An Ensemble Solution for Multivariate Time Series Clustering

Iago Vázquez[a], José R. Villar[b,*], Javier Sedano[a], Svetlana Simić[c], Enrique de la Cal[b]

[a] *Instituto Tecnológico de Castilla y León, Burgos, Spain*
[b] *Computer Science Department, University of Oviedo, Oviedo, Spain*
[c] *Department of Neurology, University of Novi Sad, Republic of Serbia*

Abstract

Technologies such as Big Data and IoT have shown the need for intelligent unsupervised processing of Multivariate Time Series (MTS), MTS clustering among them. The challenges in MTS clustering includes not only the selection of the algorithm but also the MTS representation and the similarity measurement among the instances. This study proposes an ensemble of MTS clustering methods that merges different MTS representations and distance functions, aggregating them to obtain a similarity measurement. Furthermore, a proposal for prior knowledge representation is proposed to balance the aggregation of the distances. The final clustering is performed either using k-means or hierarchical clustering.

The experimentation set up includes the implementation of the ensemble with either 4 or 5 different methods, including an MTS extension of k-Shape. The results show that the ensemble is biased towards the best methods, which helps the clustering practitioner in the selection of the most suitable prototypes. Moreover, the evaluation of the ensemble with the number of clusters set to the number of labels shows that metrics, such as the sensitivity and specificity, must drive the rule of the elbow; alternatively, this value represents the most interesting prior knowledge bit in MTS clustering.

*Corresponding author

Email addresses: iago.vazquez@itcl.es (Iago Vázquez), villarjose@uniovi.es (José R. Villar), javier.sedano@uniovi.es (Javier Sedano), svetlana.simic@mf.uns.ac.rs (Svetlana Simić), delacal@uniovi.es (Enrique de la Cal)

Further work includes the study of digital markers to compare MTS representations and distance functions and the use of external metrics to balance the aggregation of the methods.

Keywords: Multi-variate Time Series, Clustering, Prior knowledge, Ensemble of clustering

---

1. Introduction

Technologies, such as Big Data, IoT and Industry 4.0, have provide the capability to gather immense volumes of data; the main part of them comes from sensory data related to a concrete problem. These sensory data include several variables and time series are gathered for each of these variables; the combination of several variables, each being a time series is known as Multivariate Time Series (MTS). Therefore, MTS has regained the focus of the research community with the effervescence of the technologies mentioned before; MTS instance clustering (for short, MTS clustering) is one of the most challenging topics regarding MTS [1, 2, 3], either with unsupervised solutions or semi-supervised approaches [3, 4].

Nevertheless, the performance of the MTS clustering techniques varies to the considered problem or data set, affecting the confidence of the user on the obtained results. In order to cope with this situation, this research proposes ensemble of MTS clustering methods. The aim is not to generate a competitive method to compare with other state-of-the-art clustering algorithms; instead, the idea is to provide the experimenter with a tool to merge the results from different algorithms. This idea is not new and has been applied to MTS classification [5]. To group the MTS instances, this study chooses several MTS clustering methods in parallel, converts their outcomes into adjacency matrices, and aggregates the obtained results. An optional prior knowledge helps in tuning the clusters obtained so far or in the evaluation of the produced instance clusters. The solution is evaluated with several data sets from the community, showing a balanced performance in both clustering processes. Interestingly, this study extends kShape [6], a well-known and highly competitive univariate time series clustering, to deal with MTS instances.

Therefore, the main novelties in this study are:

- A proposal for an ensemble of MTS clustering is detailed with the aim of obtaining a suitable performance independently of the problem.

- The ensemble includes a wide range of TS representations and distance measurements.
- The ensemble merges the outcome from the different clustering algorithm runs using graph representation and by aggregation the graphs.
- Prior knowledge can be introduced to modulate the fusion of the different methods according to their performance.
- An MTS extension of k-Shape [6] to cope with MTS instances.

The structure of the paper is as follows. The next Section introduces the related work on the topic, while Section 3 completely describes the proposal for multi-dimensional MTS instances clustering and the ensemble of the clustering algorithms. Section 4 includes the description of the experimentation and the MTS data sets, as well as the MTS extension for k-Shape. Section 5 is devoted to the results and the discussion. Finally, the conclusions are drawn.

2. Related Work

The MTS clustering problem refers to arranging MTS instances in suitable groups so that the instances within a group share several relevant behaviours and properties. Many different methods address the MTS clustering, typically including a MTS representation and suitable distance function plus a clustering algorithm. As an example, the study in [7] proposed the Haar wavelet transformation and the k-means algorithm in the design of an ambient-air vaporizer under time-series weather conditions. The work in [8] proposed unsupervised shapelets search to differentiate MTS subsets that include common distinctive sub-sequences and the k-means algorithm, in order to create groups accordingly.

Distance measurement functions are one of the most focused research topics when dealing with MTS clustering as these functions are the base on which many clustering algorithms rely [9]. By far, Dynamic Time Warping (DTW) [10, 11, 12] is still a reference as a general distance measurement among univariate and multivariate TS, although many different alternatives have been studied. For instance, [13] proposed two distance measurements considering the magnitude and phase shift of relevant TS points according to the Pearson correlation; the second of these distances is claimed to produce similar or better clustering results than DTW. A different enhancement to DTW was presented in [14] introducing the so-called complexity-invariant distance measurement (CID). Estimated quantile auto-covariances have been proposed as a distance measurement in partitional clustering of time series, taking advantage of the ability of these correlation values to retain relevant dynamic features [15]. Finally, motif and minimum description length were proposed to group sub-sequences of time series in a process of active learning based time series segmentation in [16]. Interested readers can focus on [14, 17, 18] for further information.

MTS from photovoltaic array systems were clustered using DTW and k-means in [19]. Besides, the study in [20] proposed a hybrid distance measurement (combining the PCA similarity index and the average-based Euclidean distance) and the Fuzzy C-means clustering algorithm to group the MTS instances. PCA and a modification of the k-means algorithm have been applied to MTS clustering in [21] by projecting to the new coordinate space and then reassigning the MTS to each cluster. Fotso et al [22] proposed the u-shapelets transformation to group time series using a specific distance measurement based on eigenvector decomposition and the comparison of the autocorrelation matrices.

Hierarchical clustering [18] was applied in [23] to the clustering time series using CID distance measurement. Similarly, [24] proposed hierarchical clustering and a Fuzzy extension to DTW to the grouping of MTS; the aggregation of linear Fuzzy information granules on segments of the time series is used to modify the clusters found so far. Recently, a weighted features based clustering was proposed in [25]. In this study, DTW and shape-based distance (SBD) [6] were used to modify the fuzzy membership distance matrix according to the distortion between MTS and the difference in shape, correspondingly. Fuzzy c-means was used to cluster the fuzzy membership matrices. Paparrizos et al proposed the k-Shape and the k-MultiShapes time series clustering methods [6], the two algorithms are based on k-means and rely on two shape-based methods to calculate the centroids of time series. Interestingly, k-Shape and k-MultiShapes are both scalable and with low computational restrictions, making them suitable for high dimensional problems.

In [26, 27], recurrent Neural Networks were used to measure the similarity either between TS within the same MTS instance or between TS belonging to different MTS instances. The results show promising performance, despite the high time complexity. Furthermore, in [28], the authors proposed

Gaussian Mixture Model kernels, augmented with prior distributions learned from randomly selected features, to find similarities between MTS instances with missing data. The outcomes of each Gaussians Mixture Model are then combined using a kernel similarity matrix. In [29], a regression model-based clustering is proposed: a polynomial regression model is used to determine the most relevant variables that are later used in the clustering of the data.

On the other hand, semi-supervised MTS clustering makes use of feedback from the user in the form of pairing related instances [4]. This latter study proposed a semi-supervised MTS hierarchical clustering method by extending the COBRAS (COnstraint-Based Repeated Aggregation and Splitting) algorithm to deal with MT either using k-Shape or DTW as distance functions. Besides, [30] analyzed the performance of different MTS distance measurements when used in spectral clustering, namely, Normalized Cut (NCut). The authors proposed the use of a combination of the distance measurements and some information concerning the relationships among the instances to produce a semi-supervised MTS spectral clustering. Two options were evaluated when combining the distances measurements: a linear combination of the distances measurements and hybrid bipartite graph formation.

Interestingly, MTS clustering performance relies on the performance of the different components in the design of the solution, having these decisions a high impact on the quality of the obtained groups. Consequently, the practitioner should deal with the outcomes and decide which one is the most interesting. From his/her point of view, a tool merging the outcomes from the different methods would facilitate the decision-making process.

3. An MTS Clustering Ensemble with Prior Knowledge Assessment

In this study, we propose an ensemble of several MTS clustering algorithms to assist the practitioner in selecting the best option among the methods. This ensemble includes several different MTS representation and distance measurement functions, makes use of a clustering algorithm, and, finally, merges the outcomes of the different runs through the agglomeration of the produced graphs. Additionally, the user can provide prior knowledge to empower the runs according to their accomplishment with this knowledge, modifying the aggregation. A general overview of this proposal is included in Fig. 1; the process of running the different clustering algorithms and obtaining a final clustering solution is called, from here in after, Multi-dimensional

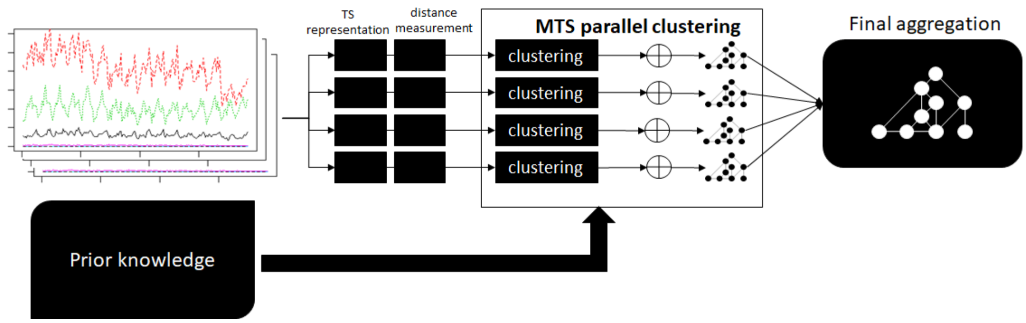

Figure 1: General Overview of the proposal. The MTS instances are represented using several techniques and clustered using a specific distance measurement function. A final merging stage aggregates the outcomes from each part. The prior knowledge modifies the weights of the methods according to their compliance with the given restrictions.

MTS Instance Clustering (MIC).

This section firstly introduces the different TS representations and the corresponding distance measurements; the reasons for their selection are also explained. Next, the algorithm for clustering the MTS instances is selected. The prior knowledge's representation is detailed afterward. Finally, the description of the ensemble is described.

3.1. Time Series Representation and Distance Measurements Functions

From now on, we call raw data the original MTS sequences. An MTS instance (ts^i) can be written as $ts^i =< \overline{x}^i_1, \overline{x}^i_2, \cdots, \overline{x}^i_M >$, where $i \in [1, L]$ is the instance position, L is the number of MTS instances and M is the number of variables. Moreover, each $\overline{x}^i_m$ represents an univariate time series, that is, $\overline{x}^i_m =< x^i_{m1}, x^i_{m2}, \cdots, x^i_{mN} >$, where N is the number of samples, m refers the variable and i the MTS instance's index within the data set. We assume an MTS data set as a collection of instances of raw MTS with arbitrary length. Note that we can store MTS for which the variables have different sampling rates provided there are some timestamps where all the sampling of all the variables coincide in time [31, 32] using polynomial interpolation. Besides, a long MTS is expected to be split in different instances; automatic segmentation of MTS can be employed in these cases to produce the set of suitable instances [24, 31, 33].

In this study, we have chosen several TS representation techniques and distance measurements. The selection of these techniques was based on including MTS representation and distance measurements from a very different nature but still competitive in the literature, to ensure broad coverage of techniques. The point is that if this proof of concept works, then the idea can be also extended to analyze different MTS clustering solutions. Additionally, k-Shape [6] has been extended to MTS and added to the ensemble to evaluate this latter assumption.

The combination of TS representation and distance measurements considered in this study are:

- *TS representation in the time domain*: **Adaptive Piecewise Constant Approximation (APCA)** [34], where each variable j in a raw TS is represented by M segments $APCA(ts^{ij})=\{< v_1^{ij}, p_1^{ij} >, \cdots, < v_M^{ij}, p_M^{ij} >\}$. The coefficients v_k^{ij} are the mean of the values of variable j in ts_i in the interval $[p_{k-1}^{ij}, p_k^{ij}]$, with $p_0^{ij} = 0$. The limits of the intervals are computed with the Haar Discrete Wavelet Transform [35].

  Three different pairs of representation and distance measurements are used: MINDIST (the distance function defined in the study with the definition of APCA), DTW and SBD (Shape-Based Distance, [6]). We combine APCA and SBD because this latter's complexity grows with the length of the time series; APCA acts as reducing the overall complexity, allowing to apply this method to a wider variety of problems.

- *TS representation in the frequency domain*: **Fast Fourier Transform (FFT)** [36, 37] computed on the z-scored raw data, limiting the coefficients to the 10 components. The distance between two univariate TS is measured with the Energy of the differences between them.

- *TS represented with the raw data*, measuring the distance among two TS with the Compression-based dissimilarity measure (**CDM**) [31]. To overcome with the problem of TS of different length, the longer TS is windowed and the CDM is averaged. Let a be the length of the shorter TS instance, then the window is of size a and the shift is $a/2$, padding the end of the longest TS with the last sample. We consider two TS of similar length whenever the differences in length do not surpass the 1.5 ratio.

In this study, we choose to aggregate the distances among all the variables to obtain the distance between two MTS instances. Given two instances ts^i and ts^j, the aggregated distance is calculated using Eq. 1, where $d(x_m^i, x_m^j)$ is the distance between variables x_m^i and x_m^j.

$$D(ts^i, ts^j) = \sqrt{\sum_{m=1}^{M} d^2(x_m^i, x_m^j)} \tag{1}$$

3.2. MTS Instance Clustering

MTS clustering aims to group similar instances of the MTS data set considering all the variables. Each pair of TS representation and distance measurement is then clustered. Although the clustering algorithm applied to each pair may vary, in this research we kept the same clustering algorithm for all the cases. Furthermore, we evaluate and compare the performance of the ensemble using two clustering algorithms: on the one hand, the well-known k-means algorithm; on the other hand, hierarchical clustering (hclust) algorithm.

While hclust represents the natural clustering method (it has already been used with APCA and MINDIST or DTW and with FFT [34], or in conjunction with CMD [31]), the use of k-means needs some justification. The wide use of k-means in the literature [8, 34, 38] and its successful performance are the main reasons to consider this clustering algorithm. Nevertheless, there is an increase in the total amount of time of the algorithms due to the k-means variability forces to repeat the clustering a predefined number of times to average the results (see Section 3.4).

Table 1 shows the different acronyms used in this research for the combination of clustering algorithm with the TS representation and distance measurement. It is worth to notice that, from now on, we call our complete approach MIC-k-means or MIC-hclust accordingly with the clustering method, including the merging procedure and the prior knowledge.

3.3. Prior Knowledge Representation

The idea of prior knowledge is to avoid the ensemble to bias towards those methods that perform better. The prior knowledge represents relationships between MTS instances that the clustering practitioner knows beforehand; therefore, this knowledge can feed the system before running the MTS clustering. In this study, we have considered two possible sources of prior knowledge: on the one hand, there are some MTS instances that the practitioner knows should be in the same cluster because they are highly related. On the other hand, there are some MTS instances that the practitioner knows should belong to different clusters because they never co-occur or are highly unrelated.

Table 1: Acronyms of the different combinations of TS representation, distance measurement and clustering algorithm. The aggregation of the results from each clustering algorithm produces the final clustering: MIC-k-means and MIC-hclust. When developing the MIC with k-means, each method is repeated a predefined number of times and the aggregation of the distances is computed.

	clustering method	
	k-means	hclust
APCA & MINDIST	k-A-MIN	h-A-MIN
APCA & DTW	k-A-DTW	h-A-DTW
APCA & SBD	k-A-SBD	h-A-SBD
FFT & Energy	k-FFT	h-FFT
Raw data & CMD	k-CMD	h-CMD
	MIC-k-means	MIC-hclust

We use a list of constraints to represent these knowledge items; a clustering solution would have higher reliability with the increasing number of accomplished constraints. Each constraint is a list of signed indexes with the following meaning:

- Each index refers to an specific MTS instance.
- Positive indexes in a constraint mean these MTS instances should be grouped together.
- A negative index in a constraint means this MTS instance should not be grouped together with the positive MTS instances contained in the constraint.
- A constraint contains at least 2 positive indexes.

An example of prior knowledge is depicted in Table 2. In this study, we limit the number of constraints to a percentage of the number of instances. We keep this percentage rather small; otherwise, it might imply that the knowledge about the problem is enough to move directly into classification instead of clustering. Additionally, we limit the number of positive indexes and the number of negative indexes to a percentage of the number of instances as well.

Table 2: An example of the prior knowledge representation.

[[1, 2, 3, -4, -5], [1, 7, 8], [1, 2, -9, -10]]

$[1, 2, 3, -4, -5]$	Instances 1, 2 and 3 shall be grouped together, but neither instance 4 nor 5 should. Nothing is said about the relationship between 4 and 5.
$[1, 7, 8]$	Instances 1, 7 and 8 shall be grouped in the same cluster.
$[1, 2, -9, -10]]$	Instances 1 and 2 shall be grouped together, but neither instance 9 nor 10 should. Nothing is said about the relationship between 9 and 10.

3.4. Merging the Different Methods

MIC includes 5 different stages to merge all the methods: i) Initialization, ii) Running each clustering method, iii) Prior Knowledge assessment, iv) Aggregation of the clustering alternatives and v) Performing the final clustering.

The *Initialization* stage receives all the parameters, such as the number of clusters K, the internal MIC clustering algorithm or the prior knowledge to use. Besides, Algorithm 1 gives details of the second stage. Ten repetitions of k-means are considered due to its variability when this method is selected. The output is a list containing an $L \times L$ similarity matrix for each TS representation and distance measurement. In the case of hclust, a similarity matrix's cell $< i, j >$ contains 1 if the instances i and j belong to the same cluster; otherwise, the cell contains 0. In the case of k-means, each cell contains the average of the number of times that the corresponding instances have been grouped in the same cluster among the repetitions.

Algorithm 1 presents the assessment due to prior knowledge. Its outcome is a vector containing the weights to the similarity matrices, which are adjusted according to the number of constraints its corresponding matrix satisfies.

The final stage is to produce the final clustering; this clustering makes use of the aggregation of the similarity matrices using the weights. In this study, we propose the following clustering options:

- Extended spectral clustering to manage similarity matrices with real values in the interval [0.0, 1.0] [39], computing the generalized spectrum

Algorithm 1: Running each clustering method

Input:
the MTS data set;
the number of clusters K;
the MIC clustering method: either k-means or hclust;
Result:
The clustering-results list with the similarity for each TS representation and distance measurement;

```
if MIC uses k-means then
    numrep ← 10;
else
    numrep ← 1;
end
clustering-results ← empty list ;
foreach TS representation and distance measurement do
    rep-results ← empty list;
    M ← Compute the distance matrix for the MTS data set;
    for i from 1 to numrep do
        rep-results[i] ← cluster using the distances in M
    end
    sim-matrix ← Average-Repetitions(rep-results);
    clustering-results.append(sim-matrix);
end
```

Algorithm 2: Assessment using the prior knowledge

```
Input:
The list SM with the similarity matrices, one per TS representation
and distance measurement;
The list APK containing the prior knowledge;
Result:
The vector weights, same length as SM;
Let weights be an empty vector;
Let sizeAPK be the number of restrictions in APK;
Let sizeSM be the length of the SM;
if sizePAK is empty then
  | return a vector containig sizeSM repetitons of 1/sizeSM;
end
foreach s ∈ 1 : sizeSM do
  | v ← 0;
  | foreach constraint R in APK do
  |   | holdR ← TRUE;
  |   | foreach cell < i, j > in SM do
  |   |   | if instances i and j have been grouped together then
  |   |   |   | if R contains i and j AND R does not hold then
  |   |   |   |   | holdR ← FALSE;
  |   |   |   | end
  |   |   | end
  |   | end
  |   | if holdR is TRUE then
  |   |   | v ← v + 1/sizeAPK;
  |   | end
  | end
  | weights.append(v) ;
end
return weights/∑weights;
```

of the similarity matrix and choosing the eigenvalues whose values are higher than the median. We limit this number to the K most significant eigenvectors. Finally, hclust produces the clusters. We identify this option with sub-index $_{CUT}$.

- Transforming the similarity matrix into an adjacency matrix using the 0.5 threshold and performing the clustering.The sub-index $_{0.5}$ identifies this option.
- Similar to the previous option but allowing uncertainty: those similarity values higher than 0.6 become a link, those smaller than 0.4 represent no link. Values in between are either linked and not linked. Sub-index $_{UNC}$ identifies this option.

4. Experimental Set Up

4.1. The Experimentation Data Sets

This study makes use of the labeled MTS data sets enumerated in Table 3, all of them available at the *Time Series Classification* site [40]. The reason for using labeled MTS data is because it allows us to evaluate the performance of the clustering algorithms.

4.2. The Experiment Set Up

This study focuses on how MIC can assist the clustering practitioner in the selection of a final solution; therefore, the performance metrics are not so important as the guidelines MIC produces, biasing towards the best solutions. Therefore, this study compares MIC versus each of the individual methods that it merges: On the one hand, the three approaches in [34]: h-A-MIN, h-A-DTW, and h-FFT. On the second hand, h-CMD [31] and an extension of k-Shape [6] to MTS problems; the next Subsection focused on detailing this latter method.

We perform the analysis using two situations: using an ensemble with the four first methods and, secondly, using an ensemble with the five methods. This allows us to evaluate the changes and improvements in the ensemble's outcome due to the incorporation of a new method, which might work better or worse than those already considered.

We set the value of K (the desired number of clusters) in two ways: On the one hand, we use the rule of the elbow to select the best K value. On the

Table 3: MTS data sets used in this research. All of them are available at the *Time Series Classification* site [40].

MTS Data set	Description
Articulary Word Recognition (AWR) [41, 42]	25 train and 25 test instances of 12 variables, each with 143 samples belonging to 25 different words.
Cricket (Cr)[42, 43]	Records the movements of the hands of 4 cricket umpires using accelerometers. A total of 12 classes, with 6 variables and 1197 samples each per instance. The data set includes 108 train instances and 72 test instances.
Epilepsy (EP)[44]	This data set includes tri-axial accelerometer data recorded for several Activities of Daily Living and simulated Epileptic seizures (up to 4 different labels). The data set includes 137 train instances and 128 test instances. Each instance includes 3 variables and 206 samples.
Finger Movements (FM) [45]	This data set has a correspondence to Benjamin Blankertz for the BCI II competition (Data set IV). The data set includes 316 train instances and 100 test instances, two possible labels. Each instance includes 28 variables, 50 samples each.
HeartBeat (HB) [46, 47]	This data set is derived from the PhysioNet/CinC Challenge 2016. The data set includes 61 instance for training and 61 for testing, two possible labels. Each instance has 61 variables and 405 samples.

other hand, we evaluate the methods with the value of K set to the number of labels.

Having used labeled MTS instances, we measure the performance of the methods with the Sensitivity (SEN) and the Specificity (SPE) metrics (see Eq. 2 and Eq. 3, respectively). Algorithm 3 computes the counters True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

$$SEN = \frac{TP}{TP + FN} \tag{2}$$

$$SPE = \frac{TN}{FP + TN} \tag{3}$$

Algorithm 3: Calculation of the performance counters.

Let n be a matrix of zeros with size $L \times 4$ for the counters;
foreach *instance i from 1 to L* **do**
  foreach *instance j from i to L* **do**
    if *i and j have same labels AND are grouped in the same cluster* **then**
      $n[i, TP] + +$; $n[j, TP] + +$;
    else if *i and j have same labels AND are grouped in different clusters* **then**
      $n[i, FN] + +$; $n[j, FN] + +$;
    else if *i and j have different labels AND are grouped in the same cluster* **then**
      $n[i, FP] + +$; $n[j, FP] + +$;
    else `/* i and j have different labels AND are grouped in different clusters */`
      $n[i, TN] + +$; $n[j, TN] + +$;
  end
end
columnSum $\leftarrow$ sum n by column;
return $\frac{1}{L} \times columnSum$;

Finally, we make use of the following prior knowledge options:

- The number of constraints (NR) takes the following possible values 0%, 5%, and 10% of the number of MTS instances.

- Both the number of positive indexes (nPIR) and the number of negative indexes (nNIR) within a constraint take the value of 5% of the number of MTS instances.
- The constraints were randomly generated introducing a variable number of positive instances (from 2 to nPIR) and a variable number of negative instances (0 to nNIR).
- The same prior knowledge rule set was applied in all the experimentation for comparison purposes.

4.3. An MTS extension for k-Shape

The high performance reported for k-Shape [6] with univariate time series suggested that its behavior with MTS instances would also be remarkable. A simple extension is proposed by i) designing an MTS-enabled SBD distance measurement (mSBD) and ii) developing an MTS Shape Extraction (mSE). We keep the same k-Shape clustering algorithm but using the two previous extended tools instead. From now on, this method is referred as k-A-Sp.

Each univariate time series within an MTS instance is represented with APCA. We analysed two percentages of reduction for APCA: 10% (k-A-$\mathrm{Sp}_{10\%}$) and the $\sqrt{N}$% (k-A-$\mathrm{Sp}_{\sqrt{(N)}}$), both calculated on the number of samples N; we keep the latter one as the best case in the subsequent ensembles. Afterwards, we propose to calculate the mSBD as the square root of the sum of the squares of the SBD for each variable (Eq. 4), where $SBD_m(i,j)$ represents the value of SBD for instances i and j computed on variable m.

$$mSBD(i,j) = \sum_{m=1}^{M} SBD_m(i,j) \tag{4}$$

Besides, the mSE generates the MTS centroids per cluster as the aggregation of the eigenvectors, one from each variable. Each eigenvector is obtained using the Shape Extraction method in [6].

5. Results and Discussion

The following tables display the obtained results: i) Table 4 shows the figures obtained without prior knowledge and finding the number of clusters with the rule of the elbow, ii) Table 5 presents the obtained numbers when using prior knowledge and setting the best number of clusters with the rule of the elbow, and iii) Table 6 shows the results obtained using the number of labels as the number of clusters.

There are several important findings from the results in Table 4. Firstly, the MTS clustering methods performed quite differently. On the one hand, h-A-MIN, h-A-DTW and h-FFT produced good results for the AWR, CR and HB data sets either concerning the number of clusters or the metrics SEN and SPE. However, the SPE values with the HB data set are surprisingly low; that is, the MTS instances with the same label were assigned to practically all the different groups.

On the other hand, the performance of the three remaining methods (namely, h-CMD, k-A-Sp$_{10\%}$ and k-A-Sp$_{\sqrt{(N)}\%}$) was really poor for the two metrics, and, even though the best number of clusters were suitable, the MTS instances were misled among all the groups. We consider the k-A-Sp$_{\sqrt{(N)}\%}$ in the ensembles due to its slightly better metric values.

It is worth mentioning the poor performance of the k-A-Sp with the two percentages of reduction. This issue was further studied, comparing the k-A-Sp and the k-Shape with univariate time series included in [6]. The results, which have not been included in this research for the sake of simplicity, suggested that the APCA reduction loses too much covariance information, penalizing the capacity to detect the correct relationships among the instances.

Table 4 also includes the mean of the metrics considering 4 or 5 methods for the sake of clarity and comparison reasons when moving forward to the results of the ensembles. For the AWR, CR and EP data sets, the performance of the ensembles was really good, not only having better metrics than the mean values in the majority of the cases but also outperforming all of the individual methods. Nonetheless, the ensembles show no clear pattern with the FM and HB data sets, in some cases performing better than the expected mean of the metrics.

In our opinion, these results give support to our hypothesis that the ensemble could help the MTS clustering practitioner in the selection of the best clustering option using employing the integration of the individual MTS clustering methods. Interestingly, the k-means seems to be more robust and to produce better results than the ensembles with the h-clust. It seems that the k-means repetitions reinforce the relationships among the MTS instances, increasing the metrics consequently.

Considering the results in Table 5, the advantages of including prior knowledge are not clear: for some methods, there are improvements with

Table 4: Results obtained when the **best number of clusters** is found using the rule of the elbow and no prior knowledge.

	AWR (25 labels)			**CR (12 labels)**			**EP (4 labels)**			**FM (2 labels)**			**HB (2 labels)**		
Method	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE
h-A-MIN	28	0.7548	0.9952	10	0.7241	0.9591	5	0.2747	0.8322	3	0.4111	0.5870	4	0.8952	0.1507
h-A-DTW	31	0.8822	0.9978	12	0.9815	0.9680	6	0.3682	0.8250	3	0.4778	0.5189	3	0.8677	0.2290
h-FFT	12	0.8691	0.9361	10	0.7199	0.9362	5	0.3207	0.7396	4	0.2813	0.7150	5	0.8757	0.1836
h-CMD	5	0.4924	0.6583	4	0.8597	0.6661	5	0.6474	0.8710	5	0.2246	0.7735	5	0.2740	0.7408
k-A-kSp$_{10\%}$	23	0.0429	0.9563	7	0.1806	0.8625	4	0.2382	0.7486	3	0.3328	0.6698	3	0.3313	0.6660
k-A-kSp$_{\sqrt{N}\%}$	23	0.0451	0.9572	7	0.1412	0.8562	4	0.2535	0.7469	3	0.3395	0.6752	3	0.3503	0.6488
4 methods mean		0.7496	0.8969		0.8213	0.8824		0.4028	0.8170		0.3487	0.6486		0.7282	0.3260
5 methods mean		0.6087	0.9089		0.6853	0.8771		0.3729	0.8029		0.3469	6099		0.6526	0.3906
	4 methods ensemble														
Method	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE
MIC-kmeans$_{CUT}$	26	0.8793	0.9953	9	0.9815	0.9411	4	0.5471	0.6136	3	0.3319	0.6662	3	0.4056	0.5655
MIC-kmeans$_{0.5}$	12	0.8204	0.9887	10	0.8750	0.9607	6	0.5432	0.8581	3	0.3576	0.6410	5	0.2429	0.7720
MIC-kmeans$_{UNC}$	7	0.9184	0.9824	17	0.7520	0.9981	5	0.6450	0.7596	3	0.3387	0.6602	3	0.4658	0.5141
MIC-hclust$_{CUT}$	16	0.8080	0.9121	13	0.8009	0.9630	4	0.7718	0.3636	4	0.2557	0.7481	5	0.2718	0.7026
MIC-hclust$_{0.5}$	25	0.7898	0.9831	13	0.8981	0.9211	7	0.6719	0.6708	3	0.4042	0.5965	4	0.5680	0.4162
MIC-hclust$_{UNC}$	23	0.7606	0.9979	19	0.6960	0.9910	6	0.6717	0.7254	4	0.2337	0.7624	6	0.1674	0.8254
	5 methods ensemble														
Method	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE
MIC-kmeans$_{CUT}$	12	0.8582	0.9815	10	0.8912	0.9671	6	0.5292	0.9193	3	0.3361	0.6631	5	0.2251	0.7870
MIC-kmeans$_{0.5}$	17	0.7920	0.9966	5	0.9954	0.8116	4	0.6601	0.5779	3	0.3605	0.6371	3	0.4491	0.5384
MIC-kmeans$_{UNC}$	17	0.8355	0.9983	5	1.0000	0.8339	4	0.6799	0.5679	3	0.3487	0.6505	3	0.5290	0.4747
MIC-hclust$_{CUT}$	25	0.7673	0.9905	13	0.6991	0.9530	7	0.6444	0.8380	3	0.3501	0.6487	4	0.4156	0.5495
MIC-hclust$_{0.5}$	12	0.8262	0.8566	6	0.8889	0.6285	5	0.6802	0.5703	5	0.2088	0.7883	5	0.2382	0.7500
MIC-hclust$_{UNC}$	12	0.9289	0.8324	6	0.9697	0.5818	5	0.7557	0.4362	5	0.2304	0.7660	5	0.2874	0.7051

Table 5: Results obtained when the **best number of clusters** is found using the rule of the elbow assisted with prior knowledge. The upper part includes the results with 5% of prior knowledge, while the bottom one includes the results with 10% of prior knowledge.

	AWR (25 labels)			**CR (12 labels)**			**EP (4 labels)**			**FM (2 labels)**			**HB (2 labels)**		
	4 methods ensemble and 5% of prior knowledge														
Method	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE
MIC-kmeans$_{CUT}$	26	0.6691	0.9846	9	0.8611	0.9514	4	0.4546	0.8206	3	0.3534	0.6446	3	0.3737	0.6574
MIC-kmeans$_{0.5}$	12	0.8625	0.9836	10	0.8796	0.9545	6	0.5807	0.9093	3	0.3396	0.6591	5	0.2353	0.7933
MIC-kmeans$_{UNC}$	7	0.9262	0.9806	17	0.7697	0.9977	5	0.6104	0.8107	3	0.3290	0.6691	3	0.4658	0.5141
MIC-hclust$_{CUT}$	16	0.7178	0.9439	13	0.2292	0.9349	4	0.6078	0.8657	4	0.2500	0.7518	5	0.2309	0.7989
MIC-hclust$_{0.5}$	25	0.7673	0.9905	13	0.8889	0.9572	7	0.6444	0.838	3	0.3857	0.6139	4	0.4214	0.5872
MIC-hclust$_{UNC}$	23	0.7606	0.9979	19	0.7224	0.9904	6	0.6717	0.7254	4	0.2337	0.7624	6	0.1674	0.8254
	4 methods ensemble and 10% of prior knowledge														
Method	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE
MIC-kmeans$_{CUT}$	26	0.6335	0.9845	9	0.7083	0.9340	4	0.4568	0.8237	3	0.3722	0.6304	3	0.4555	0.5613
MIC-kmeans$_{0.5}$	12	0.8625	0.9836	10	0.8796	0.9545	6	0.5807	0.9093	3	0.3396	0.6591	5	0.2353	0.7933
MIC-kmeans$_{UNC}$	7	0.9307	0.9752	17	0.6902	0.9994	5	0.6512	0.8338	3	0.3308	0.6676	3	0.4658	0.5141
MIC-hclust$_{CUT}$	16	0.7178	0.9439	13	0.6713	0.9583	4	0.5305	0.7756	4	0.2742	0.7318	5	0.2116	0.7884
MIC-hclust$_{0.5}$	25	0.7673	0.9905	13	0.6991	0.9530	7	0.6444	0.838	3	0.3501	0.6487	4	0.4156	0.5495
MIC-hclust$_{UNC}$	23	0.7606	0.9979	19	0.6960	0.9910	6	0.6604	0.7549	4	0.2337	0.7624	6	0.1674	0.8254
	5 methods ensemble and 5% of prior knowledge														
Method	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE
MIC-kmeans$_{CUT}$	12	0.5258	0.9739	10	0.8102	0.9370	6	0.4289	0.7228	3	0.3719	0.6258	5	0.3729	0.6237
MIC-kmeans$_{0.5}$	17	0.8196	0.9927	5	0.9931	0.8438	4	0.6152	0.7094	3	0.3386	0.6603	3	0.4122	0.5782
MIC-kmeans$_{UNC}$	17	0.8789	0.9980	5	0.9975	0.8396	4	0.6478	0.7399	3	0.3302	0.6692	3	0.4658	0.5141
MIC-hclust$_{CUT}$	25	0.6116	0.9406	13	0.6505	0.9682	7	0.5218	0.8270	3	0.2808	0.7215	4	0.2231	0.7871
MIC-hclust$_{0.5}$	12	0.9033	0.8497	6	0.9514	0.7810	5	0.6981	0.5829	5	0.2214	0.7782	5	0.2773	0.7275
MIC-hclust$_{UNC}$	12	0.8956	0.9455	6	0.9576	0.7676	5	0.6951	0.5745	5	0.1716	0.8290	5	0.2377	0.7516
	5 methods ensemble and 10% of prior knowledge														
Method	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE	K	SEN	SPE
MIC-kmeans$_{CUT}$	12	0.7484	0.9884	10	0.8935	0.9463	6	0.3551	0.7803	3	0.3824	0.6147	5	0.4005	0.5955
MIC-kmeans$_{0.5}$	17	0.8087	0.9951	5	0.9954	0.8135	4	0.6191	0.7140	3	0.3418	0.6572	3	0.5121	0.4703
MIC-kmeans$_{UNC}$	17	0.8262	0.9974	5	0.9974	0.8553	4	0.6985	0.7125	3	0.3316	0.6677	3	0.4658	0.5141
MIC-hclust$_{CUT}$	25	0.6880	0.9470	13	0.7384	0.9660	7	0.4180	0.7916	3	0.2732	0.7321	4	0.2216	0.8181
MIC-hclust$_{0.5}$	12	0.9033	0.8497	6	0.9630	0.7533	5	0.6981	0.5829	5	0.2008	0.8005	5	0.3167	0.6768
MIC-hclust$_{UNC}$	12	0.8956	0.9455	6	0.9576	0.7676	5	0.6951	0.5745	5	0.1716	0.8290	5	0.2377	0.7516

Table 6: Results obtained when the **number of clusters (K) is set to the number of classes** in the data set. There is no prior knowledge.

	AWR (25 labels)		**Cr (12 labels)**		**EP (4 labels)**		**FM (2 labels)**		**HB (2 labels)**	
	SEN	SPE	SEN	SPE	SEN	SPE	SEN	SPE	SEN	SPE
h-A-MIN	0.8319	0.9902	0.7322	0.9593	0.3143	0.7396	0.6340	0.3649	0.9474	0.0778
h-A-DTW	0.9113	0.9928	0.9352	0.9832	0.3682	0.7936	0.5057	0.4912	0.8687	0.2264
h-FFT	0.7731	0.9882	0.6782	0.9534	0.3281	0.7164	0.8168	0.182	0.9882	0.0068
h-CMD	0.1297	0.9537	0.5817	0.8413	0.6742	0.8191	0.5208	0.4771	0.5183	0.4956
k-A-kSp$_{10\%}$	0.0429	0.9594	0.088	0.9175	0.2478	0.7528	0.4992	0.4991	0.4961	0.4998
k-A-Sp$_{\sqrt{N}\%}$	0.0429	0.9599	0.0949	0.9149	0.2731	0.7514	0.4969	0.4999	0.5066	0.5014
	4 methods ensemble									
	SEN	SPE	SEN	SPE	SEN	SPE	SEN	SPE	SEN	SPE
MIC-kmeans$_{CUT}$	0.8924	0.9953	0.9259	0.9837	0.6806	0.4732	0.4971	0.5001	0.6429	0.3009
MIC-kmeans$_{0.5}$	0.7286	0.9982	0.8220	0.9704	0.6920	0.5992	0.5018	0.4948	0.8032	0.1764
MIC-kmeans$_{UNC}$	0.7600	0.9992	0.8926	0.9844	0.7084	0.6281	0.5001	0.4968	0.7762	0.1947
MIC-hclust$_{CUT}$	0.7978	0.9828	0.8102	0.9544	0.7718	0.3636	0.5474	0.4501	0.5134	0.5087
MIC-hclust$_{0.5}$	0.7898	0.9831	0.9190	0.9076	0.7845	0.3346	0.5768	0.4217	0.9175	0.0825
MIC-hclust$_{UNC}$	0.7361	0.9987	0.8943	0.9627	0.7641	0.4357	0.5566	0.4418	0.8803	0.1206
	5 methods ensemble									
	SEN	SPE	SEN	SPE	SEN	SPE	SEN	SPE	SEN	SPE
MIC-kmeans$_{CUT}$	0.6615	0.9826	0.6676	0.9574	0.4153	0.7792	0.497	0.5001	0.5085	0.4994
MIC-kmeans$_{0.5}$	0.7324	0.9983	0.8356	0.9699	0.6693	0.6005	0.5014	0.4954	0.8051	0.1740
MIC-kmeans$_{UNC}$	0.7994	0.9995	0.9219	0.9799	0.739	0.6247	0.4999	0.4970	0.7752	0.1954
MIC-hclust$_{CUT}$	0.5836	0.9793	0.6252	0.9345	0.4296	0.7880	0.4972	0.5003	0.5114	0.5156
MIC-hclust$_{0.5}$	0.6790	0.8983	0.6919	0.8908	0.6785	0.8335	0.6577	0.8422	0.6918	0.8083
MIC-hclust$_{UNC}$	0.6736	0.8999	0.6894	0.8963	0.6764	0.8436	0.6557	0.8442	0.6880	0.8121

the 5% of prior knowledge, but without a clear pattern. In other words, the benefits depend on the data set. This finding suggests that the effort in labelling and introducing the prior knowledge would barely pay off. Perhaps different prior knowledge representations, such those proposed in [26, 27], could lead to better resutls.

Comparing the performances with the number of clusters set to the number of labels showed surprising results, as can be seen in Table 6. As expected, the individual MTS instance clustering methods performed worse or equal than with the rule of the elbow. This was expected as well for the k-means ensembles but results show a different story.

Table 6 shows the results when the number of clusters is set to the number labels in the data set. As expected, the individual MTS instance clustering methods performed at most as good as with the rule of the elbow, being worse in the majority of the cases. Interestingly, the performance of the ensembles varied according to the data set: in some cases, better SEN values were obtained although at the cost of a worse SPE. This result suggest that perhaps using these metrics (or its combination with the geometric mean) in the rule of the elbow would lead to better performance groupings.

Besides, there are some concerns with the use of these ensembles. On the one hand, the use of several distance functions increases the time consumption; methods without MTS representation would possibly increase the cost of the experimentation. On the other hand, the selection of a time series representation should be carefully done to avoid problems as those commented for the k-A-Sp. Moreover, the ensembles proposed in this study show a robust performance even when several methods perform poorly on the data.

Finally, the number of labels in the MTS data set seems to have a repercussion in the ensembles as the performance is better for those with a higher number of classes. However, this remark requires further experimentation to verify it.

Some other issues that need further study. For instance, the credibility of each method can vary according to certain external metrics (the coincidence of MTS events, temporal sequences of relevant changes, error in predicting MTS instances when modelling with others [26], etc), which might increase the SEN and SPE of the ensembles. Furthermore, these external metrics can also be used as distance metrics as suggested in [27], although this approach needs some simplifications to reduce the computational costs. Similarly, studying more combinations of MTS representation and distance functions and, more importantly, finding digital markers of the plausibility of each one and each combination will lead to automatically set the ensemble according to the MTS problem.

6. Conclusion and Future Work

This study addresses the ensemble of MTS clustering methods to assist the practitioner in the selection of the best clustering options. Besides, a representation of prior knowledge is presented and evaluated with the ensemble. The ensemble includes several different MTS representation methods with an associated distance function; each of these pairs produces a similarity matrix that is aggregated. Finally, a clustering algorithm generates the final grouping.

The obtained results show the capacity of the ensemble to retain the performance of the better methods in the majority of the cases, even when half of the methods were remarkably worse than the others. Moreover, the proposed prior knowledge representation did not enhance the metrics of the ensemble, suggesting this type of knowledge is not suitable for the clustering methods. However, the number of clusters is a parameter that helps the ensemble; this prior knowledge requires further study.

Besides, the use of external metrics may help in balancing the methods to merge. Furthermore, some of these external metrics can also be used as distance measurements. An in-depth study is needed in the evaluation of the different MTS representation and the distance function to determine digital markers of the most interesting candidates according to the problem. All of these aspects represent future research.

Acknowledgements

We want to thank Dr. John Paparrizos with the University of Chicago for sharing his code with us.

This research has been funded by the Spanish Ministry of Science and Innovation, under project MINECO-TIN2017-84804-R, and by the Grant FC-GRUPIN-IDI/2018/000226 project from the Asturias Regional Government. Also, the research has been funded by project CCTT2/18/BU/0002 from the Instituto para la Competitividad Empresarial de Castilla y León.

References

[1] T. Warren Liao, Clustering of time series data—a survey, Pattern Recognition 38 (2005) 1857–1874. URL: https://www.sciencedirect.com/science/article/pii/S0031320305001305. doi:10.1016/J.PATCOG.2005.01.025.

[2] N.-A. Le-Khac, C. Fan, T. Kechadi, Clustering Approaches for Financial Data Analysis: a survey, in: 8th International conference on Data Mining (DMIN 2012), 2012, pp. 1–7.

[3] S. Aghabozorgi, A. Seyed Shirkhorshidi, T. Ying Wah, Time-series clustering - A decade review, Information Systems 53 (2015) 16–38. URL: https://www.sciencedirect.com/science/article/pii/S0306437915000733. doi:10.1016/j.is.2015.04.007.

[4] T. Van Craenendonck, W. Meert, S. Dumancic, H. Blockeel, COBRASTS: A New Approach to Semi-supervised Clustering of Time Series, in: L. Soldatova, J. Vanschoren, G. Papadopoulos, M. Ceci (Eds.), Discovery Science, Springer International Publishing, Cham, 2018, pp. 179–193.

[5] A. Bagnall, M. Flynn, J. Large, J. Lines, M. Middlehurst, A tale of two toolkits, report the third: on the usage and performance of HIVE-COTE v1.0, arXiv e-prints (2020). URL: http://arxiv.org/abs/2004.06069. arXiv:2004.06069.

[6] J. Paparrizos, L. Gravano, Fast and Accurate Time-Series Clustering, ACM Trans. Database Syst. 42 (2017) 8:1—-8:49. URL: http://doi.acm.org/10.1145/3044711. doi:10.1145/3044711.

[7] Y. Lee, J. Na, W. B. Lee, Robust design of ambient-air vaporizer based on time-series clustering, Computers & Chemical Engineering 118 (2018) 236–247. URL: https://linkinghub.elsevier.com/retrieve/pii/S0098135418308822. doi:10.1016/j.compchemeng.2018.08.026.

[8] J. Zakaria, A. Mueen, E. Keogh, Clustering Time Series Using Unsupervised-Shapelets, in: 2012 IEEE 12th International Conference on Data Mining, 2012, pp. 785–794. doi:10.1109/ICDM.2012.26.

[9] G. Morgachev, A. Goncharov, V. Strijov, Distance Function Selection for Multivariate Time-Series, in: Proceedings - 2019 International Conference on Artificial Intelligence: Applications and Innovations, IC-AIAI 2019, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 66–70. doi:10.1109/IC-AIAI48757.2019.00021.

[10] R. Bellman, R. Kabala, On adaptive control processes, IRE Transactions on Automatic Control 4 (1959) 1–9.

[11] B. Kenji Iwana, V. Frinken, S. Uchida, DTW-NN: A novel neural network for time series recognition using dynamic alignment between inputs and weights, Knowledge-Based Systems 188 (2020) 104971. URL: https://doi.org/10.1016/j.knosys.2019.104971. doi:10.1016/j.knosys.

[12] T. Han, Q. Peng, Z. Zhu, Y. Shen, H. Huang, N. N. Abid, A pattern representation of stock time series based on DTW, Physica A: Statistical Mechanics and its Applications (2020). doi:10.1016/j.physa.2020.124161.

[13] G. Jiang, W. Wang, W. Zhang, A novel distance measure for time series: Maximum shifting correlation distance, Pattern Recognition Letters 117 (2019) 58–65. doi:10.1016/j.patrec.2018.11.013.

[14] G. E. A. P. A. Batista, E. J. Keogh, O. M. Tataw, V. M. A. de Souza, CID: an efficient complexity-invariant distance for time series, Data Mining and Knowledge Discovery 28 (2014) 634–669.

[15] J. A. Vilar, B. Lafuente-Rego, P. D'Urso, Quantile autocovariances: A powerful tool for hard and soft partitional clustering of time series, Fuzzy Sets and Systems 340 (2018) 38–72. doi:10.1016/j.fss.2017.03.006.

[16] F. Madrid, S. Singh, Q. Chesnais, K. Mauck, E. Keogh, Matrix profile XVI: Efficient and effective labeling of massive time series archives, in: Proceedings - 2019 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2019, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 463–472. doi:10.1109/DSAA.2019.00061.

[17] P. Steinmann, Behavior-Based Scenario Discovery, Ph.D. thesis, Delft University of Technology, 2018. URL: http://resolver.tudelft.nl/uuid:cb4fee52-e23f-41d3-9c63-7ca7ec948304.

[18] P. Montero, J. A. Vilar, TSclust: An R package for time series clustering, Journal of Statistical Software 62 (2014) 1–43. doi:10.18637/jss.v062.i01.

[19] G. Liu, L. Zhu, X. Wu, J. Wang, Time series clustering and physical implication for photovoltaic array systems with unknown working conditions, Solar Energy 180 (2019) 401–411. doi:10.1016/j.solener.2019.01.041.

[20] C. Hora, H. Budman, A hybrid clustering approach for multivariate time series – A case study applied to failure analysis in a gas turbine, ISA Transactions 71 (2017) 513–529. URL: http://dx.doi.org/10.1016/j.isatra.2017.09.004. doi:10.1016/j.isatra.2017.09.004.

[21] H. Li, Multivariate time series clustering based on common principal component analysis, Neurocomputing 349 (2019) 239–247. doi:10.1016/j.neucom.2019.03.060.

[22] V. S. Siyou Fotso, E. Mephu Nguifo, P. Vaslin, Frobenius correlation based u-shapelets discovery for time series clustering, Pattern Recognition 103 (2020). doi:10.1016/j.patcog.2020.107301.

[23] P. Steinmann, W. L. Auping, J. H. Kwakkel, Behavior-based scenario discovery using time series clustering, Technological Forecasting and Social Change 156 (2020). doi:10.1016/j.techfore.2020.120052.

[24] L. Duan, F. Yu, W. Pedrycz, X. Wang, X. Yang, Time-series clustering based on linear fuzzy information granules, Applied Soft Computing Journal 73 (2018) 1053–1067. URL: https://doi.org/10.1016/j.asoc.2018.09.032. doi:10.1016/j.asoc.2018.09.032.

[25] H. Li, M. Wei, Fuzzy clustering based on feature weights for multivariate time series, Knowledge-Based Systems (2020). doi:10.1016/j.knosys.2020.105907.

[26] I. Vázquez, J. R. Villar, J. Sedano, S. Simic, E. A. de la Cal, A proof of concept in multivariate time series clustering using recurrent neural networks and sp-lines, in: Proceedings of the 14th International Conference on Hybrid Artificial Intelligent Systems, HAIS 2019, 2019, pp. 346–357. URL: https://doi.org/10.1007/978-3-030-29859-3_30. doi:10.1007/978-3-030-29859-3_30.

[27] I. Vázquez, J. R. Villar, J. Sedano, S. Simic, A preliminary study on multivariate time series clustering, in: Proceedings of the 14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019), 2019, pp. 473–480. URL: https://doi.org/10.1007/978-3-030-20055-8_45. doi:10.1007/978-3-030-20055-8_45.

[28] K. Ø. Mikalsen, F. M. Bianchi, C. Soguero-Ruiz, R. Jenssen, Time series cluster kernel for learning similarities between multivariate time series with missing data, Pattern Recognition 76 (2018) 569–581. URL: https://linkinghub.elsevier.com/retrieve/pii/S0031320317304843. doi:10.1016/j.patcog.2017.11.030.

[29] I. Deeva, N. O. Nikitin, A. V. Kaluyzhnaya, Pattern Recognition in Non-Stationary Environmental Time Series Using Sparse Regression, in: Procedia Computer Science, volume 156, Elsevier B.V., 2019, pp. 357–366. doi:10.1016/j.procs.2019.08.212.

[30] J. Zhou, S. F. Zhu, X. Huang, Y. Zhang, Enhancing Time Series Clustering by Incorporating Multiple Distance Measures with Semi-Supervised Learning, Journal of Computer Science and Technology 30 (2015) 859–873. doi:10.1007/s11390-015-1565-7.

[31] E. Keogh, S. Lonardi, C. A. Ratanamahatana, L. Wei, S. H. Lee, J. Handley, Compression-based data mining of sequential data, Data Mining and Knowledge Discovery 14 (2007) 99–129. doi:10.1007/s10618-006-0049-3.

[32] J. S. Anstey, D. K. Peters, C. Dawson, An Improved Feature Extraction Technique for High Volume Time Series Data, in: Proceedings of the Fourth Conference on IASTED International Conference: Signal Processing, Pattern Recognition, and Applications, SPPR'07, ACTA Press, Anaheim, CA, USA, 2007, pp. 74–81. URL: http://dl.acm.org/citation.cfm?id=1331978.1331991.

[33] E. Keogh, S. Chu, D. Hart, M. Pazzani, An online algorithm for segmenting time series, in: Proceedings 2001 IEEE International Conference on Data Mining, 2001, pp. 289–296. doi:10.1109/ICDM.2001.989531.

[34] K. Chakrabarti, E. Keogh, S. Mehrotra, M. Pazzani, Locally adaptive dimensionality reduction for indexing large time series databases, ACM Transactions on Database Systems (TODS) 27 (2002) 188–228.

[35] K. Chan, A. W. Fu, Efficient time series matching by wavelets, in: Proceedings of the 15th IEEE Int'l Conference on Data Engineering, 1999, pp. 126–133.

[36] R. C. Singleton, An Algorithm for Computing the Mixed Radix Fast Fourier Transform, IEEE Transaction on Audio and Electroacoustics 17 (1969) 93–103.

[37] IBadri Narayan Mohapatra, Rashmita Kumari Mohapatra, FFT and Sparse FFT techniques and applications, in: 2017 Fourteenth International Conference on Wireless and Optical Communications Networks, IEEE, Mumbai, 2017, pp. 1–5. doi:10.1109/WOCN.2017.8065859.

[38] M. Vlachos, J. Lin, E. Keogh, D. Gunopulos, A wavelet-based anytime algorithm for k-means clustering of time series, Proceedings of the Workshop on Clustering High Dimensional Data and Its Applications (2003) 23–30.

[39] J. Shi, J. Malik, Normalized cuts and Image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 888–905.

[40] A. Bagnall, J. Lines, A. Bostrom, J. Large, E. Keogh, The Great Time Series Classification Bake Off: a Review and Experimental Evaluation of Recent Algorithmic Advances, Data Mining and Knowledge Discovery 31 (2017) 606–660.

[41] J. Wang, A. Balasubramanian, L. M. de la Vega, J. R. Green, A. Samal, B. Prabhakaran, Word recognition from continuous articulatory movement time-series data using symbolic representations, in: ACL/ISCA interspeech workshop on speech and language processing for assistive technologies, 2013, pp. 119–127.

[42] M. Shokoohi-Yekta, B. HuHongxia, J. Wang, E. Keogh, Generalizing DTW to the multi-dimensional case requires an adaptive approach, Data Mining and Knowledge Discovery 31 (2017) 1–31.

[43] M. Ko, G. West, S. Venkatesh, M. Kumar, Online context recognition in multisensor systems using dynamic time warping, in: Proceedings of the IEEE international conference on intelligent sensors, sensor networks and information processing (ISSNIP), 2005, pp. 283–288.

[44] J. R. Villar, P. Vergara, M. Menéndez, E. de la Cal, V. M. González, J. Sedano, Generalized Models for the Classification of Abnormal Movements in Daily Life and its Applicability to Epilepsy Convulsion Recognition, International Journal of Neural Systems 26 (2016) 1650037. doi:10.1142/s0129065716500374.

[45] B. Blankertz, G. Curio, K.-R. Muller, No Title, Advances in Neural Information Processing Systems 14 (NIPS 2001) (2011).

[46] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, H. E. Stanley, hysioBank, PhysioToolkit, and PhysioNet Components of a New Research Resource for Complex Physiologic Signals, Circulation 101 (2000). doi:https://doi.org/10.1161/01.CIR.101.23.e215dy.

[47] C. Liu, D. Springer, Q. Li, B. Moody, R. A. Juan, F. J. Chorro, F. Castells, J. M. Roig, I. Silva, A. E. W. Johnson, Z. Syed, S. E. Schmidt, C. D. Papadaniil, L. Hadjileontiadis, H. Naseri, A. Moukadem, A. Dieterlen, C. Brandt, H. Tang1, M. Samieinasab, M. R. Samieinasab, R. SameniRoger, G. Mark, G. D. Clifford, An open access database for the evaluation of heart sound algorithms, Physiological Measurement 37 (2016).