# Comprehensive Annotation of Temporal Patterns in the Time Series Segmentation Benchmark (TSSB)

## Study Project - Exposé

Sunita Singh

Department of Computer Science, Humboldt-Universität zu Berlin
Supervisors: Prof. Dr. Ulf Leser, Arik Ermshaus
October 23, 2024

## 1 Introduction

Time Series Analysis is a research area that focuses on analyzing sequences of data points indexed in time to extract meaningful insights and patterns. This includes identifying trends, seasons, and random or irregular movements within the data. Examples of such time series (TS) include electrocardiograms (ECGs), stock market prices, and weather data. A key aspect to gain insight into processes is to observe local substructures, also referred to as subsequences or windows. Since the size of these windows is a key factor in most state-of-the-art time series data mining (TSDM) algorithms, it is regarded as a critical hyperparameter [3]. Thus, finding the right window size is a task in itself. If the window size is chosen too large or too small, significant patterns may be missed, leading to an incomplete or inaccurate interpretation of the data. The challenge of selecting the correct window size becomes even more pronounced in unsupervised settings, where there is no labeled data to guide the choice. Determining the correct window size is essential for the successful execution of TSDM tasks, as it enables the accurate identification of meaningful patterns and improves the overall effectiveness of the analysis.

Examples for such unsupervised TSDM tasks are anomaly detection [1], segmentation [4, 10] or motif discovery [7, 11]. As described in [3], time series anomaly detection (TSAD) aims to find erroneous or novel behaviour within expected observations. The task of time series segmentation (TSS) is to find a meaningful segmentation of time series [10]. The segments usually consist of periodically repeating substructures, with the specific substructures varying across the respective segments. Motif discovery, on the other hand, seeks to identify frequently occurring, similar subsequences, or patterns, within a time series, which can reveal underlying repetitive behaviors or structures [3]. In this study project, the primary focus will be on the task of time series segmentation.

The tasks TSAD and TSS are often applied to TS with recurring substructures. As descriped in [3] a periodic TS is one which approximately repeats a subse-

quence of values after a fixed length of time. Such subsequences are referred as a temporal pattern or period of a TS and occur consecutively. In a TS with multiple segments, the temporal pattern repeats within one segment until a changepoint is encountered, at which point the segment transitions to a new one. After the changepoint, a new temporal pattern can be identified.
In contrast to TSAD and TSS, data utilized for the task of motif discovery do not require periodicity. This distinction is significant because window size algorithms leverage the information of periodicity for optimal selection.

The selection of the appropriate window size can be accomplished manually by experts observing the data or automatically derived by algorithms. Since manually determination is time-consuming, requires expert knowledge, and may not always yield the optimal results, utilizing an algorithmic approach for this calculation presents a more efficient and potentially more accurate alternative. Several domain-agnostic methods have been proposed. To evaluate the effectiveness and quality of these different methods, benchmarking is essential. The aim of this study project is to perform a detailed annotation of temporal patterns found within the Time Series Segmentation Benchmark (TSSB). This will provide a benchmark for future research on window size selection (WSS) algorithms. In the next section we will discuss further background.

## 2 Background

Time series data, which captures the sequential behavior of processes over time, is fundamental in many analytical tasks across diverse domains.

**Definition 1 (Time Series).** *A time series $T$ is a sequence of $n \in \mathbb{N}$ real values, $T = (t_1, \ldots, t_n), t_i \in \mathbb{R}$ that measures an oberserable output of a process [3].*

To gain meaningful insights from the data, our focus extends beyond the entire time series $T$ to also include subsequences.

**Definition 2 (Subsequence).** *A subsquence (or window) $T_{s,e}$ with start offset $s$ and end offset $e$ consists of the contiguous values of $T$ from position $s$ to position $e$, i.e, $T_{s,e} = (T_s, \ldots, T_e)$ with $1 \leq s \leq e \leq n$. The lenghth (or with) of $||T_{s,e}|| = e - s + 1$ [3].*

Such subsequences can correspond to temporal patterns, which are utilized in time series segmentation and anomaly detection.

**Definition 3 (Temporal Pattern).** *A temporal pattern in a periodic time series $T$ is the longest non-repeating subsequence $T_{s,e}$, with start offset $s$ and end offset $e$, such that the values within $T_{s,e}$ approximately and consecutively repeat until a changepoint $cp$ is reached, where $1 \leq s < e \leq cp \leq n$.*

In TSS, each segment typically has a distinct temporal pattern that changes when a new segment begins. We assume that the optimal window size corresponds to the length of the temporal patterns. We can now formally define the window size selection problem within the context of periodic time series.

**Definition 4 (Window Size Selection Problem).** *The window size selection problem involves determining the optimal window length $w \in \mathbb{N}$ for a time series $T$ to accurately capture temporal patterns. The selected window size should balance capturing small recurring subsequences without introducing noise or missing key details.*

The next section will discuss related work regarding benchmarking and window size selection (WSS) algorithms.

## 3 Related Work

### 3.1 Window Size Selection Algorithms

WSS algorithms can be distinguished in the categories whole series-based and subsequence-based. Whole series-based methods analyze the global properties of the time series to identify dominant period sizes and can be further divided into frequency-based and time-based methods. In contrast, subsequence-based approaches focus on extracting local features from the time series [3].
There are six established and recent domain-agnostic WSS algorithms, which will be described in the following. (1) The *Dominant Fourier Frequency* algorithm uses the Fourier transform to identify the most dominant frequency component in the time series, assuming to be the to the optimal window size. (2) The *Autocorrelation* algorithm computes the autocorrelation function of the time series to find the lag with the highest correlation, indicating the periodicity and thus the appropriate window size. (3) Another approach, *AutoPeriod* [14], is a hybrid method that combines both Fourier and autocorrelation techniques to determine the window size by leveraging the strengths of both methods. (4) In contrast, *RobustPeriod* [15] first detrends the time series and then uses wavelet transforms and autocorrelation to identify multiple dominant periods, selecting the most significant one as the window size. (5) *Multi-Window Finder (MWF)* [5] is a differnt approach, which is a subsequence-based method that evaluates the variance in moving averages over different window sizes, choosing the window size that minimizes this variance. (6) Finally, the *Summary Statistics Subsequence (SuSS)* [2] approach compares summary statistics (mean, standard deviation, range) of subsequences with those of the entire time series to find the window size where these statistics best align.

### 3.2 Datasets and Benchmarking

As previously mentioned, benchmarking is essential for evaluating the effectiveness and quality of WSS algorithms. However, since no comprehensive annotation of temporal patterns has been proposed yet, previous research had to utilize

alternative benchmarking approaches.
In [3], Ermshaus et al. used anomaly detection, segmentation and motif discovery as downstream tasks to evaluate WSS algorithms. The window sizes computed by the WSS algorithms were provided as inputs to TSDM algorithms. Subsequently, the performance of the TSDM algorithms was compared to determine which WSS algorithm was most effective. For instance, the window sizes determined by FFT and MWF were used as inputs for TSS algorithms such as ClaSP and FLOSS. The study then assessed whether FFT or MWF resulted in better performance for the TSS algorithms and examined whether the observed differences were statistically significant.
To assess the WSS approaches on segmentation, they utilized the Time Series Segmentation Benchmark (TSSB) [13]. TSSB currently contains 75 annotated TS with 1-9 segments. Given that the dataset includes a variety of device, medical, image, motion, and other sensor data [3], it is well-suited for evaluating domain-agnostic algorithms. For the evaluation of the anomaly detection the HEX UCR Anomaly Benchmark 2021 (HUAB) [6] was used. Since there is no annotated benchmark for motif discovery, two TS of the literature were utilizied [8, 9].
Since we focus on TSS and assume that the optimal window size is equal to the length of the temporal pattern, it is not possible to definitively determine which WSS algorithm performs the best using an indirect evaluation approach. In contrast, for TSAD, such indirect evaluation can be more effective because anomalies can be significantly longer than the temporal pattern, making it challenging to capture them if the focus is solely on the length of the temporal pattern.

## 4 Project Goal

The goal of this study project is to conduct a comprehensive annotation of temporal patterns within the Time Series Segmentation Benchmark (TSSB). Subsequently, the annotated dataset will be utilized to evaluate the WSS algorithms introduced in 3.1. This annotation will enable a more precise evaluation of the WSS algorithms. Given that the annotation will be restricted to the TSSB, the study will focus exclusively on the task of time series segmentation.

### 4.1 Approach

For the annotation, we will manually review each time series in the TSSB to determine the locations of the temporal patterns. A software program called "SeriesExplorer" [12] will be utilized to assist in this annotation process by aiding in the labeling of patterns and storing the results in a CSV file. Figure 1 presents an example snippet of a time series plot with six annotated temporal patterns. Upon completing the annotation of the time series in the dataset, we will determine a window size for each time series and for each segment by calculating the differences between the annotations and subsequently computing the mean or median of these differences. For the example in Figure 1 the computation would approximately provide a value of 60 data points, representing the window size. In this study project, we will then investigate whether there are variations in window sizes within the individual time series or between different segments of the time series. The variation in window size within a segment is expected to be minimal; however, the variation between different segments may be more substantial.

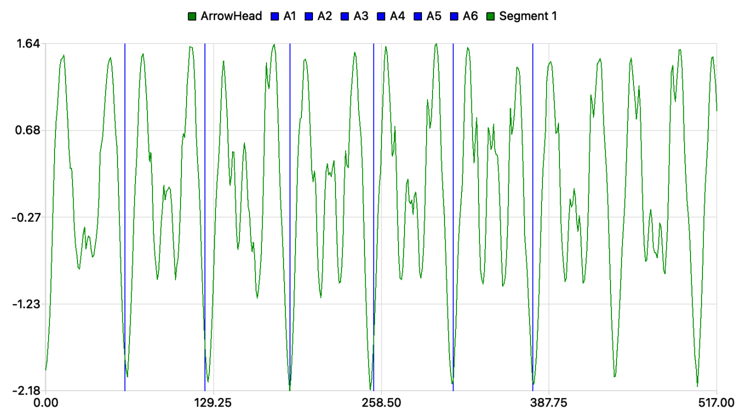


Fig. 1: Example of Time Series Plot with Annotated Temporal Patterns

### 4.2 Evaluation

After completing the annotation and determining the window sizes for each time series in the TSSB, we will proceed to evaluate the window size selection (WSS) algorithms introduced in 3.1. With the established ground truth window sizes, we will apply the Root Mean Square Error (RMSE) as our evaluation metric. For each approach, we will calculate the RMSE between the predicted window sizes and the derived window sizes from our annotations. This methodology will allow us to identify the most effective algorithm. Since this will be the first dataset with annotated temporal patterns, it will provide future research on the topic of window size selection with the opportunity to compare their results with other approaches.

## References

[1] Ane Blázquez-García et al. “A review on outlier/anomaly detection in time series data”. In: *ACM computing surveys (CSUR)* 54.3 (2021), pp. 1–33.

[2] Arik Ermshaus, Patrick Schäfer, and Ulf Leser. “ClaSP: parameter-free time series segmentation”. In: *Data Mining and Knowledge Discovery* 37.3 (2023), pp. 1262–1300.

[3] Arik Ermshaus, Patrick Schäfer, and Ulf Leser. “Window size selection in unsupervised time series analytics: A review and benchmark”. In: *International Workshop on Advanced Analytics and Learning on Temporal Data*. Springer. 2023, pp. 83–101.

[4] Shaghayegh Gharghabi et al. “Domain agnostic online semantic segmentation for multi-dimensional time series”. In: *Data mining and knowledge discovery* 33 (2019), pp. 96–130.

[5] Shima Imani and Eamonn Keogh. “Multi-window-finder: domain agnostic window size for time series data”. In: *Proceedings of the MileTS* 21 (2021).

[6] *Keogh, E., Dutta Roy, T., Naik, U., Agrawal, A: Multi-dataset timeseries anomaly detection competition (2021).* `https://compete.hexagon-ml.com/practice/competition/39/`. Accessed: 2024-07-28.

[7] JLEKS Lonardi and Pranav Patel. “Finding motifs in time series”. In: *Proc. of the 2nd Workshop on Temporal Data Mining*. 2002, pp. 53–68.

[8] Fabian Mörchen and Alfred Ultsch. “Efficient mining of understandable patterns from multivariate interval time series”. In: *Data mining and knowledge discovery* 15 (2007), pp. 181–215.

[9] Simona Petrutiu, Alan V Sahakian, and Steven Swiryn. “Abrupt changes in fibrillatory wave characteristics at the termination of paroxysmal atrial fibrillation in humans”. In: *Europace* 9.7 (2007), pp. 466–470.

[10] Patrick Schäfer, Arik Ermshaus, and Ulf Leser. “Clasp-time series segmentation”. In: *Proceedings of the 30th ACM international conference on information & knowledge management*. 2021, pp. 1578–1587.

[11] Patrick Schäfer and Ulf Leser. “Motiflets: Simple and Accurate Detection of Motifs in Time Series”. In: *Proceedings of the VLDB Endowment* 16.4 (2022), pp. 725–737.

[12] *Series Explorer*. `https://seriesexplorer.de`. Accessed: 2024-10-21.

[13] *Time Series Segmentation Benchmark (2021).* `https://github.com/ermshaua/time-series-segmentation-benchmark`. Accessed: 2024-07-27.

[14] Michail Vlachos, Philip Yu, and Vittorio Castelli. “On periodicity detection and structural periodic similarity”. In: *Proceedings of the 2005 SIAM international conference on data mining*. SIAM. 2005, pp. 449–460.

[15] Qingsong Wen et al. “RobustPeriod: Robust time-frequency mining for multiple periodicity detection”. In: *Proceedings of the 2021 international conference on management of data*. 2021, pp. 2328–2337.