A proof of concept in Multivariate Time Series clustering using Recurrent Neural Networks and SP-Lines

Iago Vázquez[1], José R. Villar[2], Javier Sedano[1], Svetlana Simić[3], and Enrique de la Cal[2]

[1] Instituto Tecnológico de Castilla y León, Pol. Ind. Villalonquejar, 09001, Burgos, Spain
{iago.vazquez,javier.sedano}@itcl.es
[2] University of Oviedo, Computer Science Department, EIMEM, Oviedo, Spain
{villarjose,delacal}@uniovi.es
[3] University of Novi Sad, Department of Neurology
Clinical Centre of Vojvodina Novi Sad, Republic of Serbia
svetlana.simic@mf.uns.ac.rs

Abstract. Big Data and the IoT explosion has made clustering multivariate Time Series (TS) one of the most effervescent research fields. From Bio-informatics to Business and Management, multivariate TS are becoming more and more interesting as they allow to match events the co-occur in time but that is hardly noticeable. This study represents a step forward in our research. We firstly made use of Recurrent Neural Networks and transfer learning to analyze each example, measuring similarities between variables. All the results are finally aggregated to create an adjacency matrix that allows extracting the groups. In this second approach, splines are introduced to smooth the TS before modeling; also, this step avoid to learn from data with high variation or with noise. In the experiments, the two solutions are compared suing the same proof-of-concept experimentation.

1 INTRODUCTION

Time Series (TS) clustering is one of the most effervescent research fields due to the Big Data and the IoT explosion. Until recently, the problem was focused on univariate TS clustering. For instance, [10] proposed use dynamic time warping and k-means to cluster the performance of a photovoltaic power plant, so to predict the meteorological conditions. Similarly, k-means was used to cluster TS and then predict the weather conditions [8]. Interested readers would read the review in [1] for a good review on this topic.

However, TS clustering has been moving from univariate to multivariate TS problems. In these problems, a TS includes more than one variable; i.e., the pollution measurements in a medium or big city includes several physical and chemical variables registered in several stations placed all around of a city. Clustering multivariate TS has been found interesting in order to perform complex event detection or to classify the current scenario. For instance, [4] proposed a Partitioning around Meroids and Fuzzy C-Meroids clustering for the problem of detecting high-value pollution records or alarms in the city of Rome.

The similarity among the variables within the TS is one of the most studied topics. PCA similarity factor was combined with the average based Euclidean distance together with a fuzzy clustering scheme in [6]. Discords have been used in multivariate TS to identify anomalies and introduce more efficient search processes [7]. Hash functions have also been used to index and to measure the similarities in multivariate TS searches [16].

Interestingly, models have been also used in measuring the similarity between multivariate TS, i.e., Gaussian Mixture Models [11]. A different approach is based on extracting features and then using these features to group the multivariate TS [5]. Feature extraction together with Self-Organized Maps [14], Hidden Markov Models [9] or Fuzzy Linear [3] are techniques that have been also proposed in solving multivariate TS. Still, this problem cannot be considered solved and a recent study found out that the combination of feature extraction and a classification stage performs better than the current approaches [2].

In this study, a similar idea of that proposed in [11] is revisited for multivariate TS. Recurrent Neural Networks (RNN) are learned to predict a variable from an example and then used to measure the similarity between the different variables. Afterward, the adjacency matrix is found for each example, then aggregated for all the examples and finally binarized to generate the final adjacency matrix. The groups are proposed based on the variables that mutually dependent. To reduce the complexity of the solution transfer learning is proposed

The organization of this manuscript is as follows. The next section describes both the previous research and the proposal for this study. Section 3 details the dataset and some method's parameters, while Section 4 includes the figures and the discussion on the results. The study ends with the conclusions.

2 Clustering multivariate TS

Basically, we use SP-Lines before modelling each of the features in an instance of a multivariate TS. Afterwards, we use the same procedure proposed in [15]. This sections explains the whole process by, first, introducing the previous study in SubSect. 2.1 and then describing how the SP-Lines are used in SubSect. 2.2.

Let's define multivariate TS dataset as the dataset containing examples, each example is a multivariate TS. A multivariate TS is an arrangement of several TS, each one belonging to a different variable. We assume all the examples having the same variables and, without loss of generality, the same sampling frequency and the same number of samples. Therefore, a multivariate TS example is a matrix of m rows of n variables, where each column represents a univariate TS. However, each example has its own number of samples.

2.1 RNN applied in multivariate TS clustering

A two stages solution was proposed in [15]: i) the first stage is devoted to find the similarities between variables in a single example, that is, in a single multivariate TS, and ii) the second stage aims to aggregate the results among the examples and extract the relationships. To find the similarities between features a RNN predicts the test subsequence of a variable, the prediction error over the remaining variables is a measurement of similarity among variables. For this preliminary study, the aggregation of the results was performed with simple thresholding followed by a graph representation. To make the process feasible, we propose to use transfer learning [12].

Finding similarities between variables from an example The procedure is depicted in Algorithm 1. Let's TS^i be the current example, $TS^i = \{X_1^i, \ldots, X_n^i\}$ $\forall i : 1, \ldots, N$, where N is the number of examples, n is the number of variables. Moreover, each variable X_j^i can be written as $X_j^i = (x_{j1}^i, \ldots, x_{jm_i}^i)$ $\forall j : 1, \ldots, n$, with m_i being the number of samples of the TS for each variable in the example i.

Algorithm 1 Computing similarities between features in an example

procedure IN-EXAMPLE-SIMILARITY(TS^i, LoRNN) ▷ LoRNN list of pre-learnt RNNs, if available
    $sim \leftarrow$ zeroes matrix of size $n \times n$
    for each variable j in TS^i **do**
        $X_j^i \leftarrow$ normalize(X_j^i)
        $RNN_j^i \leftarrow$ **Train-RNN**(X_j^i, LoRNN[j])
        $LoRNN[j] \leftarrow RNN_j^i$
        $e_j^i \leftarrow$ RMSE(RNN_j^i, test(X_j^i))
        for each variable k in TS^i, $k \neq j$ **do**
            $X_k^i \leftarrow$ normalize(X_k^i)
            $e_{jk}^i \leftarrow$ RMSE(RNN_j^i, test(X_k^i))
            $sim[j,k] \leftarrow abs(\frac{e_{jk}^i - e_j^i}{e_j^i})$
        end for
    end for
    return sim
end procedure

procedure TRAIN-RNN(X_j^i, RNN) ▷ RNN is a RNN, if available
    if is.NULL(RNN) **then**
        $RNN \leftarrow$ *full train* RNN for the train part of X_j^i
    else
        $RNN \leftarrow$ *tune* RNN for the train part of X_j^i
    end if
    return RNN
end procedure

Let's also assume that a given percentage (%TRN) of the samples of a TS is kept for training and the remaining for testing. In other words, for any variable X_j^i in example i, $(x_{j1}^i, \ldots, x_{j(\%TRN \times m_i)}^i)$ are kept for training and $(x_{j(\%TRN \times m_i)}^i, x_{jm_i}^i)$ are kept for testing.

It is possible to learn an RNN using the training part of X_j^i to predict its behavior in the testing part, let's call this RNN_j^i. Let us suppose we obtain a good model, and that the aggregation of the prediction error along the test subset for variable X_j^i is e_j^i. This prediction error can be any well-known measurement, as the Root Mean Square Error (RMSE) or similar.

The RNN_j^i is applied to predict each of the remaining variables X_k^i with $k : 1 \cdots n$ and $k \neq j$. The error obtained with RNN_j^i when predicting the test subsequence of the variable X_k^i is denoted as e_{jk}^i. This error is scaled wrt the e_j^i in order to obtain a similarity value: $E_{jk}^i = |(e_{jk}^i - e_j^i)/e_j^i|$. Values close to 0.0 means the TS can be successfully predicted by RNN_j^i. This prediction is also repeated for $-X_k^i$, that is, the normalized test sequence X_k^i is swapped wrt the time axis to consider the case the two TS X_j^i and X_k^i have a negative correlation. Consequently, the minimum of both errors is kept.

Therefore, the similarity between variable j and the remaining variables in the example i is obtained as the vector $sim_j^i = (E_{j1}^i, \cdots, E_{jn}^i)$, with $E_{jj}^i = 0$. Finally, repeating this procedure for each of the variables in the example i, a distance $n \times n$ matrix is obtained, which represents the outcome of this stage.

RNN and Transfer learning As seen in Algorithm 1 and in the previous subsection, an RNN is trained using the train part of variable X_j^i from example TS^i. As for our previous research [15]; this study makes use of the *rnn* R-package [13]. For each training process a simple grid of 12 different learning rate values (from 1/12 to 1.0), 1 to 12 as the number of epochs and 1 to 12 hidden neurons.

However, training a complete RNN from scratch for each variable and for each example makes this approach unfeasible for even small multivariate TS datasets. A simplification is clearly needed.

To do so, a simple transfer learning scheme [12] is used. For the first time, the Train-RNN is call, a NULL value is given as current RNN; thus, full learning of the RNN is performed. However, when it is not NULL, then it is the RNN_j^i trained in the first iteration of the process for variable j and example $i == 1$. We reuse this RNN model, fitting it to the current X_j^i. This adaptation is just a simple weight tuning during a reduced number of iterations (20 in this study).

Computing the similarities within a multivariate TS dataset Once the similarity matrix between the variables from an example is obtained, computing the similarity between the variables for the whole multivariate TS dataset is a matter of choosing the method. In the preliminary research, each matrix was converted to an adjacency matrix and then to aggregate the adjacency matrices.

The similarity matrix for example TS^i is converted into the SIM adjacency matrix SIM_i as follows. For each pair of variables j and k from the example

i, if similarity between j and k is smaller or equal than $th1$ ($sim_i[j,k] \leq th1$), then variable j can predict variable k (denoted as $k \lesssim j$). Thus, $SIM_i[j,k] = 1$; otherwise, $SIM_i[j,k] = 0$.

The adjacency matrices are then aggregated as just the sum of all of them. Therefore, the final aggregated adjacency matrix is $SIM_{ag} = \sum_{\forall i} SIM_i$, each cell contains an integer from 0 to N. Finally, the outcome adjacency matrix SIM_{final} is obtained by thresholding SIM_{ag} such that whenever $SIM_{ag}[j,k] \geq th2$ then $SIM_{final} = 1$, otherwise $SIM_{final} = 0$.

This binarization produces an adjacency matrix that can be represented in a graph. This visualization can help in deciding what to do with those variables that have not been grouped yet.

2.2 Using SP-lines in multivariate TS clustering

The main difference in this new study is that the variables in each instance are modeled using SP-lines. The idea was developed after the fluctuation of the signal and also because of the effects in a focused problem related with photovoltaic solar power plants. In this problem, the power of a photovoltaic panel has a strong dependence on the irradiance; the clouds that might partially cover the panel clearly decrease the generated power. But as long as the clouds can move rather fast, the consequence is that the generated power behaves with many peaks and valleys and with more fluctuation than with steady weather conditions. Thus, using SP-lines part of this extra fluctuation is absorbed but the peaks and valleys are kept.

The main part of the previous approach is used, but small variations are introduced to include the SP-lines. Previous to the RNN modeling, each TS of an instance is processed as follows:

- Filter outlier values and smooth the TS with sliding window of 5 time samples.
- Normalize the TS with the mean and standard deviation of the TS.
- If needed, segment the TS according to the behaviour of the signal. The main point is to do not substitute stationary segments with a SP-line.
- For each non-stationary segment, determine the SP-line that best fit the TS with 100 degrees of freedom.
- Join all the segments (stationary and converted to SP-lines) in a single TS; this TS is used as the variable's TS within the instance.

An example of a part of a TS and the calculated SP-line is shown in Fig. 1. The instances with the new computed TS for each variable are then used as the instance to evaluate the similarity among the variables.

3 Experiment and methods

To evaluate this preliminary study a real-world multivariate TS dataset has been used; it is the same data set used in [15], so comparisons can be presented. This

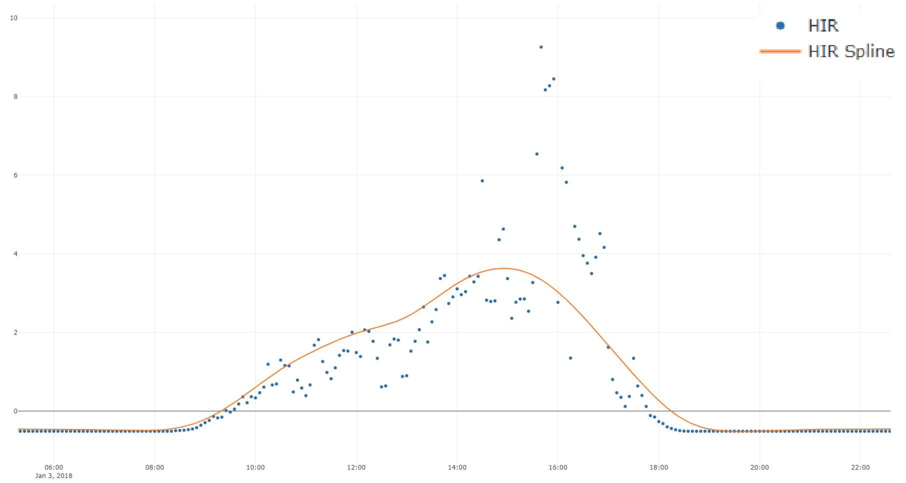

Fig. 1. Example to show the behaviour of the SP-Lines (continous lines) compared to the original TS (dotted lines). The original TS have two stationary segments at the beginning and the end of the TS. The final TS would contain the stationary segments and the inner part of the SP-line curve.

dataset includes up to $n = 11$ variables in each TS example. These multivariate TS have been extracted from a photovoltaic solar power plant, including the following variables:

- Indoor and outdoor temperatures in the weather station (TIN, TOUT)
- Horizontal and Vertical Irradiance reference measurement (HIR and VIR)
- The voltage at the weather station's battery (BV)
- The temperature of 4 photovoltaic panels linked to an inverter (T1 to T4)
- An In-panel Horizontal and Vertical Irradiance measurement (PHI, VHI)

This a toy problem used as a proof of concept. For this problem, the relationships among variables are known: the temperatures (T_x, $x \in \{1, 2, 3, 4\}$) are interrelated and so does PVI and VIR. Besides, PVI influences the behaviour of HRI and PHI (these two are interrelated). TIN and TOUT are also mutually dependent, while BV is totally independent of the others.

Each example includes data from the evolution of the magnitude of these variables for a period of four days. This period has been chosen to include long enough TS, however, larger periods could have been chosen as well. Although data are available for more than three months, in this preliminary study only the $N = 5$ examples of these data are considered. Clearly, this is a proof of concept study. A more in-depth research is needed to extract more solid conclusions. However, using this toy problem allows us to evaluate how the method behaves, its weak points and the enhancements needed to be valid for general problems.

The values of the thresholds have been set before any further analysis to $th1 = 0.07$ and $th2 = 3$ (equivalent to require that the 60% of the examples must include that relationship in order to accept a dependence).

4 Results and discusion

The results are shown from Table 1 to Table 5 and in Figures 2 and 3:

- Table 1 includes the RMSE error measurement obtained when training the RNN_j^i in time series prediction mode for both cases (with and without the SP-lines). These figures are included for information only, so the reader has an idea of how the errors evolved.
- Table 2 depicts the values of E_{jk}^i obtained for the first example in the dataset for clustering with RNN and without the SP-lines. Each line stands to using the model for the corresponding variable, including the error obtained when modelling the other variables in the instance. It is worth noticing how the error raises for those variables for which there are no relationship at all (i.e., VIR and T1), while is really small for related variables (i.e., VIR and HIR).
- Table 3 shows the figures when using the SP-lines. Similar comments to those observed for the previous table can be mentioned for this case.
- Table 4 and Tables 5 show the adjacency matrix obtained after the aggregation of the different examples and pruning with $th2 = 3$, without and with the SP-lines, respectively. The two methods produce the same adjacency matrix.
- Fig. 2 and Fig. 3 show the graph obtained from that adjacency matrix for each case; the two graphs are the same. More importantly, the graphs show the existent relationships among the features.

Aside the type of toy problem that we are using in this proof-of-concept and comparison, with a very short multivariate TS without much complexity, the behaviour of both the approach and the results seem valid. In both cases the found relationships are the expected (with variations from one method to the other). Perhaps the most suited grouping is the original one (using only the RNN with the multivariate TS raw data), but the used threshold was higher than that of the clustering with the sp-lines pre-processing.

Nevertheless, it is clear that the proposal still needs plenty of amendments as well as the structure of the method. However, the obtained results seem to be promising. Items such as different type of TS prediction techniques that might be applied provided transfer learning can be deployed, automatic setting the thresholds to adapt to the problem faced, or the definition of similarity measurements that might be more promising than the scaled RMSE are included among others in the next research to be performed. Furthermore, we do believe that this method could be directly applied in medicine and biology, especially in problems where the experts need support in the analysis of big volumes of multivariate TS.

RNN clustering			
Variable	Train Error	Variable	Train Error
VIR	0.1194	PHI	0.1471
HIR	0.1481	PVI	0.1346
T1	0.0449	TIN	0.04212
T2	0.0536	TOUT	0.0505
T3	0.0461	BV	0.1669
T4	0.0461		
SP-lines + RNN clustering			
Variable	Train Error	Variable	Train Error
VIR	2.5075	PHI	1.3457
HIR	1.3251	PVI	2.5258
T1	0.3993	TIN	0.1931
T2	0.4336	TOUT	0.2268
T3	0.4353	BV	0.1669
T4	0.4265		

Table 1. The RMSE error measurement for each of the fully trained RNN. Upper part: the previous research in [15]. Lower part: results using the SP-lines.

	T1	T2	T3	T4	TIN	BV	TOUT	VIR	VHI	HIR	PHI
T1	0	0.037	0.044	0.056	0.038	$> 10^{14}$	0.084	$> 10^{14}$	$> 10^{14}$	$> 10^{13}$	$> 10^{14}$
T2	0.038	0	0.005	0.006	0.199	$> 10^{14}$	0.065	$> 10^{14}$	$> 10^{14}$	$> 10^{14}$	$> 10^{14}$
T3	0.046	0.007	0	0.009	0.122	$> 10^{14}$	0.003	$> 10^{14}$	$> 10^{14}$	$> 10^{14}$	$> 10^{13}$
T4	0.058	0.019	0.012	0	0.149	$> 10^{14}$	0.023	$> 10^{14}$	$> 10^{14}$	$> 10^{14}$	$> 10^{14}$
TIN	0.142	0.184	0.191	0.196	0	$> 10^{14}$	0.151	$> 10^{14}$	$> 10^{14}$	$> 10^{14}$	$> 10^{14}$
BV	0.224	0.194	0.195	0.210	0.374	0	0.284	0.154	0.176	0.183	0.151
TOUT	0.036	0.074	0.080	0.078	0.129	$> 10^{14}$	0	$> 10^{14}$	$> 10^{14}$	$> 10^{14}$	$> 10^{14}$
VIR	0.503	0.489	0.487	0.486	0.498	0.343	0.438	0	0.008	0.011	0.006
VHI	0.207	0.163	0.172	0.197	0.075	0.648	0.233	0.065	0	0.007	0.070
HIR	0.189	0.146	0.155	0.180	0.112	0.685	0.271	0.075	0.007	0	0.080
PHI	0.231	0.188	0.196	0.219	0.005	0.601	0.148	0.003	0.059	0.065	0

Table 2. RNN clustering. The similarity matrix obtained with the first example from the multivariate TS dataset.

	T1	T2	T3	T4	TIN	BV	TOUT	VIR	VHI	HIR	PHI
T1	0	0.071	0.073	0.048	0.521	0.08	0.432	5.281	5.327	2.321	2.373
T2	0.097	0	0.013	0.036	0.562	0.159	0.48	4.784	4.826	2.058	2.106
T3	0.099	0.018	0	0.039	0.561	0.156	0.479	4.763	4.805	2.047	2.094
T4	0.081	0.002	0.004	0	0.553	0.139	0.47	4.881	4.924	2.11	2.158
TIN	1.042	1.224	1.229	1.178	0	0.904	0.177	11.987	12.081	5.865	5.973
BV	0.102	0.199	0.202	0.175	0.447	0	0.35	5.937	5.987	2.67	2.727
TOUT	0.731	0.887	0.891	0.847	0.163	0.629	0	10.062	10.142	4.85	4.942
VIR	0.841	0.827	0.827	0.831	0.919	0.854	0.908	0	0.007	0.472	0.463
VHI	0.839	0.826	0.825	0.828	0.914	0.854	0.904	0.007	0	0.475	0.467
HIR	0.691	0.665	0.664	0.671	0.833	0.721	0.809	0.892	0.906	0	0.016
PHI	0.7	0.674	0.673	0.68	0.843	0.726	0.823	0.863	0.877	0.015	0

Table 3. Clustering using the SP-lines. The similarity matrix obtained with the first example from the multivariate TS dataset.

	T1	T2	T3	T4	TIN	BV	TOUT	VIR	VHI	HIR	PHI
T1	0	1	1	1	0	0	0	0	0	0	0
T2	1	0	1	1	0	0	0	0	0	0	0
T3	1	1	0	1	0	0	0	0	0	0	0
T4	1	1	1	0	0	0	0	0	0	0	0
TIN	0	0	0	0	0	0	1	0	0	0	0
BV	0	0	0	0	0	0	0	0	0	0	0
TOUT	0	0	0	0	1	0	0	0	0	0	0
VIR	0	0	0	0	0	0	0	0	1	0	0
VHI	0	0	0	0	0	0	0	1	0	1	1
HIR	0	0	0	0	0	0	0	0	0	0	1
PHI	0	0	0	0	0	0	0	0	0	1	0

Table 4. RNN clustering. Final adjacency matrix obtained using $th2 = 3$.

	T1	T2	T3	T4	TIN	BV	TOUT	VIR	VHI	HIR	PHI
T1	0	1	1	1	0	0	0	0	0	0	0
T2	1	0	1	1	0	0	0	0	0	0	0
T3	1	1	0	1	0	0	0	0	0	0	0
T4	1	1	1	0	0	0	0	0	0	0	0
TIN	0	0	0	0	0	1	0	0	0	0	0
BV	0	0	0	0	0	0	1	0	0	0	0
TOUT	0	0	0	0	0	0	0	0	0	0	0
VIR	0	0	0	0	0	0	0	0	1	0	0
VHI	0	0	0	0	0	0	0	1	0	0	0
HIR	0	0	0	0	0	0	0	0	0	0	0
PHI	0	0	0	0	0	0	0	0	0	0	0

Table 5. Sp-line clustering. Final adjacency matrix obtained using $th2 = 2$.

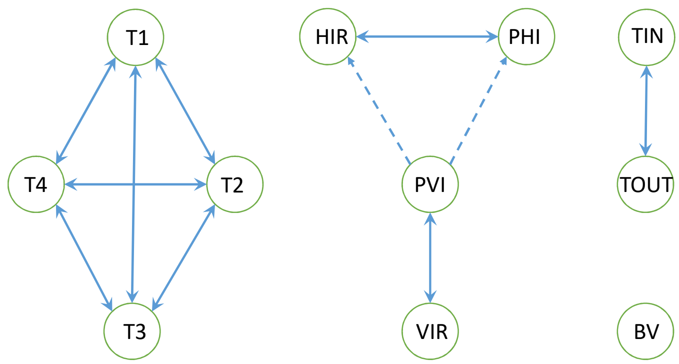

Fig. 2. RNN clustering. Final graph: the groups are clearly remarked.

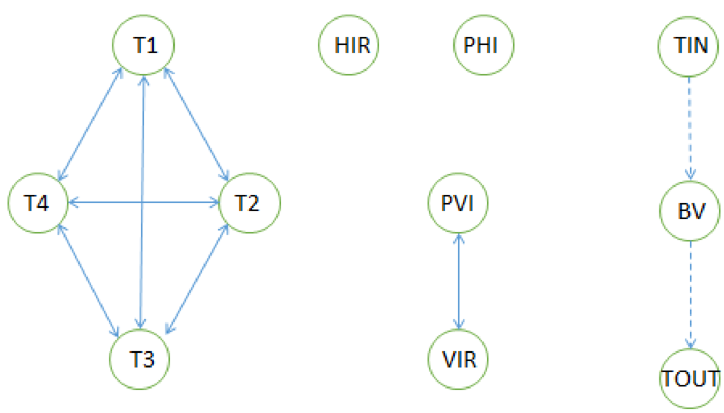

Fig. 3. SP-line clustering. Final graph: the groups are clearly remarked.

5 Conclusions

In this study two solutions for multivariate TS clustering has been compared, both of them using RNN. In the first solution, the prediction error of RNN learned for a variable within an example is used to define a similarity measurement. The aggregation of the obtained similarities among all the examples in the dataset allows developing an adjacency matrix that, finally, is used to group the variables. The second solution introduced an intermediate stage of sp-lines modeling to smooth the evolution of the TS.

A simple proof of concept has been presented, showing that the performance of the method perfectly groups the different variables. Interestingly, the first approach only has two parameters (two thresholds) that were easily tuned, while the second has the extra parameter of the degrees of freedom of the sp-lines.

We do expect to perform improvements in the algorithm, avoiding the use of thresholds, enhancing and improving the modeling method and in the similarity function, so this solution can be applied in data analysis of multivariate TS datasets. Finally, the refined method must be tested with published datasets concerning different domains, such as medicine, management or to the stock market.

ACKNOWLEDGMENT

This research has been funded by the Spanish Ministry of Science and Innovation, under project MINECO-TIN2017-84804-R.

References

1. Aghabozorgi, S., Shirkhorshidi, A.S., Wah, T.Y.: Time-series clustering – a decade review. Information Systems 53, 16 – 38 (2015), http://www.sciencedirect.com/science/article/pii/S0306437915000733
2. Bode, G., Schreiber, T., Baranski, M., Müller, D.: A time series clustering approach for building automation and control systems. Applied Energy 238, 1337 – 1345 (2019), http://www.sciencedirect.com/science/article/pii/S0306261919302089
3. Duan, L., Yu, F., Pedrycz, W., Wang, X., Yang, X.: Time-series clustering based on linear fuzzy information granules. Applied Soft Computing 73, 1053 – 1067 (2018), http://www.sciencedirect.com/science/article/pii/S1568494618305490
4. D'Urso, P., Giovanni, L.D., Massari, R.: Robust fuzzy clustering of multivariate time trajectories. International Journal of Approximate Reasoning 99, 12 – 38 (2018), http://www.sciencedirect.com/science/article/pii/S0888613X17306977
5. Ferreira, A.M.S., de Oliveira Fontes, C.H., Cavalcante, C.A.M.T., Marambio, J.E.S.: Pattern recognition as a tool to support decision making in the management of the electric sector. part ii: A new method based on clustering of multivariate time series. International Journal of Electrical Power and Energy Systems 67, 613 – 626 (2015), http://www.sciencedirect.com/science/article/pii/S0142061514007285

6. Fontes, C.H., Budman, H.: A hybrid clustering approach for multivariate time series – a case study applied to failure analysis in a gas turbine. ISA Transactions 71, 513 – 529 (2017), http://www.sciencedirect.com/science/article/pii/S0019057817305530
7. Hu, M., Feng, X., Ji, Z., Yan, K., Zhou, S.: A novel computational approach for discord search with local recurrence rates in multivariate time series. Information Sciences 477, 220 – 233 (2019), http://www.sciencedirect.com/science/article/pii/S0020025516320849
8. Lee, Y., Na, J., Lee, W.B.: Robust design of ambient-air vaporizer based on time-series clustering. Computers and Chemical Engineering 118, 236 – 247 (2018), http://www.sciencedirect.com/science/article/pii/S0098135418308822
9. Li, J., Pedrycz, W., Jamal, I.: Multivariate time series anomaly detection: A framework of hidden markov models. Applied Soft Computing 60, 229 – 240 (2017), http://www.sciencedirect.com/science/article/pii/S1568494617303782
10. Liu, G., Zhu, L., Wu, X., Wang, J.: Time series clustering and physical implication for photovoltaic array systems with unknown working conditions. Solar Energy 180, 401 – 411 (2019), http://www.sciencedirect.com/science/article/pii/S0038092X19300532
11. Mikalsen, K.Ø., Bianchi, F.M., Soguero-Ruiz, C., Jenssen, R.: Time series cluster kernel for learning similarities between multivariate time series with missing data. Pattern Recognition 76, 569 – 581 (2018), http://www.sciencedirect.com/science/article/pii/S0031320317304843
12. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. on Knowl. and Data Eng. 22(10), 1345–1359 (Oct 2010), http://dx.doi.org/10.1109/TKDE.2009.191
13. Quast, B.: Recurrent neural networks in r. https://github.com/bquast/rnn (February 2019)
14. Salvo, R.D., Montalto, P., Nunnari, G., Neri, M., Puglisi, G.: Multivariate time series clustering on geophysical data recorded at mt. etna from 1996 to 2003. Journal of Volcanology and Geothermal Research 251, 65 – 74 (2013), http://www.sciencedirect.com/science/article/pii/S0377027312000443, flank instability at Mt. Etna
15. Váquez, I., Villar, J.R., Sedano, J., Simic, S.: A preliminary study on multivariate time series clustering. In: Springer (ed.) Proceedings of 14th International Conference on Soft Computing Models in Industrial and Environmental Applications (2019)
16. Yu, C., Luo, L., Chan, L.L.H., Rakthanmanon, T., Nutanong, S.: A fast lsh-based similarity search method for multivariate time series. Information Sciences 476, 337 – 356 (2019), http://www.sciencedirect.com/science/article/pii/S0020025518308430