Interpretable Clustering of Multivariate Time Series with Time2Feat

Angela Bonifati
Lyon 1 University, Liris CNRS
Lyon, France
angela.bonifati@univ-lyon1.fr

Francesco Del Buono
University of Modena and Reggio Emilia
Modena, Italy
francesco.delbuono@unimore.it

Francesco Guerra
University of Modena and Reggio Emilia
Modena, Italy
francesco.guerra@unimore.it

Miki Lombardi
Adobe
Paris, France
mlombardi@adobe.com

Donato Tiano
University of Modena and Reggio Emilia
Modena, Italy
donato.tiano@unimore.it

ABSTRACT

This paper showcases Time2Feat, an end-to-end machine learning system for Multivariate Time Series (MTS) clustering. The system relies on interpretable inter-signal and intra-signal features extracted from the time series. Then, a dimensionality reduction technique is applied to select a subset of features that retain most of the information, thus enhancing the interpretability of the results. In addition, the system enables domain specialists to semi-supervise the process by submitting a small collection of MTS with a target cluster. This process further improves both accuracy and interpretability, by reducing the number of features used by the clustering process. The demonstration shows the application of Time2Feat to various MTS datasets, by creating clusters from MTS datasets of interest, experimenting with different settings and using the approach capabilities to interpret the clusters generated.

PVLDB Reference Format:
Angela Bonifati, Francesco Del Buono, Francesco Guerra, Miki Lombardi, and Donato Tiano. Interpretable Clustering of Multivariate Time Series with Time2Feat. PVLDB, 16(12): 3994 - 3997, 2023.
doi:10.14778/3611540.3611604

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/softlab-unimore/time2feat.

1 INTRODUCTION

A Multivariate Time Series (MTS) is a collection of multiple univariate time series (signals) that are observed simultaneously over time and provide insight into time-dependent phenomena. MTS analysis enables the examination of variable relationships over time, offering important insights into underlying phenomena. This can be useful for modeling complex systems, making data-driven decisions, and improving efficiency and productivity across various domains, such as finance and economics, environmental science, healthcare, and social science [4]. MTS analytics involves both supervised and unsupervised techniques, which cover a range of tasks, including classification, clustering, pattern discovery, and forecasting. Among these, clustering analysis has become popular in applications where sensors generate large amounts of data.

While there has been a lot of research on clustering techniques for univariate time series (UTS), the field of clustering multivariate time series is still in its early stages. Proposals adapt clustering approaches designed for UTS to MTS after applying dimensionality reduction techniques. Examples of such techniques (CSPCA [7] and MC_2PCA [6]) are based on the Principal Component Analysis (PCA), which enables the conversion of a set of correlated features in the high dimensional space into a set of uncorrelated features in the low dimensional space. Nevertheless, the resulting clusters suffer from poor explainability as the original dimensions are lost. More recently, approaches based on Deep Neural Networks (DNNs) [10], and in particular Variational Autoencoders (VAEs)[5, 8] have been used to generate MTS encodings before applying clustering methods. Although these solutions might exhibit high performance, the resulting clusters are based on latent dimensions that remain unexplainable to the end-users.

In this paper, we demonstrate Time2Feat [2], the first system that deals with explainable results of Multivariate Time Series clustering using an end-to-end feature-based pipeline. The pipeline applies clustering techniques to interpretable features automatically extracted from the signals composing the MTS. While the pipeline can be executed without any user interaction (i.e., running the *unsupervised mode*), Time2Feat can also incorporate user annotations on small dataset samples to select generated features. This kind of user's *semi-supervision* can improve the accuracy of results and the quality of explanations.

Through the demonstration, users can experiment with the end-to-end process implemented in the pipeline to generate clusters and explanations from multiple MTS datasets using both unsupervised and semi-supervised modes. The framework also enables users to evaluate the contribution of individual pipeline components to the overall analytics process by comparing results obtained with different settings, such as manual feature selection and application of different clustering techniques. Time2Feat provides a versioning

Proceedings of the VLDB Endowment, Vol. 16, No. 12 ISSN 2150-8097.
doi:10.14778/3611540.3611604

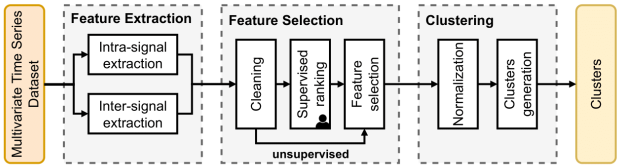


Figure 1: The Time2Feat **pipeline.**

system that allows users to record the experimented settings and revert to previous configurations if necessary.

2 THE TIME2FEAT SYSTEM

The Time2Feat data analysis pipeline is comprised of three distinct components: feature extraction, feature selection, and clustering, as depicted in Figure 1. The input to the pipeline is a Multivariate Time Series, along with the number of desired clusters. If the number of clusters is not explicitly specified, the pipeline can determine it using a heuristic approach.

There are two operational modes for Time2Feat, unsupervised mode and semi-supervised mode. In unsupervised mode, the pipeline requires no additional input beyond the Multivariate Time Series and the number of desired clusters. The pipeline automatically extracts relevant features, selects the most important ones, and performs clustering to identify the number of specified clusters.

In semi-supervised mode, the user has the option of providing a subset of labeled samples to improve the accuracy of the clustering process. This information is utilized to fine-tune the feature selection and clustering steps, ultimately resulting in improved clustering accuracy.

Feature Extraction. The goal of this component is to generate a comprehensive and detailed representation of the MTS in the dataset via the extraction of a large spectrum of features, each describing the signals composing the MTS in isolation or pairs. *Intra-signal Features* are extracted by applying the library `tsfresh` [3], which can extract 700+ features encoding the signal description from the perspective offered by a specific analysis method, such as Distribution Analysis, Statistical Analysis, etc. `tsfresh` extracts features that are interpretable for users who know how the statistical measure summarizes the time series values. Moreover, Time2Feat captures *inter-signal relationships* by quantifying the relatedness between pairs of signals using eight metrics, such as correlation and Euclidean distance. All of the features extracted via this process represent the primary characteristics of the time series. These features are entirely based on statistical calculations, which results in them being fully explainable and interpretable.

Features Extraction at work. The RacketSports dataset[1] describes four kinds of shots performed by people playing badminton or squash, i.e., clear and smash (badminton), boast backhand and boast forehand running(squash). Two sensors (an accelerometer and a gyroscope) gather data in a three-dimensional space, thus producing three signals (X, Y, Z) per sensor. Suppose we are asked to analyze the dataset and no details on the activities that MTS describes are provided to us. Time2Feat through the Feature Extraction component, generates 4722 intra-signal and 120 inter-signal features to describe the dataset. While individually interpretable, the sheer number of features can produce noise in the generation of the clusters and cannot be managed by users for their interpretation.

[1]http://www.timeseriesclassification.com/description.php?Dataset=RacketSports

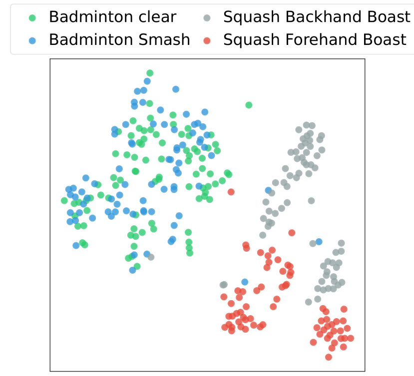


Figure 2: A t-SNE representation of the RacketSports dataset

Feature Selection. The feature extraction process generates a vast number of features, making it necessary to reduce its dimensionality to improve interpretability and clustering performance. If Time2Feat is running with the *semi-supervised mode*, the available labels are used to rank the relevance of the features for identifying a subset capable of generating clusters via an ANOVA based analysis. Irrespective of the availability of labels, the Principal Feature Analysis (PFA) technique is applied to select the most meaningful features. The PFA technique is chosen because it not only ensures conciseness but also promotes diversity by selecting the principal features that retain the maximum variability of the features in the lower-dimensional space. The proposed approach enables the selection of a smaller set of features while retaining the most relevant and diverse information for clustering tasks.

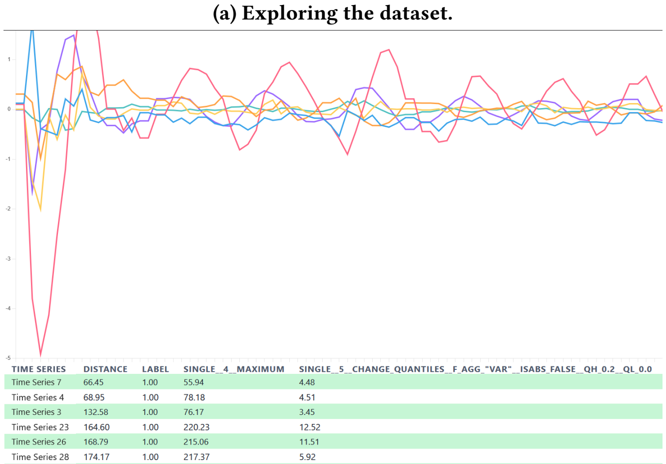
Features Selection at work. The Feature Selection component reduces the number of features to 142 intra-signal and 11 inter-signal features when running in the semi-supervised mode. If the user can provide labels for at least 20% of the MTS, the number of features is reduced to 12 intra-signal and 3 inter-signal features.

Clustering. Time2Feat includes three techniques (Hierarchical, KMeans, Spectral) for generating the clusters. The hierarchical clustering technique is selected as default, since it achieved the best accuracy in our experiments. The Time2Feat system leverages state-of-the-art heuristics (e.g., applying the well-known Elbow method) or user preferences to select the number of clusters to generate. The clusters are wholly derived from the previously selected features. Given the interpretability of these features, it is feasible to analyze the resulting clusters and gain an understanding of the factors that influenced their creation.

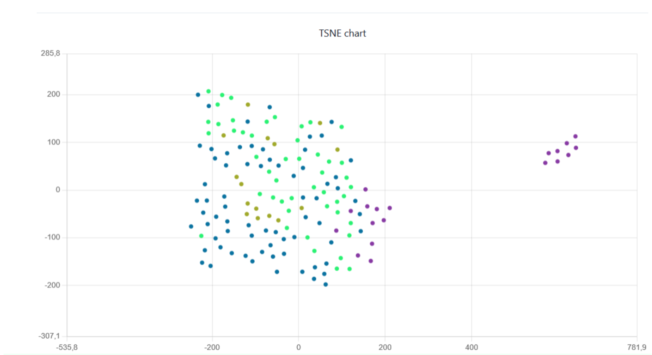
Clustering Component at work. If the user selects to generate 4 clusters and the system runs in the unsupervised mode, the quality

(a) Exploring the dataset.

(b) Clusters (unsupervised mode).



(c) Analizing the clusters.



(d) Getting details from the t-SNE.

Figure 3: Time2Feat **at work.**

of the result measured with the Adjusted Mutual Information (AMI) is around 0.35. The reason for the low quality result (that in any case overcomes the competing approaches [2]) is mainly due to the complexity of the problem. By running the semi-supervised mode and providing 20% of labels the quality of the clusters improves to 0.56. Figure 2 shows a representation of the dataset with a 2 dimensional t-SNE representation [9]. Even if it is possible to discriminate users playing badminton or squash, the difference between the shots cannot be easily identified. If the user selects to generate 2 clusters, Time2Feat is able to better discriminate between the activities. The AMI obtained with the unsupervised approach is 0.77 and the one with the semi-supervised approach is 0.86.

3 DEMONSTRATION OVERVIEW

The demonstration[2] illustrates two key capabilities of the T2F system. Firstly, it showcases how the system can automatically compute clusters of Multivariate Time Series datasets in unsupervised mode, or with users' interaction in semi-supervised mode. Secondly, it demonstrates how users can gain insights into the generated clusters by identifying the primary reasons, namely the most significant features, that led to the assignment of items selected by the users to the specified clusters.

Getting insights from the clustering pipeline. We will introduce the three main software components of the Time2Feat pipeline: the component for the feature extraction, for the feature selection and for the cluster generation. In the unsupervised mode, users only need to input their MTS datasets, and Time2Feat automatically generates clusters. We will showcase the pipeline on pre-loaded datasets (including 18 benchmark datasets from the UEA multivariate time series classification archive[1]), but users can also provide their own. The number of clusters can be determined automatically using the Elbow method or manually selected by the user. The pipeline returns the results in the form of an associative array, which can be visualized in a 2-dimensional graph through t-SNE transformation of the features. The semi-supervised mode provides

[2]See https://youtu.be/myrwmTyYdwM for a preview.

more accurate results generated by relying on fewer features, but requires some user's interaction. In particular, s/he has to provide clusters along with samples of belonging MTS. To support the sample identifications, we allow the users to start from the clusters generated via the unsupervised mode, and by means of a graphical interface, they have only to confirm for a predefined number of MTS that they belong to the same cluster. The pipeline is then executed and the clusters shown as in the unsupervised mode. The Time2Feat pipeline is fully modular, with predefined implementations for each component (tsfresh for feature extraction, PFA for feature selection, and hierarchical clustering for clustering). However, users are free to experiment with alternative implementations for each component.

Motivating Scenario 1. Figure 3 shows part of the Time2Feat GUI computing the clusters for the MTS RacketSports datasets. As a first operation, the dataset needs to be loaded into the pipeline. Time2Feat accepts the format (e.g., a list of csv time series in different files) usually adopted in the field. The pipeline starts running the unsupervised mode, where users have only to select the clustering technique to apply (e.g., KMeans, Hierarchical, Spectral) and the number of clusters to generate (the Elbow method is applied if no value is provided and either the Hierarchical clustering technique is selected). Time2Feat allows the users to evaluate the Time Series in the dataset through plots (Figure 3a). The next steps in the pipeline consist in the extraction and selection of meaningful features. No input is required from the users who can inspect the results of the process visualizing plots and analyzing a tabular representation. Time2Feat computes the clusters and the GUI shows for each time series the assigned cluster (Figure 3b). In case users are not satisfied by the results achieved, they can proceed with the semisupervised mode. We implemented a simple and quick technique, where users have only to confirm the clusters computed for some manually selected time series by flagging a checkbox. Users can associate meaningful names to the clusters. Time2Feat leverages the users' annotation to compute more accurate clusters.

Exploring the clusters. Upon running the Time2Feat pipeline and generating the clusters, the user is presented with two representations of the MTS in each cluster: a tabular representation and a graphical representation using the t-SNE algorithm. The tabular representation allows the user to visualize the key features characterizing the MTS belonging to each cluster. Furthermore, by selecting a specific MTS of interest, the user can employ it as the reference point to sort the other MTS in the cluster based on their distance from the reference. This feature enables the user to discern commonalities and differences among the MTS within the same cluster. Moreover, the distance between the selected MTS and the centroids of the cluster is displayed to facilitate comparisons with other elements in the group and other groups of MTS This approach provides a comprehensive understanding of the clustering results and can aid in data-driven decision making in a variety of domains.

Motivating Scenario 2. Users can inspect the computed clusters via a tabular (Figure 3c) and a t-SNE (Figure 3d) representation. The t-SNE is provided with a re-load button to change the seed used for the plot generation. Moreover, the points in the Figure are clickable: users can see the features of selected time series. Finally, a tabular representation of a users' selected cluster in Figure 3c can

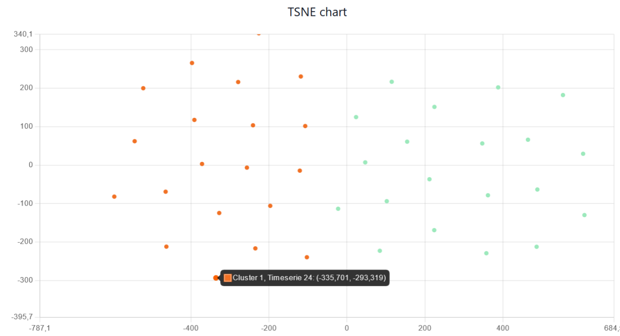


Figure 4: Getting details from the t-SNE.

show all time series belonging to the cluster, order the time series by distance (we compute the feature-to-feature euclidean distance) from a user selected reference time series, order the time series by distance from the cluster centroid. In case the users are not satisfied by the results achieved, they can re-start the process, and provides more annotations and/or change the number of clusters to compute. The t-SNE representation in Figure 4 shows the results obtained if the user decides to generate two clusters instead of four. The clusters obtained are now clearly separated.

ACKNOWLEDGMENTS

This work was partially supported by ANR (grant nr. 18-CE23-0002 QualiHealth) and by the by the Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia (project AnomalyFeats - FARD2023).

REFERENCES

[1] Anthony J. Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn J. Keogh. 2018. The UEA multivariate time series classification archive, 2018. *CoRR* abs/1811.00075 (2018).

[2] Angela Bonifati, Francesco Del Buono, Francesco Guerra, and Donato Tiano. 2022. Time2Feat: Learning Interpretable Representations for Multivariate Time Series Clustering. *Proc. VLDB Endow.* 16, 2 (2022), 193–201.

[3] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W. Kempa-Liehr. 2018. Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh - A Python package). *Neurocomputing* 307 (2018), 72–77.

[4] Francesco Del Buono, Francesca Calabrese, Andrea Baraldi, Matteo Paganelli, and Francesco Guerra. 2022. Novelty Detection with Autoencoders for System Health Monitoring in Industrial Environments. *Applied Sciences* 12, 10 (2022). https://doi.org/10.3390/app12104931

[5] Dino Ienco and Roberto Interdonato. 2020. Deep Multivariate Time Series Embedding Clustering via Attentive-Gated Autoencoder. In *PAKDD 2020 - 24th Pacific-Asia Conference on Knowledge Discovery and Data Mining (Advances in Knowledge Discovery and Data Mining)*. Singapore, Singapore. https://hal.inrae.fr/hal-02923636

[6] Hailin Li. 2019. Multivariate time series clustering based on common principal component analysis. *Neurocomputing* 349 (2019), 239–247.

[7] Hailin Li, Chunpei Lin, Xiaoji Wan, and Zhengxin Li. 2019. Feature representation and similarity measure based on covariance sequence for multivariate time series. *IEEE Access* 7 (2019), 67018–67026.

[8] Laura Manduchi, Matthias Hüser, Julia Vogt, Gunnar Rätsch, and Vincent Fortuin. 2019. DPSOM: Deep probabilistic clustering with self-organizing maps. *arXiv preprint arXiv:1910.01590* (2019).

[9] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605. http://www.jmlr.org/papers/v9/vandermaaten08a.html

[10] Chenxiao Xu, Hao Huang, and Shinjae Yoo. 2021. A Deep Neural Network for Multivariate Time Series Clustering with Result Interpretation. In *IJCNN*. IEEE, 1–8.