Novel features for time series analysis: a complex networks approach

Vanessa Freitas Silva[1] · **Maria Eduarda Silva**[2] · **Pedro Ribeiro**[1] ·
Fernando Silva[1]

Abstract

Being able to capture the characteristics of a time series with a feature vector is a very important task with a multitude of applications, such as classification, clustering or forecasting. Usually, the features are obtained from linear and nonlinear time series measures, that may present several data related drawbacks. In this work we introduce *NetF* as an alternative set of features, incorporating several representative topological measures of different complex networks mappings of the time series. Our approach does not require data preprocessing and is applicable regardless of any data characteristics. Exploring our novel feature vector, we are able to connect mapped network features to properties inherent in diversified time series models, showing that *NetF* can be useful to characterize time data. Furthermore, we also demonstrate the applicability of our methodology in clustering synthetic and benchmark time series sets, comparing its performance with more conventional features, showcasing how *NetF* can achieve high-accuracy clusters. Our results are very promising, with network features from different mapping methods capturing different properties of the time series, adding a different and rich feature set to the literature.

✉ Vanessa Freitas Silva
    vanessa.silva@dcc.fc.up.pt

    Maria Eduarda Silva
    mesilva@fep.up.pt

    Pedro Ribeiro
    pribeiro@fc.up.pt

    Fernando Silva
    fmsilva@fc.up.pt

[1] CRACS-INESC TEC, Faculdade de Ciências, Universidade do Porto, Porto, Portugal

[2] LIAAD-INESC TEC, Faculdade de Economia, Universidade do Porto, Porto, Portugal

1 Introduction

Time series, which can be thought of as collections of observations indexed by time, are ubiquitous in all domains from climate studies or health monitoring to financial data analysis. There is a plethora of statistical models in the literature adequate to describe the behaviour of time series (Shumway and Stoffer 2017). However, technological developments, such as sensors and mobile devices, lead to the gathering of large amounts of high dimensional time indexed data for which appropriate methodological and computational tools are required. With this purpose, recently, feature-based time series characterization has become a popular approach among time series data researchers (Fulcher 2018; Henderson and Fulcher 2021; Wang et al. 2006) and proved useful for a wide range of temporal data mining tasks ranging from classification (Fulcher and Jones 2014), clustering (Wang et al. 2006), forecasting (Montero-Manso et al. 2020; Talagala et al. 2018), pattern detection (Geurts 2001), outlier or anomalies detection (Hyndman et al. 2015), motif discovery (Chiu et al. 2003), visualization (Kang et al. 2017) and generation of new data (Kang et al. 2020), among others.

The main idea behind feature-based approaches is to construct feature vectors that aim to represent specific properties of the time series data by characterizing the underlying dynamic processes (Fulcher 2018; Fulcher and Jones 2017). The usual methodologies for calculating time series features include concepts and methods from the linear time series analysis literature (Shumway and Stoffer 2017), such as autocorrelation, stationarity, seasonality and entropy, but also methods of nonlinear time-series analysis based on dynamic systems theory (Fulcher et al. 2013; Henderson and Fulcher 2021; Wang et al. 2006). These methods usually involve parametric assumptions, parameter estimation, non-trivial calculations and approximations, as well as preprocessing tasks such as finding time series components, differencing and whitening thus presenting drawbacks and computation issues related to the nature of the data, such as the length of the time series.

This work contributes to the feature-based approach in time series analysis by proposing an alternative set of features based on complex networks concepts.

Complex networks describe a wide range of systems in nature and society by representing the existing interactions via graph structures (Barabási 2016). Network science, the research area that studies complex networks, provides a vast set of topological graph measurements (Costa et al. 2007; Peach et al. 2021), a well-defined set of problems such as community detection (Fortunato 2010) or link prediction (Lü and Zhou 2011), and a large track record of successful application of complex network methodologies to different fields (Vespignani 2018), including graph classification (Bonner et al. 2016).

Motivated by the success of complex network methodologies and with the objective of acquiring new tools for the analysis of time series, several network-based approaches have been recently proposed. These approaches involve mapping time series to the

network domain. The mappings methods proposed in the literature may be divided into one of three categories depending on the underlying concept: proximity, visibility and transition (Silva et al. 2021; Zou et al. 2019). Depending on the mapping method, the resulting networks capture specific properties of the time series. Some networks have as many nodes as the number of observations in the time series, as visibility graphs (Lacasa et al. 2008), while others allow to reduce the dimensionality preserving the characteristics of the time dynamics, as the quantile graphs (Campanharo et al. 2011). Network-based time series analysis techniques have been showing promising results and have been successful in the description, classification and clustering of time series. Examples include automatic classification of sleep stages (Zhu et al. 2014), characterizing the dynamics of human heartbeat (Shao 2010), distinguishing healthy from non-healthy electroencephalographic series (Campanharo and Ramos 2017) and analysing seismic signals (Telesca and Lovallo 2012).

In this work we establish a new set of time series features, *NetF*, by mapping the time series into the complex networks domain. Further, we propose a procedure for time series mining tasks and address the question whether time series features based on complex networks are a useful approach in time series mining tasks. Our proposed procedure, represented in Fig. 1 comprises the following steps: map the time series into (natural and horizontal) visibility graphs and quantile graphs using appropriate mapping methods and compute five specific topological measures for each network, thus establishing a vector of 15 features. These features are then used in mining tasks. The network topological metrics selected, average weighted degree, average path length, number of communities, clustering coefficient and modularity, measure global characteristics, are simple to compute and to interpret in the graph context and commonly used in network analysis, thus capable of providing useful information about the structure and properties of the underlying systems.
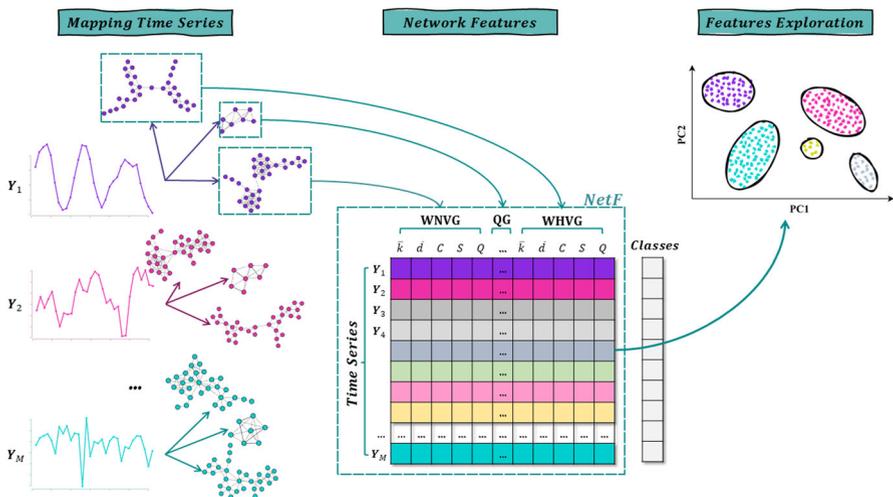

Fig. 1 Schematic diagram of the network based features approach to time series mining tasks

To investigate the relevance of the set of features *NetF* we analyse synthetic time series generated from a set of Data Generating Processes with a range of different characteristics. Additionally we consider the problem of time series clustering from a feature-based approach in synthetic, benchmark and a new time series data sets. *NetF* features are assessed against two other sets of features, *tsfeatures* and *catch22* (Hyndman et al. 2020; Kang et al. 2017; Lubba et al. 2019). The results show that network science measures are able to capture and preserve the characteristics of the time series data. We show that different topological measures from different mapping methods capture different characteristics of the data, complementing each other and providing new information when combined, rather than considered by themselves as is common in the literature. Clustering results of empirical data are balanced when compared to conventional approaches, in some data sets the proposed approach obtains better results, and in other data sets the results are quite similar between the approaches. The proposed set of features have the advantage of being always computable, which is not always possible using classical time series features.

NetF, the empirical study implementation, and the data sets presented here are available on GitHub.[1]

We have organized this paper as follows. Section 2 introduces basic concepts of time series and complex networks, setting the notation for the remainder of the paper, and also presents the mapping methods and networks measurements used. Next, in Sect. 3 the novel features of the time series proposed in this work are presented and a study of these features is carried out in order to characterize properties of the time series. In Sect. 4 the time series clustering tasks are performed as an example of application of network-based features, synthetic and empirical data sets are used, they are briefly described and compared to two other classical time series approaches. The results corresponding to the three approaches are presented. Finally, Sect. 5 presents the conclusions and some comments.

2 Background

2.1 Time series

A time series $Y = (Y_1, \ldots, Y_T)$ is a set of observations obtained over time, usually at equidistant time points. A time series differs from a random sample in that the observations are ordered in time and usually present serial correlation that must be account for in all statistical and data mining tasks. Time series analysis refers to the collection of procedures developed to systematically solve the statistical problems posed by the serial correlation. Statistical time series analysis relies on a set of concepts, measures and models designed to capture the essential characteristics of the data, namely, trend, seasonality, periodicity, autocorrelation, skewness, kurtosis and heteroscedasticity (Wang et al. 2006). Other concepts like self-similarity, non-linearity structure and chaos, stemming from non-linear science are also used to characterize time series (Bradley and Kantz 2015).

---

[1] https://github.com/vanessa-silva/NetF.

Several classes of statistical models that provide plausible descriptions of the characteristics of the time series data have been developed with a view to forecasting and simulation (Box et al. 2015). The statistical models for time series may be broadly classified as *linear* and *nonlinear*, referring usually to the functional forms of conditional mean and variance. *Linear time series models* are models for which the conditional mean is a linear function of past values of the time series. The most popular class are the AutoRegressive Moving Average (ARMA) models. As particular cases of ARMA models we have: the white noise (WN), a sequence of independent and identically distributed observations, the AutoRegression (AR) models, which specify a linear relationship between the current and past values and the Moving Average (MA) models, which specify the linear relationship between the current value and past stochastic terms. ARMA models have been extended to incorporate non-stationarity (unit root) ARIMA models and long memory characteristics, ARFIMA models. Many time series data present characteristics that cannot be represented by linear models such as volatility, asymmetry, different regimes and clustering effects. To model these effects, non-linear specifications for the conditional mean and for the conditional variance lead to different classes of *nonlinear time series models*, such as Generalized AutoRegressive Conditional Heteroskedastic (GARCH) type models specified by the conditional variance and developed mainly for financial time series, threshold models and Hidden Markov models that allow for different regimes and models for integer valued time series, INAR. Definitions, properties and details about these models are given in Appendix A.

2.2 Complex networks

Graphs are mathematical structures appropriate for modelling complex systems which are characterized by a set of elements that interact with each other and exhibit collective properties (Costa et al. 2011). Typically, graphs exhibit non-trivial topological properties due to the characteristics of the underlying system, and so they are often called complex networks.

A graph (or network), G, is an ordered pair (V, E), where V represents the set of *nodes* and E the set of *edges* between pairs of elements of V. Two nodes v_i and v_j are neighbours if they are connected by an edge $(v_i, v_j) \in E$. The edges can be termed as *directed*, if the edges connect a source node to a target node, or *undirected*, if there is no such concept of orientation. A graph is also termed *weighted* if there is a weight, $w_{i,j}$, associated with the edge (v_i, v_j).

Network science has served many scientific fields in problem solving and analyzing data that is directly or indirectly converted to networks. Currently, there is a vast literature on problems and successful applications (Vespignani 2018), as well as an extensive set of measurements of topological, statistical, spectral and combinatorial properties of networks (Albert and Barabási 2002; Barabási 2016; Costa et al. 2007; Peach et al. 2021), capable of differentiating particular characteristics of the network data. Examples include measures of node centrality (Oldham et al. 2019), graph distances (Li et al. 2021), clustering and community (Malliaros and Vazirgiannis 2013), among an infinity of them. Many of these topological measurements involve the con-

cepts of *paths* and graph *connectivity*. A path is a sequence of distinct edges that connect consecutive pairs of nodes. And, consequently, two nodes are said to be connected if there is a path between them and disconnected if no such path exists. Thus, some measurements are based on the length (number of edges) of such connecting paths (Costa et al. 2007).

2.3 Mapping time series into complex networks

In the last decade several network-based time series analysis approaches have been proposed. These approaches are based on mapping the time series into the network domain. The mappings proposed in the literature are essentially based on concepts of visibility, transition probability and proximity (Silva et al. 2021; Zou et al. 2019). In this work we use visibility graph and quantile graph methods which are based on visibility and transition probability concepts, respectively. Next, we briefly describe these methods.

2.3.1 Visibility graphs

Visibility graphs (VG) establish connections (edges) between the time stamps (nodes) using visibility lines between the observations, where nodes are associated with the natural ordering of observations. Two variants of this method are as follows.

The *Natural Visibility Graph* (NVG) (Lacasa et al. 2008) is based on the idea that each observation, Y_t, of the time series is seen as a vertical bar with height equal to its numerical value and that these bars are laid in a landscape where the top of a bar is visible from the tops of other bars. Each time stamp, t, is mapped into a node in the graph and the edges (v_i, v_j), for $i, j = 1 \ldots T, i \neq j$, are established if there is a line of visibility between the corresponding data bars that is not intercepted. Formally, two nodes v_i and v_j are connected if any other observation (t_k, Y_k) with $t_i < t_k < t_j$ satisfies the equation:

$$Y_k < Y_j + (Y_i - Y_j)\frac{(t_j - t_k)}{(t_j - t_i)}. \tag{1}$$

We give a simple illustration of this algorithm in Fig. 2.

NVGs are always connected, each node v_i sees at least its neighbors v_{i-1} and v_{i+1}, and are always undirected unless we consider the direction of the time axis (Silva et al. 2021). The network is also invariant under affine transformations of the data (Lacasa et al. 2008) because the visibility criterion is invariant under rescheduling of both the horizontal and vertical axis, as well as in vector translations, that is, each transformation $Y' = aY + b$, for $a \in \mathbb{R}$ and $b \in \mathbb{R}$, leads to the same NVGs (Silva et al. 2021).

Eventual sensitivity of NVGs to noise is attenuated by assigning a weight to the edges. Define $w_{i,j} = 1/\sqrt{(t_j - t_i)^2 + (Y_j - Y_i)^2}$ the weight associated with the edge (v_i, v_j) (Bianchi et al. 2017). This weight is related to the Euclidean distance between the points (t_i, Y_i) and (t_j, Y_j). Thus, the resulting network from *weighted* NVG (WNVG) method is a weighted and undirected graph.
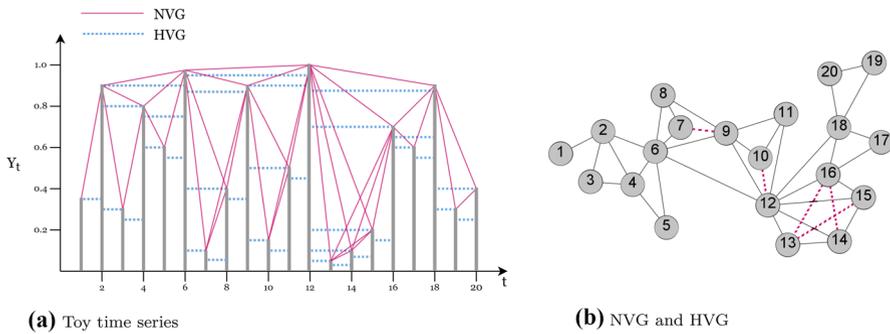
(a) Toy time series   **(b)** NVG and HVG

Fig. 2 Illustrative example of the two visibility graph algorithms. **a** Toy time series and corresponding visibility lines between data bars. Solid pink lines represent the natural visibility lines corresponding to the NVG method, and dashed blue lines represent the horizontal visibility lines corresponding to the HVG method. **b** Network generated by the corresponding mappings. The NVG is the graph with all edges, including the edges represented by the dashed pink lines, and the HVG is the subgraph that does not include these edges. *Source*: Modified from Silva et al. (2021)

The *Horizontal Visibility Graph* (HVG) (Luque et al. 2009) is a simplification of the NVG method whose construction differs in the visibility definition: the visibility lines are only horizontal (see Fig. 2). Two nodes v_i and v_j are connected if, for all (t_k, Y_k) with $t_i < t_k < t_j$, the following condition is met:

$$Y_i, Y_j > Y_k. \tag{2}$$

Given a time series, its HVG is always a subgraph of its NVG. This is illustrated in Fig. 2b where all edges present in the HVG are also present in the NVG, but the converse is not true, the edges represented by dashed pink lines. HVG nodes will always have a degree less than or equal to that of the corresponding NVG nodes. Therefore, there is some loss of quantitative information in HVG in comparison with NVG (Luque et al. 2009). However, in terms of qualitative characteristics, the graphs preserve part of the data information, namely, the local information (the closest time stamps) (Silva et al. 2021).

In a similar way to WNVG, we can assign weights to the edges of the HVG, $w_{i,j} = 1/\sqrt{(t_j - t_i)^2 + (Y_j - Y_i)^2}$, resulting in a *weighted* HVG (WHVG).

2.3.2 Quantile graphs

Quantile Graph (QG) (Campanharo et al. 2011) are obtained from a mapping based on transition probabilities. The method consists in assigning the time series observations to bins that are defined by η sample quantiles, $q_1, q_2, \ldots, q_\eta$. Each sample quantile, q_i, is mapped to a node v_i of the graph and the edges between two nodes v_i and v_j are directed and weighted, $(v_i, v_j, w_{i,j})$, where $w_{i,j}$ represents the transition probability between quantile ranges. The adjacency matrix is a Markov transition matrix:
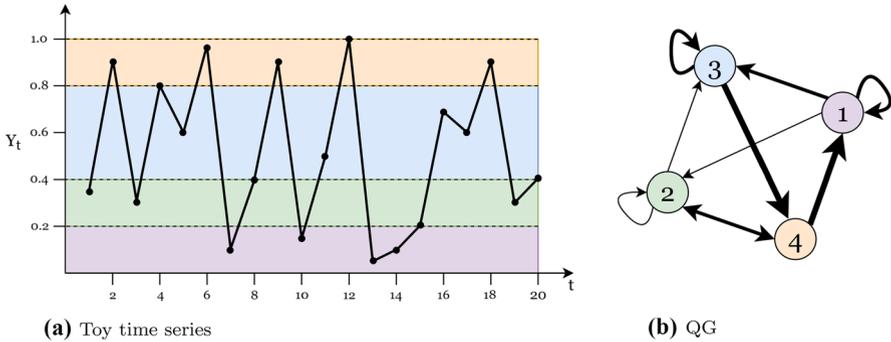
(a) Toy time series   **(b)** QG

Fig. 3 Illustrative example of the quantile graph algorithm for $\eta = 4$. **a** Toy time series with coloured regions representing the different η sample quantiles; **b** network generated by the QG algorithm. Repeated transitions between quantiles result in edges with larger weights represented by thicker lines. *Source*: Reproduced from Silva et al. (2021)

$\sum_{j=1}^{\eta} w_{i,j} = 1$, for each $i = 1, \ldots, \eta$, and the network is weighted, directed and contains self-loops.[2] We illustrate this mapping method in Fig. 3.

The number of quantiles is usually much less than the length of the time series ($\eta \ll T$). If η is too large the resulting graph may not be connected, having isolated nodes,[3] and if it is too small the QG may present a significant loss of information, represented by large weights assigned to self-loops. The causal relationships contained in the process dynamics are captured by the connectivity of the QG.

2.4 Complex networks topological measures

Complex networks have specific topological features which characterize the connectivity between their nodes and, consequently, are somehow reflected in the measurement processes (Costa et al. 2007). There is a wide range of network topology measures capable of expressing the most relevant features of a network. They include global network measurements, which measure global properties involving all elements of the network, node-level or edge-level measurements, which measure a given feature corresponding to the nodes or edges, and "intermediate" measurements, which measure features of subgraphs in the network.

Properties of centrality, distance, community detection and connectivity are central to understanding features of network structures. Centrality measures aim to quantify the importance of nodes and edges in the network depending on their connection topology. Path-based measures refer to sequences of edges that connect pairs of nodes, depend on the overall network structure and are useful for measuring network efficiency and information propagation capability. Communities and node connectivity are also very relevant features of networks, which measure how and which groups of nodes are better connected, measuring the clustering and resilience of the network.

---

[2] A self-loop is an edge that connects a node to itself.

[3] An isolated node is an node that is not connected by an edge to any other node.

In this work we propose to study the average weighted degree, $\bar{k}$, average path length, $\bar{d}$, global clustering coefficient, C, number of communities, S, and modularity, Q, representing global measures of the features described above.

The *degree*, k_i, of a node v_i represents the number of edges of v_i. It is a fairly important property that shows the intensity of connectivity in the node neighbourhood. In directed graphs we distinguish between *in-degree*, k_i^{in}, the number of edges that point to v_i, and *out-degree*, k_i^{out}, the number of edges that point from v_i to other nodes. The total degree is given by $k_i = k_i^{in} + k_i^{out}$. For weighted graphs, we can calculate the *weighted degree* by adding the edge weights instead of the number of edges. *Average path length*, $\bar{d}$, is the arithmetic mean of the shortest paths, $d_{i,j}$, among all pairs of nodes, where the path length is the number of edges, or the sum of the edges weights for weighted graphs, in the path. It is a good measure of the efficiency of information flow on a network. The *global clustering coefficient*, C, also known as *transitivity*, is a measure which captures the degree to which the nodes of a graph tend to cluster, that is, the probability that two nodes connected to a given node are also connected. In this work, we refer to network communities as the grouping of nodes (potentially overlapping) that are densely connected internally and that can also be considered as a group of nodes that share common or similar characteristics. The *number of communities*, S, is the amount of these groups on the network. The *modularity*, Q, measures how good a specific division of the graph is into communities, that is, how different are the different nodes belonging to different communities. A high value of Q indicates a graph with a dense internal community structure and sparse connections between nodes of different communities.

3 *NetF*: a novel set of time series features

Over the last decades, several techniques for extracting time series features have been developed, see (Barandas et al. 2020; Christ et al. 2018; Fulcher and Jones 2017; Fulcher et al. 2013; Hyndman et al. 2020; Lubba et al. 2019; O'Hara-Wild et al. 2021) for more details. Most of these techniques have in common the definition of a finite set of statistical features, such as autocorrelation, existence of unit roots, periodicity, nonlinearity, volatility among others, to capture the global nature of the time series.

In this work we introduce *NetF* as an alternative set of features. Our approach differs from those previously mentioned in that we leverage the usage of different complex network mappings to offer a set of time series features based on the topology of those networks. One of the main advantages of this approach comes from the fact that the mapping methods (Sect. 2.3) do not require typical time series preprocessing tasks, such as decomposing, differencing or whitening. Moreover, our methodology is applicable to any time series, regardless of its characteristics.

3.1 The 15 features of *NetF*

NetF is constituted by 15 different features, as depicted in Fig. 4. These features correspond to the concatenation of five different topological measures, as explained in
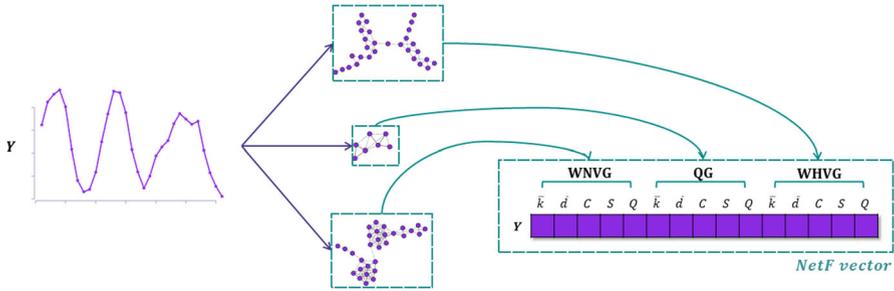
Fig. 4 Schematic diagram of the *NetF* set of features. A time series **Y** is mapped into three complex networks (WNVG, WHVG and QG) and for each of these networks five topological measures are taken ($\bar{k}$, $\bar{d}$, C, S and Q), resulting in the *NetF* vector containing 15 features

Sect. 2.4 ($\bar{k}$, the average weighted degree; $\bar{d}$, the average path length; C, the clustering coefficient; S, the number of communities; Q, the modularity), each of them applied to three different mappings of the time series, as explained in Sect. 2.3 (*WNVG*, the weighted natural visibility graph; *WHVG*, the weighted horizontal visibility graph; *QG*, the quantile graph).

Our main goal is to provide a varied set of representative features that expose different properties captured by the topology of the mapped networks, providing a rich characterization of the underlying time series.

The topological features themselves were selected so that they represent global measures of centrality, distance, community detection and connectivity, while still being accessible, easy to compute and widely used in the network analysis domain.

3.2 Implementation details

Conceptually, *NetF* does not depend on the actual details of how it is computed. Nevertheless, with the intention of both showing the practicality of our approach, as well as providing the reader with the ability to reproduce our results, we now describe in detail how we computed the *NetF* set of features in the context of this work.

To compute the WNVGs we implement the *divide & conquer* algorithm proposed in Lan et al. (2015) and for the WHVGs the algorithm proposed in Luque et al. (2009).[4] To both we added the weighted version mentioned in Sect. 2.3.1, adding the respective weights to the edges. For the QGs we chose $\eta = 50$ quantiles, as in Campanharo and Ramos (2016), and we implemented the method described in Sect. 2.3.2 to create the nodes and edges of the networks. We used the sample quantile method, which uses a scheme of linear interpolation of the empirical distribution function (Hyndman and Fan 1996), to calculate the sample quantiles (nodes) in support of the time series. To save the network structure as a graph structure, we used the igraph (Csardi and Nepusz 2006) package which also allows us to calculate the topological measures. Next, we briefly describe the methods and algorithms used by the functions we used to calculate the measures.

---

[4] https://sites.google.com/view/lucaslacasa/research-topics/visibility-graphs#h.27cb86b8e1ba43fd_148.

The **average weighted degree** ($\bar{k}$) is calculated by the arithmetic mean of the weighted degrees k_i of all nodes in the graph. In this work, the **average path length** ($\bar{d}$) follows an algorithm that does not consider edge weights, and use the breadth-first search algorithm to calculate the shortest paths $d_{i,j}$ between all pairs of vertices, both ways for directed graphs. For calculate the **clustering coefficient** (C), the function that we use in this work ignores the edge direction for directed graphs. For this reason, before we calculate C for QGs, which are directed graphs, we first transform them into an undirected graph, where for each pair of nodes which are connected with at least one directed edge the edge is converted to an undirected edge. And then, the C is calculated by the ratio of the total number of closed triangles[5] in the graph to the number of triplets.[6] The function we use in this work to calculate the **number of communities** (S) in a network, calculates densely connected subgraphs via random walks, such that short random walks tend to stay in the same community. See the Walktrap community finding algorithm (Pons and Latapy 2005) for more details. And to calculate the **modularity** (Q) of a graph in relation to some division of nodes into communities we measure how separated are the nodes belonging to the different communities are as follows:

$$Q = \frac{1}{2|E|} \sum_{i,j} \left[w_{i,j} - \frac{k_i k_j}{2|E|} \right] \delta \left(c_i, c_j \right),$$

where $|E|$ is the number of edges, c_i and c_j the communities of v_i and v_j, respectively, and $\delta(c_i, c_j) = 1$ if v_i and v_j belong to the same community ($c_i = c_j$) and $\delta(c_i, c_j) = 0$ otherwise. We performed all implementations and computations in R (R Core Team 2020), version 4.0.3 and a set of packages.

For reproducibility purposes, the source code and the datasets are made available in https://github.com/vanessa-silva/NetF.

3.3 Empirical evaluation

In this section we investigate, via synthetic data sets, whether the set of features introduced above are useful for characterizing time series data.

To this end, we consider a set of eleven linear and nonlinear time series models, denoted by Data Generating Processes (DGP), which present a wide range of characteristics summarized in Table 1. A detailed description of the DGP and computational details are given in Appendix A. For each of the DGP's in Table 1 we generated 100 realizations of length $T = 10000$. Following the steps presented in Fig. 1, we map each realization into three networks and extract the corresponding topological measures. The resulting time series features, organized by mapping, are summarized, mean and standard deviation, in Tables 5 to 7. Note that the values have been Min-Max normalized for comparison purposes since the range of the different features vary across models.

---

[5] A triangle is a set of three nodes with edges between each pair of nodes.

[6] A triplet is a set of three nodes with at least edges between two pairs of nodes.

Table 1 Summary about the data generating process (time series models) of the synthetic data

Process	Parameters	Main Property	Notation
White Noise	$\epsilon_t \sim N(0, 1)$	Noise effect	`WN`
AR(1)	$\phi_1 \in \{-0.5, 0.5\}$	Smoothness	`AR(1)-0.5`
			`AR(1)0.5`
AR(2)	$\phi_1 = 1.5, \phi_2 = -0.75$	Pseudo-periodic	`AR(2)`
ARIMA(1, 1, 0)	$\phi_1 = 0.7$	Stochastic trend	`ARIMA`
ARFIMA(1, 0.4, 0)	$\phi_1 = 0.5$	Long memory effect	`ARFIMA`
SETAR(1)	$\alpha = 0.5, \beta = -1.8, \gamma = 2,$	Regime-dependent	`SETAR`
	$r = -1$	autocorrelation[a]	
Poisson-HMM	$N = 2, \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix} \lambda \in \{10, 15\}$	State transitions	`HMM`
GARCH(1, 1)	$\omega = 10^{-6}, \alpha_1 = 0.1,$	Persistent periods of high	`GARCH`
	$\beta_1 = 0.8$	or low volatility	
EGARCH(1, 1)	$\omega = \left(10^{-6} - 0.1\sqrt{2/\pi}\right),$	Asymmetric effects of	`EGARCH`
	$\alpha_1 = 0.1, \beta_1 = 0.01, \gamma_1 = 0.3$	positive and negative shock	
INAR(1)	$\alpha = 0.5, \epsilon_t \sim Po(1)$	Correlated counts	`INAR`

Parameters, main characteristic of the data sets and notation is also included. See Appendix A for more details

[a] The two regimes have quite different autocorrelation properties: in the first the correlation is positive while in the second alternates between positive and negative values.
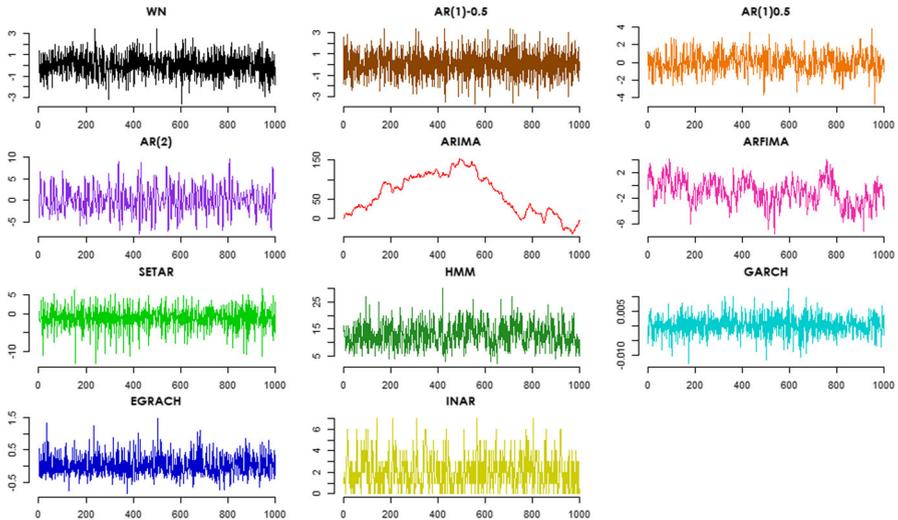
Fig. 5 Plot of one instance of each simulated time series model

WNVGs (Table 5) present lowest values for the clustering coefficient (C) for ARIMA models. Models producing time series with more than one state (HMM and SETAR) present lower average weighted degree but higher number of communities (S). The later values are comparable to those for AR(2) time series, fact that can be explained by the pseudo-periodic nature of the particular AR(2) model entertained here. WHVGs (Table 6) present average weighted degree ($\bar{k}$) approximately 0 for HMM's and approximately 1 for GARCH and EGARCH. This indicates that HMM time series have, on average, horizontal visibility for more distant points (in time and/or value), while the opposite is true for heteroskedastic time series. The clustering coefficient (C) is lowest (approximately 0) for networks obtained from INAR time series, indicating that most points have visibility only for their two closest neighbors. QGs (Table 7) present high values of average path length, ($\bar{d}$), for ARIMA, contrasting with all other DGP which present low values. On the other hand, the (C) for ARIMA presents low values while all other DGP's present high values.

The next step is to study the feature space to understand which network features capture specific properties of the time series models. Figure 6 represents a bi-plot obtained using the 15 features (5 for each mapping method) and with the two PC's explaining 68.8% of the variance. It is noteworthy that the eleven groups of time series models are clearly identified and arranged in the bi-plot according to their main characteristics. Overall, we can say that the number of communities of VGs, S, are positively correlated among themselves and are negatively correlated with the average
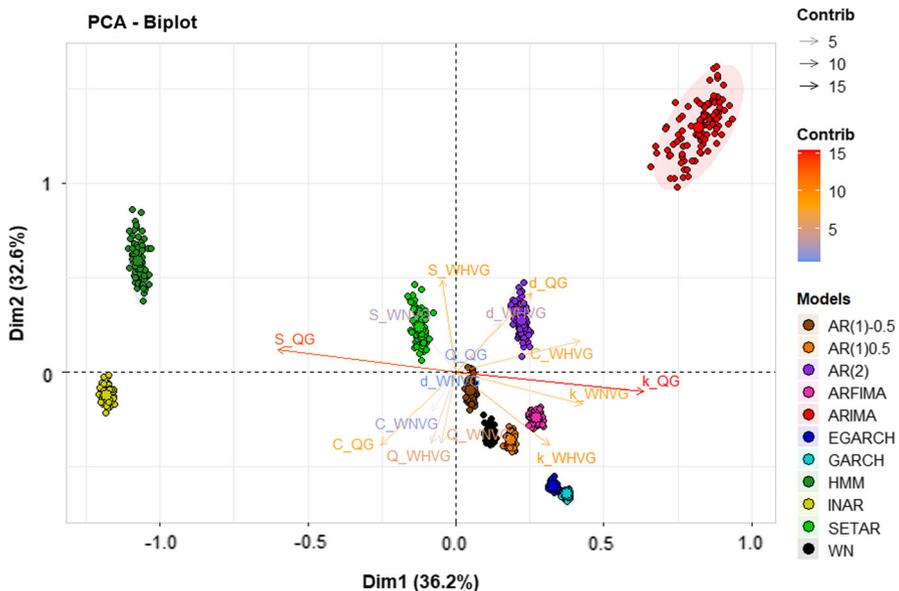
Fig. 6 Bi-plot of the first two PC's for the synthetic data set. Each Data Generating Process (DGP) is represented by a color and the arrows represent the contribution of the corresponding feature to the PC's: the larger the size, the sharper the color and the closer to the red the greater the contribution of the feature. Features grouped together are positively correlated while those placed on opposite quadrants are negatively correlated

weighted degree, $\bar{k}$, of *NetF*. The average path length, $\bar{d}$, of WHVGs and QGs and the clustering coefficient, C, of WHVG are positively correlated, but negatively to the $\bar{d}$, C and Q of WNVG, Q of WHVG and C of QGs. The features that most contribute to the total dimensions formed by the PCA are: $\bar{k}$, S, Q and $\bar{d}$ of the QGs, $\bar{k}$ of the WNVGs, and $\bar{k}$, S and Q of the WHVGs (see Fig. 10).

The (stochastic) trend of the `ARIMA`, in fact the only non-stationary DGP this data set, is represented by high average path lengths, $\bar{d}$, in WHVG and QG. Discrete states in the data, `HMM`, `SETAR`, `INAR`, are associated with the number of communities, S. The bi-plot further indicates that height average weighted degree, $\bar{k}$, mainly that of the WHVG, represents heteroskedasticity in the time series, e.g., `GARCH` and `EGARCH`. Cycles, `AR(2)`, are captured by the clustering coefficient, C.

We also did an empirical study of the *NetF* features in the context of clustering these synthetic data sets, and the results show that using the entire feature set leads to better performance than any possible subset. This showcases how the different features complement each other and how they capture different characteristics of the underlying time series. The details of this study can be seen in Appendix C.

4 Mining time series with *NetF*

In this section we illustrate the usefulness of complex networks based time series features in data mining tasks with a case study regarding time series clustering via feature-based approach (Maharaj et al. 2019). Within this mining task we analyse the synthetic data set introduced in Sect. 3.3, benchmark empirical data sets from UEA & UCR Time Series Classification Repository (Bagnall et al.), the M3 competition data from package `Mcomp` (Hyndman 2018), the set "18Pairs" from package `TSclust` (Montero and Vilar 2014) and a new data set regarding the production of several crops across Brazil (Instituto Brasileiro de Geografia e Estatística) using *NetF* and two other sets of time series features, namely *catch22* (Lubba et al. 2019) and *tsfeatures* (Wang et al. 2006), see Appendix D.

4.1 Clustering methodology

The overall procedure proposed here for feature-based clustering is represented in Fig. 7.

Given a set of time series, compute the feature vectors which are then Min-Max rescaled into the [0, 1] interval and organized in a feature data matrix. Principal Components (PC) are computed (no need of z-score normalization within PCA) and finally a clustering algorithm is applied to the PC's. Among several algorithms available for clustering analysis, we opt for *k-means* (Hartigan and Wong 1979) since it is fast and widely used for clustering. Its main disadvantage is the need to pre-define the number of clusters. This issue will be discussed within each data set example. The clustering results are assessed using appropriate evaluation metrics: *Average Silhouette* (AS); *Adjusted Rand Index* (ARI) and *Normalized Mutual Information* (NMI) when the ground truth is available.
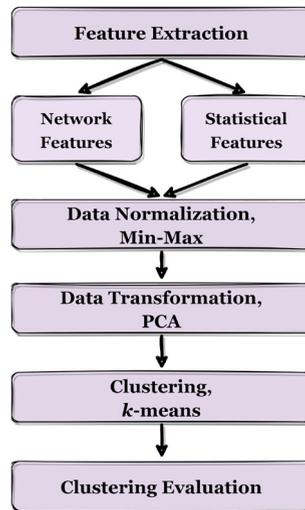
Fig. 7 Schematic diagram for the time series clustering analysis procedure

4.2 Data sets and experimental setup

We report the detailed results for the clustering exercise for the eleven data sets summarized in Table 2. The brief description of the data and clustering results for the remaining data sets is presented in Table 10.

The data sets in Table 2 belong to the UEA & UCR Time Series Classification Repository (Bagnall et al.), widely used in classification tasks, the M3 competition data from package Mcomp (Hyndman 2018) used for testing the performance of forecasting algorithms, the set "18Pairs", extracted from package TSclust (Montero and Vilar 2014) which represents pairs of time series of different domains. For all these we have true clusters and therefore clustering assessment measures ARI and NMI may be used. Additionally, we also analyse a set of observations comprising the production over forty three years of nine agriculture products in 108 meso-regions of Brazil (Instituto Brasileiro de Geografia e Estatística). We note that the size of the ElectricDevices dataset, 16575 time series, is different from the total available in the repository, as exactly 62 time series return missing values for the entropy feature of the *tsfeatures* set (see Appendix D) and so we decided to exclude these series from our analysis.

4.3 Results

First, we investigate the performance of *NetF*, *catch22* and *tsfeatures* in the automatic determination of the number of clusters k, using the clustering evaluation metrics, ARI, NMI and AS. The results (see Table 9) show overall similar values but we note that *NetF* seems to provide a value of k equal to or closer to the ground truth value (when available) more often. For the Production in Brazil data, for which there is no ground truth, values for k are obtained averaging 10 repetitions of the clustering procedure

Table 2 Brief description of the empirical time series data sets

Data set	Size of data set	T	Num. of classes	Source
18Pairs	36	1000	18	Montero and Vilar (2014)
M3 data	3003	[20, 144]	6	Hyndman (2018)
CinC_ECG_torso	1420	1639	4	Bagnall et al.
Cricket_X	780	300	12	Bagnall et al.
ECG5000	5000	140	5	Bagnall et al.
ElectricDevices	16575	96	7	Bagnall et al.
FaceAll	2250	131	14	Bagnall et al.
FordA	4921	500	2	Bagnall et al.
InsectWingbeatSound	2200	256	11	Bagnall et al.
UWaveGestureLibraryAll	4478	945	8	Bagnall et al.
Production in Brazil	108	198	9	Instituto Brasileiro de Geografia e Estatística

and using the silhouette method. The results of the 10 repetitions are represented in Fig. 12 and summarized in Table 4.

Next, fixing k to the ground truth we perform the clustering procedure. The clustering evaluation metrics, mean over 10 repetitions, are presented in Table 3.[7]

The results indicate that none of the three approaches performs uniformly better than the others. Some interesting comments follow. For the synthetic data sets and 18Pairs, *tsfeatures* and *NetF* perform better than *catch22* in all evaluation criteria. The clusters for ECG5000, ElectricDevices and UWaveGestureLibraryAll data sets produced by the three approaches fare equally well when assessed by ARI, NMI and AS. The same is true for M3 data and Cricket_X data sets, with slightly lower results. *NetF* approach seems produce better clusters for CinC_ECG_torso measured according to the three criteria, the *tsfeatures* seems to produce better clusters for FordA, and the *catch22* for FaceAll and InsectWingbeat measured according to the ARI and NMI.

Analyzing the overall results, Tables 3 and 10, we can state that *tsfeatures* and *NetF* approaches present the best ARI and NMI evaluation metrics, while *tsfeatures* achieves by far the best results in the AS. If we consider the UEA & UCR repository classification of the data sets, we note the following: the *NetF* approach presents good results for time series data of the type Image (BeetleFly, FaceFour, MixedShapes-RegularTrain, OSULeaf and Symbols), ECG (CinC_ECG_tors and TwoLeadECG) and Sensor (Wafer); the *tsfeatures* performs best for types Simulated (BME, UMD and TwoPatterns), ECG (NonInvasiveFetalECGTho), Image (ShapesAll) and Sensor (SonyAIBORobotSurface and Trace); finally the *catch22* approach presents best results for Spectro (Coffee), Device (HouseTwenty) and Simulated (ShapeletSim)

---

[7] The results for the remaining empirical time series data sets of the UEA & UCR Time Series Classification Repository are presented in Table 10.

Table 3 Clustering evaluation metrics obtained for the three approaches *NetF*, *tsfeatures* and *catch22*

Data set	ARI			NMI			AS		
	[−1, 1]			[0, 1]			[−1, 1]		
	tsf.	*cat.*	*NetF*	*tsf.*	*cat.*	*NetF*	*tsf.*	*cat.*	*NetF*
18Pairs	**0.51**	0.39	0.49	**0.89**	0.86	**0.89**	**0.42**	0.32	0.34
M3 data	**0.14**	0.13	0.13	**0.21**	0.19	0.18	**0.36**	0.22	0.31
CinC_ECG_tors	0.31	0.32	**0.45**	0.37	0.35	**0.52**	0.23	0.19	**0.31**
Cricket_X	0.15	0.15	**0.16**	**0.32**	0.28	0.30	**0.20**	0.16	0.10
ECG5000	0.29	0.28	**0.31**	**0.32**	0.29	0.30	**0.24**	**0.24**	0.16
ElectricDevices	0.20	**0.21**	0.19	**0.30**	0.29	0.29	**0.33**	0.25	0.27
FaceAll	0.15	**0.21**	0.15	0.33	**0.36**	0.29	**0.22**	0.15	0.09
FordA	**0.19**	0.01	0.01	**0.27**	0.01	0.01	**0.53**	0.33	0.29
InsectWingbeat	0.07	**0.21**	0.17	0.18	**0.37**	0.32	**0.19**	0.18	0.11
UWaveGesture	0.17	**0.20**	0.18	0.27	**0.28**	**0.28**	**0.20**	0.19	0.12
Synthetic (DGP)	0.76	0.40	**0.92**	0.91	0.66	**0.97**	**0.73**	0.39	0.68
Production Brazil	0.09	0.18	**0.30**	0.40	0.55	**0.70**	0.46	0.39	**0.61**

The values reflect the mean of 10 repetitions of the clustering analysis with number of classes equal to the ground truth (see Table 2). The values in bold represent the best results

Table 4 Clustering evaluation metrics for the different clustering analysis on Production in Brazil data set based on *NetF*, *tsfeatures* and *catch22* approaches

Approach	*k*	ARI [−1, 1]	NMI [0, 1]	AS [−1, 1]
NetF	4	**0.30**	**0.70**	**0.61**
catch22	3	0.18	0.55	0.39
tsfeatures	2	0.09	0.40	0.46

The values reflect the mean of 10 repetitions of the proposed method for different feature vectors and for the number of clusters detected according to the average silhouette metric

types. In summary *NetF* and *tsfeatures* perform better in data with the same characteristics while *catch22* seems to be more appropriate to capture other characteristics.

Regarding the data set Production in Brazil, Table 4 shows more detail on the clustering results, adding the value *k* to indicate the number of clusters that was automatically computed. We note that the 4 clusters obtained with *NetF* correspond to 4 types of goods: eggs; energy; gasoline and cattle; hypermarkets, textile, furniture, vehicles and food. Attribution plots of the clusters obtained by the three approaches are represented in Fig. 8. Note that both *tsfeature* and *catch22* put eggs and textile production in the same cluster, and *tsfeature* cannot distinguish energy. Notice also how the *NetF* approach produced the cluster with highest AS and hence the highest intra-cluster-similarity. To illustrate the relevance of the results, Fig. 9 depicts a representative time series for each cluster.
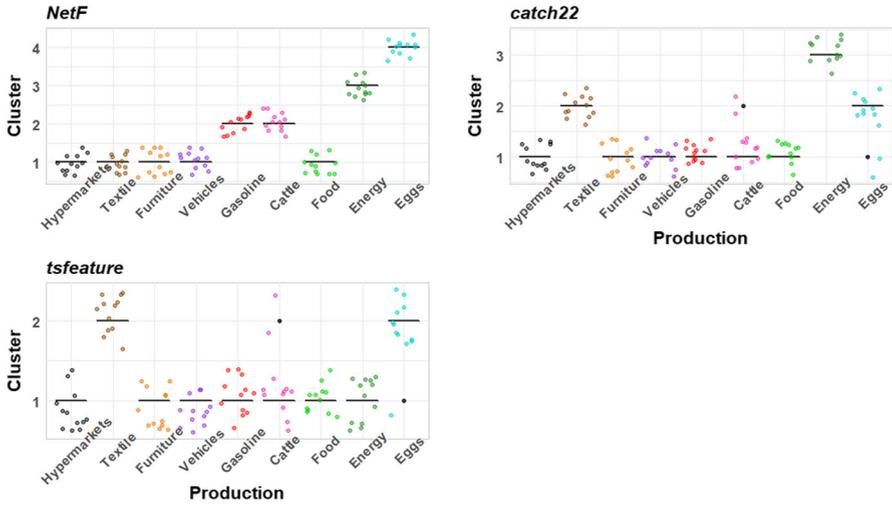
Fig. 8 Attribution of the Production in Brazil time series to the different clusters, according to each of the feature approaches. The different productions are represented on the horizontal axis and by a unique color. The time series are represented by the colored points according to its production type. The vertical axis represents the cluster number to which a time series is assigned
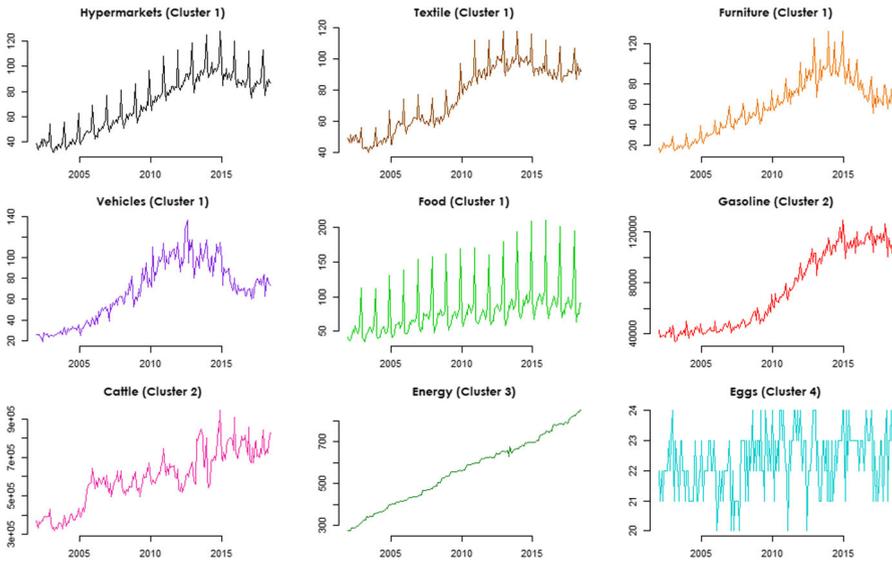
Fig. 9 Production in Brazil representative of each cluster (indicated in subtitle) obtained using the proposed approach, *NetF*

5 Conclusions

In this paper we introduce *NetF*, a novel set of 15 time series features, and we explore its ability to characterize time series data. Our methodology relies on mapping the time series into complex networks using three different mapping methods: natural and horizontal visibility and quantile graphs (based on transition probabilities). We then extract five topological measures for each mapped network, concatenating them into a single time series feature vector, and we describe in detail how we can do this in practice.

To better understand the potential of our approach, we first perform an empirical evaluation on a synthetic data set of 3300 networks, grouped in 11 different and specific time series models. Analysing the weighted visibility (natural and horizontal) and quantile graphs feature space provided by *NetF*, we were able to identify sets of features that distinguish non-stationary from stationary time series, counting from real-valued time series, periodic from non-periodic time series, state time series from non-state time series and heteroskedastic time series. The non-stationarity time series have high values of average path length and low values of clustering coefficients in their QGs, and the opposite happens for the stationary time series. Counting series have lowest value of average weighted degree, highest value of number of communities in their QGs and lowest value of clustering coefficient in WHVGs, while the opposite happens for non-counting time series. For state time series the average weighted degree value in their weighted VGs is the lowest and the number of communities is high, the opposite happens for the non-state time series. Heteroskedastic time series are identified with high average weighted degree values of their WHVGs, compared to the other DGP's.

To further showcase the applicability of *NetF*, we use its feature set for clustering both the previously mentioned synthetic data, as well as a large set of benchmark empirical time series data sets. The results for the data sets in which ground truth is available indicate that *NetF* yields the highest mean for ARI (0.287) compared to alternative time series features, namely *tsfeature* and *catch22*, with means of 0.267 and 0.228, respectively. For the NMI metric the results are similar (0.395, 0.397 and 0.358, respectively) and for AS the highest mean was found for *tsfeature*, 0.332 versus approximately 0.3 for the others. However, the higher values for AS[8] must be viewed in light of the low values of ARI and NMI which indicate an imperfect formation of the clusters. For the production data in Brazil, for which no ground truth is available, *NetF* produces clusters which group production series with different characteristics, namely, time series of counts, marked upward trend, series in the same range of values, and with seasonal component.

The results show that *NetF* is capable of capturing information regarding specific properties of time series data. *NetF* is also capable of grouping time series of different domains, such as data from ECGs, image and sensors, as well as identifying different characteristics of the time series using different mapping concepts, which stand out in different topological features. The general characteristics of the data, namely, the size of the data set, the length of the time series and the number of clusters, do not seem

---

[8] Samples are very similar inside the cluster and show little similarity inter-cluster.

to be influencing the results obtained. Also, *NetF* does not require typical time series preprocessing tasks, such as decomposing, differencing or whitening. Moreover, our methodology is applicable to any time series, regardless of the nature of the data.

The mappings and topological network measures considered are global, but it is important to clarify that they do not constitute a "universal" solution. In particular, we found that the weighted versions of the visibility graph mappings used here produce better results than their unweighted versions, as we can see in previous works (Silva 2018). In fact, formulating a set of general features capable of fully characterizing a time series without knowing both the time series properties and the intended analysis is a difficult and challenging task (Kang et al. 2020).

For related future work, we intend to add and explore new sets of topological measures both, adding local and intermediate features to *NetF*, as well as exploring other mapping methods (such as proximity graphs). We also intend to extend our approach to the multivariate case, since obtaining useful features for multidimensional time series analysis is still an open problem.

Data Availability *NetF* and data sets are available at https://github.com/vanessa-silva/NetF.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Appendices

Appendix A: Time series models

Main reference (Shumway and Stoffer 2017)

Linear models

WN The *white noise* process, ϵ_t, is a sequence of i.i.d. random variables with mean 0 and constant variance σ_ϵ^2. It is the simplest time series process that reflects information that is neither directly observable nor predictable. We generated $\epsilon_t \sim N(0, 1)$ white noise processes and refer to them as `WN`.

AR(p) We defined a process Y_t as an AR process of order p if it satisfies the following equation:

$$Y_t = \sum_{i=1}^{p} \phi_i Y_{t-i} + \epsilon_t, \tag{3}$$

where ϵ_t is the white noise and ϕ_i is the autoregressive constant. We used $p \in \{1, 2\}$, and parameters $\phi_1 \in \{-0.5, 0.5\}$ to generate AR(1) processes and $\phi_1 = 1.5$ and $\phi_2 = -0.75$ for AR(2) processes. These parameters ensure that the time series present the following characteristics: $\phi_1 = 0.5$ leads to smoother time series than $\phi_1 = -0.5$; and $\phi_1 = 1.5$ and $\phi_2 = -0.75$ generates pseudo-periodic time series. We refer to the three models generated as `AR(1)-0.5`, `AR(1)0.5` and `AR(2)`, respectively.

ARIMA(p,d,q) The *autoregressive integrated moving average* model is a generalization of the ARMA model suitable for modeling non-stationary time series. A process Y_t is an ARMA(p, q) process if it satisfies the equation:

$$Y_t = \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t, \tag{4}$$

where θ_i is the moving average constant. If a process Y_t is a non-stationary time series it can be written as an ARIMA(p, d, q) process if its dth-differences $\nabla^d Y_t = (1 - B)^d Y_t$, $d \in \mathbb{N}$, is a stationary ARMA(p, q) process. So Y_t satisfies the following equation,

$$\left(1 - \sum_{i=1}^{p} \phi_i B^i\right)(1 - B)^d Y_t = \left(1 + \sum_{i=1}^{q} \theta_i B^i\right)\epsilon_t, \tag{5}$$

where B represents the backshift operator, $BY_t = Y_{t-1}$. We use $p = 1$, $d = 1$, $q = 0$, and $\phi_1 = 0.7$ to generate ARIMA(1, 1, 0) processes with stochastic trend. We refer to these processes as `ARIMA`.

ARFIMA(p, d, q) *Autoregressive fractionally integrated moving average* model is a generalization of the ARIMA model useful for modeling time series with long range dependence. A process Y_t is an ARFIMA(p, d, q) process if it satisfies the (Eq 5) and the difference parameter, d, can take real values. We generate ARFIMA(1, 0.4, 0) processes that exhibit long memory and refer to them as `ARFIMA`.

Time series are generated from the above DGP using the R packages: `timeSeries` (Wuertz et al. 2017) and `fracdiff` (Maechler et al. 2020).

Non linear models

SETAR(1) The *self-exciting threshold autoregressive* model of order 1 specify the nonlinearity in the conditional mean. It is useful for processes with regime changes that approximate a nonlinear function by piece wise linear functions dependent on the regime (Tong 2011). This model can be presented by the following system of equations,

$$Y_t = \begin{cases} \alpha Y_{t-1} + \epsilon_t, & \text{if } Y_{t-1} \leq r \\ \beta Y_{t-1} + \gamma \epsilon_t, & \text{if } Y_{t-1} > r \end{cases}, \tag{6}$$

where r represents a real threshold. We used $\alpha = 0.5$, $\beta = -1.8$, $\gamma = 2$ and $r = -1$ and we generated time series with regimes with different autocorrelation properties: in the first the correlation is positive while in the second alternates between positive and negative values. We refer to this model as `SETAR`.

HMM *Hidden Markov models* are probabilistic models for the joint probability the random variables $(Y_1, \ldots, Y_T, X_1, \ldots, X_T)$ where Y_t is a discrete (or continuous) variable and X_t is a *hidden Markov chain* with a finite number of states, N. The following conditional independence assumptions hold (Zucchini et al. 2016):

1. $P(X_t \mid X_{t-1}, Y_{t-1}, \ldots, X_1, Y_1) = P(X_t \mid X_{t-1})$,
2. $P(Y_t \mid X_T, Y_T, \ldots, X_1, Y_1) = P(Y_t \mid X_t)$.

We used $N = 2$ and the transition matrix: $\begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$. The data are generated from a Poisson distribution with $\lambda = 10$ for the first regime and $\lambda = 15$ for the second. We refer to this model as `HMM`. Time seriea are generated using the R package `HMMpa` (Witowski and Foraita 2014).

The next nonlinear models are based on ARCH models where the mean-corrected asset return is serially uncorrelated but dependent and the dependency can be described by a simple quadratic function of its lagged values (Tsay 2010). Hereafter, ϵ_t are uncorrelated random variables, z_t represents a white noise with variance 1 and σ_t the standard deviation of ϵ_t, that is $\epsilon_t = \sigma_t z_t$.

GARCH(p, q) The GARCH model is a *generalization* of the ARCH model in which the conditional volatility is a function not only of the squares of past innovations, but also of their own past values (Cryer and Chan 2008). Thus, ϵ_t is a GARCH(p, q) process if it satisfies the following equation,

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2, \tag{7}$$

where $\omega > 0$, $\alpha_i, \beta_i \geq 0$, $\sum_{i=1}^{p} \beta_i + \sum_{i=1}^{q} \alpha_i < 1$. The conditional standard deviation can exhibit persistent periods of high or low volatility because past values of the process are fed back into the present value. We used $p = 1$, $q = 1$, $\omega = 10^{-6}$, $\alpha_1 = 0.1$ and $\beta_1 = 0.8$ to generate the GARCH(1, 1) processes and we refer to them as `GARCH`.

EGARCH(p, q) The *exponential* GARCH allows asymmetric effects of positive and negative shocks on volatility (Tsay 2010). The EGARCH(p, q) model is given

by the equation,

$$log(\sigma_t^2) = \omega + \sum_{i=1}^{q} \alpha_i \left| \frac{\epsilon_{t-i}}{\sigma_{t-i}} \right| + \sum_{i=1}^{p} \beta_i log(\sigma_{t-i}^2) + \sum_{i=1}^{q} \gamma_i \frac{\epsilon_{t-i}}{\sigma_{t-i}}, \qquad (8)$$

where $\omega = \alpha_0 - \alpha_1 \sqrt{\frac{2}{\pi}}$, α_i characterize the volatility clustering phenomena, β_i is the persistence parameter, and γ_i describes the leverage effect. The logged conditional variance allows to relax the positivity constraint of the model coefficients. To this model we choose $p = 1$, $q = 1$, $\omega = \left(10^{-6} - 0.1 \sqrt{\frac{2}{\pi}} \right)$, $\alpha_1 = 0.1$, $\beta_1 = 0.01$ and $\gamma_1 = 0.3$, and we refer to it as EGARCH.

INAR(1) The *integer-valued autoregressive* models have been proposed to model integer-valued time series, in particular, correlated counts (Silva and Oliveira 2004). These models are based on thinning (random) operations defined on the integers, where the following binomial thinning is the most common: let X be a non-negative integer valued random variable and $0 < \alpha < 1$, then $\alpha * X = \sum_{i=1}^{X} Y_i$ where $\{Y_i\}$ is a sequence of i.i.d. Bernoulli random variables, independent of X. A process Y_t is said to be an INAR(1) process if it satisfies the equation,

$$Y_t = \alpha * Y_{t-1} + \epsilon_t. \qquad (9)$$

If the innovation sequence ϵ_t and the initial distribution are Poisson, Y_t is said to be a Poisson INAR(1) process. We used $\alpha = 0.5$ and Poisson arrivals with $\epsilon_t \sim Po(1)$ to generate integer valued data with autocorrelation decaying at a rate of 0.5. We refer to this model as INAR.

Time series from the HMM and GARCH are simulated using the R packages HMMpa (Witowski and Foraita 2014) and fGarch (Wuertz et al. 2017), respectively. Time series are generated from the remaining DGP of our own implementation, available from the authors.

Appendix B: Feature evaluation in synthetic time series

The topological features of WNVGs, WHVGs and QGs obtained from the 1100 time series models are, respectively, summarized in Tables 5, 6 and 7. The table reporting the mean and standard deviation (in brackets) of the *Min-Max* normalized (across models) metrics. The columns of the tables are colored with a gradient based on the mean values: cells with a maximum value 1 are colored red, cells with the minimum value 0 are colored white and the remainder with a hue of red color proportional to its value.

Weighted natural visibility graphs

See Table 5.

Table 5 Table of mean values of the 100 instances of each DGP for each topological metric, resulting from WNVGs

Models	Average Degree $(\bar{k})$	Average Path Length $(\bar{d})$	Number of Communities (S)	Clustering Coefficient (C)	Modularity (Q)
AR(1)-0.5	0.430 (0.006)	0.457 (0.032)	0.225 (0.063)	0.615 (0.007)	0.900 (0.022)
AR(1)0.5	0.698 (0.005)	0.443 (0.037)	0.116 (0.037)	0.751 (0.009)	0.963 (0.010)
AR(2)	0.719 (0.008)	0.408 (0.035)	0.472 (0.132)	0.968 (0.012)	0.792 (0.082)
ARIMA	0.919 (0.006)	0.211 (0.095)	0.257 (0.115)	0.188 (0.079)	0.367 (0.140)
ARFIMA	0.790 (0.005)	0.412 (0.036)	0.099 (0.035)	0.766 (0.012)	0.973 (0.009)
SETAR	0.273 (0.006)	0.438 (0.041)	0.491 (0.145)	0.631 (0.007)	0.772 (0.070)
HMM	0.012 (0.005)	0.446 (0.039)	0.570 (0.150)	0.667 (0.008)	0.654 (0.093)
INAR	0.570 (0.004)	0.452 (0.118)	0.131 (0.052)	0.631 (0.011)	0.956 (0.024)
GARCH	0.994 (0.003)	0.433 (0.036)	0.070 (0.022)	0.652 (0.010)	0.991 (0.005)
EGARCH	0.940 (0.004)	0.425 (0.032)	0.066 (0.027)	0.677 (0.007)	0.980 (0.004)
WN	0.581 (0.005)	0.450 (0.034)	0.128 (0.044)	0.667 (0.007)	0.942 (0.009)

The standard deviations are presented in parentheses

Weighted horizontal visibility graphs

See Table 6.

Table 6 Table of mean values of the 100 instances of each DGP for each topological metric, resulting from WHVGs

Models	Average Degree ($\bar{k}$)	Average Path Length ($\bar{d}$)	Number of Communities (S)	Clustering Coefficient (C)	Modularity (Q)
AR(1)-0.5	0.473 (0.005)	0.004 (0.001)	0.230 (0.057)	0.557 (0.004)	0.319 (0.056)
AR(1)0.5	0.628 (0.003)	0.009 (0.001)	0.174 (0.046)	0.697 (0.004)	0.793 (0.031)
AR(2)	0.438 (0.006)	0.022 (0.002)	0.600 (0.092)	0.953 (0.004)	0.397 (0.074)
ARIMA	0.414 (0.007)	0.588 (0.161)	0.794 (0.106)	0.988 (0.005)	0.236 (0.074)
ARFIMA	0.633 (0.003)	0.030 (0.004)	0.204 (0.047)	0.785 (0.004)	0.878 (0.028)
SETAR	0.284 (0.005)	0.003 (0.001)	0.508 (0.086)	0.504 (0.004)	0.354 (0.092)
HMM	0.009 (0.003)	0.013 (0.002)	0.640 (0.116)	0.467 (0.006)	0.385 (0.105)
INAR	0.403 (0.002)	0.034 (0.005)	0.292 (0.056)	0.034 (0.010)	0.927 (0.033)
GARCH	0.998 (0.001)	0.007 (0.002)	0.064 (0.022)	0.611 (0.004)	0.777 (0.015)
EGARCH	0.900 (0.001)	0.004 (0.001)	0.059 (0.026)	0.591 (0.005)	0.747 (0.017)
WN	0.584 (0.004)	0.004 (0.001)	0.170 (0.045)	0.605 (0.004)	0.624 (0.033)

The standard deviations are presented in parentheses

Quantile graphs

See Table 7.

Table 7 Table of mean values of the 100 instances of each DGP for each topological metric, resulting from QGs

Models	Average Degree ($\bar{k}$)	Average Path Length ($\bar{d}$)	Number of Communities (S)	Clustering Coefficient (C)	Modularity (Q)
AR(1)-0.5	1.000 (0.000)	0.005 (0.000)	0.000 (0.000)	0.972 (0.003)	0.008 (0.003)
AR(1)0.5	1.000 (0.000)	0.005 (0.000)	0.027 (0.010)	0.971 (0.003)	0.374 (0.044)
AR(2)	1.000 (0.000)	0.024 (0.000)	0.044 (0.015)	0.829 (0.003)	0.816 (0.028)
ARIMA	1.000 (0.000)	0.943 (0.068)	0.062 (0.022)	0.144 (0.132)	0.398 (0.099)
ARFIMA	1.000 (0.000)	0.029 (0.003)	0.047 (0.017)	0.808 (0.009)	0.890 (0.039)
SETAR	1.000 (0.000)	0.016 (0.000)	0.027 (0.009)	0.946 (0.003)	0.245 (0.039)
HMM	0.276 (0.008)	0.001 (0.000)	0.730 (0.008)	0.998 (0.003)	0.289 (0.027)
INAR	0.000 (0.002)	0.002 (0.001)	0.981 (0.009)	0.984 (0.024)	0.493 (0.010)
GARCH	1.000 (0.000)	0.001 (0.000)	0.031 (0.012)	1.000 (0.001)	0.055 (0.016)
EGARCH	1.000 (0.000)	0.002 (0.000)	0.019 (0.010)	0.999 (0.001)	0.041 (0.016)
WN	1.000 (0.000)	0.001 (0.000)	0.029 (0.011)	1.000 (0.000)	0.047 (0.011)

The standard deviations are presented in parentheses

Principal component analysis results

See Fig. 10.
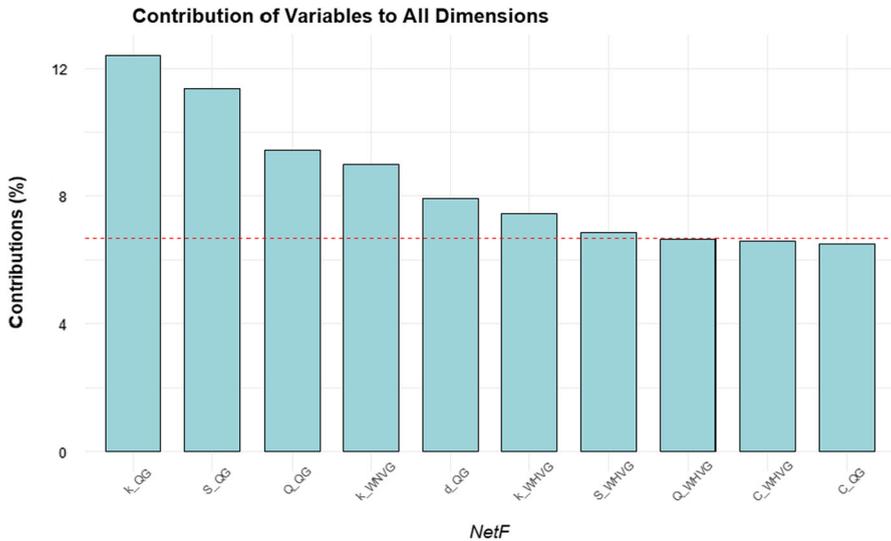
Fig. 10 Bar plot with contributions of *NetF* features to the total of all 15 principal components formed by the PCA. The red dashed line on the plot indicates the expected average contribution

Appendix C: Clustering of time series models

We analyse the performance of different combinations of the feature vectors from the WNVG, WHVG and QG mappings in a clustering task using the synthetic data set. We set the number of clusters to $k = 11$, the total of time series models, and assess the clustering results with the evaluation metrics. The results summarized in Table 8[9] indicate that joining the features obtained from the two mapping concepts (VGs and QGs) adds information that leads to improvements in the clustering results (compare the first three rows of the Table 8 with the last three). In fact, as illustrated in Fig. 11, clustering based on *NetF* can leads to a perfect attribution of the time series models samples across the 11 different clusters.

These results show that different mapping methods capture different properties from the series, as we analyzed in the Sect. 3.3, translating into a better clustering result, as we expected. If we analyse only feature vectors corresponding to one network kind, the first three rows of Table 8, we note that the WHVGs are the ones that best capture the characteristics of time series models, having high evaluation values, namely, 0.83

---

[9] The results are means from 10 repetitions of the clustering analysis. The corresponding standard deviations indicate little or none variation between the repetitions.

Table 8 Clustering evaluation metrics for the different clustering analysis resulting from different network-based feature vectors

Mappings	ARI [−1, 1]	NMI [0, 1]	AS [−1, 1]
WNVG	0.68	0.86	0.51
WHVG	0.83	0.94	0.63
QG	0.64	0.84	0.66
WNVG–WHVG	0.81	0.93	0.57
WNVG–QG	**0.84**	**0.94**	**0.67**
WHVG–QG	*0.90*	*0.96*	*0.73*
NetF	<u>0.92</u>	<u>0.97</u>	<u>0.68</u>

The values reflect the mean of 10 repetitions of the proposed method for different feature vectors and for the ground truth $(k = 11)$. The highest values are highlighted (bold, italic and underline)
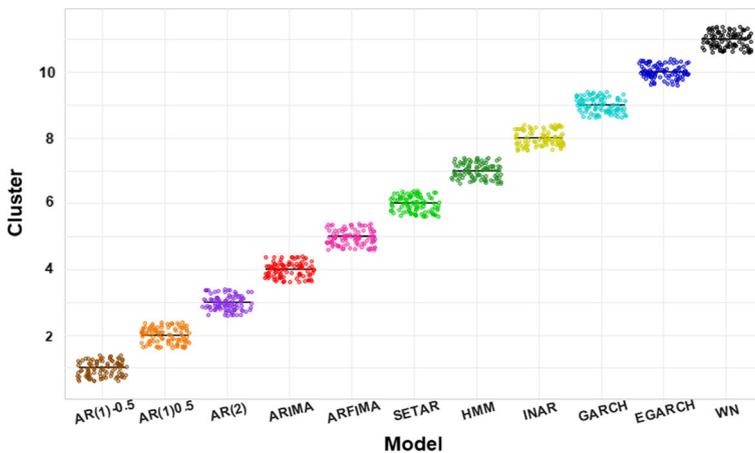
Fig. 11 Attribution of the samples corresponding to instances of time series models to the different clusters, according to *NetF*. The different models are represented on the horizontal axis and by a unique color. The time series are represented by the colored points according to its model process. The vertical axis represents the cluster number to which a time series is assigned

for ARI, 0.94 for NMI and 0.63 for AS. The last three lines of Table 8 show better results than those obtained using only WHVG features, thus showing that the resulting features of the QGs add information about certain properties of the time series models.

We still study how these seven sets of features perform in determining the number of clusters $k = 11$ using the ARI, NMI and AS evaluation metrics. The results for k obtained from features corresponding to only one kind of network, range from 8 to 13 for ARI, from 11 to 14 for NMI and 3 to 9 for AS. However, when *NetF* is used, we obtain $k = 11$ for ARI and NMI and $k = 10$ for AS (see Fig. 11).

Appendix D: Classical features

The above described procedure is applied to two further sets of features previously proposed in the literature. One is a set of time series statistical features that has been used in a variety of tasks such as clustering (Wang et al. 2006), forecasting (Kang et al. 2017; Talagala et al. 2018) and generation of time series data (Kang et al. 2020). It comprises sixteen measures calculated using the `tsfeatures` package (Hyndman et al. 2020) of the R CRAN (R Core Team 2020), namely, frequency and number and length of seasonal periods, strength of trend, "spikiness" of a time series, linearity and curvature, spectral entropy, and measures based on autocorrelation coefficients of the original series, first-differenced series and second-differenced series. These will be denoted by *tsfeatures* in the remainder of this work. The second is denominated canonical feature set, *catch22* (Lubba et al. 2019), Fulcher and Jones (2017) has been recently proposed based on a features library from an interdisciplinary time series analysis literature and Lubba et al. (2019) has been used in time series classification tasks. There are twenty two measures calculated using the `Rcatch22`[10] package (Henderson 2021) of the R CRAN (R Core Team 2020), that include properties of the distributions and simple temporal statistics of values in the time series, linear and non-linear autocorrelation, successive differences, scaling of fluctuations, and others.

Appendix E: Clustering time series with *NetF*

See Fig. 12 and Table 9.
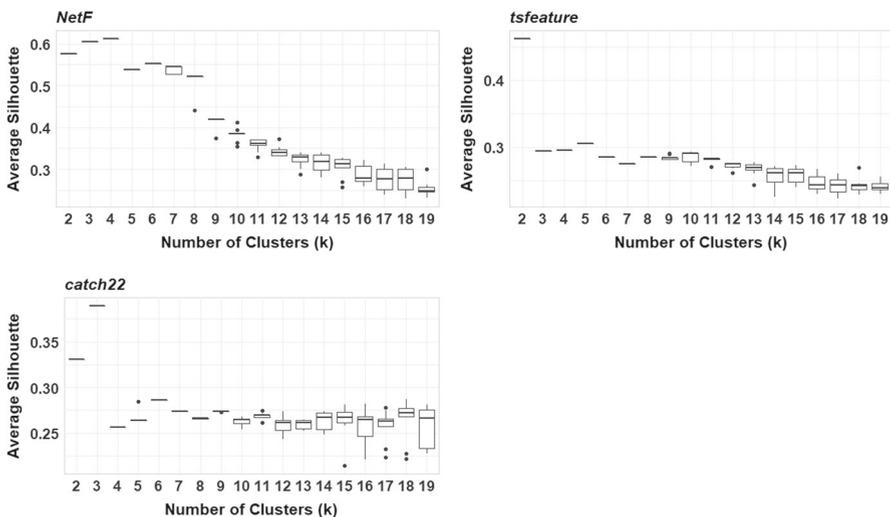
Fig. 12 Number of clusters, k, for the Production in Brazil data set using the silhouette method for 10 repetitions of the clustering analysis using the 3 features vectors: *NetF*, *catch22* and *tsfeature*

---

[10] https://github.com/hendersontrent/Rcatch22.

Table 9 Clustering evaluation metrics obtained for the two conventional approaches (*tsf.* and *cat.*) and for the proposed approach (*NetF*)

Data set	Num. of Classes	Best k			ARI [−1, 1]			NMI [0, 1]			AS [−1, 1]		
		tsf.	cat.	NetF	tsf.	cat.	NetF	tsf.	cat.	NetF	tsf.	cat.	NetF
18Pairs	18	21	19	21	**0.51**	0.41	0.39	**0.90**	0.87	0.87	**0.40**	0.37	0.28
M3 data	6	9	9	5	0.14	0.14	0.14	**0.22**	0.21	0.18	**0.31**	0.26	0.26
CinC_ECG_torso	4	4	4	5	0.31	0.32	**0.45**	0.37	0.35	**0.53**	0.23	0.19	**0.31**
Cricket_X	12	9	17	10	0.17	**0.16**	**0.16**	**0.31**	0.30	0.29	**0.20**	0.15	0.10
ECG5000	5	4	2	2	0.35	**0.45**	**0.45**	**0.37**	0.35	0.32	0.25	**0.33**	0.19
ElectricDevices	7	9	3	5	0.21	0.26	**0.28**	0.30	**0.35**	0.32	**0.31**	0.28	0.30
FaceAll	14	23	14	11	0.16	**0.20**	0.15	**0.37**	0.36	0.27	**0.20**	0.15	0.09
FordA	2	4	6	3	**0.21**	0.12	0.13	**0.30**	0.21	0.13	**0.31**	0.14	0.21
InsectWingbeatSound	11	6	11	3	0.09	**0.21**	0.20	0.19	0.36	**0.39**	**0.21**	0.18	0.17
UWaveGestureLibraryAll	8	16	10	6	0.21	0.21	0.21	**0.36**	0.30	0.30	0.19	**0.20**	0.15
Synthetic (DGP)	11	15	8	11	0.79	0.47	**0.92**	0.91	0.69	**0.97**	0.64	0.38	**0.68**

The number of clusters, k, is determined automatically by the evaluation metrics

Appendix F: Clustering results: UEA & UCR time series datasets

For some sets of benchmark empirical time series, some features of *tsfeatures* and *catch22* approaches (see Appendix D) return missing values, and some have time series with missing values. We decided not to consider these sets in our clustering analysis as they are just a few. So Table 10 present the results for 119 sets, out of a total of 129.

Table 10 Brief description of the empirical time series datasets from UEA & UCR time series repository (Bagnall et al.) and the clustering evaluation metrics obtained for the two conventional approaches (*tsfeatures* and *catch22*) and for the proposed approach (*NetF*)

Data set	M	T	k	ARI [−1, 1]			NMI [0, 1]			AS [−1, 1]		
				tsf.	cat.	Net.	tsf.	cat.	Net.	tsf.	cat.	Net.
ACSF1	200	1460	10	0.22	**0.32**	0.17	0.50	**0.56**	0.40	**0.48**	**0.30**	0.23
Adiac	781	176	37	**0.21**	**0.21**	0.11	**0.55**	**0.55**	0.43	0.20	**0.26**	0.13
ArrowHead	211	251	3	0.24	0.17	**0.34**	0.27	0.21	**0.31**	0.24	**0.27**	0.15
BME	180	128	3	**0.50**	0.36	0.40	**0.58**	0.36	0.45	**0.55**	0.20	0.23
Beef	60	470	5	**0.09**	0.05	0.04	**0.23**	0.24	0.18	0.33	**0.38**	0.18
BeetleFly	40	512	2	0.55	0.10	**0.63**	0.48	0.11	**0.55**	**0.34**	0.26	0.16
BirdChicken	40	512	2	0.07	0.55	**0.63**	0.10	**0.56**	0.53	**0.35**	0.19	0.29
CBF	930	128	3	0.31	**0.37**	**0.37**	0.34	0.39	**0.40**	**0.22**	0.19	0.13
Car	120	577	4	**0.25**	0.16	0.20	**0.35**	0.23	0.29	0.22	**0.28**	0.18
Chinatown	365	24	2	**0.33**	0.29	−0.05	**0.29**	0.21	0.03	0.23	**0.34**	0.27
ChlorineConcentration	4307	166	3	0.00	0.00	0.00	0.00	0.00	0.00	**0.46**	0.42	0.17
CinCECGTorso	1420	1639	4	0.31	0.32	**0.45**	0.37	0.35	**0.52**	0.23	0.19	**0.31**
Coffee	56	286	2	0.61	**1.00**	0.45	0.54	**1.00**	0.42	**0.32**	0.21	0.13
Computers	500	720	2	0.00	0.00	**0.07**	0.00	0.00	**0.05**	**0.46**	0.17	0.27
CricketX	780	300	12	0.15	0.15	**0.16**	**0.32**	0.28	0.30	**0.20**	0.16	0.10
CricketY	780	300	12	0.13	**0.14**	0.08	0.27	**0.28**	0.21	**0.19**	0.17	0.09
CricketZ	780	300	12	0.15	**0.16**	0.14	**0.33**	0.28	0.27	**0.19**	0.16	0.10
Crop	24000	46	24	**0.19**	0.16	0.09	**0.39**	0.34	0.26	**0.19**	0.18	0.09
DiatomSizeReduction	322	345	4	0.67	0.11	**0.69**	**0.67**	0.26	**0.67**	**0.34**	**0.34**	0.26
DistalPhalanxOutlineAgeG	539	80	3	**0.45**	0.44	0.20	**0.35**	0.34	**0.35**	**0.47**	**0.47**	0.44

Table 10 continued

Data set	M	T	k	ARI [−1, 1]			NMI [0, 1]			AS [−1, 1]		
				tsf.	cat.	Net.	tsf.	cat.	Net.	tsf.	cat.	Net.
DistalPhalanxOutlineCorr	876	80	2	0.00	0.00	**0.01**	0.00	0.00	**0.01**	0.45	0.42	**0.69**
DistalPhalanxTW	539	80	6	0.41	**0.45**	0.28	**0.47**	0.45	0.43	0.31	**0.42**	0.11
ECG200	200	96	2	**0.25**	0.07	0.03	**0.16**	0.05	0.04	**0.30**	0.19	0.16
ECG5000	5000	140	5	0.29	0.28	**0.31**	**0.32**	0.29	0.30	**0.24**	**0.24**	0.16
ECGFiveDays	884	136	2	**0.02**	0.00	0.00	**0.02**	0.00	0.01	**0.36**	0.30	0.25
ElectricDevices	16575	96	7	0.20	**0.21**	0.19	**0.30**	0.29	0.29	**0.33**	0.25	0.27
EOGHorizontalSignal	724	1250	12	**0.18**	0.16	0.12	**0.38**	0.33	0.27	**0.27**	0.22	0.13
EOGVerticalSignal	724	1250	12	0.10	**0.13**	0.06	0.26	**0.28**	0.17	**0.21**	0.20	0.11
Earthquakes	461	512	2	**0.00**	−0.03	−0.07	0.00	**0.07**	0.04	0.27	0.18	**0.52**
EthanolLevel	1004	1751	4	**0.01**	0.00	**0.01**	0.01	0.00	**0.02**	0.15	**0.27**	0.15
FaceAll	2250	131	14	0.15	**0.21**	0.15	0.33	**0.36**	0.29	**0.22**	0.15	0.09
FaceFour	112	350	4	0.26	0.36	**0.46**	0.32	0.45	**0.54**	**0.39**	0.22	0.23
FacesUCR	2250	131	14	0.14	**0.19**	0.15	0.33	**0.37**	0.28	**0.22**	0.15	0.09
FiftyWords	905	270	50	**0.19**	**0.19**	0.09	**0.57**	0.54	0.44	**0.18**	0.17	0.11
Fish	350	463	7	**0.19**	0.12	0.17	0.29	0.21	**0.30**	0.20	**0.25**	0.13
FordA	4921	500	2	**0.19**	0.01	0.01	**0.27**	0.01	0.01	**0.53**	0.33	0.29
FordB	4446	500	2	**0.27**	0.07	0.02	**0.31**	0.07	0.02	**0.48**	0.29	0.22
FreezerRegularTrain	3000	301	2	0.22	0.27	**0.31**	0.19	0.21	**0.24**	**0.52**	0.50	0.11
FreezerSmallTrain	2878	301	2	0.22	0.27	**0.32**	0.19	0.21	**0.24**	**0.52**	0.50	0.11
Fungi	204	201	18	**0.77**	0.72	0.40	**0.92**	0.90	0.68	**0.48**	0.43	0.13
GestureMidAirD1	338	360	26	**0.27**	0.25	**0.31**	0.62	0.59	**0.63**	**0.23**	**0.23**	0.15
GestureMidAirD2	338	360	26	0.24	0.25	**0.26**	0.60	0.60	**0.61**	0.21	**0.23**	0.15

Table 10 continued

Data set	M	T	k	ARI [−1, 1]			NMI [0, 1]			AS [−1, 1]		
				tsf.	cat.	Net.	tsf.	cat.	Net.	tsf.	cat.	Net.
GestureMidAirD3	338	360	26	**0.18**	0.14	0.16	**0.52**	0.48	0.50	**0.19**	**0.19**	0.13
GesturePebbleZ1	304	[100, 455]	6	0.15	0.16	**0.23**	0.23	0.22	**0.35**	0.16	0.15	**0.18**
GesturePebbleZ2	304	[100, 455]	6	0.15	0.16	**0.23**	0.23	0.22	**0.35**	0.16	0.15	**0.18**
GunPoint	200	150	2	0.00	0.00	**0.19**	0.06	0.00	**0.30**	**0.70**	0.28	0.27
GunPointAgeSpan	451	150	2	0.01	**0.10**	0.00	0.02	**0.07**	0.00	**0.48**	0.23	0.30
GunPointMaleVersusFemale	451	150	2	0.07	0.05	**0.32**	0.13	0.04	**0.37**	**0.48**	0.23	0.30
GunPointOldVersusYoung	451	150	2	0.00	0.08	**0.20**	0.00	0.06	**0.27**	**0.48**	0.23	0.30
Ham	214	431	2	0.00	0.00	**0.01**	0.00	**0.01**	**0.01**	**0.33**	0.16	0.18
HandOutlines	1370	2709	2	0.05	0.02	**0.06**	0.02	0.00	**0.11**	0.37	0.50	**0.53**
Haptics	463	1092	5	0.04	0.03	**0.07**	0.07	0.07	**0.10**	**0.38**	0.28	0.15
Herring	128	512	2	**0.00**	−0.01	−0.01	0.00	0.00	0.00	**0.22**	0.16	0.17
HouseTwenty	135	3000	2	−0.01	**0.64**	0.10	0.01	**0.61**	0.09	**0.40**	0.21	0.29
InlineSkate	650	1882	7	**0.10**	0.02	0.08	**0.22**	0.07	0.17	**0.29**	0.18	0.16
InsectEPGRegularTrain	311	601	3	0.50	0.43	**0.55**	**0.65**	0.44	0.61	**0.37**	0.22	0.19
InsectEPGSmallTrain	266	601	3	0.50	0.44	**0.52**	**0.65**	0.46	0.61	**0.37**	0.21	0.19
InsectWingbeatSound	2200	256	11	0.07	**0.21**	0.17	0.18	**0.37**	0.32	**0.19**	0.18	0.11
ItalyPowerDemand	1096	24	2	**0.04**	0.01	0.03	**0.05**	0.01	0.03	0.38	**0.40**	0.27
LargeKitchenAppliances	750	720	3	**0.21**	0.06	0.00	**0.23**	0.05	0.01	**0.35**	0.23	0.30
Lightning2	121	637	2	**0.07**	0.02	0.05	**0.14**	0.04	0.07	**0.42**	0.28	0.19
Lightning7	143	319	7	**0.22**	0.21	0.18	**0.39**	**0.39**	0.35	**0.24**	**0.24**	0.14
Mallat	2400	1024	8	**0.70**	0.69	0.53	0.80	**0.83**	0.65	**0.35**	0.32	0.13
Meat	120	448	3	**0.45**	**0.45**	0.17	0.46	**0.63**	0.18	0.29	**0.43**	0.13

Table 10 continued

Data set	M	T	k	ARI [−1, 1]			NMI [0, 1]			AS [−1, 1]		
				tsf.	cat.	Net.	tsf.	cat.	Net.	tsf.	cat.	Net.
MedicalImages	1141	99	10	**0.10**	0.03	0.06	**0.28**	0.19	0.17	**0.31**	0.21	0.17
MiddlePhalanxOutlineAgeG	554	80	3	0.42	**0.43**	0.42	0.39	0.39	0.39	**0.56**	0.49	0.40
MiddlePhalanxOutlineCorr	891	80	2	−0.01	**0.00**	−0.01	**0.01**	0.00	0.01	0.47	0.41	**0.71**
MiddlePhalanxTW	553	80	6	0.34	**0.57**	0.24	0.40	**0.43**	0.39	0.28	**0.47**	0.14
MixedShapesRegularTrain	2925	1024	5	0.44	0.23	**0.52**	0.49	0.26	**0.55**	**0.31**	0.20	0.21
MixedShapesSmallTrain	2525	1024	5	0.45	0.23	**0.51**	0.49	0.25	**0.54**	**0.31**	0.20	0.21
MoteStrain	1272	84	2	0.01	0.02	**0.17**	0.01	0.01	**0.17**	**0.48**	0.25	0.20
NonInvasiveFetalECGThor1	3765	750	42	**0.51**	0.27	0.07	**0.76**	0.58	0.30	**0.22**	0.20	0.09
NonInvasiveFetalECGThor2	3765	750	42	**0.54**	0.36	0.11	**0.79**	0.67	0.36	**0.25**	0.23	0.11
OSULeaf	442	427	6	0.29	0.28	**0.49**	0.42	0.38	**0.54**	**0.17**	0.15	0.16
OliveOil	60	570	4	**0.29**	0.18	0.10	**0.36**	0.27	0.17	0.30	**0.32**	0.10
PLAID	1074	[100, 1344]	11	**0.38**	0.27	0.25	**0.51**	0.41	0.38	**0.31**	0.22	0.18
PhalangesOutlinesCorrect	2658	80	2	0.01	**0.01**	0.00	0.00	0.00	0.00	0.33	0.37	**0.70**
Phoneme	2110	1024	39	0.06	0.07	**0.09**	0.29	0.31	**0.34**	0.14	**0.13**	**0.13**
PickupGestureWiimoteZ	100	[29, 361]	10	0.22	0.24	**0.29**	0.50	0.50	**0.55**	0.18	0.18	**0.19**
PigAirwayPressure	312	2000	52	0.11	0.04	**0.13**	0.63	0.57	**0.64**	**0.26**	0.20	0.17
PigArtPressure	312	2000	52	0.40	**0.45**	0.44	0.79	**0.82**	0.81	**0.24**	**0.24**	0.22
PigCVP	312	2000	52	0.13	0.18	**0.20**	0.64	**0.68**	0.68	**0.20**	0.14	0.16
Plane	210	144	7	**0.98**	0.86	0.97	**0.98**	0.90	0.97	**0.58**	0.34	0.37
PowerCons	360	144	2	0.13	0.00	**0.25**	0.11	0.00	**0.22**	**0.22**	0.20	**0.22**

Table 10 continued

Data set	M	T	k	ARI [−1, 1]			NMI [0, 1]			AS [−1, 1]		
				tsf.	cat.	Net.	tsf.	cat.	Net.	tsf.	cat.	Net.
ProximalPhalanxOutlineAg	605	80	3	0.55	**0.57**	0.35	0.53	**0.56**	0.45	0.49	0.40	**0.50**
ProximalPhalanxOutlineCo	891	80	2	0.05	**0.06**	0.04	0.07	**0.08**	0.05	0.44	0.49	**0.73**
ProximalPhalanxTW	605	80	6	0.43	**0.47**	0.34	0.56	**0.58**	0.48	0.29	**0.41**	0.15
RefrigerationDevices	750	720	3	0.02	0.02	0.02	0.02	0.02	0.02	**0.41**	0.27	0.27
Rock	70	2844	4	**0.36**	0.16	0.35	**0.51**	0.27	0.49	0.25	0.23	0.24
ScreenType	750	720	3	0.03	**0.04**	0.01	0.03	**0.04**	0.01	**0.31**	0.16	0.22
SemgHandGenderCh2	900	1500	2	0.00	0.00	**0.01**	**0.01**	0.00	0.00	0.29	0.23	**0.31**
SemgHandMovementCh2	900	1500	6	0.06	0.04	**0.10**	0.14	0.12	**0.20**	**0.24**	0.23	0.22
SemgHandSubjectCh2	900	1500	5	0.16	**0.26**	0.18	0.25	**0.33**	0.28	0.25	0.24	**0.26**
ShakeGestureWiimoteZ	100	[40, 385]	10	0.54	0.45	**0.56**	0.72	0.67	**0.74**	**0.29**	0.20	0.27
ShapeletSim	200	500	2	0.10	**1.00**	0.85	0.08	**1.00**	0.78	0.18	**0.33**	0.17
ShapesAll	1200	512	60	**0.37**	0.28	0.23	**0.70**	0.65	0.60	**0.26**	0.22	0.13
SmallKitchenAppliances	750	720	3	**0.19**	0.08	0.18	0.19	0.07	**0.20**	0.32	0.25	**0.50**
SonyAIBORobotSurface1	621	70	2	**0.58**	0.42	0.56	**0.53**	0.47	0.47	**0.34**	0.26	0.25
SonyAIBORobotSurface2	980	65	2	**0.55**	0.37	0.00	**0.47**	0.28	0.01	**0.37**	0.22	0.14
StarLightCurves	9236	1024	3	0.49	0.43	**0.65**	**0.57**	0.56	0.53	**0.42**	0.34	0.32
Strawberry	983	235	2	−0.05	−0.02	**0.01**	**0.11**	0.09	0.03	**0.52**	0.35	0.16
SwedishLeaf	1125	128	15	**0.46**	0.31	0.41	**0.68**	0.53	0.60	**0.25**	0.23	0.16
Symbols	1020	398	6	0.69	0.65	**0.77**	0.78	0.79	**0.84**	0.40	**0.49**	0.35
SyntheticControl	600	60	6	0.57	**0.61**	0.43	0.71	**0.74**	0.50	**0.34**	0.20	0.14
ToeSegmentation1	268	277	2	0.01	0.00	**0.05**	0.01	0.00	**0.05**	0.22	**0.23**	0.20
ToeSegmentation2	166	343	2	0.12	0.07	**0.37**	0.06	0.03	**0.26**	0.30	0.18	**0.37**

Table 10 continued

Data set	M	T	k	ARI [-1, 1]			NMI [0, 1]			AS [-1, 1]		
				tsf.	cat.	Net.	tsf.	cat.	Net.	tsf.	cat.	Net.
Trace	200	275	4	**1.00**	0.73	0.63	**1.00**	0.79	0.70	**0.65**	0.35	0.22
TwoLeadECG	1162	82	2	0.00	0.01	**0.71**	0.00	0.01	**0.61**	**0.60**	0.16	0.19
TwoPatterns	5000	128	4	**0.14**	0.00	0.01	**0.17**	0.00	0.01	**0.16**	0.15	0.11
UMD	180	150	3	**0.48**	0.17	0.35	**0.53**	0.20	0.38	**0.41**	0.24	0.21
UWaveGestureLibraryAll	4478	945	8	0.17	**0.20**	0.18	0.27	**0.28**	**0.28**	**0.20**	0.19	0.12
UWaveGestureLibraryX	4478	315	8	0.18	0.19	**0.23**	0.30	0.29	**0.33**	0.19	**0.20**	0.15
UWaveGestureLibraryY	4478	315	8	**0.22**	0.16	0.14	**0.36**	0.25	0.25	**0.22**	0.20	0.15
Wafer	7164	152	2	-0.02	-0.04	**0.99**	0.00	0.02	**0.96**	**0.56**	0.33	0.51
Wine	111	234	2	**0.03**	-0.01	0.01	**0.03**	0.00	0.01	0.39	**0.43**	0.22
WordSynonyms	905	270	25	**0.14**	0.10	0.05	**0.41**	0.34	0.26	**0.21**	0.18	0.11
Worms	258	900	5	**0.19**	0.14	0.13	**0.24**	0.22	0.22	**0.26**	0.17	0.22
WormsTwoClass	258	900	2	0.04	0.07	**0.12**	0.02	0.04	**0.08**	**0.34**	0.19	0.29
Yoga	3300	426	2	0.00	0.00	0.00	0.00	0.00	0.00	**0.29**	0.20	0.27
Mean				**0.24**	0.22	**0.24**	**0.33**	0.30	0.32	**0.33**	0.26	0.23
Win				46	28	**47**	**48**	28	43	**82**	26	16
Win (%)				38.66	23.53	**39.50**	**40.34**	23.53	36.13	**68.91**	21.85	13.45

The values reflect the mean of 10 repetitions of the clustering analysis for the ground truth, k. The values in bold represent the best results of the respective evaluation metric comparing the two approaches. M represents the size of data set, T the time series length and k the number of classes

References

Albert R, Barabási AL (2002) Statistical mechanics of complex networks. Rev Mod Phys 74(1):47. https://doi.org/10.1103/RevModPhys.74.47

Bagnall A, Lines J, Vickers W, Keogh E. The UEA & UCR time series classification repository. www.timeseriesclassification.com

Barabási AL (2016) Network science. Cambridge University Press, Cambridge

Barandas M, Folgado D, Fernandes L, Santos S, Abreu M, Bota P, Liu H, Schultz T, Gamboa H (2020) Tsfel: Time series feature extraction library. SoftwareX 11:100456. https://doi.org/10.1016/j.softx.2020.100456

Bianchi FM, Livi L, Alippi C, Jenssen R (2017) Multiplex visibility graphs to investigate recurrent neural network dynamics. Sci Rep 7:44037. https://doi.org/10.1038/srep44037

Bonner S, Brennan J, Theodoropoulos G, Kureshi I, McGough AS (2016) Deep topology classification: a new approach for massive graph classification. In: IEEE international conference on Big Data. IEEE, pp 3290–3297. https://doi.org/10.1109/BigData.2016.7840988

Box GE, Jenkins GM, Reinsel GC, Ljung GM (2015) Time series analysis: forecasting and control. Wiley, New York

Bradley E, Kantz H (2015) Nonlinear time-series analysis revisited. Chaos 25(9):4917289. https://doi.org/10.1063/1.4917289

Campanharo A, Ramos F (2017) Distinguishing different dynamics in electroencephalographic time series through a complex network approach. Proc Ser Braz Soc Comput Appl Math 5(1):0102. https://doi.org/10.5540/03.2017.005.01.0102

Campanharo AS, Ramos FM (2016) Hurst exponent estimation of self-affine time series using quantile graphs. Physica A 444:43–48. https://doi.org/10.1016/j.physa.2015.09.094

Campanharo AS, Sirer MI, Malmgren RD, Ramos FM, Amaral LAN (2011) Duality between time series and networks. PLoS ONE 6(8):e23378. https://doi.org/10.1371/journal.pone.0023378

Chiu B, Keogh E, Lonardi S (2003) Probabilistic discovery of time series motifs. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 493–498. https://doi.org/10.1145/956750.956808

Christ M, Braun N, Neuffer J, Kempa-Liehr AW (2018) Time series feature extraction on basis of scalable hypothesis tests (tsfresh-a python package). Neurocomputing 307:72–77. https://doi.org/10.1016/j.neucom.2018.03.067

Costa LdF, Rodrigues FA, Travieso G, Villas Boas PR (2007) Characterization of complex networks: a survey of measurements. Adv Phys 56(1):167–242. https://doi.org/10.1080/00018730601170527

Costa LdF, Oliveira ON Jr, Travieso G, Rodrigues FA, Villas Boas PR, Antiqueira L, Viana MP, Correa Rocha LE (2011) Analyzing and modeling real-world phenomena with complex networks: a survey of applications. Adv Phys 60(3):329–412. https://doi.org/10.1080/00018732.2011.572452

Cryer JD, Chan KS (2008) Time series analysis with applications in R. Springer, New York

Csardi G, Nepusz T (2006) The igraph software package for complex network research. InterJournal Complex Systems, 1695. http://igraph.org

Fortunato S (2010) Community detection in graphs. Phys Rep 486(3–5):75–174. https://doi.org/10.1016/j.physrep.2009.11.002

Fulcher BD (2018) Feature-based time-series analysis. In: Feature engineering for machine learning and data analytics. CRC Press, pp 87–116

Fulcher BD, Jones NS (2014) Highly comparative feature-based time-series classification. IEEE Trans Knowl Data Eng 26(12):3026–3037. https://doi.org/10.1109/TKDE.2014.2316504

Fulcher BD, Jones NS (2017) hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell Syst 5(5):527–531. https://doi.org/10.1016/j.cels.2017.10.001

Fulcher BD, Little MA, Jones NS (2013) Highly comparative time-series analysis: the empirical structure of time series and their methods. J R Soc Interface 10(83):20130048. https://doi.org/10.1098/rsif.2013.0048

Geurts P (2001) Pattern extraction for time series classification. In: European conference on principles of data mining and knowledge discovery. Springer, pp 115–127. https://doi.org/10.1007/3-540-44794-6_10

Hartigan J, Wong M (1979) A k-means clustering algorithm. Appl Stat 28:100–108

Henderson T (2021) Rcatch22: Calculation of 22 CAnonical Time-Series CHaracteristics. R package version 0.1.13

Henderson T, Fulcher BD (2021) An empirical evaluation of time-series feature sets

Hyndman R (2018) Mcomp: Data from the M-Competitions. https://CRAN.R-project.org/package=Mcomp. R package version 2.8

Hyndman R, Kang Y, Montero-Manso P, Talagala T, Wang E, Yang Y, O'Hara-Wild M (2020) tsfeatures: Time Series Feature Extraction. https://CRAN.R-project.org/package=tsfeatures. R package version 1.0.2

Hyndman RJ, Fan Y (1996) Sample quantiles in statistical packages. Am Stat 50(4):361–365. https://doi.org/10.1080/00031305.1996.10473566

Hyndman RJ, Wang E, Laptev N (2015) Large-scale unusual time series detection. In: 2015 IEEE international conference on data mining workshop (ICDMW). IEEE, pp 1616–1619. https://doi.org/10.1109/ICDMW.2015.104

Instituto Brasileiro de Geografia e Estatística - IBGE. https://www.ibge.gov.br

Kang Y, Hyndman RJ, Li F (2020) Gratis: Generating time series with diverse and controllable characteristics. Statistical Analysis and Data Mining: The ASA Data Science Journal. https://doi.org/10.1002/sam.11461

Kang Y, Hyndman RJ, Smith-Miles K (2017) Visualising forecasting algorithm performance using time series instance spaces. Int J Forecast 33(2):345–358. https://doi.org/10.1016/j.ijforecast.2016.09.004

Lacasa L, Luque B, Ballesteros F, Luque J, Nuno JC (2008) From time series to complex networks: the visibility graph. Proc Natl Acad Sci 105(13):4972–4975. https://doi.org/10.1073/pnas.0709247105

Lan X, Mo H, Chen S, Liu Q, Deng Y (2015) Fast transformation from time series to visibility graphs. Chaos 25(8):4927835. https://doi.org/10.1063/1.4927835

Li X, Cheng R, Chang KCC, Shan C, Ma C, Cao H (2021) On analyzing graphs with motif-paths. Proceedings of the VLDB Endowment 14(6):1111–1123. https://doi.org/10.14778/3447689.3447714

Lü L, Zhou T (2011) Link prediction in complex networks: a survey. Physica A 390(6):1150–1170. https://doi.org/10.1016/j.physa.2010.11.027

Lubba CH, Sethi SS, Knaute P, Schultz SR, Fulcher BD, Jones NS (2019) catch22: CAnonical Time-series CHaracteristics. Data Min Knowl Disc 33(6):1821–1852. https://doi.org/10.1007/s10618-019-00647-x

Luque B, Lacasa L, Ballesteros F, Luque J (2009) Horizontal visibility graphs: exact results for random time series. Phys Rev E 80(4):046103. https://doi.org/10.1103/PhysRevE.80.046103

Maechler M, Fraley C, Leisch F, Reisen V, Lemonte A, Hyndman RJ (2020) fracdiff: Fractionally differenced ARIMA aka ARFIMA(p,d,q) models. https://CRAN.R-project.org/package=fracdiff. R package version 1.5-1

Maharaj EA, D'Urso P, Caiado J (2019) Time series clustering and classification. CRC Press, Boca Raton

Malliaros FD, Vazirgiannis M (2013) Clustering and community detection in directed networks: a survey. Phys Rep 533(4):95–142. https://doi.org/10.1016/j.physrep.2013.08.002

Montero P, Vilar JA (2014) TSclust: An R package for time series clustering. J Stat Softw 62(1):1–43. https://doi.org/10.18637/jss.v062.i01

Montero-Manso P, Athanasopoulos G, Hyndman RJ, Talagala TS (2020) FFORMA: feature-based forecast model averaging. Int J Forecast 36(1):86–92. https://doi.org/10.1016/j.ijforecast.2019.02.011

O'Hara-Wild M, Hyndman R, Wang E (2021) feasts: Feature extraction and statistics for time series. https://CRAN.R-project.org/package=feasts. R package version 0.2.1

Oldham S, Fulcher B, Parkes L, Arnatkevičiūtė A, Suo C, Fornito A (2019) Consistency and differences between centrality measures across distinct classes of networks. PLoS ONE 14(7):e0220061. https://doi.org/10.1371/journal.pone.0220061

Peach RL, Arnaudon A, Schmidt JA, Palasciano HA, Bernier NR, Jelfs KE, Yaliraki SN, Barahona M (2021) HCGA: highly comparative graph analysis for network phenotyping. Patterns 2(4):100227. https://doi.org/10.1016/j.patter.2021.100227

Pons P, Latapy M (2005) Computing communities in large networks using random walks. In: International symposium on computer and information sciences. Springer Berlin Heidelberg, pp 284–293. https://doi.org/10.1007/11569596_31

R Core Team (2020) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

Shao ZG (2010) Network analysis of human heartbeat dynamics. Appl Phys Lett 96(7):3308505. https://doi.org/10.1063/1.3308505

Shumway RH, Stoffer DS (2017) Time series analysis and its applications. Springer, Berlin

Silva ME, Oliveira VL (2004) Difference equations for the higher-order moments and cumulants of the INAR(1) model. J Time Ser Anal 25(3):317–333. https://doi.org/10.1111/j.1467-9892.2004.01685.x

Silva VF (2018) Time series analysis based on complex networks. MSc thesis, University of Porto

Silva VF, Silva ME, Ribeiro P, Silva F (2021) Time series analysis via network science: concepts and algorithms. WIREs Data Min Knowl Discov 11(3):1404. https://doi.org/10.1002/widm.1404

Talagala TS, Hyndman RJ, Athanasopoulos G et al (2018) Meta-learning how to forecast time series. Monash Econometrics and Business Statistics Working Papers 6:18

Telesca L, Lovallo M (2012) Analysis of seismic sequences by using the method of visibility graph. EPL (Europhys Lett) 97(5):50002. https://doi.org/10.1209/0295-5075/97/50002

Tong H (2011) Threshold models in time series analysis—30 years on. Stat Interface 4(2):107–118. https://doi.org/10.4310/SII.2011.v4.n2.a1

Tsay RS (2010) Analysis of financial time series, 3rd edn. Wiley, New York

Vespignani A (2018) Twenty years of network science

Wang X, Smith K, Hyndman RJ (2006) Characteristic-based clustering for time series data. Data Min Knowl Discov 13(3):335–364. https://doi.org/10.1007/s10618-005-0039-x

Witowski V, Foraita DR (2014) HMMpa: Analysing accelerometer data using hidden Markov models. https://CRAN.R-project.org/package=HMMpa. R package version 1.0

Wuertz D, Setz T, Chalabi Y (2017) timeSeries: Rmetrics - Financial Time Series Objects. https://CRAN.R-project.org/package=timeSeries. R package version 3042.102

Wuertz D, Setz T, Chalabi Y, Boudt C, Chausse P, Miklovac M (2017) fGarch: Rmetrics - Autoregressive Conditional Heteroskedastic Modelling. https://CRAN.R-project.org/package=fGarch. R package version 3042.83

Zhu G, Li Y, Wen PP (2014) Analysis and classification of sleep stages based on difference visibility graphs from a single-channel eeg signal. IEEE J Biomed Health Inform 18(6):1813–1821. https://doi.org/10.1109/JBHI.2014.2303991

Zou Y, Donner RV, Marwan N, Donges JF, Kurths J (2019) Complex network approaches to nonlinear time series analysis. Phys Rep 787:1–97. https://doi.org/10.1016/j.physrep.2018.10.005

Zucchini W, MacDonald IL, Langrock R (2016) Hidden Markov models for time series: an introduction using R. Chapman and Hall/CRC, Boca Raton