



HAL
open science

Shape-based Outlier Detection in Multivariate Functional Data

Clément Lejeune, Josiane Mothe, Adil Soubki, Olivier Teste

► **To cite this version:**

Clément Lejeune, Josiane Mothe, Adil Soubki, Olivier Teste. Shape-based Outlier Detection in Multivariate Functional Data. Knowledge-Based Systems, 2020, 198, pp.1-18. 10.1016/j.knosys.2020.105960 . hal-02960358

HAL Id: hal-02960358

<https://hal.science/hal-02960358v1>

Submitted on 22 Aug 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

1 Highlights

2 **Shape-based outlier detection in multivariate functional data**

3 Clément Lejeune, Josiane Mothe, Adil Soubki, Olivier Teste

- 4 • A new method is introduced for detecting outliers in multivariate functional data based on the curve shape that such
5 data depict. Few work address the problem of outlier detection in multivariate functional data, and our proposal relies
6 on some curve shape features combined with state-of-the art outlier detection.
- 7 • We represent the data through some functional approximations. We propose several interpretable transformations to
8 map the resulting approximated functional data to a curve shape representation.
- 9 • We prove through experimental studies on real and synthetic data that our approach can outperform the baselines. Also
10 we show that our method performs well contrary to the baselines, whenever the proportion of outliers is high or low.
11 We discuss some issues the baselines cannot circumvent.
- 12 • We provide some recommendations regarding the kinds of curve shape representation to use with respect to the type of
13 outlier that the data set entails.

Shape-based outlier detection in multivariate functional data

Clément Lejeune^{a,b,*}, Josiane Mothe^{a,c}, Adil Soubki^b and Olivier Teste^a

^aIRIT, UMR-5505 CNRS, 118 Route de Narbonne, 31062, Toulouse, France

^bAirbus Operations, 316 Route de Bayonne, 31300, Toulouse, France

^cINSPE, Université de Toulouse, UT2J, Toulouse, France

ARTICLE INFO

Keywords:

Outlier Detection

Multivariate Functional Data

Multivariate Time Series

Multidimensional Curve Shape

ABSTRACT

Multivariate functional data refer to a population of multivariate functions generated by a system involving dynamic parameters depending on continuous variables (e.g., multivariate time series). Outlier detection in such a context is a challenging problem because both the individual behavior of the parameters and the dynamic correlation between them are important. To address this problem, recent work has focused on multivariate functional depth to identify the outliers in a given dataset. However, most previous approaches fail when the outlyingness manifests itself in curve shape rather than curve magnitude. In this paper, we propose identifying outliers in multivariate functional data by a method whereby different outlying features are captured based on mapping functions from differential geometry. In this regard, we extract shape features reflecting the outlyingness of a curve with a high degree of interpretability. We conduct an experimental study on real and synthetic data sets and compare the proposed method with functional-depth-based methods. The results demonstrate that the proposed method, combined with state-of-the-art outlier detection algorithms, can outperform the functional-depth-based methods. Moreover, in contrast with the baseline methods, it is efficient regardless of the proportion of outliers.

1. Introduction

High-dimensional data are defined as individual vectors representing a large number of measurements. They appear in various fields, such as biology, engineering, or medicine, where different sources of measurements are recorded. As a straightforward example of such data, we can consider a longitudinal study for analyzing the height of a human population, such as the Berkley growth study [44], in which a physiological parameter or variable (also termed “source”) is measured for all subjects at various time instants. Depending on the population and the number of time instants, this collection may result in high-dimensional data. Such data can be seen as realizations of a univariate function depending on time. Although a continuous function depending on a single continuous variable (e.g., time, wavelength, or frequency) underlies the data, it is finely discretized, resulting in high-dimensional vectors. Such data are referred to as functional data.

Functional data analysis (FDA) is a branch of modern statistics, the principle of which is the representation of high-dimensional measurement vectors through functions (see [32, 16] for a practical and theoretical introduction to FDA). Regarding data as functions enables recovering the true nature of the process underlying the function that generated the data. It also provides a smooth representation of the initial curves, which can be affected by measurement noise. Moreover, the FDA framework enables the handling of curves that are irregularly sampled or sampled on grids of different sizes,


where a grid refers to the discretization of a closed interval in which the continuous variable lies. This is achieved by evaluating the resulting functions on a common and arbitrary grid.

Specifically, when a single variable is recorded at each observation point (as in the previous example), that is, the underlying function $x(t) \in \mathbb{R}$, where t lies in a closed interval $\mathcal{T} \subset \mathbb{R}$, the resulting data are called univariate functional data. More generally, when p correlated variables are simultaneously recorded at each observation point, that is, $X(t) = (x_1(t), \dots, x_k(t), \dots, x_p(t)) \in \mathbb{R}^p$, these data are called multivariate functional data. In the example, if weight was measured in addition to height, these data would result as realizations of a multivariate function (in this case bivariate). In the remainder of this paper, we use lower-case letters ($x(t) \in \mathbb{R}$) and capital letters ($X(t) \in \mathbb{R}^p$) to distinguish univariate from multivariate functional data.

A typical task in FDA is outlier detection [22], which has several applications, for instance, in biology (to determine abnormal gene expression levels in time-course micro-array data [2]), in chemometrics (to determine the nature of an active substance produced by a chemical process based on near-infrared spectra data [22]), or in air pollution studies (to detect highly contaminated locations in urban areas [43]). In these fields, the data are typically functional and exhibit outlying behavior. Moreover, several parameters should be simultaneously recorded to accurately understand the studied process. Hence, outlier-detection methods should be specifically designed for multivariate functional data. Since the variables are cautiously selected by a domain expert, the outlying behavior can be detected through the potential correlation between them.

The correlation between the p variables is important in multivariate functional data because it can reveal the outlying

*Corresponding author

 clement.lejeune@irit.fr (C. Lejeune);

clement.lejeune@airbus.com (C. Lejeune); josiane.mothe@irit.fr (J. Mothe); adil.soubki@airbus.com (A. Soubki); olivier.teste@irit.fr (O. Teste)

ORCID(s):

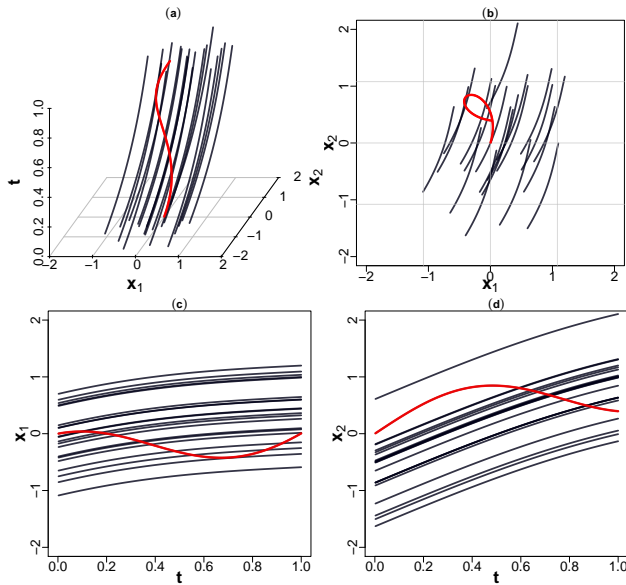


Figure 1: Example of a bivariate ($p = 2$) functional dataset (see the color version for greater clarity). (a) A dataset of 21 bivariate curves, with variables $x_{i1}(t), x_{i2}(t), i = 1 \dots 21$, is plotted along the variable and the $t \in [0, 1]$ axes. There are 20 inliers (black) and one shape outlier (red). (b) The dataset is projected along the t axis; the red curve clearly shows an outlying relationship between its variables, resulting in a different shape. This is the “view” adopted in this study. In (c) and (d), the variables x_{i1} and x_{i2} are plotted as two univariate functions with respect to t . Determining the degree of difference of the red curve without computing derived functions (e.g., derivative(s)) is not simple. Moreover, if the dataset is very large, the red curve is totally mixed with the black curves, thus rendering visual detection difficult.

tering of outliers may render these algorithms inefficient for outlier detection, owing to the well-known *class imbalance problem* [25].

Previous work on outlier detection in functional data primarily focused on the univariate case [17, 7, 28], whereas the multivariate case is more recent [4, 24, 29, 22, 26, 8]. Multivariate functional outliers can be characterized by deviations in the correlation between the variables $x_1(t), \dots, x_k(t), \dots, x_p(t)$ and, potentially, in their correlation with t . There can be scattering among functional outliers depending on how outlyingness is expressed. According to the functional-outlier taxonomy by Hubert *et al.* [22], there are two general classes: isolated and persistent outliers. An isolated outlier exhibits extreme behavior in a small part of the domain \mathcal{T} , resulting in a narrow peak in at least one of the variables. By contrast, a persistent outlier is defined as a sample in which outlyingness manifests itself in a large part of the domain. Among persistent outliers, three classes were distinguished by Hubert *et al.* as follows [22]: (i) A *shift outlier* exhibits a pattern comparable to that of a regular curve up to a random horizontal translation. (ii) A *magnitude outlier* differs in terms of range. (iii) A *shape outlier* exhibits outlyingness in local features without deviating from the regular curves at any point of the domain.

The detection of shape outliers is quite recent and is attracting increasing attention in FDA [29, 2, 26, 8]. Persistent shape outliers are difficult to detect in a curve population because the shapes are often non-linearly discriminant (Fig. 1(b)) and exhibit larger variability than isolated outliers. Considering curve discrimination in terms of shape, one can augment the curve variables by using differential analysis. This refers to adding derivatives or integrals (computed with respect to t) for each initial variable. Hence, curve shape provides information regarding “hidden outlying features” of the curve variables and the outlying relationship between them. However, as mentioned previously, the joint analysis of the p variables becomes complex as p increases (see Fig. 1). In the present study, we address this problem by using differential geometry. Specifically, we use aggregation functions (termed mapping functions) of the variables. Thereby, we implicitly consider the correlation of the variables through geometrical characterizations of curve shape. In contrast with current functional-outlier detection methods, which consider curve shape differently and only base the final detection on the resulting *depth values* (Section 2), we use both functional curve-shape features and state-of-the-art outlier-detection algorithms. Thus, the originality of the proposed approach lies in the shape characterization of the initial curves through the proposed mapping functions, combined with state-of-the-art outlier-detection algorithms.

Throughout this paper, we use the term *mapping function* to refer to analytic aggregation functions that enable capturing curve-shape features, such as curvature, length (i.e., perimeter of a shape), or tangential velocity, and consider all the variables, as a curve is viewed as a path. More precisely, a mapping function aggregates the variables through different interpretable combinations of the derivatives of the variables.

behavior of the underlying process, as discussed in [22] and shown in Fig. 1. Thus, independently analyzing each variable implies that the potential correlation between the variables is not considered, as shown in Fig. 1 (with a bivariate functional dataset), where, in (a) and (b), the variables x_1 and x_2 appear correlated, whereas in (c) and (d), they individually exhibit correlation with respect to the continuum t .

According to the definition by Aggarwal *et al.* in [1], an outlier is defined as a data point that is highly different from the others, based on some *measure*. Such a point often contains useful information regarding the abnormal behavior of the system described by the data. Outlier detection is aimed at determining an appropriate measure whereby outliers may be differentiated from inliers with a high degree of interpretability. Based on this definition, outliers, compared with inliers, represent a small part of the dataset and are scattered. Moreover, if the data dimension is high, the data are more scattered in the space (i.e., curse of dimensionality), and therefore, the probability that the outliers are scattered is higher. Hence, outlier-detection tasks are as susceptible to the curse of dimensionality as other discrimination tasks that assume well-balanced classes. However, regarding some typical algorithms for classification (e.g., logistic regression) and clustering (e.g., K-means and mean-shift), the rarity and scat-

Mapping functions have been used in shape analysis [40] that is, for curves lying in a two- or three-dimensional space (e.g., extracted from images), but not in the detection of multivariate functional outliers.

In this paper, to capture the potential outlyingness of curves through their shape, we propose mapping functions among those used in differential geometric-method in shape analysis [40]. These functions map multivariate to univariate curves; however, for efficient computation, they require curves to be smooth. Although this is the case for multivariate functional data, raw data are often noisy when sampled and we use the functional-data representation to recover smooth version of the curves. Then, using the proposed mapping functions, we map the functional representation (in the form of a curve) so that some of its shape features capture curve outlyingness. Finally, based on this new representation we use outlier-detection algorithms to assess the outlyingness of each sample and determine a threshold for flagging outliers.

The contributions of this study are summarized as follows:

- (i) We propose an end-to-end method for detecting outliers through their curve shape, which is characterized by geometrical transformations. The method is based on the functional representation of the data.
- (ii) We propose different mapping functions to capture different types of outlyingness based on curve shape.
- (iii) We demonstrate that the proposed method is superior to previous outlier-detection algorithms and, in contrast to baseline methods, performs well regardless of the proportion of outliers.

The rest of the paper is organized as follows. In Section 2 we review related work on outlier detection in both univariate and multivariate functional data. In Section 3, we discuss curve representations in the functional-data framework. In Section 4, we present the mapping functions that can capture shape outlyingness from the obtained functional representation. The experimental results are presented and discussed in Section 5. Finally, Section 6 concludes this paper.

2. Related work

2.1. Depth-based univariate functional-outlier detection

The detection of outliers in functional data is a recent topic and has primarily been addressed by extending *statistical depth*¹ to *functional depth*. Statistical depth measures the centrality of a sample relative to a dataset by providing an outward-center ordering of the samples through a score lying in $[0, 1]$. A value close to zero implies that the sample is more likely to be an outlier [45]. Statistical depth has several

theoretical properties (see [49] for details): (i) It attains its maximum value for the most centered (i.e., most representative) sample. (ii) It decreases monotonically and vanishes as the sample moves away from the center (up to infinity). (iii) It does not depend on the dataset scale. Therefore, given an outlyingness threshold, samples with a depth value close to 0 can be flagged as outliers. This type of measure has been extended to functional data and used for classification [6], ranking [17, 7], as well as outlier detection [14].

However, most of the existing functional depths are applicable to univariate functional data only. For instance, given a functional sample, the integrated depth [17], modified band depth, and modified epigraph index [28] evaluate depth pointwise, that is, at each observation point $t \in \mathcal{T}$, and then these depth values are averaged by integration over \mathcal{T} to provide a global outward-center score. The integrated depth measures the proportion of a curve that is closest to the median curve of the dataset, where the median curve is computed pointwise. The modified band depth measures the average proportion of the curve that takes values within the range of all pairwise sample combinations, where “proportion of a curve” refers to the size of the interval \mathcal{T} where the curve outlies the dataset. The modified epigraph index has a similar principle: It measures the proportion of the curve that takes values smaller than the other values of the dataset. Thus, the functional depth intuitively measures the centrality of the curve, regarding its global shape with respect to the dataset, see [28] for details. The bivariate random projection depth by Cuevas and Febrero in [6] considers specific shape information by projecting the curve and its first derivative onto random directions (e.g., directions generated according to a unit-variance Gaussian process), resulting in several bivariate vectors; a bivariate statistical depth function is then applied to these vectors and averaged over the random projections. Based on any of these functional depths, an outlyingness threshold is necessary for outlier detection. If the depth-value distribution is known, which is rare in practice, one can select the threshold as a small probability quantile (e.g., a sample with depth value lower than the 5%-quantile of this distribution is likely an outlier). Febrero *et al.* proposed in [14] estimating this threshold as the first percentile of the empirical distribution of the depth values through a bootstrap procedure.

Unfortunately, apart from the statistical point of view, these approaches do not facilitate the understanding of the nature of outlyingness. Accordingly, techniques have been developed for visually detecting univariate functional outliers. Arribas-Gil and Romo defined the outliergram in [2] to represent each sample as a bivariate vector with the modified band-depth and epigraph values. They demonstrated that these depths are quadratically related. Hence, in a two-dimensional plot, inlier samples lie on a parabola, whereas outliers are likely to be far from it. Sun and Genton [42] proposed the functional boxplot to summarize the empirical distribution of the functional data as classical boxplots computed pointwise. It was designed to visualize a univariate functional dataset, in the same spirit as that of the classical boxplot. In their method, the central region of the pointwise

¹statistical depth was not specifically proposed for functional but for multivariate data. However, we distinguish between *univariate functional depth* and *multivariate functional depth*, which were proposed specifically for functional data.

boxplots is defined as the region in \mathbb{R} where the 50% highest depth-score samples $\{x_i(t)\}_{i \leq \lfloor n/2 \rfloor}$ (i.e., the most central) lie according to the band-depth ranks [28]. The fences of the boxplots are defined by inflating 1.5 times the height of the central region. Thus, the continuum of the pointwise boxplots provides a functional boxplot. The outliers are then identified as samples falling outside the fences. In this functional boxplot, inliers and outliers rely heavily on curve magnitude. Thus, curve shape largely fails to be considered a potential outlyingness feature. In [23], Hyndman and Shang applied robust principal component analysis by considering the samples to be high-dimensional vectors and represented each sample as a bivariate vector containing the first and second principal scores. Subsequently, outliers were identified as samples outside certain high-density regions that were termed using the empirical distribution of these bivariate vectors.

2.2. Depth-based multivariate functional outlier detection

Depth-based outlier detection methods for multivariate functional data are more recent. In [4], Claeskens *et al.* generalized any given univariate functional depth to the case of multivariate functional data. This corresponds to a weighted sum of a given univariate functional depth applied to each variable $(\tilde{x}_1(t), \dots, \tilde{x}_k(t), \dots, \tilde{x}_p(t))$ pointwise and then integrated over \mathcal{T} . The selection of the weight function was also discussed. As a special case, in [24], Ieva and Paganoni proposed the multivariate band depth by using the modified band depth as the given univariate functional depth; the weights associated to the variables are constant with respect to t .

In [22], Hubert *et al.* noted that the generalization by Claeskens *et al.* [4] does not always allow the detection of all types of functional outliers, namely, shape outliers. Indeed, low-depth samples stand near the boundary of the dataset but may not be outliers. Conversely, high-depth samples may present outlyingness in their curve shape because, pointwise the curve does not exhibit any significant deviance in each variable, as this generalization is the sum of the individual univariate functional depths. To address this, the entire shape of the curve should be considered.

A few studies incorporate curve shape into a multivariate functional depth measure. Recently, Kuhnt and Rehage [26] proposed the functional tangential-angle (*FUNTA*) pseudo depth, which considers curve shape based on the intersection angles of the centered variables (i.e., the variables are scaled so that their integral over \mathcal{T} values is 0). More precisely, for each variable, *FUNTA* computes the intersection angles of a given sample x_{ik} with all the other samples $x_{jk} \forall j \neq i$, and then averages these angles over the number of intersection angles of x_{ik} and over the variables $k = 1 \dots p$. Thus, *FUNTA* separately considers the shape for each variable with respect to t , but not the shape between the p variables.

More recently, Dai and Genton [8] proposed the directional outlyingness measure (*Dir.out*), which considers curve shape through the weighted pointwise direction in \mathbb{R}^p of the vector $X(t)$ toward the median of the distribution of $X(t)$.

The purpose of the weights is the up-weighting of the directions in which the outlyingness of $X(t)$ is likely to appear. In contrast with the aforementioned multivariate functional depths, which provide a score in $[0, 1]$, the *Dir.out* depth returns a vector in $\mathbb{R}^p \times \mathbb{R}^+$ corresponding to the concatenation of the mean directional outlyingness (in \mathbb{R}^p) and the total variance of the directional outlyingness (in \mathbb{R}^+). A final outlyingness score is computed as the robust Mahalanobis distance between this vector and a mean vector of the same type computed on a subset of independent samples. Then, the upper tail of this distance distribution is approximated by an F -distribution, and the outlyingness threshold is defined as a high-probability quantile of this F -distribution. Hence, unlike in other multivariate functional depths, the outlyingness threshold provided by the *Dir.out* approach is not data-driven, as it is based on the (approximately) true distribution of the outlyingness scores. However, in this approach, the parameters should be tuned by simulation and are difficult to interpret beyond the statistical framework.

Multivariate functional depths are related to curve shape through the individual behavior of the curve variables. Here, we adopt a different approach, as we view a curve as a path in \mathbb{R}^p and process it as a geometrical shape.

As all the aforementioned multivariate functional depths yield an outlyingness score with unknown distribution (except for *Dir.out*), an outlyingness threshold can be computed from the resulting empirical distribution of the depth values through a bootstrap procedure as in the univariate case [14]. It can also be computed from a training dataset based on the receiver operating characteristic (ROC) curve.

In the experimental study (Section 5), we use the *FUNTA* and *Dir.out* functional depths as baselines because they have been demonstrated to be promising for outlier detection in multivariate functional data by regarding outlyingness as a curve-shape feature.

2.3. Geometry-based functional-outlier detection

Representing functional data in a geometric framework is a recent idea, and few studies have considered such representations for outlier detection. Recently, in [48], Xie *et al.* proposed detecting outliers in univariate functional data by decomposing each univariate functional sample into three features: translation, phase, and amplitude. The authors defined the translation of a functional sample by its mean over the observation interval \mathcal{T} . Both the amplitude and phase components are functional data extracted from the original samples. The amplitude component reflects the vertical variability of the functional data, whereas the phase component reflects the horizontal variability. Analogously to the functional boxplot by Sun and Genton [42] computed on the original dataset (although the computational methods are quite different), the authors proposed a method for constructing a functional boxplot for each of the three components so that outlying features may be identifying, and outliers may therefore be detected. Xie *et al.* extended this method to multivariate functional data and added other components such as shape orientation (reflecting rotational variability) [47]. They additionally pro-

Table 1

List of notations. A tilde always refers to objects related to the approximation functions. t_j can be an element of t_i , as well as an element of \mathcal{T} . By abuse of notation, we also use t_i to denote a vector of sampled observation points (t_1, \dots, t_{m_i}) .

| Notation | Description |
|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| $\mathcal{T} \subset \mathbb{R}$ | A closed real interval in which t lies |
| x_{ik} | Univariate function underlying the k -th variable of i for every $t \in \mathcal{T}$ |
| $X_i = (x_{i1}, \dots, x_{ip})$ | Multivariate function with p variables for every $t \in \mathcal{T}$ |
| $t_j \in \{t_1, \dots, t_{m_i}\} = t_i$ | An element of the observation points t_i (i.e., the observed discretization of \mathcal{T} with m_i points) |
| $X_i(t_j) \in \mathbb{R}^p$ | Measurement of X_i at the observation point t_j |
| $\tilde{X}_i = (\tilde{x}_{i1}, \dots, \tilde{x}_{ip})$ | Approximated multivariate function for every $t \in \mathcal{T}$ |
| $\tilde{\mathcal{T}} = \{t_1, \dots, t_j\}$ | Arbitrary discretization of \mathcal{T} |

purpose is to remove the noise, thus allowing accurate evaluations of some derived functions, such as combinations of derivatives and integral functions. This is necessary in our case, as the proposed mapping functions correspond to combinations of derivatives and integrals.

We should first select a functional representation as an approximation function. As a function is intrinsically infinite-dimensional, in FDA, it is commonly assumed that the underlying function can be approximated by a finite linear combination of non-linear basis functions. Such an approximation is called a basis expansion function [32]. We assume that x_{ik} , the k -th variable (hence a univariate function) of X_i , is to be approximated. The intuition behind the basis expansion is to combine a small number of “specific functions” (a set of given functions), each of which can capture some local features of the underlying function x_{ik} , so that x_{ik} could be recovered with a small approximation error. This approximation function can be formulated as

$$\forall t \in \mathcal{T}, \tilde{x}_{ik}(t) = \sum_{l=1}^{L_{ik}} \alpha_{ikl} \phi_l(t) = \alpha_{ik}^\top \boldsymbol{\phi}(t) \quad (1)$$

where $\boldsymbol{\phi}(t) = \{\phi_l(t)\}_{1 \leq l \leq L_{ik}}$ is a vector of orthonormal basis functions at t for some $L_{ik} \in \mathbb{N}^*$ (referred to as the basis size) with fewer basis functions than sampled observation points ($L_{ik} \ll m_i$), and $\alpha_{ik}^\top = \{\alpha_{ikl}\}_{1 \leq l \leq L_{ik}}$ is the coefficient vector, the element α_{ikl} of which is the importance of the l -th basis function.

Another choice of functional representation in FDA is to use non-parametric smoothing [16], which achieves a similar approximation, but its form is less tractable than that of the basis expansion function (for instance, to compute derivatives).

According to Eq. (1), one should select (i) the basis $\{\phi_l\}_{1 \leq l \leq L_{ik}}$ and (ii) the basis size L_{ik} .

The coefficient vector is computed from the data (see next paragraph).

The choice of the basis is data-dependent. As suggested by Ramsay and Silverman [32], when the data are smooth and periodic, the Fourier basis should be selected; when the data are smooth, a spline basis is suitable. A spline is a piecewise-polynomial function of order at least three [9]. If the data have irregularities, a wavelet basis should be preferred [31]. See [33] for other examples and details on the choice of the basis according to the data. The choice of the basis-size parameter L_{ik} depends on the selected basis. An inappropriate choice of the basis results in requiring a large L_{ik} because each basis function will focus on an irrelevant part of the data variability (low bias and high variance or, high bias and low variance); the worst case is to capture the noise, leading to over-fitting [32]. By contrast, an appropriate choice of the basis functions results in a small L_{ik} , that is, the basis is sufficiently rich to approximate an unknown function using few functions. Subsequently, once a suitable basis is selected, the bias–variance trade-off should be considered. This refers

vided useful visualization techniques for identifying outlying features (in fact, they only focused on the bivariate, $p = 2$, and trivariate, $p = 3$, cases, which are shape data extracted from images). However, when the size of the dataset and the number of variables p increase, this method is computationally costly, as the shape-based component-extraction procedures include several continuous optimization problems. Moreover, in these studies, the outlier-detection methods are based purely on the empirical distribution (through the functional boxplot) of the proposed geometrical features, whereas we map the original data to univariate functional data and subsequently use an outlier-detection algorithm. The latter can be seen as implicit non-parametric learning of the underlying distribution based on the functional data mapped to a geometric curve feature. Hence, we take advantage of both the geometrical mapping and the outlier-detection algorithm.

3. Background in functional data

This section is concerned with the handling of high-dimensional vectors of discrete noisy measurements that can be represented as smooth continuous functions; moreover we discuss how such representations can be achieved. A list of notations is provided in Table 1. The functional data representation is twofold: (i) As the \tilde{X}_i s are smooth functions the reconstructed data are noiseless. (ii) The reconstructed data are “aligned” in the sense that two reconstructed sample values $\tilde{X}_1(t_j)$ and $\tilde{X}_2(t_j)$ at t_j are comparable, as they refer to the same evaluation point $t_j \in \tilde{\mathcal{T}}$. This is not the case in raw data because one can have $t_{m_1} \neq t_{m_2}$ (the curves can be sampled on different grids).

3.1. Functional-data representation

The first step in FDA is to approximate an unknown smooth function $X_i : t \rightarrow \mathbb{R}^p$, which underlies the sample i , by another smooth approximation function $\tilde{X}_i(t)$, $\forall t \in \mathcal{T}$ through m_i discrete noisy measurements $X_i(t_1), \dots, X_i(t_{m_i})$, this is referred to as the functional approximation step. Its

to the balance between the approximation error and a reasonable L_{ik} [32]. Such a balance is generally achieved by a grid search by cross-validation for each sample i and variable k . When $\phi(t)$ and L_{ik} are specified, a computing method is required to estimate the coefficient vector α_{ik} , which is introduced in the next paragraph.

3.2. Functional-data fitting

The linearity of the basis expansion function with respect to the coefficient vector α_{ik}^\top enables its efficient estimation (assuming the data were sampled with a noise ϵ_{ij} , that is $x_{ik}(t_{ij}) = \tilde{x}_{ik}(t_{ij}) + \epsilon_{ij}$, where ϵ_{ij} is independent of $\tilde{x}_{ik}(t_{ij})$) by minimizing the least-squares criteria:

$$J(\alpha_{ikl}) = \sum_{j=1}^{m_i} (x_{ik}(t_{ij}) - \tilde{x}_{ik}(t_{ij}))^2 \quad (2)$$

or equivalently, with vector notation,

$$J(\alpha_{ik}) = \|x_{ik}(t_{i\cdot}) - \Phi_{ik}\alpha_{ik}\|^2 \quad (3)$$

where $\|\cdot\|$ stands for the l_2 -norm, and $\Phi_{ik} = (\phi_l(t_{ij}))_{1 \leq j \leq m_i, 1 \leq l \leq L_{ik}}$ is the $m_i \times L_{ik}$ matrix containing all the L_{ik} basis functions evaluated at the observation points. Thus, Φ_{ik} is a discretization over $t_{i\cdot}$ of the vector of orthonormal basis functions $a\phi(t)$ in Equation (1). As $L_{ik} \ll m_i$ and Φ_{ik} has all its columns linearly independent, by the orthonormality of the basis functions (and thus orthonormality of the columns of Φ_{ik}), $\Phi_{ik}^\top \Phi_{ik}$ is invertible. Hence, equating the gradient of J to $\mathbf{0}$ with respect to α_{ik} leads to the following minimizer:

$$\alpha_{ik}^* = (\Phi_{ik}^\top \Phi_{ik})^{-1} \Phi_{ik}^\top x_{ik}(t_{i\cdot}) \quad (4)$$

which is known as the classical least-squares solution [20].

However, as the data are fitted according to the basis functions, the *smoothness* of \tilde{x}_{ik} depends greatly on the influence on the basis functions. Consequently, \tilde{x}_{ik} may lack smoothness and overfit the data. To analyze such a noise influence, one can compute the derivative of \tilde{x}_{ik} , which is “excessively” variable if a large amount of noise remains in the approximation function. To ensure smoothness, the least-squares criteria should be minimized by penalizing the derivative(s) of \tilde{x}_{ik} with an amount $\lambda_k > 0$ as follows:

$$J_{\lambda_k}(\alpha_{ikl}) = \sum_{j=1}^{m_i} (x_{ik}(t_{ij}) - \tilde{x}_{ik}(t_{ij}))^2 + \lambda_k \int_{\mathcal{T}} (D^q \tilde{x}_{ik}(t))^2 dt \quad (5)$$

where $D^q = \frac{d^q(\cdot)}{dt^q}$ is the q -th derivative of $\tilde{x}_{ik}(t)$. More generally, D^q can be any linear combination of derivatives of x_{ik} , that is, a linear differential operator [32]. A penalization term including derivatives is also known as a *roughness penalty*. The parameter λ_k is arbitrary and can be computed by cross-validation. This is detailed in Section 5.3. Eq. (5) can be written using vector notation as follows:

$$J_{\lambda_k}(\alpha_{ik}) = \|x_{ik}(t_{i\cdot}) - \Phi_{ik}\alpha_{ik}\|^2 + \lambda_k \alpha_{ik}^\top \mathbf{R}_{ik} \alpha_{ik} \quad (6)$$

where $\mathbf{R}_{ik} = (\int_{\mathcal{T}} D^q \phi_j(t) D^q \phi_m(t) dt)_{1 \leq j \leq L_{ik}, 1 \leq m \leq L_{ik}}$ is a $L_{ik} \times L_{ik}$ positive semi-definite matrix. The matrix \mathbf{R}_{ik} contains the inner products of the q -th derivative of the L_{ik} basis functions. This matrix can be computed provided that the q -th derivative of the basis functions exists. In practice, it is common to choose $q = 1$ or $q = 2$ (i.e., to penalize the velocity or acceleration of \tilde{x}_{ik} , or a combination of both).

As J_{λ_k} remains quadratic with respect to α_{ik} , approximating \tilde{x}_{ik} with a roughness penalty is equivalent to ridge regression [21, 20]. Thus, the penalty term allows \tilde{x}_{ik} to (i) be smooth, as defined by the operator D^q and, (ii) avoid over-fitting by pushing the coefficient vector toward $\mathbf{0}$. Equating the gradient of J_{λ_k} to $\mathbf{0}$ with respect to α_{ik} leads to the following minimizer [20, 32]:

$$\alpha_{ik,\lambda}^* = (\Phi_{ik}^\top \Phi_{ik} + \lambda_k \mathbf{R}_{ik})^{-1} \Phi_{ik}^\top x_{ik}(t_{i\cdot}) \quad (7)$$

3.3. Approximation functions as building blocks

Once the coefficient vectors have been estimated for the p variables of the n samples (with or without penalization), we can consider the approximations \tilde{X}_{ik} to be smooth multivariate functions that well recover the underlying functions. Although these functions can be theoretically evaluated at an infinite number of points in \mathcal{T} , in practice, there are two methods to handle the approximations computationally (e.g., to compute *derived functions* such as derivatives and integrals):

- (i) The first method is to compute the derived functions based on the basis functions. As the basis functions are known analytically, their derived functions can also be obtained analytically. Thus, by the linearity of the basis expansion, one can easily obtain the derived functions of the approximation functions (the integral and derivative are linear operators). We illustrate this using the k -th derivative of the approximation function. We assume that an unknown function x is approximated by \tilde{x} through a basis expansion with a basis size L (in Eq. (1)), provided that the k -th derivative $\{D^k \phi_l(t)\}_{1 \leq l \leq L}$ of the basis functions exists, and the coefficient vector $\{\alpha_l\}_{1 \leq l \leq L}$ is available (or has been estimated as in Eq. (4)). The k -th derivative of \tilde{x} with respect to t is $D^k \tilde{x}$, where

$$\forall t \in \mathcal{T}, D^k \tilde{x}(t) = D^k \left(\sum_{l=1}^L \alpha_l \phi_l(t) \right) = \sum_{l=1}^L \alpha_l D^k \phi_l(t) \quad (8)$$

- (ii) The second method is to estimate the underlying functions by evaluating all the approximation functions on the same grid $\tilde{\mathcal{T}}$. Thus, from these estimates, one can compute derived functions, such as integral or derivatives, using numerical methods, such as quadrature

or finite difference schemes, respectively [41]. These methods are easy to implement, but they do not consider the basis functions and require that the arbitrary grid be sufficiently fine (so that the approximation functions are evaluated at a large number of observation points).

Thus, if the derivatives of the basis functions are known (as is the case for splines, Fourier basis functions, etc.), the derivatives of \tilde{x} are also known and need not be estimated from the raw data or the smooth reconstructions of the original data from \tilde{x} by a noise-sensitive and costly method such as finite differences. This example demonstrates the flexibility of the linear basis expansion for computing derived functions in FDA. Then, a derived function, for instance $D^1 \tilde{x}$, can be evaluated on an arbitrary grid. Such an approach is different from estimating the derivatives from an evaluation of \tilde{x} on the grid by using finite differences.

The first method is safer than the second because the analytic form of the basis functions is fully considered, and therefore the corresponding derived functions can be obtained accordingly. For instance, if the basis functions ϕ_l are B-splines (which are piecewise polynomial), we know the analytic form of $D^1 \tilde{x}$, as $D^1 \phi_l$ results in a piecewise polynomial as well. Thus, the evaluation of $D^1 \tilde{x}$ by the first method provides more accurate estimates of $D^1 x$ (which is unknown) than numerical methods applied to \tilde{x} evaluated on a fine grid of \mathcal{T} .

In the following part, we suggest some mapping functions for capturing functional outlyingness in the detection process. These mapping functions may have a complex analytical form because they involve several derivative functions (primarily first and second derivatives, as well as integral functions). Therefore, it is mandatory to have accurate evaluations of derivative functions, and accordingly we follow the first method in the computational experiments.

4. Shape-based representation for multivariate functional data

We regard a multivariate curve as a path lying in a p -dimensional space, specifically \mathbb{R}^p (see Fig. 1(a) for an example in \mathbb{R}^2), and derive mapping functions (aggregation functions of the variables), established in differential geometry, to capture shape features of the curves (e.g., length, velocity, or curvature) so that outlying features may be detected. These mapping functions have been used in shape analysis, for instance, to extract features based on the edge (bivariate curve) of an object in an image [40].

In this section, we investigate several mapping functions that enable the detection of multivariate functional outliers from the shape they exhibit in \mathbb{R}^p . Such mappings jointly consider the p variables, as they aggregate, in several ways, some derivatives (with respect to t) of the curve variables. Hence, the individual and collective variations of the variables are considered. These mapping functions take each data sample, represented by its smooth approximation function \tilde{X}_i , as

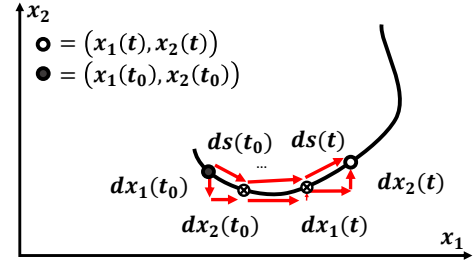


Figure 2: Arc-length mapping. The length of the curve between two observation points t_0 (dark-grey dot) and t (white dot) is defined as the sum of infinitesimal length elements $ds(t_0) \dots ds(t)$ along the curve (red diagonal arrows) for all t . The crossed-circle dots represent such points between t_0 and t .

input and return a univariate curve (i.e., the resulting aggregation) reflecting certain shape features. Hence, they provide a means to “summarize” the shape of a multivariate curve, in the sense given by the mapping function, and reduce the number of functional variables to one. The univariate function returned by a mapping function is then fed into an outlier-detection algorithm; this is detailed in Section 5. In the sequel, we simplify the notations by referring to a functional-data sample as an arbitrary curve $X = (x_1 \dots x_k \dots x_p)$ instead of $\tilde{X}_i = (\tilde{x}_{i1} \dots \tilde{x}_{ik} \dots \tilde{x}_{ip})$.

4.1. Arc-length mapping

The arc-length mapping function enables analyzing the length of a curve between two points in \mathcal{T} (see Fig. 2). Let $X(t)$ be an arbitrary curve depending on a continuous variable $t \in \mathcal{T}$. For $t_0 \in \mathcal{T}$ and $t_0 < t$, the length $s(t)$ of the curve that $X(\cdot)$ represents from t_0 to t is

$$s(t) = \int_{t_0}^t \|D^1 X(u)\| du = \int_{t_0}^t \sqrt{\sum_{k=1}^p \frac{dx_k(u)^2}{du}} du \quad (9)$$

where $\|\cdot\|$ stands for the l_2 -norm in \mathbb{R}^p . Hence, the arc-length maps an original functional-data sample to univariate functional data that represent the increases in the cumulative length of the underlying curve from the starting-point $X(t_0) = ((x_1(t_0), x_2(t_0)))$ to an arbitrary point $X(t) = ((x_1(t), x_2(t)))$ for $t > t_0$. Fig. 2 shows that the length of a bivariate curve between $X(t_0)$ and $X(t)$ is the infinite sum from t_0 to t of infinitesimal length elements $ds(\cdot)$ (aka integral), corresponding to an infinitesimal length element in each direction (x_1 and x_2) in \mathbb{R}^2 . We note that this mapping always returns a positive increasing function, as it computes the cumulative length of the initial curve. Moreover, the arc-length mapping function is not influenced by a warping (i.e., a horizontal deformation) of the curve² [40]. This mapping function can discern functional samples with a shape of

²Let $\alpha(\cdot)$ be a differentiable warping function i.e., a monotone non-decreasing function defined in $\mathcal{T} \rightarrow \mathcal{T}$. The arc-length mapping function on a warped functional datum X is equal to the arc-length mapping function on the initial unwrapped functional datum: $s(\alpha(t)) = \int_{t_0}^{\alpha(t)} \|D^1 X(\alpha(u))\| du = \int_{t_0}^t \langle D^1 X(\alpha(u)), D^1 X(\alpha(u)) \rangle^{1/2} du = \int_{t_0}^t D^1 \alpha(u) \langle D^1 X(\alpha), D^1 X(\alpha) \rangle^{1/2} du$,

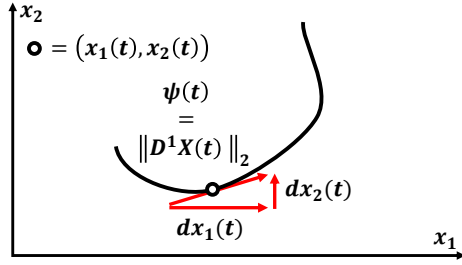


Figure 3: Velocity mapping. The norm of the tangent vector $D^1 X(t)$ (red diagonal arrow), the components of which are infinitesimal variations ($dx_1(t), dx_2(t)$) (shown by the horizontal and vertical red arrows) of the variables of the curve allows the computation of the speed at which the curve “progresses.”

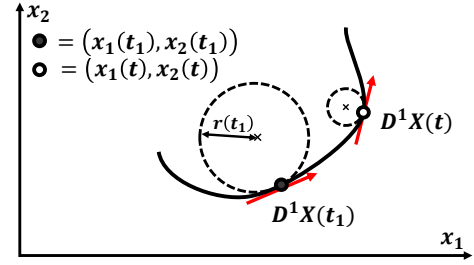


Figure 4: Curvature mapping. Curvature is defined to be the inverse of the radius of the osculating circle. In this example, in a neighborhood of the curve at t_1 (dark-grey dot), the *tangent vector* $D^1 X(t_1)$ has almost the same direction; hence, the osculating circle has a large radius ($r(t_1) = \frac{1}{\kappa(t_1)}$), resulting in a small curvature. In a neighborhood of the curve at t (white dot), the tangent vector $D^1 X(t)$ quickly changes direction; hence, the osculating circle has a lower radius, that is, a higher curvature than at t_1 .

different size, which is a global shape feature. Thereby, the detection of functional outliers can be improved when their underlying curve is longer or shorter than those of the other samples. For instance, an isolated outlier, which exhibits a peak for a small part of \mathcal{T} , induces a sharp increase in its curve length, whereas the length of other curves increases more slowly.

4.2. Velocity mapping

The velocity mapping function enables analyzing the instantaneous variations of the curve with respect to t . It has a simple interpretation when t corresponds to a time instant. In this case, velocity measures how fast a point moves on the curve. More generally, it can be interpreted as the norm of the projection of the curve onto $D^1 Y(t)$, the tangent vector to the curve at t . In Fig. 3, the velocity mapping at t of a bivariate curve is shown as the l_2 -norm $\|D^1 X(t)\|$ of the tangent vector $D^1 X(t)$ (vector of the first-order derivatives of the curve variables x_1 and x_2). It is defined as

$$\psi(t) = \|D^1 X(t)\| \quad (10)$$

and is related to the arc-length mapping by $\psi(t) = \frac{ds}{dt}$, or conversely, by $s(t) = \int_{t_0}^t \psi(t)dt$; however, these mappings capture different features. Indeed, the arc-length mapping outputs an increasing function and thus “memorizes” the local variations of the curve as t increases, whereas the velocity mapping characterizes the local variations (i.e., pointwise) with respect to t . The function returned by the velocity mapping may be regarded as a measure of the variation of the arc-length mapping. Thus, the velocity mapping can be used to identify the local outlyingness of a sample (isolated outlier).

4.3. Curvature mapping

Curvature is a notion that relates to how “bended” a curve is, or geometrically, the degree to which a curve deviates from the tangent line at a given point. An alternative interpretation

and as $D^1 \alpha(u) = \frac{d\alpha}{du}$, we have $\int_{t_0}^t D^1 \alpha(u) \langle D^1 X(\alpha), D^1 X(\alpha) \rangle^{1/2} du = \int_{t_0}^t \langle D^1 X(\alpha), D^1 X(\alpha) \rangle^{1/2} d\alpha = \int_{t_0}^t \|D^1 X(\alpha)\| d\alpha$, which implies that $s(\alpha(t)) = s(t)$.

concerns the radius of the osculating circles. At a given point t , a smaller radius of the osculating circle implies larger curvature. In fact, the radius of the osculating circle is equal to the inverse of the curvature at this point. The bivariate curve in Fig. 4 shows that at a neighborhood of t_1 where the tangent vector $D^1 X(t_1)$ has almost constant direction, the osculating circle has a larger radius $r(t_1)$ than the radius of the osculating circle at a neighborhood of t where the direction of the tangent vector $D^1 X(t)$ changes quickly. Thus, the curvature mapping function allows analyzing the change of direction of the curve with respect to t . Indeed, if the curve is a line, curvature is constant, and the curve directions remain constant as well. Curvature is defined as

$$\kappa(t) = \frac{\|D^1(\frac{D^1 X(t)}{\|D^1 X(t)\|})\|}{\|D^1 X(t)\|} \quad (11)$$

or equivalently,

$$\kappa(t) = \frac{\sqrt{\|D^1 X(t)\|^2 \|D^2 X(t)\|^2 - \langle D^1 X(t), D^2 X(t) \rangle^2}}{\|D^1 X(t)\|^3} \quad (12)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product in \mathbb{R}^P . We now provide insight into the definition of κ in Eq. (11). $\frac{D^1 X(t)}{\|D^1 X(t)\|}$ is the direction vector (i.e., the normalized tangent vector); therefore, $D^1 \frac{D^1 X(t)}{\|D^1 X(t)\|}$ is the rate of change of the direction vector, and the normalization $\|D^1 X(t)\|$ relates to the rate of change of the direction with respect to the tangent vector. Consequently, the curvature mapping can detect functional outliers with a curve that exhibits a differently bended shape than those of the other samples.

5. Experimental study

We conducted an experimental study on real and synthetic datasets to demonstrate the effectiveness of the proposed mapping functions in improving outlier detection in multivariate

functional data. The detection performance was evaluated in terms of the true detection rate (i.e., the proportion of outliers correctly detected), false detection rate (i.e., the proportion of outliers falsely detected), and area under the ROC curve (AUC).

5.1. Real data

5.1.1. ECG data

We tested the proposed approach on the real dataset used by Dai and Genton in [8]. The dataset consists of electrocardiogram (ECG) time series of the electrical activity of heart changes [19]. Such data can reveal abnormalities in heart activity. The time series are univariate and were labeled by cardiologists as *abnormal* or *normal*. This dataset has been used for time-series classification [46]. We augmented the data set by bivariate time series to demonstrate the applicability of the method to multivariate time series.

There are a total of $n = 810$ time series including 208 abnormal and 602 normal cases. All the time series have an equal size of $m_i = 86$. In contrast with Dai and Genton [8], who only considered the time series between the time stamps $t = 6$ and $t = 80$ to avoid boundary effects, we considered the entire time series to demonstrate the robustness and applicability of the proposed approach. Dai and Genton also augmented the univariate time series to multivariate by adding the first and the second derivatives. We did not follow this, as in the proposed approach, these aspects are considered (e.g., velocity mapping in Eq. (10) or curvature mapping in Eq. (11)); rather, we added the squared time series. Indeed, power is proportional to the square of voltage. Thus, in terms of interpretability, this data augmentation appears to be more relevant than that by the second derivative of voltage. We applied the same multivariate functional data augmentation to all ECG-data experiments and for all methods; we did not apply the derivative augmentation, as this would bias the interpretation of the results, that is, it would not be possible to discern whether the results were due to the specific augmentation or to the method. This would be of interest if the focus was specifically on the ECG data, but here, we use it as a real dataset example.

As in [8], to obtain a rare class of samples representing outliers, we randomly created a partition of 400 samples (i.e., the training set) out of the 810 samples by parameterizing the contamination level (i.e., the rate of *abnormal* samples) in this partition to 5%, 10%, 15%, 20%, and 25%. Then, for each contamination level, we evaluated the proposed method on the 410 remaining samples (i.e., the test set).

5.1.2. Pen-digits data

We also tested the proposed method on another real dataset consisting of $n = 10992$ bivariate time series representing pen digits (PenDig) [12]. The digits are labeled according to their class (i.e., from 0 to 9). Each digit has $m_i = 8$ observation points regularly sampled on both the horizontal and vertical coordinates. As this initial dataset cannot be considered high-dimensional, we upsampled it by linear interpolation to $m' = 200$ on the two coordinates before fitting the approximation functions.

To simulate the outlier classes, we considered a single digit to be the outlier class, and the nine other classes to be the inlier class, as in [36]. The training set was generated using 75% of the entire dataset with a contamination level equal to $c = 5\%$ (i.e., 5% of the training set are outliers). Each digit was separately considered the outlier class, and thus the experiment was conducted in 10 independent ways. Then, for each case of outlier class, we assessed the proposed method on the test set.

5.2. Synthetic data

We simulated multivariate functional data sets according to the five models proposed by Dai and Genton in [8]. To the best of our knowledge, this is the most recent study concerned with outlier detection in multivariate functional data providing performance results (detection rates). For each of the five simulation models, $n = 150$ bivariate curves were generated on a regular grid of size $m = 200$ in the real interval $[0, 1]$. Among the n curves, $c = 10\%$ (referred to as the contamination level) were outliers. Regardless of the simulation model, all uncontaminated curves were simulated according to a unique uncontaminated model U (except model 5). Hence, the models 1, 2, 3, 4 had a common uncontaminated model U (Eq. (13)) and different contamination models X_{c1} , X_{c2} , X_{c3} , X_{c4} , respectively, which generated the two classes of outliers (isolated and persistent). We recall that, compared with the rest of the dataset, isolated outliers exhibit outlying behavior in a small part of the domain \mathcal{T} , whereas persistent outliers exhibit outlying behavior in a large part of \mathcal{T} . Testing the proposed approach and the baselines using different types of outliers enables assessing the efficiency of each mapping function in a given context.

The uncontaminated model was simulated according to a bivariate Gaussian process $\mathcal{GP}(\mu(t), \Sigma(s, t))$ [34], with a constant mean function $\mu(t) = \mathbf{0}$, and a cross-covariance function C_{kr} between the two variables indexed by k and r , as follows:

$$C_{kr}(s, t) = \rho_{kr} \sigma_k \sigma_r \mathcal{M}(|s-t|; \nu_{kr}, \beta_{kr}) \quad k, r = 1, 2 \text{ and } s, t \in [0, 1]$$

where ρ_{12} is the correlation between the variables x_1 and x_2 , $\rho_{11} = \rho_{22}$ is the variance of each variable, σ_1 and σ_2 are the marginal variances,

$\mathcal{M}(h; \nu_{kr}, \beta_{kr}) = 2^{1-\nu} \Gamma(\nu)^{-1} (\beta|h|)^\nu \mathcal{K}_\nu(\beta|h|)$ is the Matérn class function [30] (\mathcal{K}_ν is a modified Bessel function [3]), $\nu_{kr} > 0$ is a smoothness parameter, and $\beta_{kr} > 0$ is a range parameter. For this simulation, we used the same parameter setting as in [8]: $\rho_{12} = 0.6$, $\rho_{11} = \rho_{22} = 1$, $\sigma_1 = \sigma_2 = 1$, $\nu_{11} = 1.2$, $\nu_{22} = 0.6$, $\nu_{12} = \nu_{21} = 1$, $\beta_{11} = 0.02$, $\beta_{22} = 0.01$, and $\beta_{12} = \beta_{21} = 0.016$. This covariance function is implemented in the R package [37]. We summarize the uncontaminated model $U(t) = (u_1(t), u_2(t))^T$ as follows:

$$U(t) \sim \mathcal{GP} = \left(\mu(t) = (0, 0)^T; \Sigma(s, t) = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \right) \quad (13)$$

The five contamination models are (we annotate the variables with an index c referring to “contamination”):

1. Model 1 (persistent magnitude outlier): $X_{c1}(t) = 4U(t)$
2. Model 2 (isolated outlier): $X_{c2}(t) = U(t)(1+11\mathbf{I}_{Z < t < Z+0.1})$ where \mathbf{I} is the indicator returning 1 if the indexed condition is true, and 0 otherwise, and Z is a uniform random variable in $[0, 0.9]$.
3. Model 3 (persistent magnitude outlier), the contamination model is different for the two variables: $X_{c3}(t) = (x_{1,c3}(t), x_{2,c3}(t))^T$, with $x_{1,c3}(t) = 1.7u_1(t)$ and $x_{2,c3}(t) = 1.5u_2(t)$.
4. Model 4 (isolated outlier): $X_{c4}(t) = U(t)(1+4\mathbf{I}_{Z < t < Z+0.1})$, with Z as in model 2.
5. Model 5 (persistent shape outlier), the new uncontaminated model is referred to as Y , and the contamination model as X_{c5} : $Y(t) = (y_1(t), y_2(t))^T$ with $y_1(t) = u_1(t) + Z_{11} \cos(4\pi t)$ and $y_2(t) = u_2(t) + Z_{12} \sin(4\pi t)$ where Z_{11} and Z_{12} are independent uniform random variables in $[2, 3]$. The contamination model X_{c5} is $x_{1,c5}(t) = u_1(t) + Z_{21} \cos(4\pi t)$ and $x_{2,c5}(t) = u_2(t) + Z_{22} \sin(4\pi t)$, where Z_{21} , Z_{22} are uniform random variables on $[4, 5]$.

5.3. Experimental protocol

5.3.1. Functional approximation

Without loss of generality, we selected $\mathcal{T} = [0, 1]$ as the domain (closed interval) of t for all the data sets. We recall that we represent all the curves in the common interval \mathcal{T} because we assume that the functional samples were generated by a random function depending on t relating to the same event in \mathbb{R}^p . For instance, when the samples are measurements of a given process depending on t , which represents time, \mathcal{T} can be viewed as the relative temporal range of the process (i.e., from the beginning at $t = 0$ to the end at $t = 1$) and $t \in \mathcal{T} = [0, 1]$ can be interpreted as the progress rate of the process.

Choice of the basis of functions For the ECG and the PenDig datasets, we approximated each variable of the bivariate time series by a basis consisting of B-splines of order eight (B-splines are piecewise-polynomial functions of order at least three, and are located at a given observation point $t \in \mathcal{T}$). Indeed, we noticed that in this dataset, the curves exhibit a smooth pattern without periodicity; hence, the B-spline basis is a suitable choice (as recommended in [32]).

For the synthetic dataset, we approximated each variable of the bivariate time series by a Fourier (sine and cosine functions) basis with a fundamental period of $T = \frac{1}{F} = 1$ (i.e., the length of \mathcal{T}). The Fourier basis was deemed suitable because we noticed low-frequency periodicity (induced by the covariance function $C_{kr}(s, t)$) over \mathcal{T} .

Application of the functional-data fitting procedure We now provide the computational details of the functional-data

fitting. Following the recommendations in [33, 15], for all datasets, we selected both the penalization λ_k and the basis size L_{ik} for the variable k of sample i through a leave-one-out cross-validation procedure over a given grid search for λ_k and L_{ik} . We penalized both the first- and second-order derivatives of \tilde{x}_{ik} to gain smoothness in the mapping-function output. We note that for all the samples of a given variable k , we equally penalized the approximations \tilde{x}_{ik} by the same λ_k to compute the coefficient vector α_{ik}^* . Then, by computing the coefficient vector α_{ik}^* according to Eq. (7), we selected the value of λ_k and $L_{ik} < m_i$ that minimize the leave-one-out cross-validation score $CV_{\lambda_k}(L_{ik})$,

$$CV_{\lambda_k}(L_{ik}) = \sum_{j=1}^{m_i} \left(x_{ik}(t_j) - \tilde{x}_{ik}^{-j}(t_j) \right)^2 \quad (14)$$

where \tilde{x}_{ik}^{-j} corresponds to the approximation of x_{ik} by L_{ik} basis functions by omitting the pair $(t_j, x_{ik}(t_j))$ in the functional-fitting step, as in Eq. (5), where the penalization is λ_k .

For the ECG and PenDig datasets, the grid search of λ_1 and λ_2 was fixed on logarithmic scale in $[-9, -1]$, with a thickness of 0.1. The grid search of L_{ik} was fixed at the integers between 35 and 60, that is, for a given integer $L_{ik} \in \llbracket 35, 60 \rrbracket$, the L_{ik} B-spline functions were regularly placed in \mathcal{T} .

For the synthetic datasets, the grid search of λ_1 and λ_2 was fixed on logarithmic scale in $[-9, -4]$, with a thickness of 0.1. The grid search of L_{ik} was fixed in $\llbracket 20, 25 \rrbracket$, that is, for a given integer $L_{ik} \in \llbracket 20, 25 \rrbracket$, the synthetic data were approximated by the first L_{ik} frequencies $2\pi \times F \times 1, \dots, 2\pi \times F \times L_{ik}$. Then, for each variable, we retained the coefficient vector associated with both the optimal regularization and basis-size parameters to recover the smooth approximation function $\tilde{X}_i = (\tilde{x}_{i1}, \tilde{x}_{i2})$.

Finally, we used the coefficient vector associated with both the optimal regularization and basis-size parameters to recover the smooth approximation functions \tilde{X}_i on a given grid and applied a mapping function to them.

5.3.2. Applying the mapping functions

We now explain the computational application of the mapping functions and then how their output was fed to an outlier-detection algorithm.

After computing the approximation functions \tilde{X}_i , we centered and scaled each variable x_{ik} with the empirical mean and standard deviation functions computed from the training set (see [32] for details on the computation of mean standard deviation functions). This scaling prevents the mapping functions from overweighting some variables with a wider range than others. Indeed,

- (i) The variables require to be scaled since the unit of the output value of the arc-length mapping function (Len_{map} in Eq. (9)) is intrinsically a length. Then, we applied the three mapping functions introduced in Section 4. As the arc-length mapping is the integral

function of the velocity mapping, the arc-length map-
ping in Eq (9) was computed from the minimum of \mathcal{T}
(i.e., $t = 0$) and was then integrated up to t for all $t \in \mathcal{T}$.
In these experiments, the integral was efficiently approx-
imated by a Riemann sum, as in this study, all the
observation points were regularly sampled in \mathcal{T} , and
therefore the sum converges to the integral. We note
that if the observation points had been irregularly sam-
pled, the integral could have not been approximated
by a Riemannian sum, and numerical techniques, such
as Simpson's or the trapezoidal rule, should have been
used instead [32].

(ii) Regarding the velocity mapping V_{map} in Eq. (10), the
first-order derivative of each variable of \tilde{X}_i was com-
puted according to Eq. (8).

(iii) The curvature mapping ($Curv_{map}$) requires the compu-
tation of both first- and second-order derivatives. Thus,
we computed them as in Eq. (8) and combined them
as in Eq. (11).

The approximation functions recover the functional data
on the entire domain \mathcal{T} . Thus, the approximation functions
can be computed on an irregular grid, and therefore the com-
putation of the mapping functions should be carefully per-
formed (e.g., (i) in the computation of an integral function).
For both V_{map} and $Curv_{map}$, which are based on derivative
functions only, simple and efficient derivative estimation
methods can be used, as mentioned in Section 3.3.

Each mapping function returns a univariate function. Thus,
applying a mapping function to all n approximation func-
tions \tilde{X}_i results in n univariate functional-data samples. We
used the resulting univariate functional data in several outlier-
detection algorithms. In practice, the functions returned by
a mapping function should be evaluated over a grid of ob-
servation points in \mathcal{T} to obtain the output samples in vector
form. As we selected $\mathcal{T} = [0, 1]$ for all datasets and the
observation points are regular, the grid is a regular discretiza-
tion $\{t_1 \dots t_j \dots t_J\}$ of \mathcal{T} with a thickness of $\frac{1}{J}$ ($t_1 = 0$ and for
 $j > 1, t_j = \frac{j}{J}$). Hence, for the outlier-detection algorithms,
the data correspond to J -dimensional numerical vectors that,
in turn, correspond to univariate functional data output by
a mapping function. We selected the thickness of the grid
as the original size of the time series for both the synthetic
and ECG datasets (ECG data set: $J = m_i = m = 86$, PenDig
dataset: $J = m' = 200$, synthetic data sets: $J = m = 200$).
An irregular grid can also be used to evaluate the approxi-
mated functions, but the computation of the mappings should
be performed cautiously, as mentioned in (i) for Len_{map} .

5.3.3. Outlier detection from the functional output of a mapping function

We detect outliers in the functional data returned by a
mapping function using a state-of-the-art outlier-detection
algorithm. To this end, we selected isolation forest (iFor) [27]
and a one-class support vector machine (OCSVM) [38]. iFor
is a bagging model that generates a large number of decision

trees grown on random subspaces. A subspace corresponds
to a subsample of features randomly selected from the full
feature space (here, $\{1 \dots j \dots J\}$). Each tree isolates the data
samples based on a random split value of a randomly selected
feature from the subspace until all the data samples have been
isolated, or all the features of the subspace have been selected.
The sample outlyingness score returned by a tree is based on
the path length between the root node and the terminal node
of a tree. Outliers are samples that are easy to isolate and
thus have short path length in the trees. The path length is
normalized in $[0, 1]$ so that if the score is close to 1, then the
sample is likely an outlier. OCSVM is a distance-based model
formulated as a constrained quadratic minimization problem,
the variables of which correspond to the radius and the center
of the smallest hypersphere containing the data. To allow
flexibility on the hypersphere boundary owing to the presence
of outliers in the training data, slack variables are introduced
in the objective function in addition to the two other variables.
The hyperparameter ν corresponds to an upper bound on the
a priori proportion of outliers in the training set. A sample is
declared as an outlier if it lies outside the fitted hypersphere.
We used the radial-basis-kernel version of OCSVM with ν
equal to the exact proportion of outliers in the training set.
The bandwidth hyperparameter of the radial basis kernel was
optimized by a 20-fold cross-validation procedure.

For the ECG and PenDig datasets, we set the number of
trees to 1000, and the subsampling size to 32 [27]. For the
synthetic datasets, we also set the number of trees to 1000,
and the subsampling size to 16. We randomly split each
dataset into a training set and a test set. As in [8], the training
set represents 50% of the data for the ECG dataset. The train-
ing set for the PenDig dataset consists of 75% of the entire
dataset. The training set contains 60% of the data for the
synthetic data. The training set was used to both fit the model
(iFor and OCSVM) and select an outlyingness threshold from
the ROC curve that discriminates inliers from outliers. We
then computed the outlyingness score of the test samples
and achieved detection using the previously computed out-
lyingness threshold. Regarding OCSVM, we finetuned the
bandwidth hyperparameter of the radial basis kernel on the
training set through a 20-fold cross-validation procedure on
the grid $\{2^{-25} \dots 2^{-5}\}$ for the ECG dataset as well as the syn-
thetic data. In addition to the true and false detection rates
(ρ_c and ρ_f , respectively), as a measure of discrimination be-
tween outliers and inliers by the proposed approach, we also
computed *AUC* from the labels of the test set.

The threshold-selection step is simple and is not part of
iFor [27] or OCSVM [38], which are both unsupervised. We
assume that the training data is labeled even if there are few
outlier samples. In real-world applications, the user has some
knowledge about the training data and can thus label inliers
and some outliers. If the training set surely has no outlier,
the proposed method only requires the modification of the
threshold selection rule. This modification is easy because
both iFor and OCSVM are unsupervised methods and output
a normalized score. Using the detection rule obtained by the
threshold, we compute two performance measures ρ_c and

ρ_f to demonstrate the complete application of the proposed method and compare it with the baselines. In fact, there are other methods for learning an outlyingness threshold, such as using a specific decision rule involving, e.g., an empirical quantile associated with a reference distribution of the outlyingness scores [8], or threshold selection from the mass-volume curve [5] when no outlier label is available, but this is beyond the scope of the present study, as we assume that the training set has low non-zero contamination level.

To assess the proposed method with respect to a ground truth and without considering a threshold, we also evaluate the results using AUC , which is a measure of discrimination between outliers and inliers. It is a standard performance measure in outlier detection [13, 27] and demonstrates that the proposed method can outperform the baselines regardless of the computed outlyingness threshold.

5.4. Baseline comparisons

We compared the proposed approach with two recent outlier-detection methods based on multivariate functional depth (Section 2).

The first baseline method is FUNTA, proposed by Kuhn and Rehage in [26] (see Section 2). It only requires centering each variable x_{ik} of each sample to a zero mean. As FUNTA has been demonstrated to be robust to noise and can handle curves of different size, we used it on the raw data without any functional data approximation. For the computation of the outlyingness threshold, we applied the same procedure as in the proposed method, that is, we selected the best outlyingness threshold for the training set using ROC and applied it to the test set. We used the R implementation proposed [35].

The second baseline method is *Dir.out* proposed by Davenport and Genton in [8] (see Section 2). We used the same parameter setting as in [8] and did not perform any functional data approximation. In this method, the outlyingness score is based on the robust Mahalanobis distance of the directional outlyingness vector computed on a subset of the data; in the present case, we computed it using the training data to obtain comparable results and to assess the performance measure on the test set. The tail of the distribution of the distances approximated by an F -distribution with degrees of freedom $(p + 1, m - p)$, where p is the number of curve variables, and m is calculated through a simulation procedure (see [8], p. 7 for details). Consequently, the outlyingness threshold is not data-driven and is computed as a quantile of probability 99,3% of an F -distribution. Then, we used the outlyingness threshold on the test set to assess performance. We used the R implementation provided by the authors.

5.5. Experimental protocol application

The performance of the proposed approach was evaluated by simulation for both the real and the synthetic data. The simulation settings for the ECG and synthetic data were as in [8]. We proceeded as follows:

- (i) We randomly generated a train/test split. For the ECG data, the training set corresponds to 50% of the full

dataset, for the PenDig data, the training set is 75% of the dataset, and for the synthetic data, the training set represents 60% of the full dataset.

- (ii) We then applied the proposed and the baseline methods. Except for *Dir.out* (baseline), which does not require outlyingness-threshold learning because the outlyingness score follows a known distribution (see Section 5.4), the outlyingness threshold was learnt on the training set based on the ROC curve.
- (iii) We evaluated the performance in terms of the true detection rate (ρ_c), false detection rate (ρ_f), and AUC on the test set.

For the ECG dataset (resp., PenDig dataset), steps (i) to (iii) were repeated 50 times for each case of the five contamination levels (resp., for the 10 outlier classes) (see end of Section 5.1), and 500 times for the synthetic data for each of the five models (Section 5.2).

The two real datasets are not used to assess the same properties of the proposed method. The ECG data are used to demonstrate the robustness of the proposed method with respect to different contamination levels for some given outliers, whereas the PenDig data are used to assess the detection performance for different outliers and a given contamination level. Thus, we only compare these two in terms of performance, in the comparison of the various methods in Section 5.6.4.

5.6. Results and discussion

We report the results for the ECG dataset in Table 2, where for each contamination level c (columns) and for each method (rows), we provide ρ_c , ρ_f , and AUC (sub-columns). The results for the PenDig dataset are shown in Table 3, where for each case of outlier class (columns), that is, a single digit, and for each method (rows), we provide the three performance measures as in Table 2. The results for the synthetic data are reported in Table 4, where for each model (columns) and for each method (rows), we provide the three performance measures as in Table 2. In these tables, the value in a cell is the average of a performance measure over the number of simulations. We discuss the results below.

5.6.1. ECG data

The results for the ECG data set (Table 2) demonstrate that the proposed method outperforms the baselines with V_{map} and $Curv_{map}$ (V_{map} and $Curv_{map}$ rows with iFor and OCSVM, which are described in Section 5.3.3).

It can be seen that both V_{map} and $Curv_{map}$ (with iFor and OCSVM), provide constant ρ_c , ρ_f , and AUC values with respect to the five contamination levels (V_{map} and $Curv_{map}$ rows). We highlight this in Fig. 5, where it can be seen that the proposed method (except for Len_{map} with both iFor and OCSVM) outperforms the baselines in terms of the three performance measures, which remain constant as the contamination level changes. This shows that the outlying features captured by these mapping functions are more robust to the contamination level than those captured by the baselines.

Table 2
Results on the ECG dataset.

Outlier detection results for the *ECG data set* with five contamination levels c . For each contamination level (columns) and each performance measure (sub-columns), we marked the best results in bold (i.e., highest correct detection rate ρ_c and AUC , and lowest false detection rate ρ_f). For all the contamination levels, the proposed method achieves the best results with V_{map} and the $Curv_{map}$. We also notice that, in the proposed method, for a given mapping function and outlier-detection algorithm, performance does not degrade when c varies, whereas for *FUNTA* and *Dir.out*, performance degrades as c increases. The proposed functions outperform state-of-the-art methods when there are few outliers.

| Methods | $c = 5\%$ | | | $c = 10\%$ | | | $c = 15\%$ | | | $c = 20\%$ | | | $c = 25\%$ | | |
|---------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | ρ_c | ρ_f | AUC | ρ_c | ρ_f | AUC | ρ_c | ρ_f | AUC | ρ_c | ρ_f | AUC | ρ_c | ρ_f | AUC |
| <i>FUNTA</i> (baseline) | 0.85 | 0.60 | 0.78 | 0.86 | 0.50 | 0.81 | 0.88 | 0.42 | 0.83 | 0.87 | 0.29 | 0.85 | 0.85 | 0.24 | 0.86 |
| <i>Dir.out</i> (baseline) | 0.88 | 0.18 | 0.90 | 0.84 | 0.16 | 0.89 | 0.75 | 0.14 | 0.89 | 0.63 | 0.13 | 0.87 | 0.55 | 0.10 | 0.86 |
| iFor(V_{map}) | 0.90 | 0.12 | 0.96 | 0.92 | 0.12 | 0.96 | 0.92 | 0.12 | 0.96 | 0.92 | 0.13 | 0.95 | 0.91 | 0.13 | 0.95 |
| iFor($Curv_{map}$) | 0.89 | 0.07 | 0.98 | 0.90 | 0.07 | 0.98 | 0.91 | 0.08 | 0.98 | 0.90 | 0.08 | 0.97 | 0.91 | 0.08 | 0.97 |
| iFor(Len_{map}) | 0.54 | 0.28 | 0.70 | 0.49 | 0.24 | 0.69 | 0.45 | 0.20 | 0.68 | 0.42 | 0.19 | 0.66 | 0.43 | 0.23 | 0.65 |
| OCSVM(V_{map}) | 0.97 | 0.10 | 0.98 | 0.97 | 0.16 | 0.97 | 0.88 | 0.17 | 0.92 | 0.90 | 0.13 | 0.94 | 0.88 | 0.18 | 0.92 |
| OCSVM($Curv_{map}$) | 0.96 | 0.17 | 0.95 | 0.96 | 0.21 | 0.93 | 0.90 | 0.20 | 0.91 | 0.91 | 0.22 | 0.91 | 0.90 | 0.23 | 0.89 |
| OCSVM(Len_{map}) | 0.79 | 0.20 | 0.86 | 0.71 | 0.23 | 0.78 | 0.54 | 0.21 | 0.67 | 0.65 | 0.27 | 0.72 | 0.58 | 0.28 | 0.66 |

1080 The outlier detection with OCSVM from V_{map} and $Curv_{map}$ does not present the same robustness to the contamination level as that with iFor in terms of ρ_f (OCSVM(V_{map}) and OCSVM($Curv_{map}$), and Fig. 2). Indeed, ρ_f increases as the contamination level c increases. Accordingly, OCSVM appears to be more suitable for datasets containing a small number of outliers. This was also observed in [11]. Despite the lower robustness, OCSVM(V_{map}) and OCSVM($Curv_{map}$) are better than the baselines, which exhibit performance degradation as the contamination level c changes. Indeed, *FUNTA* is approximately constant as c increases but degrades for small values of c in terms of ρ_f (*FUNTA* row, ρ_f columns). Conversely, *Dir.out* is as robust as OCSVM(V_{map}) in terms of ρ_f (we note that the range of ρ_f is the same for *Dir.out* and OCSVM(V_{map})) but degrades in terms of ρ_c for high values of c (*Dir.out* row, ρ_c columns). Thus, we recommend using OCSVM when the contamination level is low [*Curv_{map}*, for OCSVM and iFor, is the most efficient mapping function in terms of ρ_f (ρ_f columns, *Curv_{map}* rows), and V_{map} is the most efficient in terms of ρ_c (ρ_c columns, V_{map} rows). Len_{map} has the worst performance (Len_{map} rows, ρ_c and AUC columns).

5.6.2. PenDig data

1102 From the results on the PenDig dataset in Table 3, it can be seen that the proposed method always outperforms the baselines in terms of AUC . This implies that the baselines are not as effective in capturing shape outlying features. When the outliers are ‘0’ digits, the results by the baselines are consistent with the results on the synthetic data when some shape outliers are simulated (Model 5 in Table 4). This is not surprising, as Model 5 generates bivariate functional outliers with an elliptic shape in \mathbb{R}^2 ; hence, a zero-like shape (‘0’). An AUC value close to 0.50 implies that the detector performs as efficiently as a random method, we note that the ‘0’ outlier case is the only in which the baselines are effective. The baseline methods cannot distinguish different shape outliers with abrupt shape irregularities such as (smooth) right angles for example, when the outlier is the ‘1’, ‘4’, or ‘5’ digit.

such cases, we obtain the best results in terms of AUC with V_{map} . For more regular shapes, such as ‘3’, ‘6’, ‘8’, and ‘9’, the best results are achieved by $Curv_{map}$.

5.6.3. Synthetic data

For isolated outliers (Table 4, Model 2 and Model 4 columns), the results on the synthetic datasets demonstrate that these outliers are well detected by the baseline methods as well as the proposed with Len_{map} and V_{map} with iFor. Indeed, as an isolated outlier exhibits large deviation in a small part of \mathcal{T} , its underlying curve is longer than that of most samples. Moreover, in these models, as the first derivative is considered, the velocity quickly changes in the part of the domain where the isolated outlyingness occurs; thus, the V_{map} function is an appropriate candidate for detecting isolated outliers. *Dir.out* has the best performance in terms of both ρ_c and ρ_f . Regarding Model 2, the proposed model outperforms *FUNTA* with Len_{map} , and V_{map} with iFor. $Curv_{map}$ exhibits poor performance for the two models. This implies that it is ineffective in detecting isolated outliers. Indeed, the contamination models (Model 2 and Model 4, Section 5.2) generate stationary functional data (constant mean and only lag-dependent covariance) except in the part of \mathcal{T} where the outlyingness occurs (here, a short peak). Thus, considering the second-order variations (second-order derivatives in Eq. (11)) is irrelevant and leads to high ρ_f values (ρ_f columns and $Curv_{map}$ rows). Moreover, there is a low correlation between the curve variables, and thus $Curv_{map}$, which captures deeper correlation features (bending in the curve, see Eq. (11)) is not appropriate in this case.

For persistent magnitude outliers (Table 4, Model 1 and Model 3 columns), *Dir.out* and *FUNTA* yield the best results in terms of both ρ_c and ρ_f . We obtain highly similar results for Model 1 with V_{map} , and Len_{map} with iFor. Nevertheless, V_{map} is not as efficient for Model 3 as for Model 1. Indeed, Model 1 has high contamination (high, short peak), resulting in high velocity mapping values, and we recall that velocity and curvature relate to local variations of the curves. Consequently, as magnitude outlyingness is a global shape

Performance measures vs. contamination level

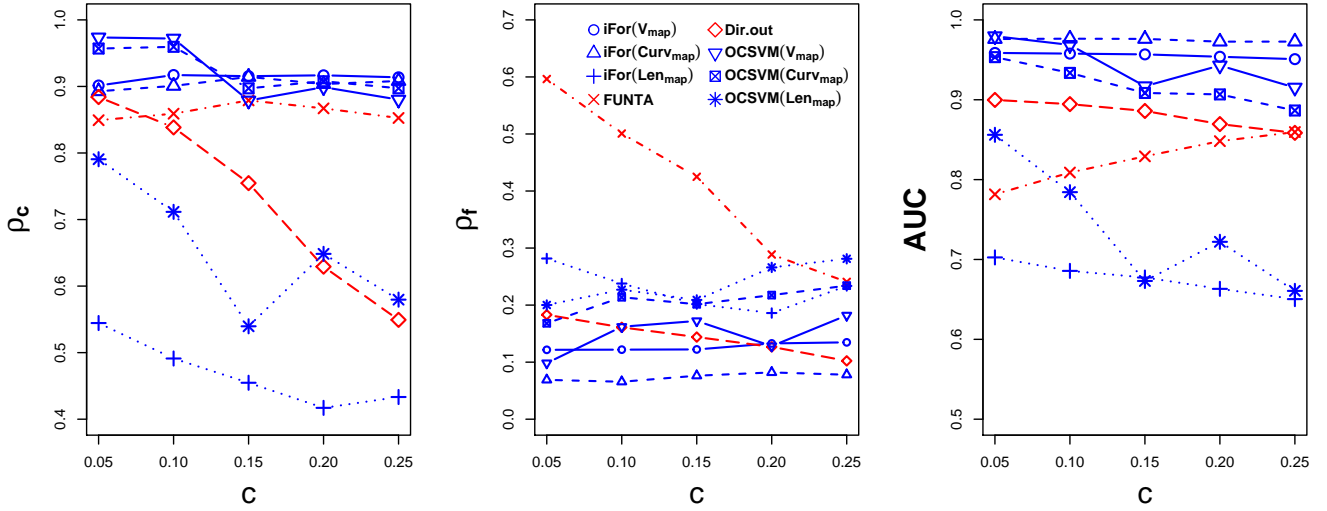


Figure 5: Performance on ECG data. The three performance measures ρ_c , ρ_f , and AUC , averaged over the number of simulations as functions of the contamination level ($c = 5\%$, $c = 10\%$, $c = 15\%$, $c = 20\%$, and $c = 25\%$) for each method (proposed in blue, and baselines in red). We notice that when the contamination level c increases, the proposed method (except for iFor(Len_{map}) and OCSVM(Len_{map})) outperforms the baselines in terms of ρ_c , ρ_f and AUC . Moreover, performance does not degrade as the contamination level changes, in contrast with that of the baselines. In terms of ρ_c , FUNTA performs as well as V_{map} and $Curv_{map}$ when used with both iFor and OCSVM but significantly degrades in terms of ρ_f (i.e., it falsely detects outliers) for low contamination levels. $Dir.out$ performs as well as the proposed method in terms of ρ_f but degrades in terms of ρ_c for high contamination levels. Hence, FUNTA performs well when the contamination level is high, and $Dir.out$ performs well when the contamination level is low.

Table 3

Results for the PenDig dataset.

Outlier detection results for the *PenDig* dataset when each of the 10 classes ('0'...'9') is considered an outlier (columns), and the nine other classes inliers. For each case of outlier class and each performance measure (sub-columns), we marked the best results in bold. It can be seen that for the ten cases, the proposed method is considerably better than the baselines, which are inefficient for this dataset except when the outliers are '0' digits.

| Methods | Outliers '0' | | | Outliers '1' | | | Outliers '2' | | | Outliers '3' | | | Outliers '4' | | |
|-----------------------|--------------|-------------|-------------|--------------|-------------|-------------|--------------|-------------|-------------|--------------|-------------|-------------|--------------|-------------|-------------|
| | ρ_c | ρ_f | AUC | ρ_c | ρ_f | AUC | ρ_c | ρ_f | AUC | ρ_c | ρ_f | AUC | ρ_c | ρ_f | AUC |
| FUNTA (baseline) | 0.49 | 0.22 | 0.60 | 0.01 | 0.21 | 0.51 | 0.22 | 0.19 | 0.58 | 0.23 | 0.20 | 0.52 | 0.23 | 0.21 | 0.53 |
| Dir.out (baseline) | 0.72 | 0.01 | 0.82 | 0.24 | 0.02 | 0.52 | 0.75 | 0.42 | 0.60 | 0.00 | 0.02 | 0.55 | 0.00 | 0.02 | 0.58 |
| iFor(V_{map}) | 0.78 | 0.05 | 0.87 | 0.44 | 0.38 | 0.79 | 0.86 | 0.15 | 0.63 | 0.61 | 0.45 | 0.66 | 0.74 | 0.09 | 0.77 |
| iFor($Curv_{map}$) | 0.82 | 0.12 | 0.92 | 0.43 | 0.60 | 0.61 | 0.87 | 0.47 | 0.57 | 0.57 | 0.38 | 0.69 | 0.81 | 0.33 | 0.63 |
| iFor(Len_{map}) | 0.63 | 0.26 | 0.59 | 0.46 | 0.56 | 0.64 | 0.59 | 0.12 | 0.65 | 0.29 | 0.23 | 0.64 | 0.78 | 0.45 | 0.56 |
| OCSVM(V_{map}) | 0.82 | 0.02 | 0.85 | 0.50 | 0.51 | 0.75 | 0.77 | 0.35 | 0.60 | 0.53 | 0.41 | 0.66 | 0.78 | 0.18 | 0.74 |
| OCSVM($Curv_{map}$) | 0.80 | 0.11 | 0.91 | 0.50 | 0.60 | 0.70 | 0.55 | 0.23 | 0.59 | 0.56 | 0.44 | 0.68 | 0.61 | 0.15 | 0.66 |
| OCSVM(Len_{map}) | 0.81 | 0.10 | 0.75 | 0.37 | 0.42 | 0.70 | 0.84 | 0.18 | 0.76 | 0.54 | 0.42 | 0.67 | 0.83 | 0.25 | 0.69 |
| Methods | Outliers '5' | | | Outliers '6' | | | Outliers '7' | | | Outliers '8' | | | Outliers '9' | | |
| | ρ_c | ρ_f | AUC | ρ_c | ρ_f | AUC | ρ_c | ρ_f | AUC | ρ_c | ρ_f | AUC | ρ_c | ρ_f | AUC |
| FUNTA (baseline) | 0.49 | 0.22 | 0.60 | 0.01 | 0.02 | 0.51 | 0.22 | 0.00 | 0.58 | 0.23 | 0.01 | 0.51 | 0.23 | 0.21 | 0.53 |
| Dir.out (baseline) | 0.43 | 0.34 | 0.59 | 0.43 | 0.17 | 0.52 | 0.43 | 0.16 | 0.65 | 0.43 | 0.17 | 0.60 | 0.43 | 0.34 | 0.61 |
| iFor(V_{map}) | 0.69 | 0.26 | 0.69 | 0.56 | 0.36 | 0.61 | 0.93 | 0.30 | 0.60 | 0.47 | 0.30 | 0.67 | 0.92 | 0.51 | 0.64 |
| iFor($Curv_{map}$) | 0.62 | 0.29 | 0.61 | 0.54 | 0.28 | 0.63 | 0.93 | 0.21 | 0.68 | 0.48 | 0.20 | 0.77 | 0.79 | 0.26 | 0.73 |
| iFor(Len_{map}) | 0.42 | 0.13 | 0.61 | 0.47 | 0.21 | 0.64 | 0.97 | 0.29 | 0.65 | 0.40 | 0.08 | 0.77 | 0.74 | 0.40 | 0.63 |
| OCSVM(V_{map}) | 0.59 | 0.04 | 0.73 | 0.55 | 0.38 | 0.56 | 0.87 | 0.22 | 0.60 | 0.58 | 0.45 | 0.63 | 0.70 | 0.25 | 0.70 |
| OCSVM($Curv_{map}$) | 0.58 | 0.18 | 0.64 | 0.61 | 0.40 | 0.61 | 0.86 | 0.19 | 0.62 | 0.56 | 0.44 | 0.66 | 0.62 | 0.14 | 0.72 |
| OCSVM(Len_{map}) | 0.67 | 0.30 | 0.62 | 0.62 | 0.47 | 0.57 | 0.79 | 0.13 | 0.61 | 0.51 | 0.24 | 0.60 | 0.88 | 0.46 | 0.67 |

1156 feature, Len_{map} is better than V_{map} and $Curv_{map}$ for detecting 61
 1157 persistent magnitude outliers (Model 1 and Model 3 columns 62
 1158 iFor(Len_{map}) row). This indicates that for detecting persistent 63
 1159 magnitude outliers, the proposed approach is more reliable 64
 1160 with Len_{map} than $Curv_{map}$ and V_{map} . 1165

For persistent shape outliers (Table 4, Model 5 column), the proposed method outperforms the baselines with iFor(Len_{map}). Furthermore, V_{map} yields results similar to those of $Dir.out$ in terms of ρ_c and AUC . Table 4 shows that the state-of-the-art FUNTA totally fails to capture shape outlyingness

Table 4
Results on the synthetic datasets.

Outlier detection results for the *synthetic data* generated by the five models (columns), as described in Section 5.2. We compared the proposed methods, $iFor(\cdot)$ and $OCSVM(\cdot)$, with the two baselines, $FUNTA$ and $Dir.out$, in terms of three performance measures (in sub-columns): correct detection rate (ρ_c), false detection rate (ρ_f), and AUC . For each model and each performance metric, we marked in bold the best results (i.e., highest ρ_c and AUC , and lowest ρ_f). $iFor$ with V_{map} and Len_{map} has a similar performance as that of the state-of-the-art methods for most of the generating models. For Model 5, $iFor(Len_{map})$ outperforms the baselines.

| Methods | MODEL 1 | | | MODEL 2 | | | MODEL 3 | | | MODEL 4 | | | MODEL 5 | | |
|----------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | ρ_c | ρ_f | AUC | ρ_c | ρ_f | AUC | ρ_c | ρ_f | AUC | ρ_c | ρ_f | AUC | ρ_c | ρ_f | AUC |
| $FUNTA$ (baseline) | 1.00 | 0.00 | 1.00 | 0.92 | 0.02 | 0.99 | 0.96 | 0.00 | 1.00 | 0.89 | 0.04 | 0.99 | 0.58 | 0.31 | 0.73 |
| $Dir.out$ (baseline) | 1.00 | 0.00 | 1.00 | 1.00 | 0.00 | 1.00 | 0.91 | 0.00 | 1.00 | 0.98 | 0.00 | 1.00 | 0.88 | 0.00 | 1.00 |
| $iFor(V_{map})$ | 0.99 | 0.00 | 1.00 | 0.91 | 0.02 | 1.00 | 0.69 | 0.25 | 0.82 | 0.77 | 0.16 | 0.92 | 0.83 | 0.13 | 0.94 |
| $iFor(Curv_{map})$ | 0.61 | 0.30 | 0.75 | 0.57 | 0.48 | 0.60 | 0.59 | 0.39 | 0.67 | 0.57 | 0.48 | 0.61 | 0.73 | 0.24 | 0.85 |
| $iFor(Len_{map})$ | 1.00 | 0.00 | 1.00 | 0.95 | 0.00 | 1.00 | 0.83 | 0.08 | 0.96 | 0.85 | 0.07 | 0.97 | 0.96 | 0.01 | 1.00 |
| $OCSVM(V_{map})$ | 0.79 | 0.22 | 0.87 | 0.82 | 0.19 | 0.91 | 0.68 | 0.35 | 0.74 | 0.65 | 0.14 | 0.84 | 0.42 | 0.14 | 0.77 |
| $OCSVM(Curv_{map})$ | 0.49 | 0.34 | 0.65 | 0.60 | 0.52 | 0.62 | 0.48 | 0.38 | 0.63 | 0.42 | 0.44 | 0.61 | 0.43 | 0.37 | 0.65 |
| $OCSVM(Len_{map})$ | 0.66 | 0.10 | 0.82 | 0.83 | 0.07 | 0.91 | 0.59 | 0.16 | 0.78 | 0.62 | 0.07 | 0.84 | 0.50 | 0.06 | 0.83 |

because it is based on the intersection angles between the samples and is computed for each variable separately. Thus it fails to consider the correlation between them (as explained in Section 1).

As V_{map} and Len_{map} achieve satisfactory results, the metric characterization (velocity and length) of the sample provides a different type of outlier detection. We note that functional-data approximation affects the geometric characterization. Indeed, functional approximation enables smoothing out a curve and properly extracting derivative-based features because the induced smoothing renders the sample differentiable (see Section 3.3); this is not a required property for the baselines $Dir.out$ and $FUNTA$. Here, we carefully monitor the functional-approximation step using leave-one-out cross-validation (Eq. 14). Thus, in contrast with the approximation step, the outlier-detection step depends greatly on the mapping-function computation.

We recommend using Len_{map} in the case of (potential) persistent magnitude or shape outliers. In practice, Len_{map} does not directly indicate whether a sample is a shape or magnitude outlier. However, as shape and magnitude are quite distinctive outlyingness classes, the class of such an outlier can be known *a posteriori* by visual inspection or by setting a magnitude threshold with respect to the magnitude of the outliers detected. If the outliers are suspected to be isolated, we recommend using V_{map} and $Curv_{map}$, as both mapping functions extract local curve features in \mathbb{R}^p . In the case of a low contamination level, both $OCSVM$ and $iFor$ are suitable (even though on the ECG data, $OCSVM$ is better for small c), whereas for high contamination levels, $iFor$ is better.

We demonstrated that each mapping function can detect multiple classes of outliers. However, identifying the class of an outlier detected by a given mapping function is not an easy task, and this issue will be addressed in future work.

5.6.4. Statistical assessment of the results

We followed the hypothesis-testing procedure recommended by Demsar [10] to compare the statistical significance

of the results obtained from all the methods tested on various datasets to assess statistical relevance. Demsar provided an evaluation protocol for a more general assessment of the difference between several classifiers used on multiple benchmark datasets. The protocol consists of two steps: First, a global significance test is conducted to determine whether there is a difference among the evaluated methods. If this is the case, the methods are pairwise compared to evaluate the gain of one over another.

We applied Demsar's protocol because the present detection task reduces to a two-class classification in the evaluation step (outliers/inliers). Erfani *et al.* [13] also used the same evaluation protocol to assess the statistical significance of several outlier detection methods. We applied the protocol for the three performance measures ρ_c , ρ_f , and AUC separately. As described in [10, 18], there are several ways of conducting the tests in the evaluation protocol, and we primarily applied it as Erfani *et al.* in [13]. Specifically, we applied the protocol as follows:

- (i) First, the Friedman test [39] was applied to detect the global statistical significance for each of the three performance measures among all the methods on all the datasets. The Friedman test can be viewed as the non-parametric version of ANOVA (where, here, a group refers to a method, and the samples in the group refer to the performance of the method on the datasets), as it is based on the ranks and thus does not make the Gaussian assumption for the performance measures for each method [10]. We conducted the Friedman test with the Iman-Davenport correction [39], as recommended in [10], to handle the well-known family-wise error rate, which can bias the p -value in a multiple-hypothesis test. We recall that in the present context, the family-wise error rate refers to the probability of erroneously asserting that one method is more reliable for detecting outliers than some of the others.
- (ii) Second, if statistical significance was detected by the Friedman test, we performed a post-hoc test to deter-

mine which methods are different. More precisely, the post-hoc test is based on the p -values returned by pairwise-comparison test applied to all pairwise comparisons of the methods. A nonparametric test can be selected for the pairwise comparisons (owing to the absence of the Gaussian assumption), such as the post-hoc Friedman's aligned ranked test [18]. As the all-pairwise-comparisons test is a special case of multiple-hypothesis test, it also suffers from the family error rate and requires a correction procedure. Thus we used the Finner correction as recommended in [18].

We separately applied this evaluation protocol to the three performance measures for the five contamination levels of the ECG data, the 10 outlier classes of the PenDig dataset, and the synthetic data to compare the methods on two distinct types of data and to demonstrate the benefit of the proposed approach on real data. Moreover, this enables assessing the difference of the methods in a given context (i.e., when the outlier class is known). For all datasets, we used a significance level of 0.1, as in [13].

We report the average ranking (vertical axis) of all methods (horizontal axis) applied to the ECG and PenDig datasets (resp., synthetic data) for each performance measure (colors) in a vertical-bar plot in Fig. 6 (resp., Fig. 7). Each bar has height equal to its average rank (1 is the best, 8 is the worst) based on the post-hoc Friedman's aligned rank test across the five contamination levels (resp., five models). For ρ_c and AUC , the ranking is given in decreasing order, and for ρ_f the ranking is given in increasing order. The above number of bars refers to the global ranking (i.e., ranks from the average ranks).

As the Friedman test yielded a significant result for the two real datasets and the synthetic data, for each performance measure (p -values are given in the discussion), we report the significance (based on the p -values) of all the pairwise comparison tests. The significance of the pairwise comparison tests of ρ_c , ρ_f , and AUC for the ECG and PenDig datasets is given in Tables 5, 6, and 7, and for the synthetic data, in Tables 8, 9, and 10. The significance (at level 0.1) of a test is indicated by \neq^* , and non-significance is indicated by $=$.

ECG data. The Friedman test rejects the null hypothesis of equivalence of the methods for the three performance measures at a significance level of 0.1. The p -values are 3.0×10^{-10} for the correction detection rate ρ_c , 3.0×10^{-10} for the false detection rate ρ_f , and 2.2×10^{-16} for AUC . Thus, we conducted a post-hoc test. Fig. 6 shows the average ranking of the methods based on the Friedman's aligned ranked test (from the best 1 to the worst 8). The p -value of each pairwise comparison in the post-hoc test is given in Tables 5, 6, and 7 for the correction detection rate, the false detection rate, and AUC , respectively, where a cell indicates whether the resulting p -value of the pairwise comparison test of the methods in the corresponding row and column is significant. The symbol $=$ indicates a p -value greater than the significance level of 0.1, allowing the acceptance of the null hypothesis of equivalence of the two methods; rejection is indicated by

\neq^* .

Based on the results in Fig. 6 and Tables 5, 6, and 7, it is seen that both V_{map} and $Curv_{map}$ outperform the baselines in terms of the three performance measures. We notice that $Dir.out$ is not significantly better than the methods with the worst performance (i.e., $iFor(Len_{map})$, $FUNTA$, and $OCSVM(Len_{map})$). $FUNTA$ is not significantly different from $iFor(Curv_{map})$ and $OCSVM(Curv_{map})$ (Tables 5 and 7, $FUNTA$ rows and columns). Thus, by considering the results on the ECG data (Table 2 and Fig. 5), which demonstrate that $FUNTA$ is almost as effective as $iFor(Curv_{map})$ and $OCSVM(Curv_{map})$ in terms of ρ_c when the contamination level is high ($c \geq 15\%$), this qualitative comparison is confirmed by the non-significance of the difference with $OCSVM(Curv_{map})$. However, in terms of ρ_f , $FUNTA$ is ineffective and is outperformed by $iFor(V_{map})$, $iFor(Curv_{map})$, $Dir.out$, and $OCSVM(Curv_{map})$ (Table 6). Even though Len_{map} yields the worst results among the three proposed mapping functions with both $iFor$ and $OCSVM$ (Table 2, Fig 6), it is not significantly different from $Dir.out$ (see $Dir.out$ columns and Len_{map} rows in Tables 5 and 7).

PenDig data. The Friedman test rejects the null hypothesis of equivalence of the methods for the three performance measures at a significance level of 0.1. The p -values are 1.5×10^{-1} for the correct detection rate, 2.8×10^{-9} for the false detection rate, and 1.1×10^{-4} for AUC . We note that there is consistency with respect to the ECG data except for the false detection rate ρ_f . Indeed, both V_{map} and $Curv_{map}$ outperform the baselines in terms of ρ_c and AUC (Tables 5 and 7). Moreover, among the three mapping functions, Len_{map} yields the worst results and is not different from $Dir.out$. However, there is an inconsistency regarding ρ_f in the PenDig data with respect to the ECG data (Fig. 6 and Table 6). Hence, as the proposed method is not ranked first in terms of the false detection rate, it may be claimed that it recognizes the outliers but tends to be excessively severe.

We note that this conclusion regarding the correct and false detection rates is drawn according to the adopted outlyingness thresholding rule, which can be modified, as discussed at the end of Section 5.3.3.

From the global ranking (Fig. 6) and the pairwise comparison tests, it may be concluded that the proposed method outperforms the baselines on both the ECG and PenDig datasets.

Synthetic data. Regarding the synthetic data, the Friedman test rejects the null hypothesis of equivalence of the methods for the three performances measures at a significance level of 0.1. The p -value is 2.4×10^{-10} for the correct detection rate, 2.4×10^{-10} for the false detection rate, and 1.0×10^{-6} for AUC . As the p -values are significantly low, we can conduct a post-hoc test to compare the methods pairwise and assess the gain of one over another. Fig. 7 shows the average ranking of the methods according to the post-hoc Friedman's aligned rank test.

The significance of each pairwise comparison (based on the p -value) in the post-hoc (Friedman's aligned rank) test is

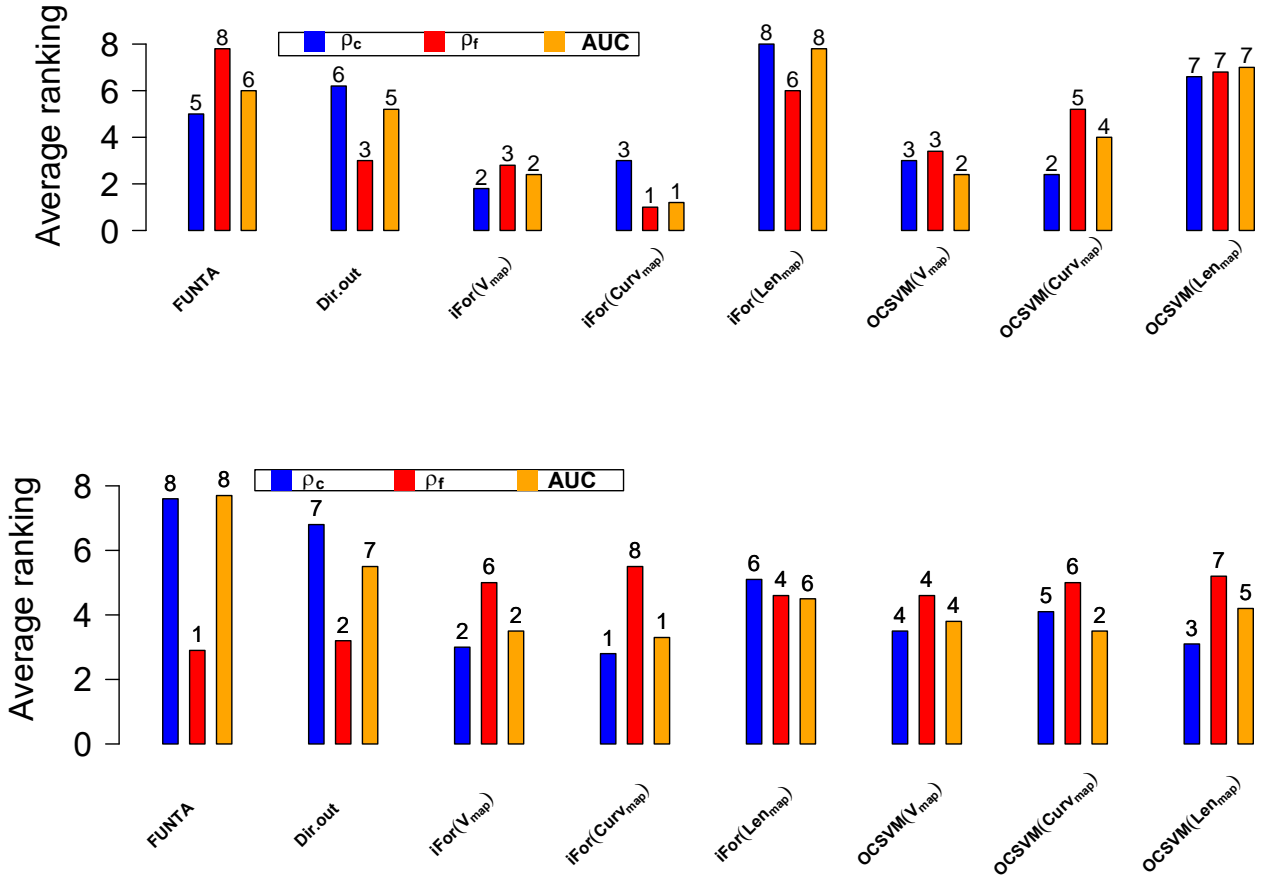


Figure 6: Ranking of the methods (1 is the best, 8 the worst) for ρ_c , ρ_f , and AUC based on the post-hoc Friedman's aligned rank test, considering the five contamination levels in the ECG data (upper bar plot) and the PenDig data (lower bar plot). For ρ_c and AUC, the ranking is given in decreasing order (i.e., for high ρ_c and AUC values, the rank tends to 1); for ρ_f , the ranking is given in increasing order (i.e., for low ρ_f values, the rank tends to 1). The y-axis represents the average ranking over the five models, and the integers on the top of the bars represent the final ranking. If there are ties, we take the average ranking.

1353 given in Tables 8, 9, and 10 for ρ_c , ρ_f , and AUC, respec370
 1354 tively. We notice that *Dir.out* is significantly equivalent to 6371
 1355 *iFor(Len_{map})*, *OCSVM(Len_{map})*, *FUNTA*, and *iFor(V_{map})*372
 1356 and these methods are ranked first, second, and third on aver373
 1357 age, respectively (Fig. 7). Thus, on the synthetic dataset, the374
 1358 baseline methods are slightly better than the proposed method375
 1359 however, based on the pairwise comparison tests, the best376
 1360 methods (*iFor(Len_{map})* and *OCSVM(Len_{map})*) are statisti377
 1361 cally equivalent. As discussed in the two previous paragraph378
 1362 the proposed method is superior on real datasets. Moreover379
 1363 in the iFor rows and OCSVM columns, it can be seen that380
 1364 there is a pairwise equivalence between iFor and OCSVM381
 1365 for (*Len_{map}*) and (*V_{map}*), that is, these two outlier-detecti382
 1366 algorithms are empirically consistent for a given mapping383
 1367 function. Therefore, we have equivalent methods to achiev384
 1368 state-of-the-art results (which cannot be improved, except fo385
 1369 MODEL 5) for the synthetic data. 1386

Overall assessment. Tables 5, 6, and 7 (in the iFor rows and OCSVM columns) show the pairwise consistency between the iFor and OCSVM algorithms for each mapping function. The same holds for the synthetic data. Thus, for a given dataset and mapping function, iFor and OCSVM achieve *statistically* the same performance results. This implies that the detection performance relies more on the outlying features provided by the mapping function than on the capacity of the outlier-detection algorithm to discover outlying features itself.

The main difference between the synthetic and the real data lies in the relationship between the variables, which is weak in the synthetic data (the correlation between the two variables is $\rho_{12} = 0.6$, Eq. (13)), whereas it is stronger in the real data. For Models 1–5, among the proposed mapping functions, *Len_{map}* achieves the best results and appears to be suitable for outlier detection if the variables are weakly correlated, whereas *V_{map}* and *Curv_{map}* are preferable if the correlation between the variables is strong. 1387
 1388

Table 5

Significance of the pairwise comparisons for the correct detection rate ρ_c on the ECG (upper table) and PenDig (lower table) datasets. The statistical significance of the difference of two given methods at level 0.1 is indicated by \neq^* , and by $=$ otherwise.

\neq^* indicates that the corresponding methods in the row and the column of the cell are significantly different at a level of 0.1, and $=$ indicates that they are not. The lower triangular part was replaced by dashes because it is equal to the upper part.

| | <i>FUNTA</i> | <i>Dir.out</i> | <i>iFor(V_{map})</i> | <i>iFor(Curv_{map})</i> | <i>iFor(Len_{map})</i> | <i>OCSVM(V_{map})</i> | <i>OCSVM(Curv_{map})</i> | <i>OCSVM(Len_{map})</i> |
|----------------------------------|--------------|----------------|------------------------------|---------------------------------|--------------------------------|-------------------------------|----------------------------------|---------------------------------|
| <i>FUNTA</i> (baseline) | x | = | = | = | \neq^* | = | = | = |
| <i>Dir.out</i> (baseline) | - | x | \neq^* | \neq^* | = | \neq^* | \neq^* | = |
| <i>iFor(V_{map})</i> | - | - | x | = | \neq^* | = | = | \neq^* |
| <i>iFor(Curv_{map})</i> | - | - | - | x | \neq^* | = | = | \neq^* |
| <i>iFor(Len_{map})</i> | - | - | - | - | x | \neq^* | \neq^* | = |
| <i>OCSVM(V_{map})</i> | - | - | - | - | - | x | = | \neq^* |
| <i>OCSVM(Curv_{map})</i> | - | - | - | - | - | - | x | \neq^* |
| <i>OCSVM(Len_{map})</i> | - | - | - | - | - | - | - | x |

| | <i>FUNTA</i> | <i>Dir.out</i> | <i>iFor(V_{map})</i> | <i>iFor(Curv_{map})</i> | <i>iFor(Len_{map})</i> | <i>OCSVM(V_{map})</i> | <i>OCSVM(Curv_{map})</i> | <i>OCSVM(Len_{map})</i> |
|----------------------------------|--------------|----------------|------------------------------|---------------------------------|--------------------------------|-------------------------------|----------------------------------|---------------------------------|
| <i>FUNTA</i> (baseline) | x | = | \neq^* | \neq^* | \neq^* | \neq^* | \neq^* | \neq^* |
| <i>Dir.out</i> (baseline) | - | x | \neq^* | \neq^* | = | \neq^* | \neq^* | \neq^* |
| <i>iFor(V_{map})</i> | - | - | x | = | \neq^* | = | = | = |
| <i>iFor(Curv_{map})</i> | - | - | - | x | = | = | = | = |
| <i>iFor(Len_{map})</i> | - | - | - | - | x | = | = | = |
| <i>OCSVM(V_{map})</i> | - | - | - | - | - | x | = | = |
| <i>OCSVM(Curv_{map})</i> | - | - | - | - | - | - | x | = |
| <i>OCSVM(Len_{map})</i> | - | - | - | - | - | - | - | x |

Table 6

Significance of the pairwise comparisons for the false detection rate ρ_f on the ECG (upper table) and PenDig (lower table) datasets.

Notation is the same as in Table 5.

| | <i>FUNTA</i> | <i>Dir.out</i> | <i>iFor(V_{map})</i> | <i>iFor(Curv_{map})</i> | <i>iFor(Len_{map})</i> | <i>OCSVM(V_{map})</i> | <i>OCSVM(Curv_{map})</i> | <i>OCSVM(Len_{map})</i> |
|----------------------------------|--------------|----------------|------------------------------|---------------------------------|--------------------------------|-------------------------------|----------------------------------|---------------------------------|
| <i>FUNTA</i> (baseline) | x | = | = | = | \neq^* | = | = | = |
| <i>Dir.out</i> (baseline) | - | x | \neq^* | \neq^* | = | \neq^* | \neq^* | = |
| <i>iFor(V_{map})</i> | - | - | x | = | \neq^* | = | = | \neq^* |
| <i>iFor(Curv_{map})</i> | - | - | - | x | \neq^* | = | = | \neq^* |
| <i>iFor(Len_{map})</i> | - | - | - | - | x | \neq^* | \neq^* | = |
| <i>OCSVM(V_{map})</i> | - | - | - | - | - | x | = | \neq^* |
| <i>OCSVM(Curv_{map})</i> | - | - | - | - | - | - | x | \neq^* |
| <i>OCSVM(Len_{map})</i> | - | - | - | - | - | - | - | x |

| | <i>FUNTA</i> | <i>Dir.out</i> | <i>iFor(V_{map})</i> | <i>iFor(Curv_{map})</i> | <i>iFor(Len_{map})</i> | <i>OCSVM(V_{map})</i> | <i>OCSVM(Curv_{map})</i> | <i>OCSVM(Len_{map})</i> |
|----------------------------------|--------------|----------------|------------------------------|---------------------------------|--------------------------------|-------------------------------|----------------------------------|---------------------------------|
| <i>FUNTA</i> (baseline) | x | = | \neq^* | \neq^* | \neq^* | \neq^* | \neq^* | \neq^* |
| <i>Dir.out</i> (baseline) | - | x | \neq^* | \neq^* | = | = | \neq^* | = |
| <i>iFor(V_{map})</i> | - | - | x | = | = | = | = | = |
| <i>iFor(Curv_{map})</i> | - | - | - | x | = | = | = | = |
| <i>iFor(Len_{map})</i> | - | - | - | - | x | = | = | = |
| <i>OCSVM(V_{map})</i> | - | - | - | - | - | x | = | = |
| <i>OCSVM(Curv_{map})</i> | - | - | - | - | - | - | x | = |
| <i>OCSVM(Len_{map})</i> | - | - | - | - | - | - | - | x |

6. Conclusion

In this paper, we proposed a method to improve the detection of different types of outliers in multivariate functional data, based on curve shape. We assumed that the original discrete curves can be well approximated by finite functional basis expansions, where the basis is specified. Based on the smooth reconstruction provided by the fitted basis expansion, we used the arc-length, velocity, and curvature mapping functions to capture latent shape features. Then, we detected the outliers from the mapped curves using outlier-detection algorithms.

Through an experimental study on real and synthetic datasets, we demonstrated that the proposed approach outperforms multivariate functional depth baselines on real data and

can perform similarly on synthetic data (except for persistent shape outliers, where the proposed method performs better). We demonstrated that, compared with the baselines, the proposed approach is robust to the variation of the contamination level. The results are consistent on both synthetic and real data.

We also discussed the ability of each of mapping function to capture outlying features depending on the type of the outliers to be detected. In future work, we will investigate more deeply the identifiability of the class(es) of outliers detected with respect to a given mapping function. Moreover, the used taxonomy [22] does not cover outliers that represent a mixture of multiple classes of outlyingness. Hence, a further step would be to identify both the outlyingness class(es) and the

Table 7

Significance of the pairwise comparisons for AUC on the ECG (upper table) and PenDig (lower table) datasets.

Notation is the same as in Table 5

| | $FUNTA$ | $Dir.out$ | $iFor(V_{map})$ | $iFor(Curv_{map})$ | $iFor(Len_{map})$ | $OCSVM(V_{map})$ | $OCSVM(Curv_{map})$ | $OCSVM(Len_{map})$ |
|----------------------|---------|-----------|-----------------|--------------------|-------------------|------------------|---------------------|--------------------|
| $FUNTA$ (baseline) | x | = | = | = | ≠* | = | = | = |
| $Dir.out$ (baseline) | - | x | ≠* | ≠* | = | ≠* | ≠* | = |
| $iFor(V_{map})$ | - | - | x | = | ≠* | = | = | ≠* |
| $iFor(Curv_{map})$ | - | - | - | x | ≠* | = | = | ≠* |
| $iFor(Len_{map})$ | - | - | - | - | x | ≠* | ≠* | = |
| $OCSVM(V_{map})$ | - | - | - | - | - | x | = | ≠* |
| $OCSVM(Curv_{map})$ | - | - | - | - | - | - | x | ≠* |
| $OCSVM(Len_{map})$ | - | - | - | - | - | - | - | x |

| | $FUNTA$ | $Dir.out$ | $iFor(V_{map})$ | $iFor(Curv_{map})$ | $iFor(Len_{map})$ | $OCSVM(V_{map})$ | $OCSVM(Curv_{map})$ | $OCSVM(Len_{map})$ |
|----------------------|---------|-----------|-----------------|--------------------|-------------------|------------------|---------------------|--------------------|
| $FUNTA$ (baseline) | x | = | ≠* | ≠* | ≠* | ≠* | ≠* | ≠* |
| $Dir.out$ (baseline) | - | x | ≠* | = | = | = | = | = |
| $iFor(V_{map})$ | - | - | x | = | = | = | = | = |
| $iFor(Curv_{map})$ | - | - | - | x | = | = | = | = |
| $iFor(Len_{map})$ | - | - | - | - | x | = | = | = |
| $OCSVM(V_{map})$ | - | - | - | - | - | x | = | = |
| $OCSVM(Curv_{map})$ | - | - | - | - | - | - | x | = |
| $OCSVM(Len_{map})$ | - | - | - | - | - | - | - | x |

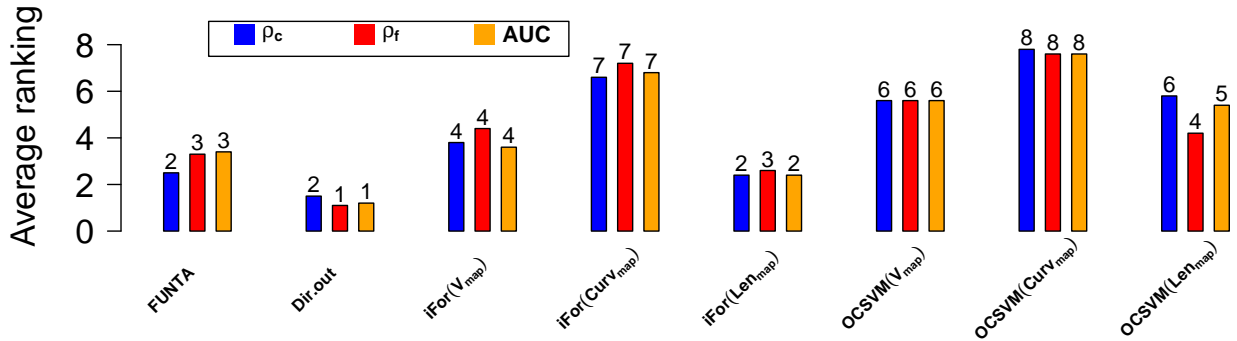


Figure 7: Ranking of the methods (1 is the best, 8 is the worst) on the synthetic datasets for ρ_c , ρ_f , and AUC based on the post-hoc Friedman's aligned rank test. For ρ_c and AUC , the ranking is given in decreasing order (i.e., for high ρ_c and AUC values, the rank is close to 1), and for ρ_f , the ranking is given in increasing order (i.e., for low ρ_f values, the rank is close to 1). The y-axis represents the average ranking over the five models, and the integers on the top of the bars represent the final ranking. If there are ties, we take the average ranking. The baseline methods are slightly better than the proposed method, but the best results by the proposed method ($iFor(Len_{map})$ and $OCSVM(Len_{map})$) are statistically equivalent to those by the baseline, as demonstrated by the pairwise comparison tests in Tables 8,9, and 10.

1417 potential mixture proportions when a sample lies in multiple 1428
 1418 classes.

1419 We did not assume any weighting of the curve variables 1429
 1420 in the mapping functions; this is left as future work. This 1430
 1421 weighting could be user-driven, as proposed for functional 1431
 1422 depth in [4], or data-driven. It is conceivable that this can en 1432
 1423 enhance outlier detection in the presence of non-outlying curve 1433
 1424 variables (when p increases). Another possible improvement 1434
 1425 would be to combine mapping functions in the same detect 1435
 1426 so that multiple outlier classes may be detected in the same 1436
 1427 dataset. 1437
 1438
 1439

Acknowledgment

We thank the French National Association for Research and Technology (ANRT) for providing us with a PhD grant.

References

- [1] Aggarwal, C.C., Yu, P.S., 2001. Outlier Detection for High-Dimensional Data, in: SIGMOD, ACM. pp. 37–46.
- [2] Arribas-Gil, A., Romo, J., 2014. Shape outlier detection and visualization for functional data: The outliergram. Biostatistics 15, 603–619. arXiv:1306.1718.
- [3] Bowman, F., 2012. Introduction to Bessel functions. Courier Corporation.
- [4] Claeskens, G., Hubert, M., Slaets, L., Vakili, K., 2014. MFHD:

Table 8Significance of the pairwise comparisons for the correct detection rate ρ_c on the synthetic dataset.

Notation is the same as in Table 5

| | <i>FUNTA</i> | <i>Dir.out</i> | <i>iFor(V_{map})</i> | <i>iFor(Curv_{map})</i> | <i>iFor(Len_{map})</i> | <i>OCSVM(V_{map})</i> | <i>OCSVM(Curv_{map})</i> | <i>OCSVM(Len_{map})</i> |
|----------------------------------|--------------|----------------|------------------------------|---------------------------------|--------------------------------|-------------------------------|----------------------------------|---------------------------------|
| <i>FUNTA</i> (baseline) | x | = | = | ≠* | = | = | ≠* | = |
| <i>Dir.out</i> (baseline) | - | x | = | ≠* | = | ≠* | ≠* | ≠* |
| <i>iFor(V_{map})</i> | - | - | x | = | = | = | ≠* | = |
| <i>iFor(Curv_{map})</i> | - | - | - | x | ≠* | = | = | = |
| <i>iFor(Len_{map})</i> | - | - | - | - | x | = | ≠* | ≠* |
| <i>OCSVM(V_{map})</i> | - | - | - | - | - | x | = | = |
| <i>OCSVM(Curv_{map})</i> | - | - | - | - | - | - | x | = |
| <i>OCSVM(Len_{map})</i> | - | - | - | - | - | - | - | x |

Table 9Significance of the pairwise comparisons for the correct detection rate ρ_f on the synthetic dataset.

Notation is the same as in Table 5

| | <i>FUNTA</i> | <i>Dir.out</i> | <i>iFor(V_{map})</i> | <i>iFor(Curv_{map})</i> | <i>iFor(Len_{map})</i> | <i>OCSVM(V_{map})</i> | <i>OCSVM(Curv_{map})</i> | <i>OCSVM(Len_{map})</i> |
|----------------------------------|--------------|----------------|------------------------------|---------------------------------|--------------------------------|-------------------------------|----------------------------------|---------------------------------|
| <i>FUNTA</i> (baseline) | x | = | = | ≠* | = | = | ≠* | = |
| <i>Dir.out</i> (baseline) | - | x | = | ≠* | = | ≠* | ≠* | = |
| <i>iFor(V_{map})</i> | - | - | x | = | = | = | ≠* | = |
| <i>iFor(Curv_{map})</i> | - | - | - | x | ≠* | = | = | = |
| <i>iFor(Len_{map})</i> | - | - | - | - | x | = | ≠* | = |
| <i>OCSVM(V_{map})</i> | - | - | - | - | - | x | = | = |
| <i>OCSVM(Curv_{map})</i> | - | - | - | - | - | - | x | ≠* |
| <i>OCSVM(Len_{map})</i> | - | - | - | - | - | - | - | x |

- 1440 Multivariate Functional Halfspace Depth. Journal of the American
1441 Statistical Association 109, 411–423. URL: <http://cran.r-project.org/package=MFHD>. 1465
- 1442 [5] Cléménçon, S., Thomas, A., 2017. Mass Volume Curves and Anomaly
1443 Ranking. Electronic Journal of Statistics doi:10.1214/18-EJS1474. 1467
- 1444 [6] Cuevas, A., Febrero, M., 2007. Robust estimation and classification
1445 for functional data via projection-based depth notions. Computational
1446 Statistics 22, 481–496. 1470
- 1447 [7] Cuevas, A., Febrero, M., Fraiman, R., 2006. On the use of the bootstrap
1448 for estimating functions with functional data. Computational Statistics
1449 and Data Analysis 51, 1063–1074. 1473
- 1450 [8] Dai, W., Genton, M.G., 2019. Directional outlyingness for multivariate
1451 functional data. Computational Statistics and Data Analysis 131, 50–
1452 65. 1476
- 1453 [9] De Boor, C., 1978. A practical guide to splines. volume 27. springer
1454 verlag New York. 1478
- 1455 [10] Demsar, J., 2006. Statistical Comparisons of Classifiers over Multiple
1456 Data Sets. Journal of Machine Learning Research 7, 1–30. 1480
- 1457 [11] Domingues, R., Filippone, M., Michiardi, P., Zouaoui, J., 2018. A
1458 comparative evaluation of outlier detection algorithms: Experimental
1459 and analyses. Pattern Recognition 74, 406–421. 1483
- 1460 [12] Dua, D., Graff, C., 2017. UCI machine learning repository. URL:
1461 <http://archive.ics.uci.edu/ml>. 1485
- [13] Erfani, S.M., Rajasegarar, S., Karunasekera, S., Leckie, C., 2016. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. Pattern Recognition 58, 121–134.
- [14] Febrero, M., Galeano, P., González-Manteiga, W., 2008. Outlier detection in functional data by depth measures, with application to identify abnormal NO_x levels. Environmetrics 19, 331–345.
- [15] Febrero-bande, M., Oviedo de la Fuente, M., 2012. Statistical Computing in Functional Data Analysis: The R package fda.usc. Journal of Statistical Software 51, 1–28.
- [16] Ferraty, F., Vieu, P., 2006. Nonparametric functional data analysis: theory and practice. Springer Science & Business Media.
- [17] Fraiman, R., Muniz, G., 2001. Trimmed means for functional data. Test 10.
- [18] García, S., Fernández, A., Luengo, J., Herrera, F., 2010. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180, 2044–2064.
- [19] Goldberger, A.L., Amaral, L.A., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K., Stanley, H.E., 2000. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. Circulation 101, e215–e220.
- [20] Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statis-

Table 10Significance of the pairwise comparisons for *AUC* on the synthetic dataset.

Notation is the same as in Table 5

| | <i>FUNTA</i> | <i>Dir.out</i> | <i>iFor(V_{map})</i> | <i>iFor(Curv_{map})</i> | <i>iFor(Len_{map})</i> | <i>OCSVM(V_{map})</i> | <i>OCSVM(Curv_{map})</i> | <i>OCSVM(Len_{map})</i> |
|----------------------------------|--------------|----------------|------------------------------|---------------------------------|--------------------------------|-------------------------------|----------------------------------|---------------------------------|
| <i>FUNTA</i> (baseline) | x | = | = | ≠* | = | = | ≠* | = |
| <i>Dir.out</i> (baseline) | - | x | = | ≠* | = | ≠* | ≠* | ≠* |
| <i>iFor(V_{map})</i> | - | - | x | ≠* | = | = | ≠* | = |
| <i>iFor(Curv_{map})</i> | - | - | - | x | ≠* | = | = | = |
| <i>iFor(Len_{map})</i> | - | - | - | - | x | ≠* | ≠* | ≠* |
| <i>OCSVM(V_{map})</i> | - | - | - | - | - | x | = | = |
| <i>OCSVM(Curv_{map})</i> | - | - | - | - | - | - | x | = |
| <i>OCSVM(Len_{map})</i> | - | - | - | - | - | - | - | x |

- tical Learning. 1554
- [21] Hoerl, A.E., Kennard, R.W., 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12, 55–67. 1555
- [22] Hubert, M., Rousseeuw, P.J., Segaert, P., 2015. Multivariate functional outlier detection. *Statistical Methods and Applications* 24, 177–202. 1558
- [23] Hyndman, R.J., Shang, H.L., 2010. Rainbow Plots , Bagplots , and Boxplots for Functional Data. *Journal of Computational and Graphical Statistics* 19, 29–45. 1561
- [24] Ieva, F., Paganoni, A.M., 2013. Depth Measures for Multivariate Functional Data. *Communications in Statistics - Theory and Methods* 42, 1265–1276. 1566
- [25] Japkowicz, N., Stephen, S., 2002. The class imbalance problem: A systematic study. *Intelligent data analysis* 6, 429–449. 1569
- [26] Kuhnt, S., Rehage, A., 2016. An angle-based multivariate functional pseudo-depth for shape outlier detection. *Journal of Multivariate Analysis* 146, 325–340. 1572
- [27] Liu, F.T., Ting, K.M., Zhou, Z.H., 2008. Isolation Forest, in: *ICDM*, pp. 413–422. 1575
- [28] López-pintado, S., Romo, J., 2009. On the Concept of Depth for Functional Data. *Journal of the American Statistical Association* 104, 718–734. 1578
- [29] López-pintado, S., Sun, Y., Lin, J.K., Genton, M.G., 2014. Simplicial band depth for multivariate functional data. *Advances in Data Analysis and Classification* 8, 321–338. 1581
- [30] Matérn, B., 2013. *Spatial variation*. volume 36. Springer Science & Business Media. 1584
- [31] Nason, G., 2008. *Wavelet Methods in Statistics with R*. Springer. 1587
- [32] Ramsay, J., Silverman, B.W., 2006. *Functional Data Analysis*. Wiley Online Library. 1590
- [33] Ramsay, J.O., Hooker, G., Graves, S., 2009. *Functional Data Analysis with R and MATLAB*. Springer Science & business Media. 1593
- [34] Rasmussen, C.E., 2003. Gaussian processes in machine learning, in: *Summer School on Machine Learning*, Springer. pp. 63–71. 1596
- [35] Rehage, A., 2016. *Functional Tangential Angle Pseudo-Depth*. URL: <https://cran.r-project.org/web/packages/FUNTA.r/package-version/0.1.0>. 1599
- [36] Ruff, L., Vandermeulen, R.A., Görnitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Uller, E., Kloft, M., 2018. Deep One-Class Classification, in: *ICML*, pp. 4390–4399. 1602
- [37] Schlather, M., Malinowski, A., Menck, P.J., Oesting, M., Storkorb, K., et al., 2015. Analysis, simulation and prediction of multivariate random fields with package randomfields. *Journal of Statistical Software* 63, 1–25. 1605
- [38] Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C., 2001. Estimating the support of a high-dimensional distribution. *Neural computation* 13, 1443–1471. 1608
- [39] Sheskin, D.J., 2003. *Handbook of parametric and nonparametric statistical procedures*. crc Press. 1611
- [40] Srivastava, A., Klassen, E.P., 2016. *Functional and Shape Data Analysis*. *Springer Series in Statistics*. URL: <http://link.springer.com/10.1007/978-1-4939-4020-2>. 1614
- [41] Stoer, J., Bulirsch, R., 2013. *Introduction to numerical analysis*. volume 12. Springer Science & Business Media. 1617
- [42] Sun, Y., Genton, M.G., 2011. Functional boxplots. *Journal of Computational and Graphical Statistics* 20, 316–334. 1620
- [43] Torres, J.M., Nieto, P.G., Alejano, L., Reyes, A., 2011. Detection of outliers in gas emissions from urban areas using functional data analysis. *Journal of Hazardous Materials* 186, 144 – 149. 1623
- [44] Tuddenham, R.D., Snyder, M.M., 1954. Physical growth of california boys and girls from birth to eighteen years. *Publications in child development*. University of California, Berkeley 1, 183–364. URL: <http://europepmc.org/abstract/MED/13217130>. 1626
- [45] Tukey, J.W., 1975. Mathematics and the picturing of data, in: *Proceedings of the International Congress of Mathematicians, Vancouver, 1975*, pp. 523–531. 1629
- [46] Wei, L., Keogh, E., 2006. Semi-supervised time series classification, in: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM. pp. 748–753. 1632
- [47] Xie, W., Chkrebti, O., Kurtek, S., Member, S., 2019. Visualization and Outlier Detection for Multivariate Elastic Curve Data. *IEEE Transactions on Visualization and Computer Graphics* . 1635
- [48] Xie, W., Kurtek, S., Bharath, K., Sun, Y., 2017. A geometric approach to visualization of variability in functional data. *Journal of the American Statistical Association* 112, 979–993. 1638
- [49] Zuo, Y., Serfling, R., 2000. General Notions of Statistical Depth Function. *The Annals of Statistics* 28, 461–482. 1641