Clustering Large-scale Time Series

Journal of Classification (2026) 43:236–274
https://doi.org/10.1007/s00357-025-09535-0

ORIGINAL RESEARCH

Clustering Large-scale Time Series

Pierpaolo D'Urso[1] · Livia De Giovanni[2] · Ruey S. Tsay[3] · Vincenzina Vitale[1]

Received: 14 February 2024 / Accepted: 16 December 2025 / Published online: 27 January 2026

Abstract
This paper proposes clustering methods for large-scale stationary time series using a fuzzy approach. Adopting partitioning around centroids (PAC) and partitioning around medoids (PAM), and focusing on distributional properties of individual series, we classify a large set of time series by transforming the series into probability density functions via nonparametric density estimation, such as the kernel estimation, and using a proper distance measure, such as the Hellinger distance, between density functions. We use simulations and two real applications to demonstrate the good performance and effectiveness of the proposed clustering methods in finite samples. The proposed methods are also applicable to the spectral density functions if one focuses on the serial dependence of individual series.

Keywords Large-scale dependent data · Kernel density estimation · Hellinger distance · Fuzzy clustering · Partitioning around centroids · Partitioning around medoids

1 Introduction

Researchers in empirical data analysis often face complicated data characterized by diverse features, including structures with different levels of complexity and different dimensions (Arabie et al., 1996). Recently, extensive research across fields like genomics, medicine, neuroscience, machine learning, finance, and meteorology has focused on analyzing high-dimensional or large-scale data (e.g., the 2016 Special Issue of the *Journal of Business & Economic Statistics* and the references therein). While methods for analyzing large-scale independent data are well-established (Gareth et al., 2014), the analysis of large-scale dependent data remains less explored. According to Tsay (2016) and Peña and Tsay (2021),

✉ Livia De Giovanni
ldegiovanni@luiss.it

Pierpaolo D'Urso
pierpaolo.durso@uniroma1.it

Ruey S. Tsay
ruey.tsay@chicagobooth.edu

Vincenzina Vitale
vincenzina.vitale@uniroma1.it

1 Department of Social Sciences and Economics, Sapienza - University of Rome, Rome, Italy

2 Department of AI, Data and Decision Sciences Luiss università di Roma, Rome, Italy

3 Booth School of Business, University of Chicago, Chicago, IL, USA

large-scale dependent data can be understood as time series or spatial processes, which may arise from either many variables (high dimensionality), many observations over long periods or at high frequencies, or both.[1] As remarked by Tsay (2016), "for time-series data, the dynamic cross-dependence between individual time series often provides valuable information for making statistical inference and for sharpening prediction of the series of interest. On the other hand, it is not easy to adequately capture the dynamic relationships between the series when the dimension is high. It is also not a simple task to comprehend the complexity of high-dimensional time-series models available in the literature."

Cluster analysis plays an important role in statistical applications, and many useful algorithms and strategies are available for clustering large-scale data, but only a few of them address the problem from a fuzzy approach (Havens et al., 2012). Moreover, existing fuzzy clustering methods have been proposed for large-scale independent data. To the best of our knowledge, there exists no literature on fuzzy clustering methods designed specifically for large-scale dependent data. The goal of this paper is to fill this gap. Within a fuzzy framework, we adopt different partitioning strategies to develop clustering methods for large-scale stationary time series, using nonparametric density estimation. We show that the fuzzy approach, combined with kernel preprocessing, can be effective for clustering large-scale dependent data. If one is interested in the serial dependence of individual series, the marginal density functions can be replaced with spectral density functions, and the proposed clustering methods can still be applied.

The importance of this dual treatment of uncertainty lies in its ability to capture complementary dimensions of variability. Probabilistic uncertainty reflects the randomness inherent in data generation, such as fluctuations in electoral turnout or financial time series volatility, while fuzzy uncertainty accounts for the imprecision of conceptual boundaries—for instance, when regions or individuals exhibit hybrid behaviors that cannot be cleanly assigned to a single group. By explicitly addressing both, the model avoids the pitfalls of oversimplification: without probabilistic modeling, we risk ignoring stochastic variability; without fuzzy modeling, we risk forcing artificial crisp boundaries where reality is ambiguous. Empirical evidence shows that models which embrace both aleatory randomness and vagueness in classification tend to produce clusters that are more stable across samples, more interpretable to domain experts, and more resilient in high-noise environments. This layered representation of uncertainty is not merely technical, but deeply practical: it equips analysts with insights that align more closely with the nuanced, overlapping, and often noisy nature of complex real-world systems. These aspects are further discussed in Remark 1.

To begin, it is important to underline that a useful and effective clustering method often depends on the objective. For example, consider $i = 1000$ time series of length T, $\{x_{it}\}$, each generated from a simple AR(1) model, say $x_{it} = 0.6\, x_{i,t-1} + e_{it}$, where $\{e_{it}\}$ are independent standard Gaussian random variables. If the goal is to group series by their marginal distributions, then all series are consistent estimates of the true density function of the AR(1) process and all series form a single cluster. In this case, the implicit objective is the dynamic dependence of the time series. If the goal is prediction, to which cross-dependence matters, then each independent series forms its own cluster, since $\text{corr}(x_{it}, x_{j,t-k}) = 0$ for $(i \neq j)$ and any integer k. Therefore, x_{jt} is not useful in prediction x_{it} for $j \neq i$. There are many ways to cluster time series. Which summary statistics to use often depends on the objective of clustering. Marginal density functions are one of the summary statistics for high-dimensional

[1] "By dependent data, we mean that observations of the variables of interest were taken over time, across space, or both. Consequently, in these data the order in time or space matters. By big data, we mean that the number of variables, k, is large or the number of data points, T, is large or both, and it could even be that $k > T$" Peña and Tsay (2021).

time series. They carry some information concerning the dynamic dependence of the series, as demonstrated in the simulations and empirical examples in Sections 4 and 5. In general, different time series models lead to different marginal distributions. The use of marginal density functions has one advantage over clustering based on autocorrelations as marginal density functions also depend on the distribution of the innovations. The marginal density functions do not lose the autocorrelation information. Consider, for instance, the marginal density functions of an MA(1) model and an AR(1) model. They have different marginal density functions, even if the innovations of the two time series share the same distributions. Marginal density functions depend on the dynamic dependence and the distribution of the innovations of a time series. They are one of the possible criteria to perform time series clustering.

In this paper, our objective is to cluster large-scale time series by focusing on some distributional properties of individual series. To this end, we employ kernel estimates of the marginal density functions of individual series. In some applications, the marginal distributions of time series are of major interest (Peña & Tsay, 2021, and the references therein). Some theoretical properties, such as the consistency of the kernel estimation of the marginal density of stationary time series, are available in Hardle et al. (1989). To this end, we propose, under a fuzzy framework, the adoption of different partitioning strategies, namely, the partitioning around centroids (PAC) and the partitioning around medoids (PAM) strategies, to derive clustering methods for a large set of stationary time series using a nonparametric density estimation and a proper distance measure between density functions.

The paper is organized as follows. Section 2 reviews existing fuzzy clustering methods for large-scale data, which are limited to the case of independent observations. Section 3 describes how to transform large-scale time series into kernel-based probability density functions, introduces the Hellinger distance for comparing these densities, and presents two fuzzy clustering methods—one based on PAC and another on PAM. Section 4 reports simulation results to evaluate the performance of the proposed methods in finite samples, Section 5 covers two empirical applications, and Section 6 offers concluding remarks and future research directions.

2 Fuzzy Clustering of Large-Scale Data

The following literature review focuses on fuzzy clustering techniques tailored to large-scale independent data, providing a basis that justifies the development of new methodologies for dependent data.

Many useful algorithms and strategies are available for clustering large-scale data. Standard clustering methods like c-means or partitioning around centroids (PAC MacQueen, 1967), partitioning around medoids (PAM) or c-medoids (Kaufman & Rousseeuw, 1990), and hierarchical methods are effective for small datasets but require adaptations for large datasets (Kriegel et al., 2009). Havens et al. (2012) categorized the clustering methods for large-scale data into three main classes:

Sampling-based methods: Fuzzy clustering method (FcM) is applied to sampled data. If the data were sufficiently sampled, the error between the cluster center locations produced by clustering the entire data set and those from the sampled data should be small.

Single pass-based methods: They sequentially load small groups of the data, clustering these manageable chunks in a single pass, and then combining the results from each chunk.

Data transformation-based methods: They summarize the structure of the data so that they can be more efficiently accessed.

As noted by Galeano and Peña (2019), two key problems in clustering large-scale data are as follows:

1. The presence of irrelevant variables, for they negatively affect the efficiency of proximity measures. This problem has been tackled by variable selection, which is carried out by adding some penalty function in the estimation criterion, such as the Lasso method. Witten and Tibshirani (2010) proposed a clustering method that can be applied to obtain sparse versions of c-means and hierarchical clustering. See also Bouveyron and Brunet (2014).
2. The curse of dimensionality, which may result in heavy computation and substantial memory requirements. This problem has been tackled by dimension reduction, which is carried out by identifying some subspace that includes the relevant information for clustering. See, for instance, Bouveyron and Brunet (2014).

However, only a few works address the problem from a fuzzy approach (Havens et al., 2012). Considering large-scale data clustering in a fuzzy framework, Shankar and Pal (1994) suggested the fast FcM (fFcM), in which FcM is applied to larger and larger nested samples until there is little change in the solution. Cheng et al. (1995) developed the multistage random FcM, which combines fFcM with a final literal run of FcM on the full data set. Other available algorithms include Cannon et al. (1986) and Kolen and Hutcheson (2002). Eschrich et al. (2003) proposed a bit-reduced FcM (brFcM) algorithm, which uses a binning strategy for data reduction. The most well-known fuzzy clustering method for large-scale data is the generalized extensible fast FcM (geFFcM) proposed by Hathaway and Bezdek (2006). This method utilizes statistics-based progressive sampling to produce a reduced data set that is sufficiently large to capture the overall nature of the data. The algorithm then clusters this reduced data set and non-iteratively extends the partition to the full data set. However, as remarked by Havens et al. (2012), the sampling method utilized in geFFcM can be inefficient, and in some cases, the data reduction is not sufficient for large-scale datasets. Hence, these authors adapted the geFFcM to a simple random sampling plus extension FcM (rseFcM) algorithm. Liao and Sheng Lin (2007) proposed the fast kernel FcM (fkFcM). Other leading clustering algorithms include the single-pass FcM (spFcM, Hore et al., 2007) and the online FcM (oFcM, Hore et al., 2009), both of which are incremental algorithms to compute an approximate FCM solution. Subsequently, a kernel-based strategy, called approximate kernel FCM (akFcM), was developed by Chitta et al. (2011) and Havens et al. (2011), which relies on a numerical approximation that uses sampled rows of the kernel matrix to estimate the solution to a c-means problem. Havens et al. (2012) compared four leading algorithms (namely, rseFCM, spFCM, oFCM, and brFCM) and four kernel FCM (kFCM) (namely, rsekFCM, akFCM, spkFCM, and okFCM) for fuzzy partitions with large-scale data. Notice that both spkFCM and okFCM have been proposed by Havens et al. (2012) as kernelized extensions of the spFCM and oFCM algorithms. Recently, Wu et al. (2017) proposed a fuzzy consensus clustering with application on large-scale data, and Shen et al. (2019) suggested a fuzzy c-means clustering for large-scale data based on a hyperplane division method to split the entire large-scale data set into disjoint subsets.

Since all the above techniques are designed for independent large-scale data, a notable gap remains in methods suitable for dependent data, such as time series. To address this limitation, the following section introduces a novel fuzzy clustering framework specifically tailored to large-scale stationary time series.

3 Fuzzy Clustering of Large-scale Time Series

Methodological approaches available in the literature to clustering time series can be classified in the following theoretical categories (Caiado et al., 2015):

Observation-based (or raw data-based) clustering: It relies on the raw data to conduct clustering analysis using a suitable distance measure (see, e.g., D'Urso et al., 2018, and references therein).

Model-based clustering: It analyzes the characteristics of the models fitted to the time series, i.e., ARIMA models (see, e.g., D'Urso et al., 2015; Maharaj, 1999; Piccolo, 1990); GARCH models (see, e.g., D'Urso et al., 2016); TAR models (see, e.g., Tong & Dabas, 1990); splines (see, e.g., García-Escudero & Gordaliza, 2005); and distribution models (see, e.g., Alonso & Maharaj, 2006; D'Urso et al., 2017; Maharaj et al., 2015; Tsay, 2016; Pértega & VilarJosé, 2010; Bouveyron et al., 2019).

Feature-based clustering: It employs certain features derived from the observed time series, i.e., autocorrelation function (see, e.g., Alonso & Maharaj, 2006; D'Urso & Maharaj, 2009); partial autocorrelation function (see, e.g., Caiado & Crato, 2009); quantile autocovariance (see, e.g., Vilar et al., 2018; Lafuente Rego et al., 2018); cross correlation (see, e.g., Dose and Cincotti, 2005; Alonso & Peña, 2019); periodogram and its transformations (see, e.g., Caiado et al., 2006; Maharaj & D'Urso, 2011); coherence (see, e.g., Maharaj & D'Urso, 2010; Euán et al., 2019); wavelet decomposition (see, e.g., D'Urso & Maharaj, 2012; D'Urso et al., 2014; Maharaj et al., 2010); hurst exponent estimates (see, e.g., Lahmiri, 2016); and cepstral (see, e.g., Martin, 2000; Maharaj & D'Urso, 2011; D'Urso et al., 2020).
For an interesting review on this topic, see also Aghabozorgi et al. (2015).

Most of the available clustering methods employ a hard boundary approach so that the boundaries between clusters are well defined. However, there are situations under which the boundary between clusters may not be as clear as one would like. For instance, consider the simple case of two normal distributions with different means, but the same variance. In this case, the boundary between the two clusters may not be clear if the two means are not sufficiently different, given the variance. To mitigate such difficulties, we employ a fuzzy approach, even though it is less commonly used in the econometric literature. Specifically, we follow the idea of transforming large-scale time series into their marginal probability density functions used in Tsay (2016) and propose clustering methods based on the fuzzy theory. In other words, we transform the high-dimensional time series into probability density functions via nonparametric density estimation, then consider a suitable distance measure for the resulting density functions, and employ fuzzy approaches to propose clustering methods. For simplicity, we employ the commonly used kernel density estimation and the well-known Hellinger distance in this paper. Our proposed fuzzy clustering methods adopt different partitioning strategies, namely, the partitioning around centroids (PAC) and the partitioning around medoids (PAM). Before providing details of the proposed methods, Fig. 1 shows a flowchart of our proposed clustering procedure.

The proposed clustering approach inherits all the benefits of its ingredients used in the methodological framework:

Fuzzy theory: Due to the difficulty in identifying a clear boundary between clusters in real applications, fuzzy clustering is more attractive than the hard clustering methods. See, for instance, McBratney and Moore (1985). In addition, the memberships indicate whether there is a second-best cluster almost as good as the best one, a scenario which hard clustering

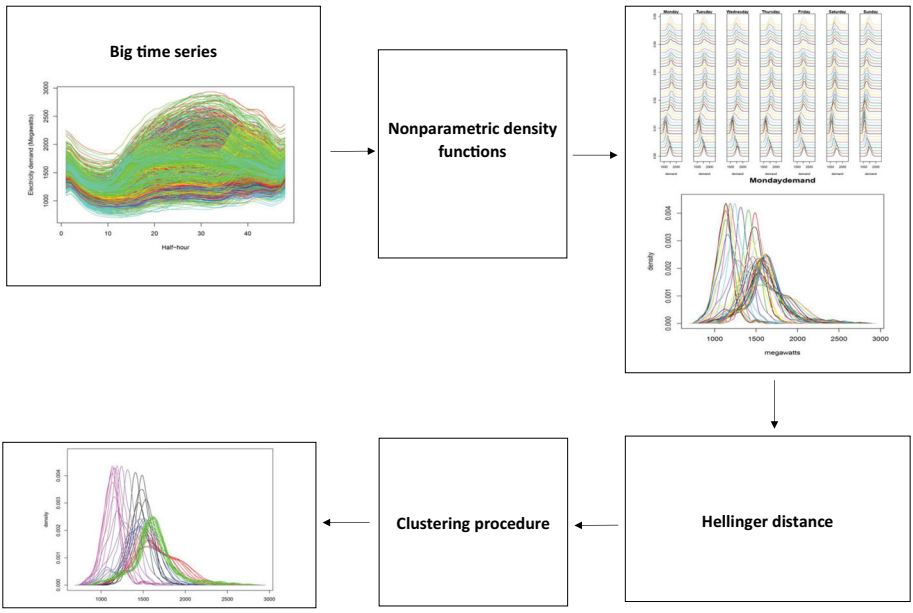


Fig. 1 A sketch of the proposed scheme for clustering large-scale time series data

methods cannot uncover (Everitt et al., 2011). Furthermore, fuzzy clustering is attractive because it is easily compatible with distribution-free methods (Hwang et al., 2007) and it is computationally efficient (McBratney & Moore, 1985; Heiser & Groenen, 1997). For more details, see D'Urso (2015).

Partitioning around centroids (PAC) and partitioning around medoids (PAM) approaches: Adopting the PAC approach, the proposed clustering procedure enjoys the benefits connected to the non-hierarchical clustering method, including the fuzzy-c-means for their simple mathematical ideas, low computational complexity, fast convergence, and easy implementation. Adopting the PAM approach, the prototypes of each cluster, henceforth the medoid time series, are observed time series instead of a "virtual" one such as the "centroids" derived with a fuzzy c-means (Bezdek, 1981). Overall, having non-fictitious representative time series available makes interpreting the obtained clusters easier, which is often useful in real applications. In fact, as remarked by Kaufman and Rousseeuw (1990), "in many clustering problems one is particularly interested in a characterization of the clusters by means of typical or representative objects [time series]. These are objects that represent the various structural aspects of the set of objects being investigated. There are many reasons for searching for representative objects [time series]. Not only can these objects provide a characterization of the clusters, but they can also be used for further analysis." Finally, the PAM approach provides a "timid robustification" of the c-means clustering as shown in García-Escudero and Gordaliza (1999) and García-Escudero et al. (2010).

Density function transformation via nonparametric kernel estimation: By transforming large-scale time series into density functions, we reduce the complexity of the empirical information represented by the large-scale time series and allow for different time series to have different sample sizes and different observation frequencies. It also enables us to

adopt a more parsimonious theoretical approach for analyzing the data. See, for instance, the discussion in Tsay (2016).

3.1 Kernel Estimation

Under the proposed approach, each time series is transformed into a density function, which is easier to handle and allows for time series with different sample sizes or different sampling frequencies. We use nonparametric methods, such as the kernel method, to estimate the marginal probability density function of a stationary time series. Limiting properties of the kernel density estimate for weakly dependent data are available in Hardle et al. (1989).

Let $\{f_t(x)\}$ be a sequence of probability density functions defined on a common domain $[\delta_1, \delta_2]$ on the real line, for $t = 1, \ldots, T$. For each density function $\{f_t(x)\}$, a random sample $\{x_{t1}, \ldots, x_{t,n_t}\}$ is observed with n_t sufficiently large. Adopting a nonparametric approach, we consider the kernel smoothing method to estimate the probability density function. Given the random sample $\{x_{t1}, \ldots, x_{t,n_t}\}$, the kernel estimator of $f_t(x)$ is as follows:

$$\hat{f}_t(x) = \frac{1}{n_t h_t} \sum_{i=1}^{n_t} K\Big(\frac{x - x_{ti}}{h_t}\Big), \tag{1}$$

where K is the Gaussian kernel, h_t is the bandwidth and $x \in [\delta_1, \delta_2]$. The bandwidth is set to 0.9 times the minimum of the standard deviation and the interquartile range divided by 1.34 times the sample size to the negative one-fifth power. The Gaussian kernel and the bandwidth h_t satisfy the conditions for consistent density estimation; see Silverman (1986). In this paper, each sample consists of realizations of a time series indexed by t, and the $\hat{f}_t(x)$ are estimated at a fixed sequence of points $\{x_1, \ldots, x_N\} \in [\delta_1, \delta_2]$. We use $N = 512$ equally spaced points in our study. For further information on nonparametric density estimation of time series data, see Gyorfi et al. (2014).

3.2 Hellinger Distance Between Time Series

A commonly used measure to quantify the difference between two continuous densities $f_s(x)$ and $f_t(x)$ is the Hellinger distance, here taken in its squared form:

$$h^2(f_s, f_t) = \frac{1}{2} \int_{\delta_1}^{\delta_2} \Big(\sqrt{f_s(x)} - \sqrt{f_t(x)}\Big)^2 dx = 1 - \int_{\delta_1}^{\delta_2} \sqrt{f_s(x) f_t(x)} dx. \tag{2}$$

By its definition, the Hellinger distance is a metric satisfying the triangle inequality. The term $\frac{1}{\sqrt{2}}$ in Eq. 2 ensures that $h(f_s, f_t) \leq 1$ for all probability distributions. Moreover, $0 \leq h(f_s, f_t) \leq 1$ and $h(f_s, f_t) = 0$ if and only if $f_s(x) = f_t(x)$. The sample version of the Hellinger distance when $\hat{f}_t(x)$ and $\hat{f}_s(x)$ evaluated at the points $\{x_1, \ldots, x_N\}$ is as follows:

$$h^2(\hat{f}_s, \hat{f}_t) = 1 - \sum_{i=2}^{N} \sqrt{\hat{f}_s(x_i) \hat{f}_t(x_i)} \times (x_i - x_{i-1}), \tag{3}$$

for a large N.

3.3 Fuzzy Partitioning Around Centroids

The proposed fuzzy partitioning around centroids based on Hellinger distance between kernel estimates (FPAC-HD-KE) is a fuzzy clustering model using centroids to summarize the features of the respective clusters (Bezdek, 1981). Each centroid represents its cluster synthetically as a weighted average of a set of features observed on the objects assigned to the cluster. The proposed FPAC-HD-KE method can be formalized in the following way:

$$\begin{cases} \min: \sum_{i=1}^{T}\sum_{c=1}^{C} u_{ic}^{m} h^2(\hat{f}_i, \hat{g}_c) = \\ \sum_{i=1}^{T}\sum_{c=1}^{C} u_{ic}^{m}\left[1 - \Delta \sum_{j=2}^{N} \sqrt{\hat{f}_i(x_j)\hat{g}_c(x_j)}\right], \\ \text{with the constraints:} \\ \sum_{c=1}^{C} u_{ic} = 1, \quad u_{ic} \geq 0, \\ \sum_{j=1}^{N} \hat{g}_c(x_j) = 1, \quad \hat{g}_c(x_j) \geq 0, \end{cases} \tag{4}$$

where $\Delta = x_2 - x_1$ is the increment of the sequence $\{x_1, \ldots, x_N\} \in [\delta_1, \delta_2]$, C is the number of clusters, m the fuzziness parameter, and T the number of observed time series. In Eq. 4, $\hat{g}_c(x)$ is the centroid density function of the c-th cluster.

The optimal solutions are as follows:

$$u_{ic} = \frac{\left[1 - \Delta \sum_{j=2}^{N} \sqrt{\hat{f}_i(x_j)\hat{g}_c(x_j)}\right]^{-\frac{1}{m-1}}}{\sum_{c'=1}^{C}\left[1 - \Delta \sum_{j=2}^{N} \sqrt{\hat{f}_i(x_j)\hat{g}_{c'}(x_j)}\right]^{-\frac{1}{m-1}}}, \tag{5}$$

$$\hat{g}_c(x_j) = \frac{\left[\sum_{i=1}^{T} u_{ic}^{m} \sqrt{\hat{f}_i(x_j)}\right]^2}{\sum_{v=1}^{N}\left[\sum_{i=1}^{T} u_{ic}^{m} \sqrt{\hat{f}_i(x_v)}\right]^2} = c\text{-th centroid estimated density function.} \tag{6}$$

The proof is provided in the corresponding Appendix 1. For m=1, we obtain the H(ard)PAM-HD-KE.

3.4 Fuzzy Partitioning Around Medoids

There are real cases in which it is more suitable to identify prototypes belonging to the data set under study to better synthesize the structural information of each cluster. Those prototypes are called medoids. Several clustering techniques based on medoids have been proposed in the literature, e.g., the partitioning around medoids (PAM) proposed by Kaufman and Rousseeuw (1990). In a fuzzy framework, Krishnapuram et al. (1999, 2001) suggested the so-called fuzzy C-medoids clustering method. The proposed FPAM based on the Hellinger

distance of the kernel estimate (FPAM-HD-KE) method for clustering density functions is given below:

$$\begin{cases} \min: \sum_{i=1}^{T}\sum_{c=1}^{C} u_{ic}^{m} h^2(\hat{f}_i, \tilde{g}_c) = \\ \quad \sum_{i=1}^{T}\sum_{c=1}^{C} u_{ic}^{m}\left[1 - \Delta \sum_{j=2}^{N} \sqrt{\hat{f}_i(x_j)\tilde{g}_c(x_j)}\right], \\ \quad \text{with the constraints:} \\ \quad \sum_{i=1}^{C} u_{ic} = 1, \quad u_{ic} \geq 0, \end{cases} \tag{7}$$

where m, again, is the fuzzy parameter, and $\tilde{g}_c(x)$ is the medoid of the cth cluster.

The optimal solutions are as follows:

$$u_{ic} = \frac{1}{\sum_{c'=1}^{C} \left\{ \frac{1-\Delta \sum_{j=2}^{N} \sqrt{\hat{f}_i(x_j)\tilde{g}_c(x_j)}}{1-\Delta \sum_{j=2}^{N} \sqrt{\hat{f}_i(x_j)\tilde{g}_{c'}(x_j)}} \right\}^{\frac{1}{m-1}}}, \tag{8}$$

$\tilde{g}_c(x_j) = c$-th medoid estimated density function.

The proof for obtaining u_{ic} is similar to that for FPAC-HD-KE (see Section 3.3). For the selection of the medoids, see Krishnapuram et al. (2001). For m=1, we obtain the H(ard)PAM-HD-KE.

3.5 Algorithms and Some Remarks

In this section, we provide the algorithms for the proposed FPAC-HD-KE and FPAM-HD-KE clustering methods:

Algorithm 1 Fuzzy partitioning around centroids (FPAC) based on Hellinger distance between kernel estimations (FPAC-HD-KE) algorithm.

1: Fix C and $max.iter$;
2: Set $iter = 0$;
3: Initialize U;
4: Set initial centroids: $\{\hat{g}_c^{(0)}(x)\}_{c=1,\ldots,C}$ (e.g., by selecting C densities among $\{\hat{f}_i(x)\}$)
5: **repeat**
6: Store the current centroids $\hat{g}_c^{\text{OLD}}(x) = \hat{g}_c(x)$, $c = 1, \ldots, C$;
7: Compute $\mathbf{u}_i$ $(i = 1, \ldots, T)$ by using (5);
8: Compute the new centroids $\hat{g}_c(x)$ for all c using (6), i.e. compute its values $\hat{g}_c(x_j)$ on the grid
9: $iter \leftarrow iter + 1$;
10: **until** $\hat{g}_c^{\text{OLD}}(x) = \hat{g}_c(x)$ for all c or $iter = max.iter$

Some remarks on the proposed methods and algorithms are in order.

Remark 1 **Addressing Multiple Sources of Uncertainty in Clustering:** In this work, we introduce a novel clustering methodology that simultaneously addresses two fundamental sources of uncertainty commonly encountered in the analysis of complex datasets. On one hand, we account for the probabilistic uncertainty stemming from the inherently stochastic

Algorithm 2 Fuzzy partitioning around medoids (FPAM) based on Hellinger distance between kernel estimations (FPAM-HD-KE) algorithm.

Fix C and $max.iter$;
Set $iter = 0$;
Initialize U;
Pick initial medoids: $\tilde{g}_c(x) = \hat{f}_{t_c}(x)$, $c = 1, \ldots, C$;
repeat
  Store the current medoids $\tilde{g}_c^{\text{OLD}}(x) = \tilde{g}_c(x)$, $c = 1, \ldots, C$;
  Compute $\mathbf{u}_i$ $(i = 1, \ldots, T)$ by using (8);
  Select the new medoids: $\hat{f}_{t_c}(x)$, $c = 1, \ldots, C$;
  for $c = 1$ to C **do**

$$q = \arg\min_{1 \le i' \le T} \sum_{i''=1}^{T} u_{i''c}^{m} \left[1 - \Delta \sum_{j=2}^{N} \sqrt{\hat{f}_{t_i}(x_j)\, \hat{f}_{t_{i''}}(x_j)} \right]$$

    return $\tilde{g}_c(x) = \hat{f}_q(x)$
  end for
  $iter \leftarrow iter + 1$;
until $\tilde{g}_c^{\text{OLD}}(x) = \tilde{g}_c(x)$ for all c or $iter = max.iter$

nature of data generation processes. On the other hand, we incorporate fuzzy uncertainty, which emerges from ambiguity in cluster definitions and overlapping group boundaries.

Our model operates within a dual-framework that integrates both probabilistic and fuzzy components to provide a more comprehensive understanding of the structure in time series data. The probabilistic aspect is managed through kernel density estimation, which allows us to work directly with the estimated probability density functions of stationary time series rather than relying on extracted features or fixed parametric models. This nonparametric approach enables us to capture the full shape of the data distribution, which is especially important in real-world applications where distributional assumptions may not hold.

At the same time, our clustering process employs the principles of fuzzy logic, allowing each observation to have degrees of membership across clusters rather than forcing hard assignments. This flexibility is crucial when analyzing large-scale or high-dimensional datasets, such as time series, where clear-cut cluster separations are rarely observed. By modeling partial membership, we can better reflect the inherent vagueness and overlapping nature of real data.

To compare the similarity between density estimates, we utilize the Hellinger distance, a robust and symmetric measure that is particularly well-suited for capturing differences between probability distributions. When combined with the kernel-based framework, this distance metric enhances our model's capacity to detect meaningful structures even in the presence of noise and non-standard distributional shapes. In summary, the proposed clustering model offers a flexible and powerful solution by jointly considering (1) the probabilistic uncertainty related to the random generation of the data (via kernel density estimation) and (2) the fuzzy uncertainty characterizing the classification process itself.

This integrated approach results in more nuanced, interpretable, and realistic clustering outcomes, particularly for challenging contexts such as stationary time series or high-variability data environments.

As a direction for future development, further enhancements could involve exploring alternative paradigms for managing uncertainty in the clustering process. These include possibilistic clustering, shadowed sets-based clustering, rough sets-based clustering, intuitionistic fuzzy clustering, evidential and credibilistic clustering, type-2 fuzzy clustering, neutrosophic clustering, hesitant fuzzy clustering, interval-valued fuzzy clustering, and picture fuzzy clustering (D'Urso, 2017). Incorporating such models would allow for even greater

expressiveness in handling diverse forms and sources of uncertainty, further strengthening the adaptability and interpretability of clustering in complex real-world applications.

Remark 2 **Inizialitazion, Local Optimum, and Stopping Rules:** The iterative algorithms for FPAC-HD-KE and FPAM-HD-KE do not guarantee the attainment of the global minimum, but they converge to, at least, a local optimum. To limit the risk of finding only the local optima and to check the stability of the solution, more than one random start is recommended. Different stopping rules have been suggested in Webb and Copsey (2011). For instance, a stopping rule considers the comparison between the successive centroids/medoids, respectively, for FPAC-HD-KE and FPAM-HD-KE. An alternative stopping criterion, which is based on suitable properties of the optimization problem connected to fuzzy PAC and fuzzy PAM, has been proposed by Ismail and Selim (1986).

Remark 3 **The Fuzziness Parameter:** The fuzziness parameter m, chosen in advance, plays an important role in the FcM clustering. A discussion on possible procedures for selecting m can be found in D'Urso (2015). For the PAC approach, Pal and Bezdek (1995) gave some heuristic guidelines regarding the best choice of m, suggesting that the level of fuzziness should be between 1.5 and 2.5. For the PAM approach, since the medoid always has a membership in one of the clusters, raising its membership to the power of m has no effect on the medoid, while all other memberships decrease to 0. Thus, when m is high, the mobility of the medoids from iteration to iteration may be lost. For this reason, a value of m between 1.0 and 1.5 is recommended (Kamdar & Joshi, 2000). In this paper, we use $m = 1.5$ in the comparison of the two proposed methods.

Remark 4 **Determining the Optimal Number of Clusters:** To determine the number of clusters, we consider the Fuzzy Silhouette (FS) index (Campello & Hruschka, 2006), which is a weighted average of individual silhouettes width, λ_i (Kaufman & Rousseeuw, 1990), with weights derived from the fuzzy membership matrix $\mathbf{U} = \{u_{ic} : i = 1, \ldots, T; c = 1, \ldots, C\}$:

$$\text{FS} = \frac{\sum_{i=1}^{T}(u_{ip} - u_{iq})^{\alpha} \cdot \lambda_i}{\sum_{i=1}^{T}(u_{ip} - u_{iq})^{\alpha}}, \qquad \lambda_i = \frac{(b_i - a_i)}{\max\{b_i, a_i\}}, \tag{9}$$

where a_i is the average distance between the i-th unit and the units belonging to the cluster p ($p = 1,...,C$) with which i is associated with the highest membership degree; b_i is the minimum (over clusters) average distance of the i-th unit to all units belonging to the cluster q with $q \neq p$; $(u_{ip} - u_{iq})^{\alpha}$ is the weight of each λ_i calculated upon $\mathbf{U}$, where p and q are, respectively, the first and second best clusters (accordingly to the membership degree) to which the i-th unit is associated; $\alpha \geq 0$ is an optional user defined weighting coefficient. The traditional Silhouette coefficients are obtained by setting $\alpha = 0$. The higher the value of FS, the better the assignment of the units to the clusters, simultaneously obtaining the minimization of the intra-cluster distance and the maximization of the inter-cluster distance. The optimal number of clusters is identified by the highest value of FS.

Remark 5 **Comparison of Partitions:** The Fuzzy Rand Index (FRI, Hüllermeier et al., 2012; Anderson et al., 2010) is adopted to compare different partitions and/or to compare a given partition with a reference one. The FRI ranges from 0 (total disagreement) to 1 (complete agreement) and is a fuzzy extension of the Rand index (RI, Campello, 2007), based on agreements and disagreements in two partitions. The RI is based on the contingency matrix associated with two crisp partitions P and Q, defined as $R = P^T Q$. If P consists of k clusters and Q consists of l clusters, then $R = [r_{i,j}]$ is a $k \times l$-matrix. In the non-fuzzy case,

the entry $r_{i,j}$ corresponds to the number of objects that belong to the i-th cluster in P and to the j-th cluster in Q. In particular:

$$\mathrm{RI} = \frac{a+d}{a+b+c+d}, \tag{10}$$

a: number of pairs of data objects belonging to the same class in P and to the same cluster in Q
b: number of pairs of data objects belonging to the same class in P, but to different clusters in Q
c: number of pairs of data objects belonging to different classes in P, but to the same cluster in Q
d: number of pairs of data objects belonging to different classes in P and to different clusters in Q

The expression (10) can also be used in the case of fuzzy partitions. This is precisely the idea of Anderson et al. (2010), where P and Q are the matrices $units \times clusters$ of the fuzzy memberships ranging in [0, 1], and summing up 1 by row.

Remark 6 **Computational Complexity and Scalability:** The time complexity of the proposed algorithms is evaluated using a unit cost for all operations, and it depends on three parameters: the number of time series T, the number of kernel estimated values of the density functions N, and the number of clusters C. As the complexity is always linear with respect to N and C, we consider the complexity with respect to T for each iteration of the algorithm. The complexity of FPAM-HD-KE is $\mathrm{O}(T^2)$, whereas the complexity of FPAC-HD-KE is $\mathrm{O}(T)$. In fact, the FPAM-HD-KE algorithm operates on each time series twice (nested cycle) as the prototype is the medoid, while the FPAC-HD-KE algorithm operates on each time series once, as the prototype is the mean. To lower the complexity of the proposed model, Step 8 in the algorithm FPAM-HD-KE can be modified so that only a subset of the time series is examined while updating the medoid for cluster c (Linearization) (Krishnapuram et al., 2001). The subset used is the set of l time series that correspond to the top l highest membership values in cluster c. The subsets for each cluster can be identified during the updating step. The complexity of the FPAM-HD-KE algorithm is reduced to $\mathrm{O}(T)$. The value of l should be proportional to the dimensionality T of the time series. The FPAM-HD-KE algorithm and the Linearized-PAM-HD-KE algorithm have been run on simulated data using $C = 2$ and $C = 5$ clusters (generative distribution functions), the value $m = 1.5$ of the power of the fuzziness degree, $N = 1000$, ten values of T (100 to 1000 step 100), $l = (0.25 * N/C)$. The results are shown in Fig. 2.

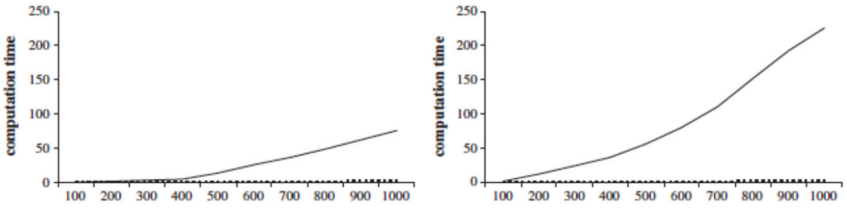


Fig. 2 Computation time of FPAM-HD-KE (solid line) and Linearized-FPAM-HD-KE (dashed line) as a function of N

4 Simulation Study

This section presents three simulation studies comparing FPAC-HD-KE and FPAM-HD-KE with other competing methods, whereas the supplementary material includes an additional simulation study that compares FPAC-HD-KE and FPAM-HD-KE with their crisp counterparts across three further experimental setups.

4.1 Simulation Setup and Competing Methods

For each study, we apply the methods FPAC-HD-KE and FPAM-HD-KE. The density functions are estimated using a Gaussian kernel, and the Hellinger distance is computed via the R package `topicmodels` (Grün & Hornik, 2011). To facilitate comparisons, as recommended by the reviewer, we also estimate the density functions using the Epanechnikov and Triangular kernels. However, as the simulation results demonstrate, no significant differences in performance are observed between these kernel choices.

For further comparisons, we incorporate FPAM-HD-KE with predefined dissimilarities. Specifically, we employ dissimilarity based on autocorrelations (d_{ACF}) and partial autocorrelations (d_{PACF}) (Galeano & Peña, 2000), as well as based on auto- and cross-correlations (d_{DM}) proposed by D'Urso and Maharaj (2009). Additionally, we include the dissimilarity based on maximum overlap discrete wavelet transform (d_{DWT}, D'Urso & Maharaj, 2012) and model-based ones, such as the Piccolo distance (d_{PIC}, Piccolo, 1990) and that derived from Maharaj's hypothesis-testing approach (d_{MAH} Maharaj, 1996; 2000). These dissimilarities are implemented in the R packages `TSclust` (Montero & Vilar, 2015) and `mlmts` (Vilar & Vilar, 2021).

The first simulation demonstrates the effectiveness of the methods in correctly classifying time series generated by different AR models of length $T = 3000$. Specifically, the first 30 time series are generated from an AR(1) model with $\phi = 0.7$. The next 20 time series are generated from an AR(1) model with $\phi = -0.5$, and the final 30 time series come from an AR(1) model with $\phi = -0.8$. These different AR processes result in distinct time series behaviors. Additionally, one further time series is introduced exhibiting a structural change, switching from the first AR process to the second at the midpoint. Figure 8, in the Appendix 2, shows both the series (a) and the corresponding Gaussian kernel densities (b). The second simulation extends the first by introducing "bumps" into certain time series at specific points. For six time series (units 1, 10, 40, 50, 65, and 70), 100 random time points are selected. A large value—three times the maximum value of the series for units 1, 40, and 65, and three times the minimum for the others—is added to the corresponding data points. This causes significant jumps in the values of these series at the selected indices. Adding large shifts at random points is useful for simulating realistic scenarios where time series undergo abrupt changes and are particularly valuable for testing clustering methods, as the presence of such shifts may influence the performance of clustering algorithms. The noise time series are presented in Fig. 9a while the Gaussian kernel density functions for all series are in b.

The third simulation increases the length of the time series to $T = 5000$ and enlarges the group sizes, considering two clusters of sizes 60 and 70, respectively. The time series in the first group are generated using an ARMA(1,1) model with parameters $\phi = 0.6$, $\theta = 0.4$, and $\sigma = 10$, while the second group is generated from an ARMA(1,1) model with $\phi = 0.65$, $\theta = 0.5$, and $\sigma = 10$. Additionally, an extra time series is introduced, created as a mixture of the two ARMA processes with a mixing proportion of 0.5. For four specific time series (units 1, 20, 90, and 110), 100 random time points are selected, and large values—either twice the maximum or minimum value of the series, chosen randomly—are added to the

corresponding data points. The time series are presented in Fig. 10a while the noise ones in b. The Gaussian kernel density functions for all series are in c. This simulation increases both the length of the time series and the size of the clusters, while simultaneously reducing the separation between clusters and introducing noise. The setup is specifically designed to challenge the methods by evaluating their performance under more complex conditions, such as less distinct clustering boundaries.

To assess the performance of different methods, we used the Fuzzy Adjusted Rand Index, alongside its crisp counterpart. In the latter case, the comparison is made between the ground truth and the crisp partition derived from the fuzzy partition by assigning units to cluster with the highest membership. The results presented are averages over 100 repetitions.

4.1.1 Simulation 1

As already described, there are 80 time series grouped in three clusters of size 30, 20, and 30, respectively, and one switching time series. We ran the algorithms with 20 random restarts, fixing $C = 3$, and considered two levels of fuzziness: $m = 1.3$ and $m = 1.5$. In Table 6, the results are shown in terms of the Adjusted Rand Index and the Fuzzy Adjusted Rand Index over 100 simulated datasets. The last column reports the execution time for one randomly selected dataset.

The results indicate that the variation between $m = 1.3$ and $m = 1.5$ is minimal for most methods. When evaluating the trade-offs between clustering quality and computational cost, kernel-based methods as well as those based on d_{DM} and d_{DWT} stand out as very fast while still achieving high performance in terms of Rand indices. This makes them highly efficient choices, especially when handling large datasets. Notably, no significant differences in performance are observed among the various kernel options. From Table 7, we can compare the membership degrees between FPAM-HD-KE and FPAC-HD-KE (with $m = 1.5$ and a Gaussian kernel). Both methods successfully identify the last time series as a switching series while accurately assigning the remaining time series to their respective clusters with high membership values.

4.1.2 Simulation 2

In this second simulation, the only difference is the inclusion of noise in specific time series (units 1, 10, 40, 50, 65, and 70). As already described, this added noise introduces significant jumps in the values of these series, leading to noticeable fluctuations. We ran the algorithms with 20 random restarts, fixing $C = 3$, and considered two levels of fuzziness: $m = 1.3$ and $m = 1.5$. Looking at Table 8, the FARI values decrease as the fuzziness parameter m goes from 1.3 to 1.5.

This reflects the expected reduction in clustering membership for higher fuzziness. However, the high ARI values indicate that the classifications remain stable. This highlights the importance of considering both FARI and ARI when evaluating clustering performance. Notably, the best performance, according to the ARI, remains unchanged at 0.998 for the proposed methods using kernel-based distance measures (Gaussian, Epanechnikov, and Triangular), even as m increases. This confirms that the classifications remain accurate despite fuzzier partitions. Among these, the FPAM-HD-KE methods, regardless of the kernel used, show the best balance between accuracy and execution time. The d_{DM}-based method is the fastest, but this comes at the cost of reduced ARI and FARI values compared to the kernel-based methods. Conversely, the other competitors achieve higher FARI values only

at $m = 1.5$, but at a significantly higher computational cost. However, their ARI values are always lower than those of our proposed methods, reinforcing the superior performance of our approach. To illustrate this, we present in Table 9 the fuzzy partitions obtained using the FPAM-HD-KE method with different dissimilarity measures: h^2 (Gaussian kernel), d_{ACF}, d_{DWT}, and d_{PIC}.[2] As observed, for the noisy time series, our method correctly assigns the highest membership to the appropriate cluster, while the other methods tend to assign high membership to the incorrect group in such cases. This discrepancy is reflected in the lower ARI index for the other methods.

4.1.3 Simulation 3

As detailed in the introduction, the third simulation tested the performance of the methods when both the length of the time series and the size of the units within the groups increased, along with a scenario in which the groups became less separated and included some noise. We ran the algorithms with 20 random restarts, fixing $C = 2$, and considered two levels of fuzziness: $m = 1.3$ and $m = 1.5$. From Table 10, the same conclusions as in the second scenario can be drawn: the kernel-based methods once again exhibit significantly higher ARI values compared to their competitors, while maintaining consistently lower computational effort. For reasons of space, we omit the same comparisons among the partitions as shown in Table 9, since the conclusions are consistent with those drawn from the second simulation.

With the only scope to show the gain in using a fuzzy approach, the crisp models PAC-HD-KE and PAM-HD-KE are also used ($m = 1.0$) in comparison with their fuzzy versions. Results are shown in the supplementary material.

5 Applications

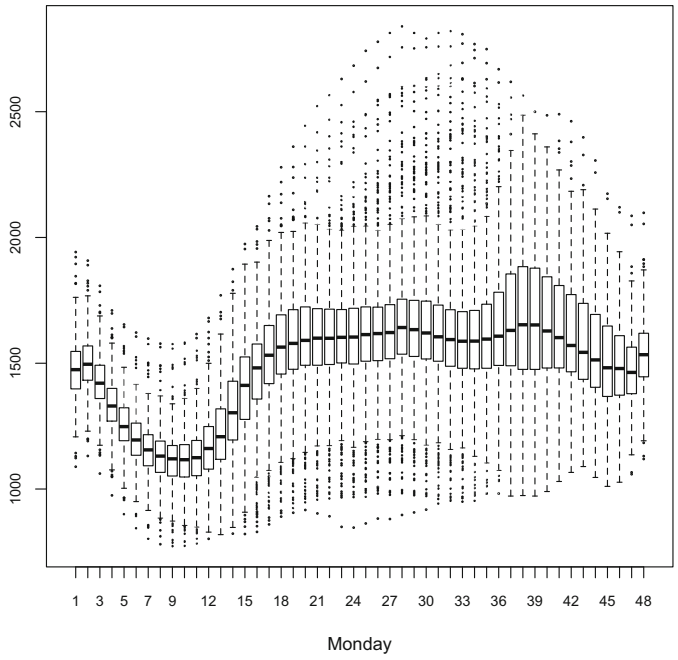
5.1 Electricity Demand Time Series in Australia

Consider the demands of electricity in half-hour intervals from Sunday to Saturday, in Adelaide, Australia, from July 6, 1997, to March 31, 2007. The data set is available in Shang and Hyndman (2018), which consists of 48 30-min interval observations (1=0:00–0:30; 48=23:30–24:00) of the demand of electricity in each day, 7 days a week, and 508 weeks. The electricity demands exhibit daily, weekly, and annual patterns. In Fig. 3, for Monday and Saturday, the boxplots of the demands for each 30-min interval are presented (other days in the Supplementary Material). The demands are relatively low when most people are asleep in the early morning hours, say from 2:00 to 6:00 a.m. The demands of electricity increase during the day.

The demand density functions of a given interval for each day were estimated by the conventional Gaussian kernel smoothing method with 508 observations using the package `topicmodels`. A common domain from 950 (700) to 2950 (2900) megawatts is used, and the density functions were evaluated at 512 equally spaced points in the common domain of the megawatts across the days. The number of clusters is chosen using the FS value. Based on the results in Table 1, the optimal number of clusters is 6 for Monday to Friday, but 7 for Saturday and Sunday. The comparison with the crisp clustering is also presented ($m = 1.0$). The results show that the fuzzy clustering improves over the crisp clustering.

The results of the method FPAC-HD-KE are shown in Table 2. The comparison, limited to Monday only, with the results in Tsay (2016) with a PAC-HD-KE method is shown in

[2] The other FPAM-HD-KE variants based on alternative dissimilarities yield similar results.



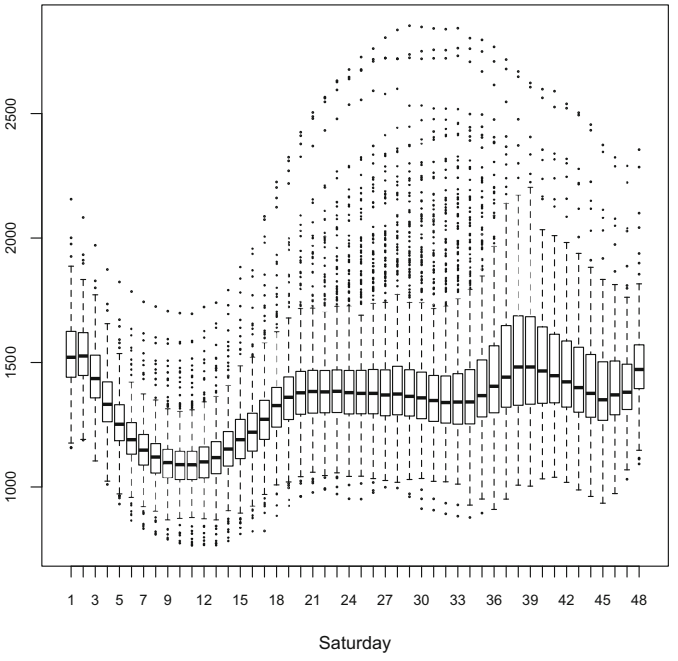


Fig. 3 Boxplot of electricity demand—Monday and Saturday

Table 1 FS for C=5, 6, 7

	m=1.0	m=1.5		
	C=6	C=5	C=6	C=7
Monday	0.45	0.56	0.61	0.56
Tuesday	0.46	0.61	0.64	0.63
Wednesday	0.42	0.51	0.52	0.50
Thursday	0.52	0.64	0.69	0.57
Friday	0.48	0.61	0.62	0.56
Saturday	0.45	0.55	0.51	0.68
Sunday	0.48	0.52	0.54	0.60

Table 2 Fuzzy membership, u highest membership; c cluster

Time	Monday		Tuesday		Wednesday		Thursday		Friday		Saturday		Sunday	
	u	c	u	c	u	c	u	c	u	c	u	c	u	c
1	0.79	4	0.72	6	0.54	6	0.43	5	0.31	1	0.82	2	0.88	7
2	0.59	4	0.58	6	0.46	6	0.39	5	0.30	1	0.76	2	0.75	7
3	0.54	4	0.37	4	0.45	4	0.47	2	0.41	6	0.34	2	0.37	7
4	0.44	3	0.40	3	0.39	3	0.42	1	0.35	5	0.30	5	0.28	2
5	0.72	3	0.78	3	0.77	3	0.78	1	0.77	5	0.46	6	0.53	1
6	0.46	1	0.44	3	0.44	3	0.43	1	0.44	5	0.83	6	0.60	4
7	0.88	1	0.84	1	0.85	1	0.87	4	0.85	4	0.49	3	0.74	4
8	0.95	1	0.95	1	0.95	1	0.96	4	0.95	4	0.93	3	0.61	5
9	0.90	1	0.91	1	0.91	1	0.91	4	0.91	4	0.95	3	0.94	5
10	0.92	1	0.92	1	0.92	1	0.92	4	0.92	4	0.91	3	0.96	5
11	0.97	1	0.96	1	0.97	1	0.97	4	0.96	4	0.91	3	0.92	5
12	0.73	1	0.74	1	0.74	1	0.75	4	0.77	4	0.97	3	0.92	5
13	0.72	3	0.70	3	0.71	3	0.69	1	0.66	5	0.92	3	0.92	5
14	0.60	3	0.54	3	0.54	3	0.56	1	0.56	5	0.44	6	0.97	5
15	0.33	2	0.56	4	0.57	4	0.55	2	0.61	6	0.94	6	0.86	5
16	0.43	2	0.66	4	0.64	4	0.66	2	0.55	6	0.86	6	0.63	5
17	0.43	2	0.37	6	0.64	6	0.75	5	0.38	1	0.29	6	0.78	4
18	0.41	6	0.40	2	0.30	5	0.34	5	0.41	2	0.29	5	0.74	4
19	0.50	6	0.45	2	0.37	5	0.34	3	0.50	2	0.42	5	0.44	1
20	0.58	6	0.45	2	0.45	5	0.46	3	0.59	2	0.44	5	0.72	1
21	0.67	6	0.52	5	0.53	5	0.62	3	0.69	2	0.41	1	0.87	1
22	0.73	6	0.65	5	0.65	5	0.75	3	0.76	2	0.44	1	0.79	1
23	0.77	6	0.75	5	0.76	5	0.81	3	0.80	2	0.45	1	0.60	1
24	0.82	6	0.79	5	0.84	5	0.83	3	0.83	2	0.43	1	0.57	3
25	0.83	6	0.79	5	0.86	5	0.85	3	0.76	2	0.38	1	0.67	3
26	0.84	6	0.81	5	0.87	5	0.85	3	0.72	2	0.38	7	0.75	3
27	0.81	6	0.81	5	0.85	5	0.83	3	0.73	2	0.50	7	0.83	3
28	0.69	6	0.72	5	0.73	5	0.70	3	0.69	2	0.49	7	0.84	3
29	0.74	6	0.76	5	0.76	5	0.74	3	0.71	2	0.71	7	0.93	3
30	0.78	6	0.80	5	0.78	5	0.79	3	0.68	2	0.79	7	0.90	3
31	0.80	6	0.84	5	0.82	5	0.83	3	0.50	2	0.74	7	0.85	3
32	0.73	6	0.78	5	0.75	5	0.74	3	0.52	1	0.67	7	0.82	3
33	0.68	6	0.72	5	0.70	5	0.64	3	0.55	1	0.67	7	0.82	3
34	0.66	6	0.69	5	0.67	5	0.59	3	0.56	1	0.69	7	0.81	3

Table 2 continued

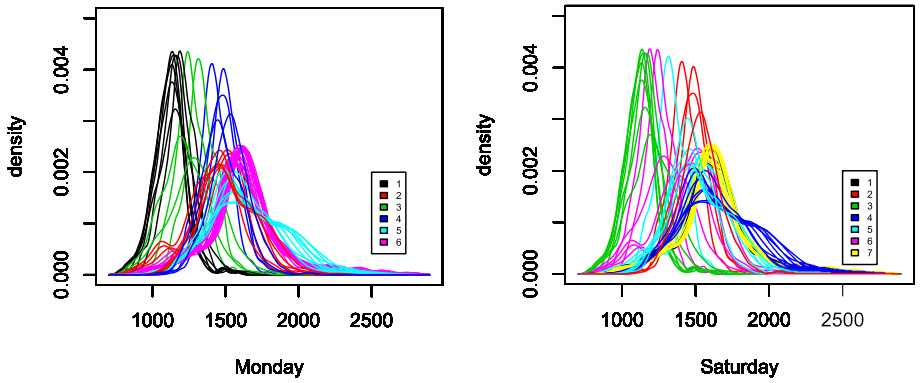
Time	Monday		Tuesday		Wednesday		Thursday		Friday		Saturday		Sunday	
	u	c	u	c	u	c	u	c	u	c	u	c	u	c
35	0.68	6	0.77	5	0.74	5	0.61	3	0.59	1	0.56	7	0.53	3
36	0.61	5	0.47	5	0.52	2	0.48	6	0.46	3	0.40	4	0.36	2
37	0.69	5	0.63	2	0.68	2	0.77	6	0.64	3	0.73	4	0.73	6
38	0.63	5	0.59	2	0.63	2	0.70	6	0.57	3	0.64	4	0.82	6
39	0.67	5	0.63	2	0.66	2	0.72	6	0.61	3	0.71	4	0.85	6
40	0.77	5	0.75	2	0.77	2	0.79	6	0.73	3	0.83	4	0.87	6
41	0.81	5	0.85	2	0.85	2	0.85	6	0.87	3	0.81	4	0.83	6
42	0.74	5	0.81	2	0.82	2	0.83	6	0.84	3	0.69	4	0.87	6
43	0.52	5	0.55	2	0.56	2	0.54	6	0.55	3	0.42	4	0.77	6
44	0.46	2	0.32	4	0.30	2	0.38	5	0.44	6	0.34	5	0.49	2
45	0.72	2	0.65	4	0.46	4	0.45	2	0.68	6	0.41	5	0.84	2
46	0.56	2	0.79	4	0.67	4	0.66	2	0.83	6	0.44	5	0.88	2
47	0.88	4	0.5	4	0.69	4	0.73	2	0.77	6	0.36	5	0.59	2
48	0.36	4	0.78	6	0.72	6	0.55	5	0.32	6	0.80	2	0.89	7

Table 3. The centroids for $m = 1.5$ are presented in Fig. 4 (Monday and Saturday), where different colors represent different clusters (other days in the Supplementary Material).

Table 3 shows the perfect matching of the clustering for Monday. Table 2 underlines that 30-min intervals 4, 6, 15–18, 44, and 48 have a membership lower than 0.5. As shown in Table 4 and clear from Fig. 3 and Table 3, time interval 4 has the second highest membership to cluster 4; time interval 6 to cluster 3—these time intervals show the second highest membership to clusters with bordering and close electricity demand time intervals; time intervals 15 and 16 to cluster 4; time interval 17 to cluster 6; time interval 18 to clusters 2 and 5; and time interval 44 to cluster 5 and time interval 48 to cluster 2—these time intervals show the second highest membership to clusters with no bordering and close electricity demand time

Table 3 Clustering electricity demands of Monday intervals. Tsay (2016) (top); FPAC-HD-KE (bottom)—C=6

cluster	Elements	Size	Calendar hours
1	36 to 43	8	5:30 $p.m.$ to 9:30 $p.m.$
2	15 to 17, 44 to 46	6	7:00 to 8:30 $a.m.$; 9:30 $p.m.$ to 11:00 $p.m.$
3	1 to 3, 47, 48	5	11:00 $p.m.$ to 1:30 $a.m.$
4	18 to 35	18	8:30 $a.m.$ to 5:30 $p.m.$
5	4, 5, 13, 14	4	1:30 to 2:30 $a.m.$; 6:00 to 7:00 $a.m.$
6	6 to 12	7	2:30 to 6:00 $a.m.$
1	6 to 12	7	2:30 to 6:00 $a.m.$
2	15 to 17, 44 to 46	6	7:00 to 8:30 $a.m.$; 9:30 $p.m.$ to 11:00 $p.m.$
3	4, 5, 13, 14	4	1:30 to 2:30 $a.m.$; 6:00 to 7:00 $a.m.$
4	1 to 3, 47, 48	5	11:00 $p.m.$ to 1:30 $a.m.$
5	36 to 43	8	5:30 $p.m.$ to 9:30 $p.m.$
6	18 to 35	18	8:30 $a.m.$ to 5:30 $p.m.$



(a) Monday (b) Saturday

Fig. 4 Density function of the centroids of the clusters: Monday, Saturday

intervals. The fuzzy nature of the model FPAC-HD-KEC catches the smooth transition in electricity demands.

Considering the demands within the days and analyzing the memberships and the centroids, we see that the resulting clusters match the daily human activities closely: sleeping hours, dinner time, working hours. As expected, the demands for electricity are relatively stable and at their lowest during the sleeping hours.

Considering the demands across the days, we see that Saturday (and Sunday) differ from the other days of the week due to a lower demand of electricity in each time interval of the day and greater variability in the intervals 20 (10:00) to 36 (18:00). This leads to the specification of 7 clusters.

5.2 $PM_{2.5}$ Time Series in Taiwan

Consider the hourly measurements of $PM_{2.5}$ collected by Airboxes at 508 locations in Taiwan (Gao & Tsay, 2020). The locations of the 508 locations are shown in Fig. 5, which mainly consist of three clusters, signifying the major cities of Taiwan, namely Taipei, Taichung, Tainan, and Kaohsiung. The latter two cities are adjacent and form a large cluster.

The isolated location outside of Taiwan denotes part of the Orchid Island of Taiwan. The method FPAM-HD-KE has been applied to the hourly measurements of $PM_{2.5}$ for March 2017 with a total of 744 observations[3]. The minimum observed value of concentration is zero, and the maximum is 176. For kernel estimation, a number of 256 equally spaced points is considered. Using the Fuzzy Silhouette criterion (Table 5), we obtain C=3 and $m = 1.3$, where 189 locations belong to cluster 1, 157 locations to cluster 2, and 162 locations to cluster 3.

The probability density function (pdf) and cumulative distribution function (CDF) of the three medoids are presented in Fig. 6.

[3] The time series are stationary and have been tested using the Augmented Dickey-Fuller (ADF) test.

Table 4 FPAC-HD-KE membership—Monday

	1	2	3	4	5	6
1	0.01	0.11	0.02	0.79	0.04	0.03
2	0.02	0.19	0.03	0.59	0.08	0.08
3	0.03	0.19	0.09	0.54	0.08	0.07
4	0.09	0.14	0.44	0.18	0.08	0.07
5	0.11	0.05	0.72	0.05	0.04	0.03
6	0.46	0.05	0.37	0.04	0.04	0.04
7	0.88	0.01	0.06	0.01	0.01	0.01
8	0.95	0.01	0.02	0.01	0.01	0.01
9	0.90	0.01	0.05	0.01	0.01	0.01
10	0.92	0.01	0.04	0.01	0.01	0.01
11	0.97	0.00	0.02	0.00	0.00	0.00
12	0.73	0.03	0.18	0.02	0.02	0.02
13	0.16	0.04	0.72	0.03	0.03	0.03
14	0.08	0.11	0.60	0.11	0.06	0.06
15	0.04	0.33	0.13	0.30	0.10	0.10
16	0.02	0.43	0.05	0.29	0.10	0.11
17	0.02	0.43	0.03	0.13	0.16	0.22
18	0.01	0.24	0.02	0.08	0.24	0.41
19	0.01	0.16	0.02	0.05	0.26	0.50
20	0.01	0.10	0.01	0.04	0.26	0.58
21	0.01	0.07	0.01	0.03	0.21	0.67
22	0.00	0.06	0.01	0.02	0.17	0.73
23	0.00	0.05	0.01	0.02	0.15	0.77
24	0.00	0.04	0.00	0.02	0.12	0.82
25	0.00	0.03	0.00	0.01	0.11	0.83
26	0.00	0.03	0.00	0.01	0.11	0.84
27	0.00	0.04	0.01	0.02	0.13	0.81
28	0.01	0.06	0.01	0.03	0.21	0.69
29	0.01	0.05	0.01	0.02	0.18	0.74
30	0.00	0.04	0.01	0.02	0.15	0.78
31	0.00	0.04	0.01	0.02	0.14	0.80
32	0.01	0.06	0.01	0.03	0.17	0.73
33	0.01	0.07	0.01	0.03	0.20	0.68
34	0.01	0.07	0.01	0.03	0.22	0.66
35	0.00	0.05	0.01	0.02	0.23	0.68
36	0.00	0.04	0.01	0.02	0.61	0.32
37	0.01	0.06	0.01	0.03	0.69	0.20
38	0.01	0.09	0.02	0.04	0.63	0.21
39	0.01	0.07	0.01	0.04	0.67	0.19

Table 4 continued

	1	2	3	4	5	6
40	0.01	0.05	0.01	0.02	0.77	0.14
41	0.00	0.05	0.01	0.02	0.81	0.11
42	0.00	0.08	0.01	0.03	0.74	0.13
43	0.01	0.21	0.02	0.07	0.52	0.18
44	0.01	0.46	0.02	0.11	0.25	0.15
45	0.01	0.72	0.02	0.12	0.07	0.06
46	0.01	0.56	0.02	0.31	0.06	0.05
47	0.00	0.08	0.01	0.88	0.02	0.02
48	0.01	0.33	0.03	0.36	0.13	0.14

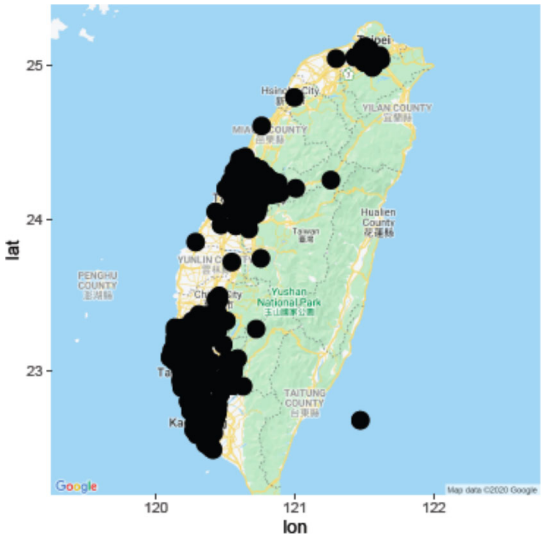


Fig. 5 Locations (latitude vs. longitude) of the 508 AirBoxes across Taiwan

Table 5 Fuzzy Silhouette for $PM_{2.5}$ time series. m is the fuzzy parameter

	m=1.0	m=1.3	m=1.5
3	0.38	0.46	0.41
4	0.25	0.36	0.31
5	0.30	0.42	0.34
6	0.27	0.35	0.30

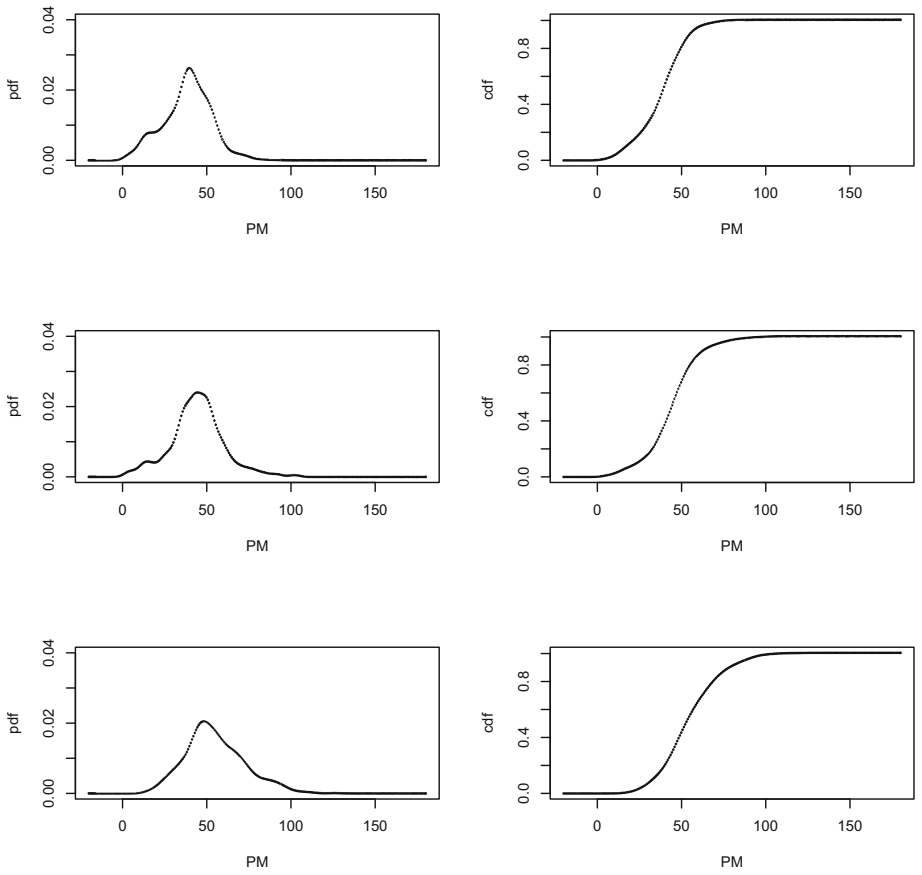


Fig. 6 Empirical cumulative distribution function and probability density function of the medoids 86, 101, and 507 from top to bottom

They show the different shapes and measures of central tendency of the distributional representatives of the three clusters; the medians are 38, 44, and 53 (values of concentration of $PM_{2.5}$). The ternary plot is presented in Fig. 7, showing the fuzziness of the clustering.

6 Conclusions

In this paper, we have introduced an innovative fuzzy clustering framework for large-scale time series, leveraging the synergy between kernel density estimation, the Hellinger distance, and fuzzy partitioning strategies such as FPAC and FPAM. This integration represents a unique methodological contribution that bridges fuzzy clustering and distributional data analysis in the context of big dependent data. We claim that this combination, applied to time series clustering, is not simply an "ordinary" integration; rather, it addresses specific theoretical and practical challenges associated with clustering complex data types, thereby advancing the state of the art in this field.

The fuzzy logic underpinning our approach provides a framework for representing partial membership, which is especially valuable when dealing with ambiguous or overlapping

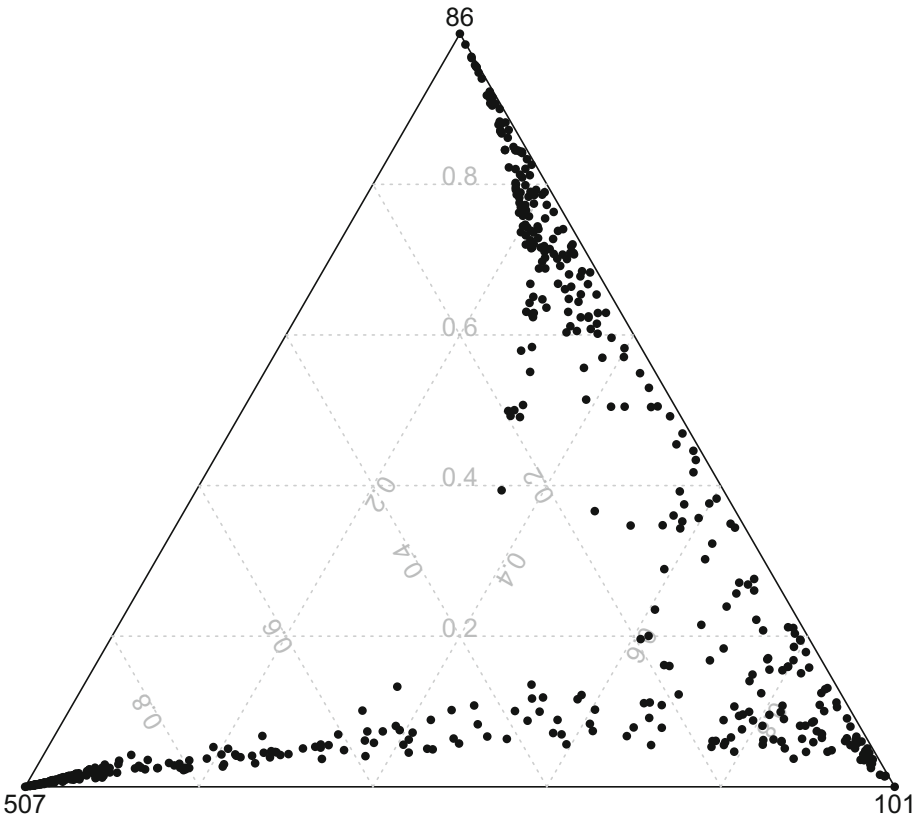


Fig. 7 Ternary plot

cluster boundaries. This becomes particularly important in real-world datasets, such as big time series, where the assignment of a unit to a single cluster is often not clear-cut. A fuzzy approach allows for more nuanced classifications, reflecting the uncertainty inherent in such complex data. Our methods also integrate a probabilistic aspect through kernel density estimation. Unlike traditional clustering, our approach works with estimated density distributions of stationary time series. This allows us to account for the full distribution, rather than relying on particular features or estimated model parameters. Moreover, by combining the Hellinger distance with kernel methods, we can cluster data that may not conform to simple parametric assumptions, thus providing greater flexibility.

Notably, our approach offers a dual advantage: it achieves computational efficiency and scalability while maintaining high accuracy, even in the context of massive time-dependent datasets and in the presence of noise data. Through rigorous simulations and real-world applications, we have demonstrated the robustness and effectiveness of our method. This extension is particularly valuable for addressing the complexities of big data, such as those encountered in big time series analysis, where traditional methods may fall. Looking ahead, future research will focus on refining the representation of dependence structures in large-scale time series and developing robust clustering techniques that can withstand the challenges posed by outliers and heavy-tailed distributions.

Supplementary Material

Section S1 - Additional simulation study. Comparison between FPACHD-KE and FPAM-HD-KE and with their crisp counterparts across three further experimental setups.
Section S2 - Application to Australian weekly electricity demand time series. The analysis of the electricity demands for the seven days of the week.

Appendix 1

A.1 Optimal Solutions of the Fuzzy Partitioning Around Centroids

Proof For determining u_{ic}, we consider the Lagrangian function:

$$L_m(\mathbf{u}_i, \lambda) = \sum_{i=1}^{T}\sum_{c=1}^{C} u_{ic}^m \left[1 - \Delta \sum_{j=2}^{N} \sqrt{\hat{f}_i(x_j)\hat{g}_c(x_j)}\right] - \lambda\left(\sum_{c=1}^{C} u_{ic} - 1\right), \tag{11}$$

where $\mathbf{u}_i = (u_{i1}, \ldots, u_{ic}, \ldots, u_{iC})'$ and λ is the Lagrange multiplier. Therefore, we set the first derivatives of Eq. 11 with respect to u_{ic} and λ equal to zero, yielding the following:

$$\frac{\partial L_m(\mathbf{u}_i, \lambda)}{\partial u_{ic}} = 0 \Leftrightarrow m u_{ic}^{m-1}\left[1 - \Delta \sum_{j=2}^{N} \sqrt{\hat{f}_i(x_j)\hat{g}_c(x_j)}\right] - \lambda = 0, \tag{12}$$

$$\frac{\partial L_m(\mathbf{u}_i, \lambda)}{\partial \lambda} = 0 \Leftrightarrow \sum_{c=1}^{C} u_{ic} - 1 = 0. \tag{13}$$

From Eq. 12, we obtain the following:

$$u_{ic} = \left(\frac{\lambda}{m\left[1 - \Delta \sum_{j=2}^{N} \sqrt{\hat{f}_i(x_j)\hat{g}_c(x_j)}\right]}\right)^{\frac{1}{m-1}} \tag{14}$$

and, by considering (13):

$$\left(\frac{\lambda}{m}\right)^{\frac{1}{m-1}} = \frac{1}{\sum_{c'=1}^{C}\left[\frac{1}{1-\Delta \sum_{j=2}^{N} \sqrt{\hat{f}_i(x_j)\hat{g}_{c'}(x_j)}}\right]^{\frac{1}{m-1}}}. \tag{15}$$

Using (15) in Eq. 14, we obtain (5).

For obtaining $\hat{g}_c(x_j)$, we have the following:

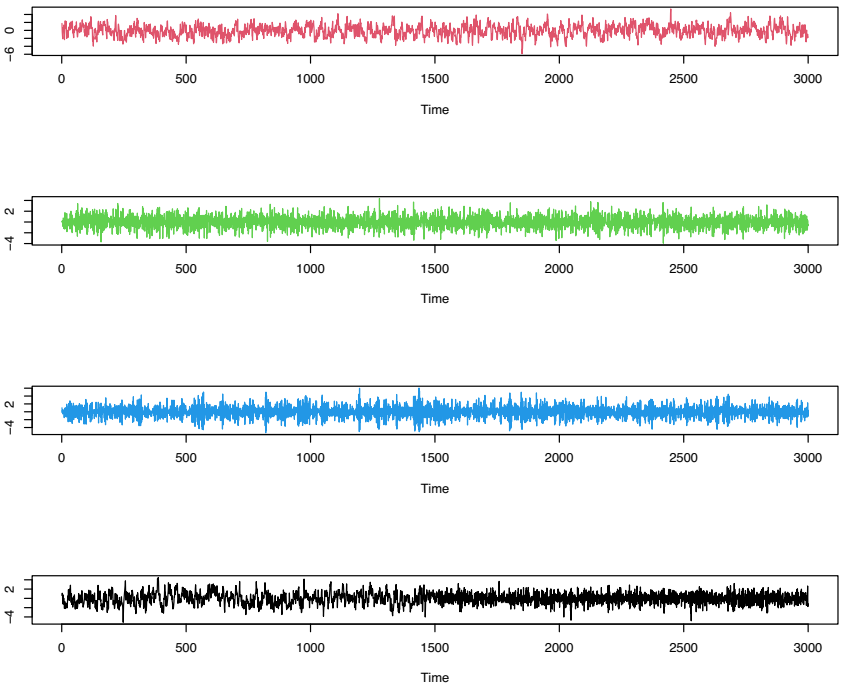
$$\begin{aligned}
&\frac{\partial}{\partial \hat{g}_c(x_j)}\left[\sum_{i=1}^{T} u_{ic}^m\left(1-\Delta\sqrt{\hat{f}_i(x_j)\hat{g}_c(x_j)}\right)-\lambda\left(\sum_{j=1}^{N}\hat{g}_c(x_j)-1\right)\right]=0\\
&\Leftrightarrow -\Delta\sum_{i=1}^{T} u_{ic}^m\sqrt{\hat{f}_i(x_j)}\frac{\partial}{\partial \hat{g}_c(x_j)}\sqrt{\hat{g}_c(x_j)}-\lambda=0\\
&\Leftrightarrow -\frac{\Delta}{2}\sum_{i=1}^{T} u_{ic}^m\sqrt{\hat{f}_i(x_j)}\frac{1}{\sqrt{\hat{g}_c(x_j)}}-\lambda=0\\
&\Leftrightarrow \hat{g}_c(x_j)=\frac{\frac{\Delta^2}{4}\left[\sum_{i=1}^{T} u_{ic}^m\sqrt{\hat{f}_i(x_j)}\right]^2}{\lambda^2}.
\end{aligned} \tag{16}$$

Taking into account that:

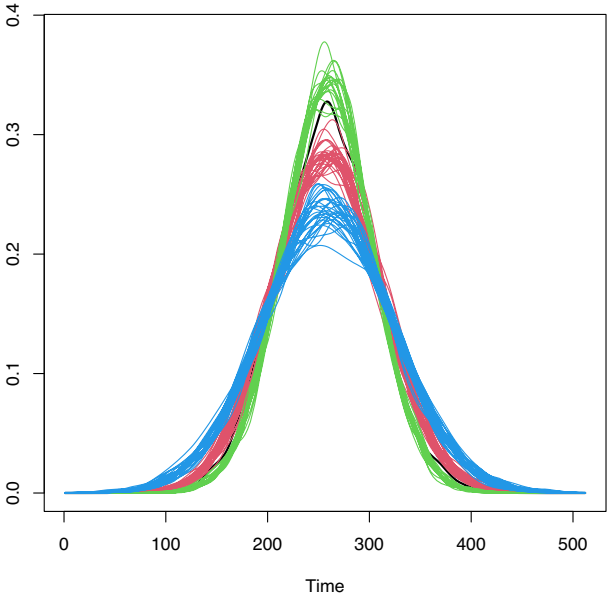
$$\begin{aligned}
&\frac{\partial}{\partial \lambda}\left[\sum_{i=1}^{T} u_{ic}^m\left(1-\Delta\sqrt{\hat{f}_i(x_j)\hat{g}_c(x_j)}\right)-\lambda\left(\sum_{j=1}^{N}\hat{g}_c(x_j)-1\right)\right]=0\\
&\Leftrightarrow \sum_{j=1}^{N}\hat{g}_c(x_j)=1\\
&\Leftrightarrow \lambda^2=\sum_{j=1}^{N}\frac{\Delta^2}{4}\left[\sum_{i=1}^{T} u_{ic}^m\sqrt{\hat{f}_i(x_j)}\right]^2.
\end{aligned} \tag{17}$$

Using (17) in Eq. 16, we obtain (6). □

Appendix 2

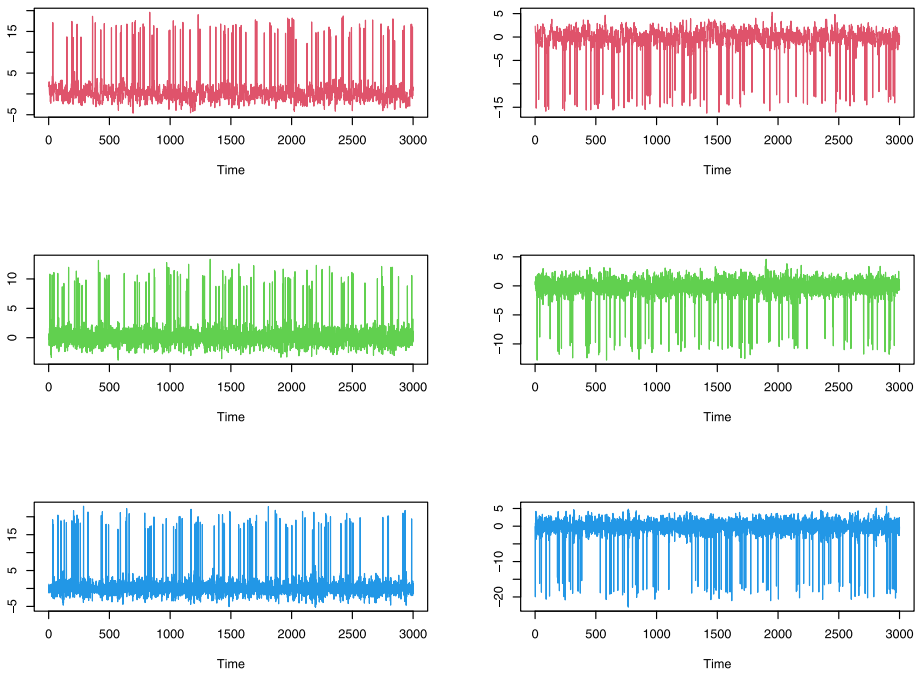


(a) Simulation 1 - time series

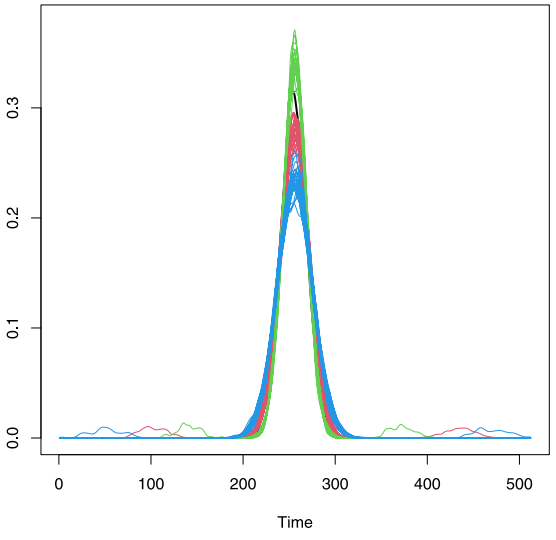


(b) Simulation 1 - kernel densities of the time series

Fig. 8 Simulation 1. The switching time series is the black one

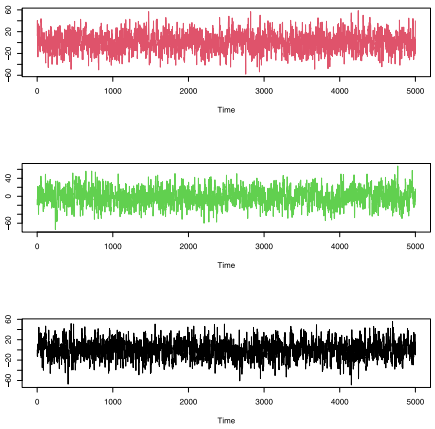


(a) Simulation 2 - noise time series (1, 10, 40, 50, 65, 70)

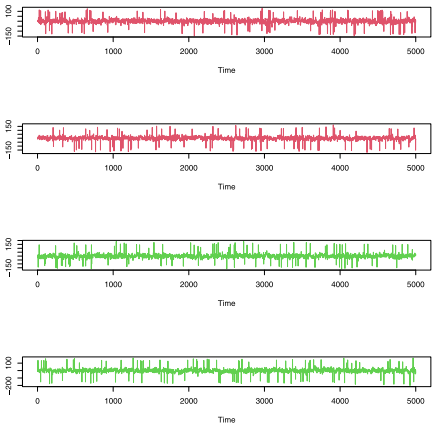


(b) Simulation 2 - kernel densities of the 81 time series

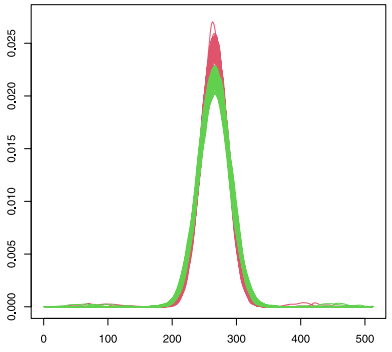
Fig. 9 Simulation 2. The switching time series is the black one



(a) Simulation 3 - time series



(b) Simulation 3 - noise time series



(c) Simulation 3 - kernel densities of the time series

Fig. 10 Simulation 3. The mixture time series is the black one

B.1 Simulation 1

Table 6 Simulation 1: clustering methods and their evaluation metrics for $m = 1.3$ and $m = 1.5$

Methods	m = 1.3			m = 1.5		
	ARI	FARI	Time in secs	ARI	FARI	Time in secs
FPAC-HD-KE (Gaussian)	0.999	0.994	0.370	0.999	0.965	0.387
FPAC-HD-KE (Epanechnikov)	0.999	0.994	0.362	0.999	0.964	0.377
FPAC-HD-KE (Triangular)	0.999	0.994	0.380	0.999	0.965	0.376
FPAM-HD-KE (Gaussian)	0.999	0.989	0.232	0.999	0.947	0.219
FPAM-HD-KE (Epanechnikov)	0.999	0.989	0.271	0.999	0.946	0.250
FPAM-HD-KE (Triangular)	0.999	0.989	0.220	0.999	0.947	0.217
FPAM-HD-KE (d_{ACF})	1.000	1.000	4.236	0.999	0.991	4.638
FPAM-HD-KE (d_{PACF})	1.000	1.000	6.210	0.998	0.990	6.336
FPAM-HD-KE (d_{DM})	1.000	0.988	0.137	0.999	0.921	0.130
FPAM-HD-KE (d_{DWT})	1.000	0.999	0.286	1.000	0.984	0.244
FPAM-HD-KE (d_{MAH})	1.000	1.000	16.916	1.000	1.000	17.319
FPAM-HD-KE (d_{PIC})	1.000	0.996	7.390	0.997	0.965	7.315

Table 7 Simulation 1: comparison of membership degrees between FPAM-HD-KE and FPAC-HD-KE with m=1.5 and Gaussian kernel

	FPAM-HD-KE			FPAC-HD-KE		
	C1	C2	C3	C1	C2	C3
1	0.98	0.01	0.02	0.99	0.00	0.01
2	0.97	0.02	0.01	0.96	0.03	0.01
3	0.96	0.01	0.03	0.98	0.00	0.01
4	0.97	0.02	0.02	0.99	0.01	0.00
5	0.97	0.02	0.01	0.99	0.00	0.00
6	0.88	0.01	0.11	0.94	0.01	0.05
7	0.98	0.01	0.02	0.99	0.00	0.01
8	1.00	0.00	0.00	1.00	0.00	0.00
9	0.95	0.03	0.01	0.98	0.01	0.00
10	0.99	0.00	0.01	0.99	0.00	0.01
11	0.99	0.00	0.01	0.99	0.00	0.00
12	0.97	0.01	0.01	0.98	0.01	0.01
13	0.98	0.01	0.01	0.99	0.00	0.00
14	0.99	0.00	0.01	1.00	0.00	0.00
15	0.98	0.01	0.01	0.99	0.00	0.01
16	0.98	0.00	0.01	0.99	0.00	0.01
17	0.94	0.01	0.05	0.97	0.01	0.02
18	0.97	0.01	0.03	0.99	0.00	0.01
19	0.97	0.02	0.01	0.99	0.00	0.00
20	0.99	0.01	0.01	1.00	0.00	0.00
21	0.98	0.01	0.01	0.99	0.01	0.00

Table 7 continued

	FPAM-HD-KE			FPAC-HD-KE		
	C1	C2	C3	C1	C2	C3
22	0.94	0.05	0.01	0.98	0.02	0.00
23	0.87	0.01	0.12	0.84	0.01	0.15
24	0.91	0.08	0.01	0.96	0.03	0.01
25	0.93	0.05	0.02	0.98	0.02	0.01
26	0.98	0.01	0.01	0.99	0.01	0.01
27	0.93	0.01	0.07	0.94	0.01	0.05
28	0.94	0.02	0.05	0.97	0.01	0.02
29	0.96	0.02	0.03	0.99	0.01	0.01
30	0.96	0.02	0.02	0.99	0.01	0.01
31	0.00	1.00	0.00	0.00	1.00	0.00
32	0.01	0.99	0.00	0.01	0.99	0.00
33	0.00	1.00	0.00	0.00	0.99	0.00
34	0.00	1.00	0.00	0.00	1.00	0.00
35	0.00	1.00	0.00	0.00	1.00	0.00
36	0.00	1.00	0.00	0.00	1.00	0.00
37	0.01	0.99	0.00	0.00	1.00	0.00
38	0.00	1.00	0.00	0.00	1.00	0.00
39	0.00	1.00	0.00	0.00	1.00	0.00
40	0.00	1.00	0.00	0.00	1.00	0.00
41	0.00	0.99	0.00	0.00	1.00	0.00
42	0.00	1.00	0.00	0.00	1.00	0.00
43	0.00	1.00	0.00	0.00	1.00	0.00
44	0.00	1.00	0.00	0.00	1.00	0.00
45	0.01	0.99	0.00	0.00	0.99	0.00
46	0.00	1.00	0.00	0.00	1.00	0.00
47	0.00	1.00	0.00	0.00	1.00	0.00
48	0.00	1.00	0.00	0.00	1.00	0.00
49	0.00	1.00	0.00	0.00	1.00	0.00
50	0.01	0.99	0.00	0.00	1.00	0.00
51	0.01	0.00	0.99	0.00	0.00	1.00
52	0.01	0.00	0.99	0.00	0.00	0.99
53	0.01	0.00	0.98	0.00	0.00	1.00
54	0.02	0.00	0.98	0.01	0.00	0.99
55	0.01	0.00	0.99	0.00	0.00	1.00
56	0.03	0.00	0.97	0.01	0.00	0.99
57	0.01	0.00	0.99	0.00	0.00	1.00
58	0.02	0.00	0.98	0.00	0.00	1.00
59	0.01	0.00	0.99	0.00	0.00	1.00
60	0.03	0.00	0.97	0.04	0.00	0.95
61	0.01	0.00	0.99	0.00	0.00	1.00
62	0.02	0.00	0.98	0.00	0.00	1.00
63	0.03	0.00	0.97	0.03	0.00	0.97

Table 7 continued

	FPAM-HD-KE			FPAC-HD-KE		
	C1	C2	C3	C1	C2	C3
64	0.01	0.00	0.99	0.01	0.00	0.99
65	0.01	0.00	0.99	0.00	0.00	1.00
66	0.00	0.00	1.00	0.00	0.00	1.00
67	0.00	0.00	1.00	0.01	0.00	0.99
68	0.01	0.00	0.98	0.00	0.00	1.00
69	0.01	0.00	0.99	0.00	0.00	0.99
70	0.07	0.00	0.93	0.09	0.00	0.91
71	0.03	0.00	0.97	0.01	0.00	0.99
72	0.03	0.00	0.97	0.01	0.00	0.99
73	0.02	0.00	0.98	0.02	0.00	0.98
74	0.01	0.00	0.99	0.01	0.00	0.99
75	0.01	0.00	0.99	0.00	0.00	1.00
76	0.01	0.00	0.98	0.00	0.00	1.00
77	0.01	0.00	0.99	0.00	0.00	1.00
78	0.01	0.00	0.99	0.01	0.00	0.99
79	0.02	0.00	0.98	0.05	0.00	0.94
80	0.04	0.00	0.96	0.02	0.00	0.98
81	0.67	0.31	0.02	0.61	0.38	0.02

B.2 Simulation 2

Table 8 Simulation 2: clustering methods and their evaluation metrics for $m = 1.3$ and $m = 1.5$

Methods	$m = 1.3$			$m = 1.5$		
	ARI	FARI	Time in secs	ARI	FARI	Time in secs
FPAC-HD-KE (Gaussian)	0.998	0.945	0.481	0.998	0.853	0.644
FPAC-HD-KE (Epanechnikov)	0.998	0.944	0.494	0.998	0.853	0.758
FPAC-HD-KE (Triangular)	0.998	0.945	0.560	0.998	0.853	0.646
FPAM-HD-KE (Gaussian)	0.998	0.933	0.204	0.998	0.874	0.245
FPAM-HD-KE (Epanechnikov)	0.998	0.933	0.209	0.998	0.874	0.215
FPAM-HD-KE (Triangular)	0.998	0.933	0.210	0.998	0.874	0.201
FPAM-HD-KE (d_{ACF})	0.929	0.899	4.178	0.929	0.891	4.296
FPAM-HD-KE (d_{PACF})	0.931	0.901	5.976	0.931	0.892	6.025
FPAM-HD-KE (d_{DM})	0.872	0.878	0.158	0.874	0.821	0.130
FPAM-HD-KE (d_{DWT})	0.848	0.859	0.331	0.848	0.856	0.362
FPAM-HD-KE (d_{MAH})	0.923	0.906	18.767	0.921	0.899	18.635
FPAM-HD-KE (d_{PIC})	0.928	0.896	7.602	0.928	0.874	7.660

Table 9 Simulation 2: comparisons among fuzzy partitions for FPAM-HD-KE method with different distance measures: h^2 (Gaussian kernel), d_{ACF}, d_{DWT}, and d_{PIC} and $m = 1.5$

	FPAM-HD-KE (Gaussian)			FPAM-HD-KE (d_{ACF})			FPAM-HD-KE (d_{DWT})			FPAM-HD-KE (d_{PIC})		
	C1	C2	C3	C1	C2	C3	C1	C2	C3	C1	C2	C3
1	0.53	0.23	0.24	0.58	0.28	0.14	0.23	0.29	0.48	0.53	0.31	0.15
2	0.99	0.01	0.00	1.00	0.00	0.00	0.99	0.01	0.00	1.00	0.00	0.00
3	0.98	0.02	0.01	1.00	0.00	0.00	0.99	0.01	0.00	0.99	0.00	0.00
4	0.99	0.00	0.01	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
5	0.97	0.02	0.01	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
6	0.99	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
7	0.97	0.01	0.01	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
8	0.99	0.00	0.01	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
9	0.98	0.01	0.01	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
10	0.53	0.22	0.26	0.43	0.40	0.17	0.25	0.29	0.45	0.40	0.41	0.19
11	0.97	0.02	0.01	1.00	0.00	0.00	0.99	0.01	0.00	1.00	0.00	0.00
12	0.99	0.00	0.01	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
13	0.99	0.00	0.01	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
14	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
15	0.99	0.00	0.01	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
16	0.99	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.99	0.00	0.00
17	0.99	0.01	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.99	0.01	0.00
18	0.98	0.01	0.01	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
19	0.97	0.01	0.02	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
20	0.99	0.01	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
21	0.97	0.01	0.02	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
22	0.96	0.01	0.03	1.00	0.00	0.00	0.99	0.01	0.00	1.00	0.00	0.00
23	0.99	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
24	0.92	0.01	0.06	1.00	0.00	0.00	0.99	0.00	0.00	1.00	0.00	0.00

Table 9 continued

	FPAM-HD-KE (Gaussian)			FPAM-HD-KE (d_{ACF})			FPAM-HD-KE (d_{DWT})			FPAM-HD-KE (d_{PIC})		
	C1	C2	C3	C1	C2	C3	C1	C2	C3	C1	C2	C3
25	0.99	0.01	0.01	1.00	0.00	0.00	0.99	0.01	0.00	1.00	0.00	0.00
26	0.97	0.02	0.01	1.00	0.00	0.00	0.99	0.01	0.00	1.00	0.00	0.00
27	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
28	1.00	0.00	0.00	1.00	0.00	0.00	0.99	0.01	0.00	1.00	0.00	0.00
29	0.77	0.03	0.19	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
30	0.99	0.01	0.01	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
31	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
32	0.01	0.99	0.00	0.00	1.00	0.00	0.01	0.98	0.00	0.00	0.98	0.02
33	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
34	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
35	0.01	0.99	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
36	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
37	0.01	0.99	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
38	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
39	0.01	0.99	0.00	0.00	0.99	0.01	0.00	1.00	0.00	0.00	0.99	0.01
40	0.26	0.67	0.07	0.13	0.68	0.19	0.13	0.20	0.67	0.13	0.66	0.20
41	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
42	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
43	0.00	1.00	0.00	0.00	0.99	0.01	0.00	1.00	0.00	0.00	0.99	0.01
44	0.01	0.99	0.00	0.00	1.00	0.00	0.01	0.99	0.00	0.00	1.00	0.00
45	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
46	0.00	1.00	0.00	0.00	0.99	0.00	0.00	1.00	0.00	0.00	1.00	0.00
47	0.01	0.99	0.00	0.00	0.99	0.01	0.00	1.00	0.00	0.00	0.99	0.01
48	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
49	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.93	0.07

Table 9 continued

	FPAM-HD-KE (Gaussian)			FPAM-HD-KE (d_{ACF})			FPAM-HD-KE (d_{DWT})			FPAM-HD-KE (d_{PIC})		
	C1	C2	C3	C1	C2	C3	C1	C2	C3	C1	C2	C3
50	0.31	0.60	0.09	0.15	0.65	0.19	0.18	0.25	0.57	0.16	0.64	0.21
51	0.01	0.00	0.99	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.03	0.97
52	0.02	0.00	0.98	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
53	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.01	0.99	0.00	0.00	1.00
54	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.01	0.99
55	0.01	0.00	0.99	0.00	0.00	1.00	0.00	0.01	0.99	0.00	0.07	0.93
56	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.02	0.98
57	0.01	0.00	0.99	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
58	0.01	0.00	0.99	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.06	0.93
59	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.01	0.99	0.00	0.00	1.00
60	0.01	0.00	0.99	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
61	0.04	0.00	0.96	0.00	0.00	1.00	0.01	0.02	0.97	0.00	0.03	0.97
62	0.01	0.00	0.99	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.04	0.96
63	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.01	0.99	0.00	0.06	0.94
64	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
65	0.30	0.07	0.63	0.07	0.76	0.17	0.25	0.30	0.45	0.12	0.67	0.21
66	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
67	0.01	0.00	0.99	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
68	0.01	0.00	0.99	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
69	0.14	0.00	0.85	0.00	0.01	0.99	0.02	0.05	0.94	0.00	0.00	1.00
70	0.27	0.07	0.67	0.08	0.76	0.17	0.26	0.30	0.44	0.12	0.66	0.21
71	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.11	0.89
72	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
73	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.03	0.97
74	0.00	0.00	1.00	0.00	0.01	0.99	0.00	0.01	0.99	0.00	0.01	0.99

Table 9 continued

	FPAM-HD-KE (Gaussian)			FPAM-HD-KE (d_{ACF})			FPAM-HD-KE (d_{DWT})			FPAM-HD-KE (d_{PIC})		
	C1	C2	C3	C1	C2	C3	C1	C2	C3	C1	C2	C3
75	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
76	0.19	0.00	0.80	0.00	0.00	1.00	0.02	0.06	0.92	0.00	0.01	0.99
77	0.01	0.00	0.99	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
78	0.01	0.00	0.99	0.00	0.00	1.00	0.00	0.01	0.99	0.00	0.00	1.00
79	0.01	0.00	0.98	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
80	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
81	0.29	0.70	0.01	0.60	0.27	0.13	0.52	0.45	0.03	0.45	0.36	0.19

B.3 Simulation 3

Table 10 Simulation 3: clustering methods and their evaluation metrics for $m = 1.3$ and $m = 1.5$

Methods	m = 1.3			m = 1.5		
	ARI	FARI	Time in secs	ARI	FARI	Time in secs
FPAC-HD-KE (Gaussian)	0.999	0.952	0.584	0.999	0.864	0.669
FPAC-HD-KE (Epanechnikov)	0.999	0.952	0.654	0.999	0.863	0.672
FPAC-HD-KE (Triangular)	0.999	0.952	0.656	0.999	0.863	0.695
FPAM-HD-KE (Gaussian)	0.998	0.943	0.451	0.999	0.859	0.408
FPAM-HD-KE (Epanechnikov)	0.998	0.944	0.381	0.998	0.859	0.376
FPAM-HD-KE (Triangular)	0.998	0.944	0.383	0.998	0.859	0.367
FPAM-HD-KE (d_{ACF})	0.932	0.898	19.239	0.932	0.827	17.026
FPAM-HD-KE (d_{PACF})	0.939	0.924	21.688	0.939	0.862	22.229
FPAM-HD-KE (d_{DM})	0.473	0.215	0.213	0.421	0.115	0.202
FPAM-HD-KE (d_{DWT})	0.936	0.894	0.648	0.936	0.790	0.537
FPAM-HD-KE (d_{MAH})	0.939	0.934	81.840	0.939	0.908	84.600
FPAM-HD-KE (d_{PIC})	0.939	0.905	37.319	0.939	0.810	37.058

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00357-025-09535-0.

Acknowledgements The authors would like to thank Editor-in-Chief Professor Paul McNicholas, the Associate Editor, and the anonymous reviewers for their valuable suggestions to improve the quality of our paper.

Author Contribution
- Conceptualization and methodology: P. D'Urso, L. De Giovanni, Ruey S. Tsay
- Software, data curation, investigation and visualization: L. De Giovanni, V. Vitale
- Writing: P. D'Urso, L. De Giovanni, Ruey S. Tsay, V. Vitale
- Supervision: P. D'Urso , L. De Giovanni, Ruey S. Tsay, V. Vitale

Funding Open access funding provided by Luiss University within the CRUI-CARE Agreement.

Data Availability The data used in the Application 5.1 is available on the website referred to in the References as Shang and Hyndman (2018). Processed data can be provided by the corresponding author on reasonable request.

Code Availability Code used in the analysis can be provided by the corresponding author on reasonable request.

Declarations

Conflict of Interest The authors declare no competing interests.

References

Aghabozorgi, S., Seyed Shirkhorshidi, A., & Ying Wah, T. (2015). Time-series clustering - a decade review. *Information Systems, 53*, 16–38.

Alonso, A., & Maharaj, E. (2006). Comparison of time series using subsampling. *Computational Statistics & Data Analysis, 50*, 2589–2599.

Alonso, A. M., & Peña, D. (2019). Clustering time series by linear dependency. *Statistics and Computing, 29*, 655–676.

Anderson, D. T., Bezdek, J. C., Popescu, M., & Keller, J. M. (2010). Comparing fuzzy, probabilistic, and possibilistic partitions. *IEEE Transactions on Fuzzy Systems, 18*, 906–918.

Arabie, P., Hubert, L., & Gaul, W. (1996). From data to knowledge: Theoretical and practical aspects of classification, data analysis, and knowledge organization. *Studies in classification, data analysis, and knowledge organization*. Springer-Verlag Berlin Heidelberg.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Norwell, MA, USA: Kluwer Academic Publishers.

Bouveyron, C., Celeux, G., Murphy, T. B., & Raftery, A. E. (2019). Model-based clustering and classification for data science: With Applications in R. Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press.

Bouveyron, C., & Brunet, C. (2014). Model-based clustering of high-dimensional data: A review. *Computational Statistics & Data Analysis, 71*, 52–78.

Caiado, J., & Crato, N. (2009). Identifying common dynamic features in stock returns. *Quantitative Finance, 10*, 797–807.

Caiado, J., Crato, N., & Peña, D. (2006). A periodogram-based metric for time series classification. *Computational Statistics & Data Analysis, 50*, 2668–2684.

Caiado, J., Maharaj, E. A., & D'Urso, P., et al. (2015). Time series clustering. In C. Hennig, M. Meila, & F. Murtagh (Eds.), *Handbook of cluster analysis* (pp. 241–264). Chapman and Hall/CRC.

Campello, R. J. G. B. (2007). A fuzzy extension of the rand index and other related indexes for clustering and classification assessment. *Pattern Recognition Letters, 28*(7), 833–841.

Campello, R. J., & Hruschka, E. R. (2006). A fuzzy extension of the silhouette width criterion for cluster analysis. *Fuzzy Sets and Systems, 157*(21), 2858–2875.

Cannon, R., Dave, J., & Bezdek, J. (1986). Efficient implementation of distinct the fuzzy c-means clustering algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI, 8*(2), 248–255.

Cheng, T. W., Goldgof, D., & Hall, L. (1995). Fast clustering with application to fuzzy rule generation. *Proceedings of 1995 IEEE International Conference on Fuzzy Systems* (vol. 4, pp. 2289–2295).

Chitta, R., Jin, R., Havens, T., & Jain, A. K. (2011). Approximate kernel k-means: Solution to large scale kernel clustering. In *Proceedings of the ACM SIGKDD International conference on knowledge discovery and data mining* (pp. 895–903).

Dose, C., & Cincotti, S. (2005). Clustering of financial time series with application to index and enhanced index tracking portfolio. *Physica A: Statistical Mechanics and its Applications, 355*, 145–151.

D'Urso, P., & Maharaj, E. A. (2009). Autocorrelation-based fuzzy clustering of time series. *Fuzzy Sets and Systems, 160*(24), 3565–3589.

D'Urso, P. (2015). Fuzzy clustering. In C. Hennig, M. Meila, F. Murtagh, & R. Rocci (Eds.), *Handbook of Cluster Analysis* (pp. 545–573). Chapman and Hall.

D'Urso, P. (2017). Informational paradigm, management of uncertainty and theoretical formalisms in the clustering framework: A review. *Information Sciences, 400–401*, 30–62.

D'Urso, P., De Giovanni, L., Maharaj, E. A., & Massari, R. (2014). Wavelet-based self-organizing maps for classifying multivariate time series. *Journal of Chemometrics, 28*(1), 28–51.

D'Urso, P., De Giovanni, L., & Massari, R. (2015). Time series clustering by a robust autoregressive metric with application to air pollution. *Chemometrics and Intelligent Laboratory Systems, 141*, 107–124.

D'Urso, P., De Giovanni, L., & Massari, R. (2016). Garch-based robust clustering of time series. *Fuzzy Sets and Systems, 305*, 1–28.

D'Urso, P., Giovanni, L., & Massari, R. (2018). Robust fuzzy clustering of multivariate time trajectories. *International Journal of Approximate Reasoning, 99*, 12–38.

D'Urso, P., De Giovanni, L., Massari, R., D'Ecclesia, R., & Maharaj, E. (2020). Cepstral-based clustering of financial time series. *Expert Systems with Applications, 161*(113), 705.

D'Urso, P., & Maharaj, E. A. (2012). Wavelets-based clustering of multivariate time series. *Fuzzy Sets and Systems, 193*, 33–61.

D'Urso, P., Maharaj, E. A., & Alonso, A. M. (2017). Fuzzy clustering of time series using extremes. *Fuzzy Sets and Systems, 318*, 56–79.

Eschrich, S., Ke, J., Hall, L. O., & Goldgof, D. B. (2003). Fast accurate fuzzy clustering through data reduction. *Transactions on Fuzzy Systems, 11*(2), 262–270.
Euán, C., Sun, Y., & Ombao, H. (2019). Coherence-based time series clustering for statistical inference and visualization of brain connectivity. *The Annals of Applied Statistics, 13*, 990–1015.
Everitt, B., Landau, S., Leese, M., & Stahl, D. (2011). *Cluster analysis* (5th ed.). London: John Wiley & Sons Ltd.
Galeano, P., & Peña, D. (2000). Multivariate analysis in vector time series. *Resenhas do Instituto de Matemática e Estatística da Universidade de São Paulo, 4*(4), 383–403.
Galeano, P., & Peña, D. (2019). Data science, big data and statistics. *TEST, 28*, 289–329.
Gao, Z., & Tsay, R. S. (2020). Modeling high-dimensional unit-root time series. arXiv:2005.03496
García-Escudero, L., & Gordaliza, A. (1999). Robustness properties of *k* means and trimmed *k* means. *Journal of the American Statistical Association, 94*(447), 956–969.
García-Escudero, L. A., & Gordaliza, A. (2005). A proposal for robust curve clustering. *Journal of Classification, 22*(2), 185–201.
García-Escudero, L. A., Gordaliza, A., Matrán, C., & Mayo-Iscar, A. (2010). A review of robust clustering methods. *Advances in Data Analysis and Classification, 4*(2–3), 89–109.
Gareth, J., Witten, D., Hastie, T., & Tibshirani, R. (2014). *An introduction to statistical learning: With Applications in R*. Incorporated: Springer Publishing Company.
Grün, B., & Hornik, K. (2011). topicmodels: An R package for fitting topic models. *Journal of Statistical Software, 40*(13), 1–30. https://doi.org/10.18637/jss.v040.i13
Gyorfi, L., Hardle, W., & Sarda, P. (2014). *Nonparametric curve estimation from time series*. Springer.
Hardle, W., Sarda, P., Vieu, P., & Gyorfi, L. (1989). Nonparametric curve estimation from time series. In *Lecture notes in statistics*. Springer.
Hathaway, R. J., & Bezdek, J. C. (2006). Extending fuzzy and probabilistic clustering to very large data sets. *Computational Statistics & Data Analysis, 51*(1), 215–234.
Havens, T., Chitta, R., Jain, A. K., & Jin, R. (2011). Speedup of fuzzy and possibilistic kernel c-means for large-scale clustering. *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)* (pp. 463–470).
Havens, T., Bezdek, J., Leckie, C., Hall, L. O., & Palaniswami, M. (2012). Fuzzy c-means algorithms for very large data. *IEEE Trans Fuzzy Systems, 20*(6), 1130–1146.
Heiser, W., & Groenen, P. (1997). Cluster differences scaling with a within-clusters loss component and a fuzzy successive approximation strategy to avoid local minima. *Psychometrika, 62*(1), 63–83.
Hore, P., Hall, L. O., & Goldgof, D. B. (2007). Single pass fuzzy c means. In *2007 IEEE International Fuzzy Systems Conference* (pp. 1–7).
Hore, P., Hall, L. O., Goldgof, D. B., Gu, Y., Maudsley, A. A., & Darkazanli, A. (2009). A scalable framework for segmenting magnetic resonance images. *Journal of Signal Processing Systems, 54*(1–3), 183–203.
Hüllermeier, E., Rifqi, M., Henzgen, S., & Senge, R. (2012). Comparing fuzzy partitions: A generalization of the Rand index and related measures. *Fuzzy Systems, IEEE Transactions on, 20*(3), 546–556.
Hwang, H., DeSarbo, W., & Takane, Y. (2007). Fuzzy clusterwise generalized structured component analysis. *Psychometrika, 72*(2), 181–198.
Ismail, M. A., & Selim, S. Z. (1986). On the local optimality of the fuzzy isodata clustering algorithm. *IEEE Transactions on Pattern Analysis & Machine Intelligence, 8*(02), 284–288.
Kamdar, T., & Joshi, A. (2000). On creating adaptive web servers using weblog mining.
Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis*. Wiley-Blackwell.
Kolen, J. F., & Hutcheson, T. (2002). Reducing the time complexity of the fuzzy c-means algorithm. *IEEE Transactions on Fuzzy Systems, 10*(2), 263–267.
Kriegel, H., Kröger, P., & Zimek, A. (2009). Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data, 3*, 1–58.
Krishnapuram, R., Joshi, A., Nasraoui, O., & Yi, L. (2001). Low-complexity fuzzy relational clustering algorithms for web mining. *IEEE transactions on Fuzzy Systems, 9*(4), 595–607.
Krishnapuram, R., Joshi, A., & Yi, L. (1999). A fuzzy relative of the k-medoids algorithm with application to web document and snippet clustering. *FUZZ-IEEE 99 1999 IEEE International Fuzzy Systems Conference Proceedings, 3*, 1281–1286.
Lafuente Rego, B., D'Urso, P., & Vilar, J. (2018). Robust fuzzy clustering based on quantile autocovariances. Statistical Papers.
Lahmiri, S. (2016). Clustering of casablanca stock market based on hurst exponent estimates. *Physica A: Statistical Mechanics and its Applications, 456*, 310–318.

Liao, L., & Sheng Lin, T. (2007). A fast spatial constrained fuzzy kernel clustering algorithm for mri brain image segmentation. *International Conference on Analysis and Pattern Recognition, 1*, 82–87.
MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297).
Maharaj, E. A. (1996). A significance test for classifying arma models. *Journal of Statistical Computation and Simulation, 54*(4), 305–331.
Maharaj, E. A. (1999). Comparison and classification of stationary multivariate time series. *Pattern Recognition, 32*(7), 1129–1138.
Maharaj, E. A. (2000). Clusters of time series. *Journal of Classification, 17*(2), 297–314.
Maharaj, E. A., Alonso, A. M., & D'Urso, P. (2015). Clustering seasonal time series using extreme value analysis: An application to spanish temperature time series. *Communications in Statistics: Case Studies, Data Analysis and Applications, 1*(4), 175–191.
Maharaj, E. A., & D'Urso, P. (2010). A coherence-based approach for the pattern recognition of time series. *Physica A: Statistical mechanics and its Applications, 389*(17), 3516–3537.
Maharaj, E. A., & D'Urso, P. (2011). Fuzzy clustering of time series in the frequency domain. *Information Science, 181*, 1187–1211.
Maharaj, E. A., D'Urso, P., & Galagedera, D. U. (2010). Wavelet-based fuzzy clustering of time series. *Journal of Classification, 27*(2), 231–275.
Martin, R. J. (2000). A metric for arma processes. *IEEE Transactions on Signal Processing, 48*(4), 1164–1170.
McBratney, A., & Moore, A. (1985). Application of fuzzy sets to climatic classification. *Agricultural and forest meteorology, 35*(1–4), 165–185.
Montero, P., & Vilar, J. A. (2015). Tsclust: An r package for time series clustering. *Journal of Statistical Software, 62*, 1–43.
Pal, N., & Bezdek, J. (1995). On cluster validity for the fuzzy c-means model. *IEEE Transactions on Fuzzy Systems, 3*, 370–379.
Peña, D., & Tsay, R. S. (2021). *Statistical learning for big dependent data*. John Wiley & Sons.
Pértega, D., & VilarJosé, A. (2010). Comparing several parametric and nonparametric approaches to time series clustering: A simulation study. *Journal of Classification, 3*(27), 333–362.
Piccolo, D. (1990). A distance measure for classifying arima models. *Journal of Time Series Analysis, 11*(2), 153–164.
Shang, H. L., & Hyndman, R. J. (2018). fds: Functional Data Sets. https://CRAN.R-project.org/package=fds, r package version 1.8.
Shankar, B., & Pal, N. (1994). An effective approach for large data sets. In *Proceedings of the Third International conference on fuzzy logic, neural nets and soft computing* (pp. 331 –332).
Shen, Y., Pedrycz, W., Chen, Y., Wang, X., & Gacek, A. (2019). Hyperplane division in fuzzy c-means: Clustering big data. *IEEE Transactions on Fuzzy Systems* 1–1.
Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. Chapman & Hall.
Tong, H., & Dabas, P. (1990). Cluster of time series models: An example. *Journal of Applied Statistics, 17*, 187–198.
Tsay, R. S. (2016). Some methods for analyzing big dependent data. *Journal of Business & Economic Statistics, 34*(4), 673–688.
Vilar, F., & Vilar, J. A. (2021). mlmts: An R package for multivariate time series clustering. Available at https://cran.r-project.org/package=mlmts
Vilar, J. A., Lafuente-Rego, B., & D'Urso, P. (2018). Quantile autocovariances: A powerful tool for hard and soft partitional clustering of time series. *Fuzzy Sets and Systems, 340*, 38–72.
Webb, A. R., & Copsey, K. D. (2011). Statistical pattern recognition, Third Edition. John Wiley & Sons.
Witten, D., & Tibshirani, R. (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association, 105*, 713–726.
Wu, J., Wu, Z., Cao, J., Liu, H., Chen, G., & Zhang, Y. (2017). Fuzzy consensus clustering with applications on big data. *IEEE Transactions on Fuzzy Systems, 25*, 1430–1445.