

**ZABBIX**

SUMMIT  
2024

# Anomaly Detection in Zabbix: Present and Future



**Aleksandrs Kalimulins**

Product Team

# Introduction

What is AI used for in IT infrastructure monitoring?



AI has become a crucial tool in IT infrastructure monitoring, providing a more proactive and efficient way to manage and optimize systems. Here are some key uses of AI in this area:

1. **Anomaly Detection:** AI algorithms can quickly identify deviations from normal behavior in IT systems. These anomalies might indicate potential issues like hardware failures, cyberattacks, or configuration errors. By detecting anomalies early, AI helps prevent these issues from escalating into serious problems.



# What is an anomaly?



Image credit: Herluf Bidstrup

# What is an anomaly?

## anomaly **noun**

anom·a·ly

ə-'nä-mə-lē ◀▶

**plural anomalies**

[Synonyms of \*anomaly\*](#) >

- 1** : something different, abnormal, peculiar, or not easily classified : something **anomalous**  
| They regarded the test results as an *anomaly*.
- 2** : deviation from the common rule : **IRREGULARITY**

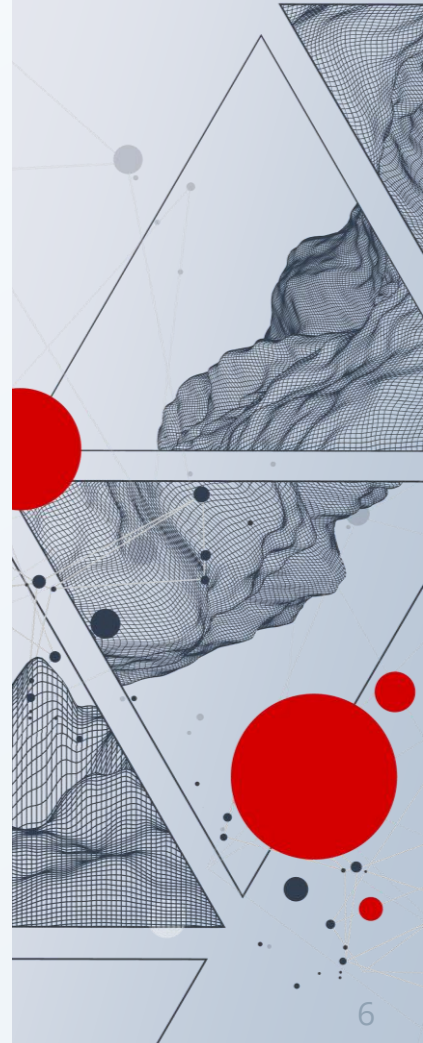
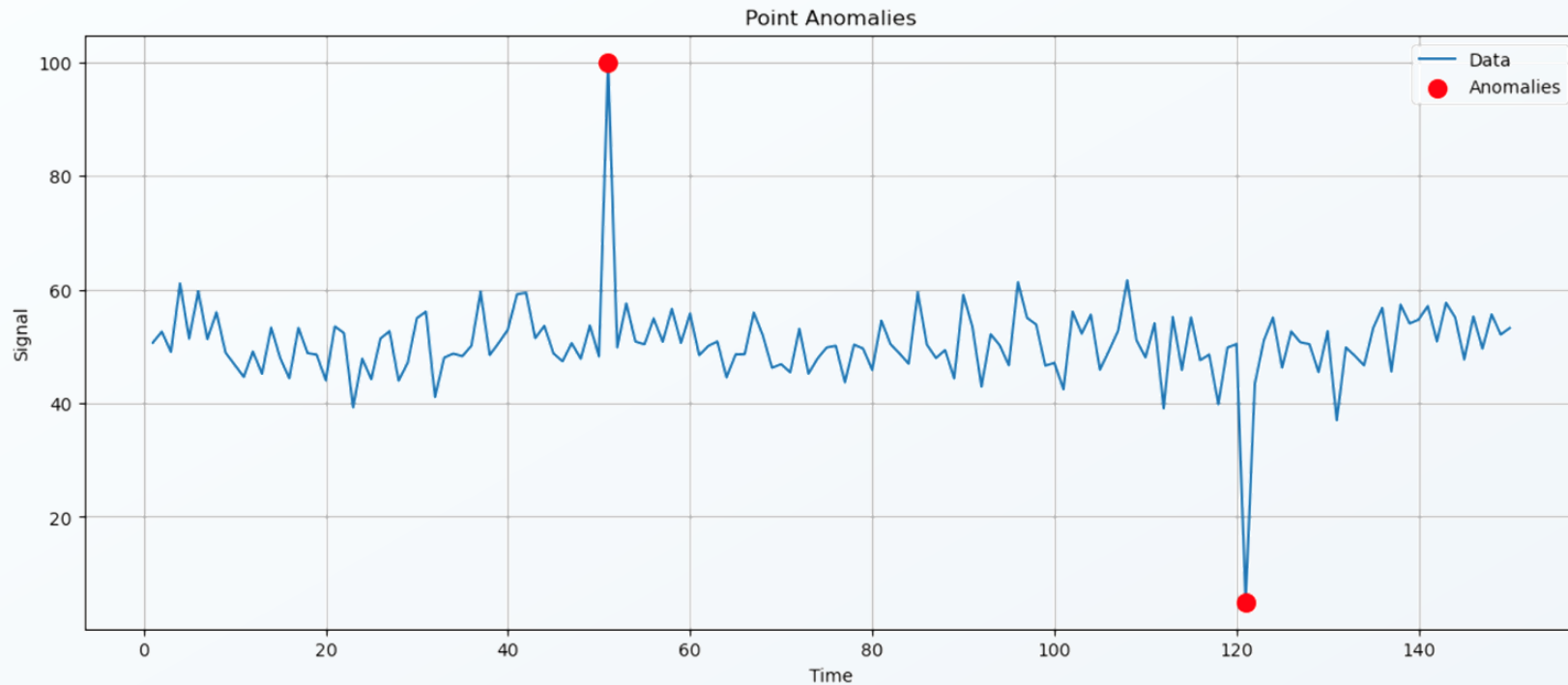


# What is an anomaly in time series?

*An anomaly in a time series is a **rare** or **unexpected** point or sequence occurring over a specified time interval, often considered unusual or undesirable.*

# Anomaly types

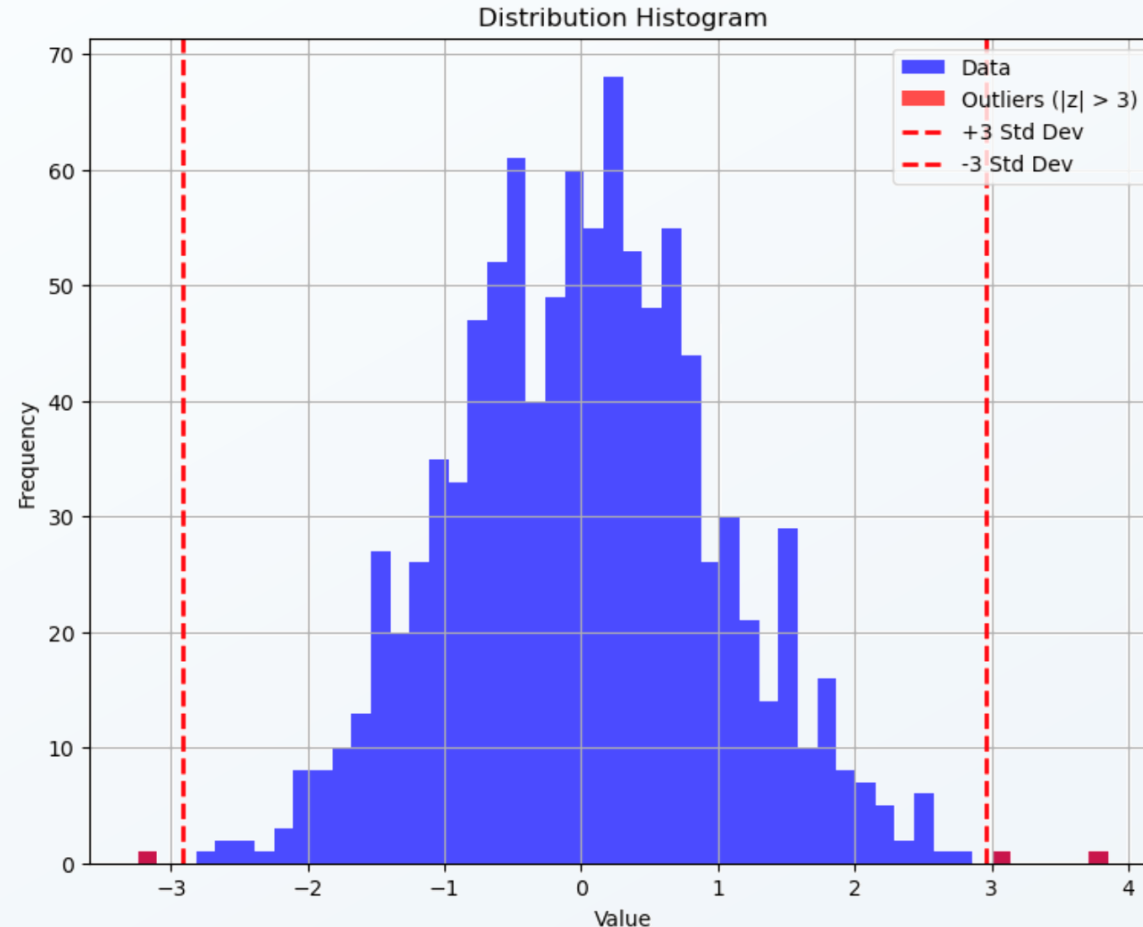
## Point-based



# Anomaly types

## Point-based

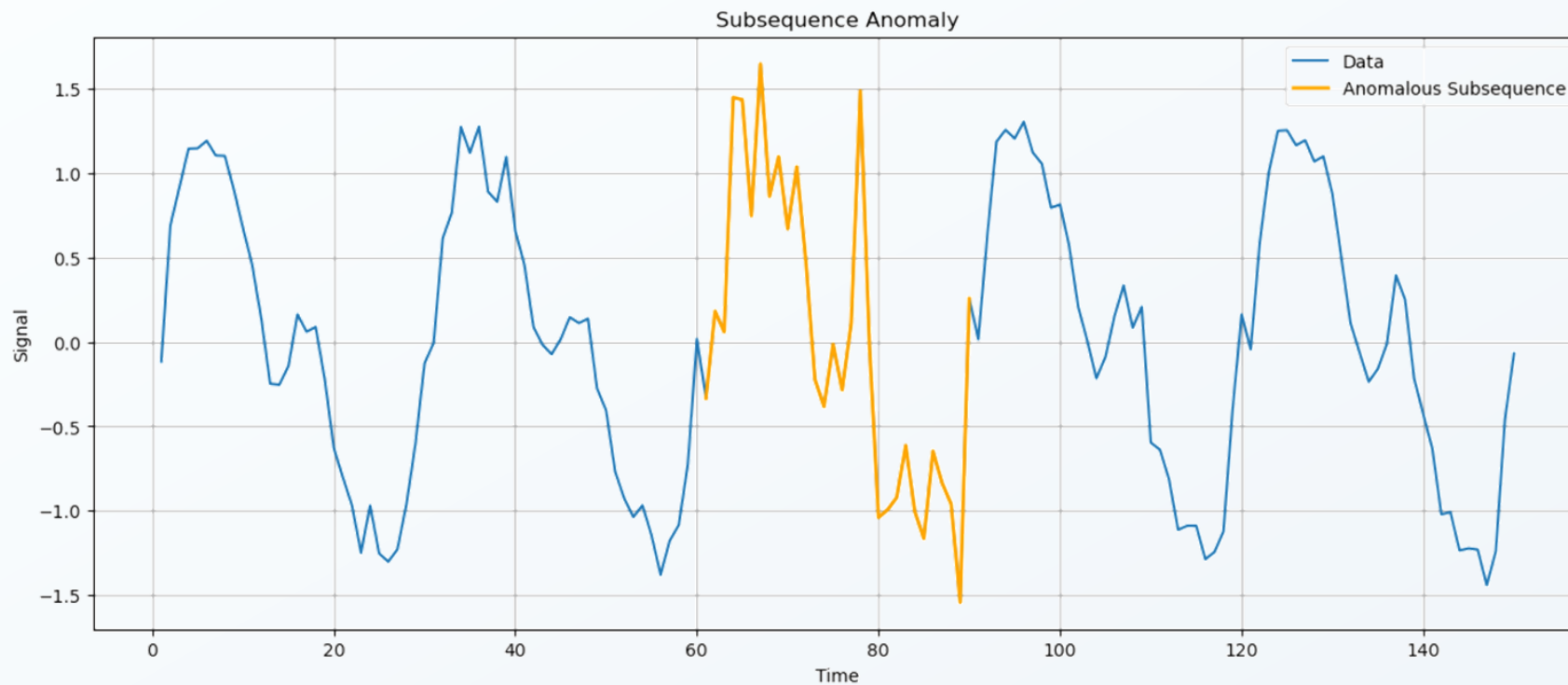
- ▶ Short duration
- ▶ Easy to spot
- ▶ Easy to detect
  - sometimes even with `stddevsamp()`





# Anomaly types

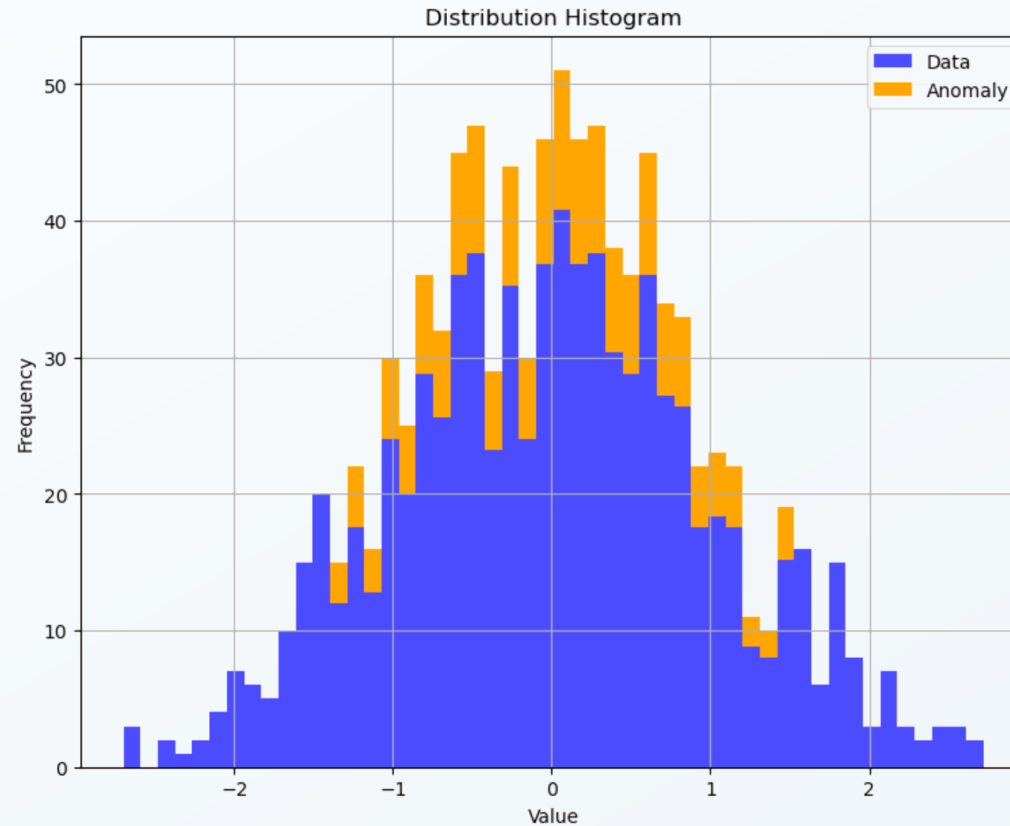
## Subsequence-based



# Anomaly types

## Subsequence-based

- ▶ Longer duration
- ▶ Harder to spot
- ▶ Difficult to detect



# How does Zabbix do it?

## Trigger functions

- ▶ `stddevpop()`, `stddevsamp()`, `mad()`
- ▶ `varpop()`, `varsamp()`
- ▶ `baselinedev()`
- ▶ `trendstl()`



# How does Zabbix do it?

## `trendstl()` - features

- ▶ implements STL anomaly detection algorithm
- ▶ decomposes data into trend, seasons, residual
- ▶ data must have pronounced seasonal pattern

# How does Zabbix do it?

## trendstl() - shortcomings

- ▶ 7 parameters (need a data science degree)
  - `trendstl(/host/key,100h:now/h-10h,100h,2h,3,"mad",1001)`
- ▶ careful choice of seasonality
- ▶ no support for multiple seasons
- ▶ subsequence-based anomalies out of scope

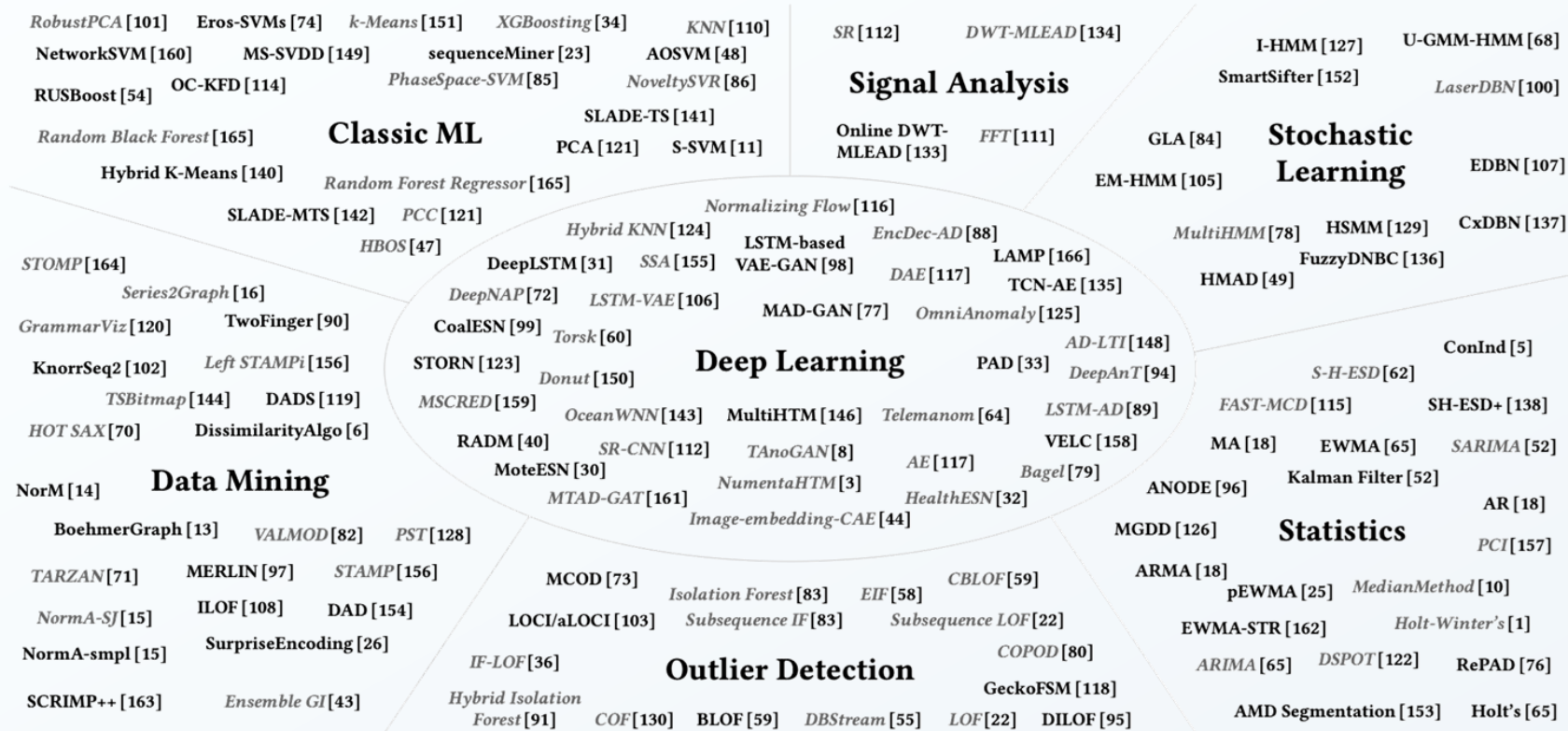
# More detection methods

## Requirements

- ▶ Performance and efficiency
- ▶ Easy to use (the less knobs and switches the better)
- ▶ Easy to understand and interpret results
- ▶ Subsequence-based anomalies
- ▶ Multiple seasonalities



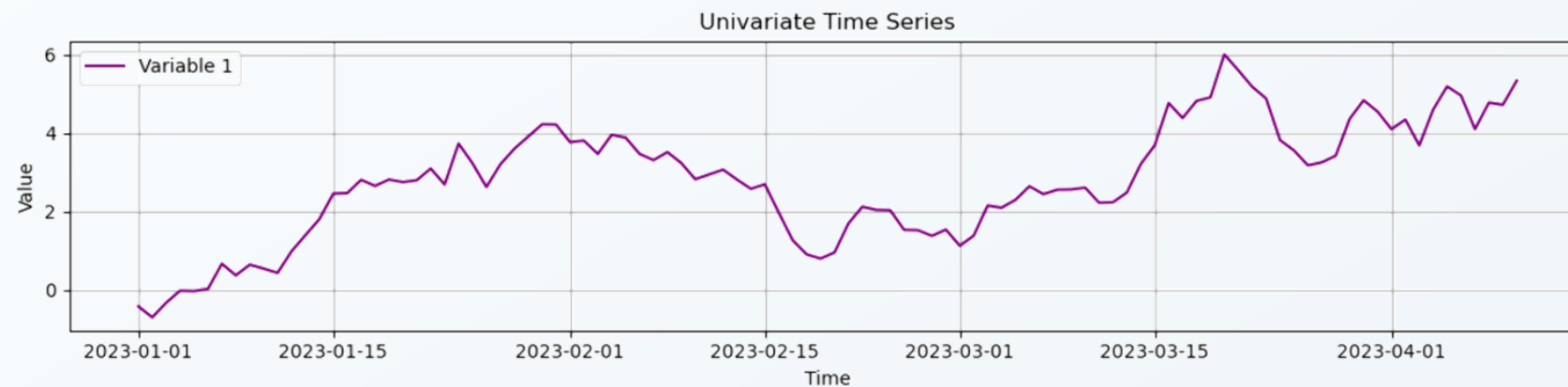
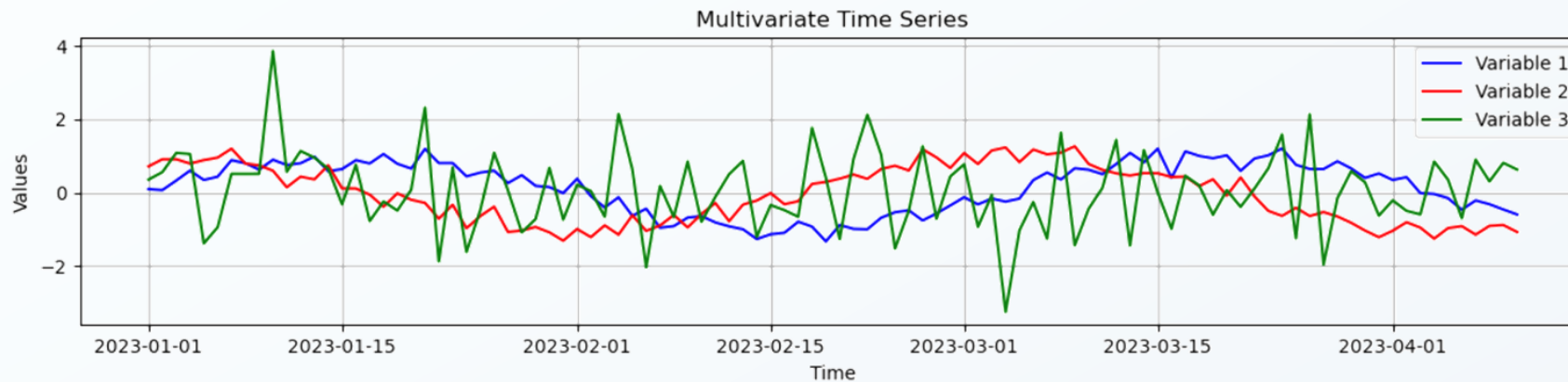
# Anomaly detection methods



"Anomaly Detection in Time Series: A Comprehensive Evaluation"

# Anomaly detection methods

## Input types



# Anomaly detection methods

## Training types

- ▶ **Supervised**
  - Manual training
- ▶ **Semi-supervised**
  - Training on a clean data
- ▶ **Unsupervised**
  - No training



Image credit: Herluf Bidstrup



# Anomaly detection methods

Quantum Alg 2023 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SYSTEMS

Ming-Chao Gu  
Qin,<sup>1</sup> Xiao-  
State Key  
Beijing University  
<sup>2</sup>Comprehensive Research Center

LOW-COUNT TIME S

Philip Renz<sup>\*,1</sup> Kurt Cutajar<sup>†</sup> Nia

Published as a conference paper at ICLR 2019

DEEP ANOMALY DETECTION WITH OUTLIER EXPOSURE

Dan Hendrycks University of California, Berkeley hendrycks@berkeley.edu

Mantas Mazeika University of Chicago mantas@ttic.edu

Thomas Dietterich Oregon State University tgd@oregonstate.edu

ABSTRACT

It is important to detect anomalous inputs when deploying machine learning systems. The use of larger and more complex inputs in deep learning magnifies the difficulty of distinguishing between anomalous and in-distribution examples. At the same time, diverse image and text data are available in enormous quantities. We propose leveraging these data to improve deep anomaly detection by training anomaly detectors against an auxiliary dataset of outliers, an approach we call Outlier Exposure (OE). This enables anomaly detectors to generalize and detect unseen anomalies. In extensive experiments on natural language processing and small- and large-scale vision tasks, we find that Outlier Exposure significantly improves detection performance. We also observe that cutting-edge generative models trained on CIFAR-10 may assign higher likelihoods to SVHN images than to CIFAR-10 images, we use OE to mitigate this issue. We also analyze the flexibility and robustness of Outlier Exposure, and identify characteristics of the auxiliary dataset that improve performance.

1 INTRODUCTION

Machine Learning systems in deployment often encounter data that is unlike the model's training data. This can occur in discovering novel astronomical phenomena, finding unknown diseases, detecting sensor failure. In these situations, models that can detect anomalies (Liu et al., 2015; Emmott et al., 2013) are capable of correctly flagging unusual examples for human intervention, carefully proceeding with a more conservative fallback policy.

Behind many machine learning systems are deep learning models (Krizhevsky et al., 2012) which can provide high performance in a variety of applications, so long as the data seen at test time is similar to the training data. However, when there is a distribution mismatch, deep neural network classifiers tend to give high confidence predictions on anomalous test examples (Nguyen et al., 2015). This can invalidate the use of prediction probabilities as calibrated confidence estimates (Guo et al., 2017), and makes detecting anomalous examples doubly important.

Several previous works seek to address these problems by giving deep neural network class a means of assigning anomaly scores to inputs. These scores can then be used for detecting out-of-distribution (OOD) examples (Hendrycks & Gimpel, 2017; Lee et al., 2018; Liu et al., 2015). These approaches have been demonstrated to work surprisingly well for complex input spaces, as images, text, and speech. Moreover, they do not require modeling the full data distribution; instead, can use heuristics for detecting unmodeled phenomena. Several of these methods detect unmodeled phenomena by using representations from only in-distribution data.

In this paper, we investigate a complementary method where we train models to detect unmodeled data by learning cues for whether an input is unmodeled. While it is difficult to model the full

Journal of Artificial Intelligence Research 46 (2013) 235-262

Exploring the Use of Data-Driven Approaches for Anomaly Detection in the Internet of Things (IoT) Environment

Eleonora Achiluzzi, Menglu Li, Md Fahd Al Georgy, and Rasha M. Alkhatib

Toronto Metropolitan University

{eachiluzzi, menglu.li, mgec}

Published at

Laboratory, Technische Universität Berlin, Germany  
Cognitive Center  
Cancer Center

gen. Dep. of Computer Science  
7077 Göttingen, Germany

Abstract—The Internet of Things (IoT) is a system that connects physical computing devices, sensors, software, and other technologies. Data can be collected, transferred, and exchanged with other devices over the network without requiring human interactions. One challenge is the development of IoT faces is the existence of anomaly data in the network. Therefore, research on anomaly detection in the IoT environment has become popular and necessary in recent years. This survey provides an overview to understand the current

TANOgan: Time Series Anomaly Detection with Generative Adversarial Networks

Md Abul Bashar  
School of Computer Science  
Centre for Data Science  
Queensland University of Technology  
Brisbane, Queensland 4000, Australia  
Email: m1.bashar@qut.edu.au

Richi Nayak  
School of Computer Science  
Centre for Data Science  
Queensland University of Technology  
Brisbane, Queensland 4000, Australia  
Email: r.nayak@qut.edu.au

Abstract—Anomaly detection in time series data is a significant problem faced in many application areas such as manufacturing, medical imaging and cyber-security. Recently, Generative Adversarial Networks (GAN) have gained attention for generation and anomaly detection in image domain. In this paper, we propose a novel GAN-based unsupervised method called TANOgan for detecting anomalies in time series when a small number of data points are available. We evaluate TANOgan with 46 real-world time series datasets that cover a variety of domains. Extensive experimental results show that TANOgan performs better than traditional and neural network models.

1. INTRODUCTION

The ubiquitous use of networked sensors and actuators in places like smart buildings, factories, power plants and data centres as well as the emergence of the Internet of Things (IoT) have resulted in generating substantial amounts of time series data. These data can be used to continuously monitor the working conditions of these environments to detect anomalies.

Proc. of the International Workshop on User Understanding from Big Data

Develop End-to-End Anomaly Detection Framework for Time Series Forecasting

Emanuele Mengoli  
École Polytechnique  
Paris, France  
em.mengoli@gmail.com

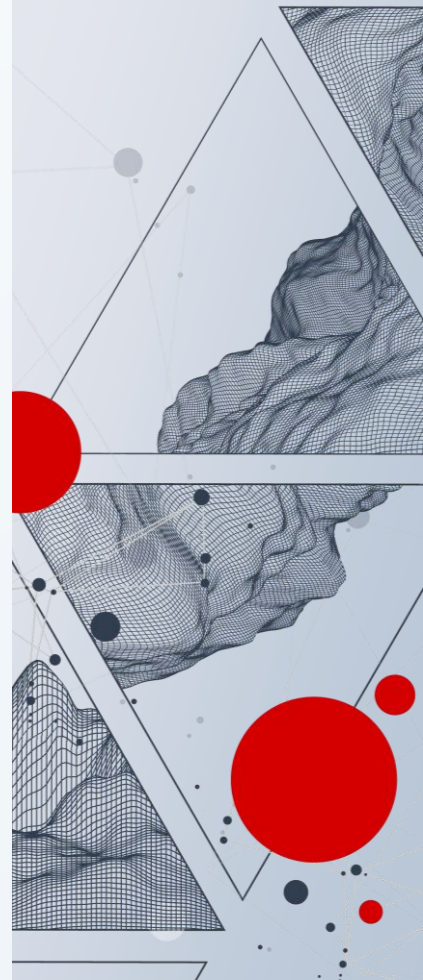
Zhiyuan Yao  
Cisco Meraki  
Paris, France  
zhiyuan@cisco.com

Abstract—Anomaly detection plays a crucial role in ensuring network robustness. However, implementing intelligent alerting systems becomes a challenge when considering scenarios in which anomalies can be caused by both malicious and non-malicious events, leading to the difficulty of determining anomaly patterns. The lack of labeled data in the computer networking domain further exacerbates this issue, impeding the development of robust models capable of handling real-world scenarios. To address this challenge, in this paper, we propose an end-to-end anomaly detection model development pipeline. This framework makes it possible to consume user feedback and enable continuous user-centric model performance evaluation and optimization. We demonstrate the efficacy of the framework by way of introducing and benchmarking a new forecasting model – named *Lachesis* – on a real-world networking problem. Experiments have demonstrated the robustness and effectiveness of the two proposed versions of *Lachesis* compared with other models proposed in the literature. Our findings underscore the potential for improving the performance of data-driven products and optimization. We demonstrate the efficacy of the framework by way of introducing and benchmarking a new forecasting model – named *Lachesis* – on a real-world networking problem. Experiments have demonstrated the robustness and effectiveness of the two proposed versions of *Lachesis* compared with other models proposed in the literature. Our findings underscore the potential for improving the performance of data-driven products and optimization.

quantifying responsibility straightforward exercise, it significantly when faced causal factors. The identification of great interest to users and supervised methods require truth data to establish causation. In this paper, we present allows continuously evaluate anomaly detection model. tion of our framework is an abnormally high amount c networking switches level. The development of an ability to continuously quantify root causes presents a non-trivial challenge. In this paper, we present a framework for root cause analysis (RCA) that leverages the power of large language models (LLMs) to analyze network logs and identify potential root causes. We begin by reprogramming the backbone language models kept intact. We begin by reprogramming the backbone language models kept intact.

ABSTRACT

Time series forecasting holds significant importance in many real-world systems and has been extensively studied. Unlike natural language processing and computer vision (CV), where a single large model can tackle multiple models for time series forecasting are often specialized, necessitating signs for different tasks and applications. While pre-trained foundation models have made impressive strides in NLP and CV, their development in domains has been constrained by data sparsity. Recent studies have revealed that large language models (LLMs) possess robust pattern recognition abilities over complex sequences of tokens. However, the challenge of effectively aligning the modalities of time series data and natural language processing (NLP) remains. In this work, we present TIME-LLM, a framework for repurposing LLMs for general time series forecasting. We begin by reprogramming the backbone language models kept intact. We begin by reprogramming the backbone language models kept intact.





# Anomaly detection methods

## Types

- ▶ **Statistical methods**
  - Z-score, Moving Average, ARIMA, ...
- ▶ **Machine learning and data analysis methods**
  - STL, FFT-based anomaly detection, ...
- ▶ **Deep learning methods**
  - LSTM autoencoders, RNN, ...

# Deep learning – is it any good?

## Deep learning methods

- ▶ Autoencoders
- ▶ Recurrent Neural Networks (RNNs)
- ▶ Generative Adversarial Networks (GANs)
- ▶ Convolutional Neural Networks (CNNs)
- ▶ LLMs (even!)

# Deep learning – is it any good?

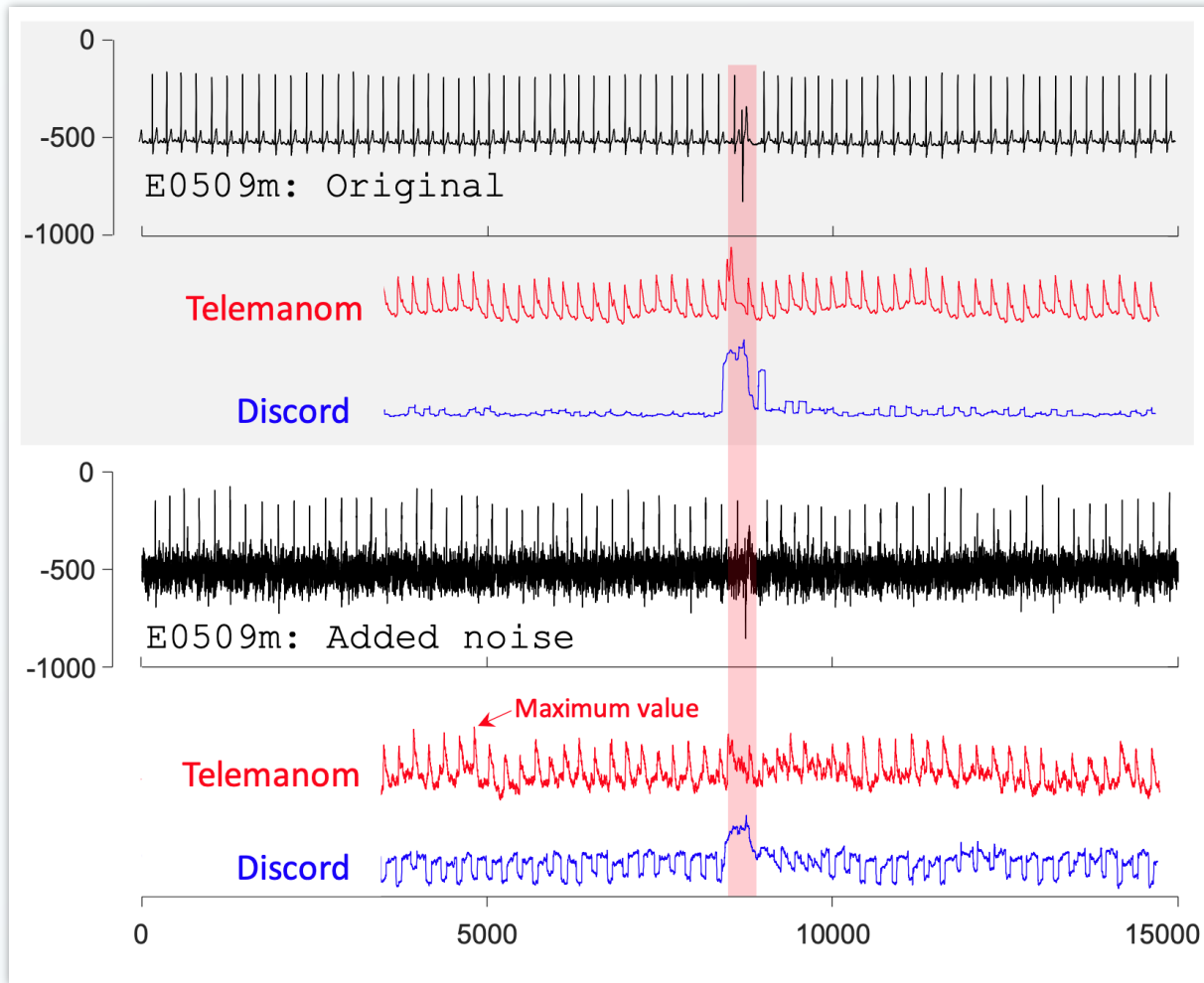
*“most anomaly detection algorithms, especially ones based on deep learning, have ten or more parameters”*

Wu, R., et al. "Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress" (2020)

*“we are not aware of a single paper that presents forceful reproducible evidence that deep learning outperforms much simpler methods”*

Wu, R., et al. "Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress" (2020)

# Deep learning – is it any good?



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



# Deep learning – is it any good?

*“We found that deep learning approaches are not (yet) competitive despite their higher processing effort on training data”*

Schmidl, S., et al. “Anomaly Detection in Time Series: A Comprehensive Evaluation.” (2022)

*“Our experiments showed that the classical machine learning methods ... outperform the deep learning methods.”*

Rewicki, F., et al. “Is it worth it? Comparing six deep and classical methods for unsupervised anomaly detection in time series” (2023)

# Deep learning – is it any good?

*“The experiments showed that the statistical approaches perform best on univariate time-series ... They also require less computation time compared to the other two classes.*

*Although deep learning approaches have gained huge attention by the ... community in the last years, our results have revealed that they are not ... able to achieve the accuracy values of the statistical methods on the univariate time-series benchmarks”*

Braei, M., et al. “Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art” (2020)

# Deep learning – is it any good?

*“... those [deep learning] methods, though potentially useful ..., do not bring much additional value for the task of TAD [anomaly detection] and their complexity is definitely not justified.”*

*What is even more worrisome, is that they managed to create up to now an illusion of progress”*

M. Saquib Sarfraz, et al. “Position: Quo Vadis, Unsupervised Time Series Anomaly Detection?” (2024)

# Deep learning – is it any good?

## Zabbix requirements

- ▶ ~~Performance and efficiency~~
- ▶ ~~Easy to use (the less knobs and switches the better)~~
- ▶ ~~Easy to understand and interpret results~~
- ▶ Subsequence-based anomalies
- ▶ Multiple seasonalities



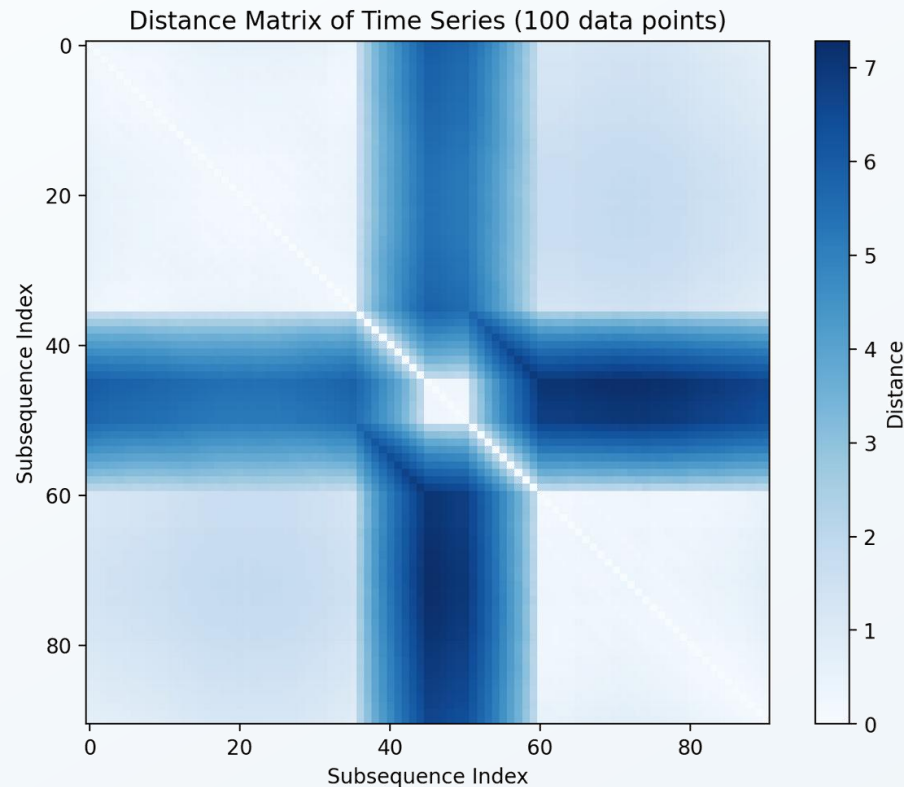
# Distance-based methods

Learn. Algorithm	TL	OOM	ERR	AUC-ROC all datasets
● Sub-LOF [22]	2 %	0 %	0 %	
▼ GrammarViz [120]	3 %	0 %	0 %	
▲ DWT-MLEAD [134]	0 %	0 %	0 %	
● VALMOD [82]	1 %	9 %	11 %	
● SAND [17]	5 %	1 %	22 %	
● Left STAMPi [156]	2 %	0 %	1 %	

“Anomaly Detection in Time Series: A Comprehensive Evaluation”

# Matrix Profile

- ▶ Method, not an algorithm
  - STAMP, STOMP, SCRIMP, SCRIMP++, SCAMP, VALMOD, MERLIN, ...
- ▶ Distance-based
  - Calculates nearest neighbours
- ▶ Subsequences of constant length
  - Motifs – similar subsequences
  - Discords – anomalies
- ▶ Fast on long subsequences
- ▶ Predictable time

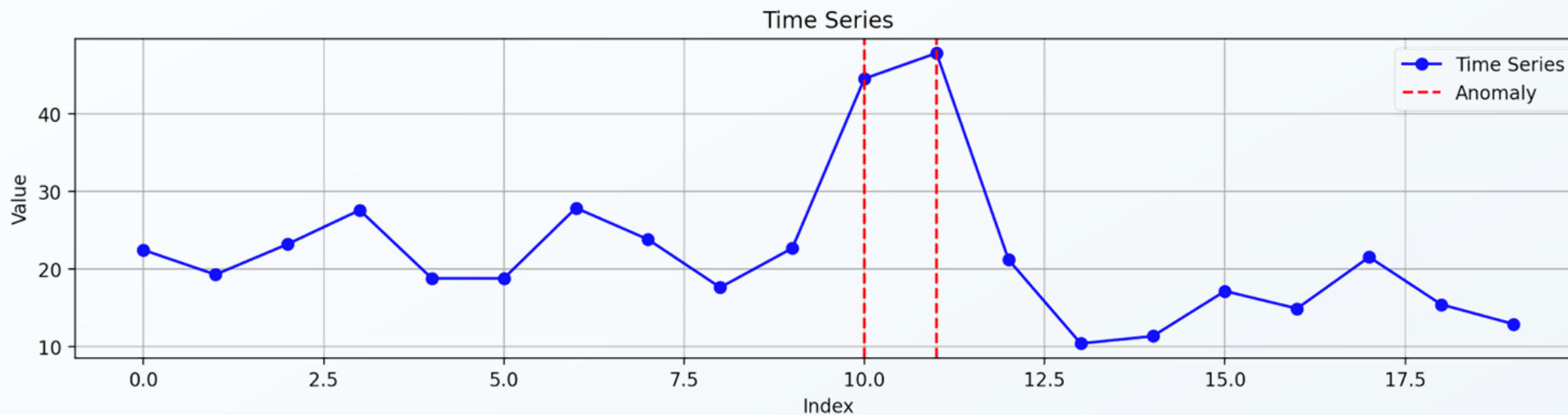


# Matrix Profile

- ▶ Applicable for subsequence and point-based anomalies
- ▶ Easily parallelizable
- ▶ No false positives
- ▶ No tuning parameters
- ▶ Used in many domains
  - IT infrastructure, IoT, space and satellites, medicine, seismology, industrial equipment, water distribution, ...

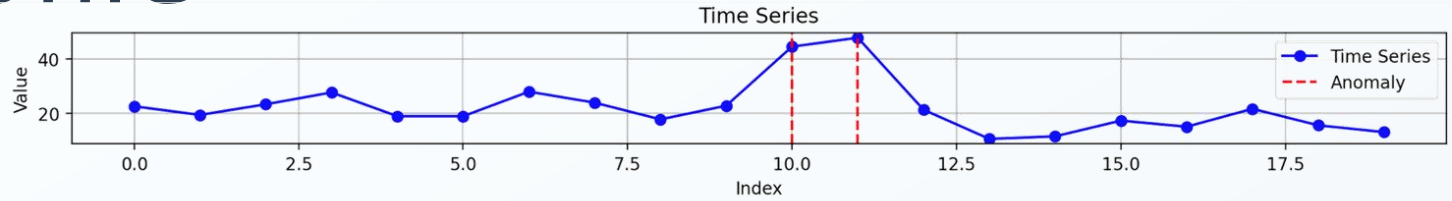
# Matrix Profile

## Calculation





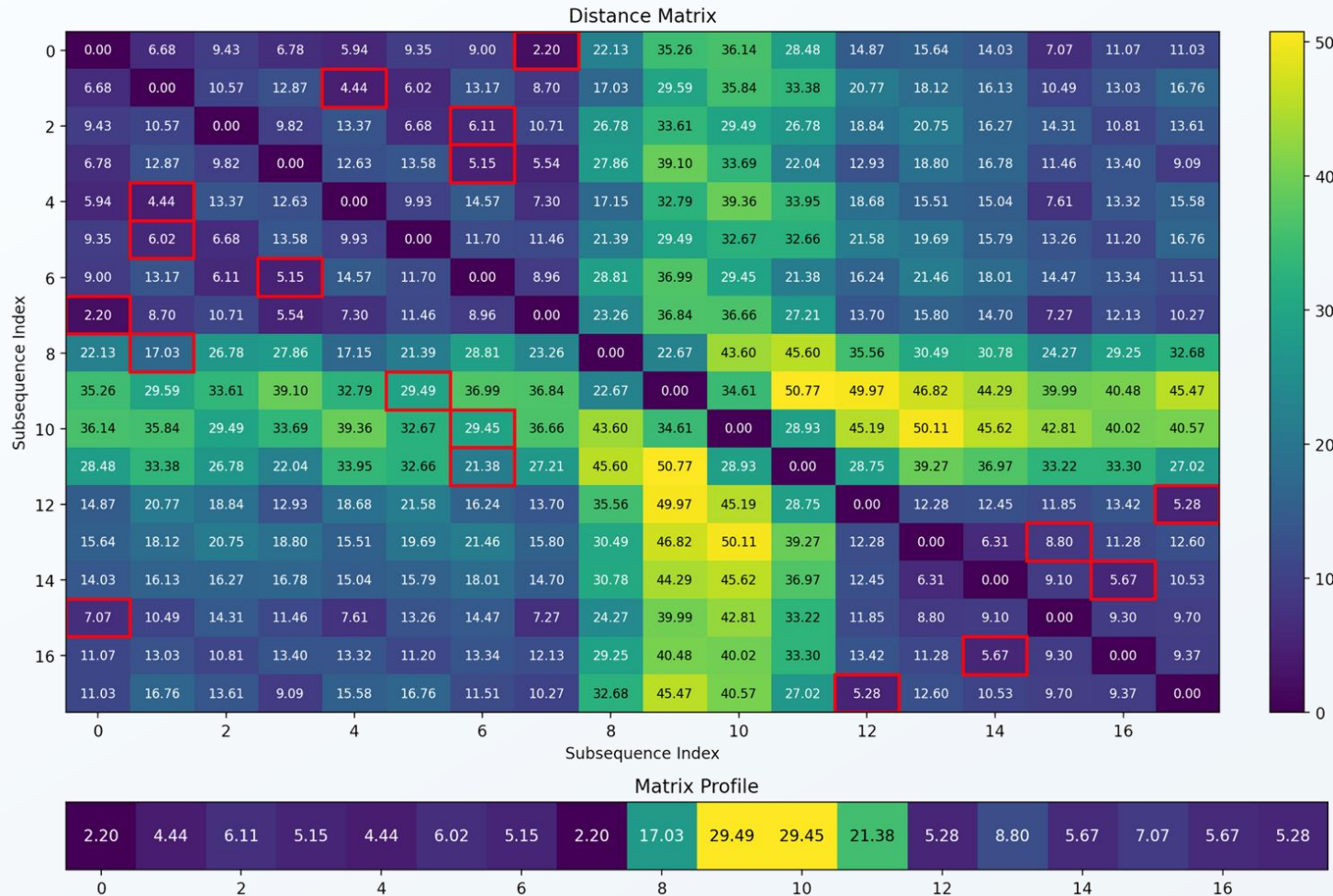
# Matrix Profile



## Calculation

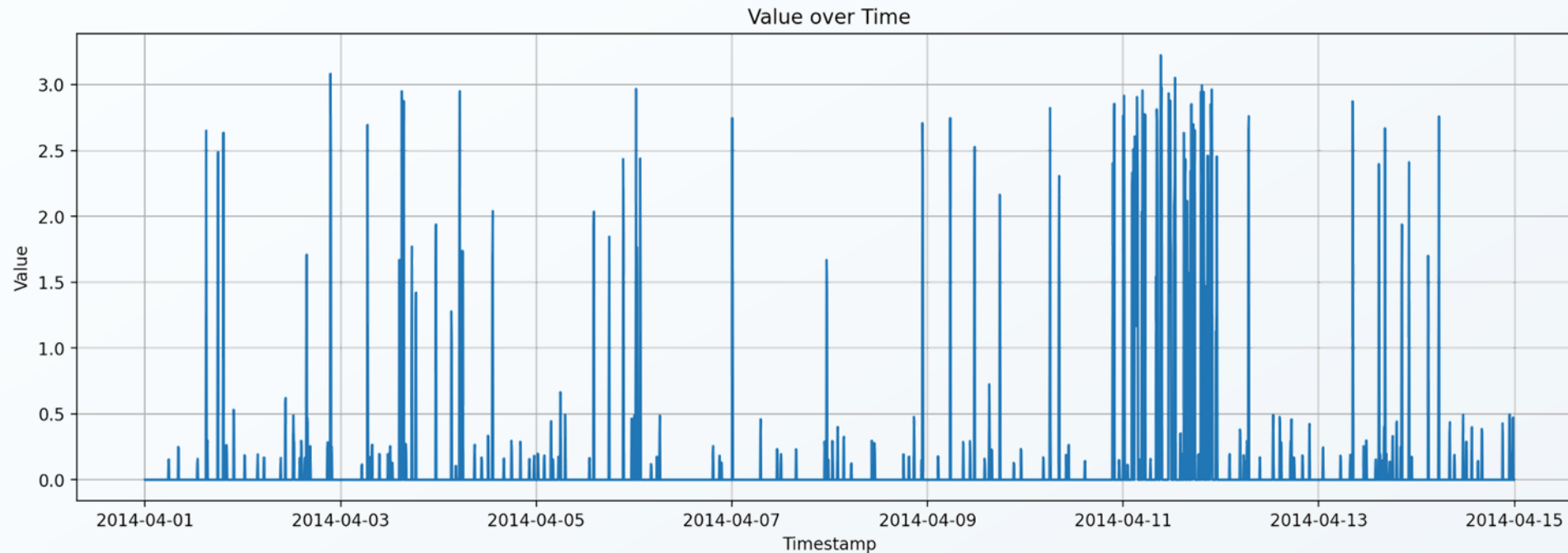
(naive approach)

- ▶ Get distances between all subsequences
- ▶ Get minimum for every row
- ▶ Compose profile



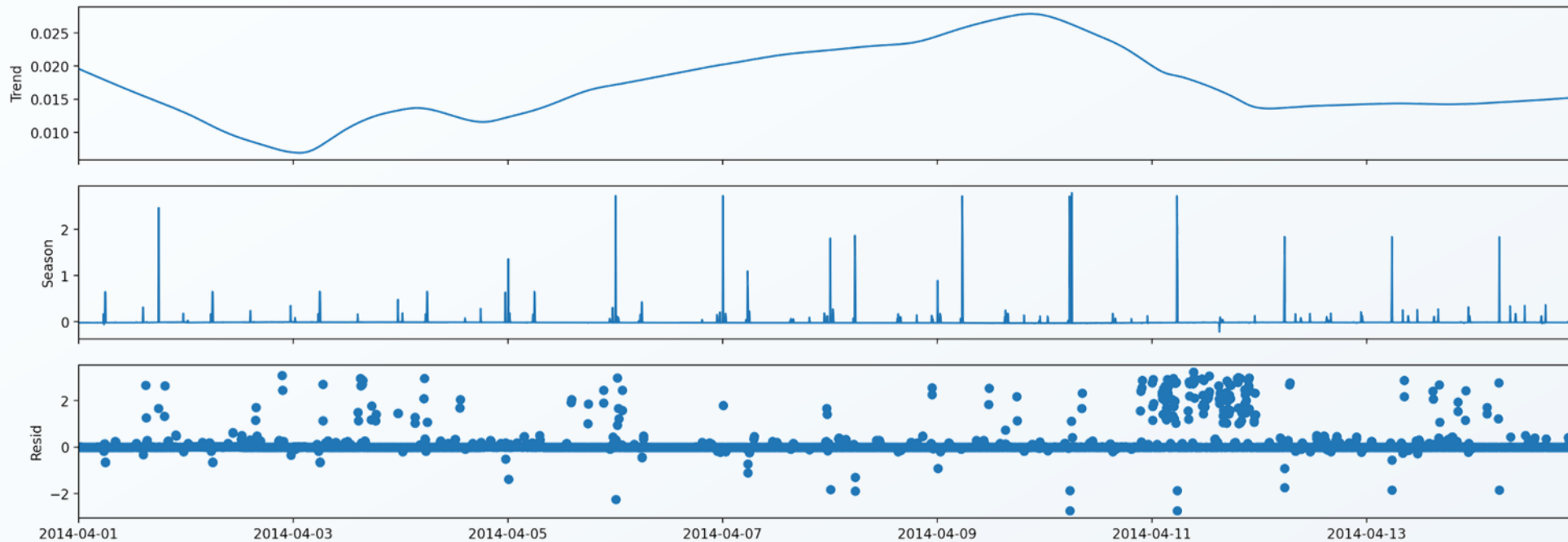
# Matrix Profile - Examples

## Raw data - load



# Matrix Profile - Examples

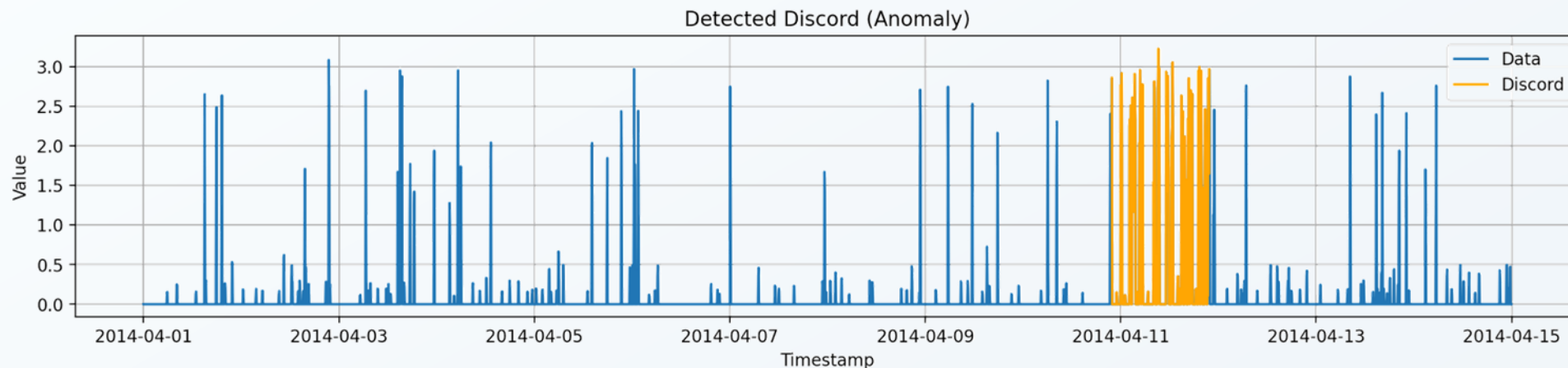
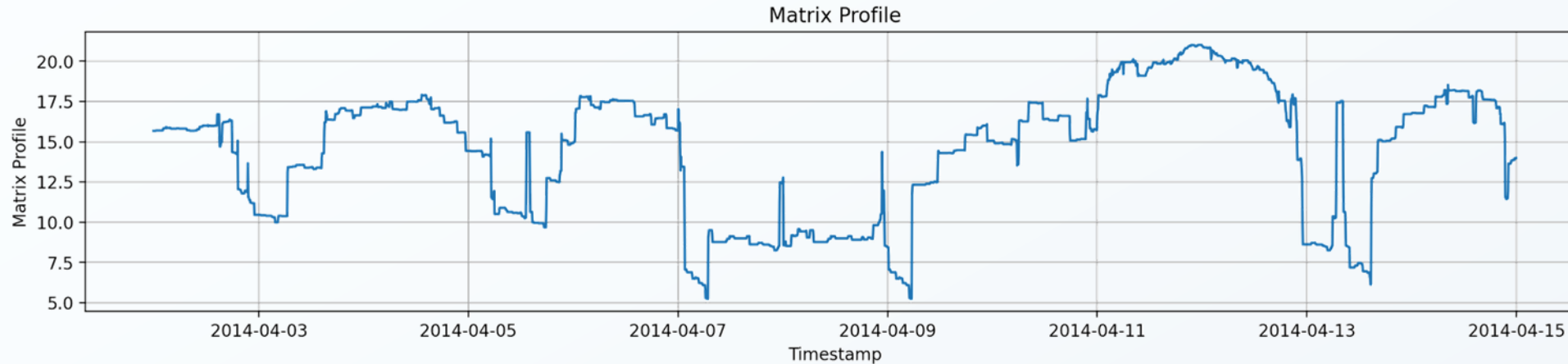
## STL analysis – not good





# Matrix Profile - Examples

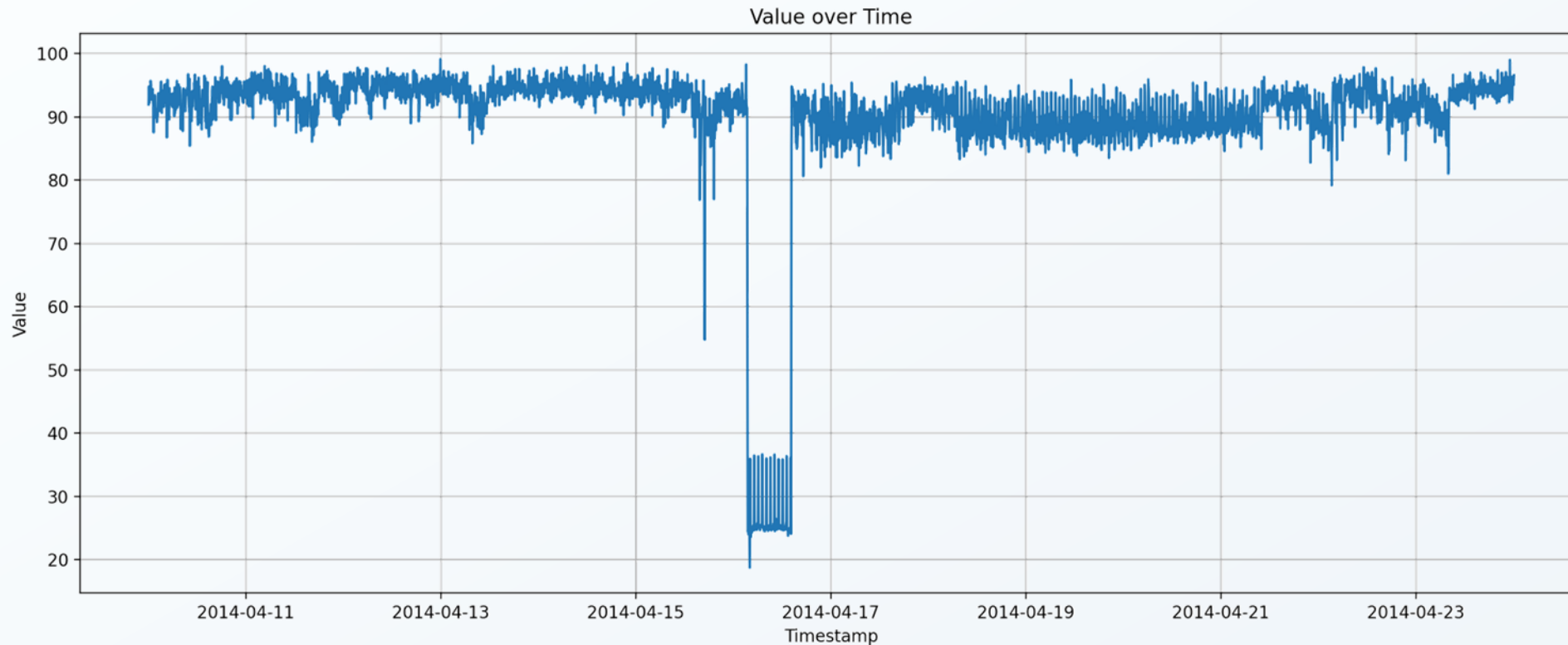
## Matrix Profile analysis





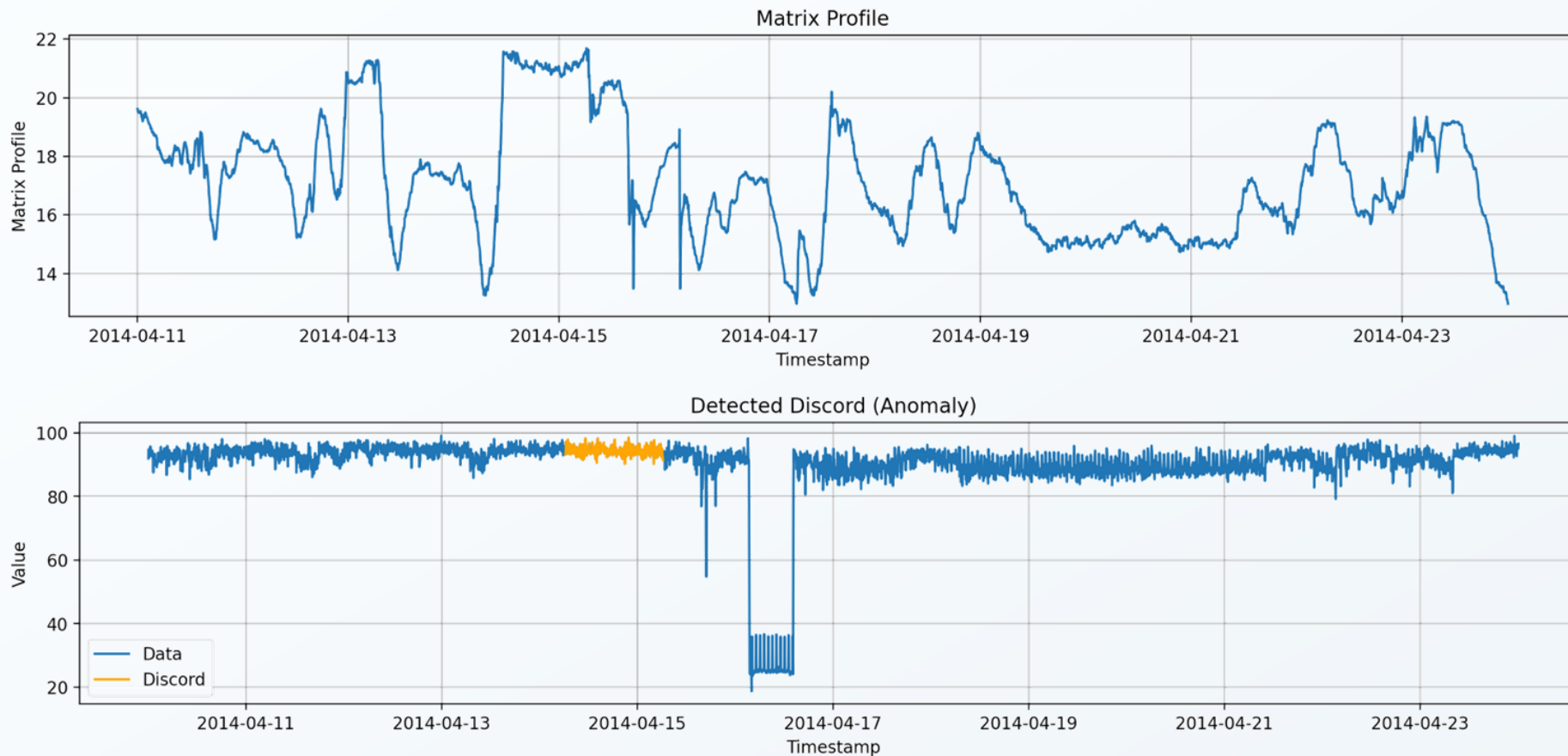
# Matrix Profile - Examples

## Raw data – CPU utilization



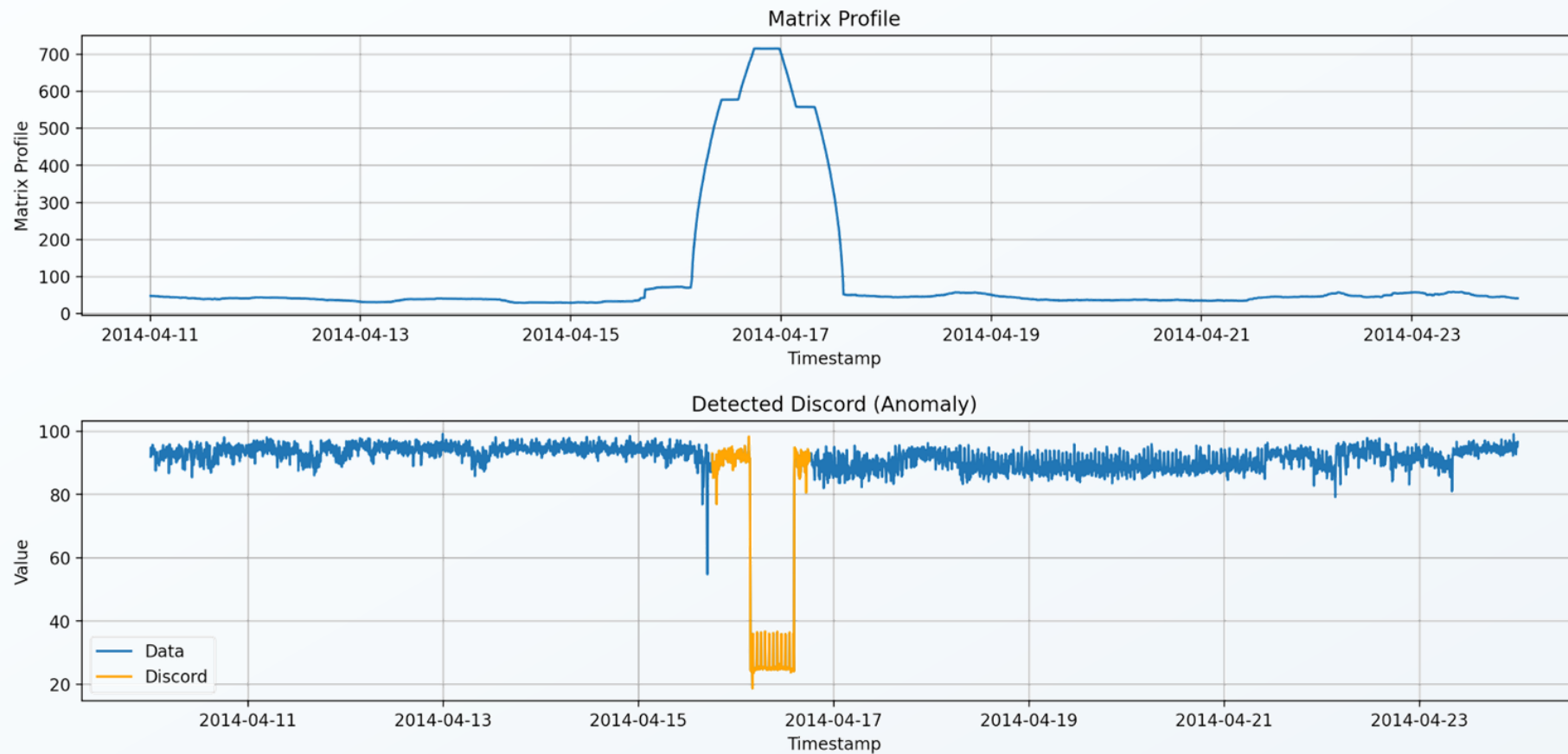
# Matrix Profile - Examples

## Matrix Profile analysis: Z-normalization



# Matrix Profile - Examples

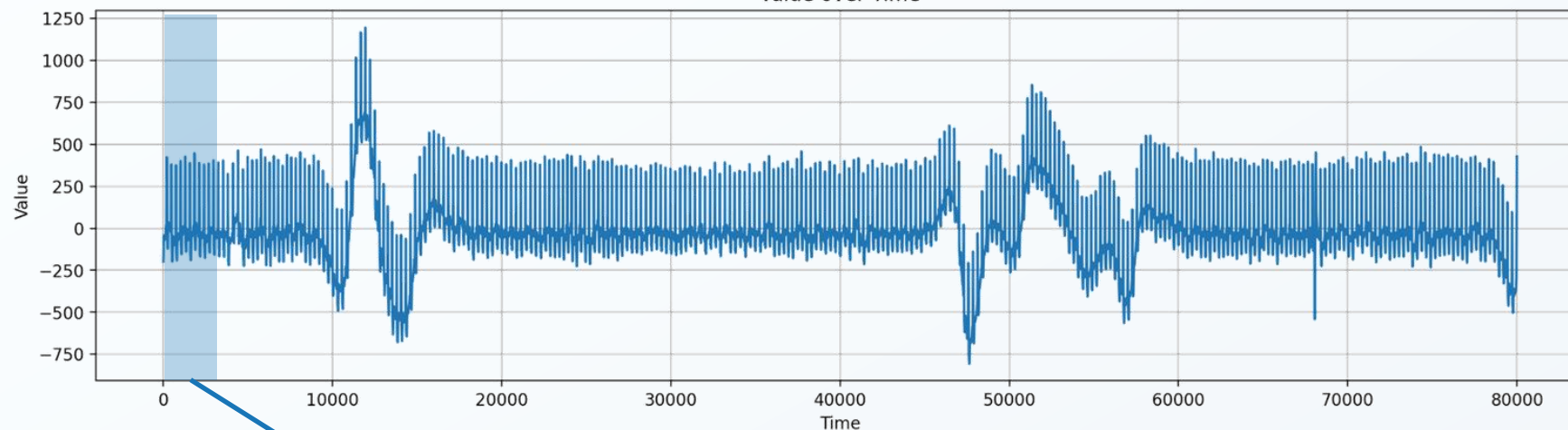
## Matrix Profile analysis: Z-normalization off



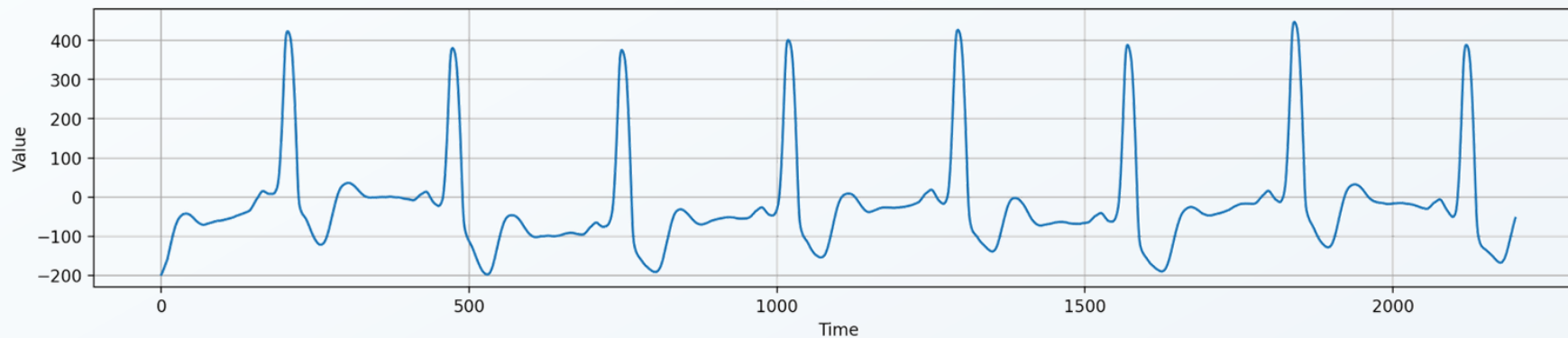
# Matrix Profile - Examples

## Raw data - ECG

Value over Time



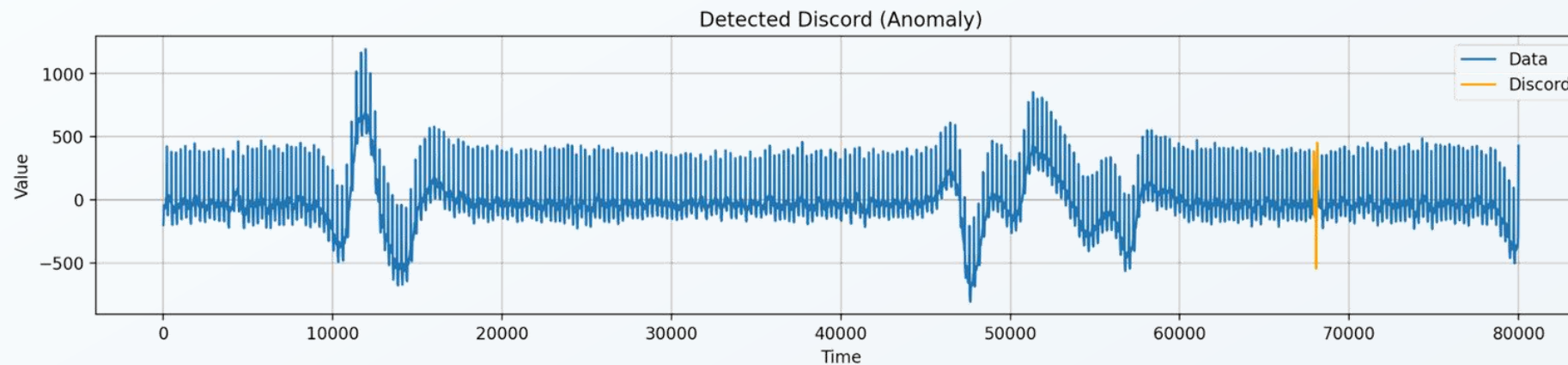
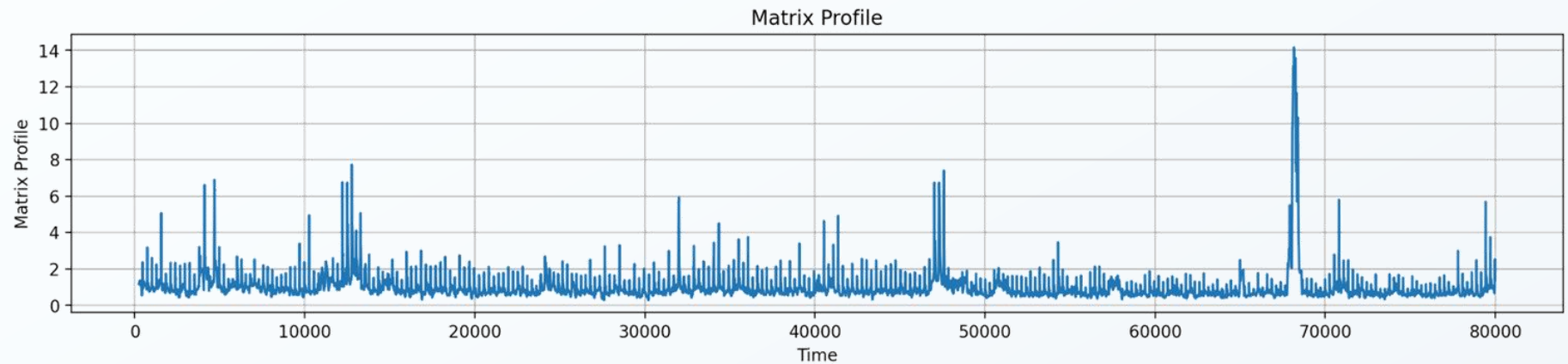
Value over Time





# Matrix Profile - Examples

## Matrix Profile analysis



# Matrix Profile

## Zabbix requirements

- ▶ Performance and efficiency
- ▶ Easy to use
- ▶ Easy to understand and interpret results
- ▶ Subsequence-based anomalies
- ▶ Multiple seasonalities

# Multiple seasonalities

## Contenders

### ► Prophet

- Released by Facebook in 2017
- Unlimited seasonalities
- Easy to use
- Designed for business data
- Primary goal is forecasting

### Forecasting at Scale

Sean J. Taylor\*†  
Facebook, Menlo Park, California, United States  
[slt@fb.com](mailto:slt@fb.com)

and

Benjamin Letham†  
Facebook, Menlo Park, California, United States  
[bletham@fb.com](mailto:bletham@fb.com)

#### Abstract

Forecasting is a common data science task that helps organizations with capacity planning, goal setting, and anomaly detection. Despite its importance, there are serious challenges associated with producing reliable and high quality forecasts – especially when there are a variety of time series and analysts with expertise in time series modeling are relatively rare. To address these challenges, we describe a practical approach to forecasting “at scale” that combines configurable models with analyst-in-the-loop performance analysis. We propose a modular regression model with interpretable parameters that can be intuitively adjusted by analysts with domain knowledge about the time series. We describe performance analyses to compare and evaluate forecasting procedures, and automatically flag forecasts for manual review and adjustment. Tools that help analysts to use their expertise most effectively enable reliable, practical forecasting of business time series.

*Keywords:* Time Series, Statistical Practice, Nonlinear Regression

# Multiple seasonalities

## Contenders

### ► MSTL

- New, published in 2021
- Extension of STL
- Unlimited seasonalities, detects seasons automatically
- Tuning parameters
- Faster than Prophet

#### MSTL: A Seasonal-Trend Decomposition Algorithm for Time Series with Multiple Seasonal Patterns

Kasun Bandara<sup>a,\*</sup>, Rob J Hyndman<sup>b</sup>, Christoph Bergmeir<sup>c</sup>

<sup>a</sup>*School of Computing and Information Systems, Melbourne Centre for Data Science, University of Melbourne*

<sup>b</sup>*Department of Econometrics and Business Statistics, Monash University*

<sup>c</sup>*Department of Data Science and AI, Monash University*

#### Abstract

The decomposition of time series into components is an important task that helps to understand time series and can enable better forecasting. Nowadays, with high sampling rates leading to high-frequency data (such as daily, hourly, or minutely data), many real-world datasets contain time series data that can exhibit multiple seasonal patterns. Although several methods have been proposed to decompose time series better under these circumstances, they are often computationally inefficient or inaccurate. In this study, we propose Multiple Seasonal-Trend decomposition using Loess (MSTL), an extension to the traditional Seasonal-Trend decomposition using Loess (STL) procedure, allowing the decomposition of time series with multiple seasonal patterns. In our evaluation on synthetic and a perturbed real-world time series dataset, compared to other decomposition benchmarks, MSTL demonstrates competitive results with lower computational cost. The implementation of MSTL is available in the R package *forecast*.

**Keywords:** Time Series Decomposition, Multiple Seasonality, MSTL, TBATS, STR

\*Corresponding Author Name: Kasun Bandara, Affiliation: School of Computing and Information Systems, Melbourne Centre for Data Science, University of Melbourne, Melbourne, Australia, Postal Address: School of Computing and Information Systems, The University of Melbourne, Victoria 3052, Australia, E-mail address: Kasun.Bandara@unimelb.edu.au



# Coming to Zabbix

## Trigger functions

- ▶ `trendmp(/host/key,eval,detect,subseq)`
  - Number of discords
- ▶ `trendmseason(/host/key,eval,detect)`
  - Anomaly rate

# Further plans and research

- ▶ Include anomaly detection in standard templates
- ▶ MERLIN, VALMOD
- ▶ Multivariate anomaly detection
- ▶ Untie from trends
- ▶ Mark anomalies on graphs

# References

- ▶ Schmidl, S., et al. *"Anomaly Detection in Time Series: A Comprehensive Evaluation."*, 2022
- ▶ Rewicki, F., et al. *"Is it worth it? Comparing six deep and classical methods for unsupervised anomaly detection in time series"*, 2024
- ▶ Braei, M., et al. *"Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art"*, 2020
- ▶ Wu, R., et al. *"Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress"*, 2020
- ▶ M. Saquib Sarfraz, et al. *"Position: Quo Vadis, Unsupervised Time Series Anomaly Detection?"*, 2024
- ▶ Keogh, E., et al. *"Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets"*, 2016
- ▶ Taylor, S.J., Letham, B. *"Forecasting at Scale"*, 2017
- ▶ Bandara, K. *"MSTL: A Seasonal-Trend Decomposition Algorithm for Time Series with Multiple Seasonal Patterns"*, 2021
- ▶ Hoang, D., et al. *"The UCR Time Series Classification Archive"*, [https://www.cs.ucr.edu/~eamonn/time\\_series\\_data\\_2018/](https://www.cs.ucr.edu/~eamonn/time_series_data_2018/)
- ▶ Numenta, *"Numenta Anomaly Benchmark"*, <https://github.com/numenta/NAB>

Thank you!

