IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 2 February 2024, accepted 5 March 2024, date of publication 18 March 2024, date of current version 22 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3377656

RESEARCH ARTICLE

Time Series Overlapping Clustering Based on Link Community Detection

YASAMIN GHAHREMANI AND BABAK AMIRI
School of Industrial Engineering, Iran University of Science and Technology, Tehran 16846-13114, Iran
Corresponding author: Babak Amiri (babakamiri@iust.ac.ir)

ABSTRACT Given the nature of time series and their vast applications, it is essential to find clustering algorithms that depict their real-life properties. Among the features that can hugely effect the options available for time series are overlapping and hierarchical properties. In this paper a novel approach to analyze time series with such features is introduced. Using the two concepts of network construction and link community detection, we have attempted to analyze and identify the mentioned properties of time series using data that is often gathered first hand. The proposed algorithm has been applied using both recent and common similarity measures on ten synthetic time series with hierarchal and overlapping features, alongside various distance measures. When testing the proposed approach, the element-centric measure of similarity indicated a clear increased accuracy for this algorithm, showing the highest accuracy when used alongside the Dynamic Time Warping distance measure. Moreover, the proposed algorithm has been very successful in identifying and forming communities for both large and small time series, thus solving another one of the main issues previous algorithms tended to have.

INDEX TERMS Network science, community detection, time series, machine learning, dynamic time warping.

I. INTRODUCTION

Today, a vast amount of data in various fields in science, such as economics [43], business [1], [26], healthcare [8], [59] and social sciences [38], is captured in the form of time series [29], [34]. Each time series, while consisting of a sequence of temporal data, can also be seen as a single object, often showing similar properties to other time series [63]. Clustering is a data mining technique which involves identifying similar features between various data series and placing them into homogenous or related groups, also known as clusters. The basis for these clusters is grouping data series with maximum inter-similarity and minimum intra-similarity. By using this technique, more knowledge can be gained regarding the datasets and therefore extracting results from the data can become more easy. Following the definition of clustering, time series clustering is a technique that its' objective is to partition time series data into clusters, groups containing time series with the maximum measure of similarity.

Time series clustering is a challenging task, for it usually includes data with high dimensions, and often large volumes and demands minimum attention to the noise that is often found in the time series, and a detailed analysis of the patterns within a time series [2]. Due to the importance of time series, and the valuable information that can be revealed through clustering such datasets, there has been an emerging interest in developing different clustering techniques [33]. Although previous methods all show a desirable accuracy for well-known datasets, many fail to capture some important features that often accompany realistic time series. While it is common practice to classify nodes to a single cluster, many real life datasets show a tendency to contribute to various clusters within their dataset, and while many of the current clustering methods work well when clusters are well separated, many break down when clusters overlap or intersect [17]. This leads to the important issue of overlapping in time series, an issue that must be considered when

The associate editor coordinating the review of this manuscript and approving it for publication was Zhigao Zheng.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

approaching and analyzing data gathered from the real world. In addition to overlapping, datasets of a hierarchical nature allow us to see beyond the macro scale of communities, and thus analyzing these two aspects together, helps us better understand datasets gathered in real life [49].

An important research domain for studying time series is performed through complex network analysis, i.e. the complex network construction of nodes and links between data samples [16], [58], [60], [69]. This method has shown promising results, as community detection methods have shown to accurately reflect structural properties and temporal changes in these datasets [5]. In this context, community detection techniques are often based on the concept of having unique nodes and attempt to detect communities in time series by clustering the nodes in the datasets. While this may be true for some datasets, it is often seen in real life that the relationships between the nodes are more unique, and therefore, in order to gain better understanding of time series, unique links must be the focus of our community detection and clusters must be defined by the membership of links within a network of time series.

In this paper, we propose the technique of link community detection for overlapping and hierarchical time series with networks formed by the K-Nearest Neighbor analysis of time series. In section II, the previous literature is reviewed. The next sections explain the steps in our proposed technique and the datasets respectfully. Finally, we discuss the results of applying the method to the datasets.

II. RELATED WORKS

Previous research has classified clustering methods into six main categories; Partitioning, Hierarchical, Grid-based, Model-based, Density-based clustering and Multi-step clustering algorithms [51], [55], [64]. Hierarchical clustering involves building a hierarchy of clusters and can be done using agglomerative or divisive algorithms [9]. Among common agglomerative algorithms are Single linkage, Average linkage, Complete Linkage and Ward's method [44], and DIANA and bisecting k-means are among the divisive algorithms [12], [28]. Hierarchical algorithms tend to be weak in quality due to the fact that they cannot adjust clusters after splitting or merging a cluster. In order to tackle this issue, hierarchical algorithms are often combined with hybrid clustering approaches. Partitioning clustering methods partition n unlabeled objects into k clusters [2], [55]. Among some of the well-known partitional algorithms are k-means and k-medoids. Compared to hierarchical algorithms, these algorithms are fast and are commonly used for time series clustering among researchers [15], [16], [17]. Since both k-means and k-medoid algorithms define cluster membership in a "crisp" manner, the fuzzy c-means and c-medoid algorithms were introduced and later on developed for the purpose of time series clustering [18], [20], [22], [23]. Model based clustering, is another category of time series clustering methods which was first mentioned and proposed by Wolfe and is based on the idea of defining clusters via a mixture model [40]. Since then, many researchers have contributed to this method of clustering [18], [19], [39], [41], [50]. This method has been often regarded as slow especially for large datasets and requires defining parameters that can contribute to the overall inaccuracy. Density based clustering algorithms are among other clustering methods and there have been many attempts for expanding them and implementing them in different fields but are often too complex to be applied for time series clustering [10], [25], [27]. Grid based and multi-step clustering methods are also known to not be useful for the analysis of real life datasets due to their complex and timely process.

The idea of applying network community detection methods for time series was first stated by Zhang [70]. What was proposed in the paper was to primarily build a 1-NN network based on the similarity between pairs of the time series. After choosing nodes having the large numbers of neighbors as candidates for the clustering, with the use of the DTW distance measure and hierarchical clustering method, each candidate node is clustered. Gao also proposed a network based clustering method for time series that allowed the detection of clusters with random shapes [20]. Another paper also working in the field of time series clustering by the use of community detection methods for networks is proposed by Ferreira et. al [16]. This paper, uses networks of datasets, whose links are our only inputs, as the main data for its algorithm and performs community detection methods on time series. The results of using this method, in comparison to other methods, has shown promising results and less complexity that although not studied and researched by previous research, could be particularly useful in the case of clustering time series that have complex characteristics, such as overlapping or hierarchical properties. The most recent article discussing the usage of community detection for overlapping datasets is also discussed through the work of Garrels et. al, where a faster and parallelized version of the FOX community detection for weighted graphs is introduced to handle the detection of such classes within solely large graphs [21], [37].

Although the first research related to community detection methods dates back to the 1970s, with research done in mathematical sociology and circuit design, Newman and Girvan's work on the modularity in complex systems is known for being one of the main researches in the field of network science research [14], [36], [45], [53]. Since then, various algorithms for community detection have been developed. These algorithms can be divided into two categories, heuristic methods and optimization methods, although both require a criterion to evaluate community partition [57]. A commonly used criterion is the modularity Q criterion proposed by Girvan-Newman, which suggests defining community structures as node groups quantitatively.

Despite the wide use of this method, due to the increasing density of intra-connections for overlapping datasets, the modularity Q algorithm cannot be used for overlapping communities. The Fastgreedy algorithm, is basically a fast implementation technique of the modularity Q algorithm as was proposed by Clauset [13]. This algorithm attempts to form new communities by merging those that show a maximum amount of improvement in the modularity score. This process continues until there are no more observations of improvement in the modularity score [46], [67]. Another algorithm is the Infomap algorithm, first mentioned by Rosvall. This algorithm finds communities using random walks and the concept of information diffusion and recognizes the most important properties of a network and showcases them in a map [16], [46], [67]. The Walktrap algorithm is a hierarchical community detection algorithm that was suggested by Latapy [48]. The basic concept of this algorithm is that short distanced random walks are likely to stay in the same community. This method, which uses the same greedy strategy as the Fastgreedy algorithm, merges two adjacent communities after calculating their distances and then updates the distance between the communities. The process in this algorithm is repeated N-1 times [16], [46], [67].

The Leading Eigenvector algorithm was also suggested by Newman. It is based on the concept of using the eigenvalues and the eigenvectors of the modularity matrix for spectral optimization of modularity.

Another community detection method was introduced as the Label Propagation method and it suggests assuming that each node in the network belongs to the same community that the majority of its' neighbors do. After listing the nodes in the networks in a random sequential order, each node is assigned to the same community the majority of its' neighbors are in [32]. The Spinglass algorithm is based on the Potts model, a generalization of the Ising model for more than two components and the main concept of this algorithm is that nodes belonging to the same spin must have edges connected. This algorithm continues for a given number of steps [52], [65], [67].

The Louvain algorithm for community detection also optimizes a quality function such as the modularity in 2 steps: 1) local moving of nodes and 2) aggregation of the network. In the first step, individual nodes are moved to a community that results in the largest increase in the value of the quality function. In the second step, an aggregate network is created with attention to the partition that was obtained in the previous step [61].

The application of the Genetic algorithm has also been studied for the purpose of community detection. Although the paper was originally set to use applications of this method in Artificial Intelligence, the method itself gained a lot of attention, due to its' novelty and for its' proposed practicality, claiming that suitable results are gained for sparse solution spaces of real-life problems. As a result of this interest, many different algorithms were proposed by various researchers, each developing the Genetic Algorithm with attention to specific details. A recent algorithm suggested is the GA-Net +, which suggests using the concept of line graph, but due to the genuine nature of the algorithm, fails to be recognized as a link based overlapping community detection method [47], [57]. A recent contribution uses multi-objective Genetic Algorithm and Fuzzy theory [31]. A link clustering algorithm for overlapping communities has also been proposed by Ahn [3], [71]. Evans has suggested that any algorithm that is capable of producing a partition of nodes, may be used for producing a partition of links [15]. Another algorithm is suggested and extends the map equation algorithm for link community detection [30]. Also, other research in the field includes the Extended Link Clustering method or ELC, which uses the Extended Link Similarity or ELS, to create denser transform matrices [24]. Another overlapping community detection algorithm tries to optimize network modularity through the use of the parliamentary optimization algorithm and attempts to analyze the overlapping communities in social networks [4]. A link clustering algorithm that is based on the density in order to improve the accuracy of overlapping community detection has also been suggested [71]. A memetic link clustering algorithm for detecting overlapping communities and identifying densely connected group of links on the weighted line graph was also suggested in 2018, but as stated by its' authors, it can be very time-consuming [35].

For this paper, we have compared classic algorithms for detecting communities in time series which many researchers before us have and still continue to compare results of their novel algorithms to. Therefore, we have attempted to compare results from our proposed method with those of the node community detection, Louvain, Fastgreedy, Walktrap, Infomap, Leading Eigenvector, Label Propagation and Spinglass algorithms.

III. THE PROPOSED METHOD

Although many clustering methods have been suggested through the work of many researchers in the field, these techniques often require some sort of presumptions regarding the attributes of the time series being studied. However, in real life, we often encounter time series that possess many characteristics, few of which we can surely make assumptions about. One characteristic that if found in clusters, can have undesirable results if used with conventional clustering methods is overlapping. Hierarchy is also another important attribute that is often assumed regarding the nature of the experiments done. Figure 1 shows overlapping properties often found in real life time series.

Despite the abundant amount of research done in the case of clustering for hierarchical overlapping datasets and link communities, our research is the first to apply link community detection methods to overlapping hierarchical time series of which no networks are predefined.

In our proposed method, the first distances between pairs of the time series are calculated. After normalizing the distances calculated, by using the method of K-Nearest Neighbors,

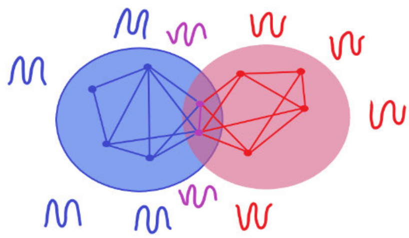

FIGURE 1. Overlapping communities.

a network of nodes with links between various time series is constructed. Finally, by using the method of link community detection, communities with the main focus on the unique links in the network are created. It is important to note that following the discussion made by Shang et. al, the nature of the datasets and the purpose of this paper, here we focus on the definition of communities that emphasize on dense internal relations, while also paying attention to the sparse relations external relations present [45], [56]. After plotting our results, we then attempt to test the accuracy of such results for various time series. Although many measures, internal and external, exist that can evaluate the accuracy of results obtained via our method, due to the limitations that indexes have, we decided to evaluate the quality of our communities using the element-centric clustering measure of similarity. This measure, which has unified the comparison of clustering with disjoint, overlapping and hierarchal characteristics, evaluates similarity between elements by the relationship induced by the cluster structure [22].

A flowchart of the steps in our proposed method has been shown in figure 2. We attempt to explain our proposed method in 5 steps using one of our synthetic time series, abbreviated as sts1.

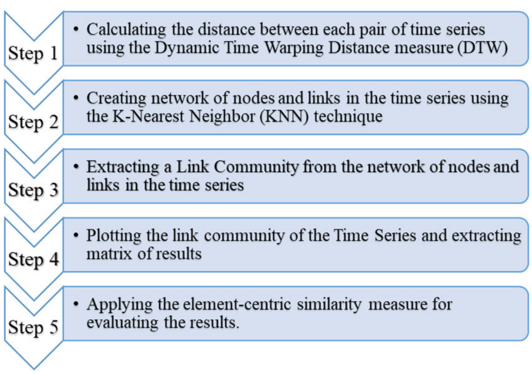

FIGURE 2. Steps of the proposed model.

Step 1: Calculating the distance between each pair of time series using the Dynamic Time Warping (DTW)

In this step, by defining each time series as a node, we attempt to find the distance between each pair of time series in our datasets. One method is by defining these distances in accordance to two classifiers: Euclidean and Non Euclidean. Euclidean space has some real-valued dimensions and dense points and there is often an understanding regarding the average of two points. On the other hand, non-Euclidean distances are based on the properties and attributes of points and not their location in space. In the case of time series distance measures, they may also be defined into four categories, Shape-based, edit-based, feature-based and structure-based [16]. Shape based measures directly compare the raw form of data for a pair of time series. Although being intuitive, parameter-free and having a linear complexity of the length of the time series, they are both misaligned in time and are sensitive to noise. The most common form is LP norms [16]. Lock step measures are also distance measures that calculate the distance for solely fixed pairs of data points, while elastic measures are designed to cover this issue. The most famous elastic measure is Dynamic Time Warping or DTW. In this method, two time series are aligned using the shortest warping path in a distance matrix. The time series axis may be compressed or stretched to achieve a reasonable fit, thus a template can match a large variety of time series. This allows us to better find patterns in time series outcomes of various types of research. Despite the Euclidean distance measure being efficient, it results are often very intuitive. The Euclidean distance fails to recognize identical time series that have a slight shifting along the time axis. In this case, the DTW measure gives intuitive distance measures, by ignoring both local and global shifts in time dimensions [54].

Another classic distance measure often mentioned by research papers in the field is the Manhattan distance measure. The Manhattan distance measure can simply be defined as the sum of absolute differences of elements. STS distance is another distance measure which considers the time series as precise linear functions and proceeds to measure differences of slopes between them [42]. DISSIM is also a distance measure designed to deal with time series that have been collected at different sampling rates. The Complexity Invariant Distance (CID) is a complexity invariance that uses the Euclidean distance with complexity differences between two time series as a correction factor [7]. Edit based distance also measures calculate the distance between two time series by calculating the minimum number of operations required to transform a time series into another [16]. The Levenshtein distance is also another method that calculates the distance by counting the number of deletions, insertions or substitutions required to transform a string to another [62].

Feature based distances extract a number of features from the time series and proceeds by comparing the extracted data. Discrete Wavelet Transform or DWT is a domain transforming technique for hierarchically decomposing sequences [68]. The Integrated Periodogram measure (INTPER) uses the integrated periodogram of time series to calculate the distance [11]. Structure based distance measures like the Hidden Markov Models or HMM, use some parametric models representing time series in order to identify higher level structures [16]. Auto Regressive Moving Average or ARMA is based on the current idea that the current value of time series can be expressed as the linear combination of past values [66].

For the calculations in this step, we can opt to use from a large variety of distance functions. In this paper, we have opted to use the Dynamic Time Warping Distance. This distance measure allows us to better convey the similarity of time series which aren't aligned. The Dynamic Time Warping method is used in the manner explained through previous research done in the field. The DTW distance matrix is shown in the figure below.

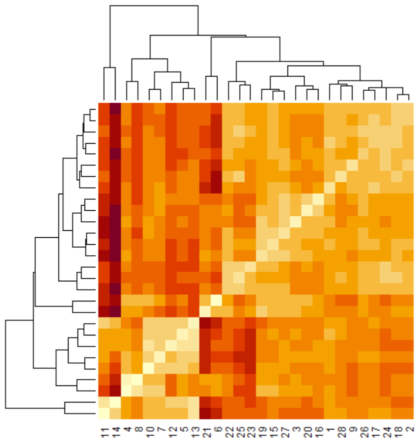

FIGURE 3. DTW distance matrix for sts1.

Step 2: Creating network of nodes and links in the time series using the K-Nearest Neighbor (KNN) technique

This technique allows us to find a primary network by analyzing the K nearest neighbors of a node. This method allows us to find links between nodes in the manner that can be used in link community detection algorithms and given in the state of real life data, this option assists us in finding properties of time series which are often the outcomes of research and can greatly contribute to finding links between each of these observations, information that given the nature of some experiments and the great number of time series, cannot be fully and accurately defined. An example of an overlapping network outcome for a synthetic and simulated time series with overlapping and hierarchical properties based on the Coffee time series is shown in figure 4.

Step 3: Extracting a Link Community from the network of nodes and links in the time series

In this step, using the network defined in the previous step, link community detection using the Louvain algorithm is used to create a community. An example of a community detection outcome is shown in figure 5, where the dendrogram for the previous network using the link community detection algorithm is calculated.

Step 4: Plotting the link community of the Time Series and extracting matrix of results

Given that in the dendrogram shown is for link community detection and that links are clustered, in order to better understand the community calculated and the properties the algorithm has identified in previous steps, we attempt to plot the outcome of our algorithm and receive the matrix including the number of nodes belonging to a single cluster,

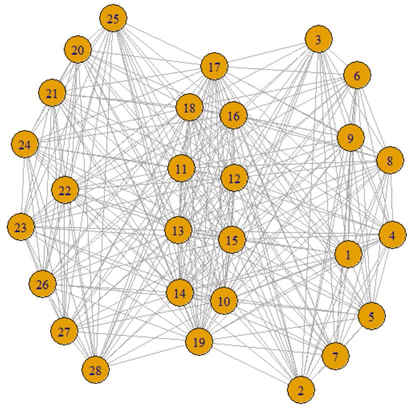

FIGURE 4. Network outcome for a series of synthetic overlapping time series based on the coffee dataset.

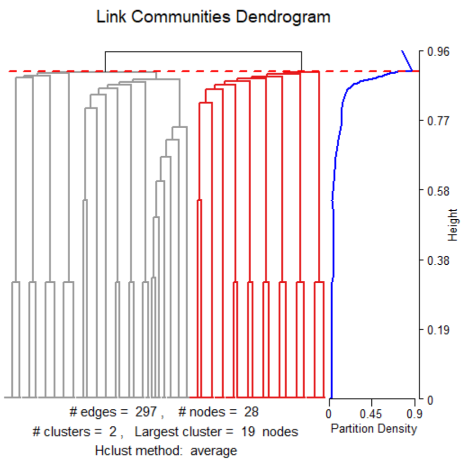

FIGURE 5. Link community detection using the average Hierarchical method for overlapping hierarchical time series.

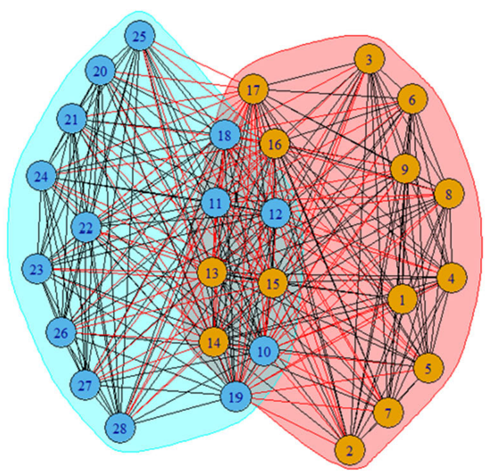

FIGURE 6. Plot of link community detection algorithm applied to a synthetic dataset with overlapping hierarchical time-series.

ones belonging to more than one cluster, given that we expect our time series to at least belong to one cluster, so that we can classify its' information, especially in the case of big data, where it is often seen that time series usually show behavior somewhat similar to previous data. Figure 6 shows the plot depicting the community of the network defined in figure 6.

Step 5: Applying the element-centric similarity measure for evaluating the results

Finally, we attempt to understand the results of applying this algorithm and compare it in terms of its' similarity to the real life classes that had previously been defined. For this purpose, both the Jaccard and Element Centric similarity measure have been used. We have explained the concept and equations for each similarity measure in previous parts.

IV. DATA

Although there are many real life time series with overlapping and hierarchical qualities, it is important to note that time series with such properties and pre-defined classes were not available. Due to the importance of using time series with such qualities and in order to prevent any under fitting and overfitting clustering, we have attempted to use time series with distinct, non-overlapping classes and have simulated datasets from these sources that have the characteristics desired. For the purpose of this research, we have used synthetic time series based on some datasets like the Coffee, SonyAIBOrobotsurface2, Epilepsy, NATOPS, PowerCons, ECG, UMD and Beef time series, all acquired and downloaded from The UEA & UCR Time Series Classification Repository [6]. In addition to the stated time series, an attempt to cluster time series with overlapping qualities, such as the Smooth time series have been depicted in the research. An overview of one of the synthetic time series is shown in figure 7. Also, table 1 includes some information regarding the synthetic time series produced.

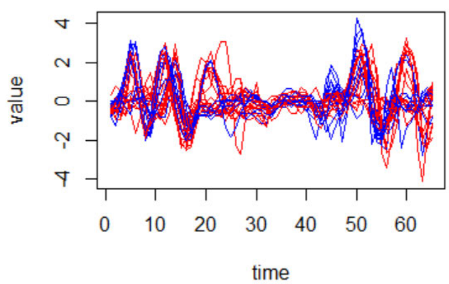

FIGURE 7. Overview of the time series in sts 3.

V. RESULTS AND DISCUSSIONS

As it had been previously mentioned, we measured our proposed method by using time series based on real distinct data and attempted to create synthetic data based on the ground truth that was available. In this section, we discuss results gathered by using the element centric similarity measure and compare community outcomes for each method. Based on the fact that using the KNN method for big data is very time consuming and that can also result in inaccuracies, we have also chosen to depict results of our algorithm for

TABLE 1. Properties of the synthetic time series used.

Time Series Name	No. of objects	Length	No. of classes
sts 1	28	286	2
sts 2	58	150	2
sts 3	34	65	2
sts 4	172	206	4
sts 5	115	96	2
sts 6	46	150	3
sts 7	190	51	6
sts 8	150	15	3
sts 9	200	144	2
sts 10	32	150	5

our relatively bigger time series using the εNN method. In the following tables, the average and standard deviation of the element centric, F-measure and Jaccard similarity measures are calculated for each algorithm of community detection, and an instance of performance of the algorithm using various distance measures, for a K and ε value, is shown and measured. It is worth noting that the proposed algorithm attempts to consider overlapping data points of more than 45 percent as clusters and fewer percentages of overlapping as overlapping clusters, something that given the nature of overlapping data, is understandable.

As noted from Table 2, our proposed method seems to find the most accurate results using the DTW distance with an accuracy of 0.85809 for a small time series such as sts 1. Based on the promising results gathered from using the DTW distance, we proceeded to use this distance measure as the basis for calculating the average and standard deviation for a variety of K values. Table 3 shows the results gathered for a variety of algorithms. As shown in the table, in terms of understanding the time series, the proposed method has been more capable of identifying similarities which are in the nature of the dataset, with a distinguishable element centric average of 0.72543, F-measure average of 0.82604 and Jaccard average of 0.75513.

For the synthetic time series 2, the DTW distance measure proved to be a great match for our proposed algorithm. Following the results, table 4 shows the Average and Standard Deviation of similarity measures for this time series when using the DTW distance, which depicts promising results. The proposed algorithm has an average value of 0.7675 of the element-centric measure, 0.7799 of the F-measure and 0.6452 of the Jaccard similarity measure, which are all noticeably higher than the other algorithms being studied in this article. To compare the results of using different distance measures, the Euclidean and Manhattan distance measures were applied to the dataset and table 5 depicts the values of similarity measures for the best K value of 19.

TABLE 2. **(a) Similarity measures between communities detected and ground truth by various algorithms using the DTW distance measure, for sts 1 with ground truth data acquired from K=18. (b) Similarity measures between communities detected and ground truth by various algorithms using the Euclidean distance measure, for sts 1 with ground truth data acquired from K=18. (c) Similarity measures between communities detected and ground truth by various algorithms using the Manhattan distance measure, for sts 1 with ground truth data acquired from K=18.**

(a)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.525138388	0.707240294	0.547077922
*Link Community	0.858097473	1.030257186	1.062402496
Fastgreedy	0.540716571	0.764397906	0.618644068
Infomap	0.849609049	0.875836431	0.779100529
Walktrap	0.567086486	0.731804586	0.577044025
Leading Eigenvector	0.592262963	0.81555334	0.688552189
Label Propagation	0.849609049	0.875836431	0.779100529
Spinglass	0.519145803	0.682056663	0.517515924

(b)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.525138388	0.707240294	0.547077922
*Link Community	0.848144253	0.986749805	0.973846154
Fastgreedy	0.540716571	0.764397906	0.618644068
Infomap	0.849609049	0.875836431	0.779100529
Walktrap	0.567086486	0.731804586	0.577044025
Leading Eigenvector	0.592262963	0.81555334	0.688552189
Label Propagation	0.849609049	0.875836431	0.779100529
Spinglass	0.532591031	0.74501574	0.593645485

(c)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.542265502	0.781151832	0.640893471
*Link Community	0.849609049	0.875836431	0.779100529
Fastgreedy	0.546480626	0.789528796	0.652249135
Infomap	0.849609049	0.875836431	0.779100529
Walktrap	0.594205475	0.879187817	0.78442029
Leading Eigenvector	0.557412016	0.826222685	0.703900709
Label Propagation	0.849609049	0.875836431	0.779100529
Spinglass	0.542265502	0.781151832	0.640893471

TABLE 3. The average and standard deviation of similarity measures for sts 1.

	Element-Centric		F-Measure		Jaccard Similarity	
Algorithm	Average	Std Dev	Average	Std Dev	Average	Std Dev
Louvain	0.461065652	0.111621725	0.691257102	0.133621848	0.542231909	0.147514215
*Link Community	0.725436119	0.137505461	0.826040681	0.211601778	0.755139925	0.310748526
Fastgreedy	0.448779741	0.11193637	0.683673684	0.132204547	0.533126578	0.146498833
Infomap	0.519325958	0.252886894	0.678419112	0.189835585	0.542311594	0.215894684
Walktrap	0.420960083	0.117370019	0.635532743	0.126539653	0.477345087	0.133538718
Leading Eigenvector	0.449523259	0.145354332	0.680372118	0.173672676	0.539050737	0.190434138
Label Propagation	0.504678703	0.175830521	0.697064025	0.149994842	0.552961705	0.167326875
Spinglass	0.410186602	0.117408864	0.628069138	0.135145839	0.470981712	0.143360024

TABLE 4. The average and standard deviation of similarity measures for sts 2.

	Element-Centric		F-Measure		Jaccard Similarity	
Algorithm	Average	Std Dev	Average	Std Dev	Average	Std Dev
Louvain	0.370518169	0.063813851	0.478357168	0.122370464	0.324002989	0.133149816
*Link Community	0.767577457	0.050681241	0.779971568	0.078176614	0.645273786	0.103714118
Fastgreedy	0.477391665	0.058218018	0.555607135	0.060062895	0.386877217	0.059238453
Infomap	0.469003443	0.070564574	0.545727127	0.056418799	0.377071559	0.051998218
Walktrap	0.350062315	0.03031653	0.451768928	0.068309862	0.294273943	0.062448042
Leading Eigenvector	0.381865426	0.080390634	0.490384169	0.124633819	0.334761387	0.133849321
Label Propagation	0.578028051	0.140059617	0.626770663	0.109354966	0.465347173	0.124568894
Spinglass	0.352908958	0.019515866	0.458184329	0.071242474	0.299997253	0.067942756

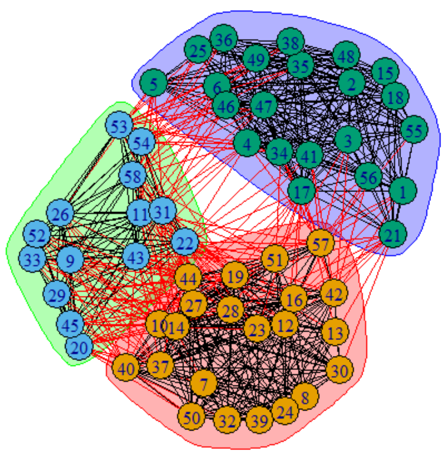

FIGURE 8. Node community detection fails to recognize overlapping properties of time series.

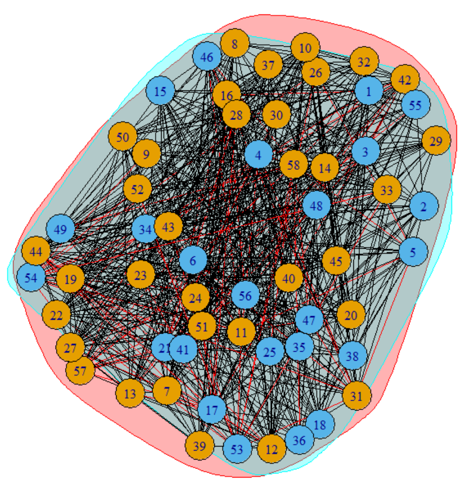

FIGURE 9. Node community detection fails to recognize overlapping properties of time series.

The first algorithm performed on this time series was the Louvain method for community detection. Figure 8 shows the outcome of our code for a K value of 19. As it may be seen, this algorithm fails to recognize and create communities with the properties mentioned, something that results in a low element- centric measure of similarity outcome when compared to the ground truth. The three similarity measures have been calculated for this time series and as it can be viewed, as a result of not recognizing overlapping, the three outcomes are very low.

TABLE 5. (a) Similarity measures between communities detected and ground truth by various algorithms using the DTW distance measure, for sts 2 with ground truth data acquired from K=19. (b) Similarity measures between communities detected and ground truth by various algorithms using the Euclidean distance measure, for sts 2 with ground truth data acquired from K=19. (c) Similarity measures between communities detected and ground truth by various algorithms using the Manhattan distance measure, for sts 2 with ground truth data acquired from K=19.

(a)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.344588676	0.432465923	0.275889328
*Link Community	0.800215731	0.841454802	0.726302956
Fastgreedy	0.512542805	0.561643836	0.39047619
Infomap	0.512542805	0.561643836	0.39047619
Walktrap	0.392241379	0.453908985	0.293584906
Leading Eigenvector	0.503842478	0.552603613	0.381791483
Label Propagation	0.838210862	0.781871776	0.641863279
Spinglass	0.344588676	0.432465923	0.275889328

(b)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.344588676	0.432465923	0.275889328
*Link Community	0.800215731	0.841454802	0.726302956
Fastgreedy	0.512542805	0.561643836	0.39047619
Infomap	0.512542805	0.561643836	0.39047619
Walktrap	0.392241379	0.453908985	0.293584906
Leading Eigenvector	0.507606677	0.555849656	0.384897361
Label Propagation	0.512542805	0.561643836	0.39047619
Spinglass	0.838210862	0.781871776	0.641863279

(c)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.623641729	0.569250986	0.397869214
*Link Community	0.745849073	0.7694339	0.62526824
Fastgreedy	0.499880545	0.560298826	0.389177168
Infomap	0.512542805	0.561643836	0.39047619
Walktrap	0.502747472	0.564567769	0.39330855
Leading Eigenvector	0.499880545	0.560298826	0.389177168
Label Propagation	0.512542805	0.561643836	0.39047619
Spinglass	0.476915266	0.531672983	0.362094395

Next we approached the Fastgreedy algorithm for community detection. Although there is a slight improvement in the recognition of clusters, this method has failed to recognize overlapping qualities and still isn't able to fully distinguish

TABLE 6. The average and standard deviation of similarity measures for sts 3.

	Element-Centric		F-Measure		Jaccard Similarity	
Algorithms	Average	Std Dev	Average	Std Dev	Average	Std Dev
Louvain	0.371777178	0.022856502	0.533089715	0.040219382	0.364258487	0.038690682
*Link Community	0.606399844	0.110776719	0.670712273	0.114397089	0.513406572	0.128693432
Fastgreedy	0.36639664	0.022121919	0.51739761	0.047501409	0.350121566	0.044577189
Infomap	0.467226723	0.093615647	0.630136703	0.066242454	0.462773048	0.071696894
Walktrap	0.386458646	0.023871702	0.559970836	0.040744679	0.389757157	0.039603136
Leading Eigenvector	0.340919092	0.030986247	0.492981301	0.03287966	0.327627673	0.028885607
Label Propagation	0.479547955	0.082159873	0.636364358	0.059660277	0.468950433	0.065228968
Spinglass	0.394149415	0.102478967	0.551620012	0.090204144	0.385411226	0.091581929

TABLE 7. (a) Similarity measures between communities detected and ground truth by various algorithms using the DTW distance measure, for sts 3 with ground truth data acquired from K=8. (b) Similarity measures between communities detected and ground truth by various algorithms using the Euclidean distance measure, for sts 3 with ground truth data acquired from K=8.

(a)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.365511551	0.523671498	0.354712042
*Link Community	0.756090809	0.80748976	0.677134446
Fastgreedy	0.357885789	0.50728863	0.33984375
Infomap	0.569406941	0.701339638	0.540048544
Walktrap	0.365511551	0.523671498	0.354712042
Leading Eigenvector	0.357885789	0.50728863	0.33984375
Label Propagation	0.569406941	0.701339638	0.540048544
Spinglass	0.569406941	0.701339638	0.540048544

(b)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.365511551	0.523671498	0.354712042
*Link Community	0.784355656	0.74644727	0.595465394
Fastgreedy	0.357885789	0.50728863	0.33984375
Infomap	0.569406941	0.701339638	0.540048544
Walktrap	0.365511551	0.523671498	0.354712042
Leading Eigenvector	0.846357363	0.795208214	0.660037879
Label Propagation	0.569406941	0.701339638	0.540048544
Spinglass	0.365511551	0.523671498	0.354712042

TABLE 7. *(Continued.)* **(c) Similarity measures between communities detected and ground truth by various algorithms using the Manhattan distance measure, for sts 3 with ground truth data acquired from K=8.**

(c)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.370087009	0.514832536	0.346649485
*Link Community	0.696372522	0.719083969	0.561382598
Fastgreedy	0.423417342	0.587807097	0.416237113
Infomap	0.569406941	0.701339638	0.540048544
Walktrap	0.387238724	0.538243626	0.368217054
Leading Eigenvector	0.357885789	0.50728863	0.33984375
Label Propagation	0.569406941	0.701339638	0.540048544
Spinglass	0.370087009	0.514832536	0.346649485

TABLE 8. **The average and standard deviation of similarity measures for sts 4.**

	Element-Centric		F-Measure		Jaccard Similarity	
Algorithms	Average	Std Dev	Average	Std Dev	Average	Std Dev
Louvain	0.346574006	0.062402876	0.391017577	0.032040484	0.243446953	0.024416654
*Link Community	0.652263171	0.015935389	0.643374514	0.054300129	0.47632578	0.059401272
Fastgreedy	0.452469853	0.031515245	0.430672514	0.006834143	0.274452359	0.005537846
Infomap	0.390408141	0.044407518	0.399763834	0.024616906	0.250075376	0.019305733
Walktrap	0.381023309	0.06425415	0.408773252	0.02287308	0.257118547	0.018021502
Leading Eigenvector	0.368819763	0.054837065	0.399034632	0.017563586	0.249379366	0.013867124
Label Propagation	0.446940835	0.043271784	0.425911521	0.015341607	0.270680552	0.012188876
Spinglass	0.337480176	0.067396198	0.378862781	0.036429648	0.234244139	0.027585678

TABLE 9. **(a) Similarity measures between communities detected and ground data by various algorithms using the DTW distance measure, for sts 4 with ground data acquired from K=26.**

(a)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.476226352	0.437676999	0.280145014
*Link Community	0.664183819	0.597885584	0.426417115
Fastgreedy	0.47539402	0.436127628	0.27887674
Infomap	0.475316777	0.434971154	0.277931717
Walktrap	0.475687746	0.435763688	0.278579192
Leading Eigenvector	0.47539402	0.436127628	0.27887674
Label Propagation	0.475687746	0.435763688	0.278579192
Spinglass	0.470126052	0.431781638	0.2753326

TABLE 9. *(Continued.)* **(b) Similarity measures between communities detected and ground data by various algorithms using the Euclidean distance measure, for sts 4 with ground data acquired from K=26. (c) Similarity measures between communities detected and ground data by various algorithms using the Manhattan distance measure, for sts 4 with ground data acquired from K=26.**

(b)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.476226352	0.437676999	0.280145014
*Link Community	0.664697938	0.5982701	0.426808403
Fastgreedy	0.47539402	0.436127628	0.27887674
Infomap	0.475316777	0.434971154	0.277931717
Walktrap	0.475687746	0.435763688	0.278579192
Leading Eigenvector	0.47539402	0.436127628	0.27887674
Label Propagation	0.475687746	0.435763688	0.278579192
Spinglass	0.470961176	0.434081209	0.277205441

(c)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.45092965	0.427582858	0.271927116
*Link Community	0.703221455	0.539658896	0.369543044
Fastgreedy	0.462513627	0.431546793	0.275141643
Infomap	0.475687746	0.435763688	0.278579192
Walktrap	0.374797146	0.386057278	0.23920135
Leading Eigenvector	0.475351607	0.436125228	0.278874777
Label Propagation	0.476582775	0.434675589	0.277690418
Spinglass	0.462513627	0.431546793	0.275141643

TABLE 10. **The average and standard deviation of similarity measures for sts 5.**

	Element-Centric		F-Measure		Jaccard Similarity	
Algorithms	Average	Std Dev	Average	Std Dev	Average	Std Dev
Louvain	0.298060805	0.042686288	0.47458471	0.05283444	0.312618342	0.047649592
*Link Community	0.803184822	0.049381193	0.979541586	0.133329817	0.988435315	0.241234742
Fastgreedy	0.514165959	0.146602101	0.637959347	0.099249352	0.475555882	0.108569741
Infomap	0.429999778	0.124310169	0.590935581	0.102968464	0.425779261	0.096716595
Walktrap	0.325550315	0.045142645	0.525279095	0.048383359	0.357469525	0.042958666
Leading Eigenvector	0.364827982	0.02867273	0.556607233	0.035741902	0.386355999	0.032513899
Label Propagation	0.497093427	0.096177787	0.635033522	0.066519176	0.46832009	0.069666548
Spinglass	0.324392878	0.068627249	0.504464275	0.091201921	0.341725541	0.079646017

members of clusters accurately. As it can be seen in table 5, the element-centric similarity measure shows a similarity of 0.51254 for the best k value of 19 and an average element-centric similarity of 0.47739 based on Table 4.

TABLE 11. (a) Similarity measures between communities detected and ground data by various algorithms using the DTW distance measure, for sts 5 with ground data acquired from K=29. (b) Similarity measures between communities detected and ground data by various algorithms using the Euclidean distance measure, for sts 5 with ground data acquired from K=29. (c) Similarity measures between communities detected and ground data by various algorithms using the Manhattan distance measure, for sts 5 with ground data acquired from K=29.

(a)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.379615545	0.577613594	0.406087679
*Link Community	0.861245681	0.817160644	0.690846682
Fastgreedy	0.861245681	0.817160644	0.690846682
Infomap	0.558488736	0.69289293	0.530096536
Walktrap	0.316250942	0.519685039	0.35106383
Leading Eigenvector	0.368570367	0.552963101	0.382134762
Label Propagation	0.5861808	0.691189558	0.528105168
Spinglass	0.379615545	0.577613594	0.406087679

(b)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.379615545	0.577613594	0.406087679
*Link Community	0.861245681	0.817160644	0.690846682
Fastgreedy	0.558488736	0.69289293	0.530096536
Infomap	0.559488746	0.687111275	0.523358349
Walktrap	0.316250942	0.519685039	0.35106383
Leading Eigenvector	0.368570367	0.552963101	0.382134762
Label Propagation	0.5861808	0.691189558	0.528105168
Spinglass	0.376385721	0.570857101	0.399440182

(c)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.380919308	0.566357461	0.395047891
*Link Community	0.794489549	0.967936803	0.937865826
Fastgreedy	0.558488736	0.69289293	0.530096536
Infomap	0.582724624	0.695076097	0.532656422
Walktrap	0.371630598	0.582897547	0.411330561
Leading Eigenvector	0.368570367	0.552963101	0.382134762
Label Propagation	0.423569657	0.601211615	0.429808841
Spinglass	0.374765133	0.560165058	0.389048106

TABLE 12. The average and standard deviation of similarity measures for sts 6.

	Element-Centric		F-Measure		Jaccard Similarity	
Algorithms	Average	Std Dev	Average	Std Dev	Average	Std Dev
Louvain	0.377976455	0.106246081	0.479731471	0.061789571	0.317642343	0.056531038
*Link Community	0.576714532	0.088736485	0.625105071	0.089604094	0.460589752	0.100433482
Fastgreedy	0.414232457	0.070839009	0.504514809	0.044326235	0.338414948	0.039208365
Infomap	0.38399471	0.072655087	0.480763787	0.05787863	0.318162952	0.049457103
Walktrap	0.373570263	0.081659974	0.469122404	0.052074945	0.307808248	0.044275377
Leading Eigenvector	0.37423981	0.080836314	0.477945925	0.043854756	0.315005526	0.037891057
Label Propagation	0.376022958	0.098761548	0.468808852	0.067919245	0.308485466	0.057416046
Spinglass	0.369627882	0.081837807	0.476455146	0.040680319	0.313590035	0.035534742

TABLE 13. (a) Similarity measures between communities detected and ground data by various algorithms using the DTW distance measure, for sts 6 with ground data acquired from K=15. (b) Similarity measures between communities detected and ground data by various algorithms using the Euclidean distance measure, for sts 6 with ground data acquired from K=15.

(a)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.685466673	0.633587786	0.463687151
*Link Community	0.638990344	0.60130719	0.429906542
Fastgreedy	0.46996293	0.521200572	0.352448454
Infomap	0.46996293	0.521200572	0.352448454
Walktrap	0.600002813	0.562734082	0.391530945
Leading Eigenvector	0.600002813	0.562734082	0.391530945
Label Propagation	0.600002813	0.562734082	0.391530945
Spinglass	0.600002813	0.562734082	0.391530945

(b)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.487873742	0.544262295	0.373873874
*Link Community	0.63163981	0.605496454	0.434202161
Fastgreedy	0.487873742	0.544262295	0.373873874
Infomap	0.487873742	0.544262295	0.373873874
Walktrap	0.487873742	0.544262295	0.373873874
Leading Eigenvector	0.487873742	0.544262295	0.373873874
Label Propagation	0.487873742	0.544262295	0.373873874
Spinglass	0.487873742	0.544262295	0.373873874

TABLE 13. *(Continued.)* (c) Similarity measures between communities detected and ground data by various algorithms using the Manhattan distance measure, for sts 6 with ground data acquired from K=15.

(c)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.487873742	0.544262295	0.373873874
*Link Community	0.638900607	0.60991663	0.438762626
Fastgreedy	0.487873742	0.544262295	0.373873874
Infomap	0.487873742	0.544262295	0.373873874
Walktrap	0.487873742	0.544262295	0.373873874
Leading Eigenvector	0.487873742	0.544262295	0.373873874
Label Propagation	0.487873742	0.544262295	0.373873874
Spinglass	0.370465272	0.48447205	0.319672131

TABLE 14. The average and standard deviation of similarity measures for sts 7.

	Element-Centric		F-Measure		Jaccard Similarity	
Algorithms	Average	Std Dev	Average	Std Dev	Average	Std Dev
Louvain	0.418238554	0.012240249	0.453516677	0.011553476	0.293316964	0.009671671
*Link Community	0.338506573	0.019768483	0.630680721	0.041670199	0.461708036	0.044512984
Fastgreedy	0.390807012	0.069945592	0.477347287	0.037977564	0.314168522	0.032307778
Infomap	0.421273536	0.03218563	0.454422828	0.029217794	0.294391991	0.023933972
Walktrap	0.436336187	0.015222589	0.475784872	0.015380939	0.312261743	0.013200951
Leading Eigenvector	0.420317553	0.006842394	0.452262519	0.008059848	0.292238057	0.006740869
Label Propagation	0.408175709	0.041091875	0.445099379	0.043541421	0.287075656	0.035145588
Spinglass	0.433104262	0.009356516	0.4692842	0.005500765	0.306592264	0.004673493

As for the infomap algorithm, although clusters have different structures and each cluster includes different nodes, yet improvements are identical to those of the Fastgreedy. Figure 9 shows the outcome of this algorithm when used on sts 2.

In addition to the methods used, other algorithms, including Spinglass, Leading Eigenvector walktrap have also been performed on the datasets, yet results were not as promising as the Label propagation and proposed algorithm. The Label propagation method comes in second when using the DTW and Euclidean distance measures, losing to the proposed link community detection algorithm. For our proposed algorithm, it can be said that accurate overlapping properties were all found within datasets using different distance measures, something that indeed shows the utility of our method when used on real life time series, with overlapping and hierarchal properties.

Tables 8, 10 and 12 depict the average and standard deviation for sts 4,5,6. The results depicted in these tables all indicate an increase in preciseness and accuracy of our proposed algorithm. As applied before, we tested other distance measures for a variety of K values and have shown the results for the best K value as an example. For sts 4, an average element-centric similarity of 0.65226 was calculated, while neither of the other algorithms have an average similarity score value above 0.45. Similar results can be seen in table 8, as the F-measure and Jaccard similarity of our proposed method depicts values larger than those obtained from using other algorithms, to the point that when comparing results of the Jaccard similarity between the proposed algorithm and the Spinglass algorithm, an increase of more than twice the value of similarity was obtained from our algorithm.

As for sts 5, which was based on the ECG dataset, an amazing average of 0.80318 was evaluated for the element-centric similarity of our algorithm, which clearly stands higher than the results of those algorithms in comparison with it. In addition to the results mentioned, both the average F-measure and Jaccard similarity measure of our algorithm is above 0.97.

TABLE 15. (a) Similarity measures between communities detected and ground data by various algorithms using the DTW distance measure, for sts 7 with ground data acquired from K=26. (b) Similarity measures between communities detected and ground data by various algorithms using the Euclidean distance measure, for sts 7 with ground data acquired from K=26. (c) Similarity measures between communities detected and ground data by various algorithms using the Manhattan distance measure, for sts 7 with ground data acquired from K=26.

(a)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.399255447	0.446696649	0.287578501
*Link Community	0.312846708	0.609554253	0.438387657
Fastgreedy	0.433231825	0.517820324	0.34936407
Infomap	0.438235839	0.471304905	0.308305369
Walktrap	0.425071362	0.458184552	0.297172111
Leading Eigenvector	0.413556211	0.44440132	0.285678643
Label Propagation	0.438246382	0.495263339	0.329136221
Spinglass	0.438246382	0.470276899	0.30742616

(b)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.399255447	0.446696649	0.287578501
*Link Community	0.31398611	0.611614278	0.440521873
Fastgreedy	0.433231825	0.517820324	0.34936407
Infomap	0.438235839	0.471304905	0.308305369
Walktrap	0.425071362	0.458184552	0.297172111
Leading Eigenvector	0.413556211	0.44440132	0.285678643
Label Propagation	0.409851051	0.449070356	0.289549148
Spinglass	0.447924347	0.482508891	0.317964888

(c)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.399255447	0.446696649	0.287578501
*Link Community	0.29357761	0.604098806	0.432766164
Fastgreedy	0.42967673	0.510340254	0.342588471
Infomap	0.410511569	0.44555774	0.286635117
Walktrap	0.420377616	0.452087501	0.292062698
Leading Eigenvector	0.342971005	0.390046214	0.242271683
Label Propagation	0.373476702	0.392854928	0.244442729
Spinglass	0.41977349	0.452052155	0.292033195

For sts 6, an average value of 0.576714532 was seen for the element-centric similarity of our algorithm, and the best K value of 15 showed a similarity value of 0.63899, which in comparison shows a similarity more than all the algorithms in

TABLE 16. The average and standard deviation of similarity measures for sts 8.

	Element-Centric		F-Measure		Jaccard Similarity	
Algorithms	Average	Std Dev	Average	Std Dev	Average	Std Dev
Louvain	0.390644444	0.079721272	0.554081391	0.080532914	0.38655481	0.080256314
*Link Community	0.926006022	0.085512266	0.989744268	0.024140031	0.980525727	0.046256674
Fastgreedy	0.468644444	0.088413287	0.63085867	0.088379026	0.4650783	0.089006665
Infomap	0.833355556	0.333288889	0.87376271	0.252474581	0.832237136	0.335525727
Walktrap	0.458577778	0.144213815	0.615121902	0.137084562	0.454944072	0.145181693
Leading Eigenvector	0.425088889	0.094932928	0.587979968	0.094950883	0.421230425	0.095570061
Label Propagation	0.8878	0.2244	0.927041789	0.145916421	0.88704698	0.22590604
Spinglass	0.415533333	0.097184147	0.578056327	0.09854575	0.411610738	0.09783639

TABLE 17. (a) Similarity measures between communities detected and ground data by various algorithms using the DTW distance measure, for sts 8 with ground data acquired from $\varepsilon = 0.5$. (b) Similarity measures between communities detected and ground data by various algorithms using the Euclidean distance measure, for sts 8 with ground data acquired from $\varepsilon = 0.5$.

(a)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.464417391	0.588790055	0.417891648
*Link Community	0.5957232	0.752425529	0.671768198
Fastgreedy	0.533750628	0.665890236	0.499154673
Infomap	0.666666667	0.747474747	0.66442953
Walktrap	0.597425292	0.719250488	0.561788622
Leading Eigenvector	0.533750628	0.665890236	0.499154673
Label Propagation	0.666666667	0.747474747	0.66442953
Spinglass	0.531900735	0.66444741	0.497511187

(b)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.461767942	0.584627374	0.413952157
*Link Community	0.5957232	0.752425529	0.671768198
Fastgreedy	0.533750628	0.665890236	0.499154673
Infomap	0.666666667	0.747474747	0.66442953
Walktrap	0.597425292	0.719250488	0.561788622
Leading Eigenvector	0.533750628	0.665890236	0.499154673
Label Propagation	0.666666667	0.747474747	0.66442953
Spinglass	0.531900735	0.66444741	0.497511187

study. This shows that for data similar to the dataset, which is based on the UMD dataset, using the algorithm can accurately understand the properties of time series and therefore detects more realistic communities.

TABLE 17. *(Continued.)* **(c) Similarity measures between communities detected and ground data by various algorithms using the Manhattan distance measure, for sts 8 with ground data acquired from $\varepsilon = 0.5$.**

(c)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.607842105	0.70595723	0.585752359
*Link Community	0.666666667	0.747474747	0.66442953
Fastgreedy	0.539286335	0.670768836	0.504645043
Infomap	0.666666667	0.747474747	0.66442953
Walktrap	0.611111111	0.729004228	0.573857204
Leading Eigenvector	0.540381025	0.672049655	0.506088276
Label Propagation	0.666666667	0.747474747	0.66442953
Spinglass	0.539286335	0.670768836	0.504645043

TABLE 18. The average and standard deviation of similarity measures for sts 9.

	Element-Centric		F-Measure		Jaccard Similarity	
Algorithms	Average	Std Dev	Average	Std Dev	Average	Std Dev
Louvain	0.613991014	0.05454358	0.771020628	0.043061569	0.628678673	0.056173661
*Link Community	0.791339386	0.037476922	0.836037438	0.07247871	0.722566626	0.10354212
Fastgreedy	0.59226698	0.016880345	0.777037657	0.041863715	0.636627213	0.054944506
Infomap	0.815344828	0	0.752351097	0	0.603015075	0
Walktrap	0.628037255	0.147693313	0.736596339	0.039774692	0.584063405	0.049466509
Leading Eigenvector	0.58553605	0.011116799	0.782052364	0.032038623	0.642853293	0.042571377
Label Propagation	0.815344828	0	0.752351097	0	0.603015075	0
Spinglass	0.573764368	0.010873003	0.726406026	0.037964304	0.571291012	0.046896916

For sts 7, we attempted to use the NATOPS time series. As depicted in tables 14 and 15, while the proposed algorithm gained a low element centric score, the value wasn't very far from the results obtained from other algorithms. This shows that in its' worst case, our algorithm can still provide useful and true information. Also, the average F-measure showed an increase in value, suggesting that the algorithm has been successful in understanding true and false memberships within clusters.

As for sts 8, very impressive results were seen. Sts 8 is based on the Smooth dataset, which has 150 objects and a length of 15. For this dataset, the average element centric value for our algorithm is 0.92600 and for $\varepsilon = 0.5$ we can see that the proposed algorithm has shown similar results to the Infomap and Label Propagation methods. Table 16 shows that despite this similarity, the overall performance of our algorithm is higher than those of the two algorithms mentioned. Not only is the element-centric similarity average higher, but also the average F-measure similarity is 0.98974 and Jaccard similarity average is 0.98052, higher than any other of the algorithms studied. In addition to the results, the standard deviation of all three similarity measures for our algorithm is smaller to those of Infomap and Label Propagation. This means that given various values for $\varepsilon = 0.5$, results are similar to one another and the algorithm has been able to successfully recognize the true communities that each link belongs to.

Finally, sts 10 results show that for this dataset, which is based on the beef dataset, the average element-centric for all algorithms are similar and about 0.27, yet for the proposed algorithm in this paper, the average is about 0.39951, the F-measure average is 0.47989 and the Jaccard similarity average is 0.317924.

For sts 9, our algorithm was capable of creating an average element-centric similarity value of 0.791339 and a best case of similarity in $\varepsilon = 0.85$, 0.835144, while also showing an average 0.83603 and 0.72256 for the F-measure and Jaccard measure of similarities, respectively. This means that our algorithm, although not creating the best element-centric value, is arguably one of the best algorithms for community detection for datasets similar to the PowerCons dataset. not creating the best element-centric value, is arguably one of the best algorithms for community detection for datasets similar to the PowerCons dataset.

TABLE 19. (a) Similarity measures between communities detected and ground data by various algorithms using the DTW distance measure, for sts 9 with ground data acquired from ε =0.85. (b) Similarity measures between communities detected and ground data by various algorithms using the Euclidean distance measure, for sts 9 with ground data acquired from ε =0.85. (c) Similarity measures between communities detected and ground data by various algorithms using the Manhattan distance measure, for sts 9 with ground data acquired from ε =0.85.

(a)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.675253607	0.804539495	0.672995461
*Link Community	0.815344828	0.752351097	0.603015075
Fastgreedy	0.605788401	0.802659359	0.670368425
Infomap	0.815344828	0.752351097	0.603015075
Walktrap	0.794177596	0.744054506	0.592425793
Leading Eigenvector	0.583112853	0.801361358	0.668559589
Label Propagation	0.815344828	0.752351097	0.603015075
Spinglass	0.568203762	0.725720287	0.569514118

(b)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.582086207	0.764474886	0.618744918
*Link Community	0.815344828	0.752351097	0.603015075
Fastgreedy	0.605788401	0.802659359	0.670368425
Infomap	0.815344828	0.752351097	0.603015075
Walktrap	0.794177596	0.744054506	0.592425793
Leading Eigenvector	0.583112853	0.801361358	0.668559589
Label Propagation	0.815344828	0.752351097	0.603015075
Spinglass	0.559562696	0.703341855	0.54242659

(c)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.500844828	0.549315068	0.378659112
*Link Community	0.815344828	0.752351097	0.603015075
Fastgreedy	0.635103448	0.733928276	0.579689336
Infomap	0.815344828	0.752351097	0.603015075
Walktrap	0.799336123	0.746047462	0.594956699
Leading Eigenvector	0.621481191	0.726212452	0.570120545
Label Propagation	0.815344828	0.752351097	0.603015075
Spinglass	0.526360502	0.612624949	0.441571259

As it can be seen in the tables 16-21, various measures of similarity calculated for the three time series, all indicate a high consistent level of accuracy. This is mainly because of the fact that in this algorithm, careful attention is given towards the identifying overlapping properties of communities, instead of detecting new classes, which all other algorithms tend to do. Given the value of importance that the element-centric similarity measure pays to these

TABLE 20. The average and standard deviation of similarity measures for sts 10.

	Element-Centric		F-Measure		Jaccard Similarity	
Algorithms	Average	Std Dev	Average	Std Dev	Average	Std Dev
Louvain	0.278930237	0.013704372	0.261451304	0.019763189	0.150513627	0.01322138
*Link Community	0.399516736	0.048845954	0.479897112	0.067288613	0.317924383	0.05863086
Fastgreedy	0.286091593	0.018377691	0.274961269	0.030039509	0.159691999	0.019948196
Infomap	0.28913556	0.018002824	0.281673326	0.021150843	0.164072767	0.01419189
Walktrap	0.277842052	0.015083486	0.261968893	0.019634901	0.150854559	0.013133008
Leading Eigenvector	0.277231198	0.010984318	0.256115731	0.01808799	0.146972795	0.012114132
Label Propagation	0.300746534	0.009612464	0.293364536	0.008157029	0.171919168	0.005551339
Spinglass	0.278991844	0.019281549	0.258900432	0.01677201	0.14879233	0.011257073

TABLE 21. (a) Similarity measures between communities detected and ground data by various algorithms using the DTW distance measure, for sts 10 with ground data acquired from K=10. (b) Similarity measures between communities detected and ground data by various algorithms using the Euclidean distance measure, for sts 10 with ground data acquired from K=10.

(a)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.300362158	0.296296296	0.173913043
*Link Community	0.359660428	0.448230668	0.288851351
Fastgreedy	0.300362158	0.296296296	0.173913043
Infomap	0.300362158	0.296296296	0.173913043
Walktrap	0.300362158	0.296296296	0.173913043
Leading Eigenvector	0.300362158	0.296296296	0.173913043
Label Propagation	0.300362158	0.296296296	0.173913043
Spinglass	0.300362158	0.296296296	0.173913043

(b)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.300362158	0.296296296	0.173913043
*Link Community	0.359660428	0.448230668	0.288851351
Fastgreedy	0.300362158	0.296296296	0.173913043
Infomap	0.300362158	0.296296296	0.173913043
Walktrap	0.300362158	0.296296296	0.173913043
Leading Eigenvector	0.300362158	0.296296296	0.173913043
Label Propagation	0.300362158	0.296296296	0.173913043
Spinglass	0.300362158	0.296296296	0.173913043

properties, we can clearly understand that this algorithm will gain a high value from this measure. But this reason alone cannot prove our statement. The use of other well-known and common similarity measures also contributes to the accuracy that the element-centric similarity measure has calculated and all show a great amount of accuracy in the communities produced which shows that the algorithm has been successful in not only identifying overlapping and hierarchal properties, but has also been successful in acknowledging clusters that solely belong to a cluster and do not show properties of those with overlapping and hierarchal properties.

TABLE 21. *(Continued.)* **(c) Similarity measures between communities detected and ground data by various algorithms using the Manhattan distance measure, for sts 10 with ground data acquired from K=10.**

(c)

Algorithms	Element-centric	F-Measure	Jaccard
Louvain	0.300362158	0.296296296	0.173913043
*Link Community	0.359660428	0.448230668	0.288851351
Fastgreedy	0.300362158	0.296296296	0.173913043
Infomap	0.300362158	0.296296296	0.173913043
Walktrap	0.300362158	0.296296296	0.173913043
Leading Eigenvector	0.300362158	0.296296296	0.173913043
Label Propagation	0.300362158	0.296296296	0.173913043
Spinglass	0.300362158	0.296296296	0.173913043

VI. CONCLUSION

Given the nature of time series and their vast usage in various fields of science, it is essential to find clustering algorithms that depict the real-life properties of such valuable datasets. In this paper, using the algorithm of link community detection, we introduce a new method that identifies properties of overlapping and hierarchy often found in real life data, by creating a network with the links within the time series and then applying link community detection algorithm to the newly defined network. For this research, data was synthetically produced based on famous time series available and results analyzed by the element-centric measure of similarity have indicated a clear increased accuracy for this algorithm with well-known distance measures, having the highest accuracy when used alongside the DTW distance measure, thus indicating how effective this algorithm can be in analyzing natural time series. The proposed algorithm has also been very successful in identifying and forming communities for both large and small time series, thus solving one of the main issues previous algorithms tend to have.

VII. FUTURE WORK

Building on the foundation laid by prior research and the findings presented in this study, it becomes evident that investigating the topological characteristics of time series and their integration into the methodologies outlined herein holds significant promise. Furthermore, delving into the creation of networks and harnessing topological data analysis (TDA) for such endeavors presents an intriguing avenue for future exploration. To this end, a comprehensive research agenda can be articulated, expanding upon the preliminary concepts introduced:

1. In-depth Study of Topological Properties in Time Series Analysis

Objective: To conduct a comprehensive investigation into how the inherent topological features of time series data influence predictive modelling and analysis. This includes exploring the persistence of certain shapes within the data and how these can be quantified and utilized for improved forecasting accuracy.

Methodology: Employ advanced topological data analysis (TDA) techniques, such as persistent homology, to extract and analyze topological signatures from various time series datasets. This will involve developing metrics for quantifying the topological robustness of time series and assessing their impact on model performance.

Expected Outcomes: The goal is to establish a clear correlation between the topological properties of time series data and their predictability. This could lead to the development of new models or the enhancement of existing ones, which explicitly incorporate topological information for better forecasting.

2. Application of TDA in Network Formation from Time Series

Objective: To explore the potential of applying topological data analysis in understanding and modeling the formation of networks derived from time series data. This involves identifying underlying patterns and structures that lead to the emergence of networks and how these can be leveraged for predictive insights.

Methodology: This research will involve creating networks from time series data, where nodes represent key features or segments of the data, and edges represent their temporal or causal relationships. TDA techniques will be applied to these networks to uncover hidden structures and dynamics. This will include using TDA to analyze the evolution of networks over time and how topological changes correlate with external events or states.

Expected Outcomes: The aim is to develop methodologies that can predict or explain the formation of networks from time series data, providing insights into the dynamics at play. This could have applications in various fields such as finance, where market dynamics are studied, or in climatology, where understanding the interaction between different climate variables is crucial.

3. Integration of TDA with Existing Time Series and Network Analysis Methods

Objective: To integrate TDA with conventional time series and network analysis methods, enhancing their capability to capture complex patterns and structures. This includes the development of hybrid models that combine the strengths of TDA and traditional statistical or machine learning approaches.

Methodology: Develop and test new algorithms that incorporate TDA features into time series forecasting and network analysis models. This will involve comparative studies to benchmark the performance of hybrid models against traditional approaches, with a focus on their ability to handle non-linear and complex data structures.

Expected Outcomes: The development of a new class of models that offer superior performance in analyzing and predicting time series and network phenomena. These models would be particularly valuable in dealing with high-dimensional, complex datasets where traditional methods struggle to capture the underlying dynamics.

4. Cross-Disciplinary Applications and Case Studies

Objective: To apply the developed methodologies and models in a range of real-world scenarios across different domains, demonstrating their versatility and effectiveness in uncovering insights from complex datasets.

Methodology: Conduct case studies in diverse fields such as economics, biology, climate science, and social media analysis, applying the new TDA-enhanced methods to solve specific problems. This will also involve collaborations with experts in these domains to ensure the relevance and applicability of the approaches.

Expected Outcomes: A portfolio of case studies that showcase the utility of incorporating topological properties into time series and network analysis. These examples will not only validate the methodologies but also highlight their potential to contribute to advancements in various scientific and industrial fields.

By pursuing these extended lines of research, the study aims to push the boundaries of how topological data analysis can be harnessed to gain deeper insights into time series and networks, opening new avenues for innovation in data analysis and predictive modelling.

REFERENCES

[1] H. Abbasimehr and F. Sheikh Baghery, "A novel time series clustering method with fine-tuned support vector regression for customer behavior analysis," *Expert Syst. Appl.*, vol. 204, Oct. 2022, Art. no. 117584.

[2] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering—A decade review," *Inf. Syst.*, vol. 53, pp. 16–38, Oct. 2015.

[3] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, Aug. 2010.

[4] B. Alatas and F. Ozbay, "Overlapping community detection in social networks using parliamentary optimization algorithm," *Int. J. Comput. Netw. Appl.*, vol. 2, no. 1, pp. 12–19, 2015.

[5] L. Anghinoni, D. A. Vega-Oliveros, T. C. Silva, and L. Zhao, "Time series pattern identification by hierarchical community detection," *Eur. Phys. J. Special Topics*, vol. 230, nos. 14–15, pp. 2775–2782, Oct. 2021.

[6] A. Bagnall, J. Lines, W. Vickers, and E. Keogh, "The UEA & UCR time series classification repository," Tech. Rep.

[7] G. E. A. P. A. Batista, E. J. Keogh, O. M. Tataw, and V. M. A. de Souza, "CID: An efficient complexity-invariant distance for time series," *Data Mining Knowl. Discovery*, vol. 28, no. 3, pp. 634–669, May 2014.

[8] S. V. Bhavani, L. Xiong, A. Pius, M. Semler, E. T. Qian, P. A. Verhoef, C. Robichaux, C. M. Coopersmith, and M. M. Churpek, "Comparison of time series clustering methods for identifying novel subphenotypes of patients with infection," *J. Amer. Med. Inform. Assoc.*, vol. 30, no. 6, pp. 1158–1166, May 2023.

[9] A. Bryant and K. Cios, "RNN-DBSCAN: A density-based clustering algorithm using reverse nearest neighbor density estimates," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1109–1121, Jun. 2018.

[10] R. J. G. B. Campello, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *WIREs Data Min. Knowl. Discov.*, vol. 10, no. 2, p. e1343, 2020.

[11] D. C. de Lucas, "Classification techniques for time series and functional data Internet version David Casado de Lucas," Ph.D. thesis, 2010.

[12] P. Cimiano, A. Hotho, and S. Staab, "Comparing conceptual, divisive and agglomerative clustering for learning taxonomies from text," in *Proc. 16th Eur. Conf. Artif. Intell.* IOS Press, 2004, pp. 435–439.

[13] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 6, Dec. 2004, Art. no. 66111.

[14] W. E. Donath and A. J. Hoffman, "Algorithms for partitioning graphs and computer logic based on eigenvectors of connection matrices," *IBM Tech. Discl. Bull.*, vol. 15, no. 3, pp. 938–944, 1972.

[15] T. S. Evans and R. Lambiotte, "Line graphs, link partitions, and overlapping communities," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 1, Jul. 2009, Art. no. 16105.

[16] L. N. Ferreira and L. Zhao, "Time series clustering via community detection in networks," *Inf. Sci.*, vol. 326, pp. 227–242, Jan. 2016.

[17] C. Fraley and A. E. Raftery, "How many clusters? Which clustering method? Answers via model-based cluster analysis," *Comput. J.*, vol. 41, no. 8, pp. 578–588, Jan. 1998.

[18] S. Fröhwirth-Schnatter and S. Kaufmann, "Model-based clustering of multiple time series," *J. Bus. Econ. Statist.*, vol. 26, no. 1, pp. 78–89, Jan. 2008.

[19] M. P. B. Gallaugher, S. D. Tomarchio, P. D. McNicholas, and A. Punzo, "Model-based clustering via skewed matrix-variate cluster-weighted models," *J. Stat. Comput. Simul.*, vol. 92, no. 13, pp. 2645–2666, Sep. 2022.

[20] Z.-K. Gao, M. Small, and J. Kurths, "Complex network analysis of time series," *Europhysics Lett.*, vol. 116, no. 5, p. 50001, Dec. 2016.

[21] T. Garrels, A. Khodabakhsh, B. Y. Renard, and K. Baum, "LazyFox: Fast and parallelized overlapping community detection in large graphs," *PeerJ Comput. Sci.*, vol. 9, p. e1291, Apr. 2023.

[22] A. J. Gates, I. B. Wood, W. P. Hetrick, and Y.-Y. Ahn, "Element-centric clustering comparison unifies overlaps and hierarchy," *Sci. Rep.*, vol. 9, no. 1, pp. 1–13, Jun. 2019.

[23] X. Golay, S. Kollias, G. Stoll, D. Meier, A. Valavanis, and P. Boesiger, "A new correlation-based fuzzy logic clustering algorithm for FMRI," *Magn. Reson. Med.*, vol. 40, no. 2, pp. 249–260, Aug. 1998.

[24] L. Huang, G. Wang, Y. Wang, E. Blanzieri, and C. Su, "Link clustering with extended link similarity and EQ evaluation division," *PLoS ONE*, vol. 8, no. 6, Jun. 2013, Art. no. e66005.

[25] H. Izakian, W. Pedrycz, and I. Jamal, "Fuzzy clustering of time series data using dynamic time warping distance," *Eng. Appl. Artif. Intell.*, vol. 39, pp. 235–244, Mar. 2015.

[26] E. Kafeza, G. Rompolas, S. Kyriazidis, and C. Makris, "Time-series clustering for determining behavioral-based brand loyalty of users across social media," *IEEE Trans. Computat. Social Syst.*, 2022.

[27] G. Karypis, E.-H. Han, and V. Kumar, "Chameleon: Hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp. 68–75, 1999.

[28] A. S. Hadi, L. Kaufman, and P. J. Rousseeuw, "Finding groups in data: An introduction to cluster analysis," *Technometrics*, vol. 34, no. 1, p. 111, Feb. 1992.

[29] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: A survey and empirical demonstration," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY, USA: Association for Computing Machinery, Jul. 2002, pp. 102–111.

[30] Y. Kim and H. Jeong, "Map equation for link communities," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 84, no. 2, Aug. 2011, Art. no. 26110.

[31] A. Kumar, D. Barman, R. Sarkar, and N. Chowdhury, ‘‘Overlapping community detection using multiobjective genetic algorithm,’’ *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 3, pp. 802–817, Jun. 2020.

[32] A. Lancichinetti and S. Fortunato, ‘‘Community detection algorithms: A comparative analysis,’’ *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 5, Nov. 2009, Art. no. 56117.

[33] H. Li, T. Du, and X. Wan, ‘‘Time series clustering based on relationship network and community detection,’’ *Expert Syst. Appl.*, vol. 216, Apr. 2023, Art. no. 119481.

[34] H. Li, X. Wu, X. Wan, and W. Lin, ‘‘Time series clustering via matrix profile and community detection,’’ *Adv. Eng. Informat.*, vol. 54, Oct. 2022, Art. no. 101771.

[35] M. Li and J. Liu, ‘‘A link clustering based memetic algorithm for overlapping community detection,’’ *Phys. A, Stat. Mech. Appl.*, vol. 503, pp. 410–423, Aug. 2018.

[36] F. Lorrain and H. C. White, ‘‘Structural equivalence of individuals in social networks,’’ *J. Math. Sociol.*, vol. 1, no. 1, pp. 49–80, Jan. 1971.

[37] T. Lyu, L. Bing, Z. Zhang, and Y. Zhang, ‘‘FOX: Fast overlapping community detection algorithm in big weighted networks,’’ *ACM Trans. Social Comput.*, vol. 3, no. 3, pp. 1–23, Sep. 2020.

[38] D. Maldeniya, M. De Choudhury, D. Garcia, and D. M. Romero, ‘‘Pulling through together: Social media response trajectories in disaster-stricken communities,’’ *J. Comput. Social Sci.*, vol. 6, no. 2, pp. 655–706, Oct. 2023.

[39] G. J. McLachlan, R. W. Bean, and D. Peel, ‘‘A mixture model-based approach to the clustering of microarray expression data,’’ *Bioinformatics*, vol. 18, no. 3, pp. 413–422, Mar. 2002.

[40] P. D. Mcnicholas, ‘‘Model-based clustering,’’ *J. Classification*, vol. 33, pp. 331–373, Nov. 2016.

[41] V. Melnykov, S. Sarkar, and Y. Melnykov, ‘‘On finite mixture modeling and model-based clustering of directed weighted multilayer networks,’’ *Pattern Recognit.*, vol. 112, Apr. 2021, Art. no. 107641.

[42] C. S. Müller-Levet, F. Klawonn, K. H. Cho, and O. Wolkenhauer, ‘‘Fuzzy clustering of short time-series and unevenly distributed sampling points,’’ in *Advances in Intelligent Data Analysis V* (Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 2810, 2003, pp. 330–340.

[43] M. A. Muda, D. D. Prastyo, and M. S. Akbar, ‘‘Clustering stock prices of industrial and consumer sector companies in Indonesia using fuzzy C-means and fuzzy C-medoids involving ACF and PACF,’’ in *Soft Computing in Data Science*, M. Yusoff, T. Hai, M. Kassim, A. Mohamed, E. Kita, Eds. Singapore: Springer, 2023, pp. 275–290.

[44] F. Murtagh and P. Contreras, ‘‘Algorithms for hierarchical clustering: An overview,’’ *WIREs Data Mining Knowl. Discovery*, vol. 2, no. 1, pp. 86–97, Jan. 2012.

[45] M. E. J. Newman and M. Girvan, ‘‘Finding and evaluating community structure in networks,’’ *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, Feb. 2004, Art. no. 26113.

[46] G. K. Orman, V. Labatut, and H. Cherifi, ‘‘Qualitative comparison of community detection algorithms,’’ 2012, *arXiv:1207.3603*.

[47] C. Pizzuti, ‘‘Overlapped community detection in complex networks,’’ in *Proc. 11th Annu. Conf. Genetic Evol. Comput.* New York, NY, USA: ACM, Jul. 2009, pp. 859–866.

[48] P. Pons and M. Latapy, ‘‘Computing communities in large networks using random walks,’’ in *Computer and Information Sciences—ISCIS*, P. Yolum, T. Güngör, F. Gürgen, and C. Özturan, Eds. Berlin, Germany: Springer, 2005, pp. 284–293.

[49] J. Qiu and Y. Hu, ‘‘Discovering the overlapping and hierarchical community structure in a social network,’’ in *Knowledge Science, Engineering and Management* (Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2013, pp. 252–262.

[50] A. E. Raftery and N. Dean, ‘‘Variable selection for model-based clustering,’’ *J. Amer. Stat. Assoc.*, vol. 101, no. 473, pp. 168–178, 2006.

[51] S. Rani and G. Sikka, ‘‘Recent techniques of clustering of time series data: A survey,’’ *Int. J. Comput. Appl.*, vol. 52, no. 15, pp. 1–9, Aug. 2012.

[52] J. Reichardt and S. Bornholdt, ‘‘Statistical mechanics of community detection,’’ *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 74, Jul. 2006, Art. no. 16110.

[53] M. Rosvall, J. C. Delvenne, M. T. Schaub, and R. Lambiotte, ‘‘Different approaches to community detection,’’ in *Advances in Network Clustering and Blockmodeling*, 2019, pp. 105–119.

[54] S. Salvador and P. Chan, ‘‘Toward accurate dynamic time warping in linear time and space,’’ *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, Oct. 2007.

[55] A. Sardá-Espinosa, ‘‘Time-series clustering in R using the dtwclust package,’’ *R J.*, vol. 11, no. 1, pp. 1–22, 2019.

[56] K.-K. Shang, M. Small, Y. Wang, D. Yin, and S. Li, ‘‘A novel metric for community detection,’’ *Europhysics Lett.*, vol. 129, no. 6, p. 68002, Apr. 2020.

[57] C. Shi, Y. Cai, D. Fu, Y. Dong, and B. Wu, ‘‘A link clustering based overlapping community detection algorithm,’’ *Data Knowl. Eng.*, vol. 87, pp. 394–404, Sep. 2013.

[58] M. Small, D. Yu, and R. G. Harrison, ‘‘Surrogate test for pseudoperiodic time series data,’’ *Phys. Rev. Lett.*, vol. 87, no. 18, Oct. 2001, Art. no. 188101.

[59] S. Subha and J. G. R. Sathiaseelan, ‘‘Effective anomaly detection approach to classify noisy data using robust noise detection and removal technique in IoT healthcare data,’’ *Social Netw. Comput. Sci.*, vol. 4, no. 5, p. 522, Jul. 2023.

[60] A. Tiwari, R. Sam, and S. Shaikh, ‘‘Analysis and prediction of churn customers for telecommunication industry,’’ in *Proc. Int. Conf. I-SMAC (IoT Social, Mobile, Anal. Cloud) (I-SMAC)*, Feb. 2017, pp. 218–222.

[61] V. A. Traag, L. Waltman, and N. J. van Eck, ‘‘From Louvain to Leiden: Guaranteeing well-connected communities,’’ *Sci. Rep.*, vol. 9, no. 1, p. 5233, Mar. 2019.

[62] M. Vlachos, G. Kollios, and D. Gunopulos, ‘‘Discovering similar multidimensional trajectories,’’ in *Proc. 18th Int. Conf. Data Eng.*, 2002, pp. 673–684.

[63] X. Wang, K. Smith, and R. Hyndman, ‘‘Characteristic-based clustering for time series data,’’ *Data Mining Knowl. Discovery*, vol. 13, no. 3, pp. 335–364, Sep. 2006.

[64] T. Warren Liao, ‘‘Clustering of time series data—A survey,’’ *Pattern Recognit.*, vol. 38, no. 11, pp. 1857–1874, Nov. 2005.

[65] F. Y. Wu, ‘‘The Potts model,’’ *Rev. Mod. Phys.*, vol. 54, pp. 235–268, Jan. 1982.

[66] Y. Xiong and D.-Y. Yeung, ‘‘Time series clustering with ARMA mixtures,’’ *Pattern Recognit.*, vol. 37, no. 8, pp. 1675–1689, Aug. 2004.

[67] Z. Yang, R. Algesheimer, and C. J. Tessone, ‘‘A comparative analysis of community detection algorithms on artificial networks,’’ *Sci. Rep.*, vol. 6, Aug. 2016, Art. no. 30750.

[68] H. Zhang, T. B. Ho, Y. Zhang, and M. S. Lin, ‘‘Unsupervised feature extraction for time series clustering using orthogonal wavelet transform,’’ *Informatica*, vol. 30, no. 3, pp. 305–319, 2006.

[69] J. Zhang and M. Small, ‘‘Complex network from pseudoperiodic time series: Topology versus dynamics,’’ *Phys. Rev. Lett.*, vol. 96, no. 23, Jun. 2006, Art. no. 238701.

[70] X. Zhang, J. Liu, Y. Du, and T. Lv, ‘‘A novel clustering method on time series data,’’ *Expert Syst. Appl.*, vol. 38, no. 9, pp. 11891–11900, Sep. 2011.

[71] X. Zhou, Y. Liu, J. Wang, and C. Li, ‘‘A density based link clustering algorithm for overlapping community detection in networks,’’ *Phys. A, Stat. Mech. Appl.*, vol. 486, pp. 65–78, Nov. 2017.

YASAMIN GHAHREMANI received the B.S. degree in industrial engineering from Sharif University of Technology, Iran, and the M.S. degree in system engineering from Iran University of Science and Technology. Her research interests include system engineering, data science, and machine learning.

BABAK AMIRI received the Ph.D. degree in information technology from The University of Sydney, in 2014. He is currently an Assistant Professor with Iran University of Science and Technology. He has a track record of publishing quality research articles in respected outlets within the fields of complex systems, machine learning, and data science. Notably, the academic community has well-received his research on topics, such as social media analysis, complex networks, customer analytics, and text analysis.