



Research paper

Time Series Clustering based on Aggregation and Selection of Extracted Features

Ali Ghorbanian and Hamideh Razavi*

Department of Industrial Engineering, Ferdowsi University of Mashhad, Mashhad, Iran.

Article Info

Article History:

Received 07 May 2023

Revised 15 June 2023

Accepted 27 June 2023

DOI:10.22044/jadm.2023.13089.2449

Keywords:

Time series, Clustering, Feature extraction, Feature selection, Data mining.

*Corresponding author: h-razavi@um.ac.ir (H. Razavi).

Abstract

In time series clustering, features are typically extracted from the time series data and used for clustering instead of directly clustering the data. However, using the same set of features for all data sets may not be effective. To overcome this limitation, this study proposes a five-step algorithm that extracts a complete set of features for each dataset including both direct and indirect features. The algorithm then selects essential features for clustering using a genetic algorithm and internal clustering criteria. The final clustering is performed using a hierarchical clustering algorithm and the selected features. Results from applying the algorithm to 81 datasets indicate an average Rand index of 72.16%, with 38 of the 78 extracted features, on average, being selected for clustering. Statistical tests comparing this algorithm to four others in the literature confirm its effectiveness.

1. Introduction

A time series is a collection of observations of a variable taken at distinct moments, which is ordered chronologically. Time series data is widely used in engineering, economics, finance, and medicine [1]. Clustering is a data mining technique used in the time series area to arrange related time series in groups without having access to the information about those groups [2, 3]. The applications of time series clustering are veritably wide. Portfolio building, watershed administration and preservation, and cancer subtype distinguishing are examples of these applications [4]. It is also employed for pre-processing data to detect anomalies [5], predict [6], and classify [7-9] time series. There are two approaches for time series clustering. In the first approach, time series are used directly for clustering. Some examples of the first approach are The Use of specific distance criteria (DTW¹, LCSS²)[10-12]; multi-stage methods[13, 14], deep learning methods [15], and weight clustering

[16]. In the alternate approach, the extracted features from the time series such as the average, minimum, and maximum of the time series, are employed for final clustering. This approach is commonly used when the original time series data is too complex or too large to be used directly in clustering algorithms. By extracting the relevant features from the time series, the dimensionality of the data is reduced, making it easier to cluster [17]. The significance of aggregation and selecting the proper feature for clustering to cluster various types of time series was an issue that other studies had previously disregarded, which was examined in this work. We have shown that selecting the proper feature for the final time series clustering can be more effective. The outcomes of our research clearly imply that the suggested algorithms are more accurate than those available in the related literature.

The presented algorithms titled aggregation and extracted of features (ASEF) have been explained in the second section. Additionally, Section 3 displays research findings, discusses and analyzes the results, and paper is concluded with Section 4.

¹ Dynamic Time Warping² Longest Common Subsequence

1.1. Literature Review

Fulcher et al. highlighted using time series features in a wide range of mining tasks, including anomaly detection, motif finding, clustering, and classification [18]. Wang *et al.* directly extracted statistical features from time series to cluster lengthy time series [19]. In another research, Räsänen and colleagues used the statistical feature to cluster and discover new knowledge on electricity data [20]. Similarly, Fulcher and Jones classified time series using statistical features [21]. Wang *et al.* also used feature extraction to cluster multivariate time series [22]. Hyndman and colleagues have also utilized this method to cluster enormous datasets and identify abnormal patterns in time series [23]. An extensive list of features that could be derived directly from a time series was presented by Barandas and colleagues [24]. As demonstrated by a two-step method in the Manakova research, using time series features for clustering can noticeably reduce the required runtime for large time series. However, this approach leads to clustering inaccuracy compared to other methods [14]. In the research mentioned above, the researchers mostly concentrated on expanding the number of features rather than the structural features of the time series to improve clustering accuracy. The use of structural features from graph mapping for time series clustering has been performed in other studies, however. Lacasa proposed a natural visibility graph for mapping time series data. This methodology is implemented in energy waste [25], financial time series [26], and heart rate time series. Lacasa *et al.* developed a simple kind of graph named the horizontal visibility graph to lessen the runtime of the natural visibility graph [27]. Mapping a time series into a quantile graph is another method that Campanharo *et al.* presented [28]. Zou *et al.* conducted a comprehensive investigation on mapping time series into graphs and their applications [29]. In a novel method, Da Silva mapped time series on various graphs and then clustered the time series based on the features extracted from these graphs [30].

In the conducted studies, the researchers have tried to increase the accuracy of the final clustering by increasing the number of extracted features. One of the important issues in this field is that not all extracted features are necessarily useful for all datasets. In many cases, a feature that can be useful feature for clustering one dataset may act as a noise source for another dataset and reduce the accuracy of clustering. Therefore, useful features must be selected for each dataset. On the other hand, in most of the

conducted studies, only the extracted features have been used separately either directly or indirectly. However, aggregating these types of features with each other can cover the weaknesses of each feature. In Section two, with several examples and in detail, it has been explained why some features can be useful in one dataset and not useful in another dataset.

Feature selection is an important step in data pre-processing, which is a critical component of data science. Feature selection refers to the process of selecting a subset of relevant features from a dataset, while discarding the irrelevant or redundant ones. The primary goal of feature selection is to improve the performance of a machine learning model by reducing the dimensionality of the input data and removing noise or irrelevant information [31, 32]. In many different types of studies, it has been utilized to improve clustering and classification performance [33-35]. Since the number of features is reduced, the runtime of the algorithms, particularly in large datasets, is also decreased. Accuracy may also improve when noisy features are eliminated [36]. Different algorithms, including metaheuristic algorithms [37, 38] and greedy algorithms [36, 39], have been utilized to achieve this goal. According to Deniz and colleagues, feature selection performed much better using metaheuristic algorithms like genetic algorithms [36]. According to the importance of feature selection in data science, this technique has not yet been implemented to select the proper features for clustering time series data sets. Based on the studies mentioned above, a wide range of features is utilized to cluster time series. In the previous research, a particular set of features have been employed to cluster various time series. This research has proved that not all time series dataset features are useful for clustering time series datasets. Furthermore, we have demonstrated that a particular set of features, which may be entirely different from another data set, is required to improve the accuracy of clustering in the time series of each dataset.

Based on this, an algorithm is developed with five main steps based on feature extraction, aggregation, and selection. The suggested algorithm is explained in the second section. Additionally, the features extracted directly and indirectly as well as the genetic algorithm for feature selection, are explained in detail in this section. Next, the dataset and the experiment results are examined in the third section. Finally, by using Wilkinson statistical test, we compared

our results to the four standout algorithms in the relevant literature.

2. Proposed Method: Aggregation and Selection of Extracted Features (ASEF)

The researchers have found that one method for clustering a dataset is to extract the features from the time series. Numerous features can be extracted from a time series dataset, however, not all of them improve clustering performance. To identify the most relevant features for clustering, researchers often use feature selection techniques. These techniques evaluate the relevance of each feature and select the subset of features that is most informative for the clustering task. When clustering one particular dataset, some features may be helpful, but the same features may adversely impact another dataset. This is due to the dependence of the features on the structure of the time series. As a result, one of the most effective ways to enhance clustering performance is to use features exclusive to each time series dataset. Figure 1 illustrates an example of a time series dataset consisting of two clusters, each having similar features, including the average, the minimum, and the maximum. As a result, the clustering performance may be reduced when the mentioned features, such as noise variables, are extracted. However, the structure of this dataset allows for the creation two unique graphs based on their mapping. As a result, the features extracted from these graphs could significantly enhance clustering performance.

Figure 1 b in the research paper presents another dataset for time series clustering, consisting of two distinct clusters. Unlike the dataset in Figure 1 a, statistical features are effective in clustering this dataset. This highlights the importance of selecting appropriate features for time series clustering, as the effectiveness of different feature sets can vary greatly depending on the specific characteristics of the time series data. However, as mentioned in the research paper, extracting features from graphs such as horizontal and natural visibility graphs may produce similar graphs, which can lead to the extraction of redundant or irrelevant features. These features can act as noise variables and reduce the overall performance of the clustering algorithm.

In light of the aforementioned information, selecting the appropriate features for a particular dataset may enhance its clustering performance. An algorithm based on feature selection and aggregation has been developed considering the mentioned cases.

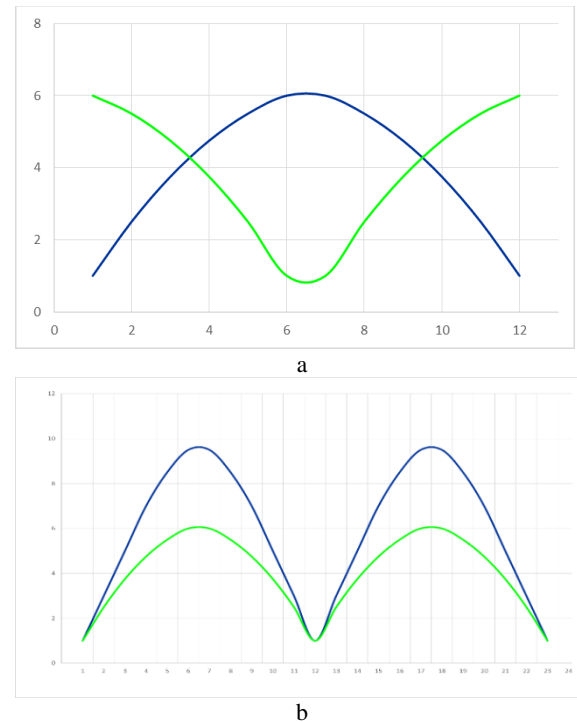


Figure 1. Effect of correct feature selection on clustering:
(a) Statistical features (average, min, max) are noise (b)
Graph features are noise.

One of the important features of this algorithm is that, unlike previous studies, it uses statistical and graph features simultaneously. Also, in this algorithm, unlike previous studies, a set of specific features is selected for each dataset using a genetic feature selection algorithm. This leads to the removal of redundant features and improves the accuracy of the final clustering. Figure 2 shows the general flowchart of the developed algorithm. Each step of the algorithm is indicated with a different color in Figure 2.

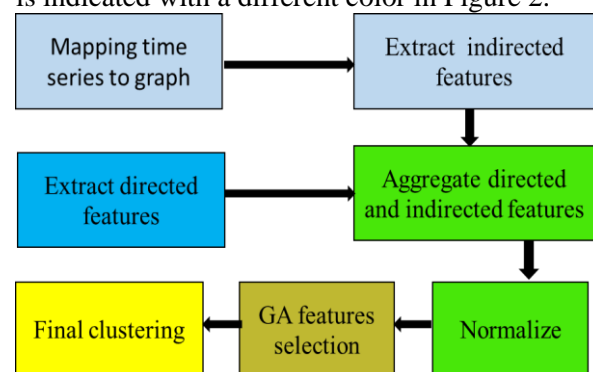


Figure 2. Flowchart of the proposed algorithm.

Therefore, an algorithm comprised of aggregation and relevant feature selection is developed based on five main steps, as below.

Step 1: The time series features are extracted indirectly by mapping the time series into quantile and visibility graphs. Five different features consisting of the average degree, mean

route length, number of communities, and density for each of the graphs, are calculated. At the end of this step, 25 features would be extracted indirectly for each time series.

Step 2: In contrast to the first phase, values are extracted directly from each time series in this stage. A particular set of features based on previous research is considered for this.

Step 3: The 78 features extracted in the first and second phases are aggregated and normalized. Because each feature's range varies, all extracted features must be normalized in this stage.

Step 4: The genetic algorithm selects relevant features to increase the overall clustering performance. The internal clustering measures have also been applied to the genetic algorithm evaluation function. Considering that the mentioned function is evaluated independently for every dataset, it is feasible for each dataset to select the relevant features of that dataset to enhance clustering performance.

Step 5: The final clustering is performed by applying the features selected in the previous stage and the hierarchical clustering algorithm. Also, the presented algorithm has been evaluated using Rand's external index.

The extracted features, visibility graph, quantile graph, aggregation and normalization, genetic algorithm for feature selection, internal and external criteria, and hierarchical algorithm for final clustering are all explained in the following sub-sections.

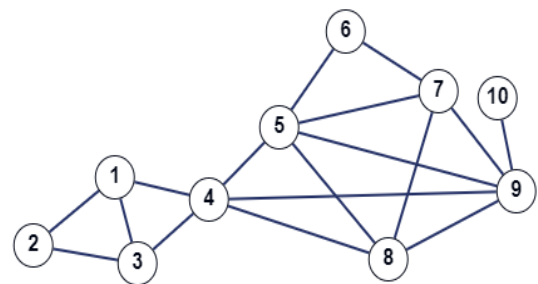
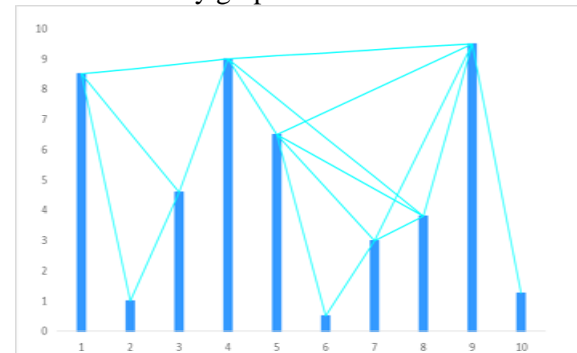
2.1. Feature extraction

According to the presented algorithm in this research, two types of features are extracted from time series. The first type consists of features indirectly extracted from time series. Two kinds of visibility and quantile graphs are used to map and extract the features. The features will be extracted from two varieties of visibility graphs, the natural and the horizontal graphs, and three different quantile graphs (10, 50, and 100).

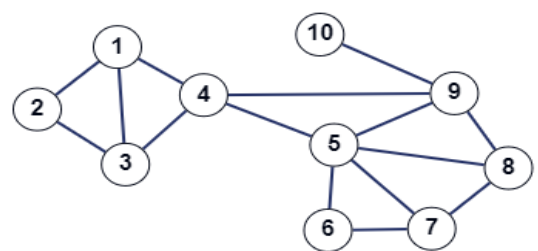
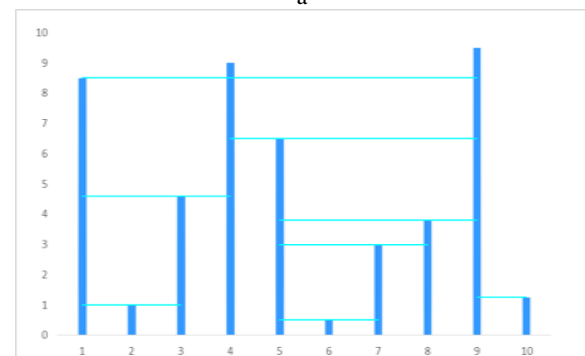
2.1.1. Visibility graph

The fundamental principle of these graphs is that there is no obstruction between two connection points. There are two varieties of natural visibility graph (NVG) and horizontal visibility (HVG) graph based on the type of relationship between two values in a time series, which can be horizontal or diagonal. These graphs may be weighted or unweighted but without a direction. This study has considered the weighted version of these graphs. Figure 3 shows how to map a time series into an NVG, and Figure 3 b exhibits how

to map a time series into an HVG. The slope of each edge indicates the weight of each edge in the natural visibility graph.



a



b

Figure 3. Mapping the time series into visibility graphs:
(a) The natural visibility graph and (b) The horizontal visibility graph.

2.1.2. Quantile graph

To create a quantile graph, The first step is to segment a time series into n pieces along the vertical axis. These pieces will function as a node in the mapped graphs. The subsequent stage is drawing a directed and weighted graph according to the number of times each item is passed on to

the next item. A quantile graph with three n values of 10, 50, and 100 is used in this study. Figure 4 illustrates the mapping of a time series into a quantile graph with $n = 5$.

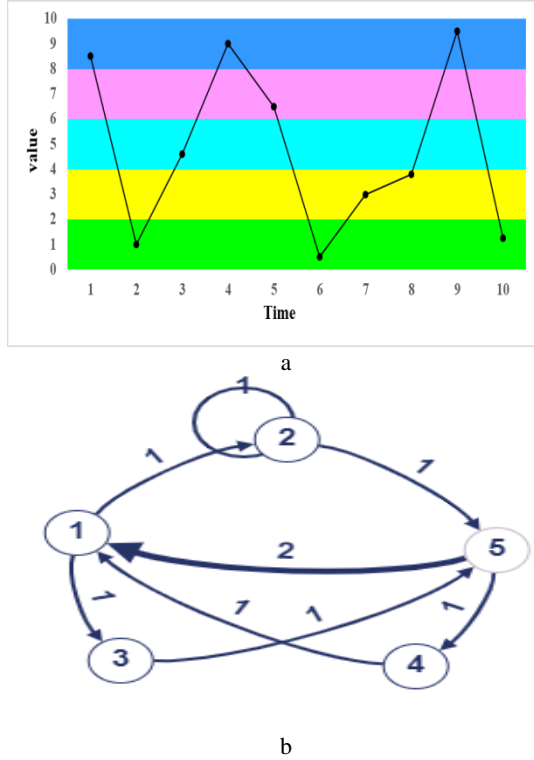


Figure 4. Mapping the time series into a quantile graph (Q5). (a) Time series. (b) Quantile graphs.

2.1.3. Features extracted from graphs

This study extracts the mean degree, mean route length, clustering degree, number of communities, and density from each dataset.

The degree determines the number of nodes connected to a single node. This feature is calculated differently, considering the graph is weighted, unweighted, directional or unidirectional. The route length is the shortest route length between all the graph nodes. A route is a string of interconnected nodes between two nodes in a graph. The clustering degree is a set of three nodes connected in a directionless graph using three edges called a triple combination. The ratio of the available triple combinations to the possible triple combination (the hypothetical complete graph) is known as the clustering degree [40]. Figure 5 depicts multiple sample graphs with different clustering degrees alongside their clustering degree calculation method.

A number of communities indicate the number of sub-graphs created using different algorithms based on the internal density of these sub-graphs and the density between them [41]. Density represents the graph density based on the number

of created subgraphs, which determines the number of edges inside communities to the number of edges between communities.

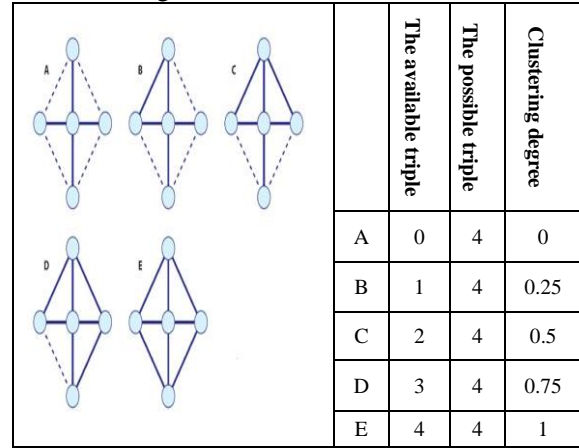


Figure 5. Multiple sample graphs with different clustering degrees.

The density value can be determined by Equation 1 [42]. Here, A_{ij} is the edge weight between the i and j nodes, k_i depicts the degree of node i , c_i determines the community of node i , and m is the total number of graph edges. Furthermore, $\delta(c_i, c_j)$ is a binary value equal to 1 if $c_i = c_j$, and equal to 0 in any other situation.

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \quad (1)$$

2.2. Direct features

These features are extracted directly from the time series. 53 features including average, variance, entropy, lumpiness, stability, maximum level shift, maximum variance shift, crossing points, autocorrelation, etc. that were taken from prior time series research are used in this work.

2.3. Aggregation and normalization

53 directly extracted features are aggregated, with 25 indirectly extracted ones in this part. Regarding the various feature extraction ranges, normalization is necessary. Equation 2 will transform all feature extraction ranges to $[0, 1]$. Here, x is the vector-based notation of an extracted feature.

$$x' = \frac{x - \min(x)}{\max x - \min x} \quad (2)$$

2.4. Feature selection

The metaheuristic genetic feature selection algorithm and internal and external criteria are described in this section.

2.4.1. Genetic feature selection algorithm

The genetic feature selection algorithm has been used to select the proper features. Each answer from this algorithm is represented by a chromosomal structure made up of genes with values of 0 and 1. Figure 6 depicts the chromosome in the genetic algorithm.



Figure 6. Chromosome in the genetic algorithm.

The crossovers between answers are made with one-point and two-point operators. Figure 7 depicts the one-point operation.

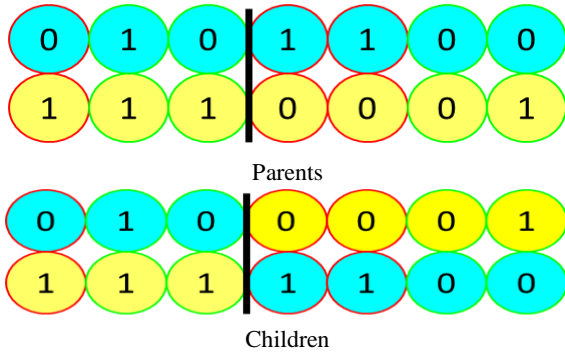


Figure 7. Crossover operation.

The mutation is another operation used to escape optimized local answers. It randomly selects a chromosome and reverses one of its genes, i.e. features. Figure 8 depicts this operation.

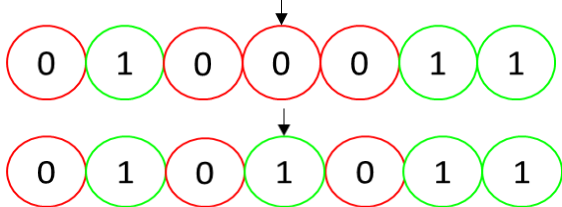


Figure 8. Mutation operation.

Considering that the labels of the objects in clustering are not revealed. As a result, the internal clustering measure is used in this study as the evaluation function for the genetic algorithm.

2.4.2. Internal measure

The object labels remain unknown because clustering, unlike classification, is an unsupervised process. Therefore, there is a need for measures capable of using the similarities inside clusters and the distinctions between them to evaluate the performance [43]. The proposed method uses the inter-group variance and Calinski-Harabasz criteria as the evaluation function for the genetic algorithm [43]. Table 1 depicts these measures, where n and k are the numbers of objects and clusters. At the same time,

x and y represent the specific objects inside the c cluster.

Table 1. Internal clustering evaluation measures.

Row	Measure	Symbol	Mathematical Equation	Optimal Value
1	Inter-group Variance	V	$\frac{1}{n-k} \sum_{i=1}^k \sum_{x \in c_i} dist(x, c_i)$	Min
2	Calinski-Harabasz	C H	$\frac{n-k}{n-1} \frac{\sum_{i=1}^k n_i dist^2(x, c_i)}{\sum_{i=1}^k \sum_{x \in c_i} dist^2(x, c_i)}$	Max

2.4.3. External criteria

The external criterion evaluation approach uses true labels to cluster all of the objects into different groups. These criteria evaluate the compatibility between the real and clustering-generated labels. The Rand Index shown in Equation 3 is used to evaluate the proposed algorithm's efficiency and compare it with the results of other studies.

$$RI = \frac{TP + TN}{TP + TN + FN + FP} \quad (3)$$

Here, TP is the number of objects with similar classes and clusters, and TN is the number of objects with different classes and clusters. On the other hand, FP shows the number of objects with different clustering and similar classes, and FN is the number of objects with similar clusters with different classes.

2.4.4. Hierarchical algorithm

As a final clustering, we will use agglomerative hierarchical clustering with complete linkage, and also, the distances between objects are considered Euclidean distances. Equations 4 and 5 are the complete distances in the hierarchical algorithm and Euclidean distance. In Equation 4, A and B are the clusters, a and b are the cluster members, and d is the distance between the two members. In Equation 5, X and Y are two points in an n -dimensional space.

$$CD = \max \{d(a, b) : a \in A, b \in B\} \quad (4)$$

$$ED(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (5)$$

3. Model Implementation

This model was implemented on 81 datasets from the UCR archive to analyze its efficiency. The intended datasets cover different subjects such as health, finances, media, and engineering. These datasets were presented for time series clustering

and contained two training and testing sections; therefore, they are aggregated for clustering [44]. The proposed algorithm is implemented for the two aforementioned internal measures with complete linkage for the final hierarchical algorithm clustering in both settings. We will use Inter-group Variance and Calinski-Harabasz in the first and second settings.

3.1. Configuring genetic algorithm

To increase the effectiveness of the proposed algorithm, the critical genetic algorithm parameters, such as iteration, population size, crossover rate, and mutation rate, were tuned using the Taguchi method. Table 2 consists of the Taguchi method levels with four variables.

Table 2: Taguchi method levels.

Parameters	Symbol	Levels
Crossover rate	P_c	0.4 – 0.6 – 0.8
Mutation rate	P_m	0.1 – 0.3 – 0.5
Iteration	It	20 – 30 – 50
Population size	Pop	100 – 200 – 300

These levels have created nine different states for each variable in the Taguchi method. Table 3 presents the average Rand Index values for the proposed method in 10 iterations for both settings on a single time series.

Table 3: Taguchi method and Rand Index values for the first and second states of the proposed algorithm.

Number of experiment	P_c	P_m	Pop	It	RI –first setting	RI –second setting
1	0.4	0.1	100	20	91.48%	91.17%
2	0.4	0.3	200	30	92.34%	91.84%
3	0.4	0.5	300	50	92.42%	92.31%
4	0.6	0.1	200	50	92.64%	92.73%
5	0.6	0.3	300	20	92.29%	92.48%
6	0.6	0.5	100	30	92.75%	92.21%
7	0.8	0.1	300	30	92.59%	92.28%
8	0.8	0.3	100	50	92.57%	92.32%
9	0.8	0.5	200	20	92.11%	92.00%

The genetic algorithm parameters were computed for both settings, considering the signal-to-noise ratio reported in Figure 9. The crossover rate of both settings is 0.6, their mutation rate is 0.5 and 0.3, and the initial population of both settings is 300 with 30 and 50 iterations.

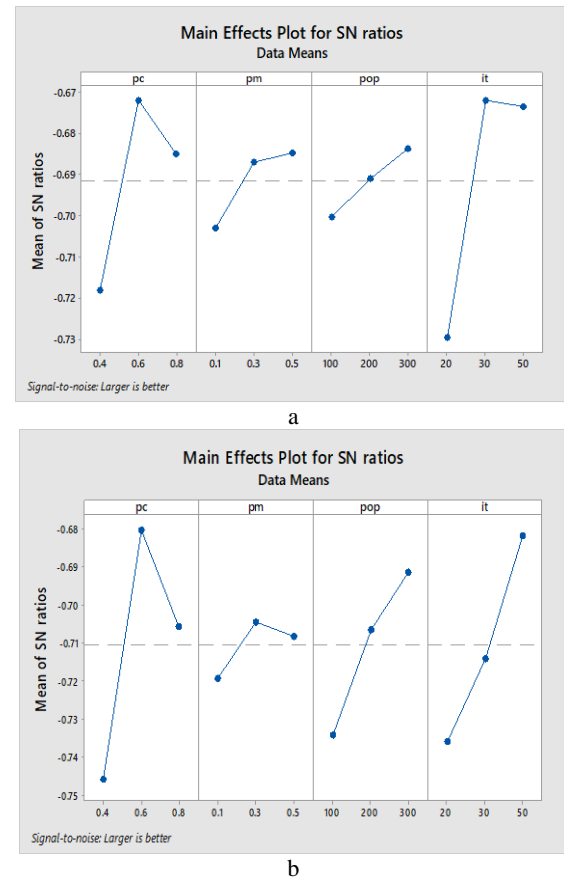


Figure 9. Signal-to-noise ratio a) First setting. b) Second setting.

3.2. Results

To evaluate the performance of the suggested algorithms, the results are compared for 81 datasets. The algorithm is run in ten subsequent iterations on each dataset based on the Taguchi method parameters. Figure 5 depicts the performance results of both algorithm settings in summary. The information from Table 4 and Table 7 shows that extracted feature aggregation for the final clustering leads to a 66% Rand index value. According to the information in Table 4 and Table 7, it can be found that if only the aggregation of the extracted features is used for the final clustering, the average value of the Rand index is almost 66%. The first and second settings increased the Rand index by 6.13 and 5.70 percent compared to the setting without feature selection. The Rand value in the first setting was the best answer for 50% of the datasets, while it was the best answer in 43.21% of the datasets in the second setting compared to the four previous algorithms. Furthermore, the first and second settings of the proposed algorithm improved the Rand index value in 75.31 and 76.54 percent of datasets.

Table 4. Statistical results of the proposed algorithm's first and second states.

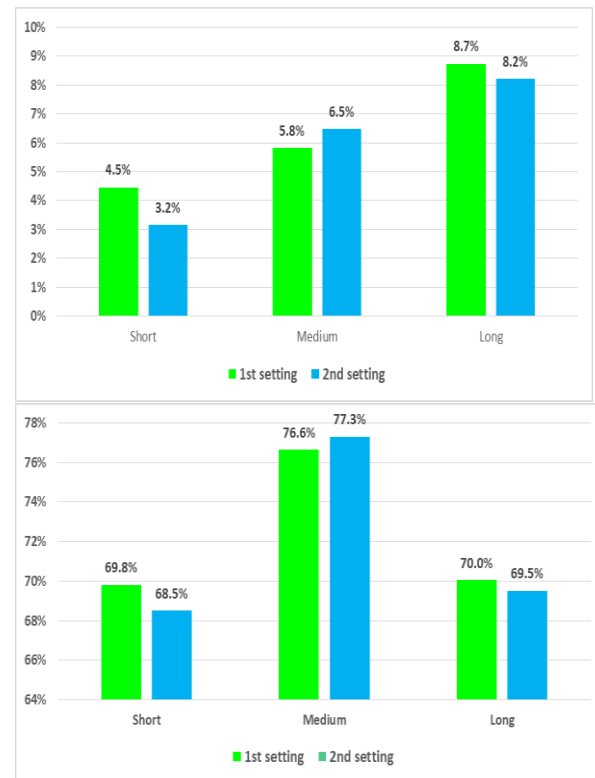
	First setting	Second setting
Average RI	72.16%	71.73%
Average improvement of the RI	6.13%	5.70%
Ratio of the best answer in previous research	49.38%	43.21%
Ratio of the dataset with the improvement of RI	75.31%	76.54%

Based on the data in Table 5, the first and second settings of the proposed algorithm selected an average of 38 features from the 78 features available. This indicates that more than half of these extracted features had no impact on the performance of the clustering. Furthermore, the minimum number of features selected by the first setting was 27, and the maximum was 46 for a specific data set. These two criteria for the second setting were equal to 26 and 51 features. It is clear from the analysis of the selected features in both settings that no feature was selected simultaneously across all data sets. Additionally, none of the features are selected at all. The presented algorithm selected the least used feature across 25 datasets in the first setting. This value was equal to 30 datasets in the second setting. Furthermore, the most used feature was selected by 52 datasets in the first setting and 51 datasets in the second setting.

Table 5. Statistical results of the selected features in both algorithm setting.

	First setting	Second setting
Average number of selected features	38	38
Minimum number of selected features	27	26
Mmaximum number of selected features	46	51
Number of the most used feature	52	51
Number of the least used feature	25	30

Considering that the datasets employed have varying lengths, a sensitivity analysis based on the length of each dataset has been performed to evaluate the efficiency of the proposed algorithm. The data sets are categorized into short, medium, and long classes based on their length. Figure 10 displays the value of the final Rand index and the improvement over the prior algorithm for the three stated classes.

**Figure 10. Rand index analysis based on length.**

As observed, the best final rand index for both algorithm settings belongs to the medium class. The long class, however, had seen the most significant increase in the rand index. The first setting of the proposed algorithm for the short and long class has a more rand index than the second setting, as seen in the figure. However, the second setting of the algorithm has performed better in the middle class, both in the Rand index and the improvement rate. Additionally, both settings of the algorithm's improvement value can be seen to have increased as the time series' length has increased. In general, the first setting of the algorithm is preferable for short and long time series, and the second setting is preferable for medium-length time series.

It should be noted that the first and second settings of the developed algorithm had poor performance in 17.28% and 16.05% of the datasets, respectively, which led to a decrease in the value of the Rand index (For both settings, the Rand index remained unchanged in 7.41% of the datasets). This performance can be due to two different reasons. The first reason is the weakness of the feature selection algorithm, which could not select suitable features for these datasets. The second reason may be the inappropriate internal criterion for these datasets. For example, in the OLI dataset, while the first setting of the algorithm decreased the Rand

index, the second setting increased it. It seems that using multi-objective feature selection algorithms in future research can somewhat reduce this problem.

3.3. Statistical analysis

In order to investigate the performance of the proposed algorithm more accurately, the Wilcoxon statistical test was used [45]. Both settings of the developed algorithm were compared with the initial state (without feature selection), and the table shows the p-value. Based on the p-value in Table 6, it can be seen that the first and second settings of the developed algorithm are better than the algorithm without feature selection. However, considering the p-value of 0.625 for comparing the two algorithm settings, it can be concluded that there is no significant difference between these two settings at 5% alpha significance level.

Table 6. Results of statistical tests of the presented algorithm.

	First setting	Second setting
without feature selection	0.000	0.000
First setting	-	0.625

Furthermore, both settings of the developed algorithm were compared with four selected algorithms in the literature based on the mentioned statistical test. The four selected algorithms are : DD_{DTW} which uses a mixture of two distance criteria with a hierarchical algorithm [12], KSC algorithm which uses specific time patterns for clustering [46]; alongside the $TS3C_{MV}$ and $TS3C_{CH}$ that have been developed based on features of time series segmentation [47]. According to the literature, the Rand index for the four mentioned algorithms, respectively, is as follows: 60.62, 60.15, 65.81, and 66.32. Table 7 displays complete information regarding the Rand index for the selected algorithm in the literature as well as two cases of the developed algorithm. The p-value for comparing the two conditions of the developed algorithm with the four selected algorithms is equal to 0.000. Considering a significant level of 5% for an alpha, can be clearly seen the higher efficiency of the proposed algorithm in terms of the Rand index compared with the four algorithms in the related literature.

Table 7. Rand index.

Dataset	Proposed algorithm			Previous algorithms			
	Without feature selection (%)	With feature selection		$TS3C_{CH}(\%)$	$TS3C_{MV}(\%)$	$DD_{DTW}(\%)$	KSC (%)
		First setting (%)	Second setting (%)				
ADI	88.2	91.0	90.9	92	92	68	95
ARR	69.6	69.1	67.7	62	63	35	63
BEE	62.9	72.1	70.3	68	68	58	71
BFL	67.2	73.8	77.6	49	49	59	50
BIR	55.0	85.8	77.6	49	49	50	54
CAR	64.8	69.7	64.7	65	65	50	68
CBF	72.4	93.0	92.3	67	67	78	56
CHL	41.7	50.3	49.5	49	47	40	53
CIN	71.3	81.6	85.2	64	64	56	69
COF	75.1	100.0	100.0	51	51	49	75
COM	50.0	54.5	50.3	50	51	50	50
CRX	81.8	86.4	85.9	85	85	78	41
CRY	81.9	82.0	84.5	84	84	69	53
CRZ	81.1	82.8	86.0	84	85	71	41
DIA	33.7	88.4	86.1	72	72	30	96
DPA	70.7	73.9	73.7	60	60	71	72
DPC	50.5	50.5	50.5	51	51	53	50
DPT	82.4	73.6	78.6	68	66	86	66
EAR	52.1	52.0	51.5	53	53	54	62
EC2	54.9	60.4	61.3	50	50	54	61
EC5	46.7	74.8	72.2	64	60	89	59
ECF	50.6	50.1	50.2	50	50	50	81
FAA	67.5	81.2	80.8	85	85	60	30
FAF	69.1	75.2	74.6	57	57	55	38
FIS	68.3	74.6	75.8	73	64	18	79
FOA	57.2	57.8	57.4	52	51	54	50
FOB	56.6	58.7	59.8	50	50	50	50
GUN	49.9	54.3	49.7	54	54	50	51
HAM	49.8	49.8	50.0	52	52	50	53
HAP	51.3	59.7	60.5	60	60	39	69
HER	50.0	49.9	50.2	50	50	51	50
INL	60.2	75.1	73.6	71	71	54	74
INS	76.1	81.4	80.9	81	81	55	69
ITA	50.1	50.1	50.5	50	50	51	64

Continued Table 7.

LAR	35.9	57.7	52.9	55	55	34	41
LI2	50.6	54.3	53.3	50	54	50	50
LI7	76.7	81.1	80.0	75	75	60	59
MAL	90.3	96.1	95.8	80	80	93	92
MEA	75.4	81.1	80.2	71	40	77	76
MED	64.7	68.4	67.5	65	65	64	47
MPA	73.1	72.9	72.9	56	56	73	73
MPC	50.0	50.0	50.0	51	51	50	50
MPT	84.0	78.2	78.5	74	82	80	81
MOT	50.3	51.0	51.2	50	50	50	58
NO1	90.5	94.5	94.5	94	95	70	95
NO2	93.5	95.5	95.5	95	95	85	97
OLI	66.8	61.2	67.6	77	77	76	85
OSU	72.6	80.9	82.9	73	73	62	29
PHA	50.0	50.0	50.0	51	51	54	51
PHO	92.0	92.2	92.4	93	93	45	51
PLA	100.0	100.0	100.0	83	80	100	92
PPA	79.7	79.1	80.2	76	76	78	76
PPC	52.1	52.1	52.1	56	56	54	53
PPT	85.0	77.5	77.5	78	78	88	81
REF	37.2	51.3	53.5	56	54	35	39
SCR	35.4	52.3	48.9	53	53	35	45
SHS	88.7	83.5	96.1	99	99	50	50
SHA	96.1	97.1	97.2	97	97	84	63
SMA	43.1	60.7	61.8	59	59	34	54
SO1	79.7	89.6	88.8	51	52	50	75
SO2	78.8	82.3	94.4	60	53	53	66
STR	52.8	51.8	50.1	50	52	50	50
SWE	88.8	92.3	93.5	88	88	35	63
SYM	89.1	94.6	93.9	81	81	89	60
SYN	87.6	95.9	90.7	78	78	88	38
TO1	49.9	50.2	50.3	51	51	51	53
TO2	68.0	50.2	51.8	50	50	67	53
TRA	92.6	98.1	99.5	84	84	87	72
TWP	49.2	61.2	62.0	64	64	85	46
TWE	51.2	96.3	50.0	64	64	50	54
UWX	66.9	79.8	79.6	78	75	80	51
UWY	75.8	81.2	80.3	78	76	82	54
UWZ	67.0	78.3	81.5	80	80	74	54
UWA	63.1	77.9	77.1	76	76	59	45
W50	93.8	94.2	94.3	94	94	92	66
WAF	73.9	62.0	63.1	50	66	53	59
WIN	49.6	50.7	51.0	57	50	50	59
WOS	82.5	87.7	88.0	87	87	87	50
WOR	42.7	63.9	63.1	60	58	62	53
WOT	50.7	54.2	54.2	51	51	50	50
YOG	49.9	50.0	50.0	51	50	50	50
Average	66.0	72.2	71.7	66.32	65.81	60.62	60.15

4. Conclusion

This paper presented a feature aggregation and selection algorithm for time series clustering using a metaheuristic genetic algorithm. First, the features are extracted from time series using direct methods and mapping into different graphs. The appropriate features that can improve the clustering efficiency are then selected for each dataset applying the inter-group variance and Calinski-Harabasz criteria as the evaluation function for the genetic algorithm. The average Rand index on 81 datasets in ten repetitions for the first and second settings of the developed algorithm is 72.2% and 71.7%, respectively. To analyze the efficiency of the proposed method, the value of the average Rand index of the first algorithm setting is compared with DD_{DTW} , KSC,

$TS3C_{MV}$, and $TS3C_{CH}$ algorithms using the statistical Wilcoxon test. The results clearly indicate the superior performance of the suggested algorithm. Furthermore, two different settings of the proposed algorithm showed that on average, only 38 of 72 extracted features are effective in the clustering process. In order to demonstrate the better performance of the proposed algorithm, a percentage of the dataset has been examined, which had the best possible solution among the developed algorithm and the four selected algorithms in the literature. In the first case, the developed algorithm had the best solution among the selected algorithms in 40 out of 81 datasets (49.4%), which is a relatively high percentage and shows the good performance of the developed algorithm. Also, this number for the second case is 35 out of 81 datasets (43.21%). While these percentages for the four selected algorithms are

12.35%, 16.05%, 14.8%, and 20.99%, respectively, which indicates a significant difference between the developed algorithm and the existing algorithms (since multiple algorithms may have the best solution simultaneously, the sum of the percentages may be greater than 100).

References

- [1] W. Lin, M. A. Orgun, and G. J. Williams, "An Overview of Temporal Data Mining," in *AusDM*, pp. 83-90, 2002.
- [2] P. Rai and S. Singh, "A survey of clustering techniques," *International Journal of Computer Applications*, Vol. 7, No. 12, pp. 1-5, Art No. 2010.
- [3] A. Ghorbanian and H. Razavi, "A new method based on ensemble time series for fast and accurate clustering," *Data Technologies and Applications*. 2023.
- [4] A. Javed, B. S. Lee, and D. M. Rizzo, "A benchmark study on time series clustering," *Machine Learning with Applications*, vol. 1, p. 100001, Art No. 2020.
- [5] A. Ghorbanian and H. Razavi, "A novel two-level clustering algorithm for time series group forecasting," *Journal of Industrial and Systems Engineering*. 2023.
- [6] D. Graves and W. Pedrycz, "Proximity fuzzy clustering and its application to time series clustering and prediction," in *2010 10th International Conference on Intelligent Systems Design and Applications: IEEE*, pp. 49-54, 2010.
- [7] S. Datta, S. Rokade, and S. P. Rajput, "Classification of uncontrolled intersections using hierarchical clustering," *Arabian Journal for Science and Engineering*, vol. 45, no. 10, pp. 8591-8606, Art No. 2020.
- [8] H. Koosha, Z. Ghorbani, and R. Nikfetrat, "A Clustering-Classification Recommender System based on Firefly Algorithm," *Journal of AI and Data Mining*, vol. 10, no. 1, pp. 103-116, Art No. 2022.
- [9] N. Esfandian, F. Jahani Bahnamiri, and S. Mavaddati, "Voice activity detection using clustering-based method in Spectro-Temporal features space," *Journal of AI and Data Mining*, vol. 10, no. 3, pp. 401-409, Art No. 2022.
- [10] G. Soleimani and M. Abessi, "DLCSS: A new similarity measure for time series data mining," *Engineering Applications of Artificial Intelligence*, vol. 92, p. 103664, Art No. 2020.
- [11] 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, no. 9, pp. 1192-1198, Art No. 2013.
- [12] M. Łuczak, "Hierarchical clustering of time series data with parametric derivative dynamic time warping," *Expert Systems with Applications*, vol. 62, pp. 116-130, Art No. 2016.
- [13] X. Zhang, J. Liu, Y. Du, and T. Lv, "A novel clustering method on time series data," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11891-11900, Art No. 2011.
- [14] N. Manakova and V. Tkachenko, "Two-stage time-series clustering approach under reducing time cost requirement," in *2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET): IEEE*, pp. 653-658, 2020.
- [15] N. Tavakoli, S. Siami-Namini, M. Adl Khanghah, F. Mirza Soltani, and A. Siami Namin, "An autoencoder-based deep learning approach for clustering time series data," *SN Applied Sciences*, vol. 2, pp. 1-25, Art No. 2020.
- [16] S. Mehrmolaei and M. R. Keyvanpour, "A comparative study on weighting-based clustering techniques: Time series data," in *2018 8th Conference of AI & Robotics and 10th RoboCup Iranopen International Symposium (IRANOPEN): IEEE*, pp. 65-72, 2018.
- [17] S. Zolhavarieh, S. Aghabozorgi, and Y. W. Teh, "A review of subsequence time series clustering," *The Scientific World Journal*, vol. 2014. 2014.
- [18] B. D. Fulcher, "Feature-based time-series analysis," *arXiv preprint arXiv:1709.08055*. 2017.
- [19] X. Wang, K. Smith, and R. Hyndman, "Characteristic-based clustering for time series data," *Data mining and knowledge Discovery*, vol. 13, no. 3, pp. 335-364, Art No. 2006.
- [20] T. Räsänen and M. Kolehmainen, "Feature-based clustering for electricity use time series data," in *International conference on adaptive and natural computing algorithms: Springer*, pp. 401-412, 2009.
- [21] B. D. Fulcher and N. S. Jones, "Highly comparative feature-based time-series classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 12, pp. 3026-3037, Art No. 2014.
- [22] X. Wang, A. Wirth, and L. Wang, "Structure-based statistical features and multivariate time series clustering," in *Seventh IEEE international conference on data mining (ICDM 2007): IEEE*, pp. 351-360, 2007.
- [23] R. J. Hyndman, E. Wang, and N. Laptev, "Large-scale unusual time series detection," in *2015 IEEE international conference on data mining workshop (ICDMW): IEEE*, pp. 1616-1619, 2015.
- [24] M. Barandas *et al.*, "TSFEL: Time series feature extraction library," *SoftwareX*, vol. 11, p. 100456, Art No. 2020.
- [25] C. Liu, W.-X. Zhou, and W.-K. Yuan, "Statistical properties of visibility graph of energy dissipation rates in three-dimensional fully developed turbulence,"

Physica A: Statistical Mechanics and its Applications, vol. 389, no. 13, pp. 2675-2681, Art No. 2010.

[26] E. Zhuang, M. Small, and G. Feng, "Time series analysis of the developed financial markets' integration using visibility graphs," *Physica A: Statistical Mechanics and its Applications*, vol. 410, pp. 483-495, Art No. 2014.

[27] B. Luque, L. Lacasa, F. Ballesteros, and J. Luque, "Horizontal visibility graphs: Exact results for random time series," *Physical Review E*, vol. 80, no. 4, p. 046103, Art No. 2009.

[28] A. S. Campanharo, M. I. Sirer, R. D. Malmgren, F. M. Ramos, and L. A. N. Amaral, "Duality between time series and networks," *PloS one*, vol. 6, no. 8, p. e23378, Art No. 2011.

[29] 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, Art No. 2019.

[30] V. A. F. da Silva, "Time Series Analysis based on Complex Networks." 2018.

[31] H. Liu and H. Motoda (2012) *Feature selection for knowledge discovery and data mining*. Springer Science & Business Media.

[32] S. Hosseini and M. Khorashadizade, "Efficient Feature Selection Method using Binary Teaching-learning-based Optimization Algorithm," *Journal of AI and Data Mining*, vol. 11, no. 1, pp. 29-37, Art No. 2023.

[33] M. A. Khan *et al.*, "A fused heterogeneous deep neural network and robust feature selection framework for human actions recognition," *Arabian Journal for Science and Engineering*, pp. 1-16, Art No. 2021.

[34] G. Ansari, T. Ahmad, and M. N. Doja, "Hybrid filter-wrapper feature selection method for sentiment classification," *Arabian Journal for Science and Engineering*, vol. 44, no. 11, pp. 9191-9208, Art No. 2019.

[35] R. Lamba, T. Gulati, and A. Jain, "A Hybrid Feature Selection Approach for Parkinson's Detection Based on Mutual Information Gain and Recursive Feature Elimination," *Arabian Journal for Science and Engineering*, vol. 47, no. 8, pp. 10263-10276, Art No. 2022.

[36] A. Deniz, H. E. Kiziloz, T. Dokeroglu, and A. Cosar, "Robust multiobjective evolutionary feature subset selection algorithm for binary classification using machine learning techniques," *Neurocomputing*, vol. 241, pp. 128-146, Art No. 2017.

[37] C. Wang, Y. Huang, M. Shao, Q. Hu, and D. Chen, "Feature selection based on neighborhood self-information," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 4031-4042, Art No. 2019.

[38] I. Tsamardinos, G. Borboudakis, P. Katsogridakis, P. Pratikakis, and V. Christophides, "A greedy feature selection algorithm for Big Data of high dimensionality," *Machine learning*, vol. 108, no. 2, pp. 149-202, Art No. 2019.

[39] E.-G. Talbi, L. Jourdan, J. Garcia-Nieto, and E. Alba, "Comparison of population based metaheuristics for feature selection: Application to microarray data classification," in *2008 IEEE/ACS International Conference on Computer Systems and Applications*: IEEE, pp. 45-52, 2008.

[40] L. Ostroumova Prokhorenkova, "Global clustering coefficient in scale-free weighted and unweighted networks," *Internet Mathematics*, vol. 12, no. 1-2, pp. 54-67, Art No. 2016.

[41] S. Kumar, B. Panda, and D. Aggarwal, "Community detection in complex networks using network embedding and gravitational search algorithm," *Journal of Intelligent Information Systems*, vol. 57, no. 1, pp. 51-72, Art No. 2021.

[42] A. Ghorbanian and M. Neyestani, "A New Approach to Community Detection in Complex Networks by Using Memetic Algorithms," *Advances in Modelling and Analysis A*, vol. 54, no. 3, pp. 384-406, Art No. 2017.

[43] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *2010 IEEE international conference on data mining*: IEEE, pp. 911-916, 2010.

[44] H. A. Dau *et al.*, "The UCR time series archive," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, pp. 1293-1305, Art No. 2019.

[45] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1-30, Art No. 2006.

[46] J. Yang and J. Leskovec, "Patterns of temporal variation in online media," in *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 177-186, 2011.

[47] 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, no. 11, pp. 5409-5422, Art No. 2020.

خوشه‌بندی سری‌های زمانی بر مبنای تجمیع و انتخاب مشخصه

علی قربانیان و حمیده رضوی*

گروه مهندسی صنایع، دانشگاه فردوسی مشهد، مشهد، ایران.

ارسال ۲۰۲۳/۰۵/۰۷؛ بازنگری ۲۰۲۳/۰۶/۱۵؛ پذیرش ۲۰۲۳/۰۶/۲۷

چکیده:

یکی از رویکردهای اصلی خوشه‌بندی سری‌های زمانی استفاده از مشخصه‌های مستخرج شده از سری‌های زمانی می‌باشد در این پژوهش نشان داده شده است که استفاده از مجموعه‌ای مشخص و ثابت از مشخصه‌ها برای همه مجموعه‌داده‌ها نمی‌تواند کارایی لازم برای خوشه‌بندی نهایی را داشته باشد. لذا یک الگوریتم بر پایه تجمیع و انتخاب مشخصه در پنج گام به‌منظور خوشه‌بندی ارائه شده است. در الگوریتم ارائه‌شده ابتدا مجموعه‌ای کامل از مشخصه‌ها به‌صورت مستقیم و غیرمستقیم برای هر مجموعه‌داده‌ها استخراج شده است. بر خلاف سایر پژوهش‌ها در این پژوهش به‌طور هم‌زمان مشخصه‌ها مستقیم و غیرمستقیم مورد استفاده قرار گرفته است. در گام بعدی با استفاده از یک الگوریتم ژنتیک و با استفاده از معیارهای داخلی خوشه‌بندی مختلف برای هر مجموعه‌داده، مشخصه‌های مفید به‌منظور خوشه‌بندی نهایی انتخاب شده است. در نهایت با استفاده از الگوریتم خوشه‌بندی سلسله‌مراتبی و مشخصه‌های منتخب خوشه‌بندی نهایی برای هر مجموعه‌داده انجام شده است. نتایج حاصله از اجرای الگوریتم ارائه‌شده روی ۸۱ مجموعه‌داده به‌وضوح نشان‌دهنده کارایی الگوریتم فوق در بهبود شاخص بیرونی رند می‌باشد به طوری که مقدار این شاخص به‌طور متوسط برابر با ۷۲٫۱۶ درصد می‌باشد. همچنین الگوریتم ارائه‌شده نشان‌دهنده که از میان ۷۸ مشخصه استخراج‌شده به‌طور متوسط ۳۸ مشخصه به‌عنوان مشخصه‌های نهایی جهت خوشه‌بندی نهایی انتخاب شده‌اند. نتایج حاصله از مقایسه الگوریتم ارائه شده با ۴ الگوریتم موجود در ادبیات با استفاده از آزمون آماری نیز بیان‌گر عملکرد بسیار خوب این الگوریتم می‌باشد.

کلمات کلیدی: سری‌های زمانی، خوشه‌بندی، استخراج مشخصه، انتخاب مشخصه، داده‌کاوی.