Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 5129735, 13 pages
https://doi.org/10.1155/2018/5129735

Research Article

Detection of Anomalies in Water Networks by Functional Data Analysis

Laura Millán-Roures,[1] Irene Epifanio ,[1,2] and Vicente Martínez[1,2]

[1] *Dept. Matemàtiques, Universitat Jaume I, 12071 Castelló, Spain*
[2] *Institut de Matemàtiques i Aplicacions de Castelló, Spain*

Correspondence should be addressed to Irene Epifanio; epifanio@uji.es

Received 21 January 2018; Revised 2 May 2018; Accepted 24 May 2018; Published 21 June 2018

Academic Editor: Anna Vila

A functional data analysis (FDA) based methodology for detecting anomalous flows in urban water networks is introduced. Primary hydraulic variables are recorded in real-time by telecontrol systems, so they are functional data (FD). In the first stage, the data are validated (false data are detected) and reconstructed, since there could be not only false data, but also missing and noisy data. FDA tools are used such as tolerance bands for FD and smoothing for dense and sparse FD. In the second stage, functional outlier detection tools are used in two phases. In Phase I, the data are cleared of anomalies to ensure that data are representative of the in-control system. The objective of Phase II is system monitoring. A new functional outlier detection method is also proposed based on archetypal analysis. The methodology is applied and illustrated with real data. A simulated study is also carried out to assess the performance of the outlier detection techniques, including our proposal. The results are very promising.

1. Introduction

The objective of a water distribution system is to supply water to meet user demand. Therefore, the water distribution system should be monitored in order to identify any fault in the system (Rodriguez et al. [1]). Detecting burst pipes, exceptional water use, or any other anomalous behavior in the water supply network is very important problem. Early detection of any anomaly in the water network is very valuable. Nowadays, thanks to technology we are able to record primary hydraulic variables in real-time. Nevertheless, all these data are useless without adequate statistical processing that enables information to be extracted for decision-making (Palau et al. [2]).

The recording of data by telecontrol systems allows data to be saved every few minutes. However, this type of systems usually presents two kinds of problems (Quevedo et al. [3]). On the one hand, poor maintenance of flow meters generates false data, i.e., somewhat noisy data, sometimes disproportionately so. On the other hand, communication problems with the sensor result in missing data. Both problems have to be solved during the first stage, which is referred to as validation/reconstruction in the hydraulic literature. In the second stage, data can be used to identify atypical behaviors in terms of water use, burst pipes, or illegal connections.

For the first stage, several alternatives have been proposed. The validation methods (finding erroneous data) include elementary signal-based ("low-level") methods (Burnell [4]), as well as other model-based ("higher level") methods. These also provide a methodology for reconstructing the signal in case of missing data. For instance, time series analysis was used by Quevedo et al. [3] or Prescott and Ulanicki [5], Kalman filters by Piatyszek et al. [6], neural networks by Valentin and Denœux [7], etc. As regards burst pipe detection, different techniques have been considered, such as multivariate principal component analysis (Palau et al. [2]), artificial neural networks (Mounce et al. [8]), a Bayesian system identification methodology (Poulakis et al. [9]), or a modified cumulative sum (CUSUM) test (Misiunas et al. [10]). In industrial quality control, the statistical monitoring of complex signals goes under the name of profile monitoring. According to Colosimo and Pacella [11], the approaches for profile monitoring share the same structure: control charts for the estimated parameters of a certain parametric model are designed.

A different point of view is taken in this paper. Note that our data are recorded every 5 minutes, so they are discretized functions. However, a continuous curve or function lies behind these data. Therefore, we propose a functional data analysis (FDA) approach for both stages. To the best of our knowledge, this is the first time this kind of data has been viewed as functional data (FD), which they are in fact.

FDA comprises statistical techniques for functional observations; i.e., a whole function is a datum. The goals of FDA are basically the same as those of any other branch of statistics. Although FDA is a relatively new field, some references have already become classics, such as Ramsay and Silverman [12], who offer an outstanding overview, Ferraty and Vieu [13], with new methodologies for studying FD nonparametrically, Ramsay and Silverman [14], who provide interesting applications in different disciplines, and Ramsay et al. [15] regarding software in this field. In the water domain, FDA has been used for analyzing water quality (Henderson [16]; Díaz-Muñiz et al. [17]; Yan et al. [18]), classifying streamflow hydrographs (Ternynck et al. [19]), or identifying major water usage patterns (Cheifetz et al. [20]), for example. Using FDA has the advantage of using all available information on shape, peak, and timing; i.e., all the information contained in the time series is taken into account.

In this paper, we propose a FDA-based method for detecting anomalous flows in urban water networks. The main novelties of this work consist of (1) treating real-time primary hydraulic variables as FD for the first time, (2) proposing a method entirely based on FDA, to validate and reconstruct those data and identify anomalies, and (3) proposing a new procedure for functional outlier detection based on functional archetype analysis (Epifanio [21]) and comparing its performance with other procedures. Section 2 presents the available data and introduces the procedures for Stages 1 and 2. A review of FDA techniques and a new proposal for functional outlier detection are also presented. In Section 3, the proposal is applied to real data and simulated data. The experiments were conducted in R (R Core Team [22]). Finally, conclusions and future work are discussed in Section 4.

2. Material and Methods

2.1. Data. Our data consist of the water inflows recorded every 5 minutes by flow meters in three district metering areas (DMA) belonging to three municipal water utilities on the eastern coast of Spain. For the sake of confidentiality, we refer to these locations as A, B, and C. There are 288 observations per day and the observations from three years (2014, 2015, and 2016) are available. Although pressure levels were also available, according to an expert in hydraulic engineering, the inflows would be more informative, so pressure levels are not considered.

It is supposed that the water demand pattern is repeated every day, but it varies at weekends, and there are also changes in demand between winter and summer (Rodriguez et al. [1]). Therefore, we decided to divide the data into 8 groups according to the four seasons and weekdays or weekends. For each group, the method is outlined in the following section. As regards holidays, we have kept them in their corresponding group, although their patterns may not correspond to weekdays or weekends. For example, Christmas Days are detected as anomalous in the set of both weekdays (years 2014 and 2015) and weekends (year 2016). The data could also be divided according the particularities of the location.

2.2. Procedure for Detecting Anomalies. The outline of the procedure (details are given in the following sections) is as follows.

Stage 1. Validation/reconstruction:

Step 1: length of the interval T (definition of the period)

Step 2: visualization (rainbow plots)

Step 3: cleaning of false data

Step 4: smoothing

Situation 1: smoothing dense FD
Situation 2: smoothing sparse FD

Stage 2. Identifying anomalies:

Phase I: iterative procedure to detect functional outliers

Phase II: system monitoring

2.3. Stage 1. Data are collected as a long time series, but as mentioned above the water demand pattern is repeated every day. The first step is to decide the length of the interval T over which functions are defined, i.e., the functional dimensionality (Ramsay and Silverman [12]). It can range from a short interval, such as 30 minutes, to a long interval (24 hours or more). Remember that data are recorded every 5 minutes (data resolution), so intervals should contain at least several observations for each function. In Section 3, the results for intervals of different lengths will be shown, although we will concentrate on medium-length results: 6 hours. Specifically, we will focus on results for night hours, from 00:00 to 6:00 a.m. since, according to an expert in hydraulic engineering, the water demand is rather stagnant during this period (Misiunas et al. [10]) and it is easier to identify anomalous behaviors.

In the next step it is advisable to examine the data by carrying out an exploratory data analysis. Visualization methods help users reason about the data and discover features that might not have been apparent using mathematical models and summary statistics, such as a different pattern over the time. A rainbow plot (Hyndman and Shang [23]) could be used, which is a simple but effective plot where data are plotted with colors according to their order in time. The colors of the functions follow the order of a rainbow, with the oldest data in red and the most recent data in violet. It is available in the R package **rainbow** (Shang and Hyndman [24]).

The advantages of an FD approach are evident in the following steps. In the fourth step, the discrete measures $y_{i1},\ldots,y_{in_i}$ recorded at time points t_{ij}, with $j = 1,\ldots,n_i$, from a functional datum i ($i = 1,\ldots,N$, N is the sample size) must be converted into a function x_i. As explained in Section 1, these values are not free from errors and smoothing should be carried out to remove those errors. Smoothing is performed by representing functions with basis functions. This has the advantage that it can be applied to data where the functions have not all been recorded at the same instant, data observations do not have to be equally spaced, and the number of sampling points can vary across cases. This is the case in the applications in question because communication problems with the sensor result in missing data.

The key point is to choose a proper basis and number of basis functions (less computation is required for a smaller number of basis functions). This is a prevalent issue in all FDA problems. Ideal basis functions should have characteristics that match those known to belong to the functions being approximated (see Ramsay and Silverman [12] for a precise and detailed explanation about smoothing FD). In addition, the individual functions can be smoothed in a different way depending on whether the number of missing data is high or low. There are therefore two possible situations.

In both situations, once the function has been expressed as a linear combination of basis functions, this allows us to evaluate the function at any desired time point and to obtain estimations for the periods with missing data and correct noisy data. However, FDA works with basis coefficients, not discretized functions, since all the information is better saved as coefficients. They provide us with the flexibility that we need together with the computational power to fit hundreds of points in a short length vector. Furthermore, coefficients allow us to express the calculations within the well-known field of matrix algebra.

Nevertheless, before moving on to the smoothing step, disproportionately noisy data (false data) should be located and removed. Smooth function assumes that a pair of adjacent data values, y_{ij} and $y_{i(j+1)}$ would be similar. Otherwise, data should be treated as multivariate rather than functional. In our application, the smoothness property makes sense.

2.3.1. Cleaning False Data. Tolerance intervals for univariate data provide limits that contain a given proportion (p) of individual observations in a population with a given level of confidence. They should not be confused with confidence intervals or prediction intervals. We can use tolerance intervals to identify unusual values.

Tolerance intervals were extended to the functional framework by Rathnayake and Choudhary [25]. They defined tolerance bands for both dense and sparse FD. They used Functional Principal Components Analysis (FPCA) in a mixed model framework to represent the measurements and the tolerance factors needed for the bands are approximated by bootstrapping. The R code is available at Rathnayake and Choudhary [25]. We have used it with $\alpha = 0.05$ and $p = 0.97$. Data points outside the tolerance bands are considered as suspected of being false data, removed, and considered as missing data.

2.3.2. Situation 1: Smoothing Dense Functional Data. When functions are densely observed, i.e., the number of missing data is low, the basis coefficients can be estimated separately for each function. This is the most common situation in our application. In the basis approach, each function x_i is expressed as a linear combination of known basis functions ϕ_k with $k = 1,\ldots,K$: $x_i(t) = \sum_{k=1}^{K} b_i^k \phi_k(t) = \mathbf{b}_i'\mathbf{\Phi}$, where $'$ stands for transpose and $\mathbf{b}_i$ indicates the vector of length K of the coefficients and $\mathbf{\Phi}$ the functional vector whose elements are the basis functions. Coefficients are estimated by ordinary least squares, although other kinds of smoothing such as the weighted least squares or regularization approaches can be considered. All of these are available in the R package **fda** (Ramsay et al. [26]). Specifically, the following expression should be minimized: $SMSSE(y|b) = \sum_{j=1}^{n_i}(y_{ij} - \sum_{k=1}^{K} b_i^k \phi_k(t_{ij})) = (\mathbf{y}_i - \mathbf{F}\mathbf{b}_i)'(\mathbf{y}_i - \mathbf{F}\mathbf{b}_i)$, where $\mathbf{F}$ is a matrix that contains the values $\phi_k(t_{ij})$. The solution is $\mathbf{b}_i = (\mathbf{F}'\mathbf{F})^{-1}\mathbf{F}'\mathbf{y}_i$, and the fitted values are $\hat{\mathbf{y}}_i = \mathbf{F}\mathbf{b}_i$.

In the application, the B-spline basis system of order 4 (cubic splines) with equally spaced knots has been selected due to its suitability for nonperiodic data (Ramsay and Silverman [12]), its great flexibility, and very fast computation. As suggested by Ramsay and Silverman [12], the number of bases K can be determined by selecting K that makes the variance decrease substantially: $s^2 = (1/(n_i - K))\sum_{j=1}^{n_i}(y_{ij} - \hat{y}_{ij})^2$. For the whole sample formed by N functions, we computed the pooled variance using 4 to 22 bases and selected the number of bases that makes it decrease substantially.

2.3.3. Situation 2: Smoothing Sparse Functional Data. When functions are sparsely observed, i.e., the number of missing data is high with long periods without data, the information from all functions should be employed to estimate the coefficients for each function (James [27]). This situation is not very common in our application, but we have used the recognized methodology developed by Yao et al. [28]: Principal components Analysis through Conditional Expectation (PACE) for illustration purposes in Section 3. Yao et al. [28] proposed a version of FPCA in which the FPC scores are framed as conditional expectations. The function $x_i(t)$ using the first K ϕ_k eigenfunctions is approximated by

$$\hat{x}_i^K(t) = \hat{\mu} + \sum_{k=1}^{K} \hat{\xi}_{ik}\hat{\phi}_k(t) \quad (1)$$

where $\hat{\mu}$ is the estimate of the mean function $E(x(t)) = \mu(t)$ and ξ_{ik} are the FPC scores. PACE uses local smoothing techniques to estimate the mean and covariance functions of the trajectories, and it was implemented by the function FPCA in the R package **fdapace** (Dai et al. [29]) (default parameters are considered in the computation). With the scores and the estimated eigenfunctions, we obtain an approximation of the functions, which can be used to predict unobserved portions of them.

2.4. Stage 2. Once the data have been cleaned, reconstructed, and smoothed we are ready to find anomalous behaviors. In statistical terms, this is equivalent to finding outliers. We distinguish two phases, as in-control chart applications (Montgomery [30]). The first phase is a retrospective analysis, where data are analyzed to find anomalies and a set of data that is free from anomalies is obtained for use in Phase II. In Phase II we monitor the process, so each successive datum is analyzed to identify whether or not it is an anomaly.

Before explaining these phases, let us review the types of functional outliers and several functional methods for detecting functional outliers. A new method for detecting functional outliers is also introduced.

2.4.1. Types of Functional Outliers. According to Febrero et al. [31], functional outliers can result from errors in measurements and recording or, despite being correctly observed functions, other causes can mean that they do not follow the same pattern as the rest of the curves. In Section 2.3.1, false data due to measurement errors are detected. In this paper, we will detect anomalous behaviors. Another question is to identify their potential root causes, which is key in statistical control quality.

Hyndman and Shang [23] distinguish two types of functional outliers: first the so-called magnitude outliers, which correspond to curves that lie outside the range of the vast majority of the data. This either could happen for a short period only (Hubert et al. [32] call these isolated outliers) or may be persistent. Persistent outliers can have the same shape as other curves, but their scale differs. Hubert et al. [32] call these amplitude outliers. The shape can even be identical but translated in time. Hubert et al. [32] call this type shift outliers. The second type is the so-called shape outliers, whose shape differs from the rest without necessarily standing out at any particular time point. Of course, outliers may exhibit a combination of these characteristics.

Unlike magnitude outliers, which are easily detected even by the naked eye, identifying shape outliers is not so easy. In fact, shape outliers may be camouflaged in the middle of the sample.

2.4.2. Methods for Detecting Functional Outliers. Several methods for functional outlier detection have been developed recently. Most of them rely on different notions of functional depth, such as Febrero et al. [33], Febrero et al. [31], Sun and Genton [34], Gervini [35], and Arribas-Gil and Romo [36], together with Hubert et al. [32] for the multivariate functional case. Others rely on robust principal components (PCs) (Hyndman and Ullah [37], Hyndman and Shang [23], and Sawant et al. [38]) and random projections (Fraiman and Svarc [39]). The R code of those methods is available in most cases. We will review them below and introduce the acronyms used to refer to them.

On the one hand, the R package **rainbow** (Shang and Hyndman [24]) implements many of those methods and others. Specifically consider the following: first, in the method by Febrero et al. [33], they use a likelihood ratio test (LRT). Second, in Febrero et al. [31] outliers are determined as functions whose depth levels are below a certain threshold. This threshold is determined by a bootstrap procedure based on either trimming (TRIM) or weighting of the sample (POND). These methods are also implemented in the R package **fda.usc** (Febrero-Bande and de la Fuente [40]). Third, in the method by Hyndman and Ullah [37] (ISFE), they use integrated square forecast errors. Fourth, the method by Rousseeuw and Leroy [41] (RMAH) uses the robust Mahalanobis distance but considering the functions as multivariate observations. Fifth, the methods that are proposed by Hyndman and Shang [23] are like the functional highest density region (HDR) boxplot. On the other hand, the R package **fda** (Ramsay et al. [26]) implements the method (FB) by Sun and Genton [34], who extend the classical boxplot to FD. The outliergram (OUG) proposed by Arribas-Gil and Romo [36] is available in the R package **roahd** (Tarabelloni et al. [42]). In the R package **mrfDepth** (Segaert et al. [43]), we find the implementation of the Functional Outlier Map (FOM) by Hubert et al. [32], improved by Rousseeuw et al. [44], which is based on functional outlyingness measures.

2.4.3. Detection of Functional Outliers by Archetype Analysis. Our proposal is based on archetype analysis (AA), which was introduced by Cutler and Breiman [45]. The objective of AA is to approximate data through a convex combination of pure or extremal types called archetypes. The premise of this is that extremes are better than central points for human interpretation (Thurau et al. [46]). Archetypes are built as a convex combination of observations. A variation of AA is archetypoid analysis (ADA), which was introduced by Vinué et al. [47]. The pure types in ADA are not a mixture of cases, but archetypoids are real cases from the sample. Archetypal analysis has been applied in different fields, but the only known references in engineering are Epifanio et al. [48], Vinué et al. [47], and Epifanio et al. [49]. AA and ADA have recently been extended to dense (Epifanio [21]) and sparse FD (Vinué and Epifanio [50]). AA can be computed with the R package **archetypes** (Eugster and Leisch [51]) and ADA with **Anthropometry** (Vinué [52]).

Besides archetypal patterns, both return the mixture coefficients contained in a $N \times p$ matrix α, where p is the number of archetypes (or archetypoids). The α coefficients indicate the amount of the contribution of each archetype (or archetypoid) to the approximation of each case. Each α_{ij} is the weight of the archetype (or archetypoid) j for the case i, and $\sum_{j=1}^{p} \alpha_{ij} = 1$ with $\alpha_{ij} \geq 0$ and $i = 1, \ldots, N$.

Archetypes are located on the boundary of the convex hull of data (Cutler and Breiman [45]). Therefore, AA and ADA are sensitive to outliers (Eugster and Leisch [53]). This is precisely what can be used to locate outliers. The basic idea is described below.

Let us assume that we have a homogeneous sample of FD, i.e., generated from a unique (unimodal) distribution. Then, it could be reasonable to think that the observed FD are between two extremal functions (two functional archetypes or archetypoids); i.e., the sample of FD belongs approximately to the band (López-Pintado and Romo [54]) given by the two extremal functions (see Figure 1 for an illustration of this idea). Therefore, the sample can be approximated by a mixture of these two extreme functions. In fact, this

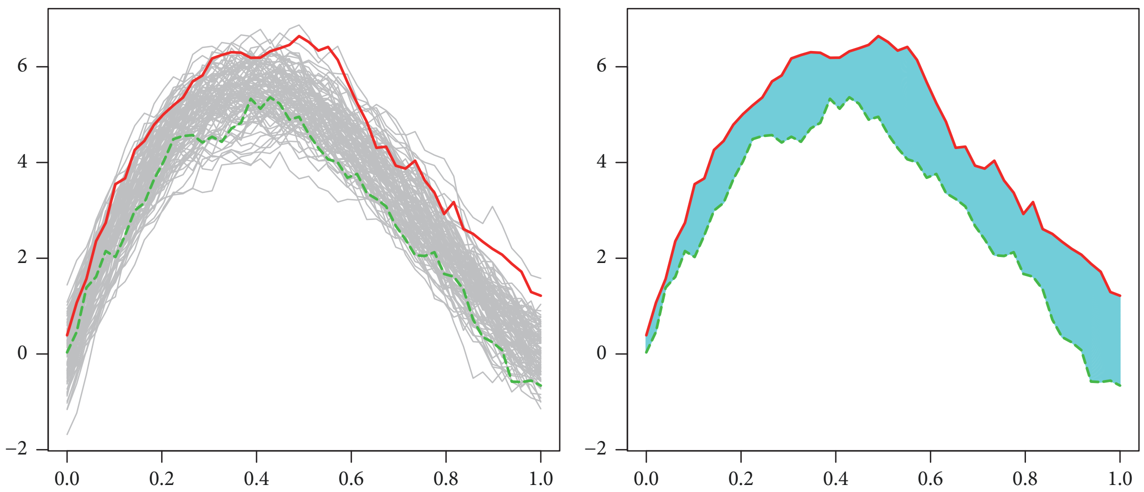


FIGURE 1: Simulated data from a unimodal distribution. Left-hand panel: functions are generated from a certain model (gray). Two archetypoids, A_1 and A_2, are computed and represented with (red) solid and (green) dashed lines. Right-hand panel: band determined by A_1 and A_2, $B(A_1, A_2) = \{(t, y) : t \in T, \min_{r=1,2} A_r(t) \le y \le \max_{r=1,2} A_r(t)\} = \{(t, y) : t \in T, y = \alpha \min_{r=1,2} A_r(t) + (1-\alpha) \max_{r=1,2} A_r(t), \alpha \in [0, 1]\}$. Most of the graphs of the other of functions are included in the light blue band given by the two archetypoids.

idea was used by D'Esposito and Ragozini [55] for ranking multivariate data. As a consequence, the α coefficients will be distributed from 0 to 1 continuously with a strictly positive density function for each archetype (or archetypoid).

Let us assume now that there is at least one outlier generated from another distribution. Then, if we consider $p = 3$ archetypes (or archetypoids), in an ideal setting one of the archetypes (or archetypoids) should be similar to the outlier (in the case of ADA, one of the archetypoids should be one of the outliers), which is an extremal function, but the other two archetypes (or archetypoids) should be the two extremal functions covering the rest of the sample. In that situation, the distribution of the α coefficients corresponding to the archetype (or archetypoid) similar to the outlier will have a special profile: only a few cases would have high values (above a certain threshold that depends on the configuration of the data). Those few cases correspond to the functions that are similar to the outlier, i.e., generated from the same distribution as the outlier, and therefore also outliers. As the other functions do not share that pattern, their α coefficients for that archetype (or archetypoid) will be small and we can observe a "hole" (an area without data) in those α values. The "hole" will separate the outliers from the rest of the sample.

If there are outliers of different types; we can compute 3 archetypes (or archetypoids) again, once the previously detected outliers have been removed from the sample, and so on. Note that if outliers are not present in the sample and we compute 3 archetypes (or archetypoids), the corresponding α coefficients will also be distributed from 0 to 1 without "holes" for each archetype (or archetypoid).

Based on these premises, our proposal is as follows. Although AA could be also used, we will focus on ADA, as the archetypoids are real cases and, therefore, in the case of outliers, one of the archetypoids should be an outlier and not a mixture of outliers, as would be the case in the AA. We refer to this method as FOADA.

(1) Compute ADA with $p = 3$ for the sample.

(2) If no "hole" is detected in the α coefficient distribution for each archetypoid, then no outlier is detected and the procedure ends.

(3) Otherwise, consider the archetypoid that has fewest cases with high (above the threshold that determines the "hole") α coefficients. Identify those cases as outliers and add them to a list of outliers. Return to Step 1, after removing the identified outliers from the sample.

This procedure depends on the detection of "holes" in the distribution of α coefficients for each archetypoid. As only 3 archetypoids are computed each time, we can visualize the distribution of α coefficients in 2D using a ternary plot without loss of information. A ternary plot makes it possible to visualize compositional 3-dimensional data in an equilateral triangle. Therefore, outliers could be detected empirically with the naked eye or with a classic multivariate detection method, such as the robust Mahalanobis distance (Rousseeuw and Leroy [41]).

Illustrative Example: Shape Outliers (Changes in the Average Shape). The following example aims to clarify the procedure. A set of $N = 100$ functions has been generated from the following model, which was used previously by Febrero et al. [31], Fraiman and Svarc [39], and Arribas-Gil and Romo [36]. $N - \lceil c \cdot N \rceil$ are generated from $X(t) = 30t(1-t)^{3/2} + \epsilon(t)$, while the remaining $\lceil c \cdot N \rceil$ functions are generated according to this contamination model: $30t^{3/2}(1-t) + \epsilon(t)$, where $t \in [0, 1]$ and $\epsilon(t)$ is a Gaussian process with zero mean and covariance function $\gamma(s, t) = 0.3\exp\{-|s - \mathrm{t}|/0.3\}$. The functions are observed at 50 equidistant points between 0 and 1. Note that those are shape outliers; their average is different.

Figure 2 shows an example of functions generated with $c = 0.02$ in the left-hand panel, the 3 archetypoids in the

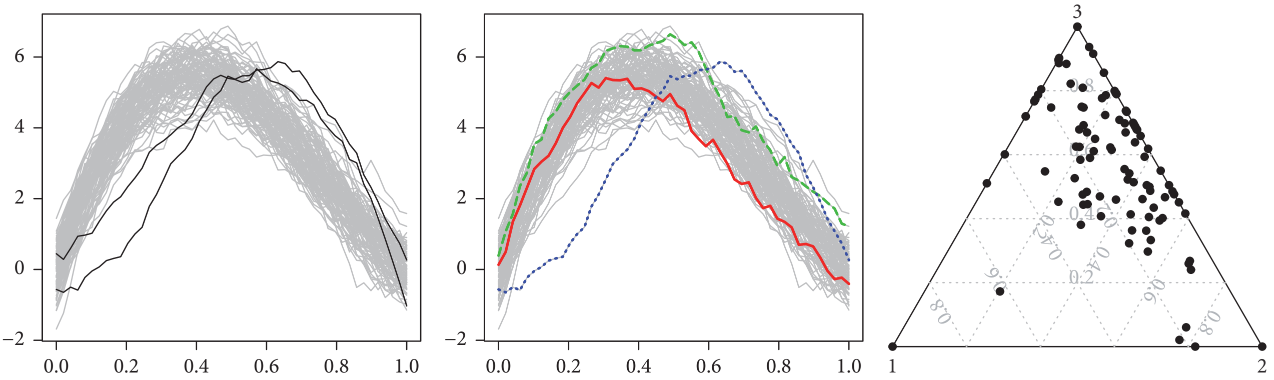


Figure 2: Simulated data with 2 outliers. Left-hand panel: functions are generated from the main model (gray) and the contamination model (black). Central panel: archetypoids are represented with (red) solid, (green) dashed, and (blue) dotted lines, respectively. Right-hand panel: ternary plot showing the α coefficients. The vertices of the triangle represent each archetypoid.

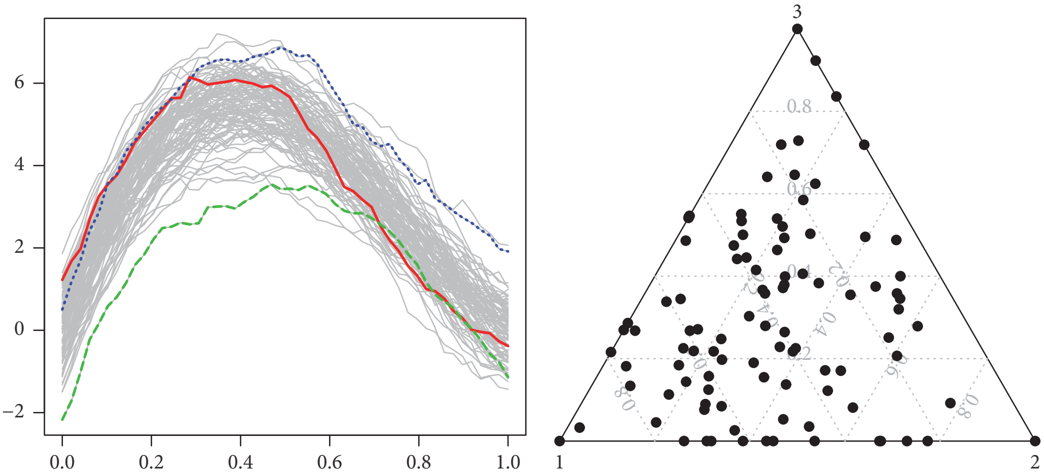


Figure 3: Simulated data with outliers removed. Left-hand panel: archetypoids are represented with (red) solid, (green) dashed, and (blue) dotted lines, respectively. Right-hand panel: ternary plot showing the α coefficients. The vertices of the triangle represent each archetypoid.

central panel, and the ternary plot in the right-hand panel. The first archetypoid is one of the outliers, while the other two archetypoids are extreme functions between which almost the entire sample is included. In the ternary plot, it is evident that almost all the components of the sample, except the two outliers, are expressed as a mixture of archetypoids 2 and 3. Only two points, the outliers, have high α values for archetypoid 1.

Now, these two outliers are removed from the sample and ADA is computed again. Figure 3 shows the 3 archetypoids obtained and the corresponding ternary plot. As before, archetypoids 2 and 3 are extreme functions resembling band functions between which almost of the sample is included. Archetypoid 1 is a function similar to archetypoid 3 in the first third of the interval and similar to archetypoid 2 in the last third of the interval. The α values are now spread out inside the triangle.

Illustrative Example: Outliers due to an Increase in Variability. Let us see how the method proceeds, when, besides shape outliers, there are also outliers because the variability is greater. Figure 4 (left-hand panel) shows the previous outliers plus two new outliers (in red). They are built by adding a white noise process to the mean of the previous main model. In the first iteration of the procedure, the shape outliers are detected and removed from the data set for the following iteration. However, no more outliers are detected in the second iteration with the raw data. Nevertheless, it should be remembered that these are functional data, so changes in variability are equivalent to changes in the derivatives. Therefore, we apply the procedure to the first derivatives of the functions in the data set (see the right-hand panel of Figure 4). In the following iteration one of the outliers is detected, while the other outlier is detected in the next iteration. No more outliers are detected. So, all the outliers are finally detected. Note that outliers in variability are shape outliers in the derivatives.

Illustrative Example: Amplitude Outliers. Figure 5 (left-hand panel) displays the previous shape outliers (in black) plus two amplitude outliers (in red), i.e., with $c = 0.02$. Therefore, the percentage of outliers in the data set is 4%. The amplitude outliers are generated by adding 2 to the mean from the main model. The amplitude outliers are detected in the

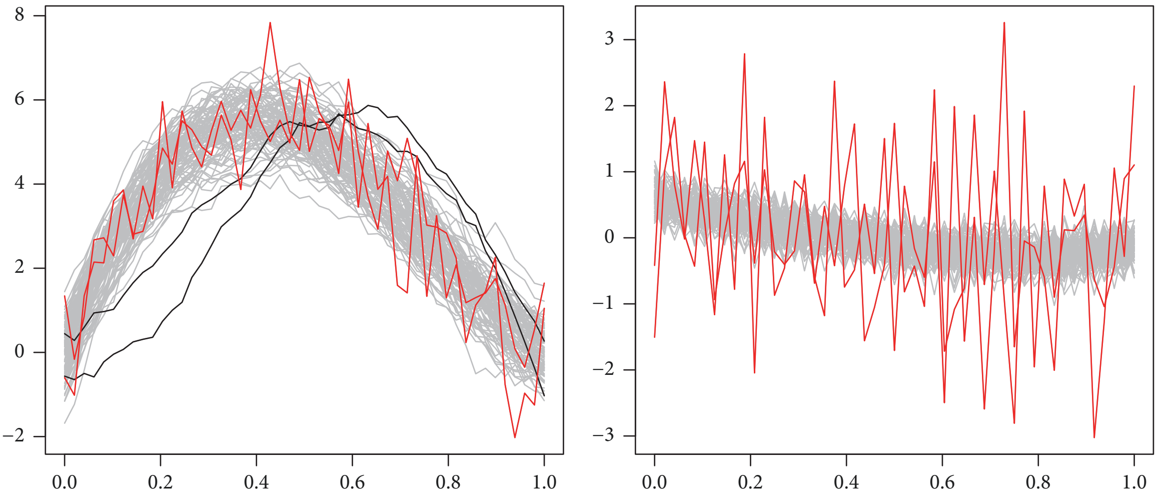


Figure 4: Simulated data (outliers in variability). Left-hand panel: functions are generated from the main model (gray) and shape outliers are in black, while outliers in variability are in red. Right-hand panel: the first derivative of the functions, once the shape outliers are removed. Outliers in variability are in red.

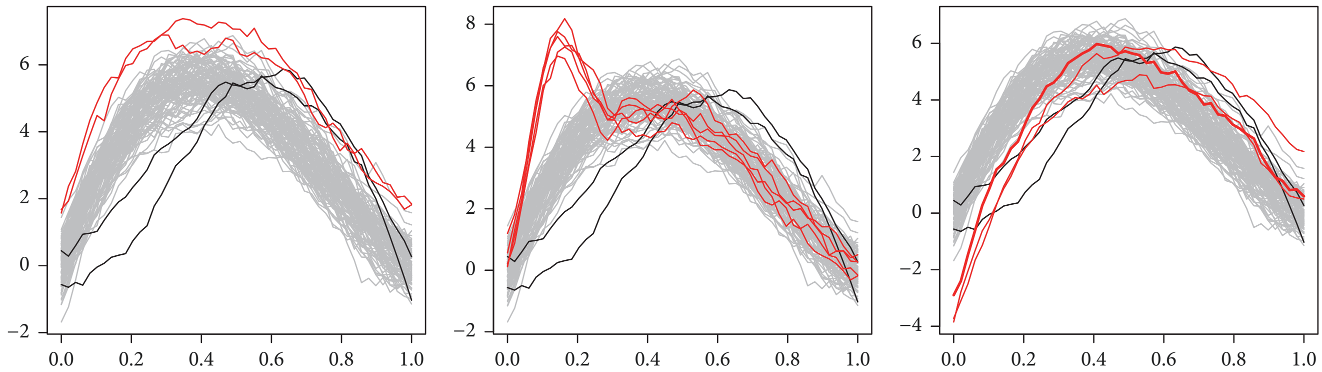


Figure 5: Simulated data (different types of outliers). Functions are generated from the main model (gray) and shape outliers are in black. Left-hand panel: outliers in amplitude are in red. Central panel: isolated outliers are in red. Right-hand panel: shift outliers are in red.

first iteration of our procedure, while the shape outliers are detected in the second iteration.

Illustrative Example: Isolated Outliers. Figure 5 (central panel) displays the previous shape outliers (in black) plus five isolated outliers (in red), i.e., with $c = 0.05$. Therefore, the percentage of outliers in the data set is 7%. The isolated outliers are generated by adding the values of a standard normal density to the first 14 observed points of the mean from the main model. The shape outliers are located in the first iteration of the procedure, while the isolated outliers are located in the following iteration.

Illustrative Example: Shift Outliers. Figure 5 (right-hand panel) shows the previous shape outliers (in black) plus three shift outliers (in red), i.e., with $c = 0.03$. Therefore, the percentage of outliers in the data set is 5%. The shift outliers are generated by translating the mean of the main model in -0.1 time units. All except one shift outliers are detected in the first iteration of the procedure. The nondetected outlier is displayed with a thicker solid red line in the right-hand panel of Figure 5.

2.4.4. Phase I. The objective of this phase is to clear the data of anomalies to ensure representative data from the in-control system. Possible functional outliers are identified using a particular method from Section 2.4.2. They should be investigated in order to determine the causes. The identified outliers are removed from the sample and the functional outlier detection method is used again. If new outliers are detected, we repeat the procedure (investigation and exclusion of outliers) until no new outliers are detected. At the end of this iterative procedure, it can be assumed that a clean sample that represents the performance of the in-control system has been obtained for use in Phase II.

In any statistical problem, having a large, representative sample is preferable to a small sample. Small samples may not cover all the information in the process and may therefore give rise to false alarms. In a somewhat similar problem, the determination of limits for multivariate case control charts, sample sizes of at least 20 and 50 are recommended (Lowry and Montgomery [56]) and even 200 would be desirable (Jensen et al. [57]).

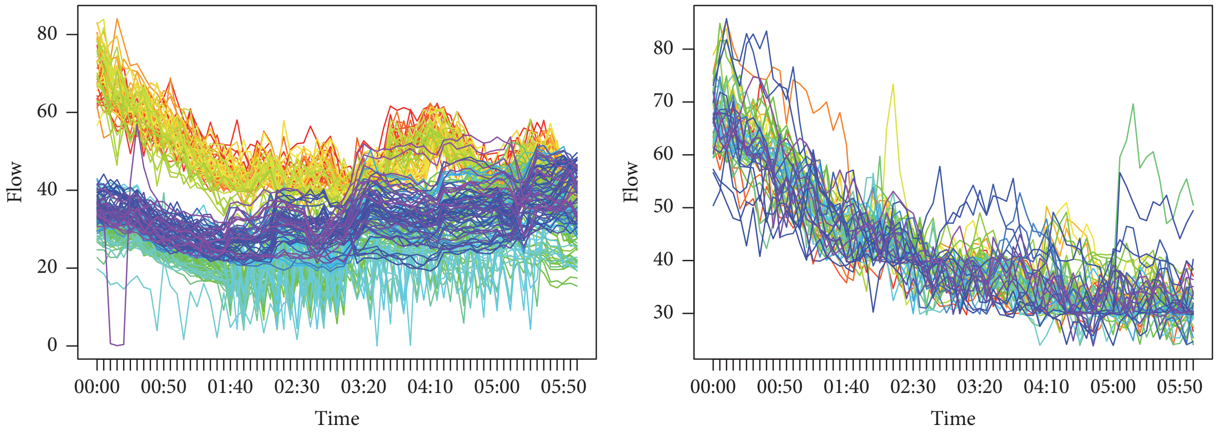


Figure 6: Rainbow plots: weekdays in summer for sector A (left) and weekend days in spring for sector B (right). Time is expressed in hours and flow in cubic meter per hour (m^3/h). Functions are ordered chronologically according to the colors of the rainbow. The oldest functions are shown in red, while the most recent functions are shown in violet.

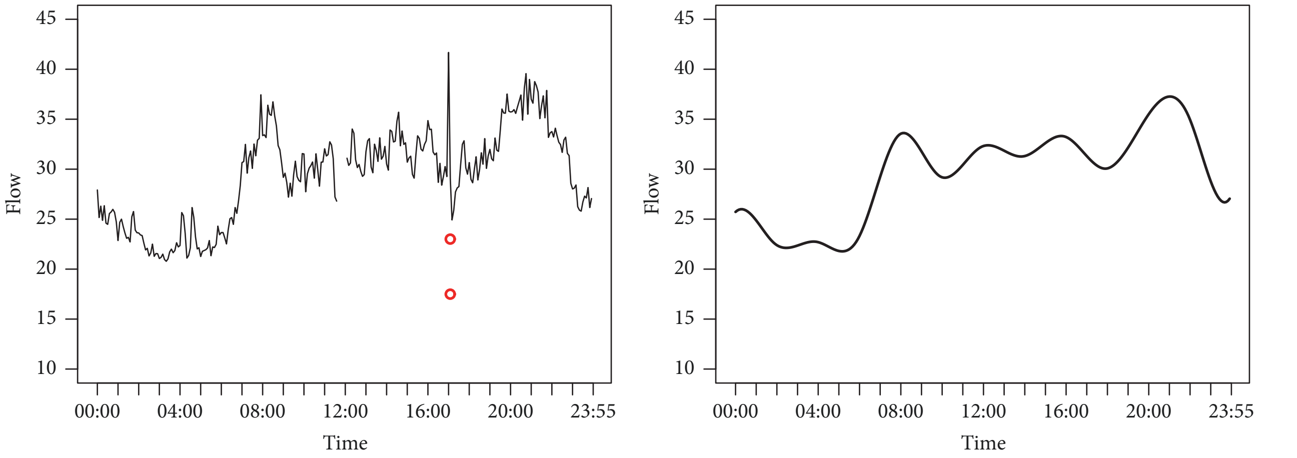


Figure 7: The left-hand panel shows raw data from 03/04/2014 with detected false data in red. It belongs to the group of weekdays in spring for sector C. The right-hand panel shows the smoothed data. Time is expressed in hours and flow in cubic meter per hour (m^3/h).

2.4.5. Phase II. The objective of this phase is system monitoring. Each new functional datum is cleaned for false data and smoothed as explained in Stage 1. It is then added to the sample obtained in Phase I and a functional outlier detection method is used to test whether this new datum is atypical. If it is detected as anomalous, it should be investigated. Otherwise, the sample obtained in Phase I can be updated by adding the new datum. In this way, the sample size can be increased. In any case, data should be visualized periodically as in Step 2 of Stage 1.

3. Results

3.1. Real Case Study. The procedure introduced in Section 2.2 has been applied to our data. For the sake of conciseness, we have decided to illustrate each of the components of the procedure. As previously discussed, unless it is stated otherwise, we focus on the results for night hours, from 00:00 to 6:00 a.m. The results for other periods will be shown in some points for illustration purposes. We begin with some examples of results of Step 2, Stage 1, and follow the illustration of each element of the procedure in order.

Figure 6 shows rainbow plots for two different statuses. In the first one, we can see how the colors show a different distribution according to year; in 2014 the functional distribution (red-yellow curves) is different from 2015 and 2016 (green-violet curves) due to external causes. They could be faults in the pipelines or a modification of the limits of the sector (opening/closing of border valves), a variation in the supply pressure of the sector, etc. This means that only data from 2015 and 2016 could be used for that sector. On the other hand, in the right-hand plot, no temporal pattern appears, so we use data from all the years in that sector.

An example of cleaning false data can be seen in Figure 7 (left-hand panel). Detected false data are plotted in red. The figure clearly shows how those points deviate from the trajectory. In this example, the length of interval T is 24 hours to show the different possibilities available for step 1. The right-hand panel of Figure 7 shows the same data but smoothed using 15 cubic B-splines basis functions, i.e., using Step 4 in situation 1. Note that the small amount of noise

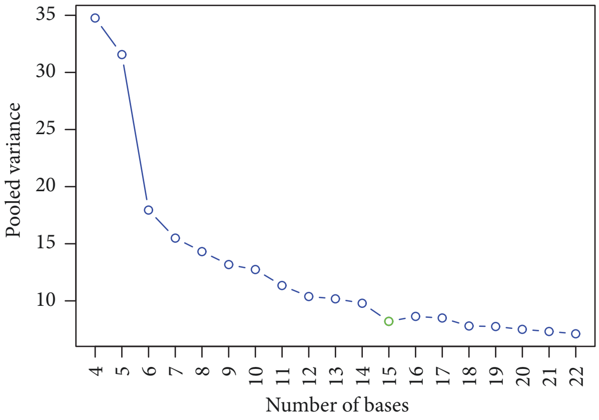


Figure 8: The relationship between the number of cubic B-splines basis functions and the variance estimate for the group of weekdays in spring for sector C. The chosen basis number is 15 (green).

has been erased and the curve has been reconstructed at the points where false data were found. Figure 8 reveals an elbow at 15, so this basis number was chosen.

In the left-hand panel of Figure 9, raw data with somewhat long periods of missing data are displayed. In the right-hand panel, those data have been smoothed and reconstructed using Step 4 in situation 2.

We now show a sample of the results from Phase I of Stage 2. Figure 10 shows two iterations for identifying outliers. For Phase II, Stage 2, the data in the right-hand panel of Figure 10 would be a sample of the in-control process and an outlier detection method should be used with that sample together with the new datum gathered for monitoring the system. A more lengthy presentation of the results is available in Millán-Roures [58], for readers who are interested in the particular results, i.e., the specific outliers detected for each sector, season, and weekdays or weekends.

Note that a leak would probably be represented as an amplitude outlier that persists until it is corrected. But, with the proposed methodology, we can also detect, for example, isolated outliers, i.e., different from the rest of data only in some part of the interval. It could correspond to some kind of fraud, for example. In short, we can detect anomalous behaviors; the following step is to check the reason: leaks, damage to measuring equipment, changes in the consumption pattern, etc.

3.2. Simulation Study. A simulation study is carried out to compare the different outlier detection methods in a controlled setting. As mentioned above, outliers that are completely outlying in shape but not so in magnitude are not easy to detect. Therefore, we begin our simulation study with this more difficult problem, considering the same models used in Section 2.4.3 with different percentages of shape outliers. Then, we concentrate on examining the results when only amplitude, isolated, and shift outliers are present, respectively. To appreciate the difficulty of each problem, remember that the left-hand panel of Figure 2 illustrates the problem with 2% shape outliers (2 of the 100 functions). The first part of the simulation study only changes this percentage, i.e., from zero to a higher percentage of outliers. In Figure 5, we see examples for the problems with amplitude, isolated, and shift outliers, the black shape outliers being discarded; i.e., shape outliers are not part of the data sets in the second part of the simulation study. In the third part of the simulation study, a new model for the data set is considered.

The contamination rates used with shape outliers are 0 (no outliers), 0.05, and 0.15 (although $c = 0.15$ is quite high and it could suggest that a subsample of the population is being generated, we have decided to use it because it was also considered in previous simulation studies such as Arribas-Gil and Romo [36]). A total of 100 simulations have been run. Default parameters have been used, except for the methods that require the coverage probability of the outlying region or trimming proportion to be specified. For those methods (LRT, TRIM, POND, and HDR), the true c values have been used. Obviously, this could give them some advantage, especially when $c = 0$. For our proposal, we have considered the method by Rousseeuw and Leroy [41] for detecting outliers in the ternary plot.

Table 1 displays the results for each method and the contamination rate. We consider that the best performance corresponds to the methods that identify many true positives (outliers as outliers) and few false positives (nonoutliers as outliers). The method that locates the highest number of outliers is FOADA, with a success rate of almost 100% in the case of $c = 0.05$, and also the highest rate in the case of $c = 0.15$. RMAH and OUG also provide very good results, although the success rate decreases slightly for OUG in the case of $c = 0.15$. The percentage of falsely identified outliers is low for these three methods, the lowest rate with these three methods corresponding to RMAH, especially when no outliers are present. TRIM and POND provide an excellent percentage of successfully identified outliers when $c = 0.05$, but the results are poor for the case of $c = 0.15$. Success rates for ISFE and HDR are medium-high. The results for FB are not very good for this problem, although the worst one is LRT, since it does not detect any outlier in any case.

Table 2 shows the results for each method and amplitude, isolated, and shift outliers with $c = 0.05$. The setting of each method is as above. For amplitude outliers, the best method is TRIM, with a success rate of 74%, followed by POND, RMAH, and FOADA. The success rate increases to 72.4% (38.9) for FOADA, if instead of using the method by Rousseeuw and Leroy [41] for detecting outliers in the ternary plot we examine it with the naked eye and consider as outliers those points with α higher than a threshold of 0.5 for the archetypoid with the fewest points nearest to them in the ternary plot. In any case, note that it is a hard problem; the amplitude outliers generated are not too far from the rest of the data (see the left-hand panel of Figure 5). As their authors explained, OUG has not been conceived to detect magnitude outliers, which explains its poor results with this kind of outliers. The best method for isolated outliers is ISFE, followed by FOADA, with 100% and 96.2% success rates, respectively. TRIM and POND also give very good results. As regards shift outliers, the best results are provided by FOADA (99.2% success rate), followed by RMAH, OUG, and TRIM.

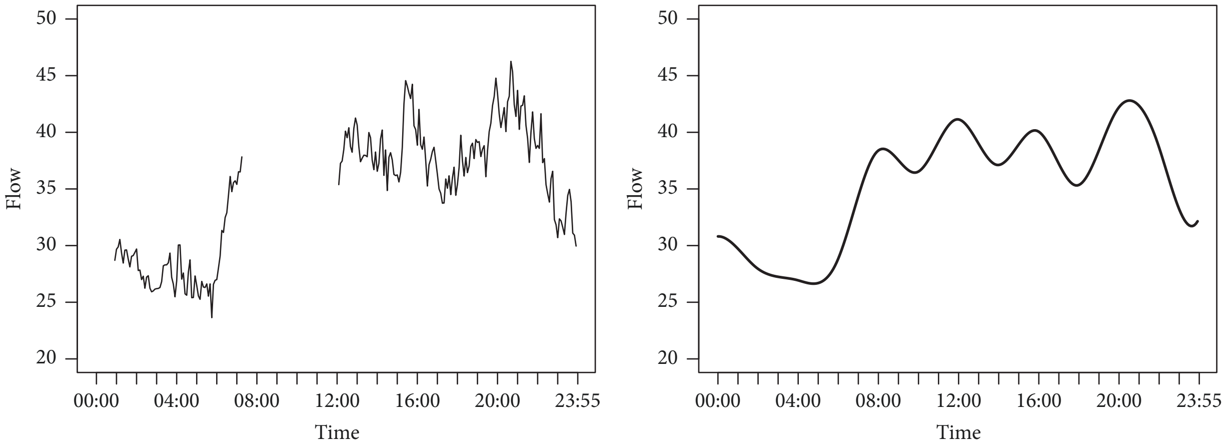


Figure 9: The left-hand panel shows raw data from 23/10/2014 with a period of missing data. It belongs to the group of weekdays in autumn for sector C. The right-hand panel shows the smoothed data. Time is expressed in hours and flow in cubic meter per hour (m^3/h).

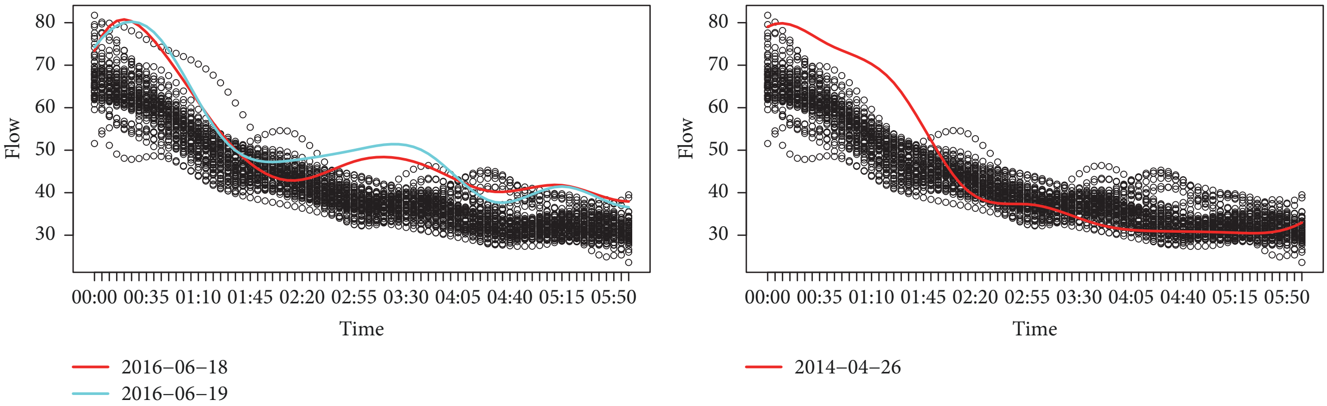


Figure 10: The left-hand panel shows the outlier detected by FB in the first iteration, while those detected in the second iteration are shown in the right-hand panel. The legends indicate the outliers. They belong to the group of weekends in spring for sector B. Time is expressed in hours and flow in cubic meter per hour (m^3/h). The procedure for outlier detection in Stage 2 is not restricted to FOADA; this is the reason why we have used, for instance, FB to illustrate this.

In the third part of the simulation study, a model similar to example 3 devised by Hyndman and Shang [23] and model 7 by Sun and Genton [34] is considered. We simulate 99 functions of the form $y_i(t) = a_i \sin(t) + b_i \cos(t)$, where $0 \leq t \leq 2\pi$, observed in steps of 0.1, and a_i and b_i follow normal distributions with mean 1 and standard deviation 0.1. One function, the atypical one, is simulated by the same functional form, but now the mean is 1.5. A data set is displayed in Figure 11. With FOADA, the method by Rousseeuw and Leroy [41] for detecting outliers in the ternary plot gives computational errors, so we consider a threshold of 0.5 and the same settings as above for the rest of the methods. Over 100 runs, the outlier is detected 100% of the times by FB, FOADA, POND, RMAH, and TRIM, 99% by HDR, 89% by FOM, and only 2% by OUG. LRT and ISFE do not detect it in any of the 100 runs. The percentage of falsely detected outliers is 0.020% for FB, 0.052% for FOADA, 0.053% for POND, 1.1% for RMAH, 1.4% for TRIM, 0.01% for HDR and FOM, and 0 for the rest of the methods. For this example, some methods achieve excellent results, but other methods are not able to detect the outlier.

4. Conclusions

We have proposed a method for monitoring and detecting anomalous flows in urban water networks based on FDA. Flow data are FD and the whole procedure is based on FDA techniques. Each phase and step has been illustrated with real data and the results are very promising. The proposed approach could be incorporated into integrated software to help decision-making for an overall water network management strategy based on computer-aided tools. We have also proposed a new method for identifying outlier functions based on archetypal analysis. This new procedure has been compared with other alternatives in a simulated setting, with very positive results. FOADA has provided consistently very good results for all the types of outliers. These results could be even improved if a finer procedure for detecting “holes” in the distribution of α coefficients for each archetypoid could be devised. Maybe this could involve estimating the distribution of compositional or closed data and finding the watersheds and catchments basins on the representation ([59], Ch. 9).

Table 1: Simulated data with shape outliers. Mean and standard deviation (in brackets) of the percentage of correctly (p_c) and falsely (p_f) identified outliers over 100 simulation runs.

	$c = 0$	$c = 0.05$		$c = 0.15$	
Method	p_f	p_c	p_f	p_c	p_f
LRT	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
TRIM	1.3 (0.7)	95.8 (15.1)	0.7 (0.8)	37.4 (35.9)	0.9 (1.2)
POND	1.8 (1.4)	96.6 (11.4)	1.2 (1.1)	2.5 (4.4)	0.2 (0.5)
ISFE	3.0 (2.0)	86.0 (29.7)	2.9 (1.8)	76.1 (33.9)	2.4 (1.8)
RMAH	1.6 (1.5)	97.4 (7.3)	0.9 (1.0)	93.3 (11.9)	0.2 (0.4)
HDR	0 (0)	65.0 (21.0)	1.8 (1.1)	62.9 (13.2)	6.6 (2.3)
FB	0.1 (0.3)	22.2 (21.4)	0.09 (0.3)	9.4 (10.8)	0.4 (0.2)
OUG	4.8 (2.9)	98.6 (5.1)	3.3 (2.2)	86.7 (12.6)	1.0 (1.2)
FOM	0.5 (0.8)	7.6 (15.5)	0.2 (0.5)	0.7 (6.7)	0.6 (2.6)
FOADA	4.8 (2.5)	99.4 (3.4)	3.4 (2.0)	96.2 (10.6)	1.4 (1.3)

Table 2: Simulated data with amplitude, isolated, and shift outliers with $c = 0.05$. Mean and standard deviation (in brackets) of the percentage of correctly (p_c) and falsely (p_f) identified outliers over 100 simulation runs.

	Amplitude		Isolated		Shift	
Method	p_c	p_f	p_c	p_f	p_c	p_f
LRT	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
TRIM	74.0 (22.7)	1.3 (1.0)	90.2 (18.7)	1.1 (1.1)	94.8 (9.7)	0.9 (1.0)
POND	63.2 (27.2)	0.7 (0.9)	84.6 (24.8)	1.0 (1.0)	90.2 (19.2)	1.1 (1.0)
ISFE	16.2 (20.4)	2.6 (2.0)	100 (0)	2.9 (2.2)	63.2 (40.5)	3.0 (1.8)
RMAH	59.0 (24.7)	1.0 (1.2)	62.6 (28.4)	0.9 (1.0)	97.6 (7.7)	1.0 (1.1)
HDR	51.4 (20.2)	2.6 (1.1)	51.8 (19.9)	2.5 (1.0)	65.0 (21.0)	1.8 (1.1)
FB	23.2 (21.4)	0.1 (0.3)	28.2 (36.5)	0.07 (0.3)	18.8 (22.2)	0.1 (0.3)
OUG	0 (0)	5.0 (3.1)	44.4 (23.4)	3.6 (2.5)	94.8 (9.3)	3.4 (2.1)
FOM	24.4 (24.5)	0.2 (0.5)	14.0 (25.8)	0.3 (0.7)	10.0 (17.4)	0.2 (0.4)
FOADA	57.4 (27.7)	2.4 (1.6)	96.2 (17.9)	2.6 (1.8)	99.2 (6.3)	3.0 (1.9)

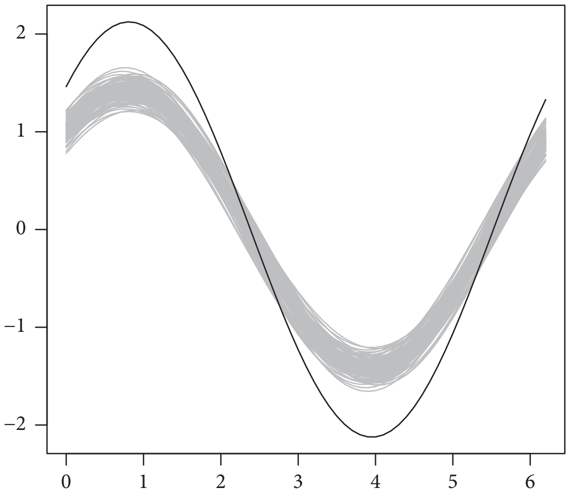


Figure 11: Simulated data with sinusoidal functions. The outlier is shown in black.

In future work, we could consider other functional variables, such as pressure, in addition to flow. Multivariate functional techniques should be used when more than one functional variable is available. As regards the functional outlier detection part, other tools could be developed. As shape outliers are the type of outliers that are most difficult to detect, techniques based on the shape of the functions could be taken into account. These ideas worked very well in classification problems, not only for univariate functions (Epifanio [60]), but also for multivariate functions, with one (Epifanio and Ventura-Campos [61]) or more arguments (Epifanio and Ventura-Campos [62]). These ideas could therefore be applied to the functional outlier detection problem.

Data Availability

The code and data for the simulation study and a script for the real case study are available at http://www3.uji.es/~epifanio/RESEARCH/fouada.rar.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to acknowledge the company FACSA for providing them with the data set. This work was carried out while Laura Millán-Roures was enjoying a grant supported by the Cátedra FACSA at Universitat Jaume I. This work has been partially supported by the Spanish Ministerio de Economía y Competitividad (AEI/FEDER, UE) Grant DPI2017-87333-R and Universitat Jaume I, UJI-B2017-13.

References

[1] H. Rodriguez, V. Puig, J. J. Flores, and R. Lopez, "Flow meter data validation and reconstruction using neural networks: Application to the Barcelona water network," in *Proceedings of the 2016 European Control Conference, ECC 2016*, pp. 1746–1751, dnk, July 2016.

[2] C. V. Palau, F. J. Arregui, and M. Carlos, "Burst detection in water networks using principal component analysis," *Journal of Water Resources Planning and Management*, vol. 138, no. 1, pp. 47–54, 2012.

[3] J. Quevedo, V. Puig, G. Cembrano et al., "Validation and reconstruction of flow meter data in the Barcelona water distribution network," *Control Engineering Practice*, vol. 18, no. 6, pp. 640–651, 2010.

[4] D. Burnell, "Auto-validation of district meter data," in *Proceedings of the CCWI '03 Conference In Advances in Water Supply Management*, pp. 13–22, 2003.

[5] S. Prescott and B. Ulanicki, "Auto-validation of district meter data," in *Water Software Systems-Theory and Applications*, vol. 2, pp. 17–28, 2001.

[6] E. Piatyszek, P. Voignier, and D. Graillot, "Fault detection on a sewer network by a combination of a Kalman filter and a binary sequential probability ratio test," *Journal of Hydrology*, vol. 230, no. 3-4, pp. 258–268, 2000.

[7] N. Valentin and T. Denœux, "A neural network-based software sensor for coagulation control in a water treatment plant," *Intelligent Data Analysis*, vol. 5, no. 1, pp. 23–39, 2001.

[8] S. R. Mounce, J. B. Boxall, and J. Machell, "Development and verification of an online artificial intelligence system for detection of bursts and other abnormal flows," *Journal of Water Resources Planning and Management*, vol. 136, no. 3, pp. 309–318, 2010.

[9] Z. Poulakis, D. Valougeorgis, and C. Papadimitriou, "Leakage detection in water pipe networks using a Bayesian probabilistic framework," *Probabilistic Engineering Mechanics*, vol. 18, no. 4, pp. 315–327, 2003.

[10] D. Misiunas, J. Vitkovský, G. Olsson, M. Lambert, and A. Simpson, "Failure monitoring in water distribution networks," *Water Science and Technology*, vol. 53, no. 4-5, pp. 503–511, 2006.

[11] B. M. Colosimo and M. Pacella, "A comparison study of control charts for statistical monitoring of functional data," *International Journal of Production Research*, vol. 48, no. 6, pp. 1575–1601, 2010.

[12] J. O. Ramsay and B. W. Silverman, *Functional Data Analysis*, Springer, New York, NY, USA, 2nd edition, 2005.

[13] F. Ferraty and P. Vieu, *Nonparametric Functional Data Analysis: Theory and Practice*, Springer, 2006.

[14] J. O. Ramsay and B. W. Silverman, *Applied Functional Data Analysis: Methods and Case Studies*, Springer Series in Statistics, Springer, Berlin, Germany, 2002.

[15] J. O. Ramsay, G. Hooker, and S. Graves, *Functional Data Analysis with R and MATLAB*, Springer, 2009.

[16] B. Henderson, "Exploring between site differences in water quality trends: a functional data analysis approach," *Environmetrics*, vol. 17, no. 1, pp. 65–80, 2006.

[17] C. Díaz Muñiz, P. J. García Nieto, J. R. Alonso Fernández, J. Martínez Torres, and J. Taboada, "Detection of outliers in water quality monitoring samples using functional data analysis in San Esteban estuary (Northern Spain)," *Science of the Total Environment*, vol. 439, pp. 54–61, 2012.

[18] F. Yan, L. Liu, Y. Li, Y. Zhang, M. Chen, and X. Xing, "A dynamic water quality index model based on functional data analysis," *Ecological Indicators*, vol. 57, pp. 249–258, 2015.

[19] C. Ternynck, M. A. B. Alaya, F. Chebana, S. Dabo-Niang, and T. B. M. J. Ouarda, "Streamflow hydrograph classification using functional data analysis," *Journal of Hydrometeorology*, vol. 17, no. 1, pp. 327–344, 2016.

[20] N. Cheifetz, Z. Noumir, A. Samé, A.-C. Sandraz, C. Féliers, and V. Heim, "Modeling and clustering water demand patterns from real-world smart meter data," *Drinking Water Engineering and Science Discussions*, pp. 1–12, 2017.

[21] I. Epifanio, "Functional archetype and archetypoid analysis," *Computational Statistics & Data Analysis*, vol. 104, pp. 24–34, 2016.

[22] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2017.

[23] R. J. Hyndman and H. L. Shang, "Rainbow plots, bagplots, and boxplots for functional data," *Journal of Computational and Graphical Statistics*, vol. 19, no. 1, pp. 29–45, 2010.

[24] H. L. Shang and R. J. Hyndman, *Rainbow: Rainbow Plots, Bagplots and Boxplots for Functional Data*, R package version 3.4, 2016.

[25] L. N. Rathnayake and P. K. Choudhary, "Tolerance bands for functional data," *Biometrics: Journal of the International Biometric Society*, vol. 72, no. 2, pp. 503–512, 2016.

[26] J. O. Ramsay, H. Wickham, S. Graves, and G. Hooker, *fda: Functional Data Analysis*, R package version 2.4.4.

[27] G. James, "The Oxford handbook of functional data analysis," in *Chapter Sparseness And Functional Data Analysis*, pp. 298–326, Oxford University Press, 2010.

[28] F. Yao, H.-G. Müller, and J.-L. Wang, "Functional data analysis for sparse longitudinal data," *Journal of the American Statistical Association*, vol. 100, no. 470, pp. 577–590, 2005.

[29] X. Dai, P. Hadjipantelis, H. Ji, H.-G. Mueller, and J.-L. Wang, *fdapace: Functional Data Analysis and Empirical Dynamics*, R package version 0.3.0, 2017.

[30] D. Montgomery, *Statistical Quality Control*, Wiley, 6th edition, 2009.

[31] M. Febrero, P. Galeano, and W. González-Manteiga, "Outlier detection in functional data by depth measures, with application to identify abnormal NOx levels," *Environmetrics*, vol. 19, no. 4, pp. 331–345, 2008.

[32] M. Hubert, P. J. Rousseeuw, and P. Segaert, "Multivariate functional outlier detection," *Statistical Methods & Applications*, vol. 24, no. 2, pp. 177–202, 2015.

[33] M. Febrero, P. Galeano, and W. González-Manteiga, "A functional analysis of NOx levels: location and scale estimation and outlier detection," *Computational Statistics*, vol. 22, no. 3, pp. 411–427, 2007.

[34] Y. Sun and M. G. Genton, "Functional boxplots," *Journal of Computational and Graphical Statistics*, vol. 20, no. 2, pp. 316–334, 2011.

[35] D. Gervini, "Outlier detection and trimmed estimation for general functional data," *Statistica Sinica*, vol. 22, no. 4, pp. 1639–1660, 2012.

[36] A. Arribas-Gil and J. Romo, "Shape outlier detection and visualization for functional data: The outliergram," *Biostatistics*, vol. 15, no. 4, pp. 603–619, 2014.

[37] R. J. Hyndman and M. Shahid Ullah, "Robust forecasting of mortality and fertility rates: a functional data approach," *Computational Statistics & Data Analysis*, vol. 51, no. 10, pp. 4942–4956, 2007.

[38] P. Sawant, N. Billor, and H. Shin, "Functional outlier detection with robust functional principal component analysis," *Computational Statistics*, vol. 27, no. 1, pp. 83–102, 2012.

[39] R. Fraiman and M. Svarc, "Resistant estimates for high dimensional and functional data based on random projections," *Computational Statistics & Data Analysis*, vol. 58, pp. 326–338, 2013.

[40] M. Febrero-Bande and M. O. de la Fuente, "Statistical computing in functional data analysis: The R package fda.usc," *Journal of Statistical Software* , vol. 51, no. 4, 2012.

[41] P. J. Rousseeuw and A. M. Leroy, *Robust Regression & Outlier Detection*, Wiley, 1987.

[42] N. Tarabelloni, A. Arribas-Gil, F. Ieva, A. M. Paganoni, and J. Romo, *Roahd: Robust Analysis of High Dimensional Data*, R package version 1.3, 2017.

[43] P. Segaert, M. Hubert, P. Rousseeuw, and J. Raymaekers, *mrfDepth: Depth Measures in Multivariate, Regression and Functional Settings*, R package version 1.0.4, 2017.

[44] P. J. Rousseeuw, J. Raymaekers, and M. Hubert, "A measure of directional outlyingness with applications to image data and video," *Journal of Computational and Graphical Statistics*, 2017.

[45] A. Cutler and L. Breiman, "Archetypal analysis," *Technometrics*, vol. 36, no. 4, pp. 338–347, 1994.

[46] C. Thurau, K. Kersting, M. Wahabzada, and C. Bauckhage, "Descriptive matrix factorization for sustainability: adopting the principle of opposites," *Data Mining and Knowledge Discovery*, vol. 24, no. 2, pp. 325–354, 2012.

[47] G. Vinué, I. Epifanio, and S. Alemany, "Archetypoids: a new approach to define representative archetypal data," *Computational Statistics & Data Analysis*, vol. 87, pp. 102–115, 2015.

[48] I. Epifanio, G. Vinué, and S. Alemany, "Archetypal analysis: Contributions for estimating boundary cases in multivariate accommodation problem," *Computers & Industrial Engineering*, vol. 64, no. 3, pp. 757–765, 2013.

[49] I. Epifanio, M. V. Ibáñez, and A. Simó, "Archetypal shapes based on landmarks and extension to handle missing data," *Advances in Data Analysis and Classification (ADAC)*, https://doi.org/10.1007/s11634-017-0297-7.

[50] G. Vinué and I. Epifanio, "Archetypoid analysis for sports analytics," *Data Mining and Knowledge Discovery*, vol. 31, no. 6, pp. 1643–1677, 2017.

[51] M. J. Eugster and F. Leisch, "From Spider-Man to Hero - Archetypal Analysis in R," *Journal of Statistical Software*, vol. 30, no. 8, pp. 1–23, 2009.

[52] G. Vinué, "Anthropometry: An R package for analysis of anthropometric data," *Journal of Statistical Software*, vol. 77, no. 6, pp. 1–39, 2017.

[53] M. J. A. Eugster and F. Leisch, "Weighted and robust archetypal analysis," *Computational Statistics & Data Analysis*, vol. 55, no. 3, pp. 1215–1225, 2011.

[54] S. López-Pintado and J. Romo, "On the concept of depth for functional data," *Journal of the American Statistical Association*, vol. 104, no. 486, pp. 718–734, 2009.

[55] M. R. D'Esposito and G. Ragozini, "A new R-ordering procedure to rank multivariate performances," *Quaderni di Statistica*, vol. 10, pp. 22–40, 2008.

[56] C. A. Lowry and D. C. Montgomery, "A review of multivariate control charts," *Institute of Industrial Engineers (IIE). IIE Transactions*, vol. 27, no. 6, pp. 800–810, 1995.

[57] W. A. Jensen, L. A. Jones-Farmer, C. W. Champ, and W. H. Woodall, "Effects of parameter estimation on control chart properties: a literature review," *Journal of Quality Technology*, vol. 38, no. 4, pp. 349–364, 2006.

[58] L. Millán-Roures, *Outliers De Datos Funcionales Para La Detección De Caudales Anómalos En El Sector Hidráulico*, Universitat Jaume I, 2017.

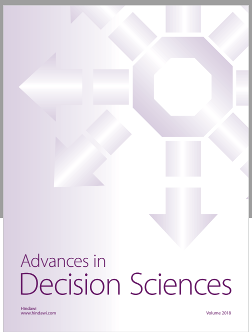
[59] P. Soille, *Morphological Image Analysis: Principles and Applications*, Springer-Verlag, Berlin, Germany, 2nd edition, 2003.

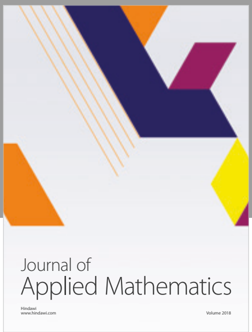
[60] I. Epifanio, "Shape descriptors for classification of functional data," *Technometrics. A Journal of Statistics for the Physical, Chemical and Engineering Sciences*, vol. 50, no. 3, pp. 284–294, 2008.

[61] I. Epifanio and N. Ventura-Campos, "Functional data analysis in shape analysis," *Computational Statistics & Data Analysis*, vol. 55, no. 9, pp. 2758–2773, 2011.

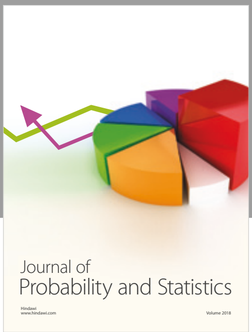
[62] I. Epifanio and N. Ventura-Campos, "Hippocampal shape analysis in Alzheimer's disease using functional data analysis," *Statistics in Medicine*, vol. 33, no. 5, pp. 867–880, 2014.

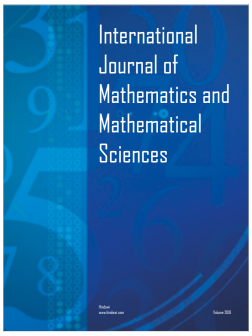
Advances in
Operations Research


Advances in
Decision Sciences

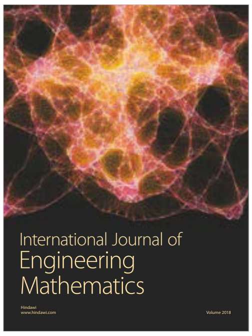

Journal of
Applied Mathematics

The Scientific
World Journal

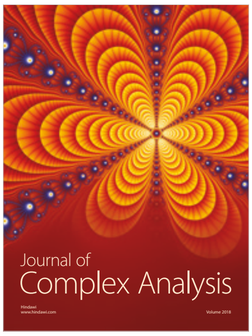

Journal of
Probability and Statistics

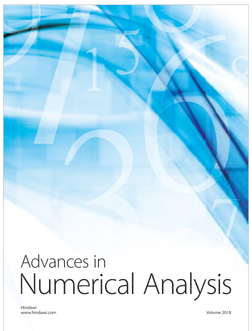

International
Journal of
Mathematics and
Mathematical
Sciences

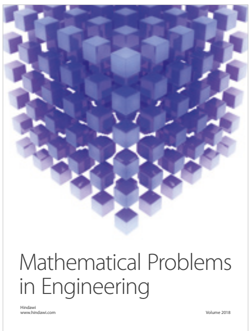
Journal of
Optimization

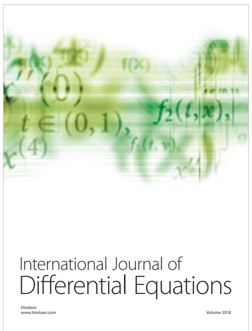

International Journal of
Engineering
Mathematics

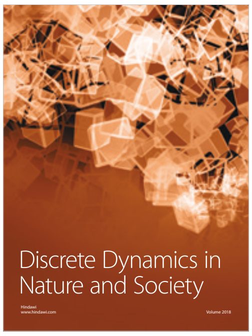
International Journal of
Analysis

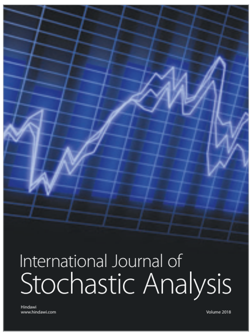

Journal of
Complex Analysis


Advances in
Numerical Analysis


Mathematical Problems
in Engineering

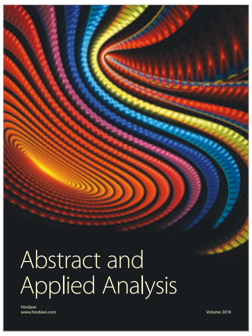

International Journal of
Differential Equations

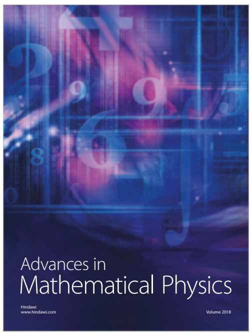

Discrete Dynamics in
Nature and Society


International Journal of
Stochastic Analysis

Journal of
Mathematics

Journal of
Function Spaces


Abstract and
Applied Analysis


Advances in
Mathematical Physics