Distribution Controlled Clustering of Time Series Segments by Reduced Embeddings

Gábor Szűcs[a], Marcell Balázs Tóth[b] and Marcell Németh[c]
Budapest University of Technology and Economics, Department of Telecommunications and Artificial Intelligence, Hungary

Keywords: Clustering, Time Series Segments, Kolmogorov–Smirnov Test, COP-KMeans, Clustering Evaluation, Confidence Score.

Abstract: This paper introduces a novel framework for clustering time series segments, addressing challenges like temporal misalignment, varying segment lengths, and computational inefficiencies. The method combines the Kolmogorov–Smirnov (KS) test for statistical segment comparison and adapted COP-KMeans for clustering with temporal constraints. To enhance scalability, we propose a basepoint selection strategy for embedding the time series segments that reduces the computational complexity from $O(n^2)$ to $O(n \cdot b)$ by limiting comparisons to representative basepoints. The approach is evaluated on diverse time series datasets from domains such as motion tracking and medical signals. Results show improved runtime performance over traditional methods, particularly for large datasets. In addition, we introduce a confidence score to quantify the reliability of cluster assignments, with higher accuracy achieved by filtering low-confidence segments. We evaluated clustering performance using the Rand Index (RI), Adjusted Rand Index (ARI), and Normalized Mutual Information (NMI). Our results demonstrate advantageous properties of the method in handling noise and different time series data, making it suitable for large scale applications.

1 INTRODUCTION

Time series clustering is a crucial task in machine learning with applications across various domains, including finance, healthcare, and industry. It involves grouping similar sequences of data points that evolve over time. However, this process poses challenges due to temporal dependencies and varying segment lengths. Traditional methods, such as whole time series and subsequence clustering (Aghabozorgi et al., 2015; Zolhavarieh et al., 2014; Caiado et al., 2015; Fujimaki et al., 2008), either struggle with large datasets or fail to account for such complexities.

Subsequence clustering, which partitions a long time series into segments for clustering, is particularly relevant for identifying recurring patterns in datasets. However, existing approaches often have difficulty in dealing with temporal misalignment and variability in segment length. Furthermore, at large datasets, the computational inefficiency of traditional clustering algorithms is a significant limitation.

In this paper, we propose a novel clustering framework that integrates the Kolmogorov-Smirnov (KS) test (Kolmogorov, 1933) for statistical comparison of segments to address temporal misalignment. To further enhance clustering, we apply COP-KMeans (Wagstaff et al., 2001), incorporating pairwise constraints to ensure that consecutive time series segments are not assigned to the same cluster, thereby preserving their temporal independence. Our method effectively handles large datasets while maintaining accuracy. The key contributions of this paper are:

- A new subsequence clustering method combining KS test statistic and adapted constrained clustering algorithm (COP-KMeans) for flexible time series segment clustering.
- A reduced-embedding-based clustering method with base point selection to decrease computational complexity.
- Comprehensive experimental results demonstrating the method's efficiency and accuracy on diverse datasets.
- A confidence score to quantify the reliability of cluster assignments, where higher accuracy achieved by filtering low-confidence segments.

[a] https://orcid.org/0000-0002-5781-1088
[b] https://orcid.org/0009-0007-7136-314X
[c] https://orcid.org/0009-0002-2835-0363

SzÅścs, G., Tóth, M. B. and Németh, M.
Distribution Controlled Clustering of Time Series Segments by Reduced Embeddings.
DOI: 10.5220/0013162700003905
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 14th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2025)*, pages 43-54
ISBN: 978-989-758-730-6; ISSN: 2184-4313
Proceedings Copyright © 2025 by SCITEPRESS – Science and Technology Publications, Lda.

The remainder of this paper is organized as follows: Section 2 reviews related work, Section 3 explains the theoretical foundations, Section 4 and Section 5 detail the proposed approach, and Section 6 presents experimental results and in the Section 7 we draw the conclusions.

2 RELATED WORKS

The literature on time series clustering can be classified into three main categories: whole time series, subsequence, and time-point clustering (Aghabozorgi et al., 2015; Zolhavarieh et al., 2014).

Whole time series clustering involves clustering a set of individual time series (where each time series is treated as a distinct instance for the clustering algorithm) based on their similarity. This category has the most extensive body of research and can be further divided into three subcategories: shape-based, feature-based, and model-based approaches.

In the shape-based approach (Meesrikamolkul et al., 2012; Li et al., 2022), the shapes of two time series are aligned as closely as possible by applying non-linear stretching and contracting along the time axes. In the feature-based approach (Hautamaki et al., 2008), raw time series data are transformed into lower-dimensional feature vectors, which are then used as inputs for traditional clustering algorithms. Finally, in model-based methods (Liao, 2005), the raw time series are converted into model parameters, and clustering is performed based on these extracted parameters using an appropriate distance measure.

Subsequence clustering (Keogh and Lin, 2005; Zolhavarieh et al., 2014) focuses on clustering subsequences extracted from a single long time series, typically through a sliding window or other segmentation techniques. The goal is to cluster these segments, and this is the primary focus of our paper.

Time-point clustering (Mörchen et al., 2005; Ertl et al., 2021), the third category, involves clustering individual time points based on their temporal proximity and the similarity of their corresponding values. This method is similar to time series segmentation but allows for some points to remain unclustered, treating them as noise.

Several methods have been proposed for subsequence clustering. For example, the MDL framework (Rakthanmanon et al., 2012) is a highly efficient approach for time series clustering and can be applied to data streams. Building on this foundational work, Li et al. (Li et al., 2012) developed a methodology to discover approximate time series motifs of varying lengths using a compression algorithm.

Some researchers have proposed selective clustering of subsequences, emphasizing that subsequence clustering is more meaningful when noisy or irrelevant subsequences are ignored and when subsequences of varying lengths are considered (Rodpongpun et al., 2012). Another approach by Yang (Yang and Wang, 2014) introduced the phase shift weighted spherical K-means algorithm (PS-WSKM) for clustering unsynchronized time series.

Rakthanmanon et al. (Rakthanmanon et al., 2013) demonstrated that time series clustering using Dynamic Time Warping (DTW) is significantly faster than methods relying on Euclidean distance.

However, one of the major challenges in subsequence time series clustering is handling large datasets with numerous segments. Most previous research has focused on relatively small datasets, but as the number of segments grows, traditional algorithms become inefficient and slow. Our goal in this paper is to accelerate these processes, making them suitable for large scale industrial applications.

3 THEORETICAL OVERVIEW

In this section, we explore the key concepts, algorithms, and methodologies that formed the foundation of our research. A crucial aspect of analyzing time series data is comparing segments to determine their degree of similarity; this requires the application of statistical tests.

3.1 Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov (K-S) (Kolmogorov, 1933) test and the Anderson-Darling (Anderson and Darling, 1952) test are commonly used to compare the distributions of two samples to determine if they come from the same distribution. The K-S test calculates the maximum difference between their empirical cumulative distribution functions (ECDFs):

$$F_X(x) = \frac{1}{n_1}\sum_{i=1}^{n_1} \mathbf{1}_{[x_i \le x]}, \quad F_Y(y) = \frac{1}{n_2}\sum_{i=1}^{n_2} \mathbf{1}_{[y_i \le y]} \quad (1)$$

The K-S statistic D is the maximum absolute difference between the ECDFs:

$$D = sup\,|F_X(x) - F_Y(x)| \quad (2)$$

The p-value quantifies the probability of observing a test statistic as extreme as the observed D, or more extreme, under the assumption that the null hypothesis is true. The complexity of the K-S test is $O(n \log n)$, where $n = n_1 + n_2$.

The K-S test is distribution-free, as its critical values do not depend on the specific distribution. This makes it computationally simpler and versatile. It excels at detecting differences near the center of distributions, which aligns with the focus of this study. In contrast, the Anderson-Darling test is more sensitive to differences in the tails of distributions but relies on the specific distribution for critical values, limiting its generalizability when such information is unavailable. Given the need for a robust and adaptable test, the K-S test is better suited to this application.

3.2 COP-KMeans

COP-KMeans (Wagstaff et al., 2001) extends K-means by incorporating pairwise constraints:

- Must-link: Points must be in the same cluster.
- Cannot-link: Points cannot be in the same cluster.

During assignment, points are assigned to the nearest centroid while respecting these constraints. The complexity is similar to K-means, $O(t \cdot n \cdot k)$, where t is the number of iterations until defined convergence criteria, n is the number of samples and k is the number of desired clusters.

3.3 Key Metrics for Clustering Evaluation

Several metrics are used to evaluate clustering performance, especially when multiple labels are present. Key metrics include the Rand Index (RI) (Rand, 1971), Adjusted Rand Index (ARI) (Hubert and Arabie, 1985), and Normalized Mutual Information (NMI) (Vinh et al., 2009).

Rand Index (RI) measures the proportion of correctly classified pairs of samples. It ranges from 0 to 1, where 1 indicates perfect agreement.

$$RI = \frac{a+d}{a+b+c+d} \tag{3}$$

where a and b are the number of pairs correctly and incorrectly clustered together; c and d are the number of pairs incorrectly and correctly clustered apart, respectively.

Adjusted Rand Index (ARI) adjusts the RI by accounting for chance. It ranges from -1 to 1, with 1 indicating perfect agreement.

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} \tag{4}$$

where $E[RI]$ is the expected RI under random clustering.

Normalized Mutual Information (NMI) measures the mutual information between two clusterings, normalized to account for different cluster sizes. It ranges from 0 to 1, where 1 indicates perfect correlation.

$$NMI(U,V) = \frac{2 \cdot I(U;V)}{H(U)+H(V)} \tag{5}$$

where $I(U;V)$ is the mutual information, and $H(U)$ and $H(V)$ are the entropies of the clusterings U and V.

These metrics provide insights into clustering performance, with ARI and NMI useful for comparing results against ground truth or when cluster sizes vary.

4 PROPOSED CLUSTERING METHOD WITH FULL EMBEDDING

In this section, we present a novel clustering method designed for time-varying segments characterized by inhomogeneous lengths and temporal misalignment. The proposed approach integrates a feature extraction process using the Kolmogorov-Smirnov (KS) test to quantify statistical differences between segment distributions, followed by embedding construction and clustering using the Constrained K-means (COP-KMeans) algorithm. Our approach ensures that consecutive segments are assigned to distinct clusters, respecting their temporal dependencies.

4.1 Problem Formulation

Clustering time-varying segments introduces significant challenges due to two key factors:

- Temporal Misalignment: Segments may exhibit significant shifts along the time axis, making direct comparison difficult.
- Inhomogeneous Lengths: The segments may have varying lengths, further complicating distance-based clustering methods.

To address these challenges, our proposed method focuses on extracting features that capture distributional differences, independent of temporal alignment or length variation.

The time-varying segments $\{s_1, s_2, \ldots, s_n\}$ are derived by dividing the continuous time series into non-overlapping, variable-length segments. The segmentation is guided by domain-specific events or changes in the signal, such as sharp transitions or activity boundaries, ensuring that each segment captures a meaningful portion of the underlying dynamics (Guijo-Rubio et al., 2021).

Given these segments, where each segment s_i has an associated length l_i, our goal is to cluster them into K distinct clusters. This is achieved by constructing meaningful embeddings for each segment based on distributional similarity, allowing robust clustering even in the presence of temporal misalignment or length variation.

4.2 Embedding Construction Using Kolmogorov-Smirnov Test Statistic

To generate appropriate embeddings, we begin by quantifying the statistical differences between each pair of segments. For this, we employ the Kolmogorov-Smirnov (KS) test, a non-parametric test that compares two sample distributions. The KS test statistic for two segments s_a and s_b, with lengths l_a and l_b, is defined as:

$$D_{KS}(s_a, s_b) = sup|F_a(t) - F_b(t)| \tag{6}$$

where $F_a(t)$ and $F_b(t)$ represent the empirical cumulative distribution functions (ECDFs) of segments s_a and s_b, respectively. The KS statistic $D_{KS}(s_a, s_b)$ measures the maximum difference between the two ECDFs, offering a robust means of comparing the statistical distributions of the segments, irrespective of their length or alignment.

Once the KS test statistic has been computed for all pairs of segments, the corresponding p-values are derived to evaluate the null hypothesis that the compared segments are drawn from the same distribution. These p-values are then used to construct an embedding vector x_i for each segment s_i. Each component of x_i captures the likelihood that the KS test statistic $D_{KS}(s_j, s_i)$ falls below a predefined threshold d. Formally, the embedding vector for segment s_i is expressed as:

$$x_i = \Big[P(D_{KS}(s_1, s_i) < d), P(D_{KS}(s_2, s_i) < d), \ldots, P(D_{KS}(s_n, s_i) < d)\Big] \tag{7}$$

These embeddings encapsulate the statistical relationship of segment s_i with all other segments, effectively reducing the original time series data into a feature space based on pairwise distributional differences. The dimensionality of each embedding is equal to n, the total number of segments.

4.3 Clustering with Adapted COP-KMeans

While standard clustering algorithms like K-means could be applied to these embeddings, they do not take into account the temporal structure of the data, where consecutive segments should belong to distinct clusters. To address this, we use the Constrained K-means (COP-KMeans) algorithm (Wagstaff et al., 2001), which incorporates pairwise constraints to guide the clustering process.

4.3.1 COP-KMeans Algorithm

COP-KMeans operates similarly to the traditional K-means algorithm but incorporates two types of constraints:

- Must-Link Constraint: Enforces that certain pairs of data points must belong to the same cluster.
- Cannot-Link Constraint: Ensures that certain pairs of data points must belong to different clusters.

In our context, the cannot-link constraint is applied to consecutive segments to prevent them from being grouped into the same cluster. This constraint reflects the temporal dependency between consecutive segments, ensuring that segments that are adjacent in time are not grouped together, as they are likely to represent distinct temporal phenomena.

4.3.2 Temporal Structure and Constraints

The use of cannot-link constraints based on temporal adjacency ensures that consecutive segments, which are known to belong to different clusters, are assigned to distinct groups.

Let s_t and s_{t+1} represent two consecutive segments. The cannot-link constraint ensures:

$$\text{Cluster}(s_t) \neq \text{Cluster}(s_{t+1}) \tag{8}$$

This constraint is enforced during the assignment step of the COP-KMeans algorithm, ensuring that the temporal structure of the data is preserved in the clustering process. In our context, the cannot-link constraint is applied to consecutive segments to prevent them from being grouped into the same cluster. This constraint reflects the temporal dependency between consecutive segments, ensuring that segments adjacent in time are not grouped together, as they likely represent distinct temporal phenomena.

This assumption is justified by prior knowledge of changepoints between segments, which we have identified using proxy methods. These changepoints could have been detected by any suitable time series segmentation algorithm, such as ClaSP (Ermshaus et al., 2023), and are considered available knowledge in our research. Therefore, the subsegments are divided by changepoints, and consecutive segments cannot be in the same cluster, reinforcing the necessity of the cannot-link constraint.

4.3.3 Algorithm Details

Let $x_i \in \mathbb{R}^n$ represent the embedding vector for segment i, constructed as described above. The steps for COP-KMeans clustering are as follows:

Step 1: Initialization. Randomly initialize K centroids $\{c_1, c_2, \ldots, c_K\}$ from the embeddings $\{x_1, x_2, \ldots, x_n\}$.

Step 2: Assignment. Assign each embedding x_i to the nearest centroid based on Euclidean distance, while respecting the cannot-link constraints. For each embedding x_i, the assignment step solves:

$$\arg\min_k \|x_i - c_k\|^2 \quad \text{subject to constraints} \tag{9}$$

The cannot-link constraint ensures that the segments are assigned to different clusters.

Step 3: Centroid Update. After assigning all segments, update the centroids as the mean of the embeddings assigned to each cluster. For cluster k, the updated centroid c_k is computed as:

$$c_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i \tag{10}$$

where C_k represents the set of embeddings assigned to cluster k, and $|C_k|$ is the number of embeddings in that cluster.

Step 4: Iteration. Repeat Steps 2 and 3 until convergence, defined as no further changes in cluster assignments or centroid locations.

4.3.4 Final Clustering Output

The final output of the clustering process is a set of cluster labels $\{l_1, l_2, \ldots, l_n\}$, where l_i denotes the cluster to which segment x_i is assigned. The clusters reflect both the statistical differences between segments, as captured by the KS test statistic, and the temporal dependencies between consecutive segments, as enforced by the cannot-link constraints. In summary, the proposed method offers a non-sensitive approach to clustering time-varying segments with inhomogeneous lengths by combining statistical feature extraction with constrained clustering.

5 REDUCED-EMBEDDING-BASED CLUSTERING METHOD WITH BASEPOINTS

In this section, we present an updated approach to address the computational challenges associated with the construction of embedding vectors. Specifically, we tackle the quadratic growth in dimensionality of the embedding vectors when performing exhaustive pairwise comparisons between segments. By utilizing a predefined set of b basepoints, we reduce the dimensionality of the embedding vectors and improve computational efficiency without significantly sacrificing accuracy in clustering.

5.1 Motivation for Basepoint Selection

The embedding vectors generated by performing a Kolmogorov-Smirnov (KS) test between all pairs of segments grow quadratically with the number of segments. This results in both an excessive computational burden and the inclusion of redundant information. To overcome this, we propose selecting a smaller set of representative basepoints, b, which is significantly smaller than the total number of segments n. By limiting comparisons to these b basepoints, the dimensionality of the embeddings is reduced. This reduction decreases the complexity of the algorithm from $O(n^2)$ to $O(n \cdot b)$ while retaining the essential characteristics of the dataset.

The motivation behind the basepoint selection is to maintain the representativeness of the clustering space by focusing on a set of diverse basepoints. This ensures that the essential characteristics of the dataset are captured without performing exhaustive pairwise comparisons.

5.2 Proposed Basepoint Selection Strategy

The idea of the basepoint selection process comes from k-means++ (Arthur and Vassilvitskii, 2007). The basepoint selection process begins by randomly selecting one segment to serve as the first basepoint. The embedding vector's initial value is then calculated for each segment based on this basepoint. Subsequent basepoints are selected iteratively by maximizing the Euclidean distance from the previously selected basepoints. This ensures that each new basepoint adds diversity and enhances the representation of the data.

The steps of the basepoint selection process are as follows:

1. Initialization: Select the first basepoint $B[g]$ randomly from the set of segments $X = \{s_1, s_2, \ldots, s_n\}$, where g=1.
2. Intermediate Embedding Vector: For each segment $s_i \in X$, compute the embedding vector's first value, $v_i[g] = f(s_{B[g]}, s_i)$, where $f(s_a, s_b)$ represents the distance metric and g=1.

3. Iterative Selection: For each subsequent basepoint $B[j]$ (j=2,3,...,b), calculate the Euclidean distance between the current basepoints and all remaining segments. Select the segment that maximizes the minimum distance from the current basepoints:
$$B[j] = \arg\max_{x_i} \min_{x_{B[1]},\dots,x_{B[j-1]}} \|x_i - x_{B[k]}\| \quad (11)$$
This iterative selection continues until b basepoints are chosen.
4. Embedding Vector Construction: For each segment $s_i \in X$, compute the embedding vector based on the selected basepoints:
$$x_i = \left[f(s_{B[1]}, s_i), f(s_{B[2]}, s_i), \dots, f(s_{B[b]}, s_i)\right] \quad (12)$$

By strategically selecting basepoints in this manner, we ensure that each new basepoint captures a unique aspect of the dataset, enhancing the goodness of the embeddings while significantly reducing computational complexity.

5.3 Complexity of the Method

The overall time complexity of the proposed method can be analyzed as follows:

- KS Test Cost: The computational cost of performing the Kolmogorov-Smirnov test between two segments of combined length L is $O(L\log L)$. Since this test is performed between the basepoints and all segments, the total cost for this step is $O(b \cdot n \cdot L\log L)$, where b is the number of basepoints and n is the number of segments.
- Euclidean Distance Calculation: The cost of calculating the Euclidean distance for each basepoint is $O(b)$, where b is the number of dimensions in the embedding vector.
- Embedding Vector Construction: The cost of constructing the embedding vector is:
$$O(b \cdot (n \cdot L\log L + b \cdot n \cdot b)) = O(b \cdot n \cdot L\log L + b^3 \cdot n) \quad (13)$$
This accounts for the KS test comparisons and the embedding vector construction based on basepoints.
- Clustering Step: The complexity of the COP-KMeans clustering algorithm is:
$$O(n \cdot b \cdot k \cdot I) \quad (14)$$
where k is the number of clusters, and I is the number of iterations required for convergence.

Thus, the total complexity of the method is:
$$O(b \cdot n \cdot L\log L + b^3 \cdot n + n \cdot b \cdot k \cdot I) \quad (15)$$
This formulation highlights the computational efficiency gained by reducing the dimensionality of the embeddings through basepoint selection.

5.4 Confidence Score Calculation

To further enhance the interpretability of the clustering results, we introduce a confidence score for each segment's assigned label. The confidence score is based on the relative distances between the segment and its two closest cluster centers.

The confidence score $C(x_i)$ for segment x_i is defined as follows:
$$C(x_i) = \begin{cases} 1 & \text{if } d_2(x_i) > 2 \cdot d_1(x_i) \\ \frac{d_2(x_i) - d_1(x_i)}{d_1(x_i)} & \text{otherwise} \end{cases} \quad (16)$$
where $d_1(x_i)$ is the distance between x_i and the closest cluster center, and $d_2(x_i)$ is the distance to the second closest cluster center. The confidence score ranges between 0 and 1:

- $C(x_i) = 1$ if the second closest cluster center is very far from the closest one, indicating high confidence.
- $C(x_i) = 0$ if the second closest center is as close as the closest, indicating low confidence.

This method provides an effective way to quantify the reliability of clustering labels, offering insights into ambiguous or uncertain assignments. This is similar to Silhouette Index, but in this case we quantify nearest centroid reliability, while the Silhouette Index measures intra-cluster density. We provide the pseudo-code implementation of the proposed method in the Algorithm 1.

6 EXPERIMENTAL RESULTS

6.1 Dataset Description

For experimental evaluation, we selected ten diverse datasets from the Time Series Segmentation Benchmark (TSSB) (Ermshaus et al., 2023). These datasets vary in domain, complexity, and length, providing a basis for testing. The datasets include:

- Plane (PLN): Time series for plane shapes.
- NonInvasiveFetalECGThorax1 (NIFECG): Fetal ECG signals with physiological noise.
- UWaveGestureLibraryX/All (UWGLX/All): Accelerometer data capturing hand gestures.
- EOGHorizontalSignal (EOGHS): Noisy eye movement data from electrooculography (EOG).
- ProximalPhalanxTW (PPTW): Hand bone motion.
- SwedishLeaf (SLF): Leaf outlines for shape.

Algorithm 1: Basepoint Selection and Clustering with Adapted COP-KMeans.

```
Data: Segments $X = \{s_1, s_2, \ldots, s_n\}$, Number
      of basepoints $b$, Number of clusters $k$,
      Distance function $f(s_a, s_b)$ (i.e.
      KS-test)
Result: Cluster labels $L = \{l_1, l_2, \ldots, l_n\}$
Initialize empty array $B$ to store basepoints;
Select the first basepoint as the first element:
  $B[1] = 1$;
for each segment $x_i$ in $X$ do
    Calculate the embedding vector's first
      value: $v_i[1] = f(s_{B[1]}, s_i)$;
end
for basepoint index $j = 2$ to $b$ do
    Find the next basepoint;
    for each segment $s_i$ in $X$ (including
      already selected basepoints) do
        Calculate Euclidean distance from the
          vectors of the current basepoints
          $\{x_{B[1]}, \ldots, x_{B[j-1]}\}$;
    end
    Select the segment with the maximum
      distance as the new basepoint:
      $B[j] = \arg\max d(x_{B[j-1]}, x_i)$;
end
for each segment $s_i$ in $X$ do
    Update the embedding vector:
      $v_i[j] = f(s_{B[j]}, s_i)$;
end
Perform adapted COP-KMeans clustering on
  the embedding vectors $V = \{v_1, v_2, \ldots, v_n\}$;
Use constraints to ensure consecutive
  segments are assigned to different clusters;
return Cluster labels $L = \{l_1, l_2, \ldots, l_n\}$
```

- Symbols (SYM): Symbolic shape representations.
- Car (CAR): Car data from driving behaviors.
- InlineSkate (INSK): Data from inline skating.

We generated larger datasets of 1,000 and 10,000 segments by randomly extracting 30–80% portions of the original time series, introducing segment length variability. Clustering performance was evaluated using RI, ARI, and NMI, averaged over 10 runs. Additionally, we assessed the impact of confidence-based label removal on clustering quality by recalculating metrics after discarding low-confidence labels.

6.2 Results of the Clustering Performance

This section presents the results for clustering using the ground truth number of clusters. The adapted COP-KMeans algorithm was run more times, and the average scores were calculated.

6.2.1 Clustering Performance with Ground Truth Number of Clusters

The Table 1 summarizes the results for each dataset, consisting of 1,000 segments, using the ground truth number of clusters. The results presented are the averages across ten runs of the adapted COP-KMeans algorithm with fifty basepoints.

Table 1: Adapted COP-KMeans with 50 Basepoints.

Dataset	RI (mean/std)	ARI (mean/std)	NMI (mean/std)
PLN	0.987/0.026	0.950/0.100	0.970/0.060
NIFECG	0.947/0.044	0.818/0.148	0.872/0.094
UWGLX	0.901/0.042	0.704/0.119	0.788/0.077
UWGLAll	0.983/0.001	0.948/0.003	0.941/0.003
EOGHS	0.680/0.020	0.202/0.047	0.259/0.028
PPTW	0.994/0.000	0.984/0.000	0.976/0.000
SLF	1.000/0.000	1.000/0.000	1.000/0.000
SYM	0.920/0.002	0.786/0.005	0.736/0.005
CAR	0.977/0.000	0.940/0.000	0.925/0.000
INSK	0.864/0.000	0.695/0.000	0.654/0.000
Average	0.925/0.014	0.803/0.042	0.812/0.027

After evaluating adapted COP-KMeans with basepoints, we perform a similar experiment without the use of basepoints. This configuration provides insight into how the method performs without basepoints.

In Table 2, the results show the performance when all segments are compared to each other, removing the basepoint constraint. This configuration is expected to yield higher accuracy because of the comparison of each segment, but this comes at the cost of significantly increased computation time. The last three columns present that the goodness indicators decrease only slightly with the basepoints, dropping to only 97%, 89% and 91% compared to the reference.

Table 2: Clustering performance comparison of adapted COP-KMeans without basepoints and the ratio of performance with and without basepoints.

Dataset	No Basepoints (BP)			With and w/o BP ratio (%)		
	RI	ARI	NMI	RI	ARI	NMI
PLN	0.987	0.952	0.971	100	99.8	99.9
NIFECG	0.981	0.932	0.945	96.5	87.8	92.3
UWGLX	0.974	0.925	0.931	92.5	76.1	84.6
UWGLAll	0.985	0.952	0.942	99.9	99.5	99.5
EOGHS	0.761	0.381	0.432	89.4	53	60
PPTW	0.996	0.989	0.984	99.5	99.9	99.2
SLF	1	1	1	100	100	100
SYM	0.932	0.818	0.771	98.7	96.1	95.5
CAR	0.966	0.909	0.887	101.1	103.4	104.3
INSK	0.956	0.9	0.859	94.0	77.2	76.1
Average	0.954	0.876	0.872	96.8	89.3	91.2

6.2.2 Effectiveness of Confidence Score in Enhancing Clustering Performance

In this section, we analyze the distribution of confidence scores within the ”Symbols” dataset to understand their relationship with clustering accuracy. Figure 1 shows how filtering segments based on confidence scores affects clustering metrics (RI, ARI, and NMI, and the horizontal axis shows the threshold of the confidence score in the filtering). As low-confidence segments are removed, the metrics improve significantly, indicating that segments with lower confidence are more likely to be mislabeled. This supports the hypothesis that higher confidence scores correlate with better clustering accuracy, and filtering based on confidence can enhance clustering performance.

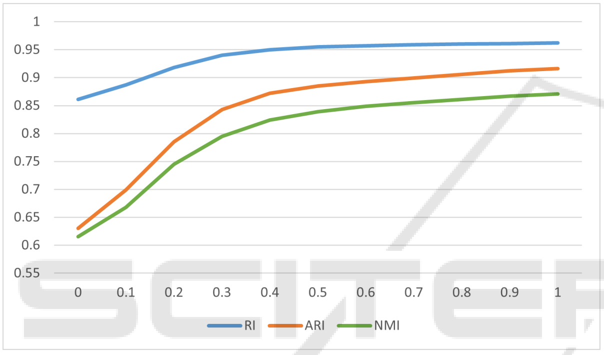

Figure 1: Impact of confidence score filtering on clustering performance metrics for the ”Symbols” dataset.

6.3 Runtime Analysis

The performance of clustering algorithms is critical not only in terms of accuracy but also computational efficiency, especially with large datasets. In this section, the runtime of the adapted COP-KMeans algorithm is evaluated under varying conditions, comparing its performance with and without basepoints. These comparisons help assess the algorithm’s scalability and applicability to large scale datasets.

Table 3 compares runtimes for two configurations: with 50 basepoints and without basepoints, all tested on 1,000 segments. This reveals the computational trade-offs across different setups.

Table 4 presents runtimes with 1,000 segments and varying basepoints (from 25 to 1,000), illustrating how runtime increases with more basepoints.

Table 5 presents runtime results with a fixed number of basepoints (50) while varying the number of segments (100, 500, 1,000, 5,000, 10,000), demonstrating how runtime scales with dataset size.

In addition, Figure 2 and 3 illustrates the average runtime of the adapted COP-KMeans algorithm as the number of segments (n) and basepoints (b) vary. The algorithm’s complexity is $O(b \cdot n \cdot L \cdot \log L + b^3 \cdot n + n \cdot b \cdot k \cdot I)$, indicating linear scaling with respect to n and power function behavior with respect to b. The plots confirm these trends, aligning with theoretical expectations for runtime behavior.

Table 3: Runtime comparison of adapted COP-KMeans configurations with 1,000 segments.

Dataset	N $\times$ B - COP (mean $\pm$ std)	N $\times$ N - COP (mean $\pm$ std)
PLN	22.8 ± 0.6	375.4 ± 12.3
NIFECG	24.1 ± 0.4	417.3 ± 11.4
UWGLX	23.1 ± 0.7	370.4 ± 11.6
UWGLAll	23.4 ± 0.5	403.0 ± 10.0
EOGHS	23.2 ± 0.5	387.6 ± 12.2
PPTW	22.3 ± 0.3	382.9 ± 7.5
SLF	22.1 ± 0.7	374.9 ± 6.4
SYM	23.3 ± 0.7	400.8 ± 6.4
CAR	21.3 ± 0.6	363.9 ± 7.3
INSK	22.9 ± 0.4	404.0 ± 7.6
Average	22.8 ± 0.5	388.0 ± 9.3

Table 4: Runtime results with varying numbers of basepoints at a fixed dataset size of 1,000 segments.

Dataset \ B	25	50	100	200	500
PLN	11.1 ± 0.6	23.7 ± 0.4	54.5 ± 0.5	142.7 ± 0.6	615.2 ± 0.7
NIFECG	11.6 ± 0.4	24.7 ± 0.3	56.9 ± 0.4	150.6 ± 0.4	659.4 ± 0.5
UWGLX	11.4 ± 0.7	23.9 ± 0.4	55.1 ± 0.4	142.6 ± 0.5	617.0 ± 0.6
UWGLAll	11.6 ± 0.5	23.8 ± 0.5	54.9 ± 0.4	146.6 ± 0.5	642.1 ± 0.5
EOGHS	11.3 ± 0.5	23.8 ± 0.4	54.1 ± 0.4	143.4 ± 0.5	630.5 ± 0.4
PPTW	11.4 ± 0.3	22.7 ± 0.4	51.4 ± 0.4	140.6 ± 0.5	617.8 ± 0.5
SLF	10.6 ± 0.7	23.0 ± 0.5	52.4 ± 0.4	140.9 ± 0.5	623.0 ± 0.5
SYM	11.8 ± 0.7	24.2 ± 0.3	54.0 ± 0.4	144.8 ± 0.5	622.3 ± 0.6
CAR	10.6 ± 0.6	22.1 ± 0.2	51.5 ± 0.5	138.0 ± 0.4	620.4 ± 0.4
INSK	11.1 ± 0.4	23.5 ± 0.4	54.9 ± 0.4	145.2 ± 0.4	636.6 ± 0.5
Average	11.3 ± 0.5	23.5 ± 0.4	54.0 ± 0.4	143.5 ± 0.5	628.4 ± 0.5

Table 5: Runtime results with a fixed number of basepoints (50) and varying numbers of segments.

Dataset \ N	100	500	1000	5000	10000
PLN	6.4 ± 0.6	14.6 ± 0.4	23.7 ± 0.4	96.5 ± 0.5	197.3 ± 0.6
NIFECG	6.3 ± 0.4	15.6 ± 0.3	24.7 ± 0.3	106.7 ± 0.4	199.8 ± 0.5
UWGLX	6.3 ± 0.7	14.4 ± 0.4	23.9 ± 0.4	96.0 ± 0.5	190.8 ± 0.6
UWGLAll	6.4 ± 0.5	14.9 ± 0.5	23.8 ± 0.5	101.6 ± 0.5	190.6 ± 0.5
EOGHS	6.4 ± 0.5	14.7 ± 0.5	23.8 ± 0.4	100.2 ± 0.5	192.4 ± 0.5
PPTW	6.3 ± 0.3	14.2 ± 0.4	22.7 ± 0.4	94.7 ± 0.5	178.2 ± 0.5
SLF	6.4 ± 0.7	14.4 ± 0.5	23.0 ± 0.5	96.1 ± 0.5	180.6 ± 0.5
SYM	6.4 ± 0.7	14.7 ± 0.3	24.2 ± 0.4	100.5 ± 0.5	184.8 ± 0.5
CAR	6.4 ± 0.6	13.9 ± 0.2	22.1 ± 0.5	92.5 ± 0.4	178.5 ± 0.4
INSK	6.4 ± 0.4	15.2 ± 0.4	23.5 ± 0.4	97.4 ± 0.4	188.5 ± 0.5
Average	6.4 ± 0.5	14.5 ± 0.4	23.4 ± 0.4	98.7 ± 0.5	188.5 ± 0.5

6.4 Sensitivity Analysis with Noise

In real-world scenarios, time series data often contains noise from environmental factors or measurement errors. Evaluating a clustering method under noisy conditions is essential to assess its stability and reliability. By introducing noise and analyzing the impact on clustering performance, we can identify both the strengths and limitations of the method, providing a basis for potential improvements.

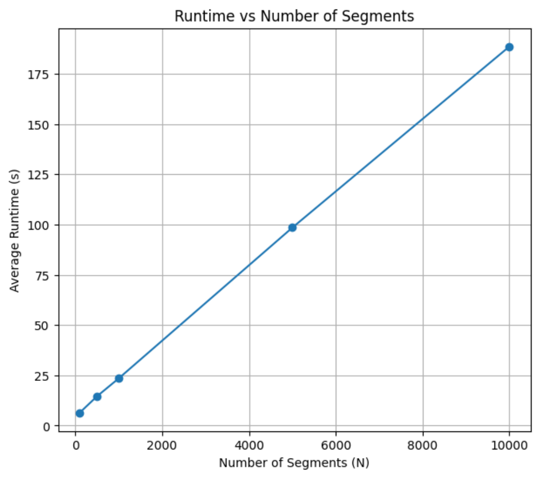

Figure 2: Average runtime of the adapted COP-KMeans algorithm based on varying segments (n).

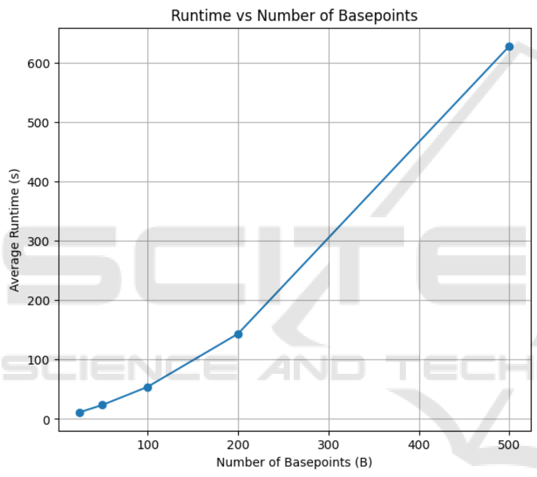

Figure 3: Average runtime of the adapted COP-KMeans algorithm based on varying basepoints (b).

6.4.1 Noise Injection Strategy

To measure the sensitivity of the clustering method, Gaussian noise is systematically injected into segments, simulating real-world sensor data imperfections. Noise is controlled by two parameters:

- Error Percentage: Defines the proportion of data points affected by noise (e.g., 10% means 10% of the segment is modified).
- Error Standard Deviation: Determines the magnitude of Gaussian noise, which has a mean of 0 and a standard deviation ranging from 0 to 1, relative to the segment's data range.

Noise is incrementally applied, with both parameters varying from 0% to 100% in 10% steps, allowing performance analysis under different noise levels.

6.4.2 Results

We evaluated the impact of noise on clustering performance using 11x11 ARI heatmaps, where the x-axis represents the error standard deviation and the y-axis indicates the sample percentage affected by noise. The adapted COP-KMeans was run 10 times for each configuration, with results averaged for accuracy. The results fall into three main categories:

1. Low Performance Without Noise: In some datasets, the clustering method struggles even without noise, indicated by low ARI values across all noise levels. Noise slightly worsens results, but the method already fails to cluster effectively (see Figure 4).

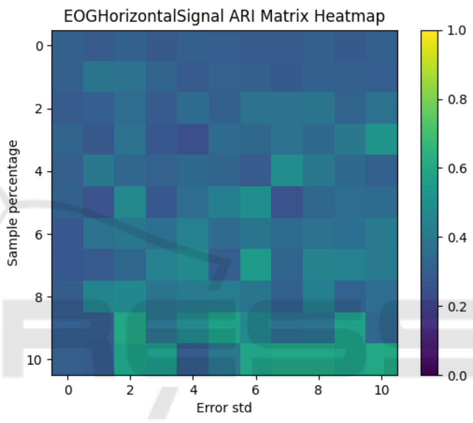

Figure 4: EOGHorizontalSignal dataset's ARI heatmap.

2. High Sensitivity to Noise: Some datasets show high clustering accuracy without noise, but performance rapidly degrades as noise increases. ARI values drop to near zero, highlighting the method's vulnerability to noise (see Figure 5).

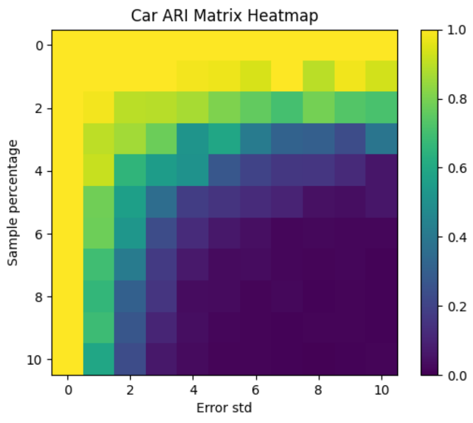

Figure 5: Car dataset's ARI heatmap.

3. Moderate Sensitivity to Noise: Other datasets maintain moderate accuracy as noise is introduced. While ARI values decline, the method still produces reasonable clusters, demonstrating some resilience to noise (see Figure 6).

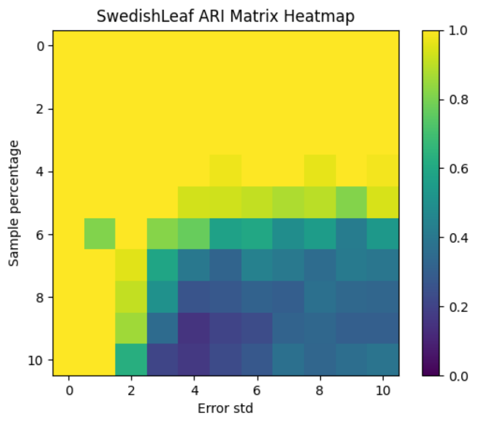

Figure 6: SwedishLeaf dataset's ARI heatmap.

This analysis underscores the method's varying sensitivity to noise, suggesting areas for improvement in noisy environments.

6.5 Confidence Score Analysis

In addition to evaluating ARI values, we analyzed the confidence scores assigned to each segment. As noise increases, the distribution of confidence scores shifts toward zero, indicating decreased reliability in clustering.

6.5.1 Confidence Score Distribution with Increasing Noise

We examined the Car dataset with three noise levels:

- Low noise: At 20% noise, confidence scores remain high for most segments, indicating reliable clustering (see Figure 7).
- Moderate noise: At 40% noise, the distribution shifts, with more segments receiving lower confidence scores, suggesting difficulties in differentiating segments (see Figure 8).
- High noise: At 100% noise, confidence scores drop significantly, with most values near zero, highlighting the method's struggle under high noise conditions (see Figure 9).

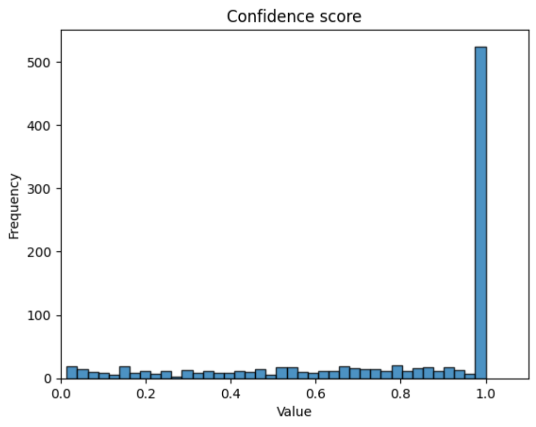

Figure 7: Car dataset's confidence distribution with 20% noise.

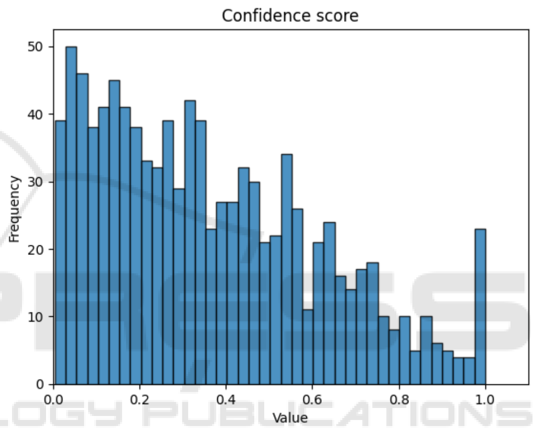

Figure 8: Car dataset's confidence distribution with 40% noise.

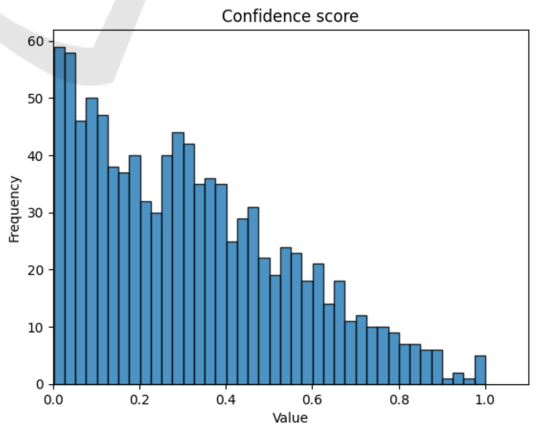

Figure 9: Car dataset's confidence distribution with 100% noise.

6.5.2 Impact of Confidence Score Threshold on Clustering Metrics

With 40% noise, increasing the confidence score threshold from 0 to 1 improves clustering metrics (RI, ARI, and NMI), but at the cost of discarding many segments. This presents a trade-off between maintaining higher confidence and retaining more data, as shown in Figure 10.

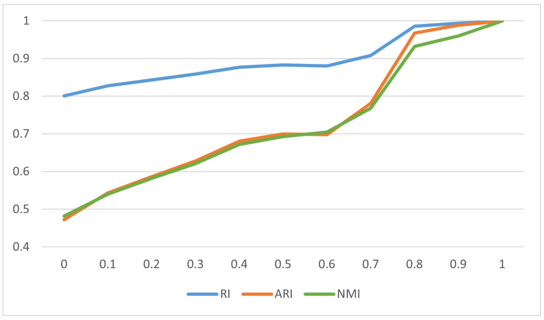

Figure 10: Impact of confidence score filtering on clustering performance metrics for the "Car" dataset with 40% noise.

At 100% noise, this trend reverses; metrics do not improve as confidence thresholds rise due to significant data loss. This illustrates that while filtering by confidence enhances performance with moderate noise, it becomes less effective in high-noise scenarios, as demonstrated in Figure 11.

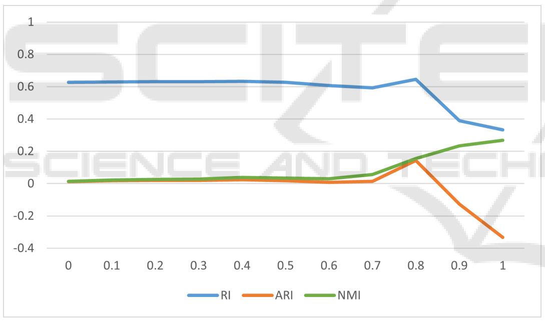

Figure 11: Impact of confidence score filtering on clustering performance metrics for the "Car" dataset with 100% noise.

7 CONCLUSION

In this paper, we presented a novel approach for clustering time-varying segments by integrating statistical methods with constrained clustering techniques. The primary contributions of this work include the development of a hybrid clustering framework that combines the Kolmogorov–Smirnov (KS) test and adapted COP-KMeans for clustering with temporal constraints. This approach addresses several key challenges in time series clustering, such as segment misalignment, varying lengths.

The core innovation lies in using the KS test statistic to generate distribution-based embeddings that capture statistical differences between segments. These embeddings provide a stable representation of the data, independent of temporal misalignment and length variation. Adapted COP-KMeans is then applied with cannot-link constraints, ensuring that consecutive segments are assigned to different clusters, preserving their temporal structure.

To improve computational efficiency, we introduced a basepoint selection strategy, reducing the dimensionality of the embedding space from $O(n^2)$ to $O(n \cdot b)$. This effectively decreases the computational complexity of the KS-based embedding construction from $O(n^2 \cdot L \log L)$ to $O(n \cdot b \cdot L \log L)$, where n is the number of segments, b is the number of selected basepoints, and L is the segment length. The experimental results demonstrate that this basepoint reduction significantly enhances runtime performance without a substantial loss in clustering accuracy (with about 10% goodness reduction, a 17-times speed-up could be achieved). This efficiency gain is crucial for large scale industrial datasets, where the traditional exhaustive pairwise comparison would be computationally prohibitive.

In addition to the basepoint strategy, we introduced a confidence score metric that quantifies the reliability of clustering assignments based on the relative distances to the nearest and second-nearest cluster centroids. This score provides a nuanced view of the clustering results, allowing low-confidence labels to be filtered out, thus improving the overall quality of the clustering.

The performance of our method was evaluated on a diverse set of benchmark time series datasets, including motion data and physiological signals. We measured clustering performance using Rand Index (RI), Adjusted Rand Index (ARI), and Normalized Mutual Information (NMI). The results demonstrate that our approach consistently outperforms traditional clustering techniques in terms of computational efficiency. Furthermore, the confidence-based filtering method provided a marked improvement in clustering quality by discarding low-confidence assignments.

While our proposed method shows strong performance across multiple datasets, several avenues for future research remain open. First, the use of alternative statistical tests and distance measures could enhance the sensitivity of the embedding space. The incorporation of more sophisticated alignment techniques, such as shape-based clustering methods or Dynamic Time Warping (DTW) (Müller, 2007), could further improve the accuracy of our approach. Second, while the basepoint selection strategy significantly reduces the dimensionality of the embeddings, optimizing the selection process remains an area for further exploration.

In conclusion, the proposed framework provides a scalable, and interpretable solution to time series clustering, combining statistical rigour with computational efficiency. Our method offers a significant step forward in addressing the challenges of temporal misalignment, variable segment lengths, and large dataset scalability in time series clustering. By balancing the theoretical rigour of statistical tests with the practical demands of large scale data analysis, this work sets the stage for future advancements in time series clustering methodologies.

ACKNOWLEDGEMENTS

Project no. KDP-IKT-2023-900-I1-00000957/0000003 has been implemented with the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the C2299763 funding scheme.

REFERENCES

Aghabozorgi, S., Shirkhorshidi, A. S., and Wah, T. Y. (2015). Time-series clustering–a decade review. *Information systems*, 53:16–38.

Anderson, T. W. and Darling, D. A. (1952). Asymptotic Theory of Certain "Goodness of Fit" Criteria Based on Stochastic Processes. *The Annals of Mathematical Statistics*, 23(2):193 – 212.

Arthur, D. and Vassilvitskii, S. (2007). K-means++: the advantages of careful seeding. In *Proc. of the 18th annual ACM-SIAM symposium on Discrete algorithms*, page 1027–1035. SIAM, Philadelphia, PA, USA.

Caiado, J., Maharaj, E. A., and D'Urso, P. (2015). Time series clustering. In *Handbook of Cluster Analysis*, pages 241–263. CRC Press.

Ermshaus, A., Schäfer, P., and Leser, U. (2023). Clasp: parameter-free time series segmentation. *Data Mining and Knowledge Discovery*, 37(3):1262–1300.

Ertl, B., Meyer, J., Schneider, M., and Streit, A. (2021). Semi-supervised time point clustering for multivariate time series. In *The 34th Canadian Conference on Artificial Intelligence*. Springer.

Fujimaki, R., Hirose, S., and Nakata, T. (2008). Theoretical analysis of subsequence time-series clustering from a frequency-analysis viewpoint. In *International Conference on Data Mining*, pages 506–517. SIAM.

Guijo-Rubio, D., Durán-Rosal, A. M., Gutiérrez, P. A., Troncoso, A., and Hervás-Martínez, C. (2021). Time-series clustering based on the characterization of segment typologies. *IEEE Transactions on Cybernetics*, 51(11):5409–5422.

Hautamaki, V., Nykanen, P., and Franti, P. (2008). Time-series clustering by approximate prototypes. In *19th International Conference on Pattern Recognition*, pages 1–4. IEEE.

Hubert, L. J. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218.

Keogh, E. and Lin, J. (2005). Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and Information Systems*, 8:154–177.

Kolmogorov, A. (1933). Sulla determinazione empirica di una legge di distribuzione. *Giornale dell'Istituto Italiano degli Attuari*, 4:83.

Li, Y., Lin, J., and Oates, T. (2012). Visualizing variable-length time series motifs. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 895–906. SIAM.

Li, Y., Shen, D., Nie, T., and Kou, Y. (2022). A new shape-based clustering algorithm for time series. *Information Sciences*, 609:411–428.

Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern Recognition*, 38(11):1857–1874.

Meesrikamolkul, W., Niennattrakul, V., and Ratanamahatana, C. A. (2012). Shape-based clustering for time series data. In *Advances in KDDM: 16th Pacific-Asia Conference, PAKDD, Kuala Lumpur, Malaysia, May 29-June 1, 2012, Part I*, pages 530–541. Springer.

Mörchen, F., Ultsch, A., and Hoos, O. (2005). Extracting interpretable muscle activation patterns with time series knowledge mining. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 9(3):197–208.

Müller, M. (2007). *Dynamic Time Warping*, pages 69–84. Springer.

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., and Keogh, E. (2013). Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(3):1–31.

Rakthanmanon, T., Keogh, E. J., Lonardi, S., and Evans, S. (2012). Mdl-based time series clustering. *Knowledge and Information Systems*, 33:371–399.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

Rodpongpun, S., Niennattrakul, V., and Ratanamahatana, C. A. (2012). Selective subsequence time series clustering. *Knowledge-Based Systems*, 35:361–368.

Vinh, N. X., Epps, J., and Bailey, J. (2009). Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th ICML*, pages 1073–1080.

Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. (2001). Constrained k-means clustering with background knowledge. In *Proceedings of the 18th ICML*, volume 1, pages 577–584.

Yang, T. and Wang, J. (2014). Clustering unsynchronized time series subsequences with phase shift weighted spherical k-means algorithm. *Journal of Computers*, 9(5):1103–1108.

Zolhavarieh, S., Aghabozorgi, S., and Teh, Y. W. (2014). A review of subsequence time series clustering. *The Scientific World Journal*, 2014(1):312521.