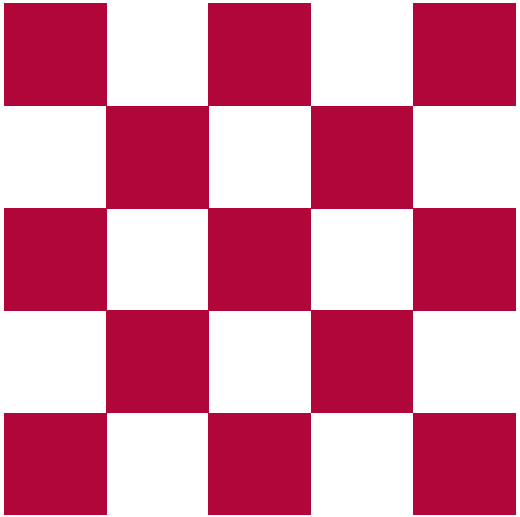
Covid Regulations

Please, register to this event with the Corona Warn-App!

Large-Scale Time Series Analytics
F-E.06, Campus II, HPI

Please, always wear your mask and sit in a checkboard fashion.



When you sit, you can take the mask off.

Course: Large-Scale Time Series Analytics

Deep Dive into Time Series Anomaly Detection

Sebastian Schmidl
Phillip Wenig

Agenda

1. Organization
   - Introduction Round
   - Team Building
   - Fix Dates
2. Definitions
   - Time Series
   - Anomaly
3. Algorithms
   - Unsupervised
   - Semi-Supervised
   - Supervised
   - Preliminary Evaluation Results
4. Tools
   - GutenTAG
   - TimeEval
   - UltraMine
5. Research Example: S2G++
6. Homework

Organization

Introduction round

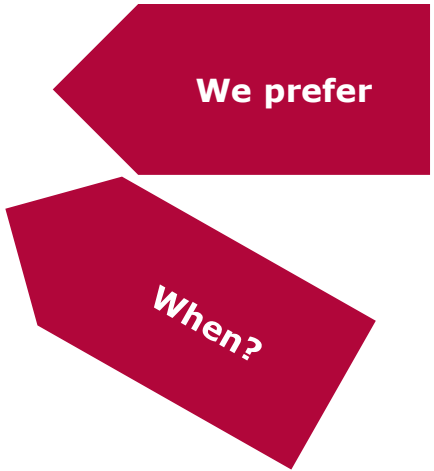
- Name
- Interests
  - Technologies
  - Research fields
- Hobbies?
- …

Team Building

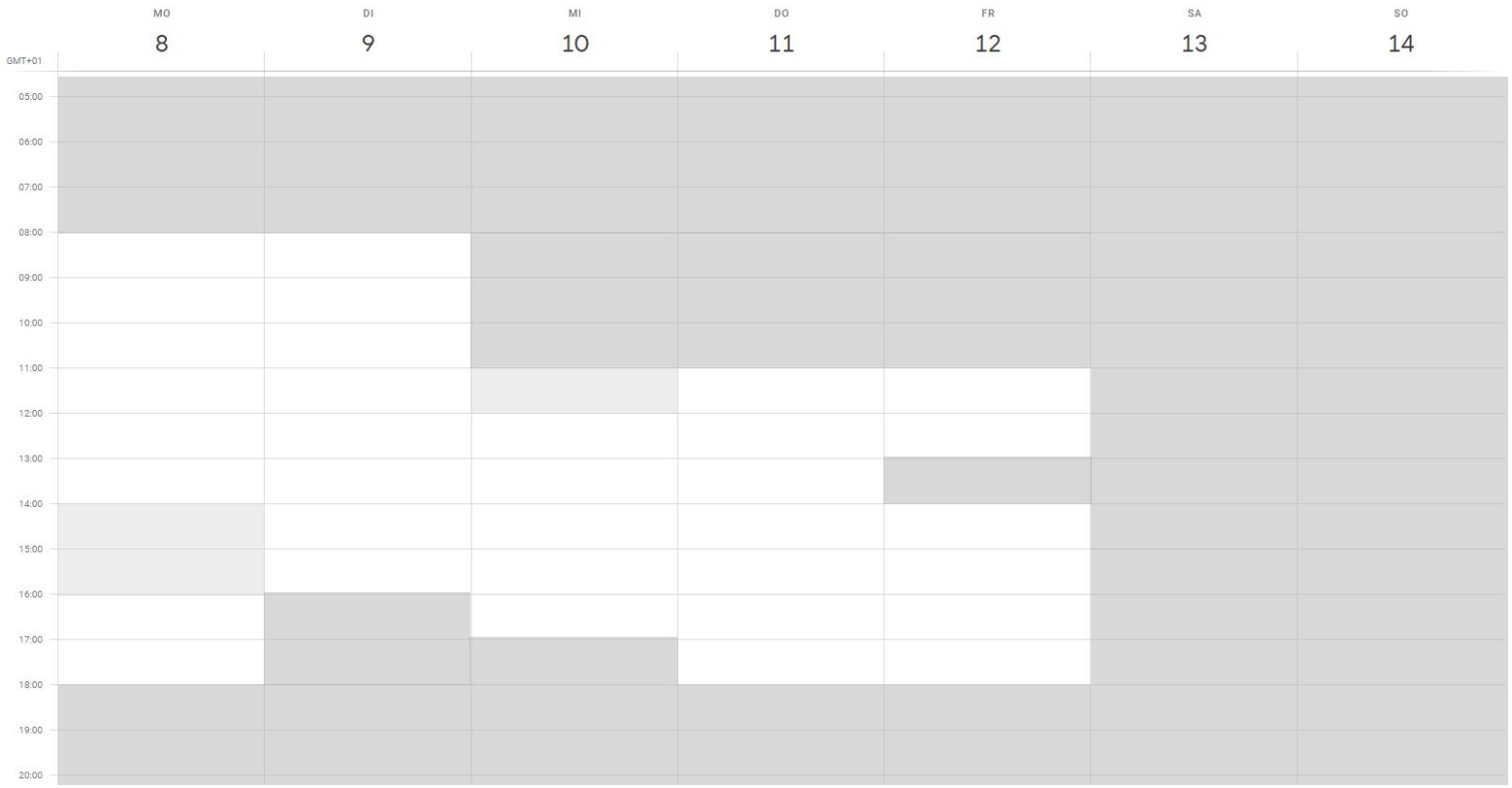
If you haven‘t found a team partner, do so now.

Fix Dates

1. Bi-weekly
   - Starting this week
   - All together
   - Every team presents current progress
   - Meeting room F-2.10 (most likely)
2. On-demand
   - Every team sends us date proposals
   - Minimum once every 2 weeks
   - More if needed



Fix Dates



Schmidl & Wenig
Large-Scale TSA
Winter 2021/22
Chart **8**

Definitions

Time Series

Ordered set

Time series = data series

In this study, we investigate algorithms for the detection of anomalous subsequences in time series data. A *time series* (or *data series* in general) is an ordered set $T = \{T_1, T_2, \ldots, T_m\}$ of m real-valued, potentially multi-dimensional data points $T_i \in \mathbb{R}^n$. A *subsequence* $T_{i,j} = \{T_i, \ldots, T_j\} \subseteq T$ is a contiguous segment of T with length $|T_{i,j}| = j - i$ and $|T_{i,j}| \geq 1$. Our evaluation assumes that the data points are equidistant, which is true for most real-world time series and relieves the algorithms from interpreting the concrete continuous measures (time, mass, angle, etc.); data series not following this assumption need to be discretized.

Real-valued multi-dimensional data points

equidistant

Subsequence = contiguous segment of time series with length >=1

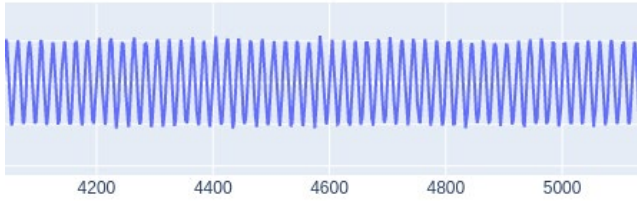
Time Series

or timestamps

Index	value_0
0	0.199
1	0.323
2	0.339
3	0.401
4	0.356
5	0.299
6	0.200
7	0.154
…	…

```
import pandas as pd

time_series = pd.read_csv(PATH_TO_CSV)
time_series.plot()
```



Time Series

Index	value_0	value_1
0	0.199	0.199
1	0.323	0.323
2	0.339	0.339
3	0.401	0.401
4	0.356	0.356
5	0.299	0.299
6	0.200	0.200
7	0.154	0.154
…	…	…

Multivariate

- Same time
- Same subject
- But multiple (different) channels

Time Series

Index	value_0	value_1	is_anomaly
0	0.199	0.199	0
1	0.323	0.323	0
2	0.339	0.339	0
3	0.401	0.401	0
4	0.356	0.356	0
5	0.299	0.299	1
6	0.200	0.200	1
7	0.154	0.154	1
…	…	…	

Labeling

- Single point labels

Or potentially overlapping subsequences

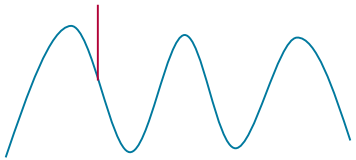
Anomaly

Definitions

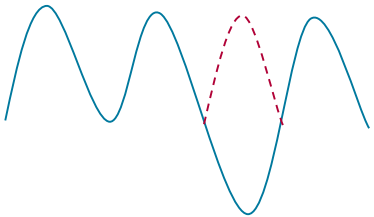
- Largest distance from rest of data points/subsequences
- Highest error in reconstructing learned behavior
- Most isolated part(s) of time series
- Part of the data set that cannot be explained by the underlying function that the data set follows.
- …

Examples

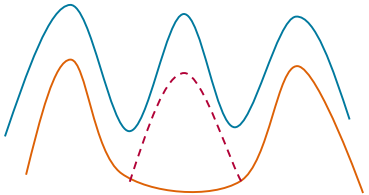
Single Point Extrema



Pattern Change

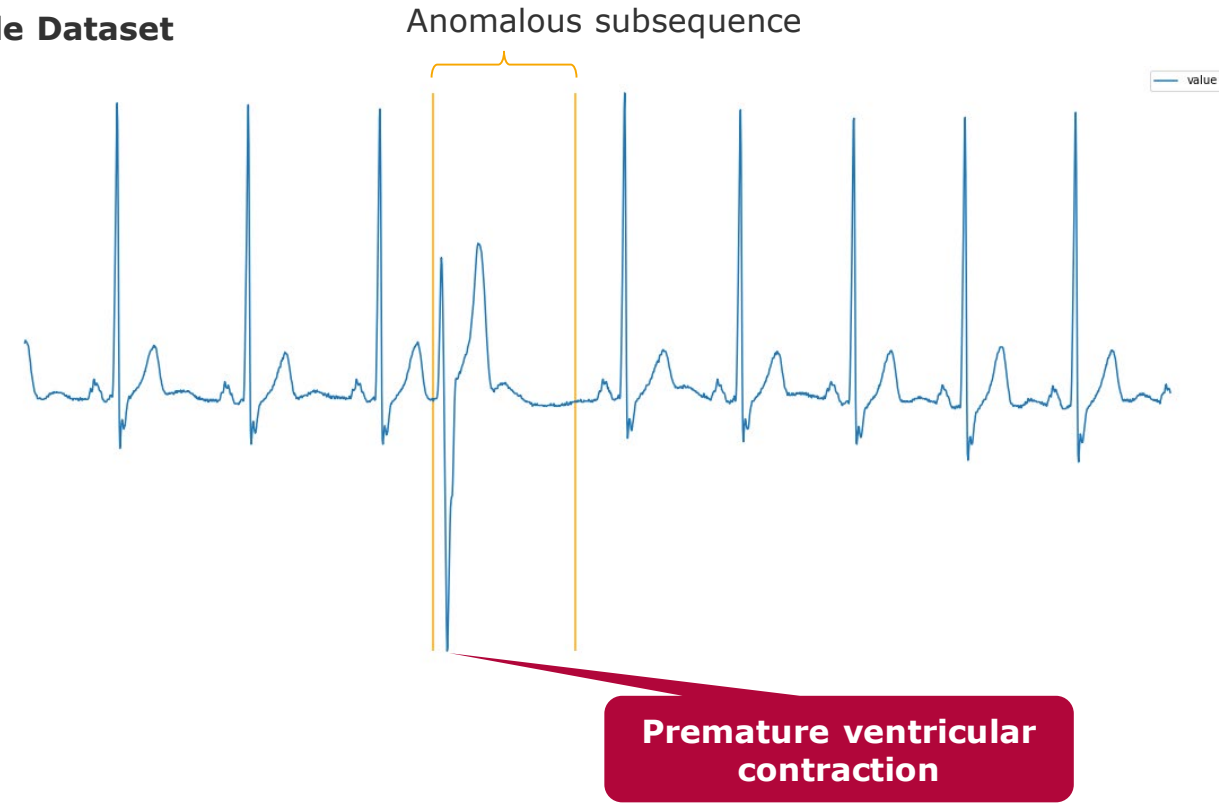


Correlation Anomaly



Anomaly

Example Dataset

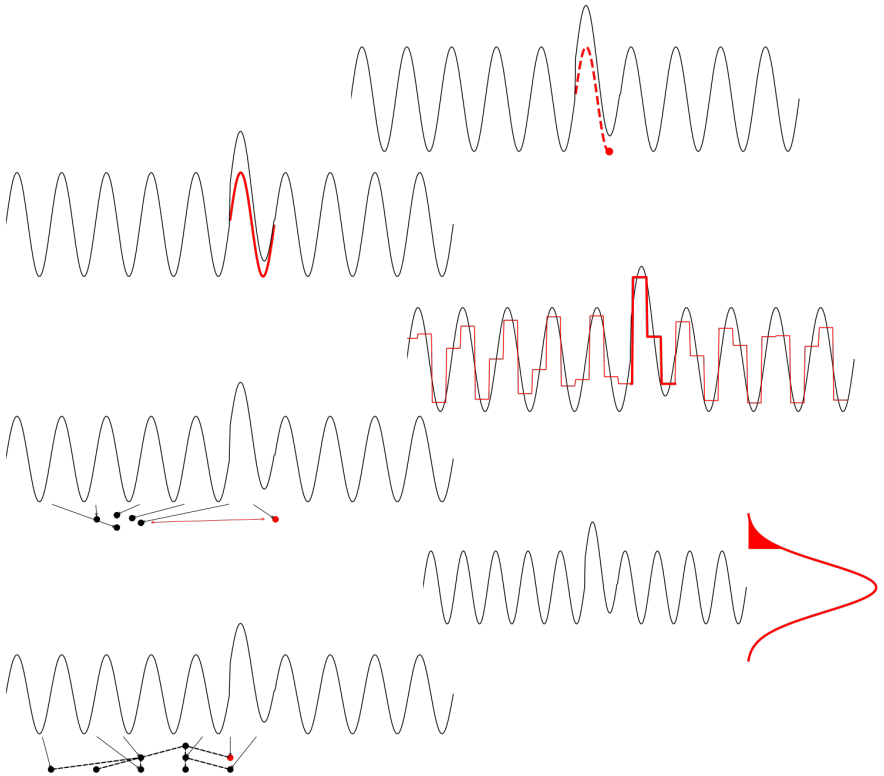


HPI

Algorithms

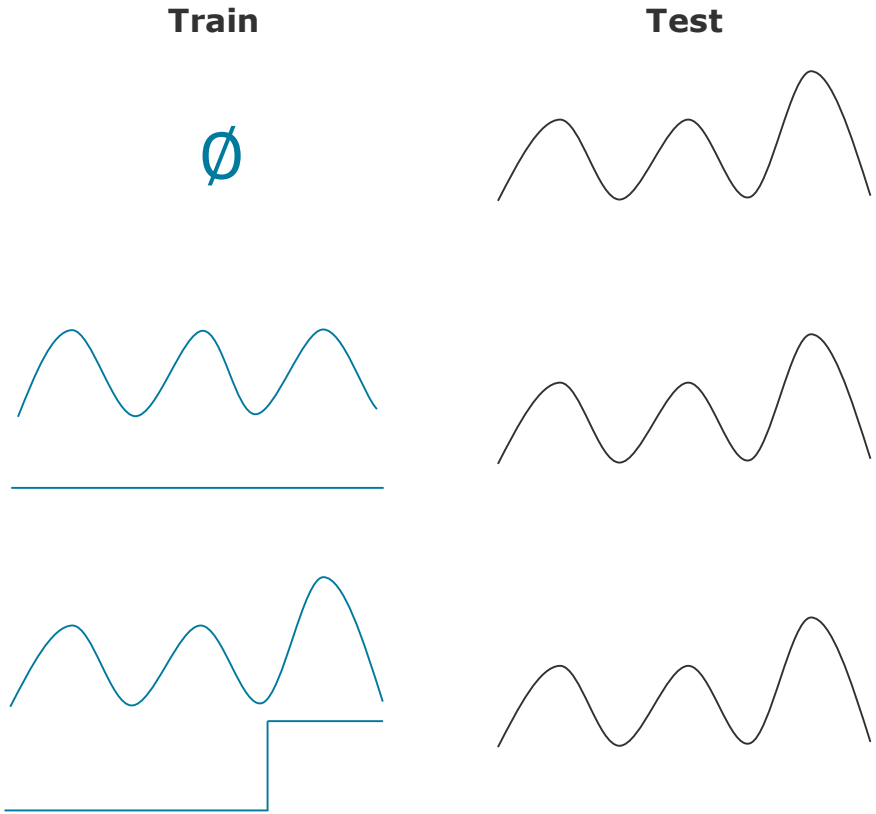
Algorithm Method Families

- Forecasting methods
- Reconstruction methods
- Encoding methods
- Distance methods
- Distribution methods
- Isolation tree methods



Algorithm Learning Types

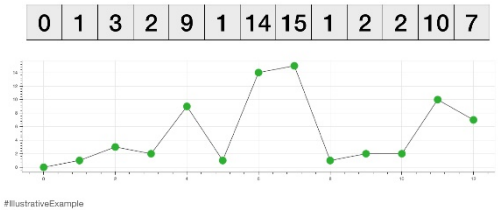
- Unsupervised
- Semi-supervised
- Supervised



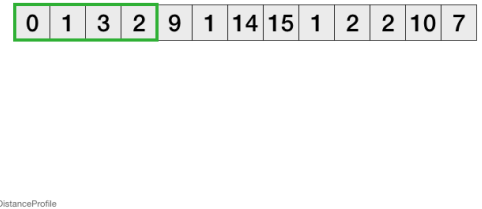
Unsupervised Algorithm Example

Matrix Profile

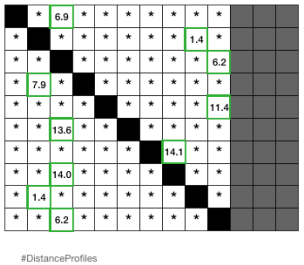
Time Series with Length, n = 13



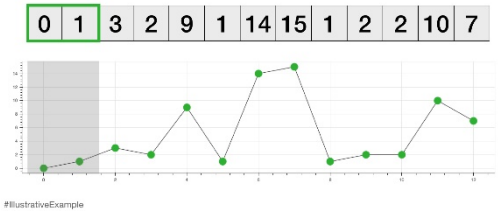
Pairwise Euclidean Distance



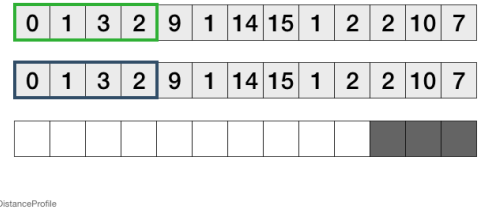
Matrix Profile



Subsequence



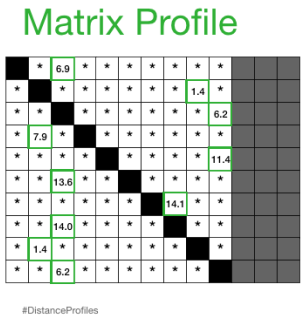
Pairwise Euclidean Distance

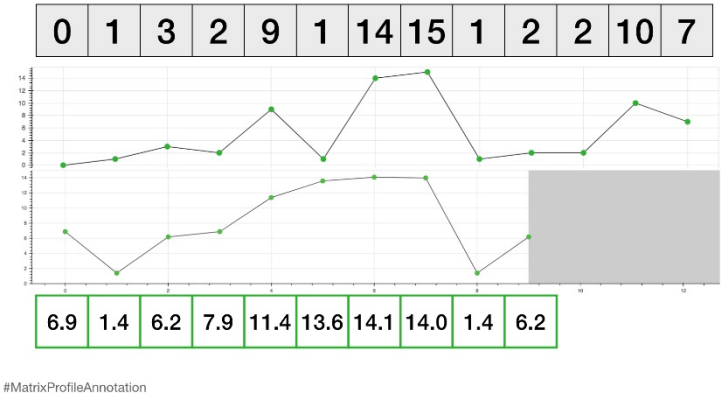


doi:10.1109/ICDM.2016.0085, source: https://stumpy.readthedocs.io/en/latest/Tutorial_The_Matrix_Profile.html

Unsupervised Algorithm Example

Matrix Profile



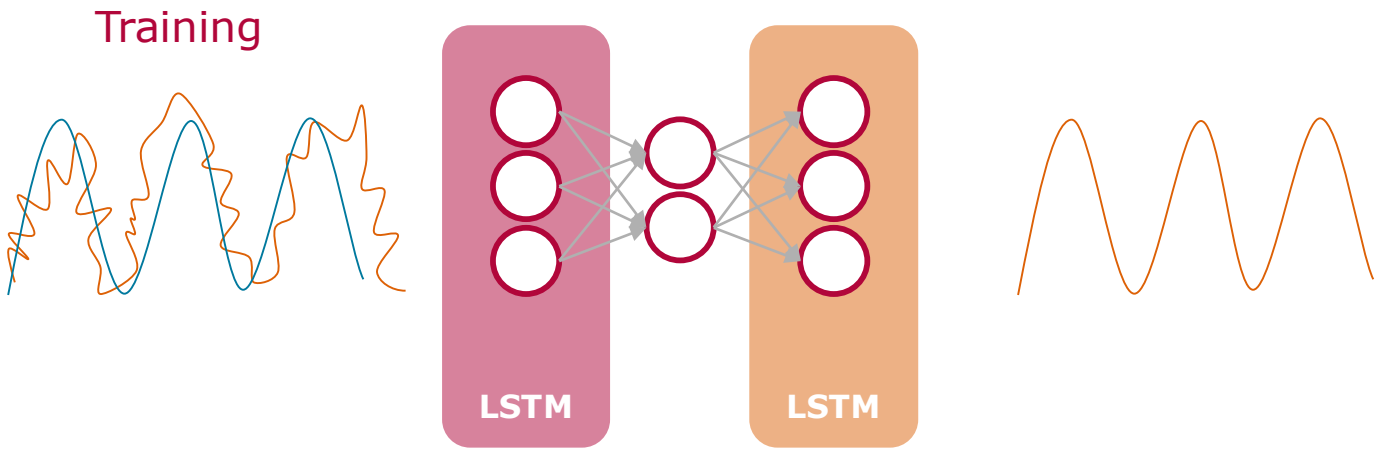


Schmidl & Wenig
Large-Scale TSA
Winter 2021/22
Chart **20**

doi:10.1109/ICDM.2016.0085, source: https://stumpy.readthedocs.io/en/latest/Tutorial_The_Matrix_Profile.html

Semi-Supervised Algorithm Example

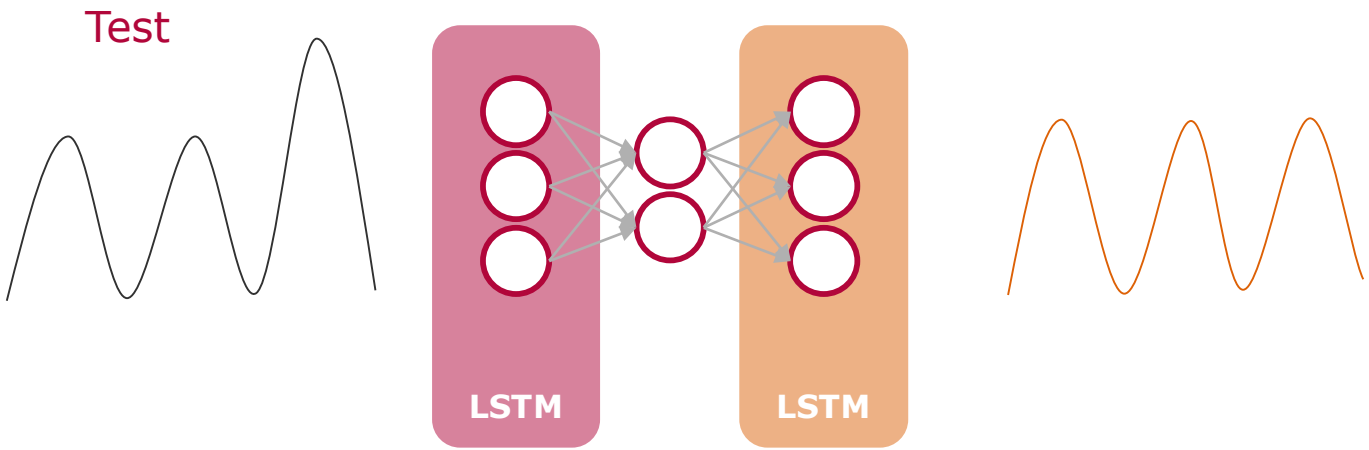
- Encoder Decoder Anomaly Detector (EncDec-AD)



arXiv:1607.00148

Semi-Supervised Algorithm Example

- Encoder Decoder Anomaly Detector (EncDec-AD)

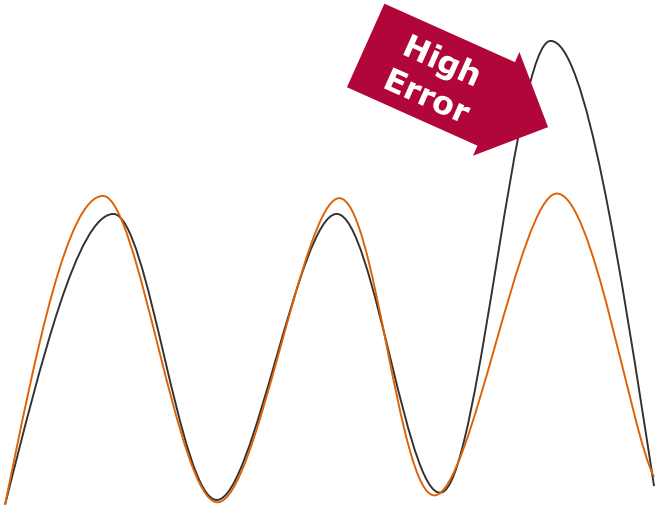


arXiv:1607.00148

Semi-Supervised Algorithm Example

- Encoder Decoder Anomaly Detector (EncDec-AD)

Test

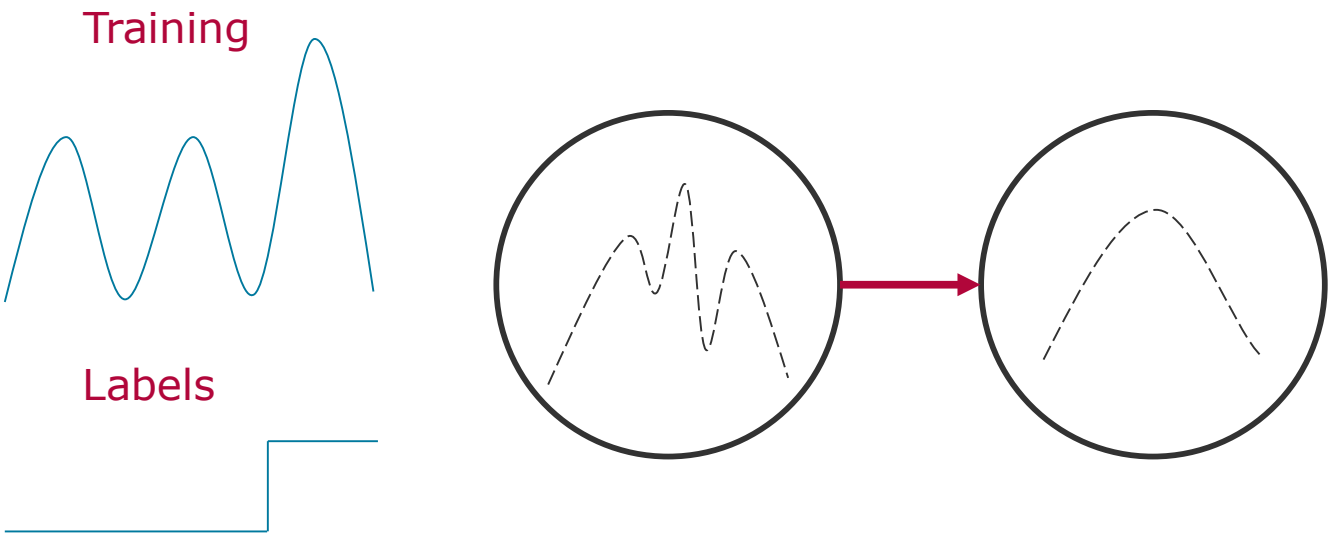


Anomaly Score

arXiv:1607.00148

Supervised Algorithm Example

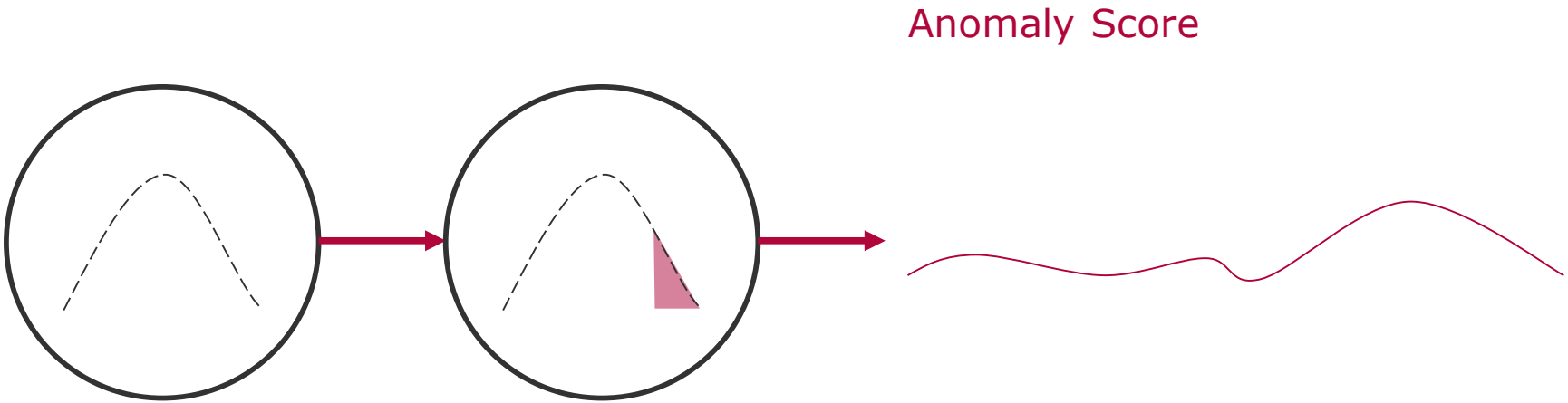
- Normalizing Flows



arXiv:1912.09323
Further reading (mathematics!!!): http://akosiorek.github.io/ml/2018/04/03/norm_flows.html

Supervised Algorithm Example

- Normalizing Flows



arXiv:1912.09323
Further reading (mathematics!!!): http://akosiorek.github.io/ml/2018/04/03/norm_flows.html

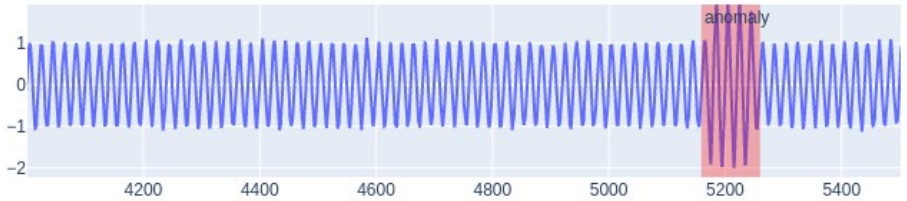
Preliminary Evaluation Results

HPI

Tools

GutenTAG

A Good Timeseries Anomaly Generator




```yaml
timeseries:
  - name: demo
    length: 10000
    channels: 1
    semi-supervised: true
    supervised: true
    base-oscillation:
      kind: sinus
      frequency: 10.0
      amplitude: 1.0
      variance: 0.05
    anomalies:
      - position: end
        length: 100
        kinds:
          - kind: amplitude
            parameters:
              amplitude_factor: 2.0
    # - position: beginning
```

Output format

Time Series Properties

Anomalies Properties

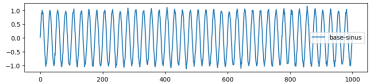
https://gitlab.hpi.de/akita/guten-tag

GutenTAG

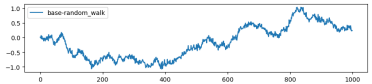
A Good Timeseries Anomaly Generator

Base oscillations

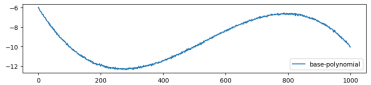
Sine wave



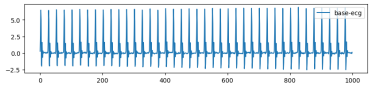
Random walk



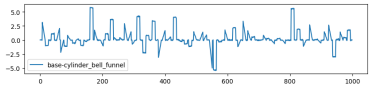
Polynomial



ECG

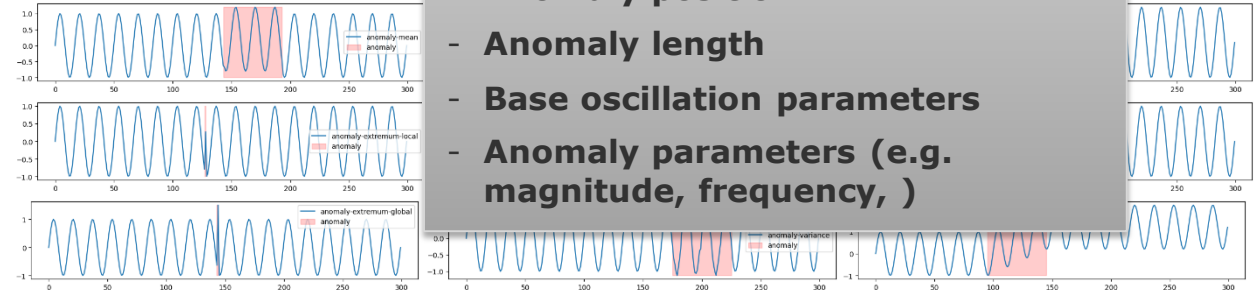


Cylinder-bell-funnel



Variation of

- **Noise**
- **Trend**
- **#dimensions**
- **#anomalies (similar or different)**
- **Anomaly position**
- **Anomaly length**
- **Base oscillation parameters**
- **Anomaly parameters (e.g. magnitude, frequency,)**



Anomaly types

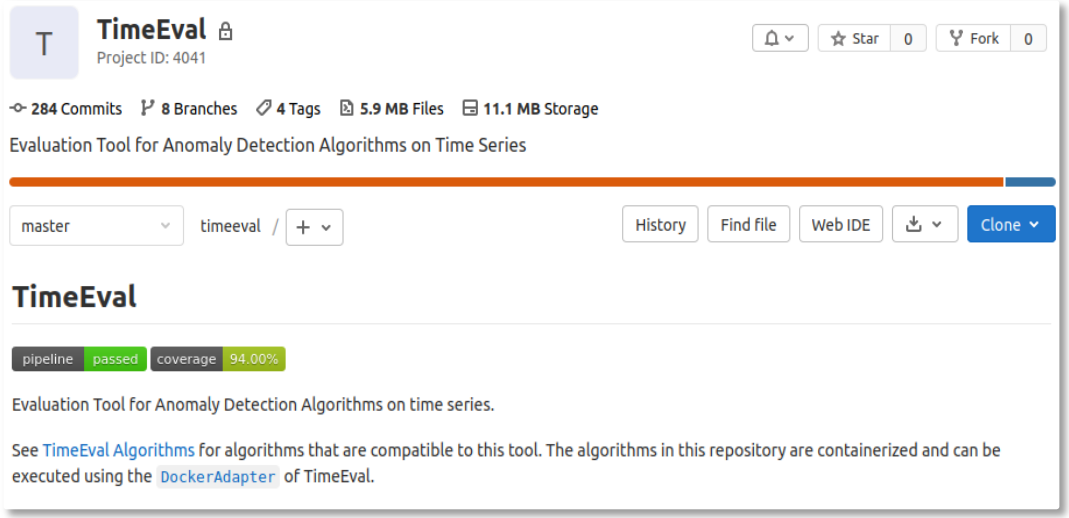
https://gitlab.hpi.de/akita/guten-tag

GutenTAG

A Good Timeseries Anomaly Generator

TimeEval

- Integrated benchmark datasets
- **Adapter** architecture for algorithm integration (FunctionAdapter, DockerAdapter, ...)
- Automatic **quality scoring using different metrics**
- **Automatic timing** of the algorithm execution
- **Distributed** experiment execution
- Output and logfile tracking for subsequent inspection / debugging

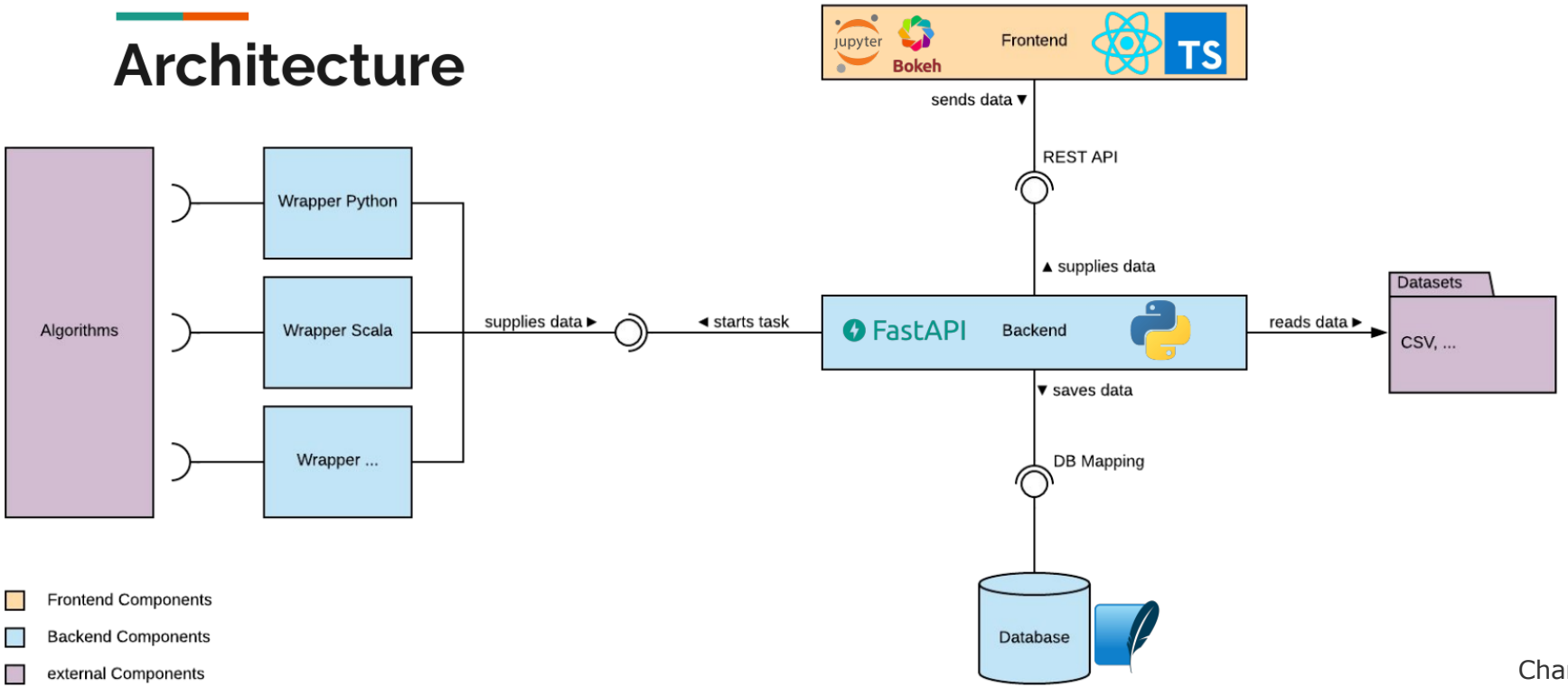


https://gitlab.hpi.de/akita/timeeval

TimeEval

UltraMine

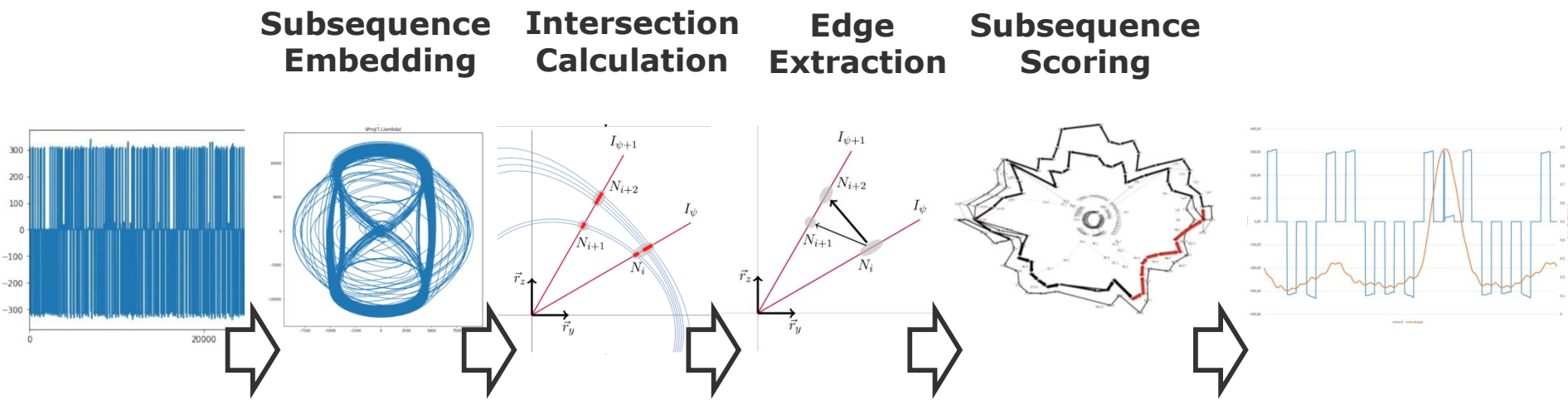
Architecture



UltraMine

Research Example: DADS
(more efficient)

Series2Graph



Schmidl & Wenig
Large-Scale TSA
Winter 2021/22
Chart **36**

doi:10.14778/3407790.3407792

DADS

Distributed Detection of Sequential Anomalies in Univariate Time Series

Johannes Schneider · Phillip Wenig · Thorsten Papenbrock

Received: date / Accepted: date

Abstract The automated detection of sequential anomalies in time series is an essential task for many applications, such as the monitoring of technical systems, fraud detection in high-frequency trading, or the early detection of disease symptoms. All these applications require the detection to find *all* sequential anomalies possibly *fast* on potentially very *large* time series. In other words, the detection needs to be effective, efficient and scalable w.r.t. the input size. Series2Graph is an effective solution based on Graph-embeddings that is robust against re-occurring anomalies, can discover sequential anomalies of arbitrary length and works without training data. Yet, Series2Graph is neither efficient nor scalable due to its single-threaded approach; it can, in particular, not process arbitrarily large sequences due to the memory constraints of a single machine.

In this paper, we propose our Distributed Anomaly Detection System, short DADS, which is an efficient and scalable adaptation of Series2Graph. Based on the actor programming model, DADS distributes the input time sequence, intermediate state and the computation to all processors of a cluster in a way that minimizes communication costs and synchronization barriers. Our evaluation shows that DADS is orders of magnitude faster than S2G, scales almost linearly with the number of processors in the cluster and can process much larger input sequences due to its scale-out property.

J. Schneider
Hasso Plattner Institute, University of Potsdam
E-mail: johannes.schneider@hpi.de

P. Wenig
Hasso Plattner Institute, University of Potsdam
E-mail: phillip.wenig@hpi.de

T. Papenbrock
Hasso Plattner Institute, University of Potsdam
E-mail: thorsten.papenbrock@hpi.de

Keywords distributed programming, sequential anomaly, actor model, data mining, time series

1 Sequential Anomaly Detection

Univariate time series are ordered sequences of one-dimensional, real-valued records [3, 18, 31, 38, 1]. The ordering is usually time-related – hence the name – but sometimes also follows other continuous measures, such as size, distance, or speed. Anomalies in time series denote subsequences with significant, i.e., particularly infrequent values or patterns. In general, we distinguish between point anomalies a.k.a. outliers and sequential anomalies a.k.a. collective anomalies [9]. Because point anomalies are sequential anomalies of size one, sequential anomalies cover point anomalies. For example, a temperature reading of 50°C in an ordinary room is a point anomaly; a sudden temperature increase of 10°C directly followed by a drastic drop of 15°C is a sequential anomaly, if common temperature changes are slow.

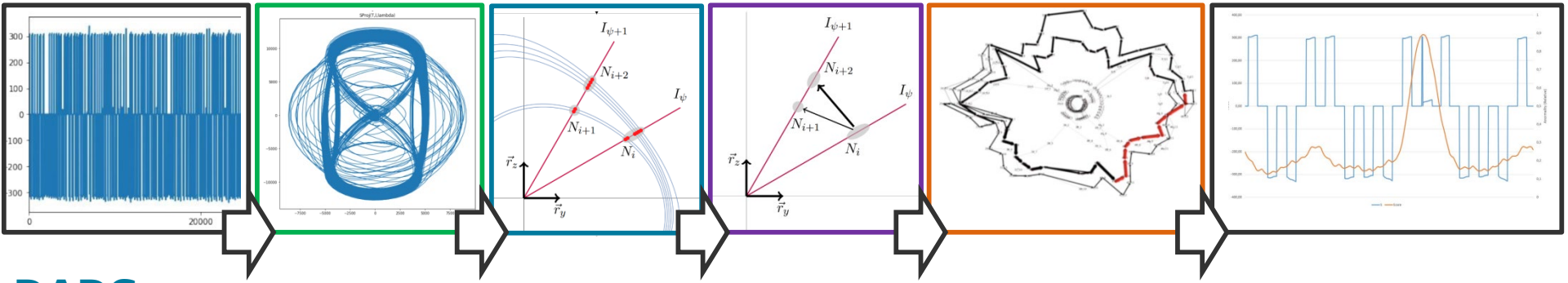
Anomalies often indicate meaningful events in time series and are, therefore, important for various applications. Examples for such applications range from intrusion detection in digital networks [17, 28], over finding unexpected phenomenons in astrophysical measurements [37], to medical studies of disease cases [13]. Figure 1 shows another example from aircraft engineering: We see a time series T of engine measurements with various regular spikes and one anomalously lean spike. None of the records in the anomaly is an outlier, but its shape is clearly different from other patterns that we find in the sequence. An anomaly score, which is depicted in red in Figure 1, is a proper indicator of this anomaly. In this paper, we propose a scalable algorithm that efficiently calculates such anomaly scores.

Sequential anomalies are hard to detect, because they often differ in length, vary in degree of anomaly and might

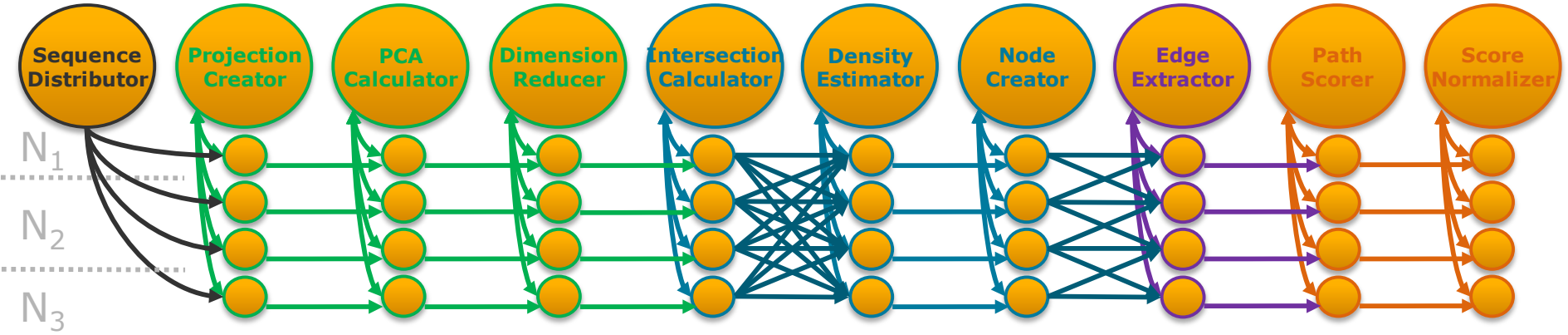
- **D**istributed **A**nomaly **D**etection **S**ystem (DADS)
- Distributed implementation of the Series2Graph (S2G) algorithm
- Distributes computation as well as temporary data
- Reactive / minimizes communication barriers
- Implemented using Akka

doi:10.1007/s00778-021-00657-6

DADS



DADS: Subsequence Embedding | Node Extraction | Edge Extraction | Subsequence Scoring



DADS

|T| (x10^6)	S2G	S2G$^+$	DADS (1P,1T)	DADS (1P,20T)	DADS (12P,20T)
0.01	4	4	6	5	7
0.1	35	29	12	7	7
1	388	297	145	19	9
10	5,897	3,077	904	145	26
50	TL	15,947	4,575	727	106
100	ML	ML	ML	ML	206
200	ML	ML	ML	ML	401
300	ML	ML	ML	ML	586
400	ML	ML	ML	ML	802
500	ML	ML	ML	ML	986

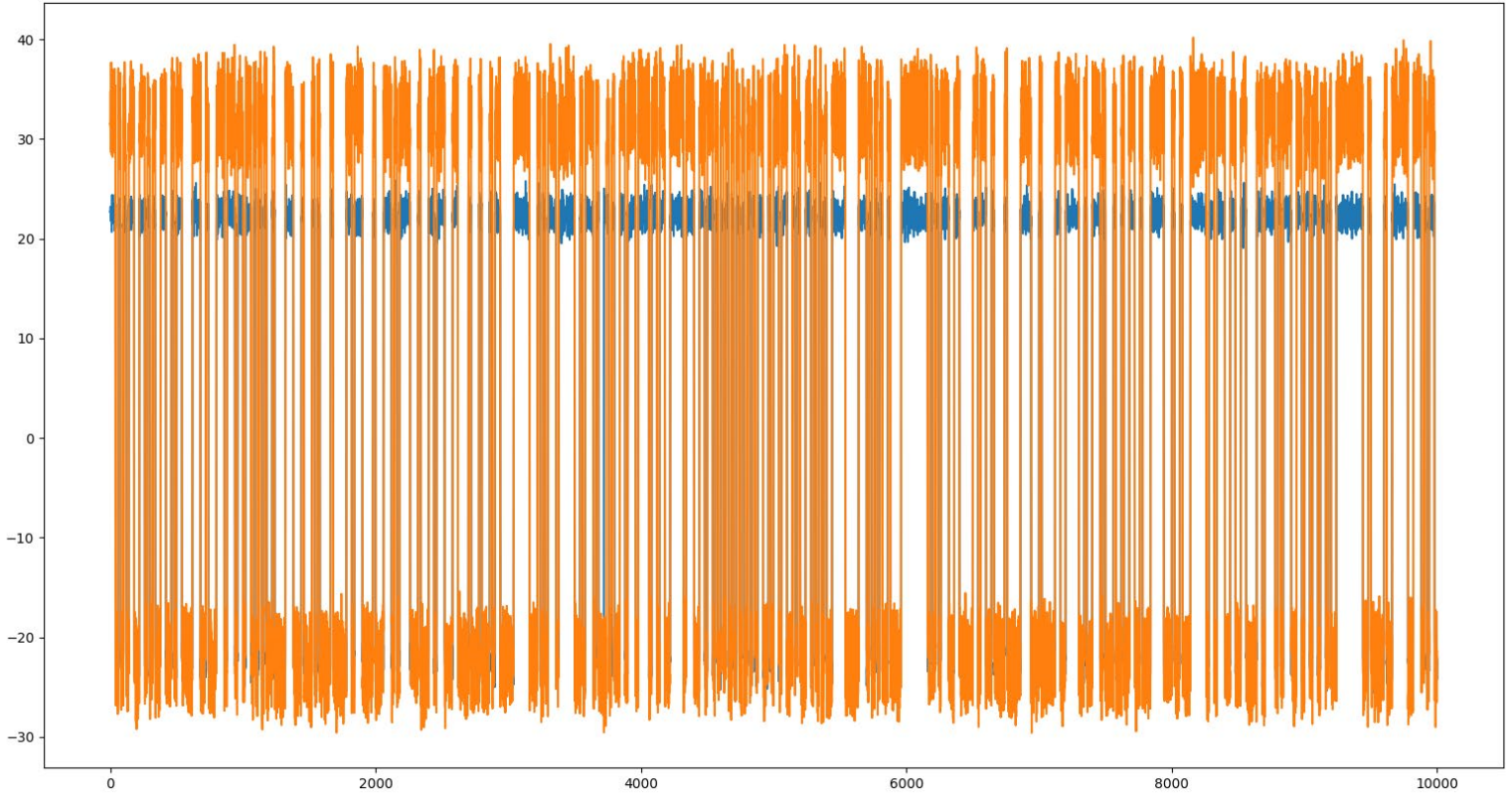
runtime in seconcds

doi:10.1007/s00778-021-00657-6

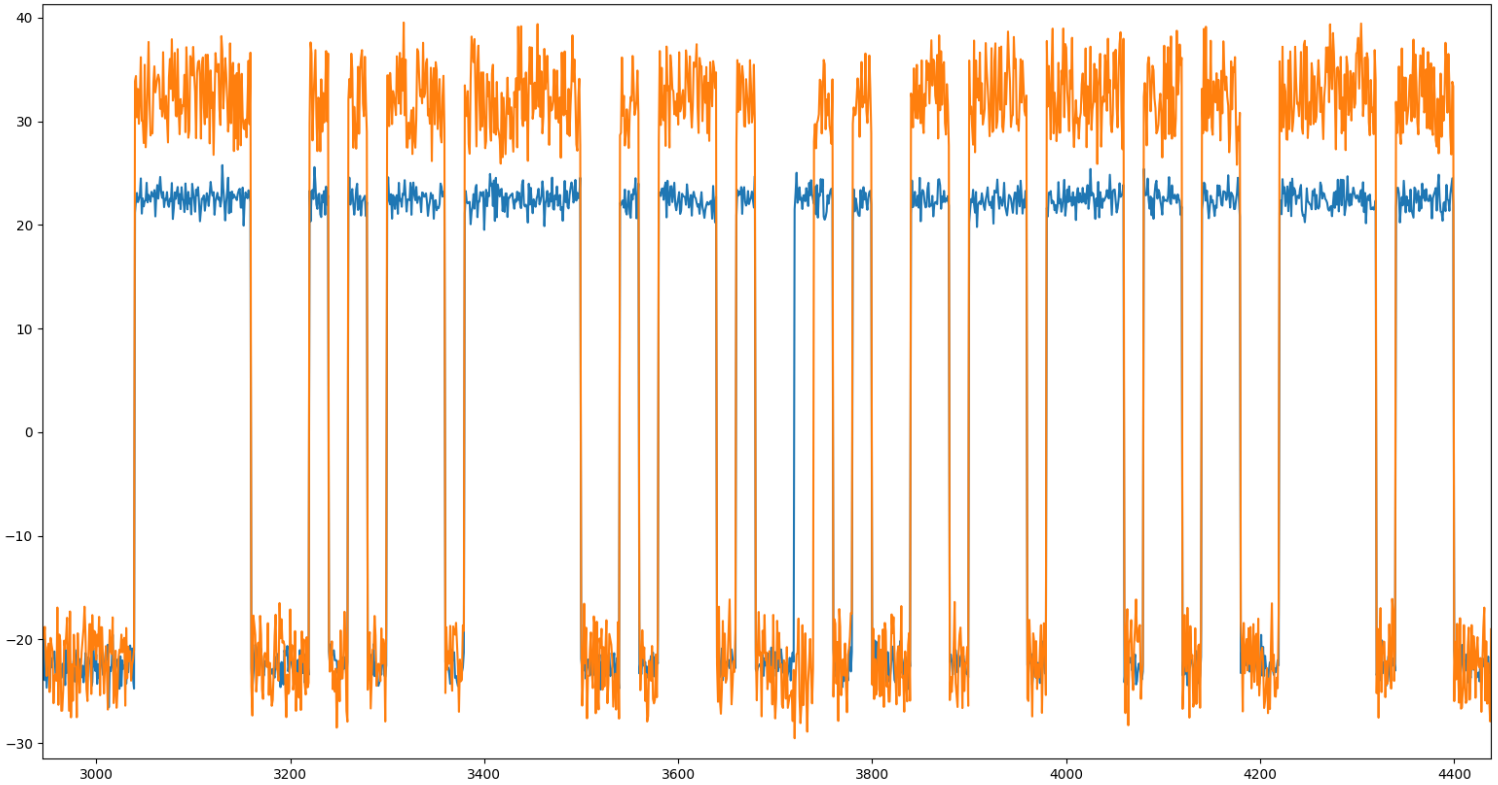
Research Example: S2G++

(more capable; work in progress)

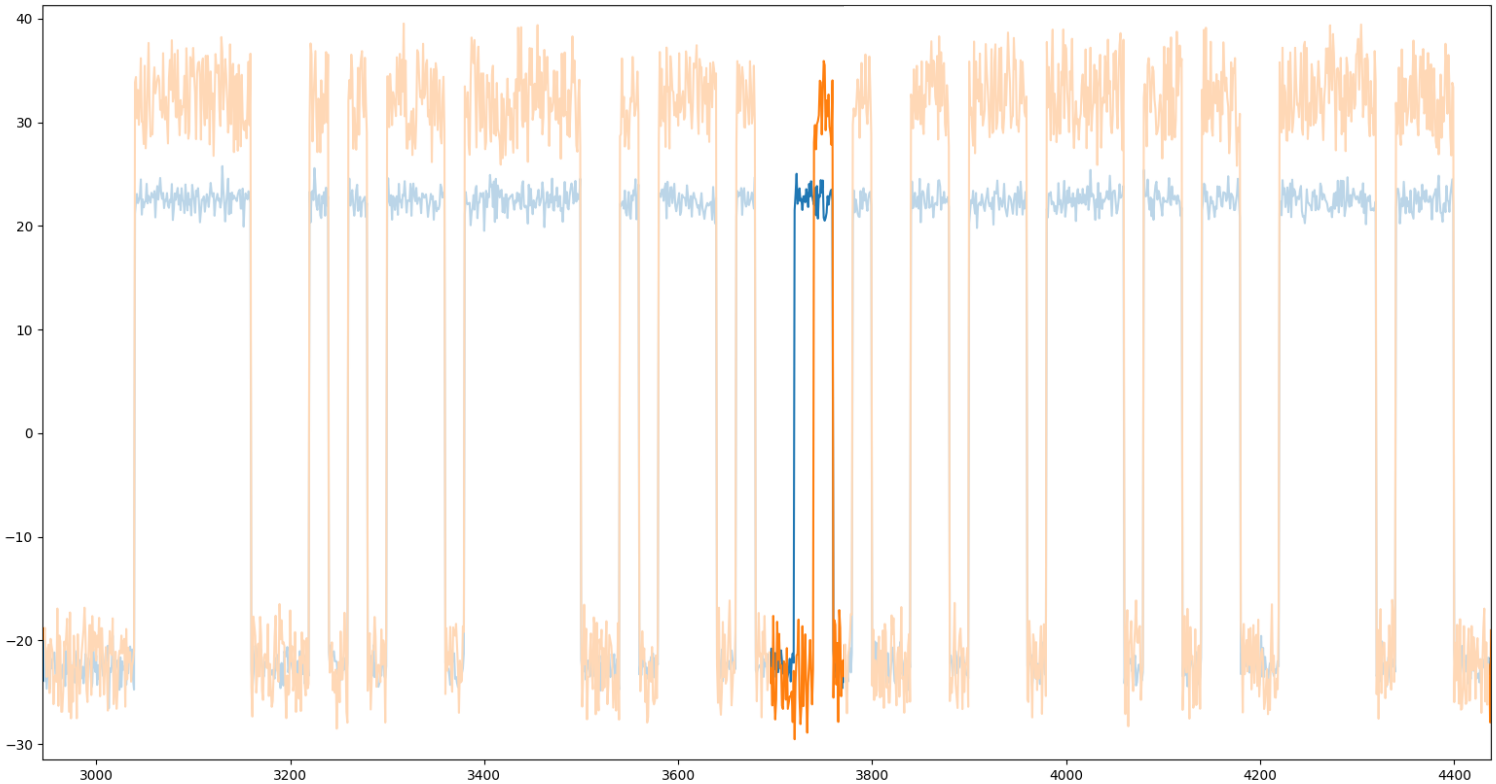
Series2Graph



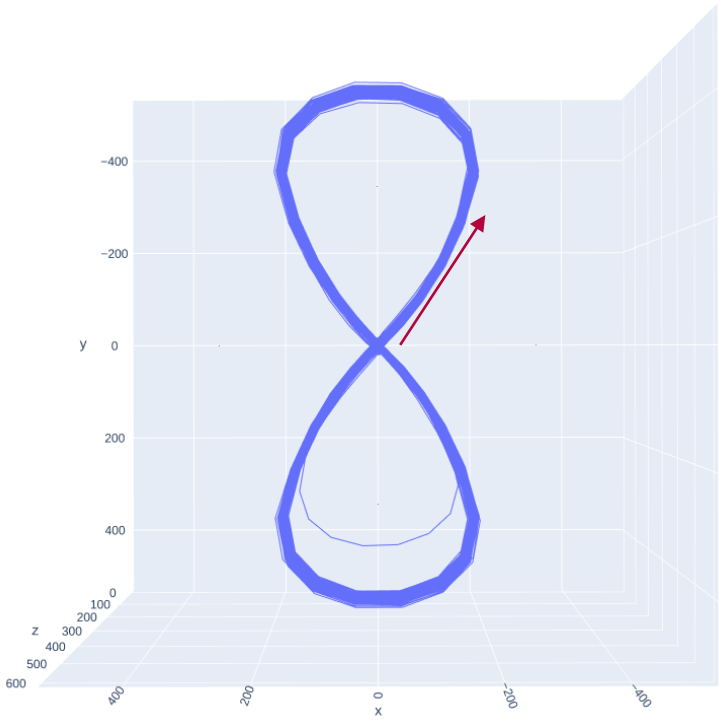
Series2Graph

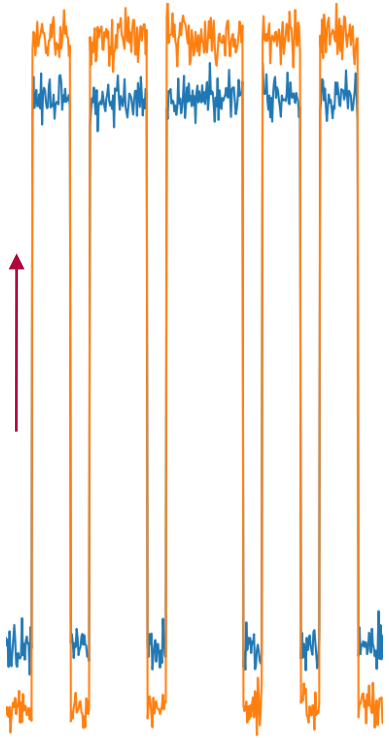


Series2Graph

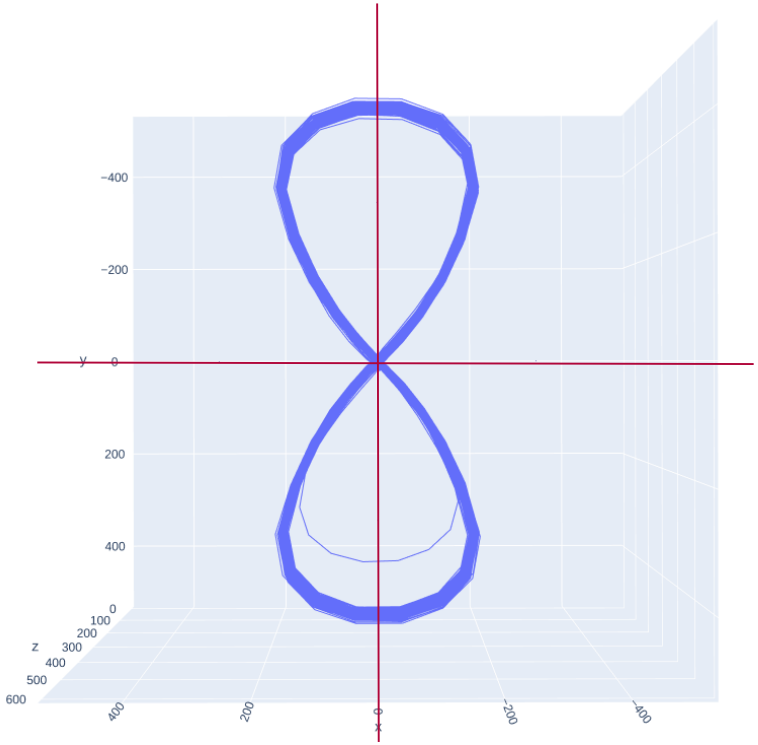


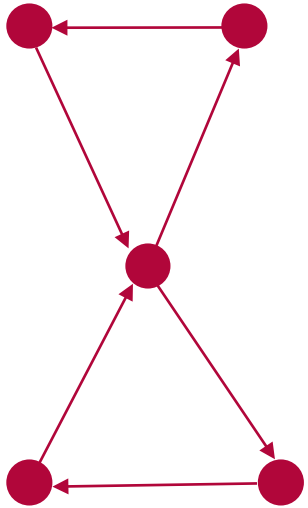
Series2Graph





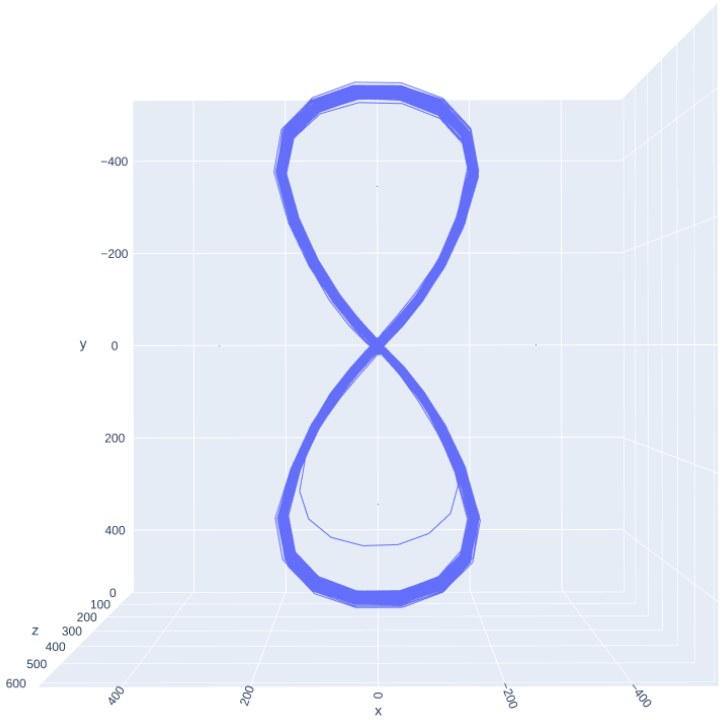
Series2Graph



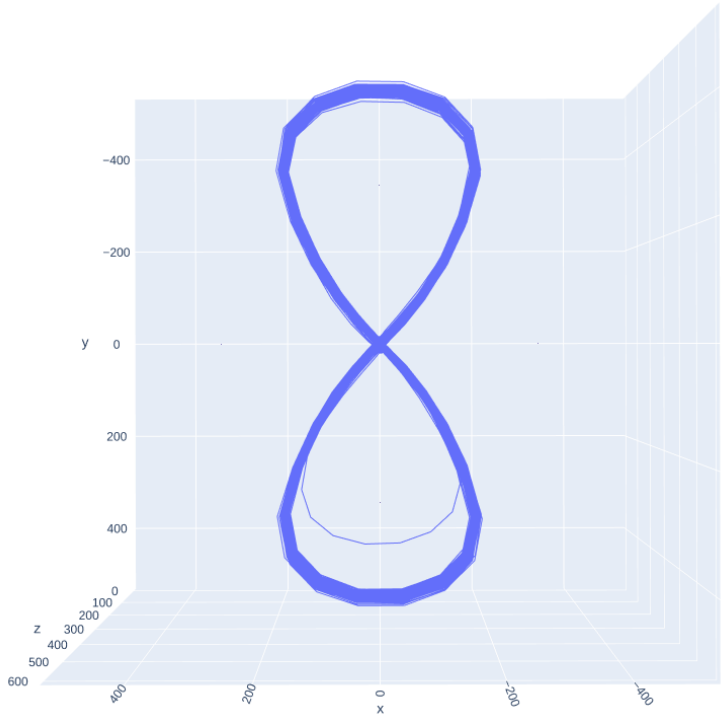


Series2Graph++

Time Series 1

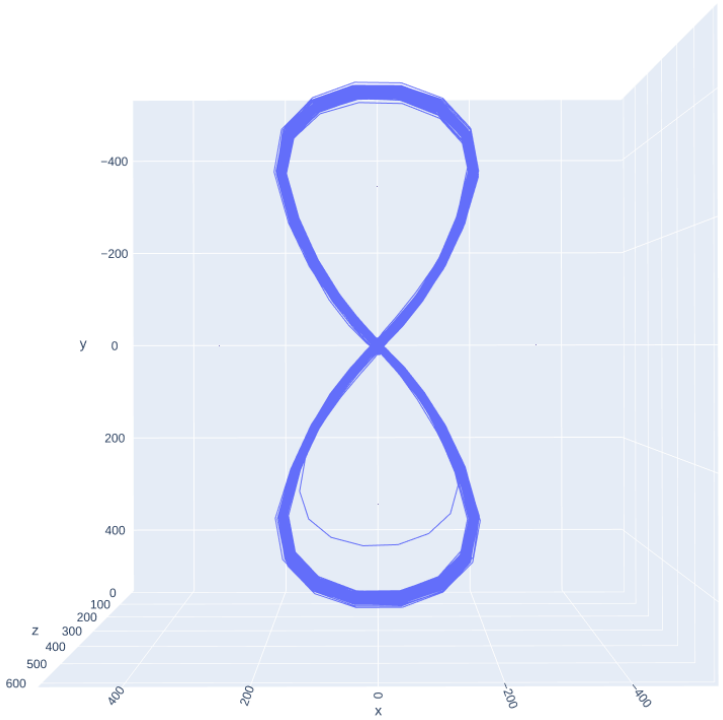


Time Series 2

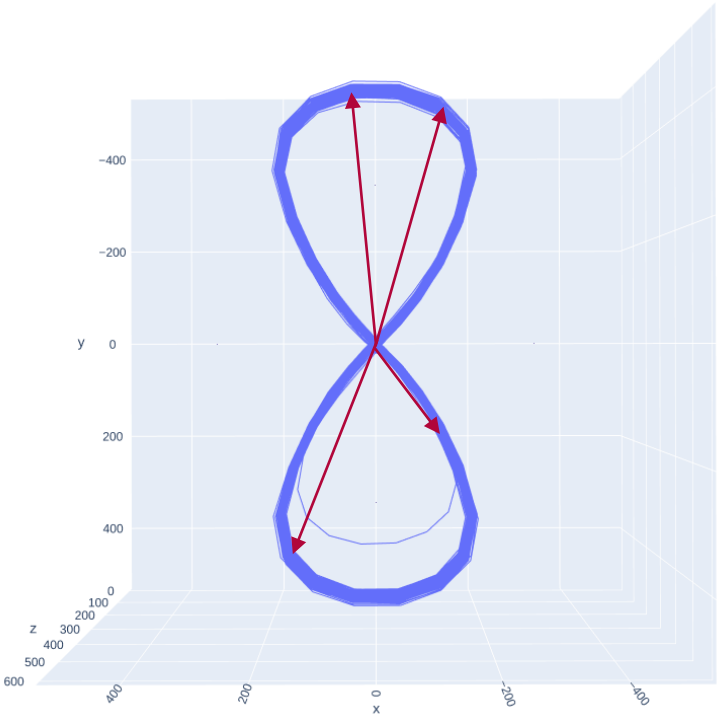


Series2Graph++

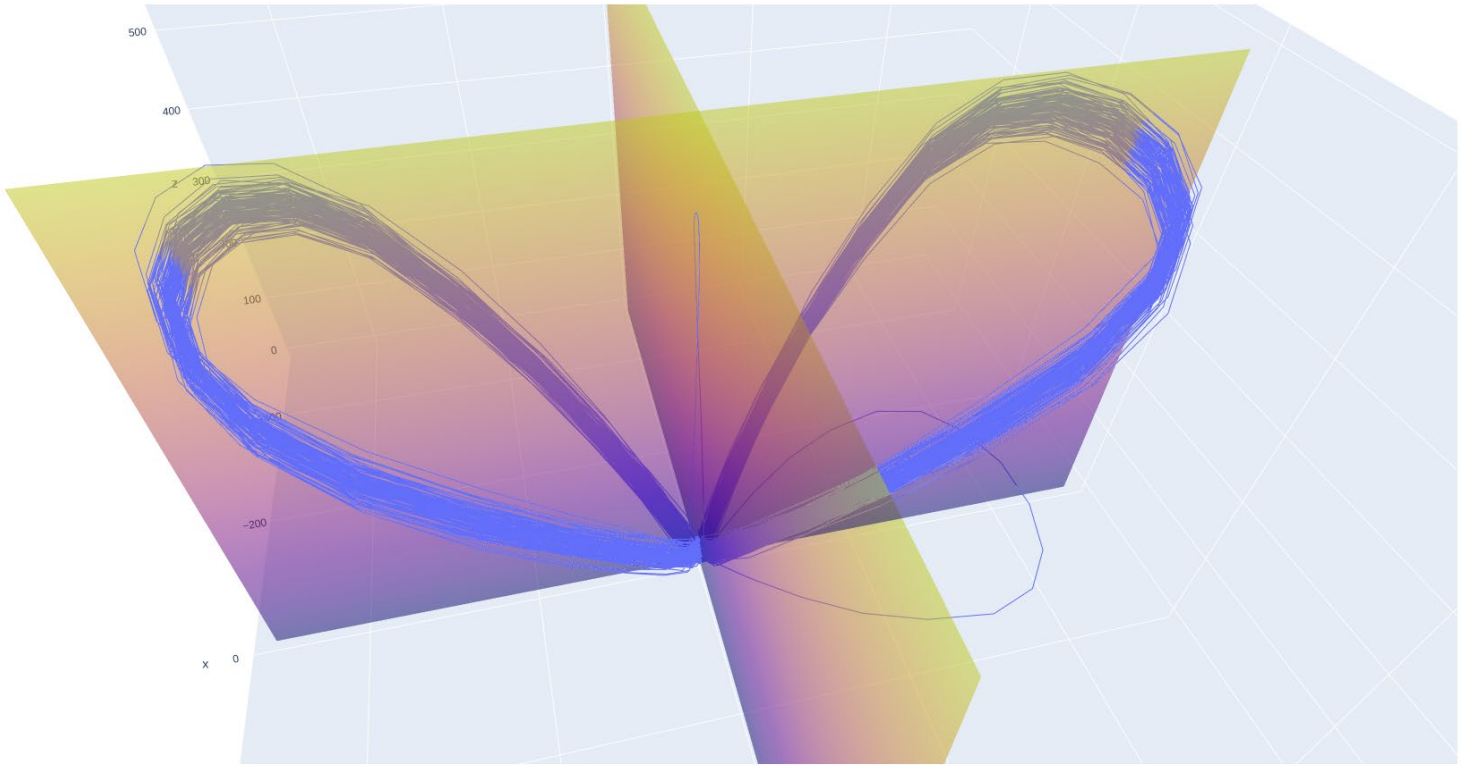
Time Series 1



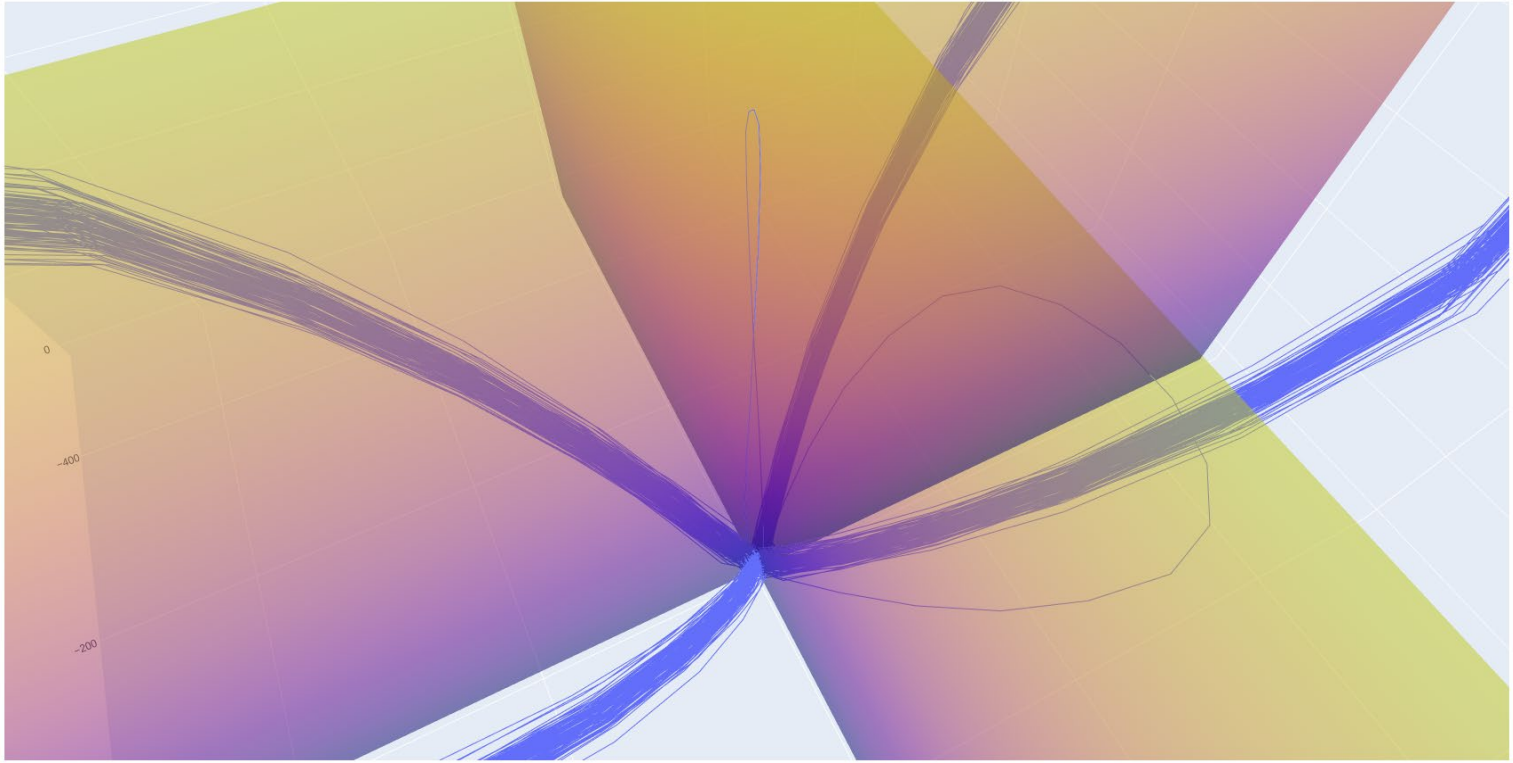
Time Series 2



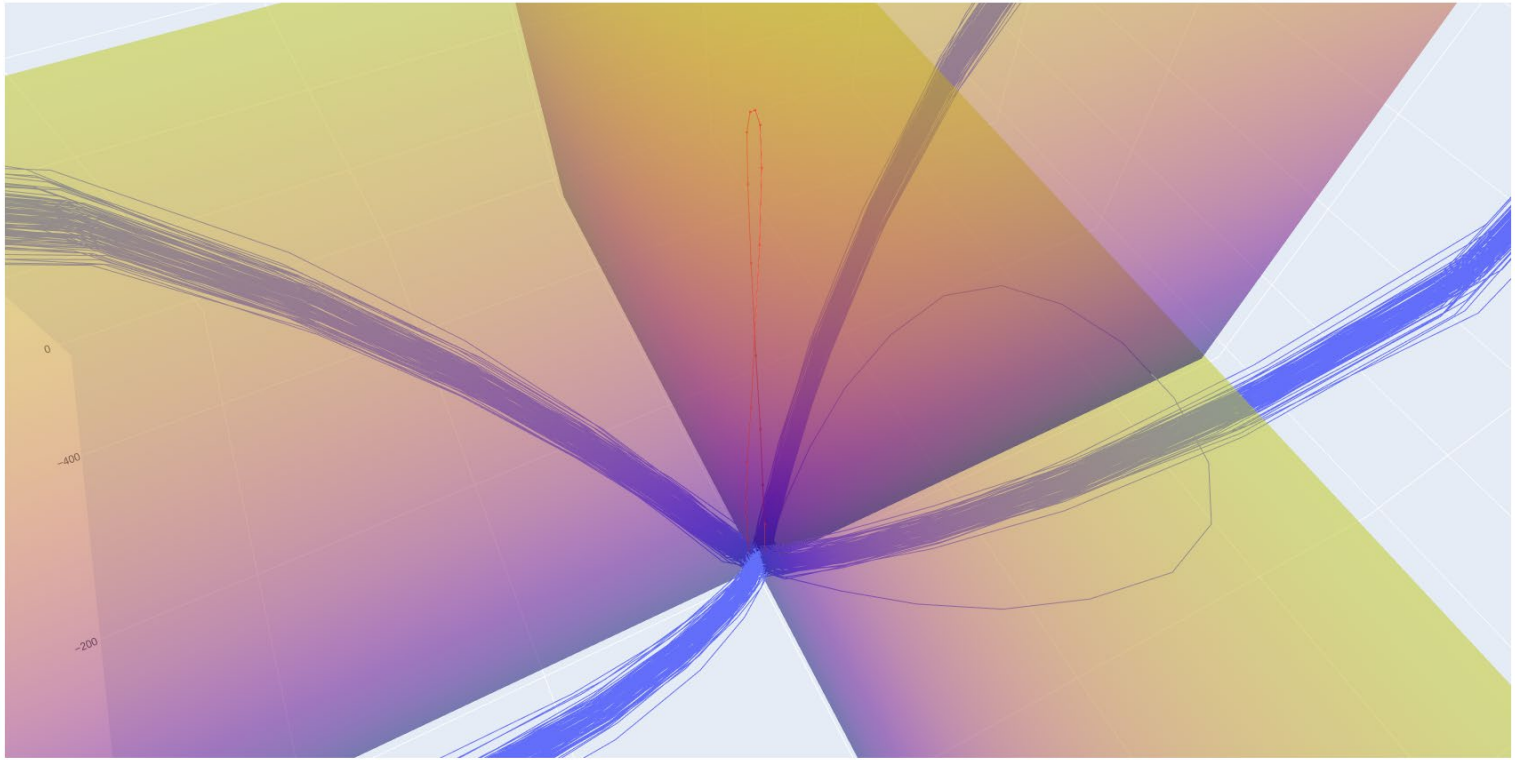
Series2Graph++



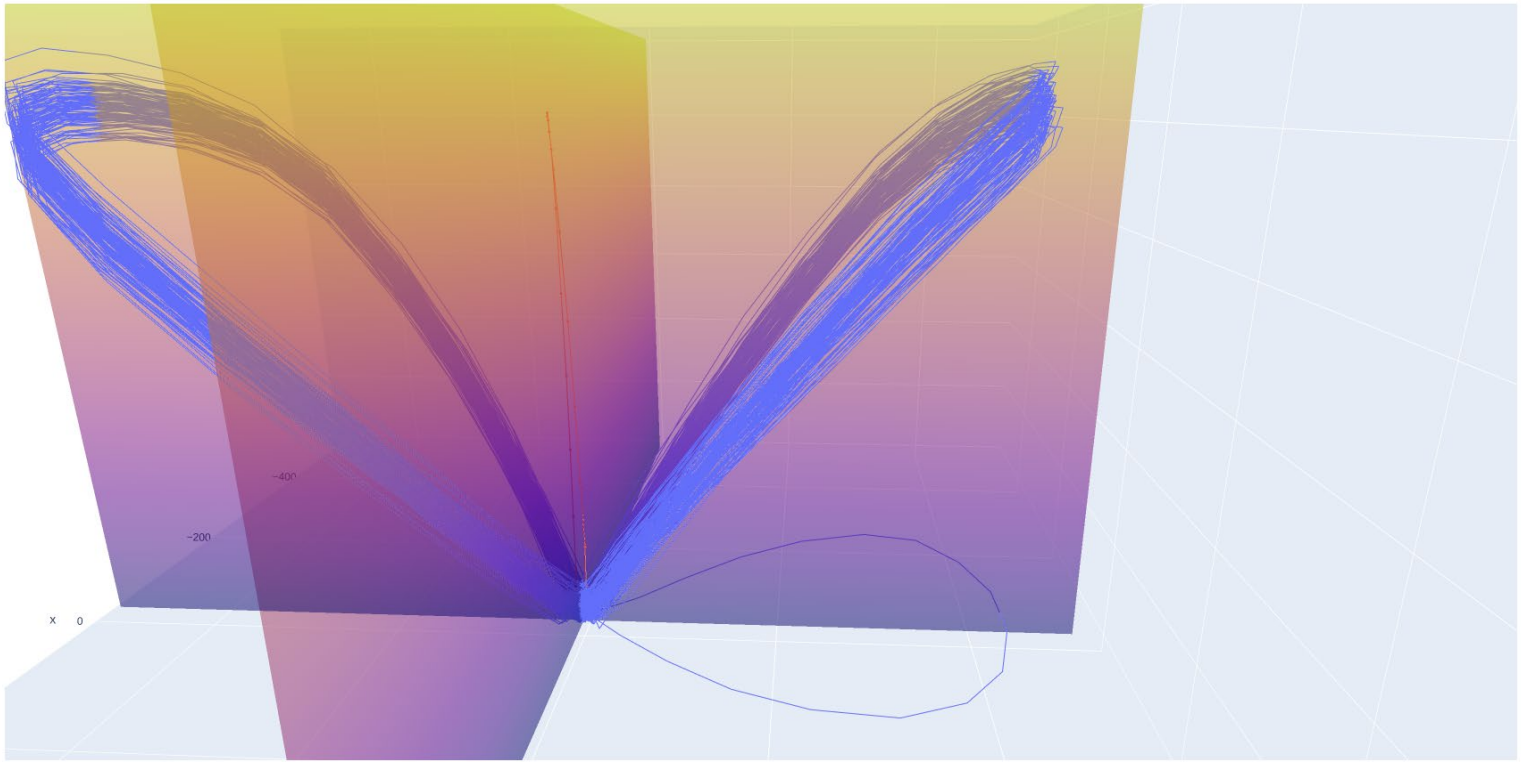
Series2Graph++



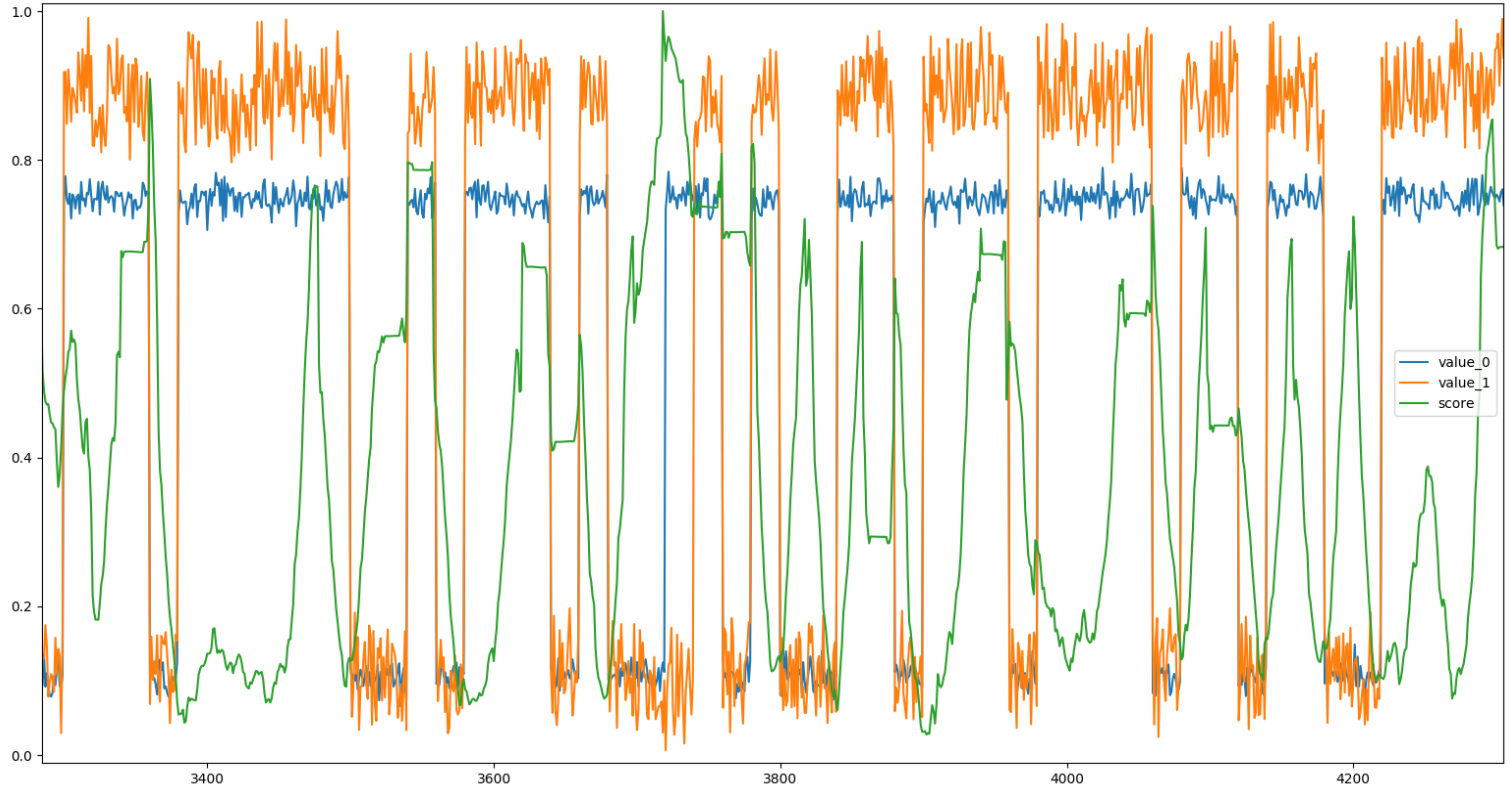
Series2Graph++



Series2Graph++



Series2Graph++



Homework

Homework

- Literature: Draft of our experimental evaluation paper

Anomaly Detection in Time Series: A Comprehensive Evaluation

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

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

Thorsten Papenbrock Philipps University of Marburg Marburg, Germany papenbrock@informatik.uni-marburg.de

ABSTRACT

Detecting anomalous subsequences in time series data is an important task in many areas, ranging from manufacturing processes over finance applications to health care monitoring. An anomaly can indicate important events, such as production faults, delivery bottlenecks, systems 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 scientifically evaluates and compares the different approaches. For this reason, choosing the best detection technique for a given anomaly detection task is a difficult challenge.

With this paper, we present a comprehensive, scientific study that carefully evaluates most state-of-the-art anomaly detection algorithms. We collected and re-implemented a significant amount of 71 anomaly detection algorithms from different domains and evaluated them on 200 time series datasets. The algorithms have been selected from different algorithm families and detection approaches to represent the entire spectrum of anomaly detection techniques. In the paper, we provide a concise overview of the techniques and their commonalities, we evaluate their individual strengths and weaknesses and, thereby, consider factors, such as effectiveness, efficiency, ease of use, and robustness. Our experimental results should ease the algorithm selection problem and open up new research directions.

PVLDB Reference Format:
Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. Anomaly Detection in Time Series: A Comprehensive Evaluation. PVLDB, 16(1): XXX-XXX, 2022.
doi:XX.XX/XXX.XX

PVLDB Availability Tag:
The source code of this research paper has been made publicly available at http://vldb.org/pvldb/format_vol13.html.

*Both authors contributed equally to this work.

Proceedings of the VLDB Endowment, Vol. 16, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

2021-11-03 15:26 Page 1 of 1–13.

Figure 1: An example time series.

1 ANOMALY DETECTION WILDERNESS

A data series is a chronologically ordered sequence of data points. The data points describe some object or process property based on a continuous measure, such as temperature (e. g., physics), mass (e. g., chemistry), angle (e. g., astronomy), position (e. g., geology), or speed (e. g., mechanical engineering). If the order is based on time, the sequence is generally referred to as a *time series*. Regardless of the ordering measure, the recording of the data points usually follows discrete, equally-spaced intervals. For this reason and because most data series analytics algorithms are agnostic of the reference measure, we use the terms *data series*, *time series*, and *sequence* interchangeably in this study.

The data points of a time series record one or multiple real-valued variables. Each variable models one channel of the time series. If the data points consist of only one variable, the time series is called univariate; otherwise, it is *multivariate*. Figure 1 shows an example with two time series: a univariate time series on XXXXXXX and a multivariate time series on XXXXXXX. An *anomaly* in such time series is a point (e. g., outlier) or point sequence (e. g., irregularity) that deviates w. r. t. some measure, model, or embedding from the regular patterns of the sequence. In multivariate time series, these pattern deviations can occur in any single channel and also in the correlation of channels (e. g., skew). Anomalous subsequences might have varying lengths and they might re-appear in the same time series. Depending on the domain of a time series, its anomalies can describe important events, such as heart failures in cardiology [4], structural defects in jet turbine engineering [76], or ecosystem disturbances in earth sciences [10].

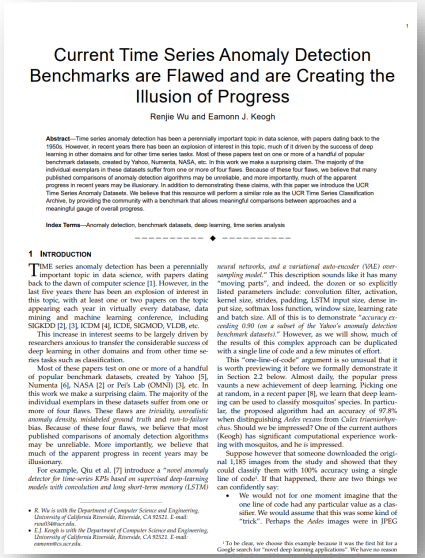
Don’t share or publish!

Find in the PDF in our seminar OwnCloud share.

- Introduction to subsequence anomaly detection in time series
- Overview about current State-of-the-Art methods
- Categorization of methods
- Experiment setup and assumptions

Homework

- Literature: Ongoing discussion on how to assess time series anomaly detection methods, e.g.



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 illusionary. In addition to demonstrating these claims, with this paper we introduce the UCR Time Series Anomaly Datasets. 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.

This increase in interest seems to be largely driven by researchers anxious to transfer the considerable success of deep learning in other domains and from other time series tasks such as classification.

Most of these papers test on one or more of a handful of popular benchmark datasets, created by Yahoo [5], Numenta [6], NASA [2] or Pei's Lab (OMNI) [3], 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. These flaws are *triviality, unrealistic anomaly density, mislabeled ground truth* and *run-to-failure* bias. Because of these four flaws, we believe that most published comparisons of anomaly detection algorithms may be unreliable. More importantly, we believe that much of the apparent progress in recent years may be illusionary.

For example, Qiu et al. [7] introduce a *"novel anomaly detector for time-series KPIs based on supervised deep-learning models with convolution and long short-term memory (LSTM) neural networks, and a variational auto-encoder (VAE) oversampling model."* This description sounds like it has many "moving parts", and indeed, the dozen or so explicitly listed parameters include: convolution filter, activation, kernel size, strides, padding, LSTM input size, dense input size, softmax loss-function, window size, learning rate and batch size. All of this is to demonstrate *"accuracy exceeding 0.90 (on a subset of the Yahoo's anomaly detection benchmark datasets)."* However, as we will show, much of the results of this complex approach can be duplicated with a single line of code and a few minutes of effort.

This "one-line-of-code" argument is so unusual that it is worth previewing it before we formally demonstrate it in Section 2.2 below. Almost daily, the popular press vaunts a new achievement of deep learning. Picking one at random, in a recent paper [8], we learn that deep learning can be used to classify mosquitos' species. In particular, the proposed algorithm had an accuracy of 97.8% when distinguishing *Aedes* terans from *Culex triaeniorhynchus*. Should we be impressed? One of the current authors (Keogh) has significant computational experience working with mosquitos, and he is impressed.

Suppose however that someone downloaded the original L3S5 images from the study and showed that they could classify them with 100% accuracy using a single line of code. If that happened, there are two things we can confidently say:

- We would not for one moment imagine that the one line of code had any particular value as a classifier. We would assume that this was some kind of "trick". Perhaps the *Aedes* images were in JPEG

• R. Wu is with the Department of Computer Science and Engineering, University of California Riverside, Riverside, CA 92521. E-mail: rwu5346@ucr.edu.
• E.J. Keogh is with the Department of Computer Science and Engineering, University of California Riverside, Riverside, CA 92521. E-mail: eamonn@cs.ucr.edu.

[1] To be clear, we choose this example because it was the first hit for a Google search for "novel deep learning applications". We have no reason

Source:
https://doi.org/10.1109/TKDE.2021.3112126

- Current datasets
  - are too **trivial**
  - have unrealisticly **dense** anomalies
  - have **mislabeled** data
  - have more anomalies in the **last** points

Homework

- Check out the material in our seminar share (E-Mail with link follows)
  - Datasets
  - Algorithms
  - Literature (Papers)
  - Metadata and evaluation results
  - …
- Decide on your team direction (goal):
  - More reliable
  - More accurate
  - More efficient
  - More capable

Further Survey/Review papers

Chandola, Varun, Arindam Banerjee, and Vipin Kumar. **Anomaly Detection: A Survey**. ACM Computing Surveys 41, no. 3 (2009): 1–58. https://doi.org/10.1145/1541880.1541882.

Gupta, Manish, Jing Gao, Charu C. Aggarwal, and Jiawei Han. **Outlier Detection for Temporal Data: A Survey.** IEEE Transactions on Knowledge and Data Engineering (TKDE) 26, no. 9 (2014): 2250–67. https://doi.org/10.1109/TKDE.2013.184.

Choudhary, Dhruv, Arun Kejariwal, and Francois Orsini. **On the Runtime-Efficacy Trade-off of Anomaly Detection Techniques for Real-Time Streaming Data.** ArXiv:1710.04735, October 12, 2017. http://arxiv.org/abs/1710.04735.

Chalapathy, Raghavendra, and Sanjay Chawla. **Deep Learning for Anomaly Detection: A Survey.** ArXiv:1901.03407, 2019. http://arxiv.org/abs/1901.03407.

Blázquez-García, Ane, Angel Conde, Usue Mori, and Jose A. Lozano. **A Review on Outlier/Anomaly Detection in Time Series Data.** ArXiv:2002.04236, February 11, 2020. http://arxiv.org/abs/2002.04236.

Braei, Mohammad, and Sebastian Wagner. **Anomaly Detection in Univariate Time-Series: A Survey on the State-of-the-Art.** ArXiv:2004.00433, April 1, 2020. http://arxiv.org/abs/2004.00433.