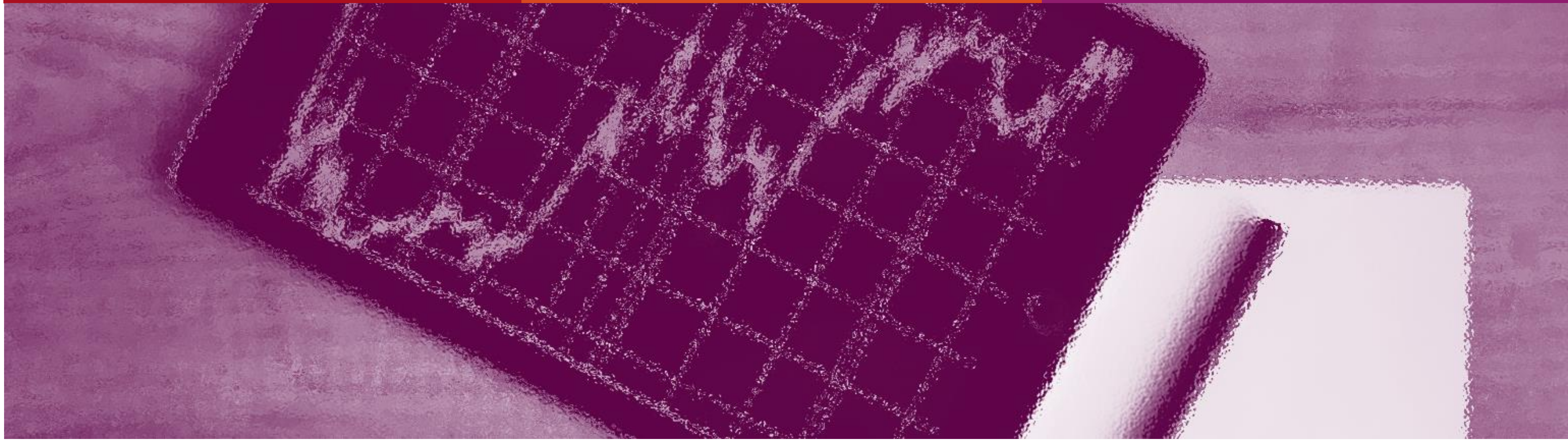


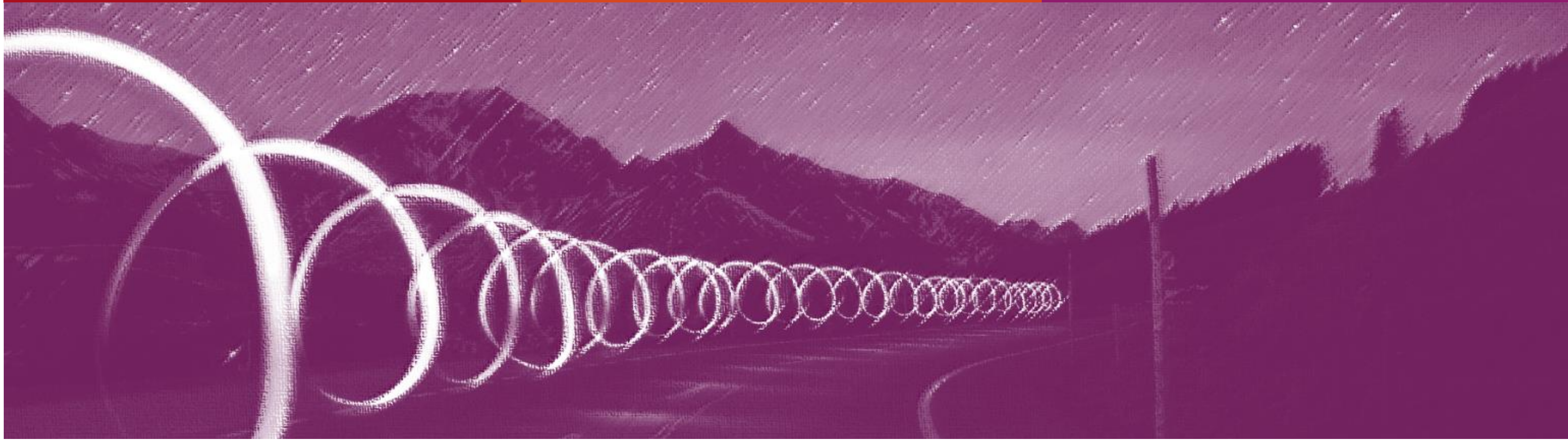
Time Series Encodings with Temporal Convolutional Networks

Unsupervised Representation Learning & Anomaly Detection

Markus Thill, Wolfgang Konen, and Thomas Bäck
{markus.thill, wolfgang.konen}@th-koeln.de, t.h.w.baeck@liacs.leidenuniv.nl

Technology
Arts Sciences
TH Köln

Universiteit
Leiden



Introduction

Motivation & introductory Example

Motivation: Anomaly Detection in Time Series

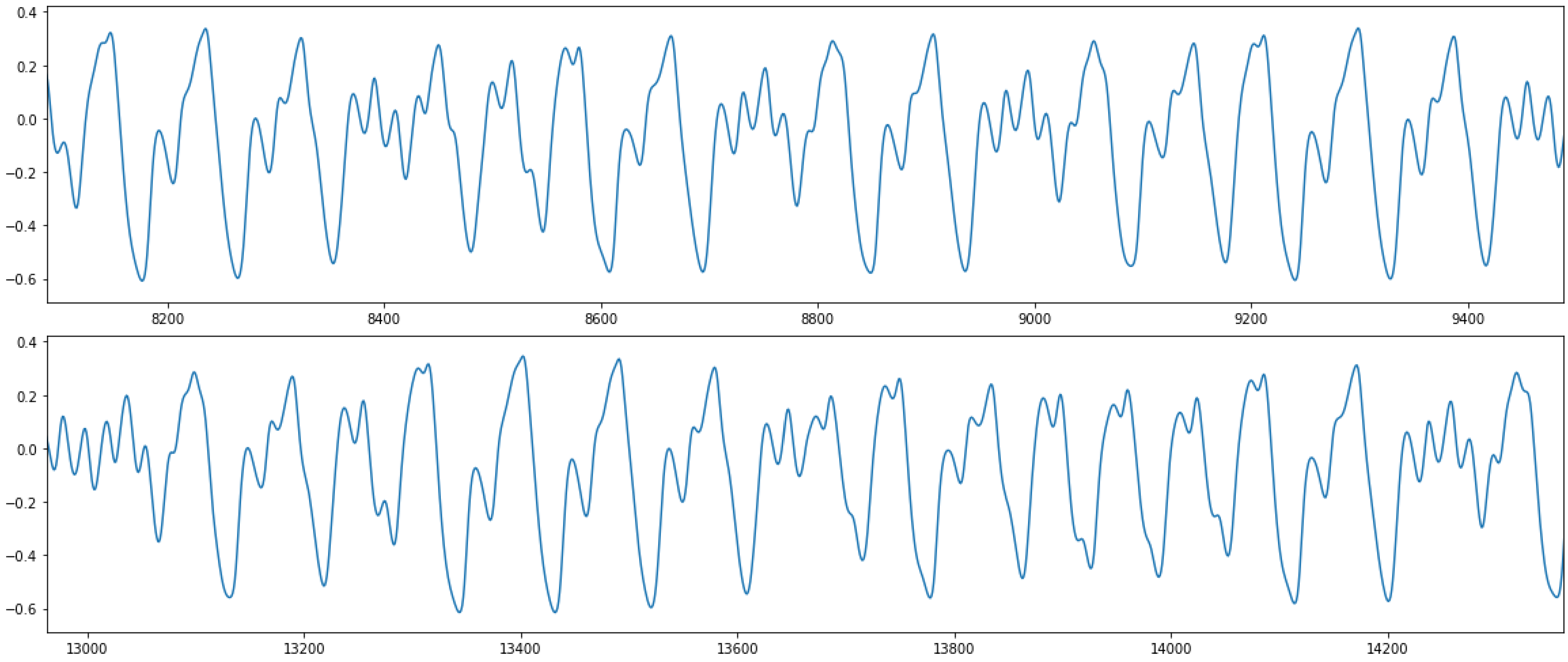
What is an Anomaly?

– An event that deviates from what ist standard, normal or expected
  – Context-based/Temporal Anomaly: value is unusual given the current temporal context
  – Concept Change: statistical properties of certain (target-) variables change over time

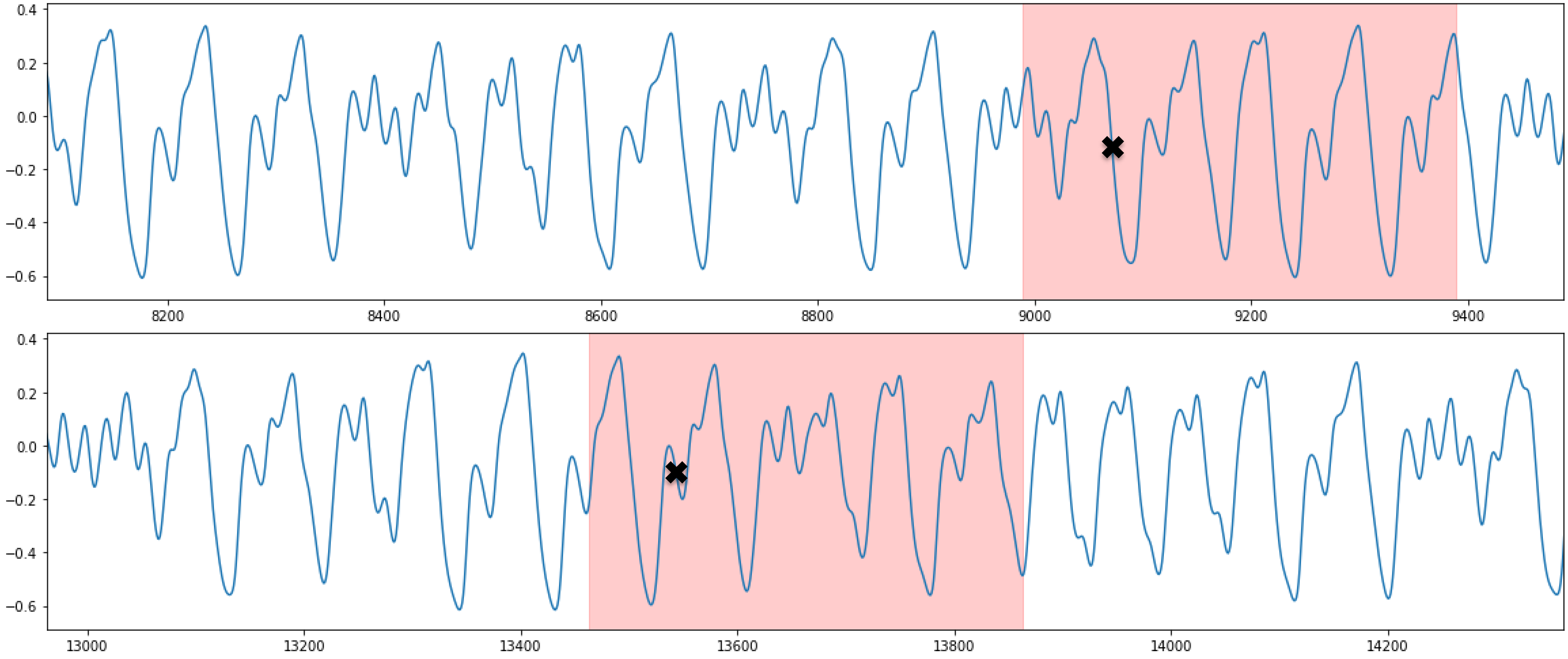
Application of Anomaly Detection Algorithms

– Fraud Detection, E-Commerce, Process Industry, Network Monitoring Tasks,
– Predictive Maintenance, Fault Detection
– Electrocardiogram Readings (ECG)

Motivation


0.4
0.2
0.0
−0.2
−0.4
−0.6
8200
8400
8600
8800
9000
9200
9400
13000
13200
13400
13600
13800
14000
14200

Motivation


0.4
0.2
0.0
−0.2
−0.4
−0.6
8200
8400
8600
8800
9000
9200
9400
0.4
0.2
0.0
−0.2
−0.4
−0.6
13000
13200
13400
13600
13800
14000
14200

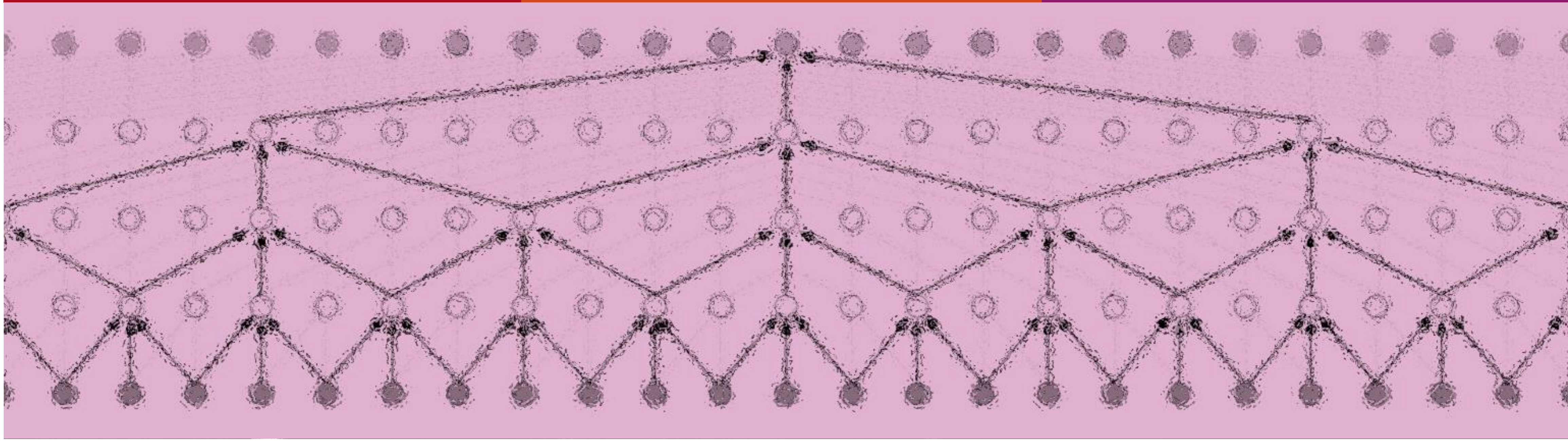
Contributions of the Paper

Temporal Convolutional Autoencoder (TCN-AE)

- An Autoencoder for time series (sequences) based on TCNs
- Applications for time series:
  - Representation learning
  - Unsupervised anomaly detection

Mackey-Glass Anomaly Benchmark (MGAB)

- Synthetic benchmark based on chaotic Mackey-Glass time series
- Well defined, but non-trivial anomalies
- Steerable parameters to generate customized benchmarks



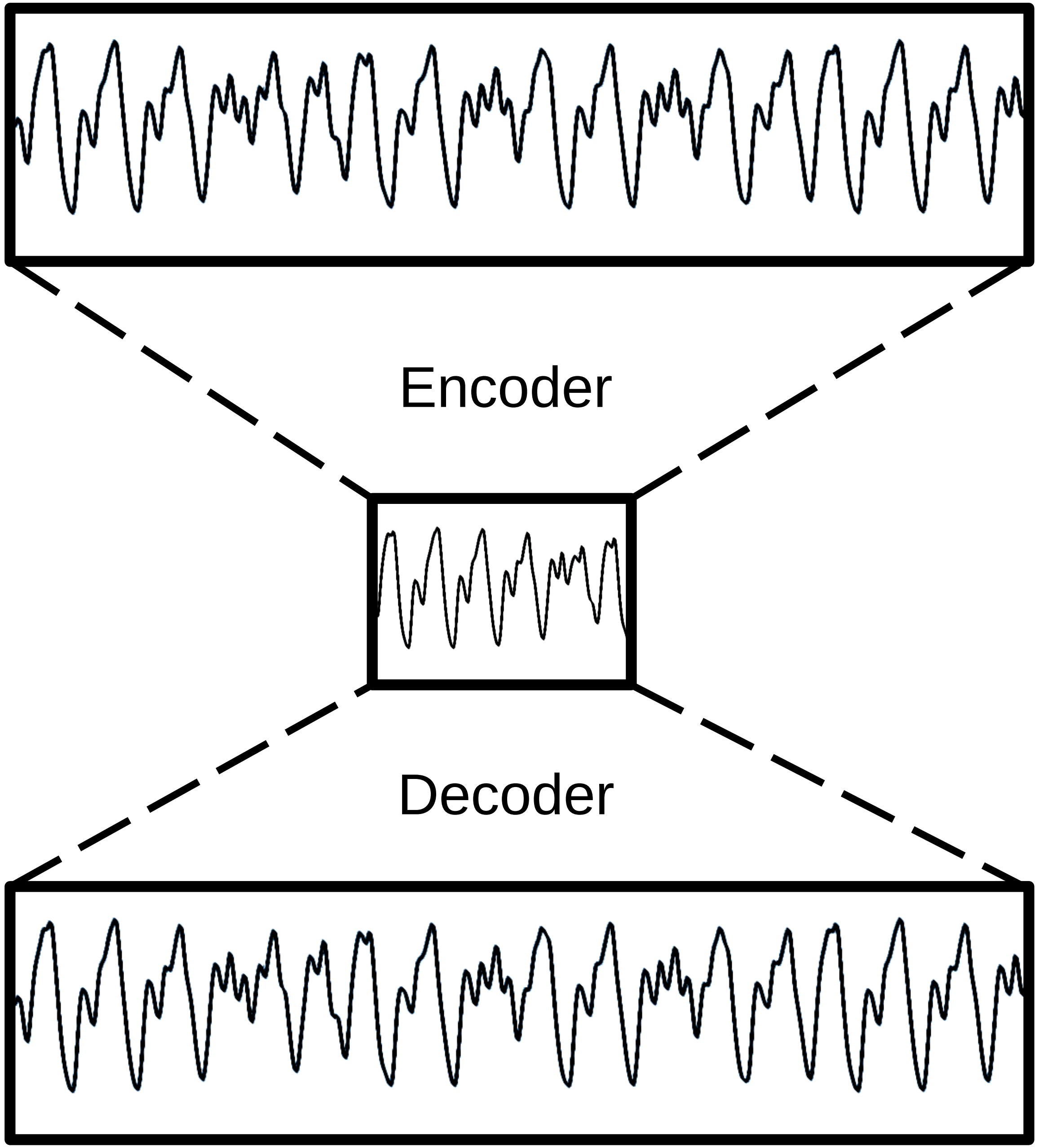
The Temporal Convolutional Autoencoder

An Autoencoder for sequential / Time Series Data

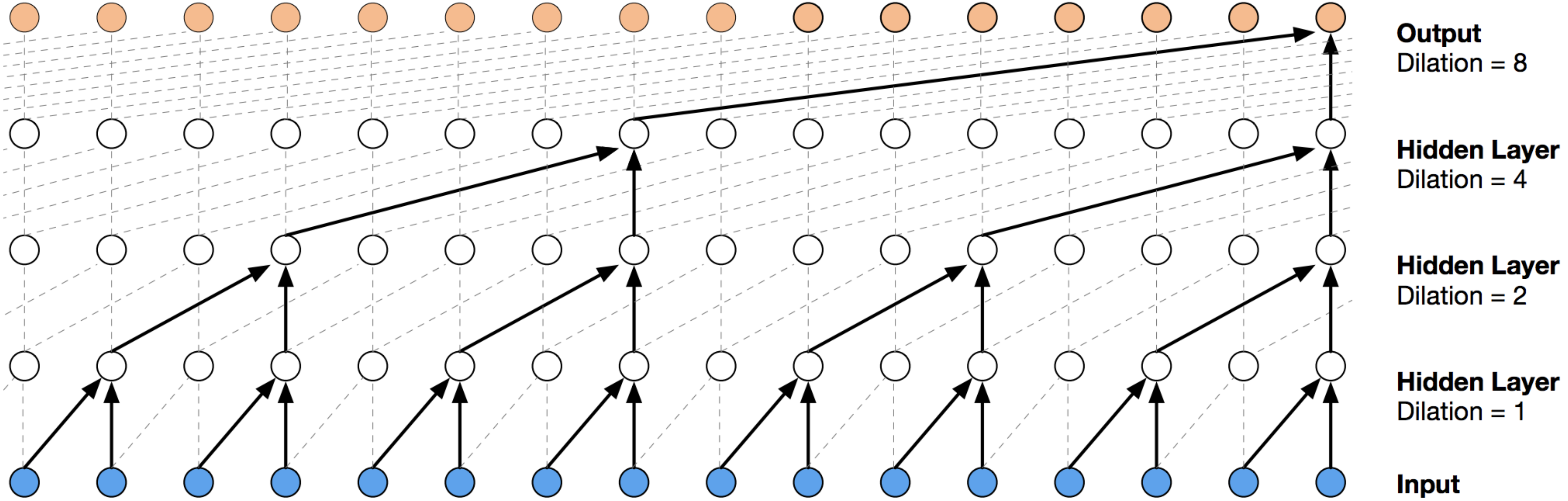
TCN-AE: Intuition

General Idea of TCN-AE Architecture

- Encode a sequence of length T into a significantly shorter sequence of length T/k
- Subsequently, reconstruct (decode) the original sequence from the compressed sequence
- Reconstruction error as indicator for anomalous behavior
- Approach similar to classical (dense) Autoencoder
- *But*: Uses dilated convolutional layers
  - more flexible w.r.t. input size
  - Large receptive field due to dilated filters



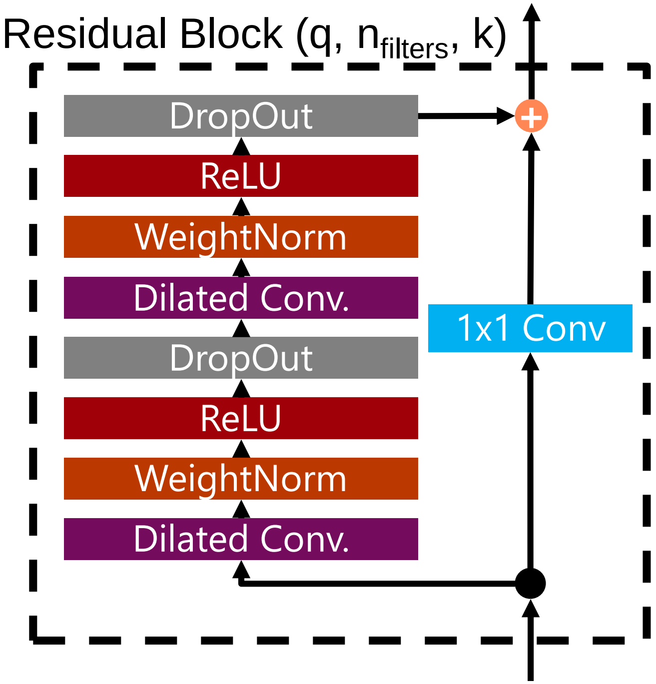
TCN-AE: Dilated Convolutions

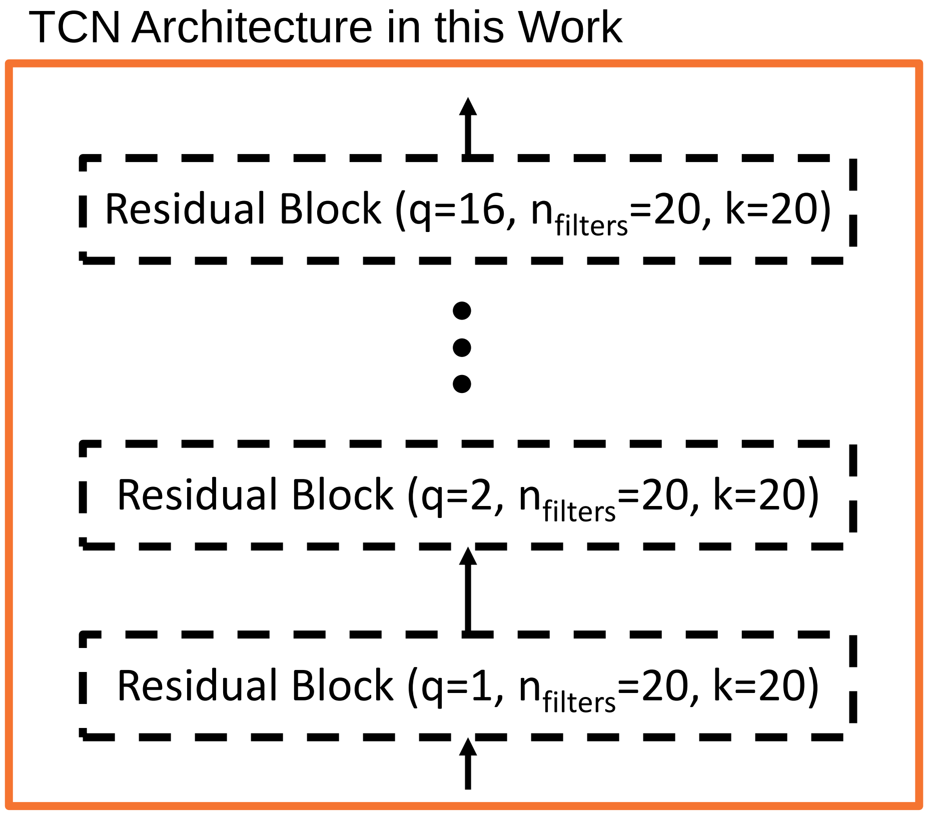


Oord, A. et al.: „WaveNet: A Generative Model for Raw Audio“. *arXiv:1609.03499 [cs]*, 2016. http://arxiv.org/abs/1609.03499.

Temporal Convolutional Networks (TCNs)

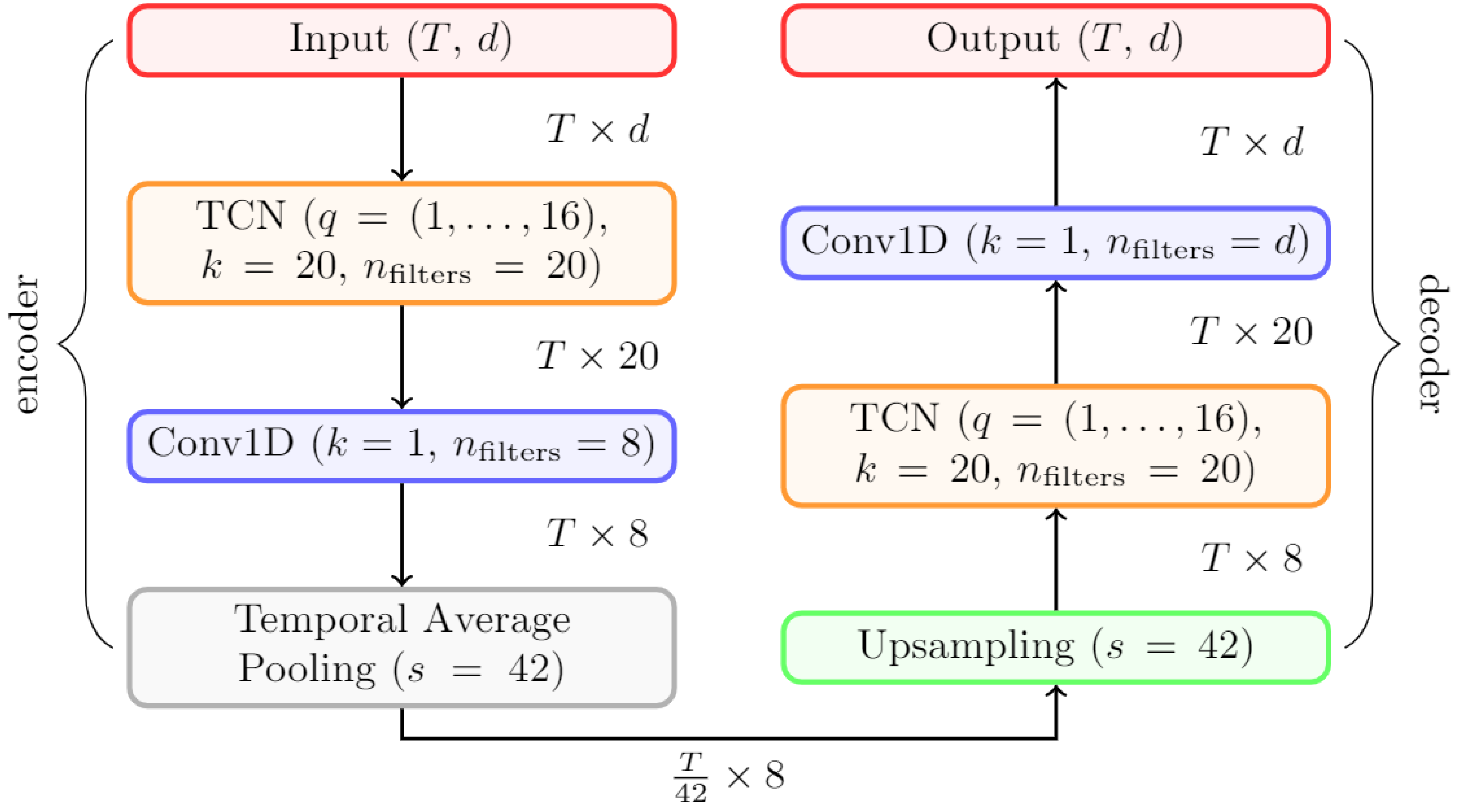
- TCN: Dilated Convolutions + Residual Connections + Weight Norm. + Dropout
- Mainly described by:
  - A list of dilation rates (q), e.g. (1,2,4,8,16)
  - The number of filters ($n_{filters}$)
  - The kernel size (k)

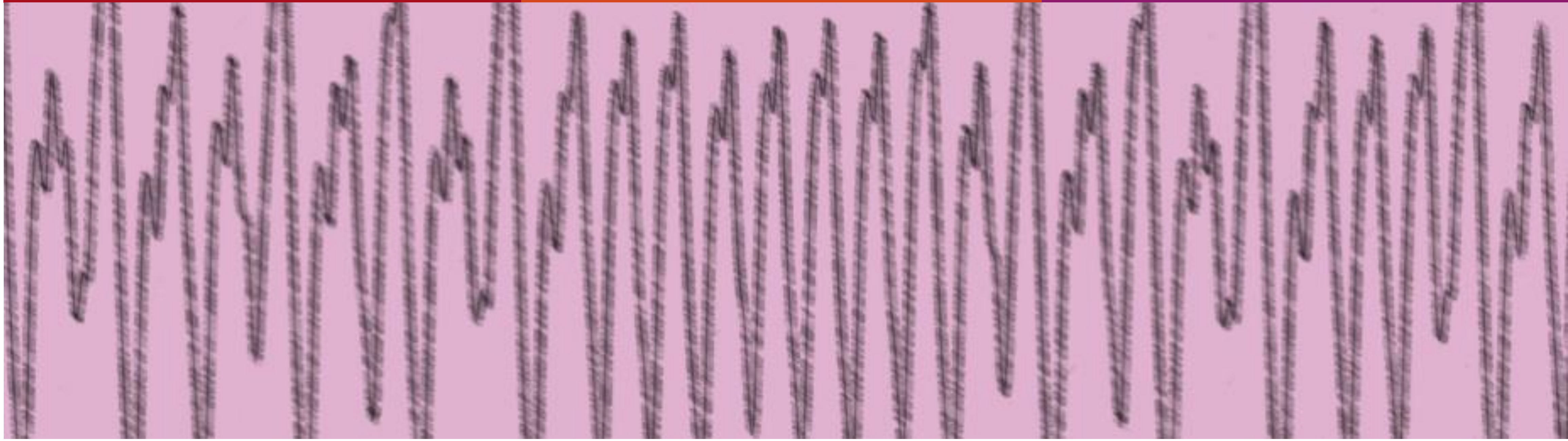




Bai, Shaojie, J. Zico Kolter, und Vladlen Koltun. „*An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*“. *arXiv:1803.01271*

The TCN Autoencoder (TCN-AE)





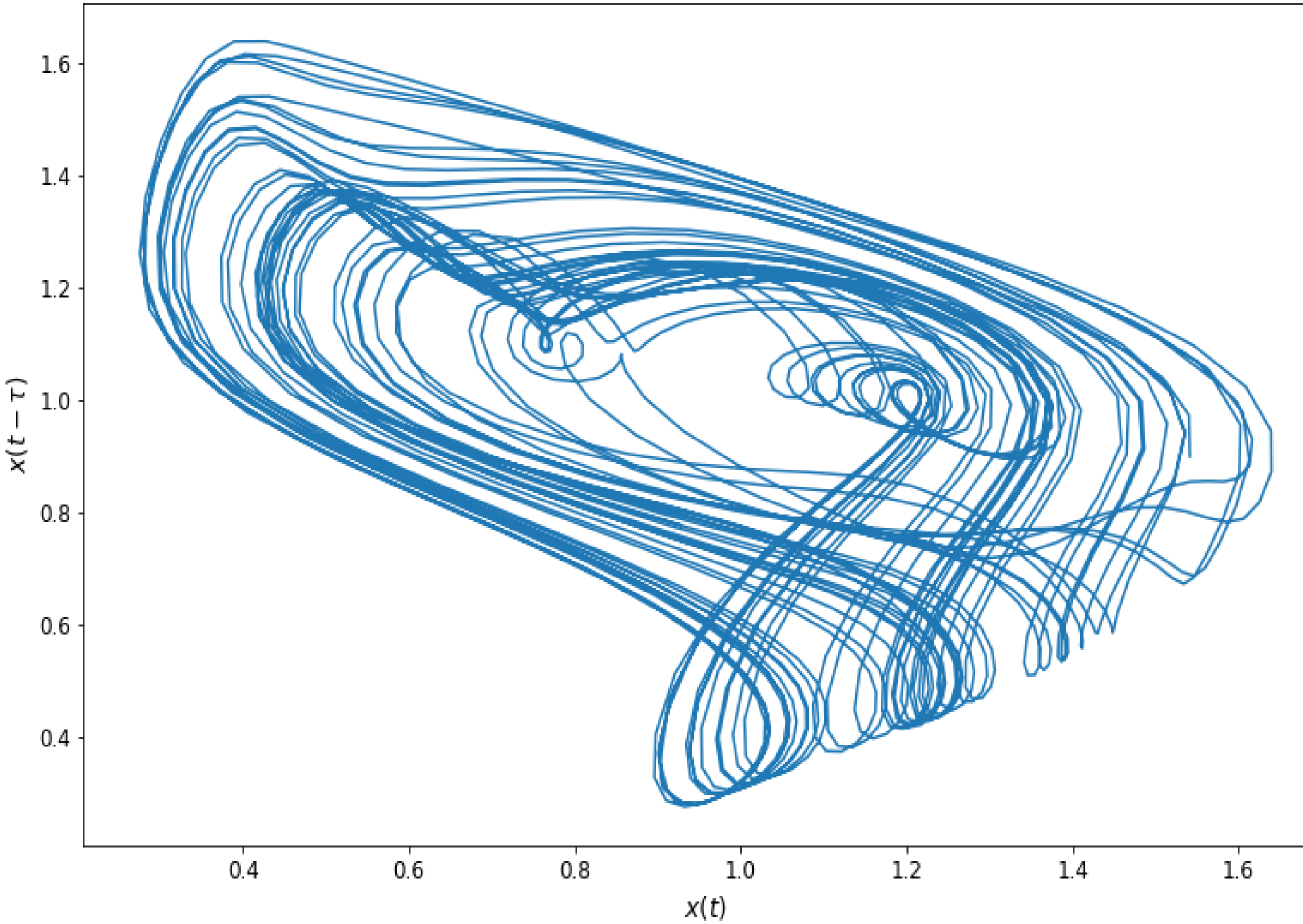
The Mackey-Glass Anomaly Benchmark

The Mackey-Glass Equation

- The Mackey-Glass equation: a so-called non-linear time delay differential equation (DDE), which is defined as:

$$\frac{dx}{dt} = \beta \cdot \frac{x(t-\tau)}{1+x(t-\tau)^n} - \gamma\, x(t)$$

- exhibits various patterns of chaotic and periodic dynamics
- Time delay embedding to visualize chaos
- Maximum Lyapunov Exponent:
  - Value of $\lambda_{mle} > 0$ indicates chaotic behavior
- Integrate with JiTCDDE package for Python



Mackey-Glass Anomaly Benchmark (MGAB)

Anomaly Insertion Process (Rough Sketch)

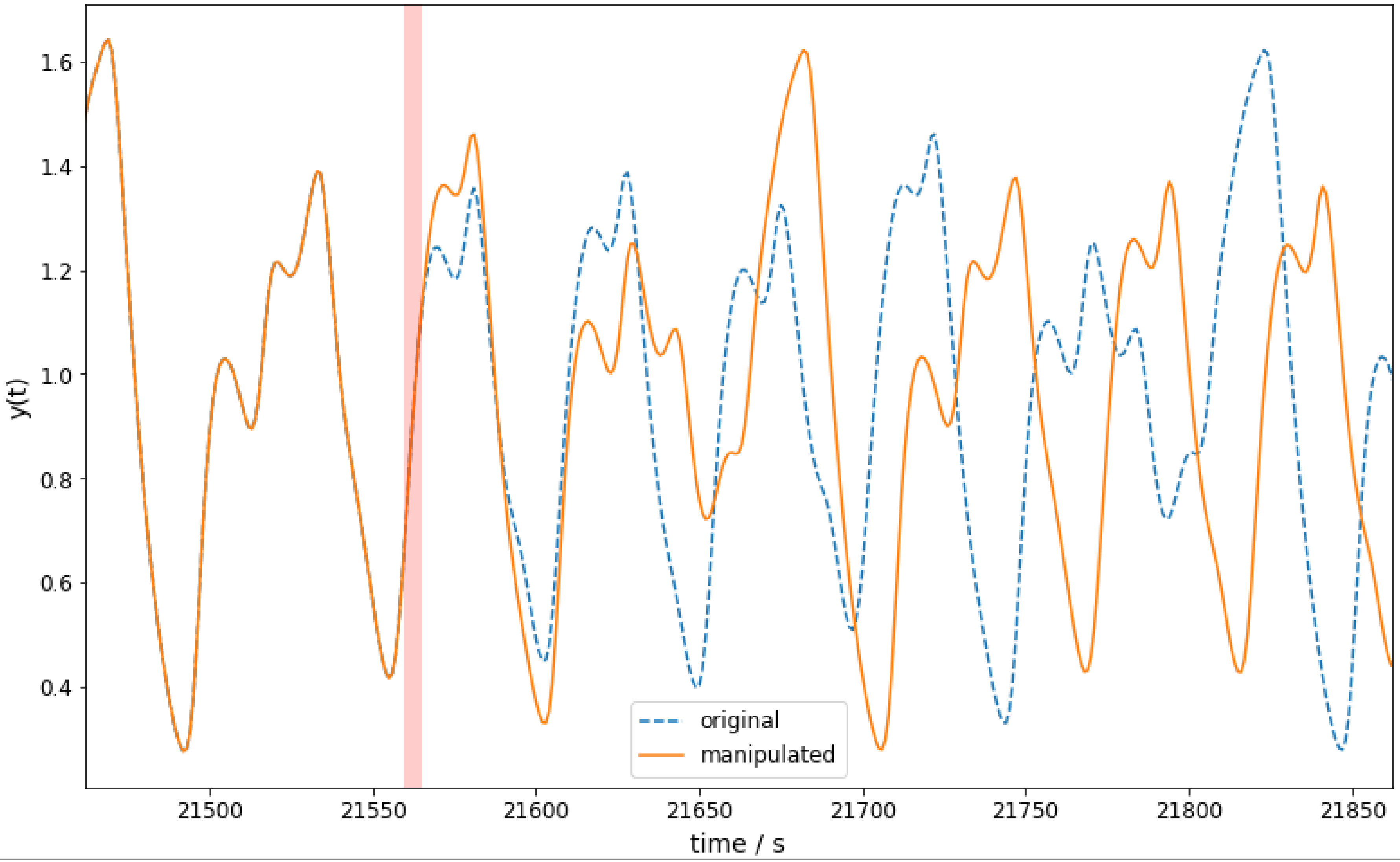
1. Create sufficiently long MG time series $x(t)$
2. Compute first 3 derivatives $\dot{x}(t),\ \ddot{x}(t),\ \dddot{x}(t)$
3. Randomly select a time t_i
4. Now find t_m, such that

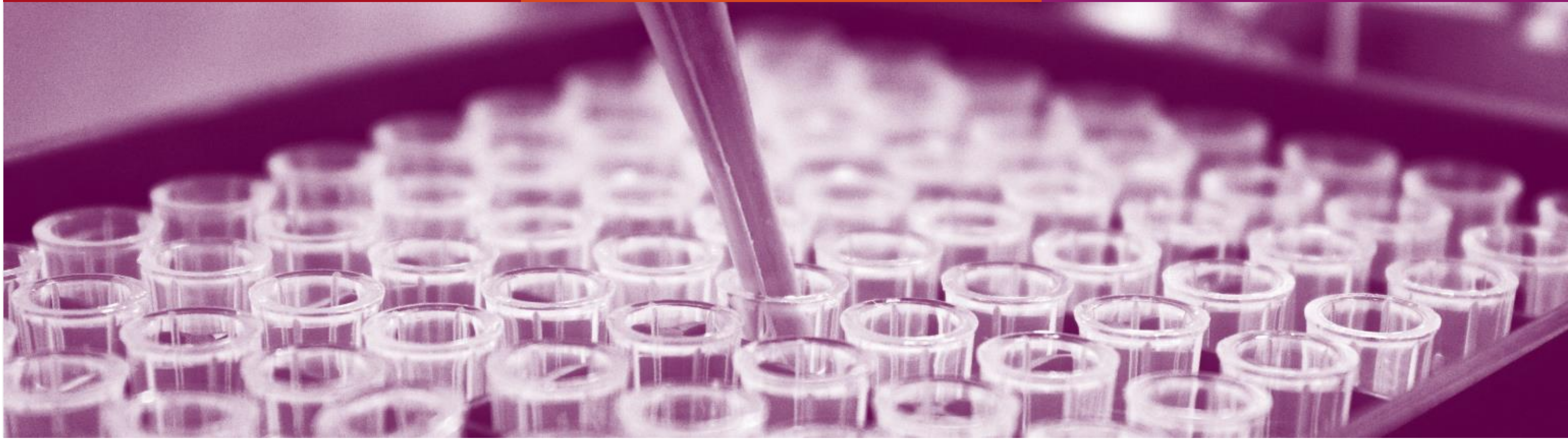
$$\mathrm{m} = \underset{i+T<k<i+T+M}{\arg\min} \left\| \begin{pmatrix} x(t_i) \\ \dot{x}(t_i) \\ \ddot{x}(t_i) \\ \dddot{x}(t_i) \end{pmatrix} - \begin{pmatrix} x(t_k) \\ \dot{x}(t_k) \\ \ddot{x}(t_k) \\ \dddot{x}(t_k) \end{pmatrix} \right\|$$

5. Remove the segment between t_i and t_m and stick both ends back together
6. Go to 3. Repeat until enough anomalies are inserted

GitHub: https://github.com/MarkusThill/MGAB

Mackey-Glass Anomaly Benchmark (MGAB): Insertion Process





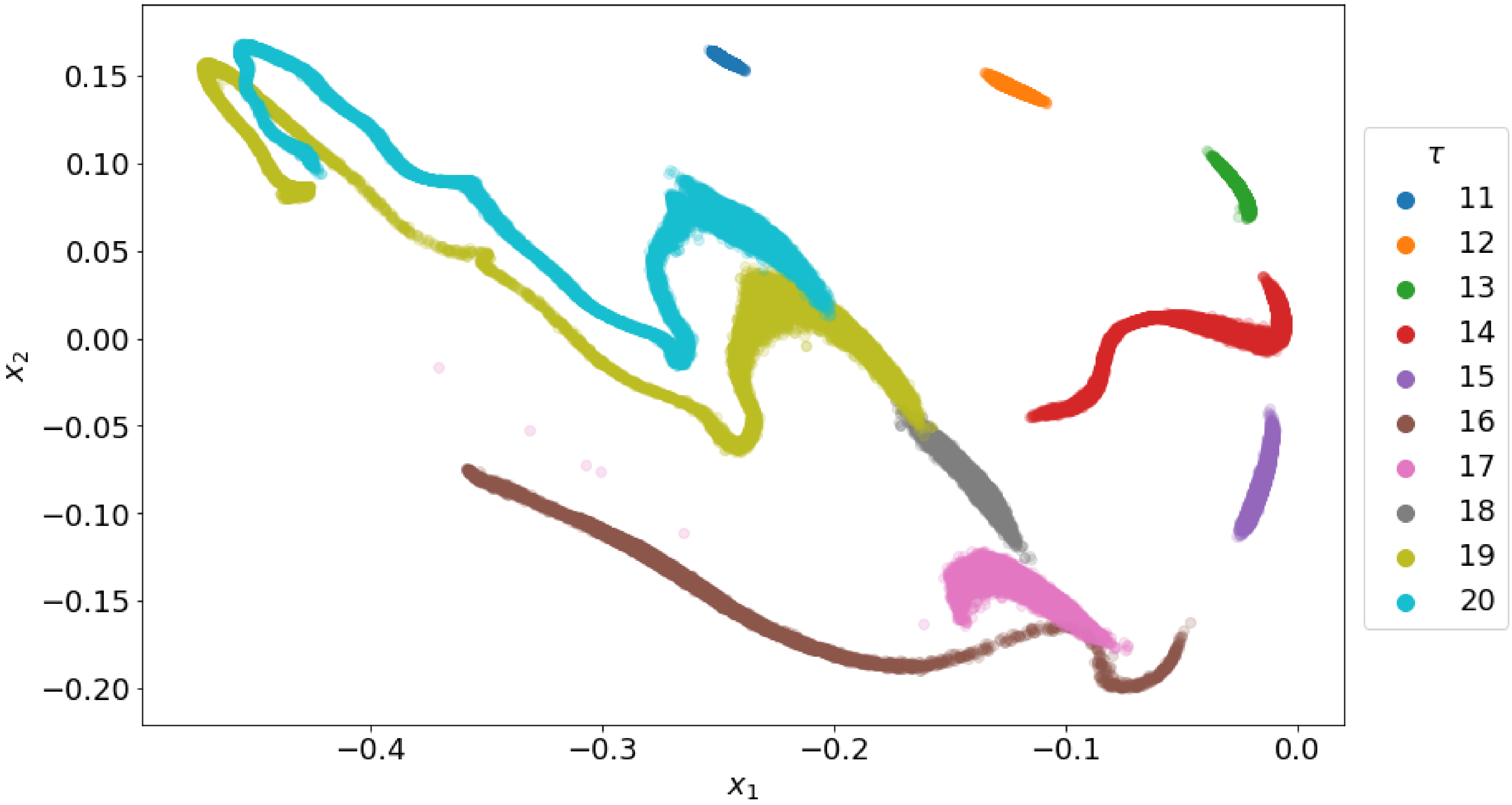
Experiments

Learning Time Series Representations & Anomaly Detection

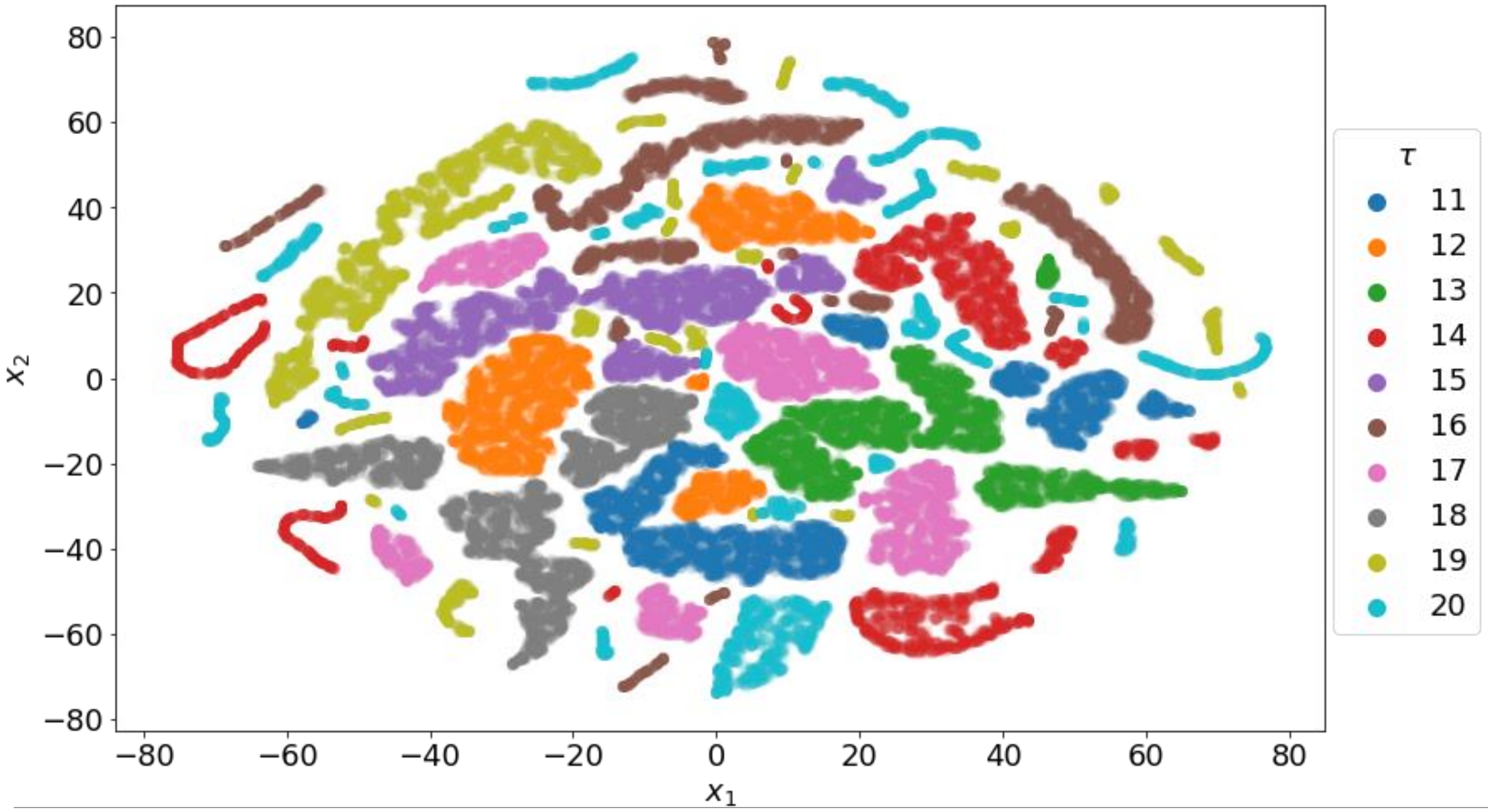
TCN-AE: Learning Time Series Representations

- Assess the capabilities of the TCN-AE architecture in learning representations of time series
- Train a TCN-AE model using many different MG time series with a varying time delay parameter τ
- 10^5 different Mackey-Glass time series (10^4 for each τ in the range of $\tau = 11 \dots 20$
- Each time series of length 256 is encoded into a 2-dimensional compressed representation
- compression rate of 128

Learning Time Series Representations
Results: TCN-AE



TCN-AE: Learning Time Series Representations
Results: Comparison with t-SNE



Anomaly Detection Task: Experimental Setup

Benchmark

- 10+1 MG time series of length 100 000 with 10 anomalies each (∑ 100+10 anomalies)
- Parameters: $\tau=18$, $n=10$, $\beta=0.25$, $\gamma=0.1$, $h=0.9$
- Train on 1 time series, test on 10 others
- Select anomaly threshold based on 10% of the anomaly labels (similarly to CV)

Algorithms

- TCN-AE: This work
- NuPIC [2]: Based on Numenta's HTM algorithm
- DNN-AE [3]: Based on a Deep Autoencoder
- LSTM-ED [4]: Encoder-Decoder Model
- LSTM-AD [5]: Our Previous Work

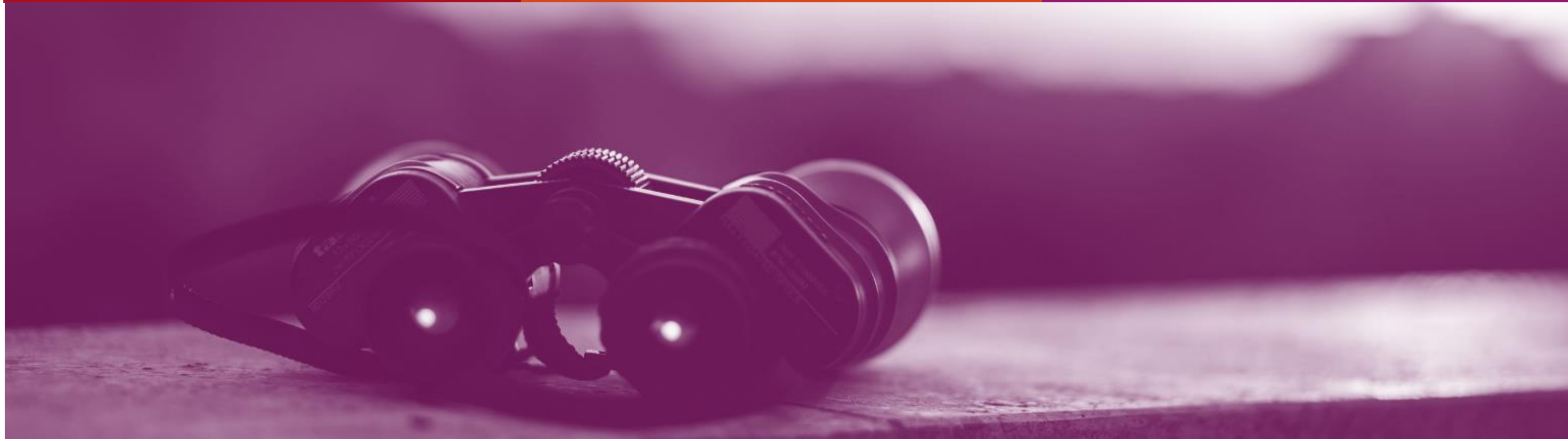
Table 1: Number of Trainable Weights for each Algorithm

Algorithm	Trainable Weights
NuPIC	-
TCN-AE (This Work)	**164,451**
LSTM-ED	244,101
DNN-AE	241,526
LSTM-AD	464,537

Anomaly Detection Task: Results for MGAB

Table 2: The results shown here (mean and standard deviation of 10 runs and 10 sub-sequences, are for the sum of TP, FN and FP over all 10 time series.

Algorithm	TP	FN	FP	Precision	Recall	F_1-Score
NuPIC [2]	3.00 ± 0.00	97.00 ± 0.00	132 ± 0.00	0.02 ± 0.00	0.03 ± 0.00	0.03 ± 0.00
LSTM-ED [4]	14.60 ± 5.86	85.40 ± 5.86	57.00 ± 20.43	0.21 ± 0.08	0.15 ± 0.06	0.17 ± 0.06
DNN-AE [3]	91.79 ± 1.22	8.21 ± 1.22	62.58 ± 13.65	0.60 ± 0.06	0.92 ± 0.01	0.72 ± 0.04
LSTM-AD [5]	88.80 ± 2.59	11.20 ± 2.59	0.62 ± 0.61	0.99 ± 0.01	0.89 ± 0.03	0.94 ± 0.01
TCN-AE [this work]	$\mathbf{90.54 \pm 1.72}$	$\mathbf{9.46 \pm 1.72}$	$\mathbf{0.20 \pm 0.47}$	$\mathbf{1.00 \pm 0.01}$	$\mathbf{0.91 \pm 0.02}$	$\mathbf{0.95 \pm 0.01}$



Conclusion & Outlook

Summarizing Remarks and Future Work

Conclusion and Future Work

Summary

- We proposed temporal autoencoder Architecture TCN-AE
- Evaluation of TCN-AE on two tasks:
  - Representation Learning
  - Anomaly Detection (unsupervised)
- Introduction of the Mackey-Glass Anomaly Benchmark (MGAB)
  - Synthetic benchmark with non-trivial anomalies
  - Steerable generation of custom time series

Future Work

- Gain more insights from the representations learnt by the TCN-AE
- Enhancements of the TCN-AE architecture, for example:
  - Reduce sensitivity towards dilation rate (Feature Reuse)
  - Investigate other types of bottlenecks than plain down-sampling
- Real-world anomaly detection tasks, e.g. ECG signals or industrial monitoring tasks

Thank You!