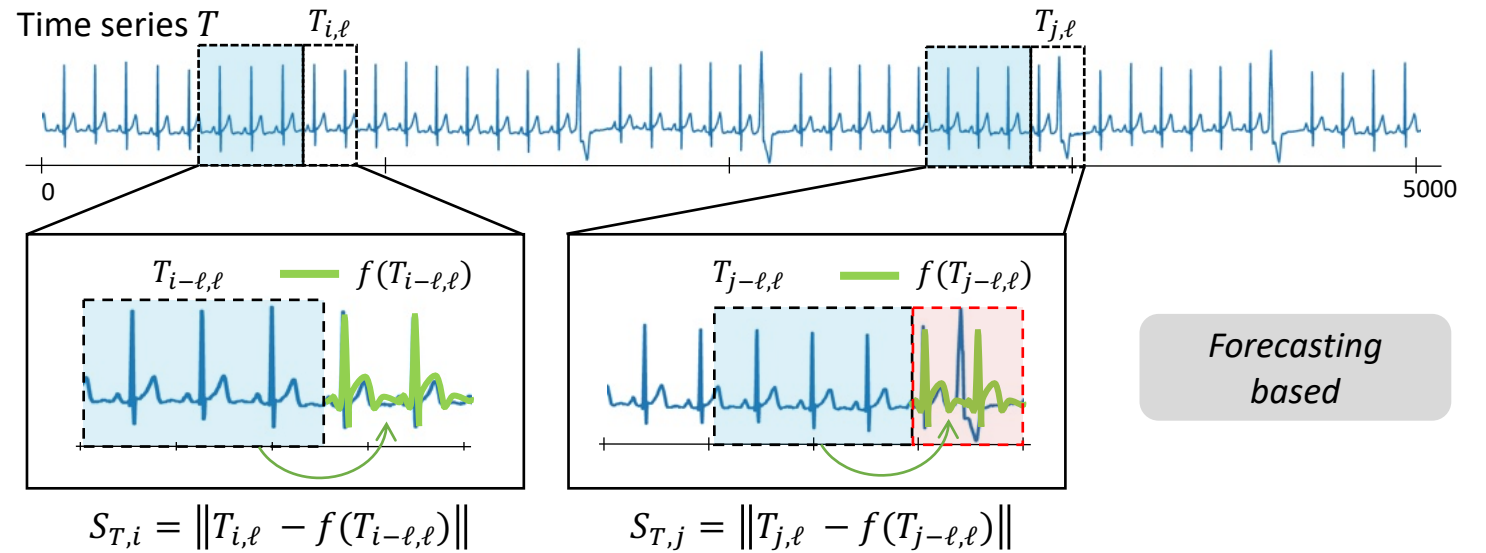
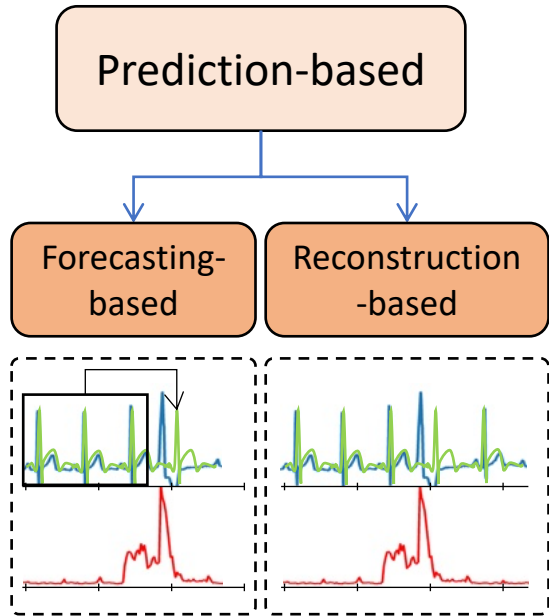
Anomaly Detection methods:

Prediction-based



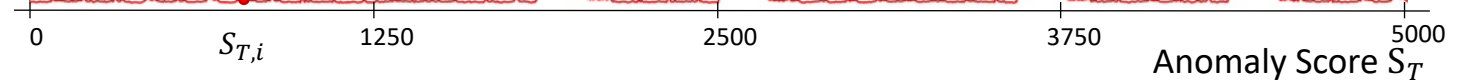
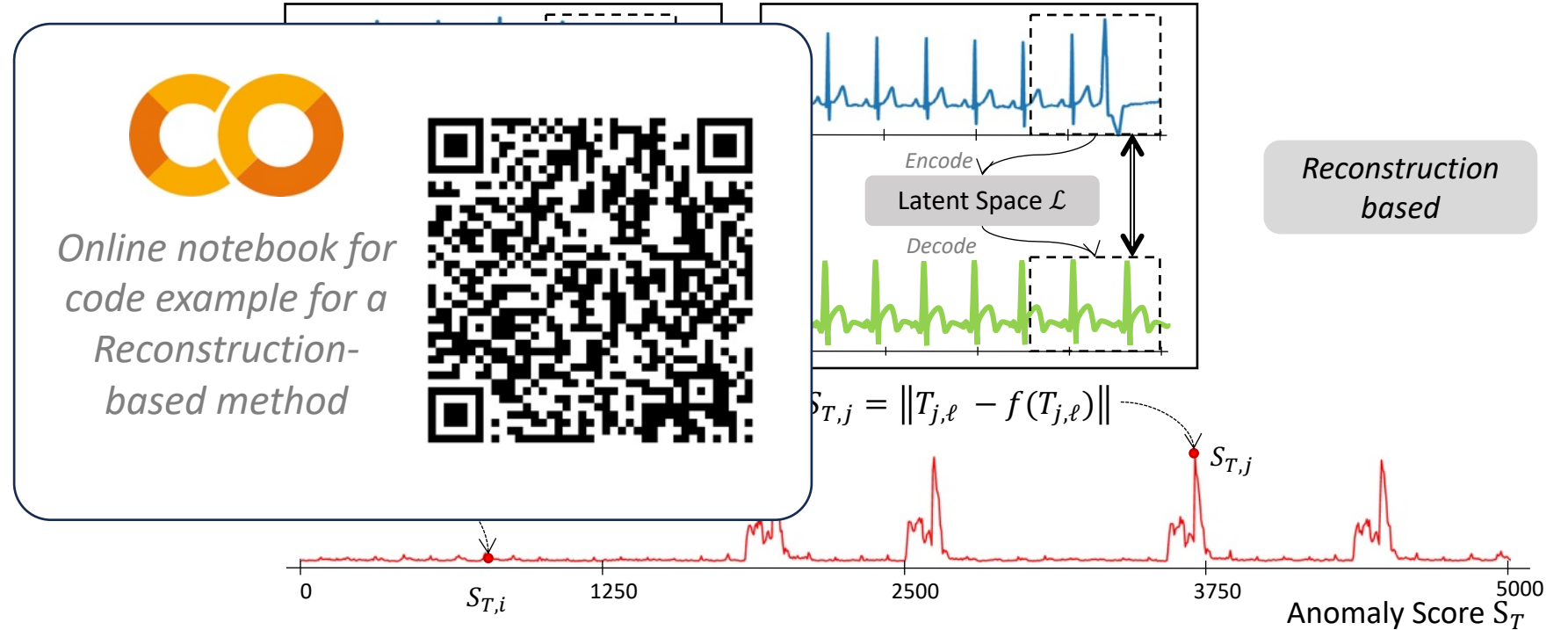
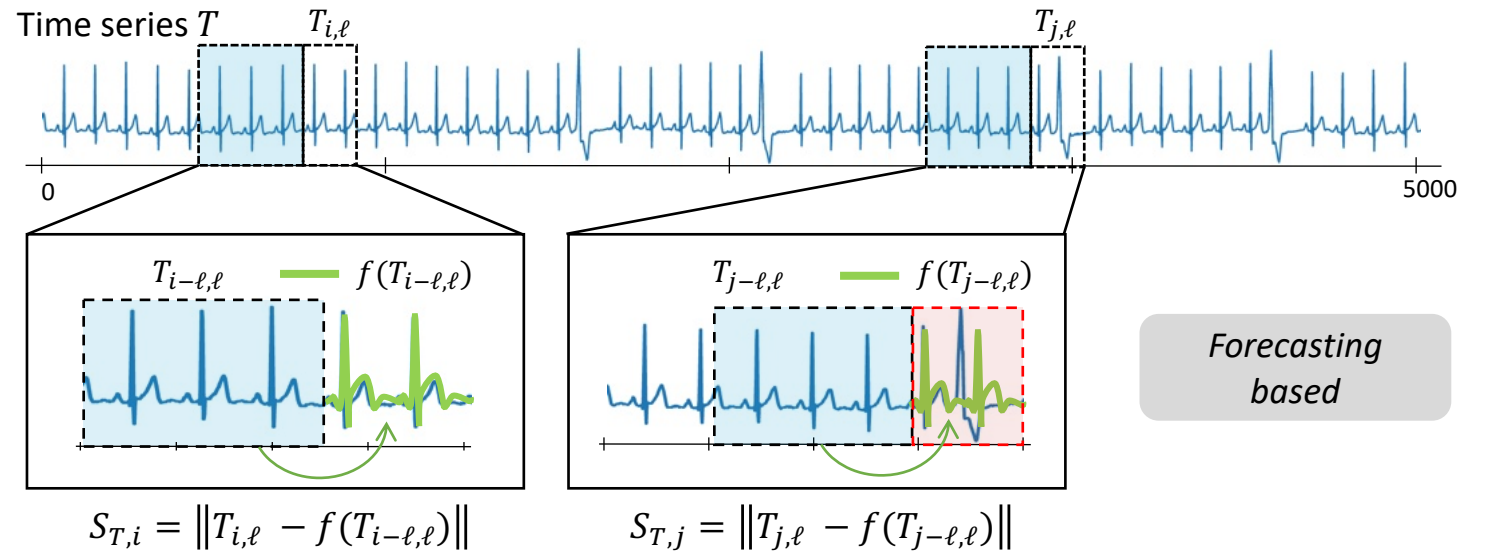
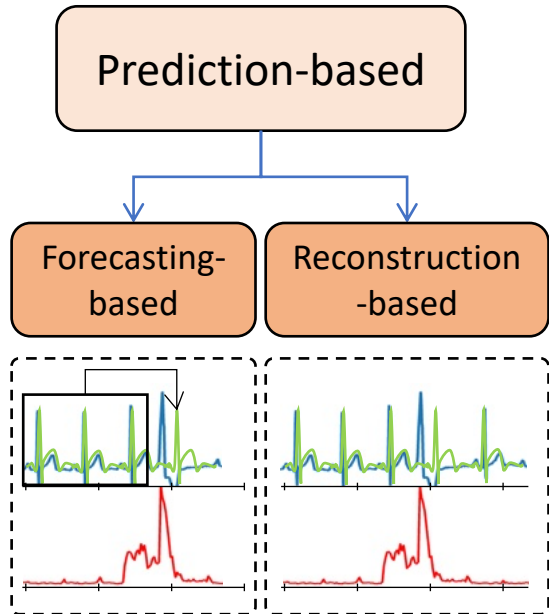
Anomaly Detection methods:

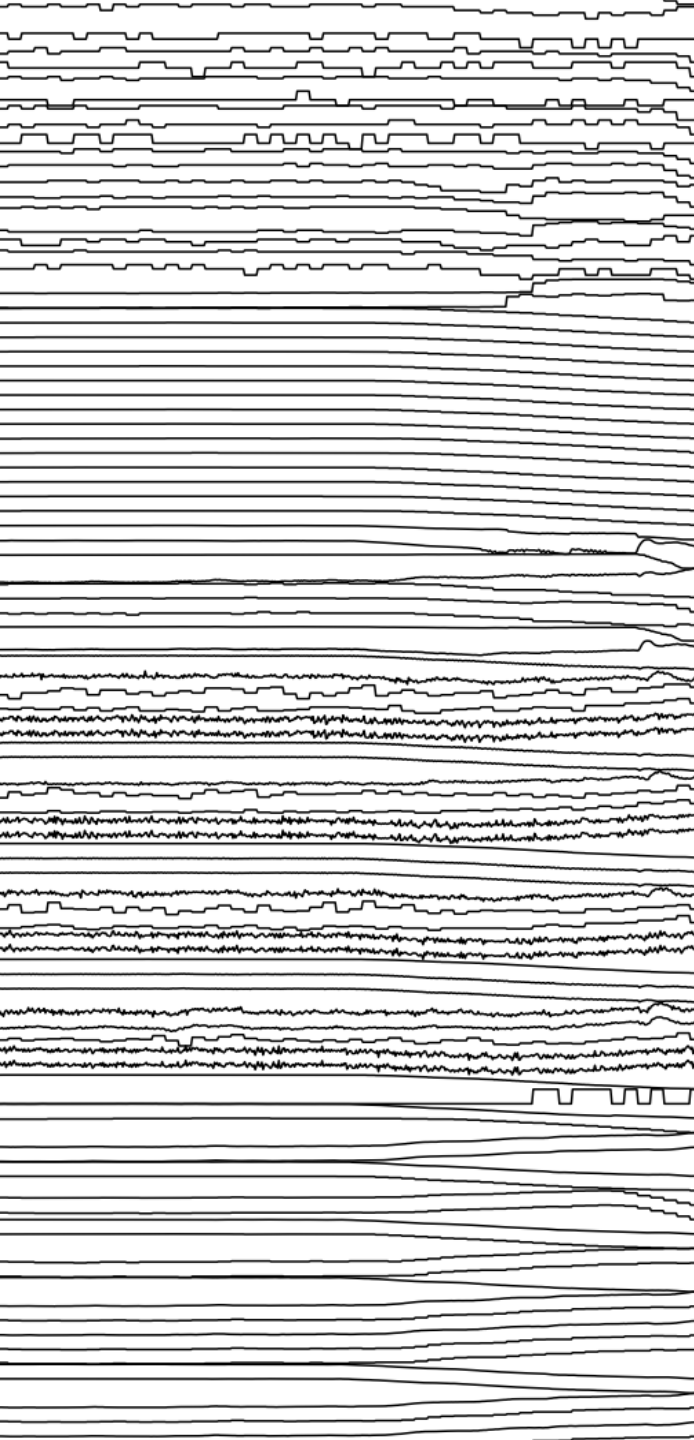
Prediction-based



Anomaly Detection methods:

Prediction-based

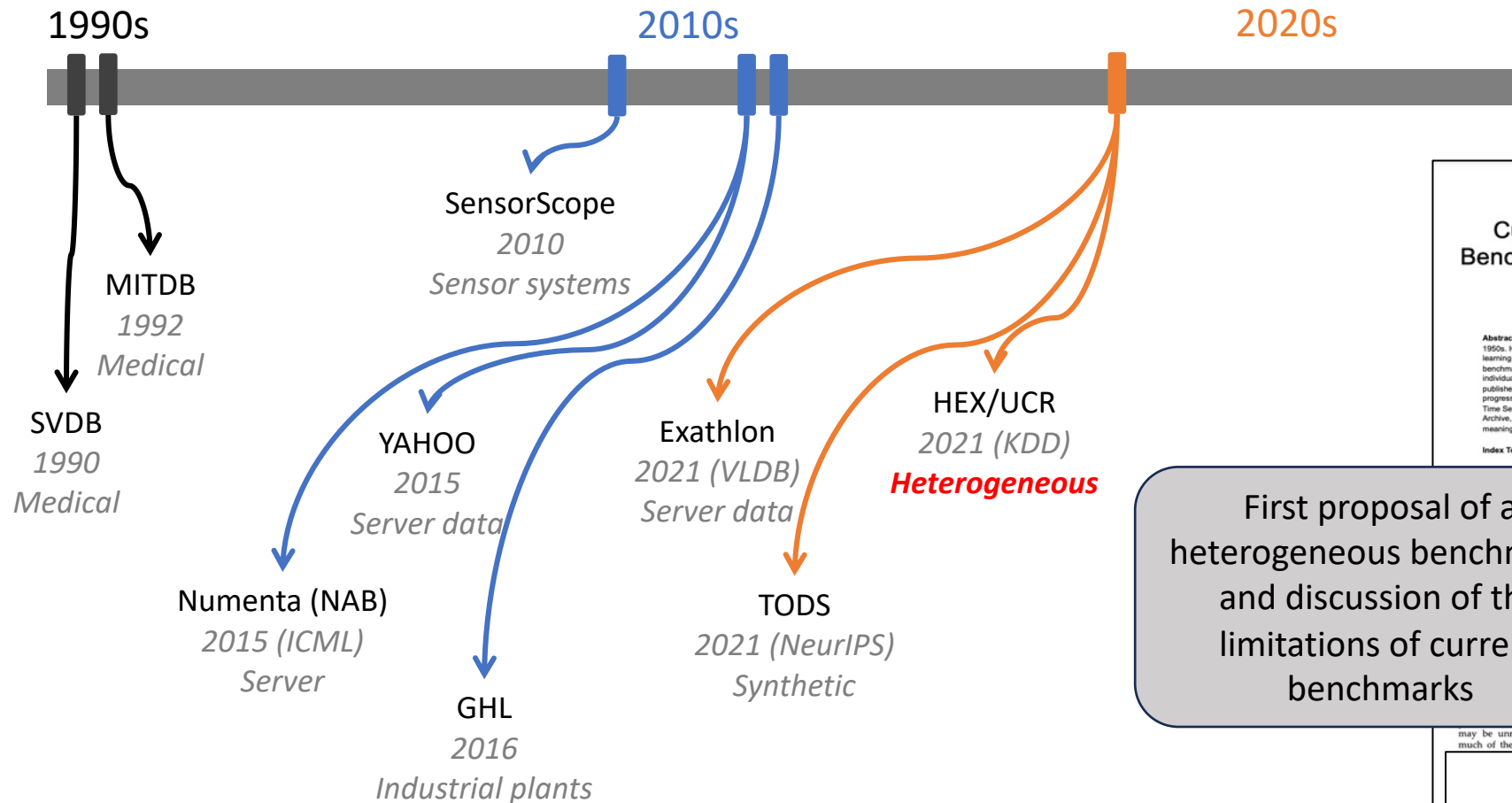




Part III.

Benchmark, Evaluation Measures, and Automatic Solutions

Anomaly Detection methods: *Existing benchmark*



Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress
Renjie Wu and Eamonn J. Keogh

Abstract—Time series anomaly detection has been a perennially important topic in data science, with papers dating back to the 1950s. However, in recent years there has been an explosion of interest in this topic, much of it driven by the success of deep learning in other domains and for other time series tasks. Most of these papers test on one or more of a handful of popular benchmark datasets, created by Yahoo, Numenta, NASA, etc. In this work we make a surprising claim. The majority of the individual exemplars in these datasets suffer from one or more of four flaws. Because of these four flaws, we believe that many published comparisons of anomaly detection algorithms may be unreliable, and more importantly, much of the apparent progress in recent years may be illusory. In addition to demonstrating these claims, with this paper we introduce the UCR Time Series Anomaly Archive. We believe that this resource will perform a similar role as the UCR Time Series Classification Archive, by providing the community with a benchmark that allows meaningful comparisons between approaches and a meaningful gauge of overall progress.

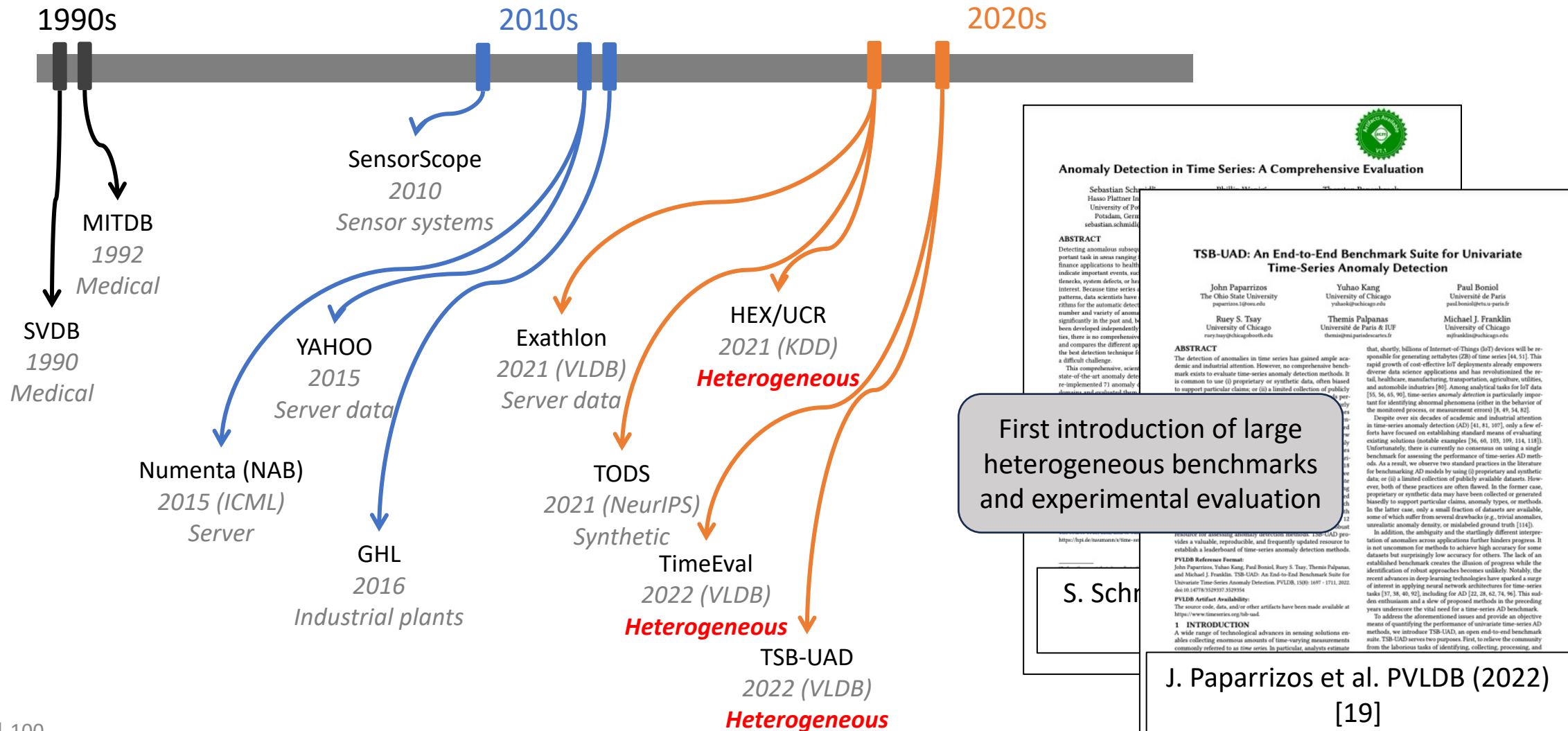
Index Terms—Anomaly detection, benchmark datasets, deep learning, time series analysis

seen a perennially important topic in data science, with papers dating back to the 1950s. However, in recent years there has been an explosion of interest in this topic, much of it driven by the success of deep learning in other domains and for other time series tasks. Most of these papers test on one or more of a handful of popular benchmark datasets, created by Yahoo, Numenta, NASA, etc. In this work we make a surprising claim. The majority of the individual exemplars in these datasets suffer from one or more of four flaws. Because of these four flaws, we believe that many published comparisons of anomaly detection algorithms may be unreliable, and more importantly, much of the apparent progress in recent years may be illusory. In addition to demonstrating these claims, with this paper we introduce the UCR Time Series Anomaly Archive. We believe that this resource will perform a similar role as the UCR Time Series Classification Archive, by providing the community with a benchmark that allows meaningful comparisons between approaches and a meaningful gauge of overall progress.

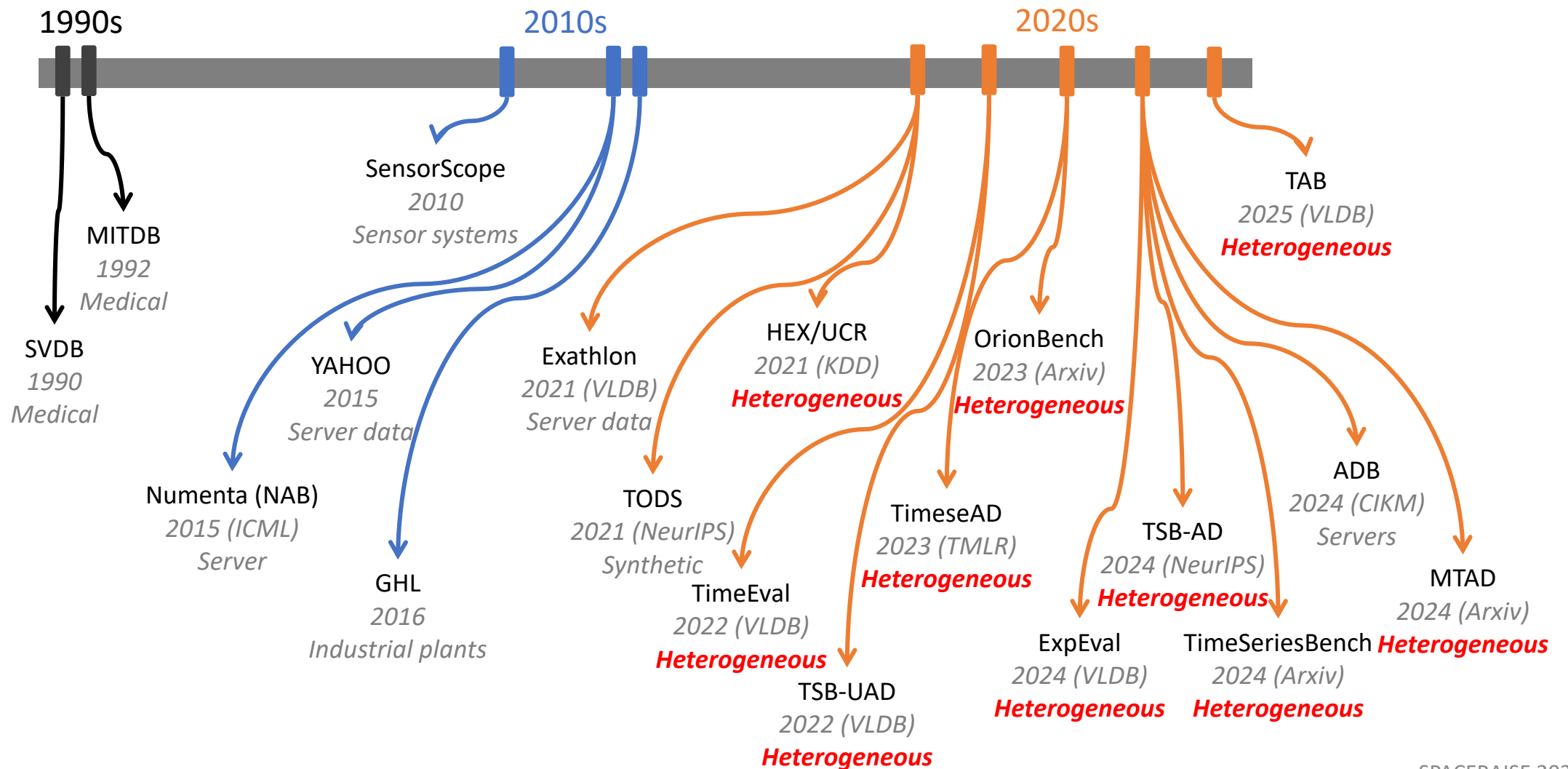
First proposal of a heterogeneous benchmark, and discussion of the limitations of current benchmarks

R. Wu et al. TKDE (2021) [18]

Anomaly Detection methods: *Existing benchmark*



Anomaly Detection methods: *Existing benchmark*



Anomaly Detection methods: *Existing benchmark*

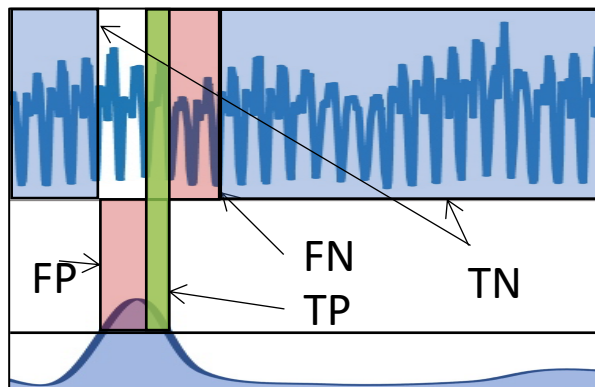
1990s

2010s

2020s

How should we **evaluate** the **accuracy** of time series anomaly detection?

Traditional point-wise and threshold-dependent measures are over-penalizing



orScope
2010
systems

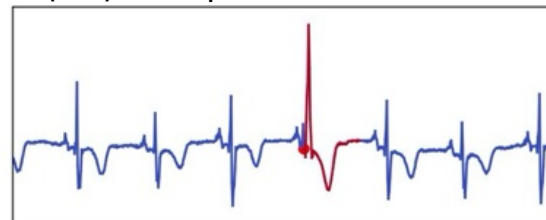
Exa
2021
Serv

plants

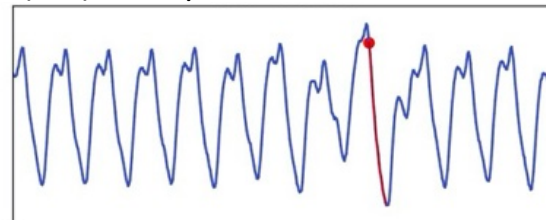
Can we **rely** on one **unique** method?

Observations from recent heterogeneous benchmarks (i.e., TSB-UAD, TSB-AD, TimeEval, and TAB) confirm that there are significant differences in performance across datasets

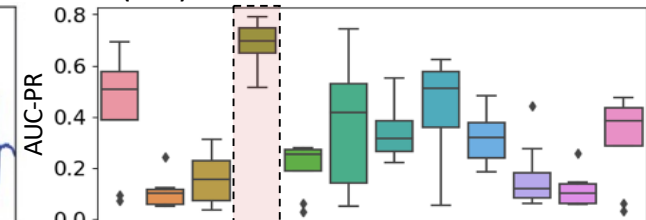
(a.1) Example from ECG dataset



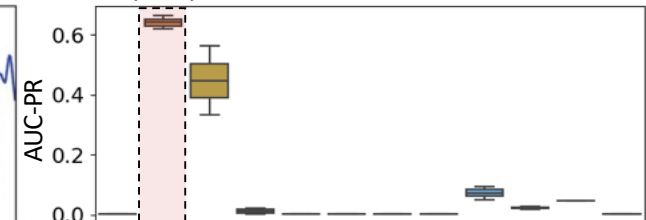
(b.1) Example from MGAB dataset



(a.2) ECG best detector: **NormA**



(b.2) MGAB best detector: **LOF**



(iv)
POUS

Anomaly Detection methods: *Existing benchmark*

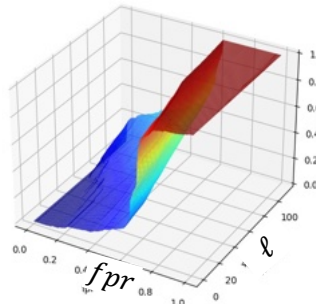
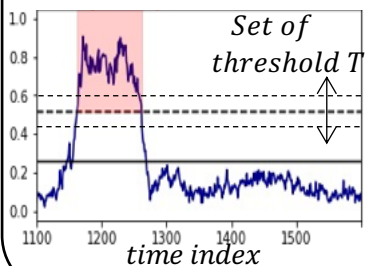
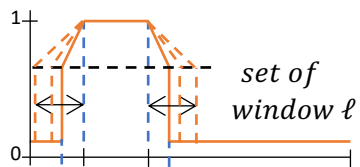
1990s

2010s

2020s

How should we **evaluate** the **accuracy** of time series anomaly detection?

Recent time series specific measures (such as VUS) have been introduced



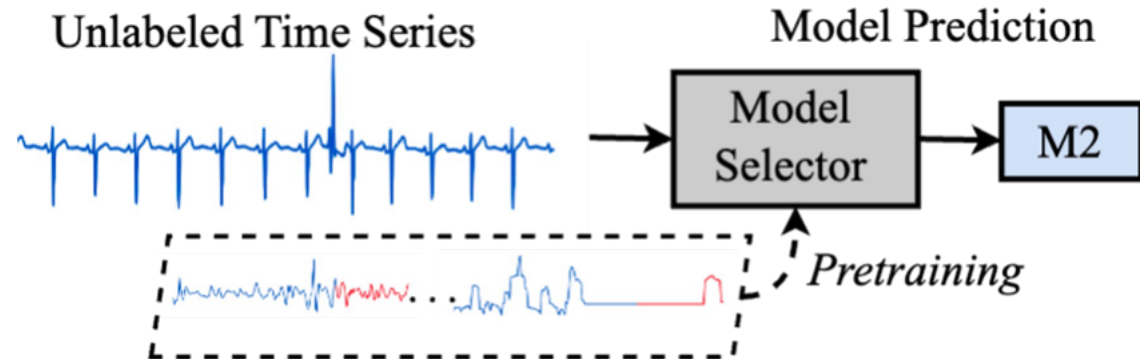
orScope
2010
systems

Exa
2021
Serv

plants

Can we **rely** on one **unique** method?

Model Selection and ensembling significantly outperform each individual detectors



(iv)
EUS

Anomaly Detection methods: *Existing benchmark*

1990s

2010s

2020s

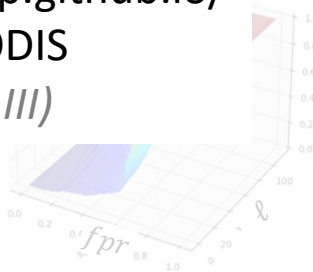
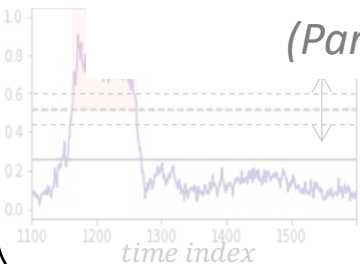
How should we **evaluate** the **accuracy** of time series anomaly detection?

Recent time series specific measures (such as VUS) have been

More info on previous lectures:



<https://bonioldp.github.io/TwinODIS>
(Part III)



orScope
2010
systems

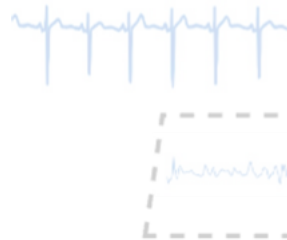
Exa
2021
Serv

plants

Can we **rely** on one **unique** method?

Model Selection and ensembling significantly outperform each individual detectors

Unlabeled Time Series



More info on previous lectures:

<https://bonioldp.github.io/TwinODIS>
(Part III)

Model Prediction



retraining

(iv)
POUS

Conclusion

If you are interested in anomaly detection in time series...

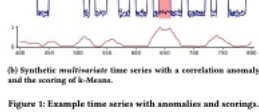
Anomaly Detection in Time Series: A Comprehensive Evaluation

Sebastian Schmid¹
Hasso Plattner Institute,
University of Potsdam
Potsdam, Germany
sebastian.schmid@hpi.de

Phillip Wenig¹
Hasso Plattner Institute,
University of Potsdam
Potsdam, Germany
phillip.wenig@hpi.de

Thorsten Papenbrock²
Philips University of Marburg
Marburg, Germany
papenbrock@informatik.uni-
marburg.de

ABSTRACT
Detecting anomalous subsequences in time series data is an important task in areas ranging from manufacturing processes over finance applications to health care monitoring. An anomaly can indicate important events, such as production faults, delivery bottlenecks, system defects, or heart flicker, and is therefore of central interest. Because time series are often large and exhibit complex patterns, data scientists have developed various specialized algorithms for the automatic detection of such anomalous patterns. The number and variety of anomaly detection algorithms has grown significantly in the past and, because many of these solutions have been developed independently and by different research communities, there is no comprehensive study that systematically evaluates and compares the different approaches. For this reason, choosing the best detection technique for a given anomaly detection task is a difficult challenge.



1 ANOMALY DETECTION WILDERNESS

<https://github.com/HPI-Information-Systems/TimeEval>

S. Schmidl et al. PVLDB (2022) [5]

TSB-UAD: An End-to-End Benchmark Suite for Univariate Time-Series Anomaly Detection

John Paparrizos¹
The Ohio State University
paparrizos@osu.edu

Yuhao Kang¹
University of Chicago
yuhao.kang@uchicago.edu

Paul Boniol²
University de Paris
paul.boniol@etu.u-paris.fr

Ruey S. Tsay¹
University of Chicago
rueytsay@chicago Booth.edu

Themis Palpanas¹
University de Paris & IUF
themis@uniparisdesarts.fr

Michael J. Franklin¹
University of Chicago
mjfranklin@uchicago.edu

ABSTRACT
The detection of anomalies in time series has gained ample academic and industrial attention. However, no comprehensive benchmark exists to evaluate time-series anomaly detection methods. It is common to use (i) proprietary or synthetic data, often biased to support particular claims; or (ii) a limited collection of publicly available datasets. Consequently, we often observe methods performing exceptionally well in one dataset but surprisingly poorly in another, creating an illusion of progress. To address the issues above, we thoroughly studied over one hundred papers to identify, collect, process, and systematically format datasets proposed in the past decades. We summarize our effort in TSB-UAD, a new benchmark to ease the evaluation of univariate time-series anomaly detection methods. Overall, TSB-UAD contains 13766 time series with labeled anomalies spanning different domains with high variability of anomaly types, ratios, and sizes. TSB-UAD includes 18 previously proposed datasets containing 1980 time series and we contribute two collections of datasets. Specifically, we generate 958 time series using a principled methodology for transforming 138 time-series classification datasets into time series with labeled anomalies. In addition, we present data transformations with which we introduce new anomalies, resulting in 10828 time series with varying complexity for anomaly detection. Finally, we evaluate 12 representative methods demonstrating that TSB-UAD is a robust resource for assessing anomaly detection methods. TSB-UAD provides a valuable, reproducible, and frequently updated resource to establish a leaderboard of time-series anomaly detection methods.

<https://github.com/TheDatumOrg/TSB-UAD>

J. Paparrizos et al. PVLDB (2022) [19]

Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress

Renjie Wu and Eamonn J. Keogh

Abstract—Time series anomaly detection has been a perennially important topic in data science, with papers dating back to the 1950s. However, in recent years there has been an explosion of interest in this topic, much of it driven by the success of deep learning in other domains and for other time series tasks. Most of these papers test on one or more of a handful of popular benchmark datasets, created by Yahoo, Numenta, NASA, etc. In this work we make a surprising claim. The majority of the individual exemplars in these datasets suffer from one or more of four flaws. Because of these four flaws, we believe that many published comparisons of anomaly detection algorithms may be unreliable, and more importantly, much of the apparent progress in recent years may be illusory. In addition to demonstrating these claims, with this paper we introduce the UCR Time Series Anomaly Archive. We believe that this resource will perform a similar role as the UCR Time Series Classification Archive, by providing the community with a benchmark that allows meaningful comparisons between approaches and a meaningful gauge of overall progress.

Index Terms—Anomaly detection, benchmark datasets, deep learning, time series analysis

1 INTRODUCTION
TIME series anomaly detection has been a perennially important topic in data science, with papers dating back to the dawn of computer science [1]. However, in the last five years there has been an explosion of interest in this topic, with at least one or two papers on the topic appearing each year in virtually every database, data mining, and machine learning conference, including SIGKDD [2], [3], ICDM [4], ICDE, SIGMOD, VLDB, etc. A large fraction of this increase in interest seems to be largely driven by researchers anxious to transfer the considerable success of deep learning in other domains and

<https://wu.renjie.im/research/anomaly-benchmarks-are-flawed/>

R. Wu et al. TKDE (2021) [18]

A review on outlier/anomaly detection in time series data

ANE BLÁZQUEZ-GARCÍA and ANGEL CONDE, Iberlan Technology Research Centre, Basque Research and Technology Alliance (BRTA), Spain
USUE MORI, Intelligent Systems Group (ISG), Department of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU), Spain
JOSE A. LOZANO, Intelligent Systems Group (ISG), Department of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU), Spain and Basque Center for Applied Mathematics (BCAM), Spain

Recent advances in technology have brought major breakthroughs in data collection, enabling a large amount of data to be gathered over time and thus generating time series. Mining this data has become an important task for researchers and practitioners in the past few years, including the detection of outliers or anomalies that may represent errors or events of interest. This review aims to provide a structured and comprehensive state-of-the-art on outlier detection techniques in the context of time series. To this end, a taxonomy is presented based on the main aspects that characterize an outlier detection technique.

Additional Key Words and Phrases: Outlier detection, anomaly detection, time series, data mining, taxonomy, software

1 INTRODUCTION
Recent advances in technology allow us to collect a large amount of data over time in diverse research areas. Observations that have been recorded in an orderly fashion and which are correlated in time constitute a time series. Time series data mining aims to extract all meaningful knowledge from this data, and several mining tasks (e.g., classification, clustering, forecasting, and outlier detection) have been considered in the literature [Ealing and Agon 2012; Fu 2011; Ratanamahatana et al. 2010].
Outlier detection has become a field of interest for many researchers and practitioners and is now one of the main tasks of time series data mining. Outlier detection has been studied in a variety of application domains such as credit card fraud detection, intrusion detection in cybersecurity, or fault diagnosis in industry. In particular, the analysis of outliers in time series data examines anomalous behaviors across time [Gupta et al. 2014a]. In the first study on this topic, which was conducted by Fox [1972], two types of outliers in univariate time series were defined: type I, which affects a single observation; and type II, which affects both a particular observation and the subsequent observations. This work was first extended to four outlier types [Tsay 1988], and then to the case of multivariate time series [Tsay et al. 2000]. Since then, many definitions of the term outlier and numerous detection methods have been proposed in the literature. However, to this day, there is still no consensus on the terms used [Caretto et al. 2019]; for example, outlier observations are often referred to as anomalies, discordant observations, discords, exceptions, aberrations, surprises, peculiarities or contaminants.

Authors' addresses: Ane Blázquez-García, ablagarc@iberlan.es, Angel Conde, aconde@iberlan.es, Iberlan Technology Research Centre, Basque Research and Technology Alliance (BRTA), P.I.M. Anizkorriaketa, 2, Arcazio-Mondragón, 48940, Spain; Ute Mori, ute.mori@ehu.es, Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU), Alameda de Deusto, 45, 48940, Spain.

A. Blazquez-Garcia et al. ACM Computing Survey (2021) [24]

Conclusion

Time Series Anomaly Detection

Paul Boniol, Qinghua Liu, John Paparrizos, and Themis Palpanas.



Video (*EDBT 2023 Tutorial*)



<https://www.youtube.com/watch?v=96869qimXAA&t=1s>



Slides (*VLDB 2024 Tutorial*)



<https://drive.google.com/file/d/1Vyz6H0E16lpbVZXgtiZVnU9le8zAJaog/view>



Short Survey (*KDD 2025*)



<https://dl.acm.org/doi/10.1145/3711896.3736565>



Survey (*2025 preprint*)



<https://arxiv.org/pdf/2412.20512>

Anomaly Detection

Sebastian Schmidl
Hasso Plattner Institut
University of Potsdam
Potsdam, Germany
sebastian.schmidl@hpi.de

ABSTRACT

Detecting anomalous subsequences is a non-trivial task in areas ranging from finance applications to health care. It indicates important events, such as system failures, system defects, or heart file interest. Because time series are often noisy, data scientists have developed various methods for the automatic detection of anomalies. This paper surveys the number and variety of anomaly detection techniques that have been developed independently and compares the different approaches to provide a taxonomy of the state-of-the-art anomaly detection techniques. In the paper, we evaluate the effectiveness, efficiency, and results should ease the algorithm selection process in new research directions.

This comprehensive, scientific state-of-the-art anomaly detection survey re-implemented 71 anomaly detection techniques and evaluated them on 91 real-world datasets. The datasets have been selected from different domains and their characteristics are summarized in the paper. The survey also provides a taxonomy of the techniques and their common characteristics. The survey is available as a preprint on arXiv and as a book chapter in the proceedings of the VLDB 2024 conference.

http://
Informa

S. Schmidl et al. PVLDB (2022) [5]

J. Paparrizos et al. PVLDB (2022) [19]

R. Wu et al. TKDE (2021) [18]

A. Blazquez-Garcia et al. ACM Computing Survey (2021) [24]

Conclusion

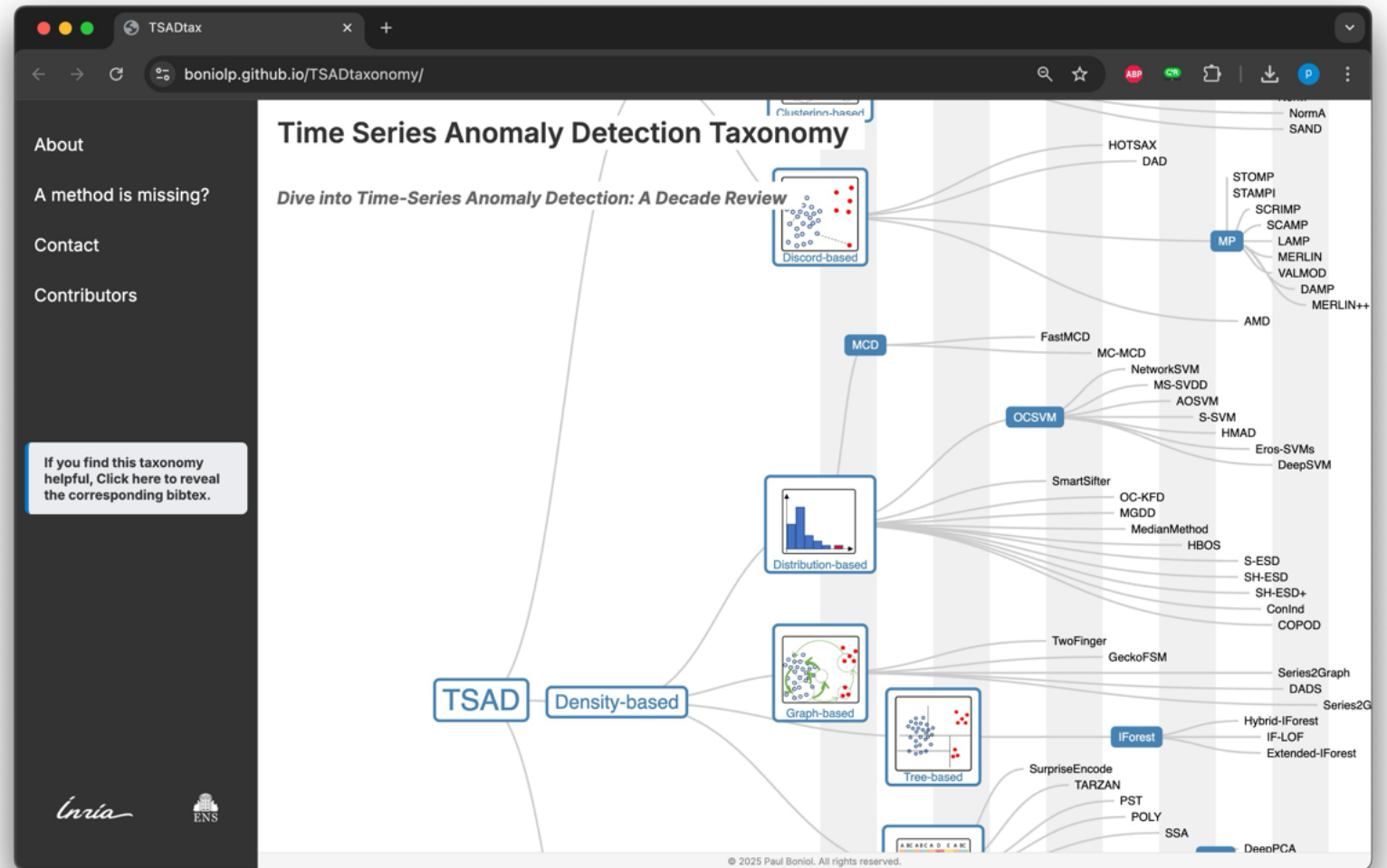
Online interactive
Taxonomy



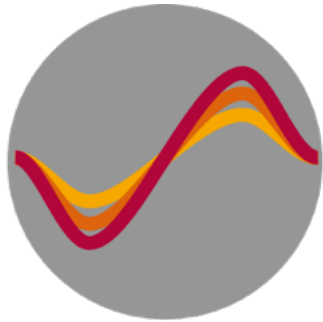
<https://bonioldp.github.io/TSADtaxonomy/>



<https://github.com/bonioldp/TSADtaxonomy>



Conclusion

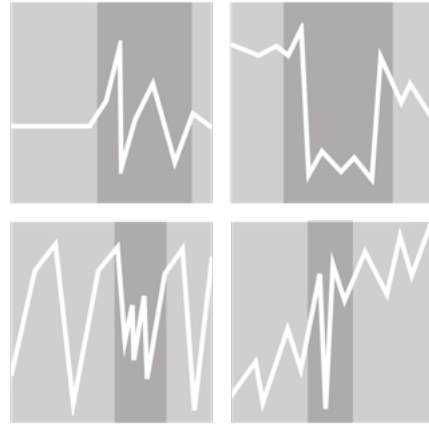


TimeEval

PyPI v1.1.4, Python 3.7 – 3.9

```
Pip install TimeEval
```

github.com/TimeEval/TimeEval

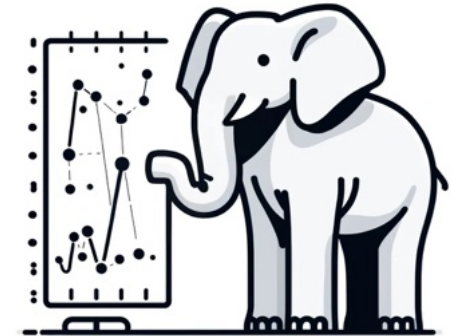


PyPI v0.0.3, Python 3.8 – 3.12

```
Pip install tsb-uad
```

github.com/TheDatumOrg/TSB-UAD

TSB-AD

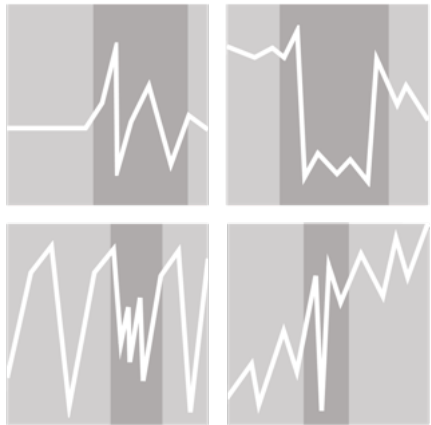


No PyPI, Python 3.9

```
Pip install TSB-AD
```

github.com/TheDatumOrg/TSB-AD

Conclusion



PyPI v0.0.3, Python 3.8 – 3.12

```
Pip install tsb-uad
```

github.com/TheDatumOrg/TSB-UAD

```
import os
import numpy as np
import pandas as pd
from TSB_UAD.models.iforest import IForest
from TSB_UAD.models.feature import Window
from TSB_UAD.utils.slidingWindows import find_length
from TSB_UAD.vus.metrics import get_metrics

df = pd.read_csv('data/benchmark/ECG/MBA_ECG805_data.out', header=None).to_numpy()
data = df[:, 0].astype(float)
label = df[:, 1]

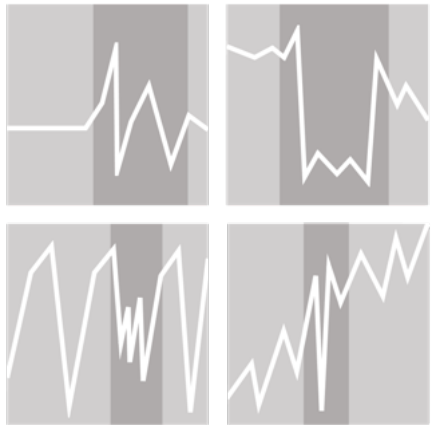
slidingWindow = find_length(data)
X_data = Window(window = slidingWindow).convert(data).to_numpy()

clf = IForest(n_jobs=1)
clf.fit(X_data)
score = clf.decision_scores_

score = MinMaxScaler(feature_range=(0,1)).fit_transform(score.reshape(-1,1)).ravel()
score = np.array([score[0]]*math.ceil((slidingWindow-1)/2) + list(score) + [score[-1]]*(s

results = get_metrics(score, label, metric="all", slidingWindow=slidingWindow)
for metric in results.keys():
    print(metric, ':', results[metric])
```

Conclusion



PyPI v0.0.3, Python 3.8 – 3.12

```
Pip install tsb-uad
```

github.com/TheDatumOrg/TSB-UAD

```
import os
import numpy as np
import pandas as pd
from TSB_UAD.models.iforest import IForest
from TSB_UAD.models.feature import Window
from TSB_UAD.utils.slidingWindows import find_length
from TSB_UAD.vus.metrics import get_metrics

df = pd.read_csv('data/benchmark/ECG/MBA_ECG805_data.out', header=None).to_numpy()
data = df[:, 0].astype(float)
label = df[:, 1]

slidingWindow = find_length(data)
X_data = Window(window = slidingWindow).convert(data).to_numpy()

clf = IForest(n_jobs=1)
clf.fit(X_data)
score = clf.decision_scores_

score = MinMaxScaler(feature_range=(0,1)).fit_transform(score.reshape(-1,1)).ravel()
score = np.array([score[0]]*math.ceil((slidingWindow-1)/2) + list(score) + [score[-1]]*((s

results = get_m
for metric in r
print(metri
```

```
AUC_ROC : 0.9216216369841076
AUC_PR : 0.6608577550833885
Precision : 0.7342093339374717
Recall : 0.4010891089108911
F : 0.5187770129662238
Precision_at_k : 0.4010891089108911
Rprecision : 0.7486112853253205
Rrecall : 0.3097733542316151
RF : 0.438214653167952
R_AUC_ROC : 0.989123018780308
R_AUC_PR : 0.9435238401582703
VUS_ROC : 0.9734357459251715
VUS_PR : 0.8858037295594041
Affiliation_Precision : 0.9630674176380548
Affiliation_Recall : 0.9809813654809071
```

Conclusion

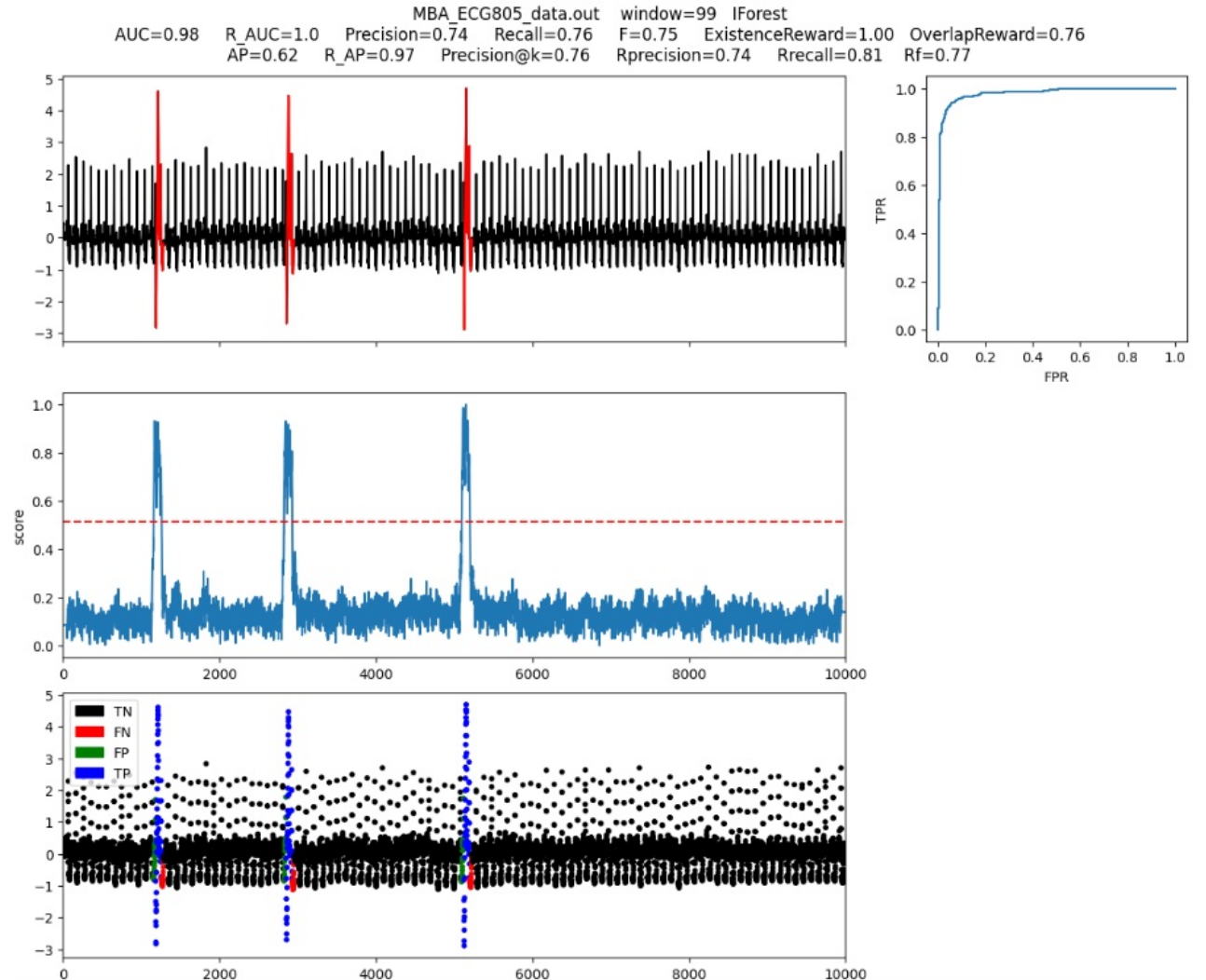


PyPI v0.0.3, Python 3.8 – 3.12

```
Pip install tsb-uad
```

github.com/TheDatumOrg/TSB-UAD

```
modelName='IForest'  
clf = IForest(n_jobs=1)  
x = X_data  
clf.fit(x)  
score = clf.decision_scores_  
score = MinMaxScaler(feature_range=(0,1)).fit_transform(score.reshape(-1,1)).ravel()  
score = np.array([score[0]*math.ceil((slidingWindow-1)/2) + list(score) + [score[-1]]*((slidingWindow-1)//2)])  
  
plotFig(data, label, score, slidingWindow, fileName=name, modelName=modelName)
```



Thanks for attending!



paul.boniol@inria.fr



<https://boniold.github.io/>