

Article

A Novel Hybrid Algorithm to Forecast Functional Time Series Based on Pattern Sequence Similarity with Application to Electricity Demand

Francisco Martínez-Álvarez ^{1,*}, Amandine Schmutz ², Gualberto Asencio-Cortés ¹ and Julien Jacques ²

¹ Data Science & Big Data Lab, Universidad Pablo de Olavide, ES-41013 Seville, Spain; guaasecor@upo.es

² Université de Lyon 2, 69000 Lyon, France; aschmutz@lim-group.com (A.S.); julien.jacques@univ-lyon2.fr (J.J.)

* Correspondence: fmaralv@upo.es

Received: 14 September 2018; Accepted: 3 December 2018; Published: 28 December 2018



Abstract: The forecasting of future values is a very challenging task. In almost all scientific disciplines, the analysis of time series provides useful information and even economic benefits. In this context, this paper proposes a novel hybrid algorithm to forecast functional time series with arbitrary prediction horizons. It integrates a well-known clustering functional data algorithm into a forecasting strategy based on pattern sequence similarity, which was originally developed for discrete time series. The new approach assumes that some patterns are repeated over time, and it attempts to discover them and evaluate their immediate future. Hence, the algorithm first applies a clustering functional time series algorithm, i.e., it assigns labels to every data unit (it may represent either one hour, or one day, or any arbitrary length). As a result, the time series is transformed into a sequence of labels. Later, it retrieves the sequence of labels occurring just after the sample that we want to be forecasted. This sequence is searched for within the historical data, and every time it is found, the sample immediately after is stored. Once the searching process is terminated, the output is generated by weighting all stored data. The performance of the approach has been tested on real-world datasets related to electricity demand and compared to other existing methods, reporting very promising results. Finally, a statistical significance test has been carried out to confirm the suitability of the election of the compared methods. In conclusion, a novel algorithm to forecast functional time series is proposed with very satisfactory results when assessed in the context of electricity demand.

Keywords: time series; functional data; forecasting; pattern sequence similarity

1. Introduction

Functional Data Analysis (FDA) is a new field of statistics that has emerged in the last three decades with the aim of adapting statistical methodologies for this kind of data. Broadly interpreted, FDA deals with functional random variables. Some comprehensive references are found in [1], which review relevant methods for FDA, including smoothing, basis function expansion, principal component analysis, canonical correlation, regression analysis, and differential analysis. This paper is focused on Functional Time Series (FTS), which can be defined as a time sequence of functional observations [2]. Note that most of the issues found in FDA can be extended to FTS.

While there is a number of approaches dealing with discrete time series, there are few contributions paying attention to FTS [3]. In particular, the forecasting of FTS has not been extensively studied, as compared to other fields of research: this is the main motivation for proposing a novel forecasting algorithm.

On the one hand, a successful forecasting algorithm, the Pattern Sequence Forecasting algorithm (PSF), was introduced in [4] and an R package later developed [5]. The authors proposed a new paradigm of forecasting based on the search for pattern sequence similarities in historical data. However, the algorithm could only be applied to discrete time series.

On the other hand, a clustering algorithm for multivariate functional data, funHDDC, was introduced in [6]. An extension was later proposed [7] along with its R package [8] in 2018. After introducing multivariate functional principal component analysis, a parametric mixture model, based on the assumption of the normality of the principal component scores, was defined and estimated by an EM-like algorithm. Its efficiency and usefulness have been shown in many case studies.

Thus, a hybrid approach making the most of both PSF and funHDDC, hereafter called funPSF, is proposed. This combination turns PSF into an FTS forecasting algorithm and funHDDC into a clustering algorithm able to deal with univariate time series. Additionally, some improvements have been added to the original approaches: (i) a novel voting system to calculate the optimal number of clusters that combines four well-known indices is proposed; (ii) the way PSF generates the outputs is modified in order to weight those patterns closer in time; (iii) a different data normalization is carried out in order to minimize the impact of very low values, typically occurring in some time series; and (iv) different funPSF models are simultaneously applied to the same time series, taking into consideration different features that different periods of time may exhibit in a time series.

The performance of this approach was tested on real-world time series. Namely, the challenging problem of accurately forecasting electricity demand in Uruguay was addressed, providing an interesting tool for the industrial sector. Eight years were considered, and the reported results outperformed well-established methods, such as artificial neural networks (ANN) or autoregressive integrated moving average (ARIMA), in terms of accuracy.

The rest of the paper is structured as follows. Section 2 presents an exhaustive revision of the state-of-the-art related literature. Section 3 introduces the proposed methodology and the description of the funPSF algorithm, which was developed to be applied to FTS. Section 4 shows the results obtained by the approach in the electric energy markets of Uruguay, for the years 2008–2014, including measures of their quality. This section also includes comparisons between the proposed method and other techniques, as well as statistical significance tests. Section 5 summarizes the main conclusions drawn and future works. Finally, the Abbreviations reports the nomenclature.

2. Related Works

This section first reviews some relevant works in the field of FTS forecasting and, later, how PSF and funHDDC have been successfully applied to different domains. Furthermore, their impact in both time series forecasting and clustering functional data domains is discussed. Note that a very interesting work comparing different time series forecasting methods can be found in [9].

Functional data can be found in numerous real-life applications from different fields of applied sciences. Hence, the study of demographic curves (mortality and fertility rates for each age observed every year) can be found in [10]. A short-term approach for traffic flow forecasting was introduced in [11]. The volatility in time series was studied, by proposing a functional version for the ARCH model [12]. The Italian gas market was also forecasted by means of constrained FTS, as introduced in [13]. The analysis of the evolution of intraday particulate matter concentration was reported in [14]. Another FTS approach to forecast volatility in foreign exchange markets was introduced in [15]. Rainfall was forecasted with FTS in [16]. Recently, different models to forecast electricity demand by means of FTS were introduced in [17], with a locally-recurrent fuzzy neural architecture, in [18], with a random vector functional link network, and in [19], with a new Hilbertian ARMAX model. The same authors also addressed the challenging problem of forecasting electricity prices [2]. The forecasting of chaotic time series with functional link extreme learning ANFIS was proposed in [20].

The PSF algorithm was first published in 2011 [4]. It was developed to deal with discrete time series and proposed, for the first time, to use clustering methods to transform a time series into a sequence of values. Recently, a robust implementation in R was also published [5]. PSF has also been successfully applied in the context of wind speed [21], solar power [22], and water forecasting [23]. Since its first publication, several authors have proposed modifications for its improvement. In [24], the authors modified PSF to forecast outliers in time series. Later, Fujimoto et al. [25] chose a different clustering method, in order to find clusters with different shapes. In 2013, Koprinska et al. [26] combined PSF with artificial neural networks. Five modified PSF were jointly used in [27]. Another relevant modification can be found in [28], in which the authors suggested minute modification to minimize the computation delay, and they opposed using a voting system to find the optimal number of clusters given its expensive computational cost.

On the other hand, funHDDC [6] is a functional data clustering algorithm [29]. Earlier, in 2012, a clustering algorithm for functional data in low-dimensional subspace was introduced in [30]. By contrast, the authors in [31] proposed a method based on wavelets for detecting patterns and clusters in high-dimensional time-dependent functional data one year later. A method based on similar shapes of FTS was proposed for short-term load forecasting in [32]. A similar approach was proposed one year later, in 2014, making use of techniques for clustering FTS, as well [33]. An approach for clustering functional data on convex function spaces, in which the authors focused on cases with forms of the observations known in advance, can be found in [34]. Another interesting approach, based on the study of the depth measure for functional data, was introduced in [35]. A robust clustering algorithm for stationary FTS was introduced in [36]. Finally, current efforts are directed towards the development of clustering algorithms for multivariate functional data [7,37–39].

In light of the above reviewed papers, the development of a new hybrid algorithm for forecasting FTS is justified. It has been shown that PSF can achieve competitive results in a variety of fields. Moreover, FDA is an emerging field of research in which hardly no forecasting algorithms have been proposed.

3. Methodology

Hybridization is a very common strategy currently. In this sense, the proposed method has combined two existing techniques (PSF for forecasting and funHDDC for functional clustering) in order to develop a new hybrid method, making the most of both of them, able to forecast functional time series. This section describes the methodology proposed to forecast FTS. The funPSF flowchart is illustrated in Figure 1, in which seven well-differentiated steps can be seen. These steps are now listed and detailed.

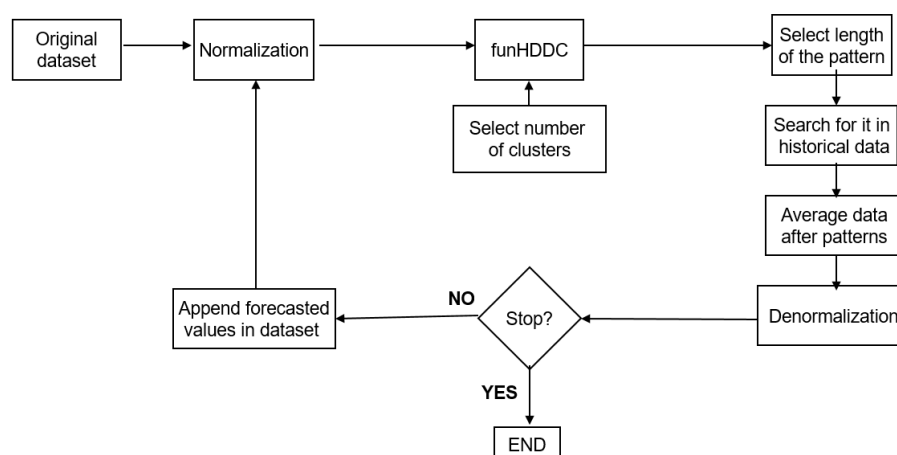


Figure 1. Flowchart of the proposed algorithm.

Step 1: Data normalization. It is well known that data can range in large intervals for some variables. In order to minimize this effect, data are usually normalized. Although there are several strategies, the one selected for funPSF is the unity-based normalization, which brings all values into the range $[0, 1]$:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (1)$$

where X' denotes the normalized data, X the original data, $X_{\max}$ the maximum value, and $X_{\min}$ the minimum value.

Step 2: Selection of the optimal number of clusters, K . One of the main issues in clustering is the determination of the optimal number of partitions. There exist several indices; however, all of them search for optimality by optimizing different aspects [40]. For this reason, in this work, a majority voting system is used that combines the results of four well-known methods: silhouette [41], Dunn [42], Davies–Bouldin [43], and the Bayesian Information Criterion (BIC) [44]. Their application can lead to two situations: more than two indices (the majority) agree on choosing the same K (and then, this K is selected) or all four methods propose different K . When the second situation takes place, the second best values of the four indices are also considered (along with the first best values). The K selected is, then, the one pointed to by the majority of all the cases. Further, the best values can be analyzed until one K has more votes than the others.

Step 3: Application of the funHDDC functional clustering data algorithm. Let $\mathbf{X}_1, \dots, \mathbf{X}_n$ be an independent and identically distributed sample of multivariate curves, such that $\mathbf{X}_i = \mathbf{X}_i(t)_{t \in [0, T]}$ with $\mathbf{X}_i(t) = (X_i^1(t), \dots, X_i^p(t))' \in \mathbb{R}^p$. Under some regularity assumption, these multivariate curves can be represented as an infinite linear combination of orthogonal eigenfunctions: $\mathbf{f}_j$:

$$\mathbf{X}_i(t) = \mu(t) + \sum_{j \geq 1} C_{ij} \mathbf{f}_j(t), \quad t \in [0, T]. \quad (2)$$

This decomposition is known as the Karhunen–Loeve expansion. It is worth noting that the computational cost of the Karhunen–Loeve transform is low as long as the observed curves are approximated in a finite basis function with a reasonable number of basis functions, which is the case in the present work.

In practice, the functional expressions of the observed curves are not known, and we only have access to discrete observations at a finite set of times. A common way to reconstruct the functional form of data is to assume that each curve can be approximated by a linear combination of basis functions (e.g., splines or Fourier): $X_i^j(t) = \sum_{r=1}^{R_j} p_{ir}^j(X_i^j) \phi_r^j(t)$.

One consequence is that only a finite number of eigenfunctions can be computed in Equation (2), and the computational cost of the Karhunen–Loeve expansion computation is consequently low (more details in [6]).

funHDDC [7] is a model-based clustering algorithm for either univariate or multivariate functional data, which estimates for each curve the cluster $Z_i \in \{1, \dots, K\}$ to which it belongs. This algorithm relies on a Karhunen–Loeve decomposition per cluster, in order to propose a fine representation of the curves, and on the following distributional assumptions:

$$Z_i \sim \mathcal{M}(\pi_1, \dots, \pi_K) \quad \text{and} \quad \mathbf{C}_i | Z_i = k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

where $\mathbf{C}_i = (C_{ij})_j$.

Moreover, in order to provide a parsimonious modeling, a specific parametrization of the variance matrix $\boldsymbol{\Sigma}_k$ is considered.

The inference of the model parameters $\boldsymbol{\theta} = (\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)_k$ is done through an EM algorithm:

1. Initialization $\boldsymbol{\theta}^{(0)}$;
2. Repeat until convergence:

- (a) E step: compute $t_{ik}^{(q)} = \mathbb{E}[Z_i | \mathbf{C}_i^{(q)}, \boldsymbol{\theta}^{(q)}]$;
- (b) M step: update the model parameters as follows:
- $\pi_k^{(q+1)} = \frac{\eta_k^{(q)}}{n}$ with $\eta_k^{(q)} = \sum_{i=1}^n t_{ik}^{(q)}$;
 - $\boldsymbol{\mu}_k^{(q+1)} = \frac{1}{\eta_k^{(q)}} \sum_{i=1}^n t_{ik}^{(q)} \mathbf{C}_i$;
 - the elements $\boldsymbol{\Sigma}_k^{(q+1)}$ are obtained from the diagonalization of $W^{1/2} D_k^{(q)} W^{1/2}$ where $D_k^{(q)} = \frac{1}{\eta_k^{(q)}} \sum_{i=1}^n t_{ik}^{(q)} (\mathbf{C}_i - \boldsymbol{\mu}_k^{(q)}) (\mathbf{C}_i - \boldsymbol{\mu}_k^{(q)})^t$, and W contains the inner products between the basis function $(\phi_r^j)_{rj}$.

It is worth noting that the EM algorithm exhibits some convergence issues [45] that may lead to the failure of the whole procedure. However, these issues are not expected to occur with time series with daily or seasonal patterns, which is the usual situation in which funPSF can be successfully applied.

Figure 2 depicts an illustrative case, in which $K = 4$. It can be seen that once funHDDC is applied, the original time series is transformed into a sequence of labels: $L = \{1, 3, 3, 1, 3, 3, 1, 3, 3\}$. Furthermore, for simplicity, it is considered that each pattern comprises one day (24 h), since it is a very common situation in problems in which funPSF can be efficiently applied.

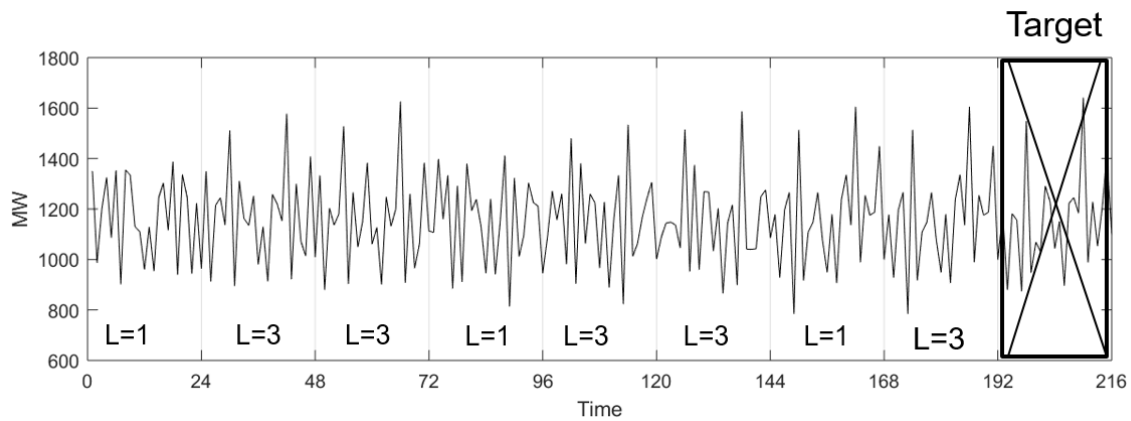


Figure 2. Application of funHDDC to the time series and transformation into a sequence of labels.

Step 4: Selection of the length of the pattern sequence or window, W . The optimal number of labels forming W is determined by minimizing the forecasting error when funPSF is applied to a training set. In practice, W is calculated by means of cross-validation, whose validity to be used for evaluating time series prediction was recently discussed in [46]. The number of folds may vary according to the type of problem to be addressed. Figure 3 shows how W is determined when the training set is one year, and 12 folds are considered. It can be appreciated that the initial dataset is split into twelve folds (this number is used for simplicity and for a better physical understanding, since each fold represents one month). Then, every fold is used as test data, and the remaining eleven folds as training data. Errors for tests are reported when varying W from 1–10 and are denoted as $e_{month}\{W = 1, \dots, 10\}$. The average errors for $W = 1, \dots, 10$ are denoted as $\mathbf{e}_{1, \dots, 10}$. Finally, the W that leads to minimal error is selected.

Step 5: Search for the pattern sequence within the historical data. At this stage of the algorithm, the dataset has already been discretized and the length of the pattern sequence determined. Hence, the sequence pattern, with length W , including the last (most recent) W labels, is retrieved, and the pattern sequence, S_W , is created. S_W is searched for within the historical data, and every time there is a match, the values identifying the next h are obtained.

Step 6: Generation of the forecast. Finally, all these values are averaged, weighting their contribution according to the distance to the present. That is, retrieved values closer to the target h

have weights higher than those that occurred further into the past. Mathematically, it can be expressed as follows:

$$\widehat{X}' = \frac{1}{M} \sum_{i=1}^M d_i^{-1} X_i'^m \quad (3)$$

where $\widehat{X}'$ is the forecasted sample, M is the number of matches encountered in the historical data, $X_i'^m$ are the i th actual h values existing after a match, and d_i the distance (or time elapsed) between the match and the h values that we want to predict.

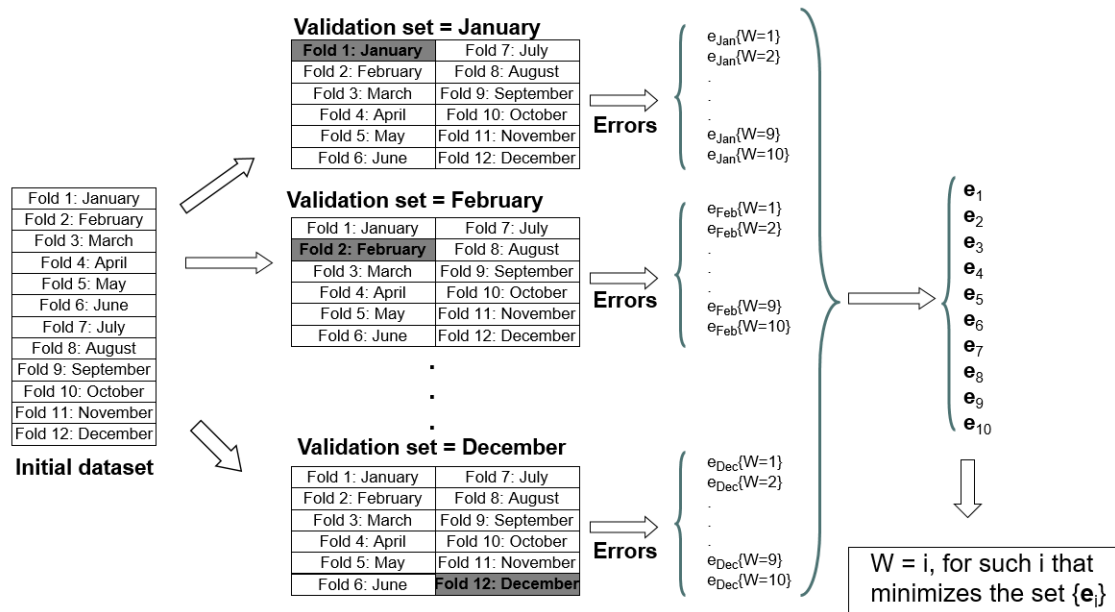


Figure 3. Determining the length of the pattern sequence by means of cross-validation.

Please note that Equation (3) does not use outputs' feedback and this could limit the accuracy of the predictions, as discussed in [47]. However, original PSF and other published variants have shown prediction robustness despite this fact.

Figure 4 illustrates Steps 5 and 6. For this particular case, $h = 24$, $W = 2$ and the pattern sequence $S_W = \{1, 3\}$. Two matches are depicted and the result of the weighted average for the values found just after the matches is the forecast itself.

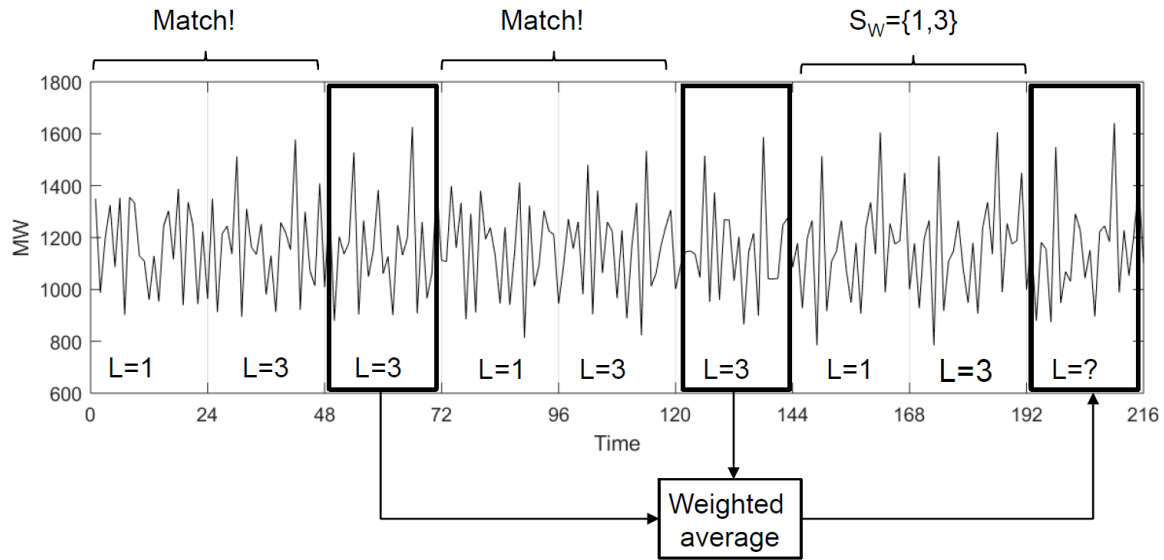


Figure 4. Determining the length of the pattern sequence by means of cross-validation.

Step 7: Data denormalization. Once the forecast is done, this value must be denormalized, leading to the final forecast, $\hat{X}$. In this case, it consists of multiplying the forecast by the same value by which the data were normalized in Step 1:

$$\hat{X} = \hat{X}^I (X_{max} - X_{min}) + X_{min} \quad (4)$$

Finally, Figure 5 shows the pseudocode for funPSF, where D denotes the dataset, K the number of clusters, L the labeled dataset such as $L = [L_1, L_2, \dots, L_{d-2}, L_{d-1}]$, W the pattern sequence length or window, T the test set, and $\hat{X}(d)$ the forecasts for all data in T .

Input: D, K, L, W, T

Output: $\hat{X}(d)$

funPSF()

$ES_d \leftarrow \{\}$

$\hat{X}(d) \leftarrow 0$

for each data $d \in T$

$S_W^{d-1} \leftarrow [L_{d-W}, L_{d-W+1}, \dots, L_{d-2}, L_{d-1}]$

for each j such as $X(j) \in D$

$S_W^j \leftarrow [L_{j-W+1}, L_{j-W+2}, \dots, L_{j-1}, L_j]$

if ($S_W^j = S_W^{d-1}$)

$ES_d \leftarrow ES_d \cup j$

for each $j \in ES_d$

$\hat{X}(d) \leftarrow \hat{X}(d) + X(j+1)$

$\hat{X}(d) \leftarrow \hat{X}(d) / \text{size}(ES_d)$

$D \leftarrow D \cup \hat{X}(d)$

$[L_1, L_2, \dots, L_{d-1}, L_d] \leftarrow \text{funHDDC}(D, K)$

$d \leftarrow d + 1$

return $\hat{X}(d)$ for all data in T

Figure 5. A general scheme of the algorithm funPattern Sequence Forecasting (funPSF).

4. Results

The application of funPSF to real-world data is reported in this section. Section 4.1 describes the dataset used related to electricity demand from Uruguay. The quality measures to assess the funPSF performance are introduced in Section 4.2. Finally, the results achieved are shown and discussed in Section 4.4.

4.1. Data Description

This section describes the dataset that was used to evaluate the funPSF performance. It consists of data from 2007–2014 of electricity demand in Uruguay, retrieved on an hourly basis (70128 samples). The full dataset is available upon request from authors.

Data did not exhibit a significant number of outliers, since the application of the test proposed in [48] showed that less than 0.01% of the values were outliers. This fact led to the discarding of the use of methods particularly designed to deal with volatility or spike values.

In particular, the data distribution for different days of the week is represented, as well as the distribution for the full dataset is illustrated in Figure 6. Axis x for all subfigures denotes the amount of electricity in MW, while axis y denotes the accumulated frequency for such values. Differences in mean and skewness can be appreciated in different days of the week. In general, while working days were slightly left-skewed, it can be appreciated that weekends were slightly right-skewed, which means that lower consumption was reported on weekends, as expected. This fact justified generating, at least, different models for different types of days.

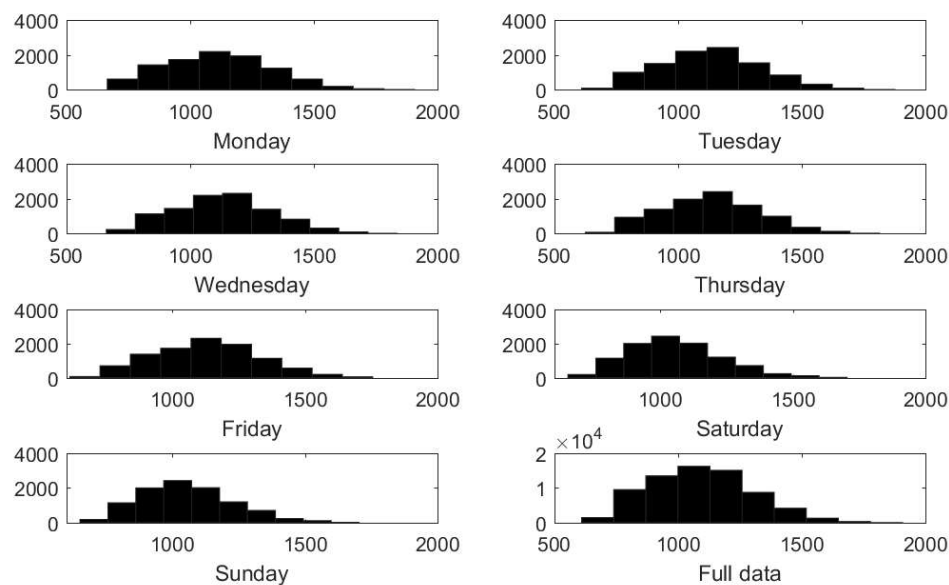


Figure 6. Data distribution for days of the week, as well as for the full dataset.

4.2. Quality Measures

Given the nature of the data to be processed, three metrics were chosen to assess the performance of funPSF.

The first one was the mean absolute error (MAE), which is typically used in the electricity time series context. Operators are interested in the total amount of energy that must be generated, and that is the reason why they typically use absolute errors.

The second widely-used metric is the mean absolute percentage error (MAPE). This error provides information that is easier to interpret and compare.

Finally, the Root Mean Squared Error (RMSE) was also provided.

The associated formulas are:

$$MAE = |D_h - \hat{D}_h| \quad (5)$$

$$MAPE = \frac{|D_h - \hat{D}_h|}{D_h} \quad (6)$$

$$RMSE = \sqrt{\overline{(D_h - \hat{D}_h)^2}} \quad (7)$$

where D_h denotes the actual demand and $\hat{D}_h$ the predicted demand for period h , and the bar above the squared differences is the mean.

4.3. Hardware and Software Resources

The experiments were carried out on the cluster physically located at the Data Science & Big Data Lab, Pablo de Olavide University. Each node forming the cluster shares the features listed in Tables 1 and 2.

Table 1. Features of the nodes: Hardware.

Hardware	
Processors	Intel(R) Core(TM) i7-5820K CPU
Cores	6 (12 threads)
Clock speed	2.13 GHz
Cache	16 MB
Network	Gigabit (1 Gb/s)
RAM	48 GB

Table 2. Features of the nodes: Software.

Software	
Framework	RStudio 1.1
OS	Ubuntu 16.04.1 LTS

4.4. Performance Assessment

To assess the performance of funPSF, years 2013 and 2014 have been selected as test sets. The remaining years (2008–2012) have been used as training test sets. The standard 70–30% data division into training and test was applied. In particular, five out of seven years were used for training, which represents 71.43%. As shown in the original PSF paper, the algorithm needs no more than two years to find patterns in the historical data to perform accurate forecasts. As a consequence, it was determined that five years was a suitable choice.

Another relevant issue is the prediction horizon, h , which was set to one day given the typology of the problem, since the electricity industry typically plans the amount of energy that needs to be generated one day in advance.

In Section 4.1, it is shown that different days may have different data distributions. For this reason, while configuring the experimental setup, it was decided to use two strategies. First, funPSF was applied to the entire dataset, generating one single model regardless of the day of the week it had to be forecasted. Second, seven different models were generated, one per day, in order to better discover pattern sequences associated with different days of the week and festivities. It will be referred to as the 7-funPSF model in the rest of the paper.

The first parameter that must be calculated is the number of clusters that funHDDC must create, K . Values for K from 2–10 were considered in this study. The application of the majority voting system proposed in Section 3 yielded $K = 4$. Table 3 summarizes the results of all the indices individually proposed for the best and second best values. It can be seen that Dunn and BIC proposed $K = 4$ as the best values, and Davies–Bouldin and silhouette proposed $K = 3$. Since no K had more votes than others, the second best values had to be observed. In this case, both Davies–Bouldin and silhouette also pointed to $K = 4$ as the best election, whereas Dunn proposed $K = 3$ and BIC $K = 6$. Therefore,

since $K = 4$ obtained four votes, it became the most voted K and was selected as the number of clusters to be created.

Table 3. Determining the optimal number of clusters.

K	Silhouette	Dunn	Davies–Bouldin	BIC
Best	3	4	3	4
Second best	4	3	4	6

The second, and last, parameter required by funPSF is the length of the pattern sequence to be search for, W . Values for W from 2–10 were considered in this study. The cross-validation approach applied in the training set discovered that $W = 2$ was the best value ($MAE = 56.12$ and $MAPE = 3.98\%$).

Moreover, ANN and ARIMA models have been applied. Both of them have been trained with the same training and test sets. The configuration found to be optimal was: (i) for ANN: 24 input neurons, one hidden layer, backpropagation, and the sigmoid activation function; (ii) ARIMA(0, 1, 1).

Table 4 shows a performance summary for all four considered methods, and both MAE and MAPE errors. It can be seen that 7-funPSF clearly outperformed all of them. It can be seen that 7-funPSF obtained $MAE = 68.33$ on average. This value represents a 28.07% improvement if compared to funPSF. Furthermore, the improvement reached 35.80% and 53.37% if compared to ANN and ARIMA, respectively. Analogously, 7-funPSF obtained the best MAPE: 5.63%. The relative reported improvements were 29.15%, 42.47%, and 58.00% when compared to funPSF, ANN, and ARIMA, respectively.

Table 4. Summary of the achieved results in terms of mean absolute error (MAE) (MW), mean absolute percentage error (MAPE) (%), and Root Mean Squared Error (RMSE).

K	funPSF	7-funPSF	ANN	ARIMA
MAE	95.85	68.33	106.44	146.56
MAPE	7.94	5.63	9.78	13.40
RMSE	123.92	93.18	141.20	142.37

A thorough analysis of how MAE, MAPE, and RMSE behaved in all methods is reported in Tables 5–13. Table 5 shows how MAE was hourly distributed. It can be seen that 7-funPSF obtained lower errors for every hour, and overall, the error remained constant (except for Hours 19–21, where a moderate error increase can be seen). By contrast, ARIMA exhibited the worst behavior for 23 out of 24 h (all of them except for Hour 1). funPSF showed better performance than ANN in almost all the hours, except for Hours 8–13. However, ANN’s performance was very poor for the nighttime (Hours 1–6).

Table 5. MAE summary for hours of the day.

Hour	funPSF	7-funPSF	ANN	ARIMA
1	61.17	59.89	148.53	123.04
2	53.81	53.75	88.75	116.14
3	48.52	48.05	76.05	103.54
4	46.84	45.62	91.62	103.48
5	50.38	45.85	71.85	99.74
6	61.46	49.94	96.94	120.84
7	85.27	52.78	80.78	135.63
8	119.46	59.30	92.30	156.46
9	128.83	64.36	93.36	159.11
10	122.46	62.71	109.71	157.90
11	114.57	61.20	92.20	148.67
12	112.18	61.14	86.14	153.19
13	107.36	64.56	111.56	148.76
14	108.44	71.09	101.09	161.72
15	112.54	76.18	101.18	161.23
16	113.93	79.46	127.46	161.64
17	110.22	78.31	125.31	155.12
18	105.02	76.23	122.23	150.75
19	121.00	102.10	135.10	179.46
20	131.11	115.82	159.82	202.16
21	119.38	98.31	124.31	183.41
22	99.10	75.52	107.52	158.77
23	86.88	68.33	98.33	139.92
24	80.42	69.48	112.48	136.74
Average	95.85	68.33	106.44	146.56

Another relevant piece of information that operators need to know is how algorithms behave in seasonal terms. For this reason, Table 6 shows how MAE was distributed over months. The behavior was similar for all studied methods. They reached the worst values for winter time (December and January) and for the beginning of the summer (June). 7-funPSF, again, obtained the best results for every month, whereas ARIMA obtained the worst results for all months. ANN slightly outperformed funPSF only in May (104.27 versus 106.77).

The third analysis carried out consisted of evaluating how the algorithms behaved for different day typologies. Specifically, Table 7 analyzes MAE values distinguishing between regular days and festivities. For regular days, MAE was almost constant for all methods. However, funPSF reported the worst value for Mondays. This fact is due to the search strategy followed, in which pattern sequences incorporate changes from weekends to working days. Fortunately, this effect is completely removed in 7-funPSF. By contrast, the situation for festivities was different. First, the reported errors were quite higher than those of regular days. Second, funPSF performed better than 7-funPSF. A possible justification is that with 7-funPSF, the size of training sets became smaller, and the forecasting of these few days should not be done by using this strategy. Overall, ANN and ARIMA kept reporting worse results than funPSF and 7-funPSF. Consequently, the development of new models for forecasting these particular days is suggested.

Table 6. MAE (MW) summary for months.

Hour	funPSF	7-funPSF	ANN	ARIMA
January	121.97	103.93	128.93	181.14
February	97.18	75.86	105.86	154.23
March	71.02	43.84	73.84	114.51
April	88.90	57.98	100.98	135.73
May	106.77	68.27	104.27	153.82
June	115.70	84.01	127.01	169.03
July	87.35	63.43	98.43	142.87
August	72.66	48.38	98.38	121.83
September	79.79	52.16	95.16	128.87
October	73.19	43.15	85.15	115.40
November	106.23	76.19	116.19	161.05
December	128.36	102.02	143.02	180.23
Average	95.85	68.33	106.44	146.56

Table 7. MAE (MW) summary for the type of day.

Day	Regular Days				Festivities			
	funPSF	7-funPSF	ANN	ARIMA	funPSF	7-funPSF	ANN	ARIMA
Monday	156.22	65.20	94.53	143.87	134.66	64.41	145.07	142.60
Tuesday	84.08	65.64	84.18	141.42	257.26	359.83	312.26	253.67
Wednesday	80.12	67.08	86.34	152.47	153.18	202.39	256.73	258.32
Thursday	89.00	73.24	101.83	133.09	129.00	133.31	142.87	201.54
Friday	87.61	76.15	111.06	140.29	79.29	106.58	121.72	163.51
Saturday	72.11	69.24	117.02	145.00	42.41	40.15	99.03	241.80
Sunday	100.00	54.80	120.34	153.32	87.33	42.70	117.77	145.61
Average	95.59	67.34	102.19	144.21	126.16	135.62	170.78	201.01

Similar conclusions can be drawn for MAPE, but with slight nuances. Table 8 reports the MAPE hourly distribution. 7-funPSF achieved the best results for all 24 h, with constant error. However, the behavior for ANN and ARIMA was quite different since their associated MAPE oscillated, combining small with high errors. Nevertheless, ARIMA reached the worst results for 21 out of 24 h, and ANN only slightly outperformed funPSF in four hours.

MAPE results in terms of months had no remarkable situations (see Table 9). 7-funPSF obtained the best results for all months; funPSF obtained the second best results for 11 out of 12 months (December being the worst month, even worse than ANN and ARIMA); ARIMA reached the worst results for 11 out of 12 months; and ANN achieved results better than ARIMA, but worse than funPSF in 11 out of 12 months. Overall, funPSF, 7-funPSF, and ANN showed constant errors over time (no high fluctuations were reported throughout the different months).

MAPE differences between regular days and festivities are shown in Table 10. It is worth noting that 7-funPSF and funPSF obtained quite constant values for regular days and obtained the best and second best values for all considered measures. ANN and ARIMA showed higher fluctuations in their MAPE and reported the second worst and worst values for all days (except for two cases). As for festivities, again, funPSF showed better performance than the remaining methods, on average. It reached the best value in five out of seven days. 7-funPSF only obtained better results for two cases. ANN and ARIMA alternated between the worst and second worst results. Broadly speaking, the reported errors for festivities were quite larger than those of regular days, thus suggesting the development of particular models for these kinds of days.

Finally, Tables 11–13 report the accuracy in terms of RMSE. Almost the same conclusions can be drawn as those of MAPE, since both metrics are similar (except for the square root). Table 11 shows RMSE for every hour in the day. All methods exhibited two differentiated trends: from 4:00–21:00,

RMSE was constantly increasing; however, from 21:00–4:00, errors decreased. There were no hours apparently in which the algorithms worked particularly well. Analogously, it seems that there were no hours in which the errors reached values that were particularly high. 7-funPSF obtained the best results in 21 out of 24 h. The remaining three best results were obtained by PSF (from 1:00–3:00). Neither ANN nor ARIMA reached the smallest error at any hour.

Table 8. MAPE summary for the hour of the day (%).

Hour	funPSF	7-funPSF	ANN	ARIMA
1	5.33	5.21	5.38	10.82
2	5.12	5.10	8.48	13.00
3	4.88	4.83	7.10	9.68
4	4.86	4.71	5.80	13.44
5	5.27	4.77	5.66	8.29
6	6.34	5.09	7.23	10.60
7	8.49	5.17	10.72	11.91
8	11.33	5.46	5.86	15.70
9	11.48	5.63	15.23	20.94
10	10.29	5.21	7.28	14.34
11	9.29	4.92	8.43	11.30
12	8.86	4.78	10.43	11.08
13	8.39	4.98	13.35	17.46
14	8.48	5.46	6.83	18.41
15	8.92	5.91	14.97	13.89
16	9.11	6.21	6.40	11.11
17	8.86	6.18	11.69	14.28
18	8.39	6.03	8.56	13.70
19	9.32	7.94	8.39	15.56
20	9.76	8.75	17.16	16.40
21	8.37	6.99	16.82	17.77
22	6.82	5.24	8.43	8.02
23	6.29	4.97	9.97	14.23
24	6.34	5.49	14.55	9.54
Average	7.94	5.63	9.78	13.40

Table 9. MAPE summary for the month (%).

Month	funPSF	7-funPSF	ANN	ARIMA
January	10.17	8.75	14.10	16.24
February	8.30	6.40	8.70	10.82
March	6.67	4.08	9.08	9.67
April	6.58	3.87	10.34	11.14
May	7.49	4.88	7.95	12.83
June	8.03	5.05	9.73	13.19
July	8.52	6.19	11.08	13.18
August	8.14	5.82	8.45	18.06
September	7.40	5.28	10.92	12.17
October	6.64	4.35	7.09	20.64
November	7.11	4.67	8.56	13.04
December	10.20	8.13	11.43	9.80
Average	7.94	5.62	9.78	13.40

Table 10. MAPE summary for the type of day (%).

Day	Regular Days				Festivities			
	funPSF	7-funPSF	ANN	ARIMA	funPSF	7-funPSF	ANN	ARIMA
Monday	12.25	5.28	19.30	25.61	10.51	5.60	20.18	24.33
Tuesday	6.53	5.24	13.10	7.09	32.55	44.95	46.51	44.83
Wednesday	6.28	5.27	8.00	20.27	15.32	20.47	23.24	34.97
Thursday	6.93	5.74	13.55	18.16	13.65	14.08	28.26	20.71
Friday	6.91	6.01	11.02	14.32	6.98	9.44	11.88	24.93
Saturday	6.27	5.82	9.34	15.72	4.00	3.56	18.84	13.31
Sunday	9.97	5.06	14.26	10.75	8.64	3.98	20.87	17.29
Average	7.88	5.49	12.65	15.99	13.09	14.58	24.25	25.77

Table 11. RMSE summary for the hour of the day.

Hour	funPSF	7-funPSF	ANN	ARIMA
1	79.11	79.21	118.63	119.41
2	73.00	74.32	108.83	107.84
3	68.20	69.02	98.81	103.20
4	66.16	65.77	96.57	101.87
5	68.27	64.40	95.65	103.25
6	77.45	66.53	102.11	104.87
7	100.52	67.02	116.15	114.47
8	139.49	79.47	149.05	143.03
9	152.67	86.58	156.51	159.09
10	148.60	86.28	165.66	164.56
11	140.82	84.56	152.15	157.58
12	138.15	84.81	134.84	138.15
13	134.03	89.64	164.42	167.23
14	136.79	98.12	143.62	143.91
15	142.12	105.26	158.49	158.74
16	144.23	109.68	155.01	160.63
17	139.65	106.87	159.44	159.69
18	131.36	99.80	164.19	165.10
19	148.13	125.33	160.33	161.06
20	159.52	141.37	187.04	188.27
21	149.88	124.69	159.83	156.41
22	128.41	98.02	163.12	162.75
23	111.13	88.72	133.68	133.42
24	101.56	89.14	144.72	142.53
Average	123.92	93.18	142.37	141.20

As for Table 12, it summarizes the estimated RMSE distributed over months. As expected, 7-funPSF obtained the best results for all twelve months. Only RMSE for May, November, and December exceeded 100, contrary to what the other methods reached: PSF only obtained RMSE less than 100 in three months, and ARIMA and ANN in two months. April and May were months that were particularly difficult to predict for ANN and ARIMA, with RMSE greater than 200. These values were quite unexpected, especially taking into account that the worst value for PSF was less than 150. Anyway, the behavior shown matched that of MAPE or MAE.

Table 13 summarizes RMSE disaggregated between regular days and festivities. It is especially remarkable that RMSE reached values that were extremely high for Tuesday and Wednesday when these days were festivities. This fact can be explained because they belong to central days of the week, and therefore, it is difficult to forecast such a pattern without external information. Overall, except for these two cases, the results were smooth, and no remarkable differences were reported between the best and worst values for all the methods.

Table 12. RMSE summary for the month.

Month	funPSF	7-funPSF	ARIMA	ANN
January	121.57	99.60	128.53	131.57
February	88.62	56.62	90.13	76.31
March	112.70	78.87	131.02	130.85
April	134.19	89.88	199.27	207.99
May	147.41	107.42	212.22	234.47
June	112.48	80.89	118.75	111.29
July	93.93	64.13	109.20	108.78
August	100.78	67.52	109.69	108.06
September	90.38	56.48	98.64	94.92
October	136.24	99.03	166.41	156.41
November	166.16	136.69	174.10	168.03
December	150.96	134.33	170.44	166.45
Average	123.92	93.18	142.37	141.20

Table 13. RMSE summary for the type of day.

Day	Regular Days				Festivities			
	funPSF	7-funPSF	ANN	ARIMA	funPSF	7-funPSF	ANN	ARIMA
Monday	185.06	86.96	197.24	201.66	160.26	89.33	182.31	190.01
Tuesday	109.87	88.84	112.19	117.92	297.31	396.56	388.40	376.32
Wednesday	103.08	90.92	111.63	118.21	181.39	231.48	247.91	262.93
Thursday	116.50	98.37	124.97	124.99	151.17	161.60	164.88	170.75
Friday	110.64	100.38	115.01	118.37	102.80	141.73	153.04	160.24
Saturday	93.99	92.06	100.87	102.55	55.12	56.84	70.32	88.32
Sunday	123.27	76.29	131.36	132.60	105.21	58.61	110.67	120.68
Average	120.34	90.54	127.61	130.90	150.47	162.30	188.22	195.61

Finally, a test of statistical significance was carried out. First, the D'Agostino–Pearson test [49] concluded that data obtained for this study did not meet the criteria of normality ($p < 0.005$). For this reason, a non-parametric test was selected to validate statistically the differences in the efficiency for all models. The non-parametric process followed was described in [50] and involved the use of the Friedman test. After executing the test, the obtained p -value was less than 0.0001. Therefore, the null hypothesis was rejected, and it can be concluded that the differences in the models were statistically significant.

5. Conclusions

A novel algorithm to forecast functional time series was proposed in this paper, as a result of the gap found after reviewing related literature. The main principle underlying the algorithm lies in the discovery of patterns within the historical data. To achieve such a goal, a functional clustering algorithm is first applied to the time series. This step results in transforming the time series into a sequence of labels, thus discretizing the original series. Later, a sequence of labels of arbitrary length, preceding the sample to be forecasted, is searched for. As the last step, the values occurring after matching the pattern sequence and the historical data are averaged, with weights associated with their distance. Real-world data related to electricity demand were used to assess the performance of the algorithm. Very promising results were reported and compared to other methods. It is also shown that the proposed approach obtained error values almost constant over time. The code is available for researchers in a ready-to-use R package (available upon acceptance). The main limitation of the proposed approach lies in its dependency on seasonal patterns within data: for chaotic time series or just for time series with no clear daily or seasonal patterns, funPSF might not work as desired. Alternatively, this method provides fast and accurate forecasts in the context of electricity demand.

Future works are directed towards the development of novel funPSF models particularly adapted to dealing with anomalous data. The handling of multivariate functional time series is also a challenging task that will be addressed. In this sense, clustering algorithms to analyze multivariate data, as well as more suitable metrics must be used to fulfil such a task.

Author Contributions: Conceptualization, F.M.-Á. and J.J.; methodology, F.M.-Á. and J.J.; software, G.A.-C. and A.S.; writing, F.M.-Á. and J.J.; funding acquisition, F.M.-Á.

Funding: The authors would like to thank the Spanish Ministry of Economy and Competitiveness for the support under Projects TIN2014-55894-C2-R, P12-TIC-1728, and TIN2017-88209-C2-1-R. Francisco Martínez-Álvarez acknowledges support from José Castillejo Program CAS17/00379.

Acknowledgments: The authors also thank Prof. Jairo Cugliari for providing information about the datasets.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

Algorithm

PSF	Pattern Sequence Forecasting
funPSF	Functional Pattern Sequence Forecasting
ANN	Artificial Neural Networks
ARIMA	Autoregressive Integrated Moving Average
ARCH	Autoregressive Conditional Heteroskedasticity
ARMAX	Autoregressive Moving Average with Exogenous inputs model
EM	Expectation-Maximization
ANFIS	Adaptive Neuro-Fuzzy Inference System

Variables

K	Number of clusters
W	Length of the window
D	Dataset
L	Labeled dataset
T	Test set
S_W	Pattern sequence
M	Number of matches
h	Prediction horizon

Metrics

MAE	Mean Absolute Error
MAPE	Mean Absolute Error
RMSE	Root Mean Squared Error

References

1. Ramsay, J.; Silverman, B.W. *Functional Data Analysis*; Springer: Berlin, Germany, 2005.
2. Portela-González, J.; Munoz, A.; Alonso-Pérez, E. Forecasting Functional Time Series with a New Hilbertian ARMAX Model: Application to Electricity Price Forecasting. *IEEE Trans. Power Syst.* **2018**, *33*, 289–341.
3. Horváth, L.; Kokoszka, P. Functional Time Series. *Springer Ser. Stat.* **2012**, *200*, 289–341.
4. Martínez-Álvarez, F.; Troncoso, A.; Riquelme, J.C.; Aguilar-Ruiz, J.S. Energy time series forecasting based on pattern sequence similarity. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 1230–1243. [[CrossRef](#)]
5. Bokde, N.; Asencio-Cortés, G.; Martínez-Álvarez, F.; Kulat, K. PSF: Introduction to R Package for Pattern Sequence Based Forecasting Algorithm. *R J.* **2017**, *9*, 324–333.
6. Bouveyron, C.; Jacques, J. Model-based Clustering of Time Series in Group-specific Functional Subspaces. *Adv. Data Anal. Classif.* **2011**, *5*, 281–300. [[CrossRef](#)]
7. Schmutz, A.; Jacques, J.; Bouveyron, C.; Chèze, L.; Martin, P. Clustering Multivariate Functional Data in Group-Specific Functional Subspaces, 2017. Available online: <https://hal.inria.fr/hal-01652467/> (accessed on 14 September 2018).

8. Schmutz, A.; Jacques, J.; Bouveyron, C. funHDDC: Univariate and Multivariate Model-Based Clustering in Group-Specific Functional Subspaces; R package version 2.2.0, CRAN Repository. Available online: <https://rdrr.io/cran/funHDDC/> (accessed on 7 December 2018).
9. Cincotti, S.; Gallo, G.; Ponta, L.; Raberto, M. Modelling and forecasting of electricity spot-prices: Computational intelligence vs. classical econometrics. *AI Commun.* **2014**, *27*, 301–314.
10. Hyndman, R.J.; Shang, H.L. Forecasting functional time series. *J. Korean Stat. Soc.* **2009**, *38*, 199–211. [[CrossRef](#)]
11. Lippi, M.; Bertini, M.; Frasconi, P. Short-Term Traffic Flow Forecasting: An Experimental Comparison of Time-Series Analysis and Supervised Learning. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 871–882. [[CrossRef](#)]
12. Hormann, S.; Horvath, L.; Reeder, R. A functional version of ARCH model. *Econ. Theory* **2013**, *29*, 267–288. [[CrossRef](#)]
13. Canale, A.; Vantini, S. Constrained functional time series: Applications to the Italian gas market. *Int. J. Forecast.* **2016**, *32*, 1340–1351. [[CrossRef](#)]
14. Shang, H.L. Functional time series forecasting with dynamic updating: An application to intraday particulate matter concentration. *Econ. Stat.* **2017**, *1*, 184–200. [[CrossRef](#)]
15. Kearney, F.; Cummins, M.; Murphy, F. Forecasting implied volatility in foreign exchange markets: A functional time series approach. *Eur. J. Financ.* **2017**, *24*, 1–14. [[CrossRef](#)]
16. Yasmeen, F.; Hameed, S. Forecasting of Rainfall in Pakistan via Sliced Functional Times Series (SFTS). *World Environ.* **2018**, *8*, 1–14.
17. Bebarta, D.K.; Bisoi, R.; Dash, P.K. Locally recurrent functional link fuzzy neural network and unscented H-infinity filter for short-term prediction of load time series in energy markets. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Bhubaneswar, India, 15–17 October 2015; pp. 1394–1399.
18. Qiu, X.; Suganthan, P.N.; Amaratunga, G.A.J. Electricity load demand time series forecasting with Empirical Mode Decomposition based Random Vector Functional Link network. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Budapest, Hungary, 9–12 October 2016; pp. 1394–1399.
19. Portela-González, J.; Munoz, A.; Sánchez-Úbeda, E.F.; García-González, J.; González, R. Residual demand curves for modeling the effect of complex offering conditions on day-ahead electricity markets. *IEEE Trans. Power Syst.* **2017**, *32*, 50–61. [[CrossRef](#)]
20. Nhabangue, M.F.C.; Pillai, G.N.; Sharma, M.L. Chaotic time series prediction with functional link extreme learning ANFIS (FL-ELANFIS). In Proceedings of the IEEE International Conference on Power, Instrumentation, Control and Computing, Thrissur, India, 18–20 January 2018; pp. 1–6.
21. Bokde, N.; Troncoso, A.; Asencio-Cortés, G.; Kulat, K.; Martínez-Álvarez, F. Pattern sequence similarity based techniques for wind speed forecasting. In Proceedings of the International work-conference on Time Series, Granada, Spain, 18–20 September 2017; pp. 786–794.
22. Wang, Z.; Koprinska, I.; Rana, M. Pattern sequence-based energy demand forecast using photovoltaic energy records. In Proceedings of the International Conference on Artificial Neural Networks, Sardinia, Italy, 11–14 September 2017; pp. 486–494.
23. Gupta, A.; Bokde, N.; Kulat, K.D. Hybrid Leakage Management for Water Network Using PSF Algorithm and Soft Computing Techniques. *Water Resour. Manag.* **2018**, *32*, 1133–1151. [[CrossRef](#)]
24. Martínez-Álvarez, F.; Troncoso, A.; Riquelme, J.C.; Aguilar-Ruiz, J.S. Discovery of motifs to forecast outlier occurrence in time series. *Pattern Recog. Lett.* **2011**, *32*, 1652–1665. [[CrossRef](#)]
25. Fujimoto, Y.; Hayashi, Y. Pattern sequence-based energy demand forecast using photovoltaic energy records. In Proceedings of the IEEE International Conference on Renewable Energy Research and Applications, Nagasaki, Japan, 11–14 November 2012; pp. 1–6.
26. Koprinska, I.; Rana, M.; Troncoso, A.; Martínez-Álvarez, F. Combining pattern sequence similarity with neural networks for forecasting electricity demand time series. In Proceedings of the IEEE International Joint Conference on Neural Networks, Dallas, TX, USA, 4–9 August 2013; pp. 940–947.
27. Shen, W.; Babushkin, V.; Aung, Z.; Woon, W.L. An ensemble model for day-ahead electricity demand time series forecasting. In Proceedings of the International Conference on Future Energy Systems, Berkeley, CA, USA, 21–24 May 2013; pp. 51–62.

28. Jin, C.H.; Pok, G.; Park, H.W.; Ryu, K.H. Improved pattern sequence-based forecasting method for electricity load. *IEEE Trans. Electr. Electron. Eng.* **2014**, *9*, 670–674. [[CrossRef](#)]
29. Jacques, J.; Preda, C. Functional data clustering: A survey. *Adv. Data Anal. Classif.* **2014**, *8*, 231–255. [[CrossRef](#)]
30. Yamamoto, M. Clustering of functional data in a low-dimensional subspace. *Adv. Data Anal. Classif.* **2012**, *6*, 219–247. [[CrossRef](#)]
31. Antoniadis, A.; Brossat, X.; Cugliari, J.; Poggi, J.M. Clustering Functional Data using Wavelet. *Int. J. Wavel. Multiresolut. Inf. Process.* **2013**, *11*. [[CrossRef](#)]
32. Paparoditis, E.; Sapatinas, T. Short-term load forecasting: The similar shape functional time-series predictor. *IEEE Trans. Power Syst.* **2013**, *28*, 3818–3825. [[CrossRef](#)]
33. Chaouch, M. Clustering-based improvement of nonparametric functional time series forecasting: Application to intra-day household-level load curves. *IEEE Trans. Smart Grid* **2014**, *5*, 411–419. [[CrossRef](#)]
34. Antoniadis, A.; Brossat, X.; Cugliari, J.; Poggi, J.M. Clustering Functional Data on Convex Function Spaces. *Methodol. Appl. Stat. Inference* **2016**, *16*, 105–114.
35. Kwon, A.; Ouyang, M. Clustering of Functional Data by Band Depth. In Proceedings of the International Conference on Bio-inspired Information and Communications Technologies, New York, NY, USA, 3–5 December 2016; pp. 510–515.
36. Rivera-García, D.; García-Escudero, L.A.; Mayo, A.; Ortega, J. Robust Clustering for Time Series Using Spectral Densities and Functional Data Analysis. In Proceedings of the International Work-Conference on Artificial Neural Networks, Cadiz, Spain, 14–16 June 2017; pp. 142–154.
37. Ieva, F.; Paganoni, A.M.; Tarabelloni, N. Covariance-based Clustering in Multivariate and Functional Data Analysis. *J. Mach. Learn. Res.* **2016**, *17*, 1–21.
38. Park, J.; Ahn, J. Clustering multivariate functional data with phase variation. *Biometrics* **2017**, *73*, 324–333. [[CrossRef](#)] [[PubMed](#)]
39. Tokushige, S.; Yadohisa, H.; Inada, K. Crisp and fuzzy k-means clustering algorithms for multivariate functional data. *Comput. Stat.* **2017**, *22*, 1–16. [[CrossRef](#)]
40. Sabo, K.; Scitovski, R. An approach to cluster separability in a partition. *Inf. Sci.* **2015**, *305*, 208–218. [[CrossRef](#)]
41. Kaufman, L.; Rousseeuw, P.J. *Finding Groups in Data: An Introduction to Cluster Analysis*; Wiley: Hoboken, NJ, USA, 1990.
42. Dunn, J. Well separated clusters and optimal fuzzy partitions. *J. Cybern.* **1974**, *4*, 95–104. [[CrossRef](#)]
43. Davies, D.L.; Bouldin, D.W. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *1*, 224–227. [[CrossRef](#)]
44. Schwarz, G.E. Estimating the dimension of a model. *Ann. Stat.* **1978**, *6*, 461–464. [[CrossRef](#)]
45. Wu, C.F.J. On the convergence properties of the EM algorithm. *Ann. Stat.* **1983**, *11*, 95–103. [[CrossRef](#)]
46. Bergmeir, C.; Hyndman, R.J.; Koo, B. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Comput. Stat. Data Anal.* **2018**, *120*, 70–88. [[CrossRef](#)]
47. Capizzi, G.; Lo-Sciuto, G.; Napoli, C.; Tramontana, E. Advanced and Adaptive Dispatch for Smart Grids by Means of Predictive Models. *IEEE Trans. Smart Grid* **2017**, *6*, 6684–6691. [[CrossRef](#)]
48. Gelper, S.; Fried, R.; Croux, C. Robust forecasting with exponential and Holt-Winters smoothing. *J. Forecast.* **2010**, *29*, 285–300.
49. D’Agostino, R.B.; Belanger, A.; D’Agostino, R.B., Jr. A suggestion for using powerful and informative tests of normality. *Am. Stat.* **1990**, *44*, 316–321.
50. García, S.; Herrera, F. An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. *J. Mach. Learn. Res.* **2008**, *9*, 2677–2694.

