Journal of Artificial Intelligence and Data Mining (JAIDM), x(x): xxx-xxx, xxxx

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

A Novel Combination of Segmentation, Ensemble Clustering and Genetic Algorithm for Clustering Time Series

Ali Ghorbanian[1*] and Zahra Ghorbani[2]

1. Department of Industrial Engineering, Esfarayen University of Technology, Esfarayen, Iran.
2. Edinburgh Business School, Heriot-Watt University, Edinburgh, Scotland (UK).

Article Info

Article History:
Received 12 February 2024
Revised 29 March 2024
Accepted 02 May 2024

DOI:10.22044/jadm.2024.14170.2526

Keywords:
Time-series clustering, Ensemble clustering, Segmentation, Genetic algorithm.

**Corresponding author: a.ghorbanian@esfarayen.ac.ir (A. Ghorbanian).*

Abstract

Increasing the accuracy of time-series clustering while reducing execution time is a primary challenge in the field of time-series clustering. Researchers have recently applied approaches, such as the development of distance metrics and dimensionality reduction, to address this challenge. However, using segmentation and ensemble clustering to solve this issue is a key aspect that has received less attention in previous research. In this study, an algorithm based on the selection and combination of the best segments created from a time-series dataset was developed. In the first step, the dataset was divided into segments of equal lengths. In the second step, each segment is clustered using a hierarchical clustering algorithm. In the third step, a genetic algorithm selects different segments and combines them using combinatorial clustering. The resulting clustering of the selected segments was selected as the final dataset clustering. At this stage, an internal clustering criterion evaluates and sorts the produced solutions. The proposed algorithm was executed on 82 different datasets in 10 repetitions. The results of the algorithm indicated an increase in the clustering efficiency of 3.07%, reaching a value of 67.40. The obtained results were evaluated based on the length of the time series and the type of dataset. In addition, the results were assessed using statistical tests with the six algorithms existing in the literature.

1. Introduction

Time-series clustering is an unsupervised learning process that deals with objects that are not marked or labeled. Clustering of time series can be utilized both directly and indirectly in various industries and services, including electric energy, natural gas consumption, water consumption, or even healthcare. One of the main applications of time-series clustering is identifying consumers' behavioral patterns to manage consumer demand at different times [1]. Other applications of time series clustering are in the preprocessing step for forecasting time series data and for anomaly detection as well as classification [2-9]. As an example, Hatamlou and Deljavan have used this technique to forecast the price of gold in three clusters.

Time-series clustering includes three categories: whole, sequence, and point-based. In whole time series clustering, unlike the other two types, a set of time series is grouped into clusters based on similarity measures, such as having the minimum distance from each other in various groups. In addition, there are two different approaches for clustering of time series. In the first approach, the time series itself is used for clustering. However, in the second approach, features extracted directly or indirectly from a time series will be used for the final clustering [10]. Due to the different nature of time series data in terms of diversity, one of the main approaches in clustering this data type is to

use specialized distance measures, among which Dynamic Time Warping (DTW) and Longest Common Subsequence (LCSS) can be mentioned [11, 12].

This study presents a relatively fast and accurate algorithm for clustering various types of time series, relying on the importance of segmentation and ensemble clustering. The algorithm highlights the significance of segmenting a time series and utilizes an ensemble clustering technique to achieve efficient clustering. This article aims to select appropriate segments using an internal criterion and a metaheuristic algorithm and finally combine these segments using ensemble clustering. Implementing the proposed algorithm on multiple diverse datasets demonstrates high accuracy and relatively low execution time. Moreover, the proposed algorithm exhibits good stability due to the repeated iterations.

This paper is divided into five sections. We present related work in the first section. The second section explains the presented approach and the algorithms employed for segmentation and ensemble clustering. In the third section, the results obtained from implementing the proposed algorithm on multiple datasets are evaluated based on various measures. Results of the presented algorithm, sensitivity, and statistical analysis are presented in section four. Our conclusions are drawn in the final section.

1.1. Literature Review

One of the main approaches in clustering time series is specialized distance measures. Rahim Khan and Zakarya used the LCSS measure for clustering time series data [13]. Also, Soleimani and Abessi increased the clustering accuracy in various datasets by modifying the LCSS measure to a fuzzy version [12]. To cluster long time series data, Kamalzadeh et al. first introduced a distance measure using specific geometric relationships for this type of data. They then utilized this distance measure for clustering time series data [14]. Wang et al. have shown that the difference in the area under two curves can also be a suitable measure for calculating the distance between two time series [15]. In another study, D'Urso et al. applied the fuzzy DTW measure to calculate distances and cluster multivariate time series data [7].

Furthermore, different combinations of distance measures such as DTW, DDTW, and LCSS are usually used [16-18]. Despite the widespread use of these types of distance measures, computational time remains one of the significant challenges. For example, a measure like DDTW can result in an effective computational cost, with execution times reaching up to 80 hours per dataset, which is quite noticeable. Reducing the computational time, several approaches have been considered. One of these approaches is using multi-stage algorithms, where the first phase aims to reduce the dimensionality of the main problem, which causes a reduction in the clustering time in the second phase. Some seminal research in this area includes the studies by Aghabozorgi et al. [19], Zhang et al. [20], and Manakova and Tachenko [21]. Izakian and Mesgari have proposed a technique for clustering time series data using a particle swarm optimization (PSO) approach. The proposed technique was able to find (near) optimal cluster centers during the clustering process [22].

However, Wang et al. have employed a different approach to reduce the computational time. They extract features such as variance, first-order correlation, linearity, curvature, seasonality, peak points, and trough points from a time series. Then, they utilize these features for the final clustering [23]. Zou et al. have taken a different approach, mapping a time series into recurrence and visibility graphs. Then, they utilize the features extracted from these graphs for clustering time series data [24].

Furthermore, Ferreira and Zhao have employed a different approach, mapping a time series into a complex network using various techniques. They then perform the final clustering utilizing the concept of community detection in a complex network [25].

Indeed, performing feature extraction directly can somewhat reduce the computational time of the algorithm. However, it may also lead to a decrease in the final clustering accuracy. On the other hand, using graph-based mapping methods can introduce significant computational overhead.

Another modern approach in this field utilizes autoregressive methods and information theory [1, 26]. Indeed, recent research has shown that the segmentation of a time series can significantly improve the clustering accuracy of a time series dataset. Guijo-Rubio et al. revealed that instead of directly extracting features from a time series, it is possible to transform them into segments using specific algorithms. Then, they utilized the characteristics of these segments and their similarity for the final clustering. This approach has achieved high accuracy, albeit with a relatively higher computational cost [27]. In another study, Bonacina et al. demonstrated that combining segmentation and transforming segments into complex networks can yield better

results [28].

In general, recent studies can be categorized into three main groups: distance metrics, dimensionality reduction, and clustering algorithms. The first category consists of studies in which researchers aim to introduce or improve a specific distance metric for a time series. In some studies, the performance of combining these metrics has been investigated. Although using these specific distance metrics for a time series generally yields good clustering results, it incurs significant time costs, leading to inefficiencies in utilizing such distance metrics.

The second category, known as dimensionality reduction, is approached using various methods. The goal of these studies has often been to reduce execution time and make clustering more practical; however, this has resulted in decreased clustering accuracy. For this purpose, researchers have utilized multistage algorithms, where the primary focus of these algorithms is dimension reduction in the initial phase. Another dimensionality reduction method is feature extraction, which is performed directly or indirectly.

The third category, which is widely observed in the literature, involves the use of diverse clustering algorithms, including the utilization of new algorithms or combining different methods. However, these approaches are often time-consuming and do not always achieve acceptable accuracy for all datasets. Based on the literature review, the main challenge in the field of time-series clustering is to present an algorithm that can simultaneously increase clustering accuracy while maintaining reasonable execution time costs. An efficient clustering algorithm should prioritize not only high accuracy but also reasonable time costs. In research conducted in this field, the focus is often one-dimensional. Some methods emphasize increasing accuracy without considering the execution time, whereas others focus solely on the execution time without considering the accuracy.

One method used to improve the clustering accuracy for various types of data is the utilization of ensemble clustering, which has not received significant attention in previous research. Recent studies have demonstrated that segmentation can yield favorable results in time-series clustering. This paper presents an algorithm based on segmentation and ensemble clustering to enhance clustering accuracy while maintaining reasonable time costs.

Although distance measures for time series clustering might initially appear desirable, as has been widely applied in previous research, the approach suffers from some serious drawbacks, including a significant computational cost. So, there is an urgent need for an accurate and fast time series clustering algorithm in the related literature. The developed algorithm utilizes a combined approach of segmentation and ensemble clustering to enhance accuracy while simultaneously reducing execution time by applying computationally efficient distance metrics, such as Euclidean distance. This allows for an increase in clustering accuracy within a reasonable timeframe

The main principle of this algorithm is to select suitable segments from a dataset and combine them for the final clustering. A combination of a genetic algorithm and an internal clustering criterion is employed to select and combine segments, known as ensemble clustering.

2. Proposed Method

Dividing a time series dataset into equal segments, some segments may effectively represent the existing clusters in the dataset while others may not accurately do this. Figure 1 displays a dataset with two distinct clusters. As evident from the figure, segments 3 and 4 effectively separate the two existing clusters from each other. However, segments 1 and 2 have difficulty in distinguishing the existing clusters.

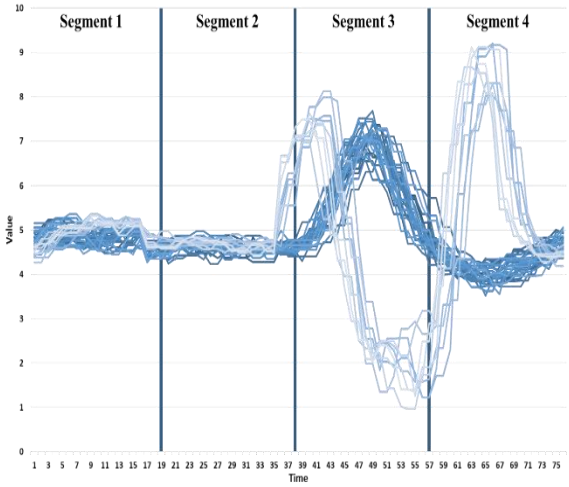


Figure 1. Segmentation of a time series with fixed length.

The objective of the proposed approach is to select suitable segments and combine them for the final clustering. As explained, not all segments in a time-series dataset represent the correct number of clusters. If we can separate the correct segments from incorrect ones during a process, we can utilize the selected correct segments for the final clustering. The aim of the presented

approach is to select suitable segments and combine them for final clustering. The proposed algorithm selects appropriate segments using the concepts of segmentation, ensemble clustering, genetic algorithm, and internal criterion.

To achieve this, the initial dataset was transformed into a fixed number of predetermined segments. Subsequently, each segment is clustered separately using an existing clustering algorithm. The clustering results for each segment are then stored. Essentially, this stage determines the segments that correctly represent the overall dataset clustering. In the next stage, the different segments are combined using ensemble clustering. The final clustering result was obtained from a combination of various segments. Consequently, a better choice of segments leads to a better final clustering result. The goal is to select suitable segments and combine them for the final clustering.

When the number of segments is small, all possible cases can be considered. However, a small number of segments may lead to improper identification of segments with the correct cluster number. As the number of segments increased, the total number of cases became significantly high, making it practically infeasible to examine all combinations. For this purpose, a genetic algorithm was employed to identify the best combination of segments. Owing to the nature of clustering compared with classification, an external criterion cannot be used as the objective function for the genetic algorithm. Therefore, an internal criterion was used to evaluate the solutions generated by the genetic algorithm.

The three-step proposed algorithm utilizes the concepts of segmentation, ensemble clustering, genetic algorithm, and an internal criterion to select suitable segments, as follows:

First Step (Segmentation): In the initial stage of this approach, the dataset is divided into equal segments.

Second Step (Clustering): In this stage, each segment created in the first step is clustered by a clustering algorithm, and the final clustering results for each segment are stored.

Third Step (Segment Selection): In the final stage, a metaheuristic algorithm and ensemble clustering are utilized to select suitable segments, and the final clustering is performed by combining these segments.

Figure 2 depicts the overall framework of the proposed algorithm, which will be described in detail in subsequent sub-sections.

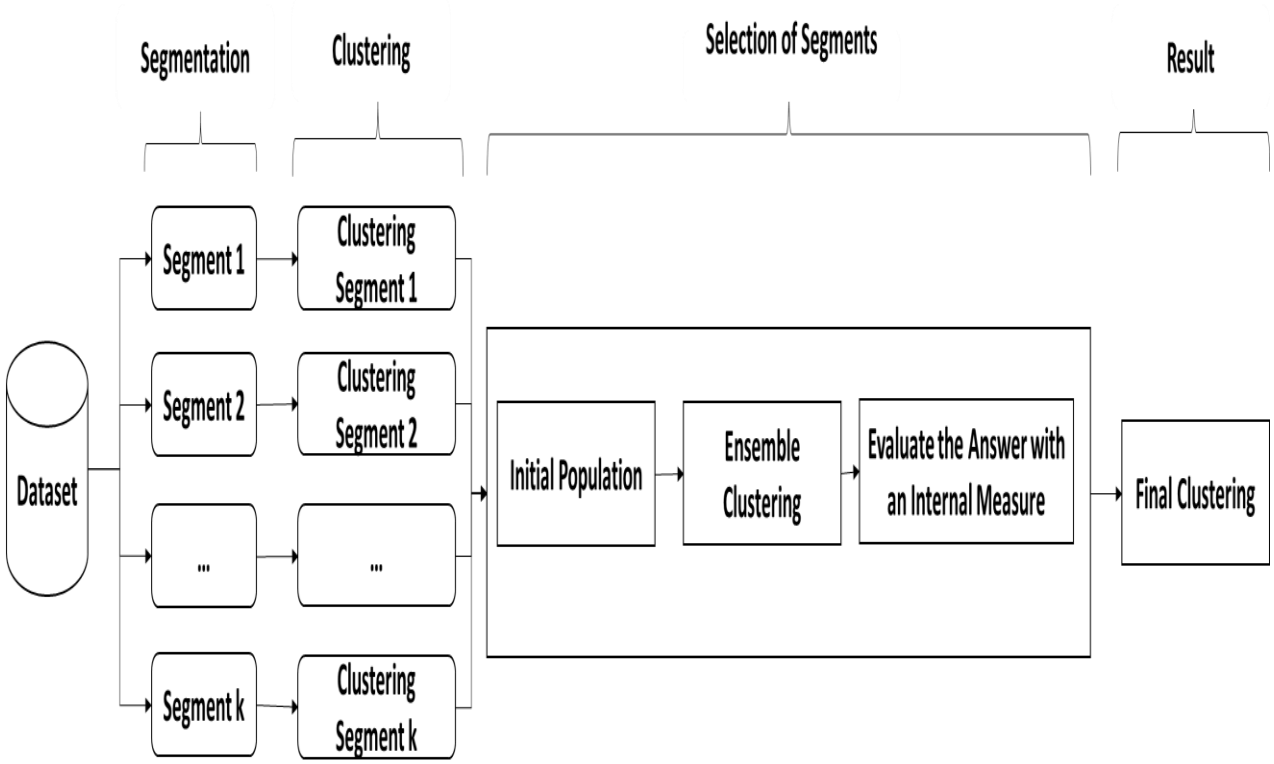


Figure 2. The Proposed algorithm.

2.1. Segmentation

A time series can be divided into smaller segments using two different approaches, where the arrangement of these segments creates the original time series. In the first approach, specific algorithms can transform a time series into segments of varying lengths [29, 30]. The second approach, also used in the developed algorithm, defines a window of length L, and the time series is divided into equal segments with lengths equal to L [31]. Figure 1 illustrates this type of segmentation.

2.2. Clustering

In the second step of the proposed approach, the segments created in the first step have been clustered using a hierarchical agglomerative algorithm. This algorithm uses the complete distance (d_{max}) as the linkage distance between clusters. Equation (1) represents this type of distance. A and B represent the clusters, and a and b represent the objects of each cluster, while d denotes the distance between two objects [16].

$$d_{max} = \max\{d(a,b) : a \in A, b \in B\} \quad (1)$$

To calculate the distance between two time series, different distance metrics can be used, such as Euclidean distance, Dynamic Time Warping (DTW), and Longest Common Subsequence (LCSS) [32-34]. Considering the computational cost of DTW and LCSS metrics, this study utilizes the Euclidean distance [35-37]. If two-time series, X and Y, of length N are given, the Euclidean distance (ED) can be computed using equation (2). However, it should be noted that the Euclidean distance is applicable only when the two-time series have the same length [32].

$$ED(X,Y) = \sqrt{\sum_{t=1}^{n} (x_t - y_t)^2} \quad (2)$$

2.3. Segment Selection

The third and final step of the approach involves selecting suitable segments from the created segments and combining them. This process uses a metaheuristic algorithm, ensemble clustering, and an internal criterion. A solution generated in the genetic algorithm is represented using a binary gene representation of zeros and ones. If a gene value is one, it indicates the selection of a segment from the available segments. On the other hand, if the gene value is zero, it indicates the non-selection of a segment. Figure 3 demonstrates the process of generating a solution in the genetic algorithm.

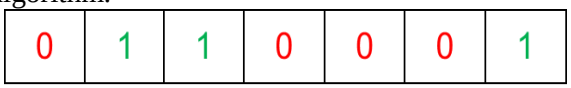


Figure 3. Represent the solution (Chromosome) in genetic algorithm.

In the next step, in this phase, only the selected segments are combined using an ensemble clustering algorithm, resulting in a final solution. In the last step, an internal criterion is used as an activity function in the genetic algorithm to calculate the fitness function value for the generated solution to evaluate the solution.

2.3.1. Genetic algorithm

Based on the previous explanations, the genetic algorithm has been used to select appropriate segments. The utilized algorithm includes two leading operators: crossover and mutation. The crossover operator combines two parent solutions to create new solutions, known as offspring, based on the representation of the solution for this problem. Two types of operators, single-point and two-point crossover, have been used for the crossover operator. Figure 4 represents a single-point crossover operator. In this operator, a random point is selected in both parents and by swapping the segments of the parents from the selected point, two new offspring solutions are created.

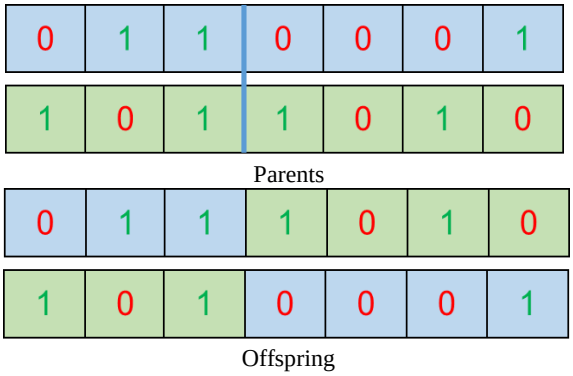
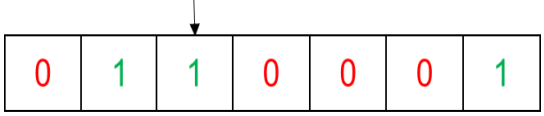


Figure 4. Crossover operation.

Furthermore, the genetic algorithm utilizes a mutation operator to escape from a local optimum. In this operator, initially, a gene is randomly selected within the chromosome, and its value is inverted. In other words, if the value is zero, it changes to one, and if it is one, it changes to zero. Figure 5 represents the mutation operator used. Additionally, the algorithm's parameters have been adjusted using the Taguchi method.



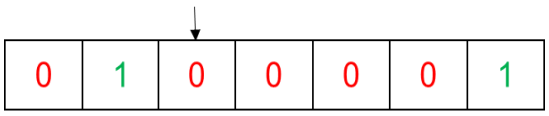


Figure 5. Mutation operation.

2.3.2. Ensemble clustering

One of the methods used to achieve higher accuracy in clustering is ensemble clustering. This approach creates multiple solutions for a dataset and combines them to obtain the final solution. The ensemble clustering approach consists of two main steps: generation and consensus. Typically, homogeneous and heterogeneous methods are used in the generation step to produce initial solutions. Additionally, various techniques, such as pairwise similarities and graph theory, are employed in the aggregation step. Figure 6 illustrates the combined clustering approach's overall concept and different components. An extended algorithm based on graph theory called LWGP is utilized in the presented framework. In this method, the distances between the created clusters for each object are computed initially. Then, using these distances and a clustering algorithm, the final labels for each object are determined [38].

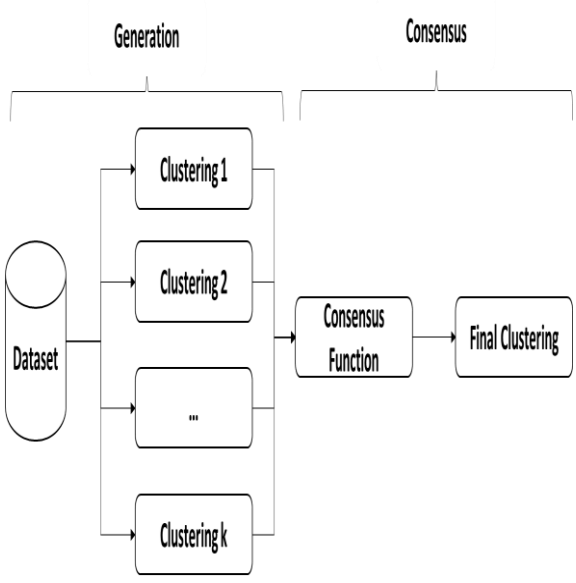


Figure 6. Ensemble clustering.

2.3.3. Internal and External Measures

Internal and external measures are two indices used to assess a dataset's clustering accuracy. In this study, the internal measure has been utilized as the fitness function for the genetic algorithm. During the selection phase of the genetic algorithm, since the final labels of the objects are unknown, the internal measure of inter-group variance has been employed as the fitness function for the genetic algorithm. A lower value for this measure indicates a better quality. Equation (3) represents the calculation method for this measure. In this context, n and k represent the number of objects in the dataset and number of clusters, respectively. Further, x represents an object and ci denotes the ith cluster.

$$\frac{1}{n-k}\sum_{i=1}^{k}\sum_{x\in c_i} d\left(x, c_i\right) \quad (3)$$

Additionally, an external measure has been used to assess and compare the algorithm's accuracy with existing algorithms. The specific external measure utilized in this study is the Rand Index (RI). Suppose TP represents the number of objects with the same class and cluster. In that case, TN represents the number of objects that have different classes and clusters, FP represents the number of objects that have different clusters but the same class, and finally, FN represents the number of objects that have the same cluster but different classes. Then, Equation (4) represents the calculation method for the Rand Index.

$$RI = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

3. Implementing the proposed model

The algorithm's performance has been evaluated by running it on 82 different datasets from the UCR website. For this purpose, the model parameters have been adjusted, including the number of segments created for each dataset and the genetic algorithm parameters.

3.1. Parameter Tuning

The number of segments created for each dataset can vary. If the number of segments is too small, the probability of identifying ideal segments decreases. Conversely, if the number of segments is too large, the characteristics of each segment will be lost. Therefore, a logarithmic relationship has been used to determine the number of segments created for each dataset. This relationship ensures that the number of segments increases slowly as the time series length increases.

Additionally, it provides that a sufficient number of segments is created even for shorter lengths. Equation (5) represents the calculation method for determining the number of segments (k). In this equation, L represents the length of the time series dataset.

$$k = \log_{10} L \quad (5)$$

Additionally, a Taguchi design has been utilized to optimize the parameters of the genetic

algorithm. In this design, the parameters of the number of iterations, population size, crossover rate, and mutation rate have been optimized. Table 1 displays the levels used in the Taguchi design, representing the specific values assigned to each parameter for optimization purposes.

Table 1. Taguchi method levels.

Parameters	Symbol	Levels
Crossover rate	*pc*	0.4 – 0.65 – 0.9
Mutation rate	*pm*	0.1 - 0.3 – 0.5
Population size	*pop*	100 - 200 - 300
Iteration	*IT*	10 - 20 - 30

Table 2 displays nine experiments resulting from the design and the outcomes of ten iterations of the provided algorithm for a time series dataset.

Table 2. Taguchi method and Rand Index values for the proposed algorithm.

Number of experiment	*pc*	*pm*	*pop*	*It*	RI
1	0.40	0.1	100	10	71.00%
2	0.40	0.3	200	20	72.38%
3	0.40	0.5	300	30	72.27%
4	0.65	0.1	200	30	72.30%
5	0.65	0.3	300	10	72.58%
6	0.65	0.5	100	20	72.42%
7	0.90	0.1	300	20	72.23%
8	0.90	0.3	100	30	72.23%
9	0.90	0.5	200	10	72.33%

Based on the reported signal-to-noise ratio for the provided algorithm (Figure 7), the parameter values for crossover rate, mutation rate, population size, and number of iterations are set to 0.65, 0.3, 300, and 20, respectively.

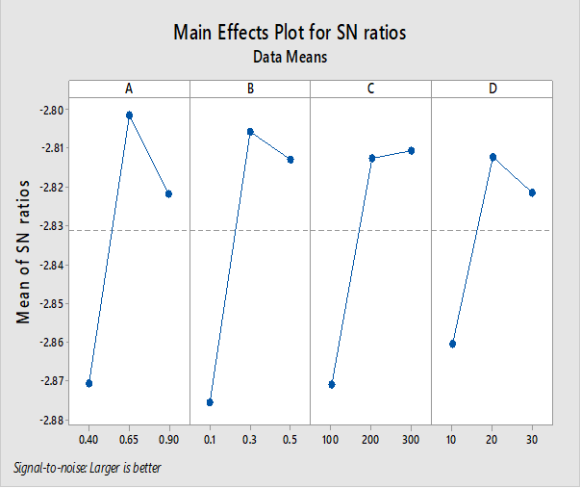


Figure 7. Signal-to-noise ratio.

4. Results

The presented algorithm has been evaluated by performing ten iterations for each dataset and examining the Rand index and execution time. To assess the algorithm's performance, the Rand index of the proposed approach was compared to those without segmentation. According to the information in Table 3, the average Rand index for 82 datasets, without segmentation, is reported as 64.33, while the Rand index for the developed algorithm is 67.40, indicating a 3.07% improvement in the Rand index. Additionally, considering the improvement value, it can be observed that the algorithm has improved the Rand index value in 46 datasets compared to without segmentation case. The average standard deviation of 10 iterations is 0.38%, indicating relatively good stability of the algorithm. In the best case, the developed algorithm has achieved a 46.97% improvement in the Rand index for one dataset. The maximum and minimum values of the Rand Index for the developed algorithm were 95.20% and 36.86%, respectively. In contrast, these values are 94.58% and 30.59%, respectively, for the non-segmented case.

4.2. Sensitivity Analysis

In this section, the performance of the algorithm is evaluated based on two assessment criteria. In the first part, a sensitivity analysis was conducted concerning the length of the time series, and in the second part, the algorithm's performance was examined for the type of time series under investigation. In this section, the efficiency of the proposed algorithm has been examined according to the length of each time series dataset. For this purpose, the datasets under investigation are divided into three categories: short (less than 200), medium (between 200 and 500), and long (greater than 500) based on their length. The Rand index value and the improvement achieved are examined for all three introduced classes. According to the information in Figure 8, it can be observed that the algorithm performs best in the medium-length classes. In this class, the algorithm has managed to increase the Rand index by 3.8 compared to the without-segmentation case. It can also be seen that the algorithm's performance in the short-length class does not differ significantly from the medium-length class, with an improvement value of 3.4. However, the algorithm's performance in the long-length class is relatively lower than the other two classes, which is noticeable. Furthermore, the medium-length datasets have the highest Rand index value of 70.5%, significantly different from the other two classes. Overall, it can be concluded that the algorithm performs best in the medium-length class, followed by the short and long-length

categories, respectively

Table 3. Results of Rand index and Standard Deviation for the proposed algorithms.

Data sets	Number of cluster	Rand Index without segmentation	Rand Index	Standard Deviation	Improve	Data sets	Number of cluster	Rand Index without segmentation	Rand Index	Standard Deviation	Improve
ADI	37	86.89%	92.03%	0.34%	5.14%	**MPA**	2	58.06%	73.95%	0.00%	15.89%
ARR	3	34.45%	62.63%	0.25%	28.18%	**MPC**	3	52.86%	49.96%	0.00%	-2.90%
BEE	5	58.81%	66.85%	0.03%	8.03%	**MPT**	2	82.63%	81.33%	0.58%	-1.30%
BFL	2	51.92%	53.33%	0.00%	1.41%	**MOT**	6	51.01%	57.63%	0.00%	6.62%
BIR	2	49.23%	49.36%	0.29%	0.13%	**NO1**	42	89.72%	94.20%	0.60%	4.48%
CAR	4	61.51%	66.47%	0.02%	4.96%	**NO2**	42	93.73%	94.78%	0.23%	1.05%
CBF	3	64.20%	63.26%	0.00%	-0.94%	**OLI**	4	84.07%	89.04%	0.00%	4.97%
CHL	3	39.96%	52.79%	0.00%	12.83%	**OSU**	6	73.24%	70.25%	1.16%	-2.99%
CIN	4	63.83%	65.66%	0.15%	1.82%	**PHA**	2	53.96%	50.00%	0.00%	-3.97%
COF	2	50.13%	74.08%	2.21%	23.95%	**PHO**	39	92.74%	86.45%	0.23%	-6.28%
COM	2	49.90%	49.90%	0.00%	0.00%	**PLA**	7	91.19%	94.56%	0.42%	3.36%
CRX	12	82.41%	83.45%	0.30%	1.04%	**PPA**	3	41.73%	78.57%	0.14%	36.83%
CRY	12	83.24%	78.45%	3.90%	-4.80%	**PPC**	2	56.52%	53.31%	0.00%	-3.21%
CRZ	12	84.87%	81.71%	0.47%	-3.16%	**PPT**	6	81.75%	74.84%	0.24%	-6.91%
DIA	4	30.59%	77.56%	0.96%	46.97%	**REF**	3	55.25%	36.86%	3.74%	-18.39%
DPA	3	71.72%	72.85%	0.00%	1.14%	**SCR**	3	44.68%	53.66%	1.33%	8.98%
DPC	2	52.71%	49.96%	0.00%	-2.76%	**SHS**	2	49.75%	49.77%	0.00%	0.02%
DPT	6	87.42%	80.48%	4.80%	-6.93%	**SHA**	60	94.20%	95.20%	0.36%	0.99%
EAR	2	52.30%	56.76%	0.34%	4.46%	**SMA**	3	41.14%	44.18%	0.19%	3.03%
EC2	2	60.38%	62.31%	0.00%	1.93%	**SO1**	2	53.45%	59.60%	0.00%	6.14%
EC5	5	84.87%	84.03%	0.00%	-0.83%	**SO2**	2	56.41%	59.50%	0.00%	3.09%
ECF	2	49.99%	50.09%	0.00%	0.10%	**STR**	2	52.26%	50.24%	0.00%	-2.01%
FAA	14	83.70%	85.09%	0.53%	1.39%	**SWE**	15	52.11%	84.97%	0.57%	32.86%
FAF	4	67.82%	74.90%	0.00%	7.08%	**SYM**	6	77.86%	89.67%	0.00%	11.81%
FIS	7	71.00%	73.52%	1.21%	2.51%	**SYN**	6	79.08%	81.03%	0.00%	1.94%
FOA	2	50.24%	50.04%	0.00%	-0.20%	**TO1**	2	49.95%	49.82%	0.00%	-0.13%
FOB	2	50.02%	49.99%	0.00%	-0.03%	**TO2**	2	53.56%	49.70%	0.00%	-3.86%
GUN	2	50.07%	49.75%	0.00%	-0.32%	**TRA**	4	75.12%	74.90%	0.01%	-0.22%
HAM	2	49.92%	51.89%	0.00%	1.97%	**TWP**	2	61.29%	61.27%	2.45%	-0.03%
HAN	2	53.94%	67.25%	0.08%	13.31%	**TWE**	4	50.34%	51.08%	0.00%	0.74%
HAP	5	55.80%	66.84%	0.36%	11.05%	**UWX**	8	80.79%	82.97%	0.00%	2.19%
HER	2	50.21%	50.05%	0.00%	-0.16%	**UWY**	8	83.08%	82.04%	0.00%	-1.04%
INL	7	50.95%	72.30%	0.11%	21.35%	**UWZ**	8	81.93%	83.04%	0.04%	1.11%
INS	11	85.51%	85.97%	0.32%	0.46%	**UWA**	8	85.76%	88.04%	0.13%	2.28%
ITA	2	51.43%	50.01%	0.00%	-1.42%	**W50**	50	94.58%	93.78%	0.07%	-0.80%
LAR	3	53.16%	51.04%	0.00%	-2.12%	**WAF**	2	53.44%	53.44%	0.00%	0.00%
LI2	2	60.00%	50.36%	0.00%	-9.64%	**WIN**	2	49.88%	49.58%	0.00%	-0.29%
LI7	7	74.74%	71.76%	0.98%	-2.98%	**WOS**	25	88.83%	86.94%	0.70%	-1.90%
MAL	8	92.89%	90.79%	0.12%	-2.10%	**WOR**	5	63.44%	63.69%	0.53%	0.26%
MEA	3	77.04%	72.49%	0.00%	-4.55%	**WOT**	2	49.81%	49.95%	0.06%	0.14%
MED	10	64.18%	65.30%	0.02%	1.13%	**YOG**	2	50.00%	49.99%	0.00%	-0.01%
Average		64.33%	67.40%	0.38%	3.07%						

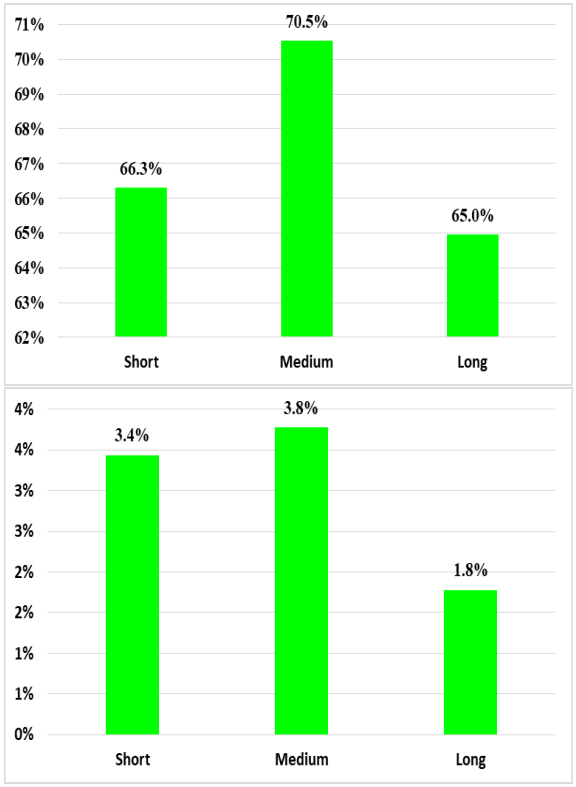


Figure 8. The performance of the developed algorithms according to the length of the time series.

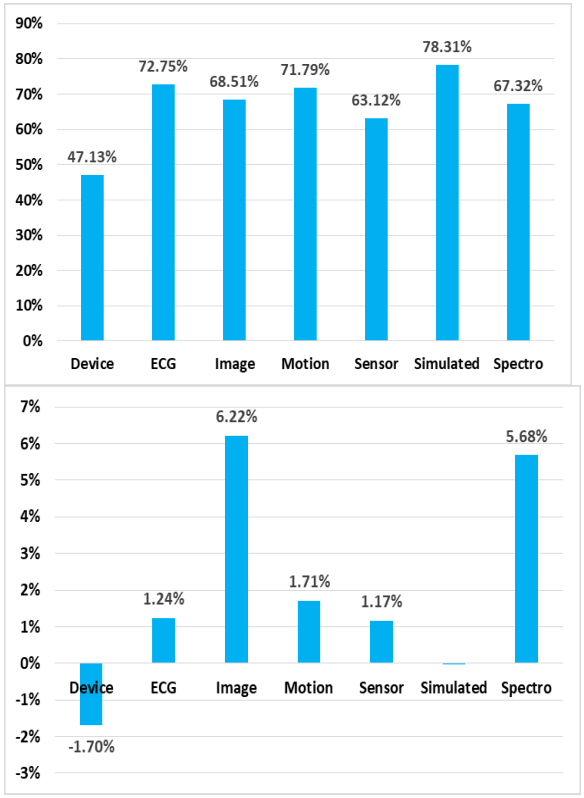


Figure 9. The performance of the developed algorithms according type of dataset.

According to the UCR website, the datasets used have been categorized into 6 types: Device, ECG, Image, Motion, Sensor, Simulated, and Spectro. The number datasets for each type are 5, 6, 28, 14, 18, 5, and 6. The Rand index and improvement value were examined for seven types of time series which were introduced previously. Based on the Figure 9 and the Rand index, the best and worst average Rand index was observed for the Simulated and Device, respectively. The average Rand index values for these two data types were 78.31 and 47.13, respectively. Additionally, based on the figure data, it can be seen that the increase in the Rand index for the developed algorithm on the device type is -1.70, whereas it is 0 for the simulated type. The increase values were positive for the other five types, with the highest being 6.22 for the image type. Considering both the Rand index and the value change, it is evident that the algorithm's performance is acceptable for six of the seven types of time series, with only one type showing poor performance.

To investigate the execution time of the algorithm, 82 datasets have been divided into three categories based on their size. The first category of data sets whose number of objects is less than 500 (small), the second category of data sets whose number of objects is between 500 and 1000 (medium), and the third category of data sets whose number of objects is more than 1000 (big).

According to Figure 10, the average execution time for small data sets is equal to 151 seconds. With the doubling of the size of the data set, the algorithm execution time has almost doubled and increased linearly. However, for large data sets, the average execution time equals 5312 seconds, which shows an exponential increase.

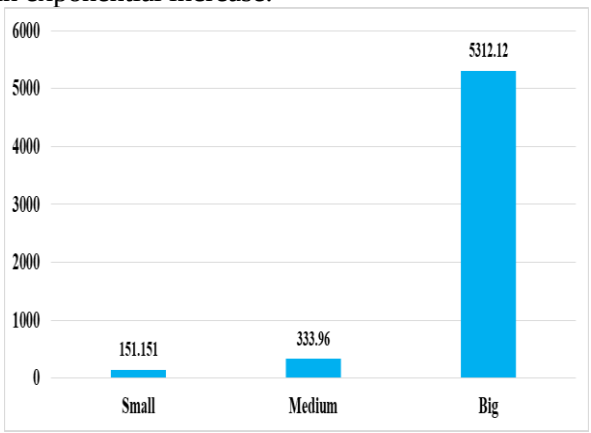


Figure 10. The performance of the Run time of the developed algorithms according to size of the dataset.

4.3. Statistical analysis

Based on the two criteria of the Rand index and execution time, six algorithms from the literature have been selected for comparing these criteria. The six mentioned algorithms are DTW, which uses Dynamic Time Warping distance metrics with a hierarchical algorithm, which uses

Derivative Dynamic Time Warping distance metrics with a hierarchical algorithm, DD_{DTW}, which combines two distance metrics with a hierarchical algorithm [16]; KSC algorithm, which utilizes specific temporal patterns for clustering [39]; $TS3C_{\mathrm{MV}}$ and $TS3C_{\mathrm{CH}}$ algorithms, which are developed based on temporal segmentation characteristics [27]. According to the information in Table 4, it can be observed that the developed algorithm outperforms all six introduced algorithms in terms of the Rand index. Additionally, in terms of execution time, the developed algorithm performs better, with an average of 1899 seconds compared to the other six algorithms.

In summary, the developed algorithm demonstrates superior performance in terms of the Rand index and execution time compared to the six introduced algorithms.

Table 5 presents a complete display of the Rand index and execution time for the six selected algorithms and the developed algorithm for each dataset. According to the information in Table 5, it can be observed that the developed algorithm provides the best solution among the five algorithms for 20% of the datasets

Table 4. Rand index and run time of the presented algorithm and six previous studies.

	Proposed algorithm	$TS3C_{\mathrm{CH}}$	$TS3C_{\mathrm{MV}}$	DD_{DTW}	**KSC**	**DTW**	D_{DTW}
Rand Index	67.40	66.10	65.68	60.55	60.26	58.20	45
Run Time	1,899	3,270	3,284	23,6467	17,365	-	-

Table 5. Comparison of the Rand Index and running time of the proposed algorithm and six other selected algorithms.

Data set	**Rand Index**							**Run time**				
	Proposed algorithm	$TS3C_{\mathrm{CH}}$	$TS3C_{\mathrm{MV}}$	DD_{DTW}	**KSC**	**DTW**	D_{DTW}	**Proposed algorithm**	$TS3C_{\mathrm{CH}}$	$TS3C_{\mathrm{MV}}$	DD_{DTW}	**KSC**
50W	93.78	94	94	92	66	92	65	681	1478	1503	104285	17667
ADI	92.03	92	92	687	95	71	19	348	887	904	32221	5374
ARR	62.63	62	63	35	63	35	35	48	361	362	4603	84
BEE	66.85	68	68	58	71	42	56	40	188	188	1247	63
BFL	53.33	49	49	59	50	59	50	24	136	136	642	21
BIR	49.36	49	49	50	54	50	51	25	117	117	688	19
CAR	66.47	65	65	50	68	48	32	43	392	392	8371	142
CBF	63.26	67	67	78	56	42	33	383	976	978	24830	584
CHL	52.79	49	47	40	53	40	40	7,541	6556	6601	763587	2623
CIN	65.66	64	64	56	69	48	28	2,296	11648	11653	763587	27772
COF	74.08	51	51	49	75	50	49	23	98	98	385	10
COM	49.90	50	51	50	50	50	50	319	2699	2700	210113	413
CRX	83.45	85	85	78	41	71	37	249	1708	1715	82970	4356
CRY	78.45	84	84	69	53	70	23	294	1752	1759	83710	4488
CRZ	81.71	84	85	71	41	71	35	281	1680	1687	78563	3807
DIA	77.56	72	72	30	96	30	30	65	623	624	20433	303
DPA	72.85	60	60	71	72	71	71	114	312	314	3091	83
DPC	49.96	51	51	53	50	53	53	325	484	487	8536	45
DPT	80.48	68	66	86	66	80	86	125	303	306	3231	194
EAR	56.76	53	53	54	62	51	55	212	2246	2247	93262	308
EC2	62.31	50	50	54	61	54	55	32	122	122	651	18
EC5	84.03	64	60	89	59	88	47	8,272	5019	5088	736065	16214
EFC	50.09	50	50	50	81	51	51	344	942	945	19578	136
FAA	85.09	85	85	60	30	60	36	1,720	1876	1893	763587	5236
FAF	74.90	57	57	55	38	54	42	51	270	270	2449	90
FIS	73.52	73	64	18	79	17	17	113	930	932	42707	1265

FOA	50.04	52	51	54	50	50	50	16,103	16265	16316	763587	36823
FOB	49.99	50	50	50	50	50	50	13,385	12187	12222	763587	28637
GUN	49.75	54	54	50	51	50	50	34	134	134	2009	10
HAM	51.89	52	52	50	53	50	50	45	557	558	14039	152
HAN	67.25	50	55	55	69	54	55	5,400	20404	20408	763587	34052
HAP	66.84	60	60	39	69	22	39	220	2302	2303	763587	3881
HER	50.05	50	50	51	50	51	52	37	273	273	7475	38
INL	72.30	71	71	54	74	54	17	599	5598	5600	763587	11234
INS	85.97	81	81	55	69	20	21	1,978	3661	3685	474905	10473
ITA	50.01	50	50	51	64	50	50	467	242	246	1233	20
LAR	51.04	55	55	34	41	34	34	556	3906	3908	369379	711
LI2	50.36	50	54	50	50	50	50	35	543	544	9478	128
LI7	71.76	75	75	60	59	63	56	67	323	324	3125	180
MAL	90.79	80	80	93	92	93	81	3,370	14510	14530	763587	18388
MEA	72.49	71	40	77	76	77	34	38	279	279	4718	86
MED	65.30	65	65	64	47	60	46	422	646	651	19954	1062
MPA	73.95	56	56	73	73	70	73	143	250	251	4086	70
MPC	49.96	51	51	50	50	53	50	337	395	396	9306	65
MPT	81.33	74	82	80	81	79	79	140	286	289	3319	204
MOT	57.63	50	50	50	58	50	50	696	717	721	20456	496
NO1	94.20	94	95	70	95	64	16	3,252	17115	17226	763587	218650
NO2	94.78	95	95	85	97	82	56	2,632	13594	13676	763587	208416
OLI	89.04	77	77	76	85	74	74	47	139	139	2444	66
OSU	70.25	73	73	62	29	58	29	179	1097	1100	60765	670
PHA	50.00	51	51	54	51	54	54	2,756	1499	1515	77558	200
PHO	86.45	93	93	45	51	42	17	2,506	15477	15525	763587	364703
PLA	94.56	83	80	100	92	96	95	66	149	149	2047	53
PPA	78.57	76	76	78	76	77	77	144	235	236	4909	82
PPC	53.31	56	56	54	53	54	52	317	418	421	8195	32
PPT	74.84	78	78	88	81	79	87	147	319	321	3851	221
REF	36.86	56	54	35	39	34	35	609	3969	3972	422843	900
SCR	53.66	53	53	35	45	33	35	549	4022	4024	355109	1358
SHS	49.77	99	99	50	50	50	50	51	1005	1006	13879	111
SHA	95.20	97	97	84	63	77	46	1,114	3738	3777	546585	51107
SMA	44.18	59	59	34	54	34	34	645	4048	4050	379869	1487
SO1	59.60	51	52	50	75	50	51	173	349	350	2770	74
SO2	59.50	60	53	53	66	53	53	397	607	610	9040	149
STR	50.24	50	52	50	50	52	52	539	1331	1334	92052	366
SWE	84.97	88	88	35	63	35	26	549	985	996	36092	1620
SYM	89.67	81	81	89	60	89	17	552	2774	2779	248454	2867
SYN	81.03	78	78	88	38	88	24	110	361	365	2112	235
TO1	49.82	51	51	51	53	50	50	62	527	528	8315	121
TO2	49.70	50	50	67	53	69	61	40	338	339	5258	79
TRA	74.90	84	84	87	72	87	62	56	325	325	4986	114
TWP	61.27	64	64	85	46	72	25	8,839	6465	6532	581050	4602
TWE	51.08	64	64	50	54	50	50	558	703	707	11903	240
UWZ	82.97	78	75	80	51	79	14	7,535	7470	7523	763587	47423

UWY	82.04	78	76	82	54	80	13	7,282	8951	9014	763587	48751
UWZ	83.04	80	80	74	54	75	13	7,327	7264	7315	763587	47554
UWA	88.04	76	76	59	45	59	13	9,470	18754	18799	763587	167079
WAF	53.44	50	66	53	59	53	68	21,546	4087	4158	763587	1682
WIN	49.58	57	50	50	59	50	50	30	105	105	1273	31
WOS	86.94	87	87	87	50	84	24	481	1463	1476	98233	8212
WOR	63.69	60	58	62	53	62	30	133	1605	1606	82527	1085
WOT	49.95	51	51	50	50	51	51	93	1620	1621	77497	676
YOG	49.99	51	50	50	50	50	50	6,923	7959	7983	763587	4848
Average	**67.40**	**66.10**	**65.68**	**60.55**	**60.26**	**58.20**	**45**	**1,899**	**3270**	**3284**	**236467**	**17365**

To examine the Rand Index and execution time of the proposed algorithms more accurately, a non-parametric statistical test called the Wilcoxon signed-rank test was employed [٤٠]. This test was used to assess one sample before and after the influence of a given factor. It uses the concept of differences in ranks to investigate the significant differences between the two samples. The null (H_0) and alternative hypotheses (H_1) used in this study are represented by equations (6) and (7) for Rand Index and execution time: In this context, μ_0 represents the mean Rand Index and execution time for the developed algorithm, and μ_1 denotes the mean Rand Index and execution time for the six selected algorithms.

$$H_0 : \mu_0 = \mu_1$$
$$H_1 : \mu_0 \geq \mu_1 \quad (6)$$

$$H_0 : \mu_0 = \mu_1$$
$$H_1 : \mu_0 \leq \mu_1 \quad (7)$$

Table 6 displays the p-values and Wilcoxon Statistic for the Rand index and execution time. Assuming an alpha value of 10% for this test, it can be observed that the p-value for the six selected algorithms is less than 10% in both the Rand index and execution time criteria. Therefore, it can be concluded that the developed algorithm outperforms the six selected algorithms in both the Rand index and execution time criteria, with a confidence level of 90%.

5. Conclusion

In previous studies, the focus has been chiefly on using novel distance measures. Although using these measures has somewhat improved the clustering accuracy in different datasets, it has also increased the computational cost, rendering the use of these algorithms less efficient in practice. The developed algorithm's most significant strength lies in increasing clustering accuracy while reducing the execution time cost, making it highly effective in practice.

This research proposed a combined segmentation and clustering algorithm for clustering time series data in three main steps. The primary basis of the proposed algorithm is the selection and combination of suitable dataset segments. In this approach, a dataset is initially divided into equal segments. Then, appropriate segments are selected using an iterative algorithm and combined to obtain the final solution. The developed algorithm was implemented on 82 datasets, with an average Rand index of 67.40 and an execution time of 1899 seconds. The obtained results demonstrate that the developed algorithm improved the Rand index by 3.07% compared to the non-segmented approach. Sensitivity analysis of the developed algorithm showed that it performs best on time series with average lengths. Additionally, the developed algorithm was compared to six selected algorithms (DTW, D_{DTW}, DD_{DTW}, KSC, $TS3C_{\mathrm{MV}}$, and $TS3C_{\mathrm{CH}}$) in terms of the Rand index and execution time using the Wilcoxon statistical test, indicating its superior performance in both the Rand index and execution time compared to these six algorithms. The Rand index and improvement value were examined for seven types of time series which were introduced previously. The best and worst average Rand index was observed for the Simulated and Device, respectively.

Table 6. Results of statistical tests of the presented algorithm and six previous studies.

		$TS3C_{\mathrm{CH}}$	$TS3C_{\mathrm{MV}}$	DD_{DTW}	KSC	DTW	D_{DTW}
Rand Index	P-value	0.05	0.086	0.00	0.025	0.00	0.00
	Wilcoxon Statistic	2058.5	1997.0	2551.0	2125.5	2740.5	3043.0
Run time	P-value	0.00	0.00	0.00	0.00	-	-
	Wilcoxon Statistic	414.0	407.0	0.0	764.5	-	-

References

[1] M. Maleki, H. Bidram, and D. Wraith, "Robust clustering of COVID-19 cases across US counties using mixtures of asymmetric time series models with time varying and freely indexed covariates," *Journal of Applied Statistics.* vol. 50, pp. 2648–2662, 2022.

[2] M. Castán-Lascorz, P. Jiménez-Herrera, A. Troncoso, and G. Asencio-Cortés, "A new hybrid method for predicting univariate and multivariate time series based on pattern forecasting," *Information Sciences.* vol. 586, pp. 611-627, 2022.

[3] P. Laurinec, M. Lóderer, M. Lucká, and V. Rozinajová, "Density-based unsupervised ensemble learning methods for time series forecasting of aggregated or clustered electricity consumption," *Journal of Intelligent Information Systems.* vol. 53, pp. 219-239, 2019.

[4] S. Xu, H. K. Chan, E. Ch'ng, and K. H. Tan, "A comparison of forecasting methods for medical device demand using trend-based clustering scheme," *Journal of Data, Information and Management.* vol. 2, pp. 85–94, 2020.

[5] T. M. Dantas and F. L. C. Oliveira, "Improving time series forecasting: An approach combining bootstrap aggregation, clusters and exponential smoothing," *International Journal of Forecasting.* vol. 34, pp. 748-761, 2018.

[6] J. Li, H. Izakian, W. Pedrycz, and I. Jamal, "Clustering-based anomaly detection in multivariate time series data," *Applied Soft Computing.* vol. 100, p. 106919, 2021.

[7] P. D'Urso, L. De Giovanni, and R. Massari, "Trimmed fuzzy clustering of financial time series based on dynamic time warping," *Annals of Operations Research.* vol. 299, pp. 1379-1395, 2021.

[8] S. Datta, S. Rokade, and S. P. Rajput, "Classification of uncontrolled intersections using hierarchical clustering," *Arabian Journal for Science and Engineering.* vol. 45, pp. 8591-8606, 2020.

[9] A. Hatamlou and M. Deljavan, "Forecasting gold price using data mining techniques by considering new factors," *Journal of AI and Data Mining.* vol. 7, pp. 411-420, 2019.

[10] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering–a decade review," *Information Systems.* vol. 53, pp. 16-38, 2015.

[11] L. Wang and P. Koniusz."Uncertainty-DTW for time series and sequences," *presented at the European Conference on Computer Vision*, 2022, pp. 176-195.

[12] G. Soleimani and M. Abessi, "DLCSS: A new similarity measure for time series data mining," *Engineering Applications of Artificial Intelligence.* vol. 92, p. 103664, 2020.

[13] M. A. Rahim Khan and M. Zakarya, "Longest common subsequence based algorithm for measuring similarity between time series: a new approach," *World Applied Sciences Journal.* vol. 24, pp. 1192-1198, 2013.

[14] H. Kamalzadeh, A. Ahmadi, and S. Mansour, "Clustering time-series by a novel slope-based similarity measure considering particle swarm optimization," *Applied Soft Computing.* vol. 96, p. 106701, 2020.

[15] X. Wang, F. Yu, W. Pedrycz, and J. Wang, "Hierarchical clustering of unequal-length time series with area-based shape distance," *Soft Computing.* vol. 23, pp. 6331-6343, 2019.

[16] M. Łuczak, "Hierarchical clustering of time series data with parametric derivative dynamic time warping," *Expert Systems with Applications.* vol. 62, pp. 116-130, 2016.

[17] R. Ma and R. Angryk."Distance and density clustering for time series data," *presented at the 2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2017, pp. 25-32

[18] T. Górecki, "Classification of time series using combination of DTW and LCSS dissimilarity measures," *Communications in Statistics-Simulation and Computation.* vol. 47, pp. 263-276, 2018.

[19] S. Aghabozorgi, T. Ying Wah, T. Herawan, H. A. Jalab, M. A. Shaygan, and A. Jalali, "A hybrid algorithm for clustering of time series data based on affinity search technique," *The Scientific World Journal.* vol. 2014, 2014.

[20] X. Zhang, J. Liu, Y. Du, and T. Lv, "A novel clustering method on time series data," *Expert Systems with Applications.* vol. 38, pp. 11891-11900, 2011.

[21] N. Manakova and V. Tkachenko."Two-stage time-series clustering approach under reducing time cost requirement," *presented at the 2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, 2020, pp. 653-658.

[22] Z. Izakian and M. Mesgari, "Fuzzy clustering of time series data: A particle swarm optimization approach," *Journal of AI and Data Mining.* vol. 3, pp. 39-46, 2015.

[23] R. J. Hyndman, E. Wang, and N. Laptev."Large-scale unusual time series detection," *presented at the 2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, 2015, pp. 1616-1619.

[24] Y. Zou, R. V. Donner, N. Marwan, J. F. Donges, and J. Kurths, "Complex network approaches to nonlinear time series analysis," *Physics Reports.* vol. 787, pp. 1-97, 2019.

[25] L. N. Ferreira and L. Zhao, "Time series clustering via community detection in networks," *Information Sciences.* vol. 326, pp. 227-242, 2016.

[26] H. Liu, J. Zou, and N. Ravishanker, "Clustering high-frequency financial time series based on information theory," *Applied Stochastic Models in Business and Industry.* vol. 38, pp. 4-26, 2022.

[27] D. Guijo-Rubio, A. M. Durán-Rosal, P. A. Gutiérrez, A. Troncoso, and C. Hervás-Martínez, "Time-Series Clustering Based on the Characterization of Segment Typologies," *IEEE Transactions on Cybernetics.* vol. 51, pp. 5409-5422, 2020.

[28] F. Bonacina, E. S. Miele, and A. Corsini, "Time Series Clustering: A Complex Network-Based Approach for Feature Selection in Multi-Sensor Data," *Modelling.* vol. 1, pp. 1-21, 2020.

[29] A. Koski, M. Juhola, and M. Meriste, "Syntactic recognition of ECG signals by attributed finite automata," *Pattern Recognition.* vol. 28, pp. 1927-1940, 1995.

[30] E. J. Keogh and M. J. Pazzani."An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback," *presented at the Knowledge Discovery and Data Mining*, 1998, pp. 239-243.

[31] E. Keogh, S. Chu, D. Hart, and M. Pazzani (2004), "Segmenting time series: A survey and novel approach," in *Data Mining in Time Series Databases*, M. Last Ed.: World Scientific, pp. 1-21.

[32] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," *ACM Sigmod Record.* vol. 23, pp. 419-429, 1994.

[33] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems.* vol. 7, pp. 358-386, 2005.

[34] M. Djukanovic, G. R. Raidl, and C. Blum, "Finding Longest Common Subsequences: New anytime A∗ search results," *Applied Soft Computing.* vol. 95, p. 106499, 2020.

[35] M. Paterson and V. Dančík."Longest common subsequences," *presented at the International Symposium on Mathematical Foundations of Computer Science*, 1994, pp. 127-142.

[36] R. Lin, A. King-lp, and H. S. S. K. Shim."Fast similarity search in the presence of noise, scaling, and translation in time-series databases," *presented at the Proceeding of the 21th International Conference on Very Large Data Bases*, 1995, pp. 490-501.

[37] M. Vlachos, G. Kollios, and D. Gunopulos."Discovering similar multidimensional trajectories," *presented at the Proceedings 18th International Conference on Data Engineering*, 2002, pp. 673-684.

[38] D. Huang, C.-D. Wang, and J.-H. Lai, "Locally weighted ensemble clustering," *IEEE Transactions on Cybernetics.* vol. 48, pp. 1460-1473, 2017.

[39] J. Yang and J. Leskovec."Patterns of temporal variation in online media," *presented at the Proceedings of the Fourth ACM International Conference on Web Search and Data mining*, 2011, pp. 177-186.

[40] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research.* vol. 7, pp. 1-30, 2006.

یک ترکیب نوآورانه از قطعه‌بندی، خوشه‌بندی ترکیبی و الگوریتم ژنتیک به‌منظور خوشه‌بندی سری‌های زمانی

علی قربانیان[1,*] و زهرا قربانی[2]

[1] گروه مهندسی صنایع، مجتمع آموزش عالی فنی و مهندسی اسفراین، اسفراین، ایران.

[2] ادینبرا بیزنس اسکول، دانشگاه هریوت وات، ادینبرا، اسکاتلند.

ارسال ۲۰۲۴/۰۲/۱۲؛ بازنگری ۲۰۲۴/۰۳/۲۹؛ پذیرش ۲۰۲۴/۰۵/۰۲

چکیده:

افزایش دقت خوشه‌بندی سری‌های زمانی هم‌زمان با کاهش زمان اجرا یکی از چالش‌های اصلی در حوزه خوشه‌بندی سری‌های زمانی می‌باشد. در سالیان اخیر پژوهشگران از رویکردهایی مانند توسعه معیارهای فاصله و کاهش ابعاد برای حل این چالش استفاده نموده‌اند. با این وجود یکی از مواردی که در پژوهش‌های پیشین کمتر موردتوجه قرار گرفته است استفاده از قطعه‌بندی و خوشه‌بندی ترکیبی می‌باشد به‌منظور حل این مسئله می‌باشد. یک الگوریتم بر مبنای انتخاب و ترکیب بهترین قطعات ایجادشده از یک مجموعه داده سری زمانی توسعه داده شده است. در گام اول یک مجموعه داده به قطعاتی با اندازه یکسان تقسیم می‌گردند، در گام دوم هر یک از قطعات ایجادشده با استفاده از الگوریتم سلسله مراتبی خوشه‌بندی می‌شوند. در گام سوم و اصلی یک الگوریتم ژنتیک قطعات مختلف را انتخاب می‌نماید و با استفاده از خوشه‌بندی ترکیبی، با یک دیگر ترکیب می‌نماید. نتیجه خوشه‌بندی قطعات انتخاب‌شده به‌عنوان خوشه‌بندی نهایی مجموعه داده انتخاب می‌گردد. در این گام یک معیار درونی خوشه‌بندی جواب‌های ایجادشده را ارزیابی و مرتب می‌نماید. الگوریتم ارائه‌شده روی ۸۲ مجموعه داده مختلف در ۱۰ تکرار اجرا شده‌است. نتایج الگوریتم ارائه‌شده نشان‌دهنده افزایش کارایی خوشه‌بندی به میزان ۳,۰۷ درصد و رسیدن به عدد ۶۷,۴۰ می‌باشد. نتایج حاصله با توجه طول سری زمانی و نوع مجموعه داده مورد ارزیابی قرار گرفته‌است. همچنین نتایج حاصله با استفاده از تست آماری با ۶ الگوریتم موجود در ادبیات نیز مورد ارزیابی قرار گرفته‌است.

کلمات کلیدی: خوشه‌بندی سری‌زمانی، خوشه‌بندی ترکیبی، قطعه‌بندی، الگوریتم ژنتیک.