Scalable Outlier Detection Methods for Functional Data

by

Oluwasegun Taiwo Ojo

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in

Mathematical Engineering

Universidad Carlos III de Madrid

Advisors:
Antonio Fernández Anta
Rosa E. Lillo

October 11, 2022

This work has been supported by:

institute IMDEA networks

To the Lamb who sits on the throne.

Acknowledgements

I am grateful to God for his grace and providence over the course of my research.

I am indebted to my advisors, Antonio Fernández Anta and Rosa Lillo Rodríguez for believing in me and giving me the opportunity to do my doctoral research under their supervision. I am grateful for their guidance, willingness to share their knowledge, encouragement, and their help in various forms all through the period of my research and preparation of this thesis. I am especially thankful to Antonio Fernández Anta for his tremendous support in helping me settle down in my early days in Spain and for his amazing mentorship in making me a researcher.

I am also grateful to Prof. Marc G. Genton for hosting me at KAUST, Saudi Arabia, for a fruitful three months research stay. I am grateful for his valuable insights, ideas, and feedback over the course of our collaboration.

I want to thank IMDEA Networks Institute for funding my research and providing a conducive research environment. I appreciate the admin staff at IMDEA Networks Institute for their support on various administrative, financial and immigration matters. I should also thank my fellow colleagues and the professors at IMDEA Networks for the stimulating discussions and ideas we shared over lunch and countless coffee breaks.

My sincere gratitude goes to my family at Immanuel Baptist Church, Madrid, for their incredible support over the course of my research. Although I cannot list them all, I am especially grateful to Sorin and Paula, Pat and Lami, Emi and Adela, Mrs. Injae Lee, Moises, Eric, Charles, Elizabeth, Keith, and Mixy for the love and care shown to me, and for making me feel at home in Madrid.

Special thanks to Prof. Femi Barnabas Adebola, Dr. Olusola Makinde, and Dr. Olusegun Eweemoje for their mentorship and support back at the Federal University of Technology, Akure. I am also grateful for the immense support I have received from my friends: Adeyeye Ebenezer, Soetan Ayodele, Margaret Ikape, and Antonio Elias Fernández.

Finally, thanks to my family: Abimbola, Akinwale, and Mrs. Toyin Ojo for their encouragement, and to my fiancée, Yufei Deng, for always cheering me on.

Published, Submitted, and Preprint Contents

The following lists include the research papers I have co-authored during the course of my PhD that are included in this thesis.

Published Contents and Software

- Ojo, O. T., Fernández Anta, A. F., Lillo, R. E., and Sguera, C. (2021), "Detecting and Classifying Outliers in Big Functional Data," *Advances in Data Analysis and Classification*, 2021, 1–36.
  - `https://doi.org/10.1007/s11634-021-00460-9`.
  - `https://arxiv.org/abs/1912.07287`
  - First-author.
  - Fully included in Chapter 3.
  - Whenever material from this source is included in this thesis, it is singled out with typographic means and an explicit reference.
- Ojo, O. T., Lillo, R. E., & Fernández Anta, A. (2021). ***fdaoutlier****: Outlier Detection Tools for Functional Data Analysis*. R package version 0.2, 9000.
  - `https://cran.r-project.org/package=fdaoutlier`.
  - Software package
  - Presented in Chapter 2.
  - Whenever material from this source is included in this thesis, it is singled out with typographic means and an explicit reference.

Submitted Contents

- Ojo, O. T., Fernández Anta, A., Genton, M. G., and Lillo, R. E. (2022), "Multivariate Functional Outlier Detection with the FastMOUD Indices," *Under Review*.
  - `https://arxiv.org/abs/2207.12803`.
  - Also available at `https://e-archivo.uc3m.es/handle/10016/35665`.
  - First-author.
  - Fully included in Chapters 4 and 5.
  - Whenever material from this source is included in this thesis, it is singled out with typographic means and an explicit reference.

Preprints

- Ojo, O., Lillo, R. E., and Anta, A. F. (2021). "Outlier Detection for Functional Data with R Package fdaoutlier," *arXiv:2105.05213*.
  - `https://arxiv.org/abs/2105.05213`.
  - First-author.
  - Fully included in Chapter 2.
  - Whenever material from this source is included in this thesis, it is singled out with typographic means and an explicit reference.

Invited Talks and Conference Presentations

The following list includes the various conferences and workshops where I have presented some parts of the work in this thesis.

- "Improvements to the Massive Unsupervised Outlier Detection (MUOD) Algorithm," *III International Workshop on Advances in FDA (IWAFDA)*, May 23 - 24, 2019, Cantabria, Spain.
- "Detecting and Classifying Outliers in Big Functional Data", *VI Workshop on Complex Sociotechnical Systems*, February 13 - 14, 2020, Burgos, Spain.
- "FastMUOD indices: theory and applications for outlier detection in FDA," *XXXIX Congreso Nacional de Estadística e Investigación Operativa (SEIO2022)*, June 7 - 10, 2022, Granada, Spain.

- "FastMUOD Indices - Theory and applications for outlier detection in functional data analysis" *Jornadas de Concurrencia y Sistemas Distribuidos (JCSD)*, June 15 - 17, 2022, las Navas del Marqués, Ávila, Spain.

Other Research Merits

The following list includes other research papers I have co-authored during the course of my PhD that are not included in this thesis.

- Bamisile, O., Cai, D., Oluwasanmi, A., Ejiyi, C., Ukwuoma, C. C., **Ojo, O.**, ... & Huang, Q. (2022). "Comprehensive assessment, review, and comparison of AI models for solar irradiance prediction based on different time/estimation intervals". *Scientific Reports*, 12(1), 1-26.

- Bamisile, O., **Ojo, O.**, Yimen, N., Adun, H., Li, J., Obiora, S., & Huang, Q. (2021). "Comprehensive functional data analysis of China's dynamic energy security index". *Energy Reports*, 7, 6246-6259.

- Baquero, C., Casari, P., Fernandez Anta, A., García-García, A., Frey, D., Garcia-Agundez, A., ... & Sanchez, I. (2021). "The CoronaSurveys system for COVID-19 incidence data collection and processing". *Frontiers in Computer Science*, 3, 641237.

- Garcia-Agundez, A., **Ojo, O.**, Hernández-Roig, H. A., Baquero, C., Frey, D., Georgiou, C., ... & Fernandez Anta, A. (2021). "Estimating the COVID-19 prevalence in Spain with indirect reporting via open surveys". *Frontiers in Public Health*, 9, 658544.

- Meuser, T., **Ojo, O. T.**, Bischoff, D., Fernández Anta, A., Stavrakakis, I., & Steinmetz, R. (2020, June). "Hide Me: Enabling Location Privacy in Heterogeneous Vehicular Networks". In *International Conference on Networked Systems* (pp. 11-27). Springer, Cham.

Abstract

Recent technological advances have led to an exponential growth in the volume of data generated. The quest to make sense of these data, some of which are usually complex, has led to recent interest in development of statistical methods for analysing data with complex structures. One such field of interest is functional data analysis (FDA), which deals with the analysis of data that can be considered as functions, curves, or surfaces observed over a domain set. Outlier detection is a challenging but important part of the exploratory analysis process in FDA because functional observations can exhibit outlyingness in various ways compared to the bulk of the data. This thesis addresses the problem of detecting and classifying outliers in functional data with three main contributions.

First, the **fdaoutlier** R package is presented in Chapter 2. The package contains implementations of some of the state-of-the-art functional outlier detection methods in the literature. Some of the methods implemented include directional outlyingness, magnitude-shape plot, sequential transformations, total variation depth, and modified shape similarity index. Detailed illustrations of the functions of the package are provided, using various simulated and real functional datasets curated from the functional outlier detection literature. Overviews of the functional outlier detection methods implemented in the package are also presented in Chapter 2. This chapter therefore, serves as a review of some of the current literature in outlier detection for functional data.

Next, two new methods, named 'Semifast- MUOD' and 'Fast-MUOD', are presented in Chapter 3. These methods work by computing for each curve three indices (magnitude, amplitude and shape index) that measure the outlyingness of that curve in terms of its magnitude, amplitude and shape. 'Semifast- MUOD' computes these indices with respect to (w.r.t.) a random sample of the dataset, while 'Fast-MUOD' computes these indices w.r.t. to the point-wise or L_1 median. The classical boxplot is then used as a cutoff on the three indices to identify curves that are outliers of different types. A by-product of the methods is an unsupervised classification of the outliers into different types, without the need for visualisation. Performance evaluation of the methods, using various real and simulated datasets, shows that Fast-MUOD is the better of the two

new proposed methods for outlier detection, in addition to being very scalable. Comparisons with latest functional outlier detection methods in the literature also show superior or comparable outlier detection performance.

In Chapter 4, some theoretical properties of the Fast-MUOD indices are presented. These include some definitions of the indices, as well as convergence proofs of the sample approximations. Some properties of the indices under simple transformations are also presented in this chapter. Finally, three techniques are presented in Chapter 5 for extending the Fast-MUOD indices to outlier detection in multivariate functional data observed on the same domain. These techniques include the use of random projections and identifying outliers on the marginal components of the multivariate functional data. The use of random projections showed the best result in performance evaluations with various real and simulated datasets.

Chapter 6 contains some concluding remarks and possible future research work.

Resumen

Los recientes avances tecnológicos han provocado un crecimiento exponencial del volumen de datos generados. La búsqueda de sentido a estos datos, algunos de los cuales suelen ser complejos, ha provocado un reciente interés por el desarrollo de métodos estadísticos para analizar datos con estructuras complejas. Uno de estos campos de interés es el análisis de datos funcionales (FDA), que se ocupa del análisis de datos que pueden considerarse como funciones, curvas o superficies observadas sobre un conjunto de dominios. La detección de valores atípicos es una parte desafiante pero importante del proceso de análisis exploratorio en el FDA, ya que las observaciones funcionales pueden presentar valores atípicos de diversas maneras en comparación con el grueso de los datos. Esta tesis aborda el problema de la detección y clasificación de valores atípicos en datos funcionales con tres contribuciones principales.

En primer lugar, el paquete R **fdaoutlier** se presenta en el capítulo 2. El paquete contiene implementaciones de algunos de los métodos de detección de valores atípicos funcionales más avanzados de la literatura. Algunos de los métodos implementados incluyen la perificidad direccional ('directional outlyingness'), el gráfico de magnitud-forma ('magnitude-shape plot'), las transformaciones secuenciales ('sequential transformations'), la profundidad de la variación total ('total variation depth') y el índice de similitud de forma modificado ('modified shape similarity index'). Se proporcionan ilustraciones detalladas de las funciones del paquete, utilizando varios conjuntos de datos funcionales simulados y reales curados de la literatura de detección de valores atípicos funcionales. En el capítulo 2 también se presenta un resumen de los métodos de detección de valores atípicos funcionales implementados en el paquete. Por lo tanto, este capítulo sirve como revisión de parte de la literatura actual sobre la detección de valores atípicos para datos funcionales.

A continuación, dos nuevos métodos, denominados 'Semifast- MUOD' y 'Fast-MUOD', se presentan en el capítulo 3. Estos métodos trabajan calculando para cada curva tres índices (magnitud, amplitud e índice de forma) que miden la perificidad de esa curva en términos de su magnitud, amplitud y forma. El método 'Semifast- MUOD' calcula estos índices con respecto a una muestra aleatoria del conjunto de datos, mientras que

'Fast-MUOD' calcula estos índices con respecto a la mediana puntual o L_1. Se utiliza el boxplot clásico como límite de los tres índices para identificar las curvas que son valores atípicos de diferentes tipos. Un subproducto de los métodos es una clasificación no supervisada de los valores atípicos en diferentes tipos, sin necesidad de visualizarlos. La evaluación del rendimiento de los métodos, utilizando varios conjuntos de datos reales y simulados, muestra que Fast-MUOD es el mejor para la detección de valores atípicos de los dos nuevos métodos propuestos, además de ser muy escalable. Las comparaciones con los últimos métodos funcionales de detección de valores atípicos de la literatura también muestran un rendimiento superior o comparable en la detección de valores atípicos.

En el capítulo 4, se presentan algunas propiedades teóricas de los índices Fast-MUOD. Éstas incluyen algunas definiciones de los índices, así como pruebas de convergencia de las aproximaciones muestrales. También se presentan en este capítulo algunas propiedades de los índices bajo transformaciones simples. Por último, en el capítulo 5 se presentan tres técnicas para ampliar los índices Fast-MUOD a la detección de valores atípicos en datos funcionales multivariantes observados en el mismo dominio. Estas técnicas incluyen el uso de proyecciones aleatorias y la identificación de valores atípicos en los componentes marginales de los datos funcionales multivariantes. El uso de proyecciones aleatorias mostró los mejores resultados en las evaluaciones de rendimiento con varios conjuntos de datos reales y simulados.

El capítulo 6 contiene algunas observaciones finales y posibles trabajos de investigación futuros.

Contents

List of Figures

List of Tables

Chapter 1

Introduction: Outlier Detection for Functional Data

1.1 Outlier detection

The problem of detecting and dealing with outliers is age old and common. Experimental scientists have always had to deal with observations that appear to be arbitrarily "different" and "unrepresentative" of the bulk of the data obtained from their experiments. Nevertheless, concretely defining what exactly is an outlier is challenging and subjective. Various attempts have been made in the literature to define outliers. For example, a common idea is that outliers are observations that either come from the extremes of the data distribution or from another data distribution entirely (Hawkins, 1980). On the other hand, Barnett and Lewis (1994) described outlier(s) as *"an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data"*. The phrase *"appears to be inconsistent"* echoes the subjective nature of declaring an observation as an outlier. Identifying such "different" observations is however crucial in exploratory analysis because outliers are known to bias the results of many statistical analysis procedures. Moreover, identifying outliers may reveal previously unknown behaviours about the data generating process. This has led to a recent interest in robust statistics and outlier detection. Robust statistics deals with development of statistical analysis methods that are resistant to outliers, while outlier detection deals with statistical methods for identifying outliers in a data.

However, it is often the case that a robust statistical method may help to identify outliers in a dataset. For example, when a random variable $\boldsymbol{X}$ generates independently and identically distributed values $\boldsymbol{x}_i \in \mathbb{R}^d, i = 1, \ldots n$, for $n, d \in \mathbb{N}$, it is common to use a robust minimum covariance determinant (MCD) estimate of the mean and covariance

matrix to find outliers. This is done by computing the robust Mahalanobis distance for each observation $\boldsymbol{x}_i$ and comparing this distance to the quantiles of the Chi-squared distribution. Standard (non-robust) estimates of the location and scatter are less effective in the described outlier detection procedure because they are biased by outliers which leads to a masking effect– the misidentification of outliers as non-outliers. In this example, the MCD estimates of the location and scatter are robust to outliers, thereby preventing the masking effect. Non-parametric robust statistical methods, usually based on ranks and data depths, are also useful for finding outliers, especially in multivariate observations. A statistical *depth* measure (Tukey, 1975) provides a centre-outward ordering of the multivariate observations. Observations with low depth values are then further scrutinized and maybe declared as outliers. The "non-parametric" nature of these methods is especially convenient as they do not require strict assumptions about the distribution of the data.

This thesis focuses on the problem of detecting outliers in functional data. Functional data are observations that are assumed to be realisations of a function or stochastic process defined over a domain. We will consider as outliers both (functional) observations that lie on the extreme of the distribution (of the data generating process) and observations that come from another distribution (or data generating process). We provide a brief overview of functional data analysis (FDA) in the next section.

1.2 Functional Data Analysis

Functional data analysis (Ramsay, 1982) deals with the analysis of observations that can be considered as realizations of functions defined over some domain set $\mathcal{I}$. In reality, these observations are observed at a finite resolution (or discrete points) over the domain, but it is natural to assume that these discrete measurements are evaluations of an underlying stochastic process $X : \mathcal{I} \mapsto \mathbb{R}$. The domain set $\mathcal{I}$ can be an interval $[a, b] \subset \mathbb{R}$, or more complex structures, e.g., a sphere. It is usual to assume that X is in the space of square-integrable functions over $\mathcal{I}$ ($L^2(\mathcal{I})$), with inner product defined as:

$$\langle f, g \rangle = \int_{\mathcal{I}} f(t)g(t)dt, \qquad f, g \in L^2(\mathcal{I}).$$

The norm induced by this inner product is given by

$$\|f\| = \sqrt{\langle f, f \rangle},$$

and the distance between any two functions $f, g \in L^2(\mathcal{I})$ is given by $\|f - g\|$. Sometimes, a vector in $\mathbb{R}^d,\ d \in \mathbb{N}$, is observed at each point of the domain, i.e., we have that

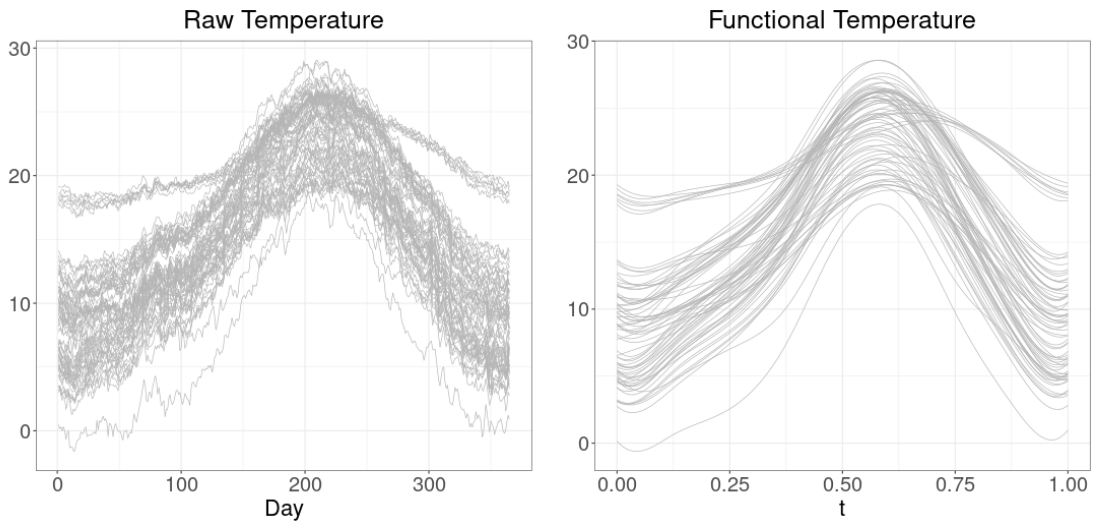

Figure 1.1: Daily average temperature in degree Celsius between 1980-2009 measured at 73 weather stations in Spain. Left: the raw daily average temperature values. Right: the smoothed daily average temperature values.

$\boldsymbol{X} : \mathcal{I} \mapsto \mathbb{R}^d$, in which case we have a multivariate functional data. Conceptually, a multivariate functional data can also be considered as a vector of stochastic processes $\boldsymbol{X} = [X^{(1)}, X^{(2)}, \ldots, X^{(p)}]^\top$ with each process having different domains and dimensions: $X^{(j)} : \mathcal{I}_j \mapsto \mathbb{R}^{d_j},\ d_j \in \mathbb{N},\ j = 1, \ldots, p$.

Figure 1.1 shows an example of a sample of functional observations. The left plot of Figure 1.1 shows the average daily temperature (in degree Celsius) between 1980-2000, observed at 73 weather stations in Spain (each observation or curve corresponds to a weather station). This data on the left plot of Figure 1.1 can be represented as

$$x_n(t_j) \in \mathbb{R},\ t_j \in [1, 365],\ n = 1, \ldots, 73,\ j = 1, \ldots, 365.$$

This means the 73 curves are measured only at finite specific points t_j, and their values at all points $t \in [1, 365]$ are unknown. However, it is natural to assume that the values of $x_n(t_j)$ come from curves $\{x_n(t)\}_{n=1}^{73}$, and that the values of these curves exist at any point $t \in [1, 365]$.

It is a usual first step in FDA to approximate the observed curves $x_n(t_j)$ as a linear combination of some standard basis functions:

$$x_n(t) \approx \sum_{m=1}^{M} c_{nm}\phi_m(t),$$

where $\phi_m(t)$ are some standard basis functions like splines and Fourier basis functions.

Typically, the number of basis functions M is smaller than $\#t_j$, so the approximation helps in dimension reduction by concisely representing each curve x_n by a dimension M vector of coefficients $[c_{n1}, c_{n2}, \ldots, c_{nM}]^\top$. Because the common basis functions used in FDA are smooth, the approximation also has a smoothing effect on the curves $x_n(t)$. This smoothing effect can be seen in the right plot of Figure 1.1 in which the raw temperature curves ($x_n(t_j)$) have been approximated with 11 B-Spline basis functions. A general overview of FDA can be found in Ramsay and Silverman (2006) and Kokoszka and Reimherr (2017), while Hsing and Eubank (2015) provides a review of the key theoretical foundations of functional data analysis.

In this thesis, we focus on outlier detection in functional data in which the domain set $\mathcal{I}$ is an interval $[a, b] \subset \mathbb{R}$. For multivariate functional data, we assume the component processes have the same domain and dimension, i.e., $\mathcal{I}_j = \mathcal{I}$ and $d_j = d \in \mathbb{N}$ for $j = 1, \ldots, p$.

1.3 Outlier Detection in Functional Data Analysis

A sample of functional data comprises curves or functions evaluated at a finite grid. Detecting outliers in such sample is challenging because functional observations can exhibit different outlying behaviours. For example, a curve may display magnitude outlyingness, in which case it is shifted above or below the bulk of the data. Also, a curve may have a different shape compared to the bulk of the data, without standing out at any point of the domain; such a curve is usually referred to as a shape outlier. Moreover, a curve may either be outlying throughout the domain or in a small part of the domain, with the former referred to as *persistent* outliers and the latter referred to as *isolated* outliers (Hubert et al., 2015). Figure 1.2 shows examples of different types of functional outliers.

Nevertheless, certain functional outlier detection methods in the literature are well suited to identifying a specific type of outlier; e.g., *outliergram* (Arribas-Gil and Romo, 2014) and *functional boxplot* (Sun and Genton, 2011) are suited to detecting shape and magnitude outliers, respectively. Consequently, it is of interest to develop a functional outlier detection method that is capable of detecting different types of outliers. The methods proposed in this thesis simultaneously target three different types of outliers viz. magnitude, shape, and amplitude outliers. The methods proposed also work quite well in identifying persistent and isolated outliers.

Apart from identifying an outlying curve, it is beneficial to know what type of outlier such a curve is. This helps to understand why such curve is flagged as an outlier, and enables selective targeting of different outlier types (for instance, an analyst may

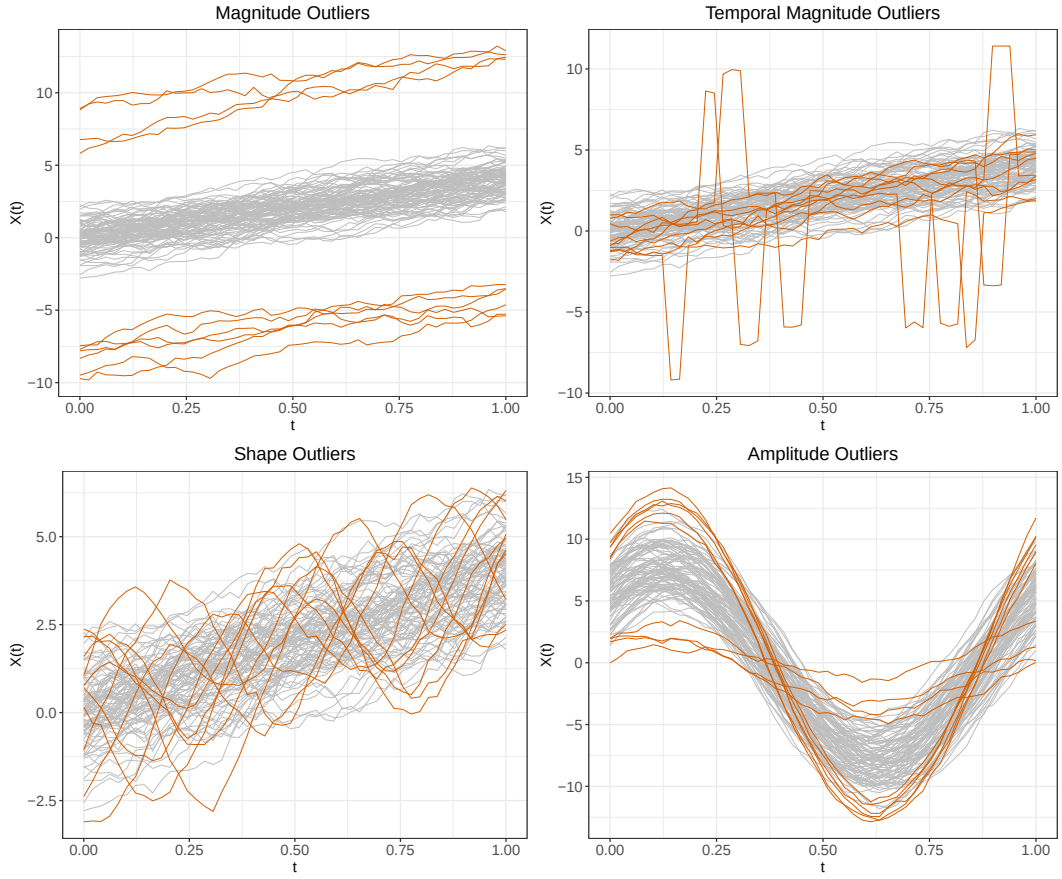

Figure 1.2: Examples of the different types of functional outliers. Curves in orange are outliers.

only be interested in magnitude outliers). While some current state-of-the-art outlier detection methods provide clues as to the type of an outlier, visualisation of some metrics of the data are often needed to get this information; which may be difficult when the data is large. Examples of these methods include the *magnitude-shape plot* (MS-plot) and *functional outlier map* (FOM), in which magnitude and shape outliers appear on the right (corner) and top (corner) of their plots, respectively (Dai and Genton, 2018; Rousseeuw et al., 2018). In addition to targeting different types of outliers, the outlier detection methods proposed in Chapters 3 and 5 also classify the identified outliers, unsupervised, without the need for visualization, making them valuable when the data is large.

Finally, the thesis addresses the problem of scalablity in functional outlier detection. Although some current methods in the literature target and classify different types of outliers unsupervised, e.g., Nagy et al. (2017), these methods are often computationally intensive and not scalable. Fast-MUOD, proposed in Chapter 3 can process over 1 mil-

lion observations in under 10 seconds, consequently lending itself to outlier detection in "big" functional data.

Therefore, the objective of the thesis is to add a scalable outlier detection method to the existing exploratory toolbox for functional data, that targets different types of outliers, while also classifying those outliers, unsupervised.

1.4 Outline of Thesis

The rest of the thesis starts with Chapter 2 which provides a review of some of the latest functional outlier detection methods in the literature and presents their implementation in the **fdaoutlier** R package. Detailed examples and scenarios on the use of the package are also presented in this chapter.

Chapter 3 presents the Semifast-MUOD and Fast-MUOD methods for detecting and classifying outliers in big functional data. These methods identify functional outliers by computing for each curve, a magnitude, an amplitude, and a shape index, that respectively target magnitude, amplitude, and shape outliers. A comprehensive simulation study was conducted to test the outlier detection performance and scalability of the proposed methods compared to other state-of-the-art functional outlier detection methods. Then, Fast-MUOD is illustrated with three real datasets consisting of weather data from Spain, population growth data, and a greyscale video data.

Chapter 4 explores some theoretical properties of the Fast-MUOD indices. The definitions of the indices, together with their corresponding sample and finite-dimensional approximations are presented. The properties presented in this chapter describe the behaviours of the Fast-MUOD indices under simple transformations and why these behaviours make the indices suitable for outlier detection in functional data.

Chapter 5 presents three techniques for using the Fast-MUOD indices for outlier detection in multivariate functional data. The three techniques include detecting outliers marginally, and the use of random projections. The techniques were tested on various simulated multivariate functional data sets with random projections showing effective outlier detection performance when compared with other multivariate functional outlier detection methods in the literature. Fast-MUOD with random projections is then illustrated on example multivariate functional datasets consisting of characters handwriting data and a color video data.

Some concluding remarks and outlook for future research are presented in Chapter 6. Apart from the results in the thesis, I worked on other topics during my PhD. These include monitoring the COVID-19 prevalence using surveys, energy security, predictive modelling of solar irradiance, and location privacy in vehicular networks.

Chapter 2

Outlier Detection Methods for Functional Data and R Package fdaoutlier

This chapter is reprint of:

Ojo, O., Lillo, R. E., & Fernández Anta, A. (2021). "Outlier Detection for Functional Data with R Package **fdaoutlier**". arXiv:2105.05213

The article presents the following software package:

Ojo, O. T., Lillo, R. E., & Fernández Anta, A. (2021). ***fdaoutlier**: Outlier Detection Tools for Functional Data Analysis*. R package version 0.2, 9000.

Abstract:

Outlier detection is one of the standard exploratory analysis tasks in functional data analysis. We present the R package **fdaoutlier** which contains implementations of some of the latest techniques for detecting functional outliers. The package makes it easy to detect different types of outliers (magnitude, shape, and amplitude) in functional data, and some of the implemented methods can be applied to both univariate and multivariate functional data. We illustrate the main functionality of the R package with common functional datasets in the literature.

2.1 Introduction

Outlier detection is a common task when carrying out exploratory data analysis. Identifying possible outliers is essential during the exploratory analysis process, because outliers can significantly bias statistical analyses. The process of dealing with identified outliers may also provide new insights into the nature of the data generating process. In functional data analysis (FDA), observations are treated as functions observed on a domain. These functional observations can exhibit various outlyingness properties as pointed out by Hubert et al. (2015). For instance, an observation can be shifted from the mass of the data. Such outliers are referred to as magnitude outliers in the FDA literature. On the other hand, an observation can be a shape outlier because it differs in shape from the mass of the data (even if it lies completely inside the mass of the data). For periodic functional observations, an observation may be outlying because it has an amplitude different from the mass of the data. Finally, any of the aforementioned outlyingness properties can be exhibited by a functional observation in a subset of the domain or all through the domain. Consequently, identifying outliers in FDA is challenging as there are many possible ways a functional observation can exhibit outlyingness.

Much work has been done regarding identifying outliers in the FDA context, with their corresponding software implementations made available in R (R Core Team, 2022). A number of these methods have been obvious applications of a notion of functional depths, which induces a centre outward ordering on a sample of curves. For instance, the functional boxplot (Sun and Genton, 2011) uses the (modified) band depth to define a 50% central region for the sample of curves with outliers identified as curves lying outside 1.5 times the central region in any part of the domain. In R, the functional boxplot is available in the **fda** package (Ramsay et al., 2022) with options to use the fast exact (modified) band depth defined by bands of two functions, proposed in Sun et al. (2012).

The **fda.usc** package (Febrero-Bande and de la Fuente, 2012) in R implements three functional outlier detection methods. The first method, proposed in Febrero et al. (2007), uses a likelihood ratio statistics to detect outlying curves (with cutoff determined by a bootstrap procedure). The other two methods identify outliers by comparing the depth values of the functions to a cutoff also obtained by a bootstrap procedure, based on either trimming of suspicious curves or placing more weights on the deeper curves (Febrero et al., 2008). These three methods are also implemented in the **rainbow** package (Han, 2011), together with the functional bagplot and the functional highest density region plot (Hyndman and Shang, 2010). The **rainbow** package also contains the integrated square forecast errors method for detecting functional outliers proposed in

Hyndman and Ullah (2007).

Nagy et al. (2017) proposed the j^{th} order integrated and infimal depths for identifying shape outliers, with implementations available in the **ddalpha** package (Pokotylo et al., 2019). Rousseeuw et al. (2018) in their work proposed a directional outlyingness (DO) measure, its functional extension (fDO), and the variability of directional outlyingness (vDO). Then, they used the functional outlier map, a scatter plot of the fDO versus vDO, to identify outliers with cutoffs determined by the standardized logarithm of the combined functional outlyingness(LCFO) measure. The functional outlier map can also be used with the adjusted outlyingness (AO) measure proposed in Brys et al. (2005) (see also Hubert and Van der Veeken, 2008, and Hubert et al. 2015), rather than the DO measure. These methods are available in the **mrfDepth** package (Segaert et al., 2020). Finally, the **roahd** package (Ieva et al., 2019) contains an implementation of the outliergram method proposed in Arribas-Gil and Romo (2014), as well as its multivariate generalisation proposed in Ieva and Paganoni (2020).

More recently proposed outlier detection methods include: the directional outlyingness for multivariate functional data proposed in Dai and Genton (2019) and further elaborated into the magnitude-shape plot (MS-plot) in Dai and Genton (2018); the total variation depth (TVD) and modified shape similarity index (MSS) proposed in Huang and Sun (2019); and the CRO-FADALARA method, based on archetypoids proposed in Vinue and Epifanio (2020b), and available in the **adamethods** package (Vinue and Epifanio, 2020a). Dai et al. (2020) also proposed detecting and classifying outliers using some sequence of transformations, e.g., shifting a curve to its centre and normalising it using the L_2 norm.

The objective of this paper is to describe the **fdaoutlier** package which aims to extend the available facility for outlier detection (in FDA context) for R, with implementations of some of the latest outlier detection methods. The **fdaoutlier** package's main contributions are:

- Implementations of the directional outlyingness and MS-plot outlier detection methods proposed in Dai and Genton (2019) and Dai and Genton (2018).

- An implementation of the TVD and MSS proposed in Huang and Sun (2019). The **fdaoutlier** implementation of TVD/MSS is written in C++ using R's `.Call` interface which leads to significant computational efficiency as TVD and MSS are computationally intensive.

- An implementation of the sequential tranformation method described in Dai et al. (2020).

- An implementation of the massive unsupervised outlier detection (MUOD) method proposed in Azcorra et al. (2018).

- Various depth and ordering methods, including extremal depth, one and two-sided extreme rank length depth, directional quantile, among others, useful for ordering functional observations (e.g., in functional boxplots).

In the next section, we provide a brief overview of the implemented outlier detection methods and demonstrate their implementations in **fdaoutlier** using simulated data. In Section 2.3, we apply **fdaoutlier** on two common datasets in the FDA outlier detection literature, replicating some of the analyses done in the literature. We then conclude in Section 2.4 with some remarks and a future outlook of **fdaoutlier**.

2.2 Outlier detection methods

We provide a brief primer of the implemented methods in the **fdaoutlier** package, and then describe their implementations. For illustrating the methods, we use the convenience functions `simulation_model1()` - `simulation_model9()` implemented in **fdaoutlier** to generate data with different types of outliers. These functions are useful for the rapid development and testing of new outlier detection methods and were curated from the functional outlier detection literature. Figure 2.1 shows plots of sample data generated by these nine models produced by calling `simulation_model*(plot = TRUE)`.

2.2.1 Directional outlyingness and MS-plot

The directional outlyingess for multivariate functional data proposed in Dai and Genton (2019) provides a way to measure not only the point-wise outlyingness of a functional observation but also the direction of outlyingness of that observation with respect to (w.r.t.) the rest of the data. Formally, let $\mathbf{Y} : I \to \mathbb{R}^d$ be a stochastic process in the space of real continuous functions $C(I, \mathbb{R}^d)$ defined on a compact interval I. Let the probability distribution of $\mathbf{Y}$ be $F_{\mathbf{Y}}$. At each evaluation point, $t \in I$, $\mathbf{Y}(t)$ is a d-variate vector with probability distribution $F_{\mathbf{Y}(t)}$. The directional outlyingness for multivariate data is defined as:

$$\mathbf{O}(\mathbf{Y}, F_{\mathbf{Y}}) = o(\mathbf{Y}, F_{\mathbf{Y}}) \cdot \mathbf{v},$$

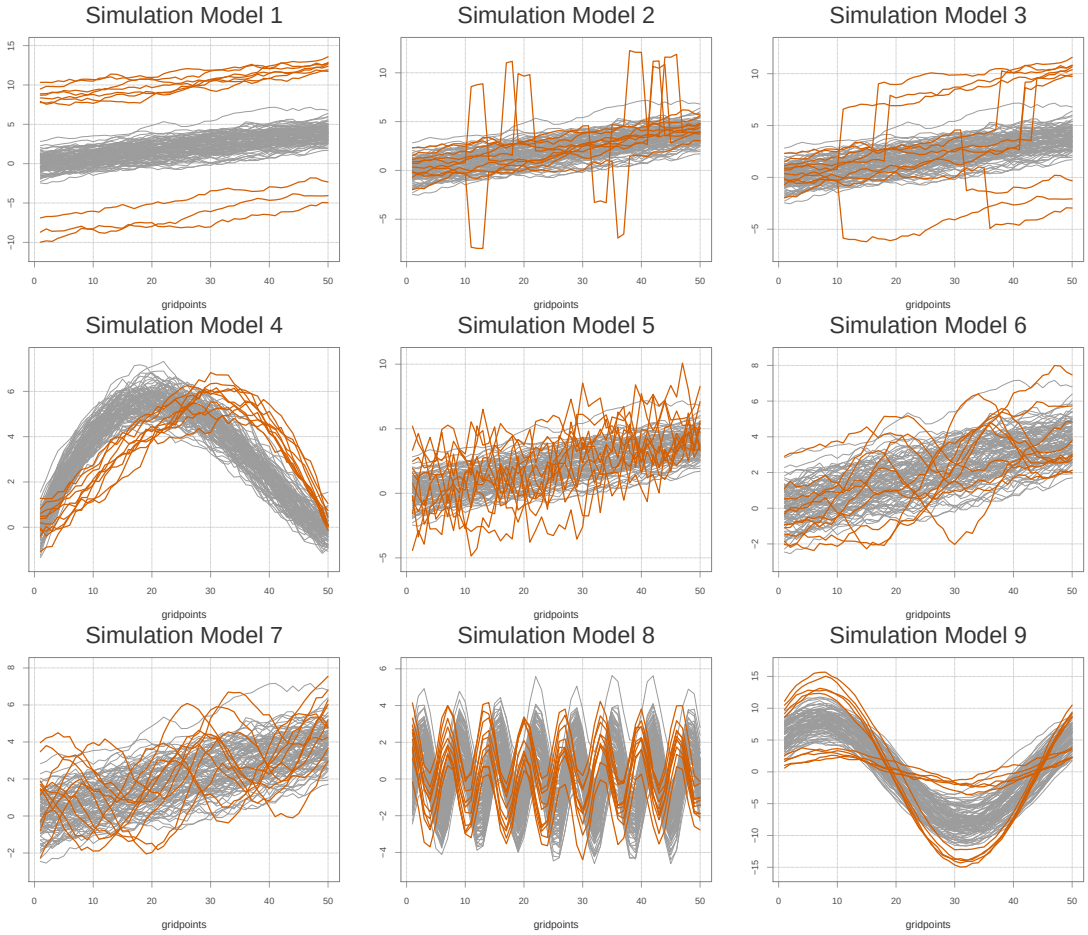

Figure 2.1: Simulation models: Plot of sample of data generated by each simulated model in **fdaoutlier**. Curves in orange are outliers.

where $o(\mathbf{Y}, F_{\mathbf{Y}})$ is the outlyingness of $\mathbf{Y}$ w.r.t. to $F_{\mathbf{Y}}$ and $\mathbf{v}$ is the spatial depth defined at point t by

$$\mathbf{v}(t) = \frac{|\mathbf{Y}(t) - \mathbf{Z}(t)|}{\|\mathbf{Y}(t) - \mathbf{Z}(t)\|},$$

with $\mathbf{Z}(t)$ being the unique median of $\mathbf{Y}(t)$ w.r.t. $F_{\mathbf{Y}(t)}$ (deepest point of $F_{\mathbf{Y}(t)}$). $\mathbf{v}(t)$ is a unit vector pointing from $\mathbf{Z}(t)$ to $\mathbf{Y}(t)$. Dai and Genton (2019) recommends using a distance-based outlyingness measure, like the Stahel-Donoho outlyingness defined by:

$$SDO(\mathbf{Y}(t), F_{\mathbf{Y}(t)}) = \sup_{\|\mathbf{u}\|=1} \frac{\|\mathbf{u}^\top \mathbf{Y}(t) - \text{median}(\mathbf{u}^\top \mathbf{Y}(t))\|}{\text{mad}(\mathbf{u}^\top \mathbf{Y}(t))}.$$

Thus, the Stahel-Donoho type directional outlyingness is given by:

$$\mathbf{O}(\mathbf{Y}, F_{\mathbf{Y}}) = SDO(\mathbf{Y}, F_{\mathbf{Y}}) \cdot \mathbf{v}.$$

Then the functional directional outlyingness (FO) is defined, to capture the *overall* outlyingness for functional data, as:

$$\text{FO}(\mathbf{Y}, F_{\mathbf{Y}}) = \int_I \|\mathbf{O}(\mathbf{Y}(t), F_{\mathbf{Y}(t)})\|^2 w(t) dt,$$

where $w(t)$ is a weight function defined on I. The mean directional outlyingness (**MO**) and variation of directional outlyingness (VO) were defined as :

$$\mathbf{MO}(\mathbf{Y}, F_{\mathbf{Y}}) = \int_I \mathbf{O}(\mathbf{Y}(t), F_{\mathbf{Y}(t)}) w(t) dt,$$

and

$$\text{VO}(\mathbf{Y}, F_{\mathbf{Y}}) = \int_I \|\mathbf{O}(\mathbf{Y}(t), F_{\mathbf{Y}(t)}) - \mathbf{MO}(\mathbf{Y}, F_{\mathbf{Y}})\|^2 w(t) dt.$$

These quantities measure the magnitude outlyingness and shape outlyingness of a functional observation, respectively. Dai and Genton (2019) further showed the relationship:

$$\text{FO}(\mathbf{Y}, F_{\mathbf{Y}}) = \|\mathbf{MO}(\mathbf{Y}, F_{\mathbf{Y}})\|^2 + \text{VO}(\mathbf{Y}, F_{\mathbf{Y}}),$$

which decomposes the total functional outlyingness into the magnitude outlyingness and the shape outlyingness.

In practice, the functional observations are observed at a finite number of points, say p, on I, i.e., at points $t_1, t_2, \ldots, t_p \in I$. The finite dimensional version of $\mathbf{MO}(\mathbf{Y}, F_{\mathbf{Y}})$ is defined as:

$$\mathbf{MO}_p(\mathbf{Y}, F_{\mathbf{Y}}) = \frac{1}{p} \sum_{i=1}^{p} \mathbf{O}(\mathbf{Y}(t_i), F_{\mathbf{Y}(t_i)}) w(t_i),$$

and the finite dimensional version of VO can be defined in a similar manner.

After obtaining the **MO** and VO for each curve, MS-plot is then a scatterplot of the points $(\mathbf{MO}^\top, \text{VO})^\top$. To detect outliers, a multivariate data whose columns are the **MO**s and VOs is formed, and a robust Mahalanobis distance is computed for each of the $(\mathbf{MO}^\top, \text{VO})^\top$ pair in this data. The robust covariance matrix is estimated using the minimum covariate determinant (MCD) estimator (Rousseeuw and Driessen, 1999). The distribution of these robust distances is approximated using the F distribution (Hardin and Rocke, 2005). Any observation with a robust distance greater than the cutoff obtained from the tails of the F distribution is flagged as an outlier.

The directional outlyingness and MS-plot methods procedures are implemented mainly through the `dir_out()` and `msplot()` functions in **fdaoutlier**. These functions accept a matrix or data frame of dimension $n \times p$ for a univariate functional data, or an array of dimension $n \times p \times d$ for multivariate functional data (where n is the num-

ber of functions/curves, p is the number of evaluation points in the domain, and d is the dimension of the functional data with, $d \geq 2$ for multivariate functional data). The `dir_out()` function computes the directional outlyingness matrix $\mathbf{O}(\mathbf{Y}, F_{\mathbf{Y}})$, the mean directional outlyingness $\mathbf{MO}(\mathbf{Y}, F_{\mathbf{Y}})$ and the variation of directional $\text{VO}(\mathbf{Y}, F_{\mathbf{Y}})$, while the `msplot()` function finds outliers using the mean and variation of outlyingness with the F approximation.

We illustrate identifying outliers with `msplot()` using `simulation_model5()` to generate data of 100 curves, out of which 10 are shape outliers with a different covariance structure. The generated curves are observed on 50 domain points over the interval $[0, 1]$. A call to `simulation_model5()` returns a list containing the matrix of generated data and a vector containing the indices of the true outliers.

```
R> simdata <- simulation_model5(n = 100, p = 50,
+                                outlier_rate = 0.1, seed = 2)
R> dt <- simdata$data
R> dim(dt)

[1] 100  50

R> simdata$true_outliers

 [1]  6 10 20 21 34 38 48 49 66 93
```

Next we pass the generated data to the `msplot()` function to detect the outliers in `dt`. By default, `msplot()` also produces a plot of the VO against the MO (or $||\mathbf{MO}||$ in the case of a multivariate functional data) and returns a list containing `outliers` which is a vector of the indices of detected outliers. The plotting function can be turned off by setting the parameter `plot = FALSE`.

```
R> ms <- msplot(dts = dt, return_mvdir = T, plot = FALSE)
R> ms$outliers

 [1]   6  10  20  21  34  38  48  49  51  66  93 100
```

Setting the additional parameter `return_mvdir = TRUE` ensures that vectors of the mean and variation of outlyingness (MO and VO) of each curve are returned by `msplot()` (a matrix is returned for **MO** in the case of a multivariate functional data).

```
R> head(ms$mean_outlyingness)

[1]  0.04718408 -0.76134612  1.30502807  0.20414153  0.81537363
[6]  2.27956644
```

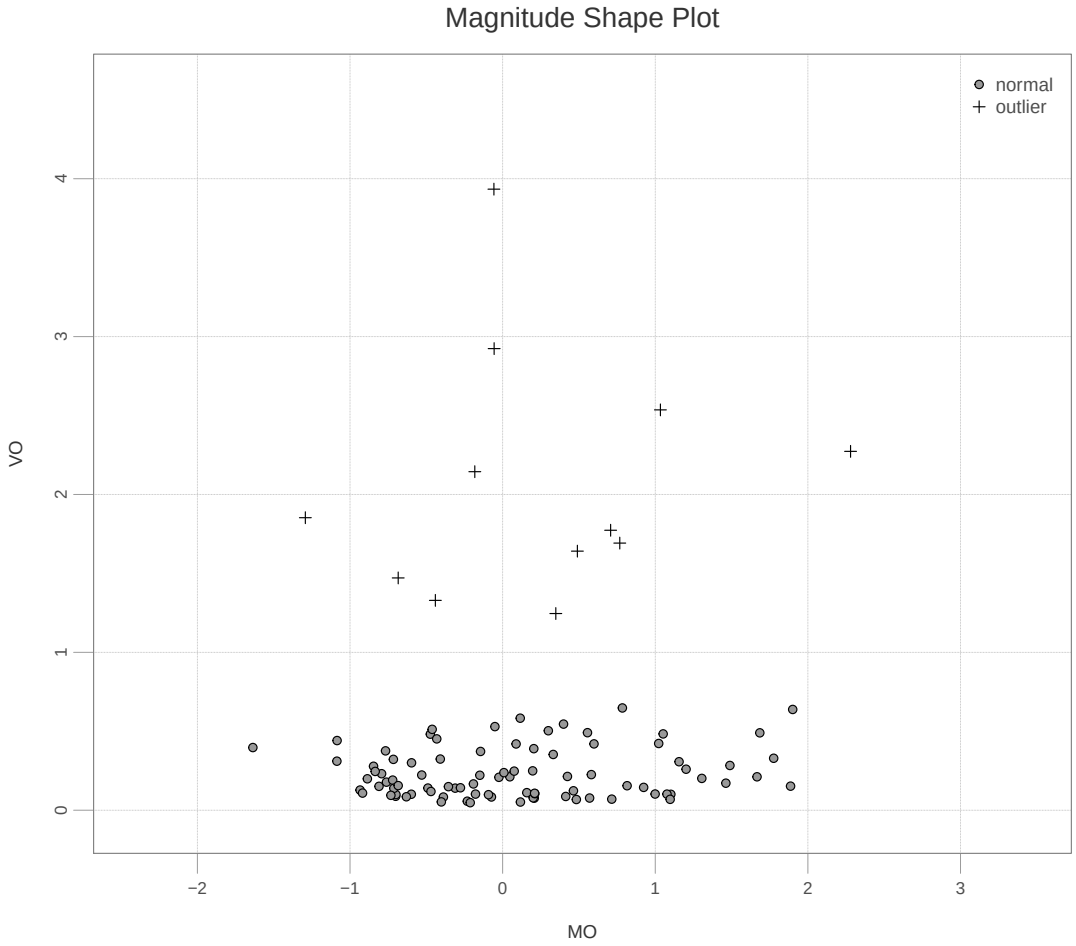

Figure 2.2: MS-Plot: Plot of the VO against the MO.

```
R> head(ms$var_outlyingness)
```

```
[1] 0.2115806 0.1777422 0.2015300 0.3898691 0.1550651 2.2730378
```

The MS-plot produced by the `msplot()` function when the parameter `plot = TRUE` is shown in Figure 2.2. Additional parameters `plot_title`, `title_cex`, `show_legend`, `ylabel` and `xlabel` can be passed to `msplot()` to further customise the MS-plot generated. If the aim is to compute either the MO and VO or the directional outlyingness matrix (or array for multivariate functional data), without the need for identifying outliers, then the `dir_out()` function, which is called by `msplot()` can be used directly. This returns a list containing the mean and variance of outlyingness, and the directional outlyingness matrix (if the parameter `return_dir_matrix = TRUE`).

```
R> simdir <- dir_out(dts = dt, return_dir_matrix = T)
R> head(simdir$mean_outlyingness)

[1]  0.04718408 -0.76134612  1.30502807  0.20414153  0.81537363
[6]  2.27956644

R> head(simdir$var_outlyingness)

[1] 0.2115806 0.1777422 0.2015300 0.3898691 0.1550651 2.2730378

R> dim(simdir$dirout_matrix)

[1] 100  50
```

2.2.2 Total variation depth and modified shape similarity index

Suppose $Y : I \to \mathbb{R}$ is a stochastic process defined on the interval I in $\mathbb{R}$. Let the distribution of Y be F_Y. For a function y, let $R_y(t)$ be the indicator function:

$$R_y(t) = \mathbb{1}\{Y(t) \leq y(t)\},$$

for $t \in I$. The functional total variation depth (Huang and Sun, 2019) of the function y w.r.t. F_Y is then defined as:

$$TVD(y, F_Y) = \int_I D_y(t)w(t)dt,$$

where $w(t)$ is a weight function and $D_y(t)$ is the pointwise total variation depth given by:

$$D_y(t) = \text{var}\{R_y(t)\} = \mathbb{P}\{Y(t) \leq y(t)\}\mathbb{P}\{Y(t) > y(t)\}.$$

The constant weight function $w(t) = \frac{1}{|I|}$ is suggested in (Huang and Sun, 2019) but other weight functions (that place more emphasis on different regions of the interval) can be used in the formulation of the functional total variation depth. The pointwise total variation depth $D_y(t)$ can be decomposed into a shape and magnitude component by breaking up the variance $\text{var}\{R_y(t)\}$ using the law of total variance:

$$D_y(t) = \text{var}\{R_y(t)\} = \text{var}[\mathbb{E}\{R_y(t)|R_y(s)\}] + \mathbb{E}[\text{var}\{R_y(t)|R_y(s)\}],$$

for $s, t \in I$ and $s = t - \Delta$. The shape similarity index of the functional observation y in a given time span Δ is then the weighted ratio of the shape component $\text{var}[\mathbb{E}\{R_y(t)|R_y(s)\}]$ to the total variation depth over the interval I:

$$SS(y, \Delta) = \int_I S_y(t, \Delta) v(t, \Delta) dt,$$

where

$$S_y(t, \Delta) = \begin{cases} \text{var}[\mathbb{E}\{R_y(t)|R_y(s)\}]/D_y(t) & D_y(t) \neq 0 \\ 1 & D_y(t) = 0, \end{cases}$$

and the weight function $v(t, \Delta)$ is the normalised changes in $y(t)$ over the interval I:

$$v(t, \Delta) = \frac{|y(t) - y(t - \Delta)|}{\int_I |y(t) - y(t - \Delta)|}.$$

The shape similarity index is a measure of shape outlyingness with small indices associated with shape outliers. However, when $D_y(t)$ is very small, the shape similarity index may not be small enough, so Huang and Sun (2019) further defined the modified shape similarity index (MSS) by shifting $(y(t - \Delta), y(t))$ to the centre. The modified shape similarity index is defined as:

$$MSS(y, \Delta) = \int_I S_{\tilde{y}}(t, \Delta) v(t, \Delta) dt,$$

where $\widetilde{y}$ is given by:

$$\widetilde{y}(s, \Delta) = \begin{cases} \text{median}\{Y(s)\} & s = t \\ f(s) - f(s + \Delta) + \text{median}[Y(s + \Delta)] & s = t - \Delta, \end{cases}$$

and

$$S_{\tilde{y}}(t, \Delta) = \text{var}(\mathbb{E}[R_{\tilde{y}}(t)|R_{\tilde{y}}(s)])/D_{\tilde{y}}(t)\}.$$

Details of the empirical versions of the total variation depth, the shape similarity index and its modified version are presented in the Appendix of Huang and Sun (2019). The total variation depth and the modified similarity index are implemented in the `total_variation_depth()` function of **fdaoutlier** using C++ through R's `.Call` interface for a fast and efficient computation. This function accepts only a matrix, calling it suffices to compute both the total variation depth and the modified shape similarity index, and it returns a list containing both the total variation depth and the modified shape similarity index:

```
R> tvdepth <- total_variation_depth(dt)
R> head(tvdepth$mss)

[1] 0.6388217 0.6208659 0.6975233 0.6853633 0.6618313 0.2167154

R> head(tvdepth$tvd)
```

```
[1] 0.224578 0.156484 0.114386 0.197932 0.178926 0.072398
```

In order to identify outliers, shape outliers are first identified and removed using a classical boxplot on the modified shape similarity indices. A functional boxplot is then used on the remaining curves (to identify magnitude outliers) using the total variation depth to identify their 50% central region (relative to the original number of curves). The `tvdmss()` function implements these steps to detect the magnitude and shape outliers. It returns a list containing the indices of the magnitude outliers, shape outliers, and the combined (shape and magnitude) outliers. This is illustrated using the generated data `dt` from the previous section.

```
R> tvoutlier <- tvdmss(dts = dt)
R> tvoutlier$shape_outliers

 [1]  6 10 20 21 34 38 48 49 66 93

R> tvoutlier$magnitude_outliers

NULL

R> tvoutlier$outliers

 [1]  6 10 20 21 34 38 48 49 66 93
```

In this case the total variation depth identifies all the shape outliers correctly and does not detect any magnitude outliers when compared to the index of the true outliers of the generated data:

```
R> simdata$true_outliers

 [1]  6 10 20 21 34 38 48 49 66 93
```

Additional arguments can be passed to the function parameters `emp_factor_mss`, `emp_factor_tvd`, and `central_region_tvd` of `tvdmss()` to control the classical boxplot of the modified shape similarity index and the functional boxplot of the total variation depth.

2.2.3 Outlier detection using sequential transformations

Dai et al. (2020) proposed using some sequence of transformations to identify and classify functional outliers. By transforming the functional data, it is possible to turn shape

outliers into magnitude outliers, consequently making it easier to identify shape outliers. More formally, let $\{Y_i\}_{i=1}^n$ be a set of functional observations in the space of continous functions $C(I)$ defined on an interval $I \in \mathbb{R}$. Suppose that $Y_i \sim F_Y$, and let Γ be a transformation that is also defined on $C(I)$. Furthermore, let $F_{\Gamma(Y)}$ be the distribution of the transformed data $\{\Gamma(Y_i)\}_{i=1}^n$. Dai et al. (2020) proposed the following algorithm for functional outlier detection and taxonomy.

Algorithm 1: Functional outlier detection using sequential transformations.

1 Identify from $\{Y_i\}_{i=1}^n$ the magnitude outliers using the functional boxplot, and denote the set of identified outliers by S_0. These are the Γ_0-outliers (magnitude outliers).
2 Apply transformation Γ_1 on $\{Y_i\}_{i=1}^n$ to get $\{\Gamma_1(Y_i)\}_{i=1}^n$.
3 Repeat step 1 on $\{\Gamma_1(Y_i)\}_{i=1}^n$ to obtain the set of detected outliers S_1; $S_1 \setminus S_0$ are the Γ_1-shape outliers.
4 Apply transformation Γ_2 on $\{\Gamma_1(Y_i)\}_{i=1}^n$ to get $\{\Gamma_2 \circ \Gamma_1(Y_i)\}_{i=1}^n$.
5 Repeat step 1 on $\{\Gamma_2 \circ \Gamma_1(Y_i)\}_{i=1}^n$ to obtain the set of detected outliers S_2; $S_2 \setminus S_1 \cup S_0$ are the $\Gamma_2 \circ \Gamma_1-$shape outliers.
6 Repeat steps 4 and 5 if more transformations are considered.

Dai et al. (2020) proposed the following useful (sequence of) transformations to identify and classify outliers:

Shifting and normalization of Curves: $\mathcal{T}_2 \circ \mathcal{T}_1 \circ \mathcal{T}_0(Y_i)$

This sequence involves first identifying the magnitude outliers using functional boxplot. This is the $\mathcal{T}_0$ transformation and the identified outliers are the $\mathcal{T}_0$ outliers (magnitude outliers). The second transformation $\mathcal{T}_1$ involves shifting the raw curves Y_i to their centres:

$$\mathcal{T}_1(Y)(t) = Y(t) - \lambda(I)^{-1} \int_I Y(t)dt,$$

where $\lambda(I)$ is the Lebesgue measure of the interval I. The $\mathcal{T}_1$ outliers are then identified using functional boxplot (step 3 of Algorithm 1). The third transformation $\mathcal{T}_2$ involves normalizing the centered curves, i.e., $\{\mathcal{T}_1(Y_i)\}_{i=1}^n$, with their L_2 norms:

$$\mathcal{T}_2(Y)(t) = \frac{\mathcal{T}_1(Y)(t)}{\left[\int_I \{\mathcal{T}_1(Y)(t)\}^2 dt\right]^{1/2}}.$$

Derivatives of curves: $\mathcal{D}_2 \circ \mathcal{D}_1 \circ \mathcal{D}_0(Y_i)$

- The $\mathcal{D}_0$ transformation first involves identifying the magnitude outliers using a functional boxplot without transforming the data (same as $\mathcal{T}_0$). These are the $\mathcal{D}_0$ outliers.

The second transformation $\mathcal{D}_1$ involves finding the derivative of the curves, and the third transformation $\mathcal{D}_2$ computes the derivative of $\mathcal{D}_1(Y_i)$ again. After each transformation, outliers are identified using functional boxplot as indicated in Algorithm 1. $\mathcal{D}_1$ and $\mathcal{D}_2$ transforms are implemented in **fdaoutlier** by differencing the observed points of the functions on the domain.

Directional outlyingness: $\mathcal{O}(\mathbf{Y}(t))$

For multivariate functional data $\{\mathbf{Y}_i\}_{i=1}^n$ taking values in $\mathbb{R}^d$, the directional outlyingness transformation $\mathcal{O}$ is especially useful. This transformation changes the multivariate functional observation $\mathbf{Y}_i$ to univariate functional data Y_i by finding the pointwise directional outlyingness described in Section 2.2.1 (Dai and Genton, 2019). The univariate functional data (of the outlyingness values) can then be investigated for outliers, e.g., using functional boxplot with a one-sided ordering like the (one-sided) extreme rank length depth (see Myllymäki et al., 2017, and Dai et al. 2020).

Other transformations and sequences suggested in Dai et al. (2020) include elimination of phase variations using a warping function:

$$\mathcal{R}(Y)(t) = Y(r(t)), \tag{2.1}$$

where $r(t)$ is a warping function on I. Eliminating phase variations using $\mathcal{R}(Y)$ may make it easier to detect shape outliers. Other possible sequences of transformations are: $\mathcal{D}_1 \circ \mathcal{T}_1 \circ \mathcal{T}_0(Y)$ and $\mathcal{D}_2 \circ \mathcal{D}_1 \circ \mathcal{T}_2 \circ \mathcal{T}_1 \circ \mathcal{T}_0(Y)$.

In the intermediate steps of identifying outliers using functional boxplots, possible depths and outlyingness measures that can be used to order the functions are: modified band depth (MBD) of López-Pintado and Romo (2009), j^{th} order integrated depth of Nagy et al. (2017) (FD_j), the L^∞ depth (Long and Huang, 2015), and extreme rank length depth (ERLD) of (Myllymäki et al., 2017). Other methods include the robust Mahalanobis distance (RMD) of the $(\mathbf{MO}^\top, \text{VO})^\top$ pair, obtained from the directional outlyingness in Section 2.2.1, and directional quantile (DQ) (Myllymäki et al., 2017). DQ, RMD, and L^∞ are distance-based, while MBD, FD_j, and ERLD are based on ranks. Dai et al. (2020) suggested using the distance-based methods, especially when the number of evaluation points on the interval I is small, as rank-based methods might suffer from a large number of ties. The distance-based methods also achieved the best results for detecting shape outliers in the simulation tests consisting of various shape outliers conducted in Dai et al. (2020). However, some transformations may require the use of specific ordering methods, e.g., the one-sided ERLD is best used with the $\mathcal{O}$ transformation since it generates univariate functional data made up of point-wise directional

outlyingness, and we want to consider only large values of these curves as extremes (rather than use a typical functional depth like MBD which considers both small and large values of curves as extremes). The **fdaoutlier** package implements all the transformations mentioned in Dai et al. (2020) except for the $\mathcal{R}(Y)(t)$ transformation which involves the use of a warping function. The ordering measures: band depth (BD) and MBD, L^∞, DQ, RMD, TVD, and ERLD (both one and two-sided) are available in **fdaoutlier** for ordering the functions in the intermediate functional boxplots.

The `seq_transform()` function in **fdaoutlier** finds outliers using sequential transformations. Like the other functions in **fdaoutlier**, `seq_transform()` accepts a matrix or data frame (of size n observations by p evaluation points) for a univariate functional data and an array (of size n observation by p evaluation points by d dimension). The sequence of transformations to apply on the data is specified to the `sequence` parameter which accepts a character vector containing a combination of the following strings: `"T0"`, `"D0"`, `"T1"`, `"T2"`, `"D1"`, `"D2"`, and `"O"`. The strings `"T0"`, `"T1"` and `"T2"` represent the tranformations $\mathcal{T}_0$, $\mathcal{T}_1$ and $\mathcal{T}_2$ respectively, while the strings `"D0"`, `"D1"`, and `"D2"` represent $\mathcal{D}_0$, $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively. The string `"O"` indicates the outlyingness transformation $\mathcal{O}(\mathbf{Y})(t)$. Thus, to specify the sequence of tranformations: $\mathcal{D}_1 \circ \mathcal{T}_1 \circ \mathcal{T}_0(Y)$, one should pass the argument `c("T0", "T1", "D1")` to the parameter `sequence` in the call to `seq_tranform()`, i.e., set `sequence = c("T0", "T1", "D1")`. We provide some examples below on the use of the `seq_transform()` function for detecting outliers using some suggested sequences in Dai et al. (2020). First we generate some data with outliers from `simulation_model4()`:

```
R> simdata4 <- simulation_model4(n = 100, p = 50,
+                                outlier_rate = 0.05,
+                                deterministic = T, seed = 50)
R> dt4 <- simdata4$data
```

Next, we call the `seq_transform()` function using the sequence $\mathcal{T}_2 \circ \mathcal{T}_1 \circ \mathcal{T}_0(Y)(t)$ while specifying MBD as the ordering function of choice for the intermediate functional boxplots.

```
R> seq1 <- seq_transform(dts = dt4,
+                        sequence = c("T0", "T1", "T2"),
+                        depth_method = "mbd")
R> seq1$outliers

$T0
integer(0)
```

```
$T1
[1] 43 53

$T2
[1] 43 53 96
```

`seq_transform()` returns a list of named lists, one of which is named `outliers`. The `outliers` list contains named vectors of the indices of the outliers found at each step of the sequence of transformations. The names of the vectors in `outliers` are the different transformations conducted at each step. In this example, the sequence $\mathcal{T}_2 \circ \mathcal{T}_1 \circ \mathcal{T}_0(Y)(t)$ (with modified band depth) identifies only three of the five true outliers contained in the simulated data:

```
R> unique(unlist(seq1$outliers))

[1] 43 53 96

R> simdata4$true_outliers

[1] 20 43 53 70 96
```

Next we try the sequence $\mathcal{D}_1 \circ \mathcal{T}_1 \circ \mathcal{T}_0(Y)(t)$ but now with the total variation depth in the intermediate functional boxplot:

```
R> seq2 <- seq_transform(dts = dt4,
+                        sequence = c("T0", "T1", "D1"),
+                        depth_method = "tvd")
R> seq2$outliers

$T0
integer(0)

$T1
[1] 43 53

$D1
integer(0)
```

The sequence $\mathcal{D}_1 \circ \mathcal{T}_1 \circ \mathcal{T}_0(Y)(t)$ with total variation depth identifies only two of the five true outliers as seen below:

```
R> unique(unlist(seq2$outliers))

[1] 43 53

R> simdata4$true_outliers

[1] 20 43 53 70 96
```

Another suggested sequence is the sequence $\mathcal{D}_2 \circ \mathcal{D}_1 \circ \mathcal{D}_0(Y)(t)$. We use this sequence but now with the L^∞ depth as the ordering function:

```
R> seq3 <- seq_transform(dts = dt4,
+                        sequence = c("D0", "D1", "D2"),
+                        depth_method = "linfinity")
R> seq3$outliers

$D0
integer(0)

$D1
integer(0)

$D2
integer(0)
```

This time, the sequence $\mathcal{D}_2 \circ \mathcal{D}_1 \circ \mathcal{D}_0(Y)(t)$ does not identify any of the outliers. This is not surprising as the sequence $\mathcal{D}_2 \circ \mathcal{D}_1 \circ \mathcal{D}_0(Y)(t)$ is advisable for identifying pure magnitude (captured by $\mathcal{D}_0$) and pure shape outliers (captured by $\mathcal{D}_1$ and $\mathcal{D}_2$) and this result is in line with the results of the simulation tests conducted in Dai et al. (2020) where the sequence $\mathcal{D}_2 \circ \mathcal{D}_1 \circ \mathcal{T}_0(Y)(t)$ performed the worst on this simulation model (See Table 4 of Dai et al. (2020)). Note that the sequence $\mathcal{D}_2 \circ \mathcal{D}_1 \circ \mathcal{D}_0(Y)(t)$ can also be specified with the `sequence` argument `c("D0", "D1", "D1")` or `c("D0", "D2", "D2")` or `c("T0", "D1", "D2")` since both `"D1"` and `"D2"` do the same thing, i.e., perform a lag-1 differencing and `"D0"` and `"T0"` also do the same thing (identify magnitude outliers in the raw untransformed data). When there are repeated transformations in the argument passed to `sequence` (e.g., when `sequence = c("D0", "D1", "D1")` is passed), a warning is shown and the labels of the output outliers list are changed, so that outliers for the two $\mathcal{D}_1$ transformations are accessed with `output$outliers$D1_1` and `output$outliers$D1_2` respectively:

```
R> seq4 <- seq_transform(dt = dt4,
+                        sequence = c("D0", "D1", "D1"),
+                        depth_method = "linfinity")
R> seq4$outliers

$D0
integer(0)

$D1_1
integer(0)

$D1_2
integer(0)
```

Sometimes, it may be necessay to inspect or save the intermediate transformed data for further analysis. Each intermediate transformed data can be saved by setting the parameter `save_data = TRUE` in the call to `seq_tranform()`. These data can then be assessed with the form `object_name$transformed_data$transform`:

```
R> seq5 <- seq_transform(dt = dt4,
+                        sequence = c("D0", "D1", "D1"),
+                        depth_method = "mbd", save_data = T)
R> str(seq5$transformed_data$D1_1)

 num [1:100, 1:49] 0.409 0.659 0.657 0.307 0.397 ...
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : chr [1:49] "2" "3" "4" "5" ...
```

As mentioned earlier, the $\mathcal{O}(\mathbf{Y})(t)$ transformation should be used with a one-sided ERLD ordering (in the functional boxplot) so that only large values of the resulting outlyingness data are considered as extremes:

```
R> seq6 <- seq_transform(dt = dt4,
+                        sequence = "O",
+                        depth_method = "erld",
+                        erld_type = "one_sided_right")
R> seq6$outliers

$O
[1] 18 43 53 70
```

The additional parameter `erld_type` specifies whether large values should be considered as extremes (`erld_type = "one_sided_right"`), or small values should be considered as extremes (`erld_type = "one_sided_left"`) or both small and large values should be considered as extremes (`erld_type = "two_sided"`). The two sided ordering is used by default if `erld_type` is not specified.

2.2.4 Massive unsupervised outlier detection

The massive unsupervised outlier detection (MUOD) detects and classifies outliers into shape, magnitude, and amplitude outliers. It was proposed in Azcorra et al. (2018) as a support method to identify influential users in social networks. MUOD works by computing for each curve three indices which measure outlyingness in terms of shape, magnitude, and amplitude. The shape index of Y_i w.r.t. F_Y denoted by $I_S(Y_i, F_Y)$ is defined as

$$I_S(Y_i, F_Y) = \left| \frac{1}{n} \sum_{j=1}^{n} \hat{\rho}(Y_i, Y_j) - 1 \right|,$$

where $\hat{\rho}(Y_i, Y_j)$ is the Pearson correlation coefficient between the observed points of curves Y_i and Y_j, given by

$$\hat{\rho}(Y_i, Y_j) = \frac{\text{cov}(Y_i, Y_j)}{s_{Y_i} s_{Y_j}}, \qquad s_{Y_i}, s_{Y_j} \neq 0.$$

The magnitude and amplitude indices of Y_i w.r.t. F_Y are defined are:

$$I_M(Y_i, F_Y) = \left| \frac{1}{n} \sum_{j=1}^{n} \hat{\alpha}_j \right|,$$

and

$$I_A(Y_i, F_Y) = \left| \frac{1}{n} \sum_{j=1}^{n} \hat{\beta}_j - 1 \right|,$$

respectively, where

$$\hat{\beta}_j = \frac{\text{cov}(Y_i, Y_j)}{s^2_{Y_j}}, \qquad s^2_{Y_j} \neq 0,$$

$$\hat{\alpha}_j = \bar{x_i} - \hat{\beta}_j \bar{x}_j,$$

and

$$\bar{x_i} = \frac{\sum_{t \in \mathcal{I}} Y_i(t)}{p}.$$

Generally, magnitude outliers will have larger magnitude indices, and the same applies to shape and amplitude outliers. To identify a cutoff for the indices, Azcorra et al. (2018) proposed to use a *"tangent"* method, which searches for the line tangent to the maximum index and uses as cutoff the point where this tangent line intercepts the x-axis. This method is especially problematic and prone to false positives, as pointed out by Vinue and Epifanio (2020b). A alternative is to use a classical boxplot on the indices to identify extremely large indices.

MUOD is implemented in **fdaoutlier** and can be accessed through the `muod()` function. The tangent method or the classical boxplot can be specified for determining the indices cutoffs. The function returns a list containing the outliers and the MUOD indices. The outliers list contains vectors with names `shape`, `amplitude` and `magnitude` all containing the indices of the detected shape, amplitude, and amplitude outliers, respectively.

```
R> simdata1 <- simulation_model1(n = 100, p = 50,
+                                outlier_rate = 0.1, seed = 2)
R> moutlier <- muod(dts = simdata1$data, cut_method = "tangent")
R> moutlier$outliers

$shape
[1] 100  58  35  79  21  40  50  14

$amplitude
[1]  51 100  58  14  94

$magnitude
 [1] 20 21 66 48 38 34 10 49  6 93 67

R> moutlier2 <- muod(dts = simdata1$data, cut_method = "boxplot")
R> moutlier2$outliers

$shape
[1] 100  58  35  79  21  40

$amplitude
[1]  51 100  58

$magnitude
 [1] 20 21 66 48 38 34 10 49  6 93
```

Furthermore, the muod magnitude (I_M), amplitude (I_A) and shape (I_S) indices can be accessed for further analysis:

```
R> str(moutlier$indices)
```

```
'data.frame':        100 obs. of  3 variables:
 $ shape    : num  0.1007 0.092 0.0845 0.1153 0.0721 ...
 $ magnitude: num  0.404 0.437 1.188 1.21 0.371 ...
 $ amplitude: num  0.08081 0.26492 0.00309 0.51515 0.11212 ...
```

2.3 Usage examples

In this section, we demonstrate the use of the **fdaoutlier** package on some common real datasets in FDA literature. In particular, we replicate some of the application examples from Dai and Genton (2018) and Dai et al. (2020). First, we analyse the Spanish ('aemet') weather dataset, followed by the population growth dataset. These datasets have been analysed extensively in the literature, and they provide meaningful applications for outlier detection in functional data analysis.

The Spanish weather data contains the daily average temperature, log precipitation, and wind speed of 73 Spanish weather stations measured between 1980-2009. This data was analysed in Dai and Genton (2018), Dai and Genton (2019), and Dai et al. (2020) where the directional outlyingness, MS-plot, and sequential transformation methods were proposed. The data is originally available in the **fda.usc** package (with the name `aemet`) but a stripped-down version is also made available in **fdaoutlier** (with the name `spanish_weather`). In this analysis, we focus on the average temperature and log precipitation, and the aim is to find outlying curves (weather stations with outlying weather data). Following Dai and Genton (2019), we smooth the data with 11 B-spline basis functions by obtaining a smoothing matrix (without roughness penalty) using the **fda.usc** package.

```
R> library("fda.usc")
R> data("spanish_weather")
R> b_spline <- create.bspline.basis(c(0, 365), nbasis = 11)
R> smoothing_matrix <- S.basis(tt = 0.5:364.5, basis =  b_spline)
R> temp <- spanish_weather$temperature %*% smoothing_matrix
R> logprecip <- spanish_weather$log_precipitation %*% smoothing_matrix
```

A plot of the original and smoothed versions of the temperature and log precipitation data is shown in Figure 2.3. Next, we apply MS-plot on the individual smoothed tem-

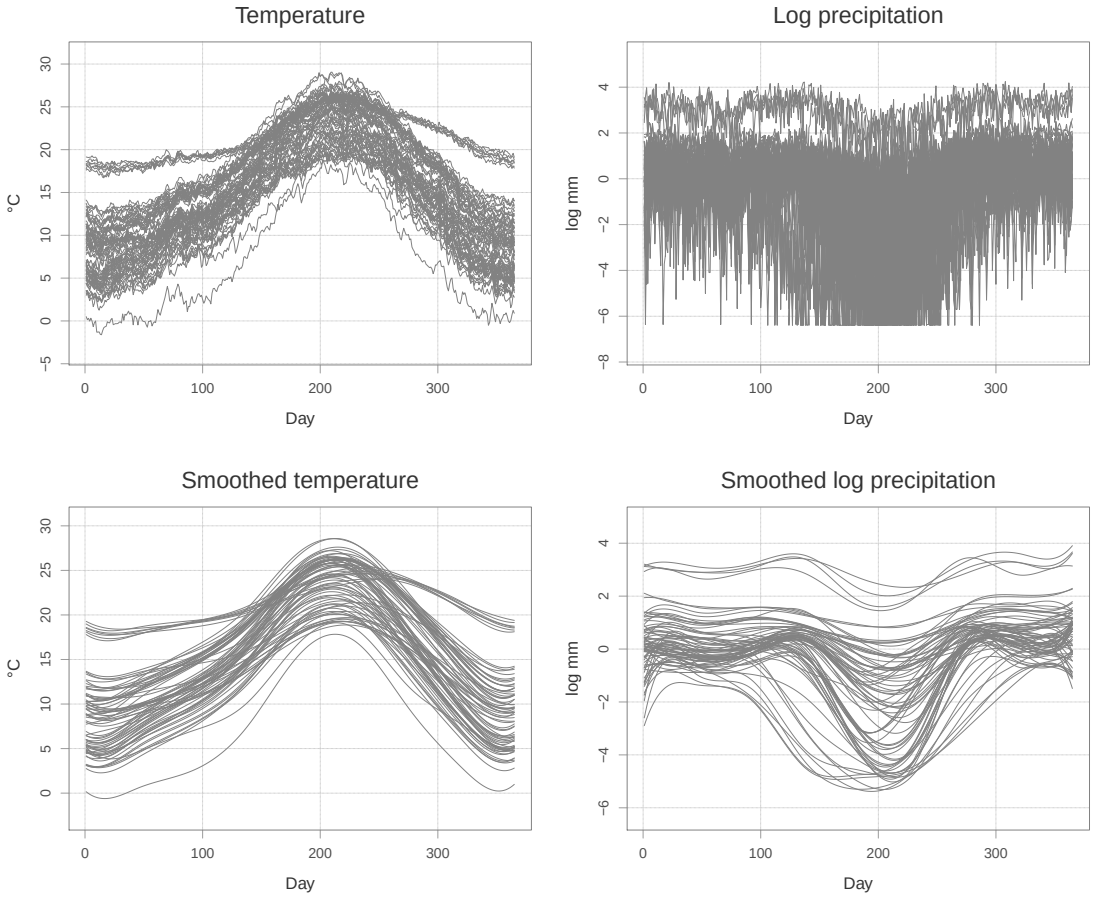

Figure 2.3: Plot of temperature and log precipitation and their smoothed (with 11 B-spline basis) versions.

perature and log precipitation data using the `msplot()` function to detect the marginal outliers.

```
R> temp_ms <- msplot(dts = temp, plot = F)
R> logprecip_ms <- msplot(dts = logprecip, plot = F)
```

Using the indices of the outliers returned, we can identify the weather stations detected as marginal outliers using the station information data (`station_info`).

```
R> head(spanish_weather$station_info$name[temp_ms$outliers])
```

```
[1] "A CORUÑA"
[2] "A CORUÑA/ALVEDRO"
[3] "SANTIAGO DE COMPOSTELA/LABACOLLA"
[4] "ASTURIAS/AVILÉS"
```

```
[5] "OVIEDO"
[6] "TARIFA"
```

```
R> head(spanish_weather$station_info$name[logprecip_ms$outliers])
```

```
[1] "LOGROÑO/AGONCILLO"
[2] "FUERTEVENTURA/AEROPUERTO"
[3] "LANZAROTE/AEROPUERTO"
[4] "LAS PALMAS DE GRAN CANARIA/GANDO"
[5] "COLMENAR VIEJO/FAMET"
[6] "MADRID/TORREJÓN"
```

Using the vectors of the mean and variation of directional outlyingness returned by `msplot()`, we can make a plot of the outliers detected and the plots of VO against MO as shown in Figure 2.4.

We can also apply `msplot()` on the multivariate functional data constructed by combining both the smoothed temperature and log precipitation using an `array` in order to detect and identify the joint temperature and log precipitation outliers:

```
R> joint_dt <- array(data = c(as.vector(temp),
+                     as.vector(logprecip)),
+                     dim = c(nrow(temp), ncol(temp), 2))
R> joint_ms <- msplot(joint_dt, plot = F)
R> joint_ms$outliers
```

```
 [1]  1  2  3  9 20 21 31 33 34 35 36 39 44 52 55 56 57 58 59 60 66 70
```

```
R> head(spanish_weather$station_info$name[joint_ms$outliers])
```

```
[1] "A CORUÑA"
[2] "A CORUÑA/ALVEDRO"
[3] "SANTIAGO DE COMPOSTELA/LABACOLLA"
[4] "ASTURIAS/AVILÉS"
[5] "TARIFA"
[6] "SANTANDER/PARAYAS"
```

Figure 2.5 shows the plot of the variation of outlyingness VO against the norm of the mean of outlyingness $\|\mathbf{MO}\|$ produced by **fdaoutlier** (by setting `plot = TRUE` in the call to `msplot()`).

Another option to detect joint outliers is to use the directional outlyingness transformation $\mathcal{O}(\mathbf{Y})(t)$ (together with a one-sided ERLD in the functional boxplot) in Dai et al. (2020). This can be achieved using the `seq_transform()` function in **fdaoutlier**.

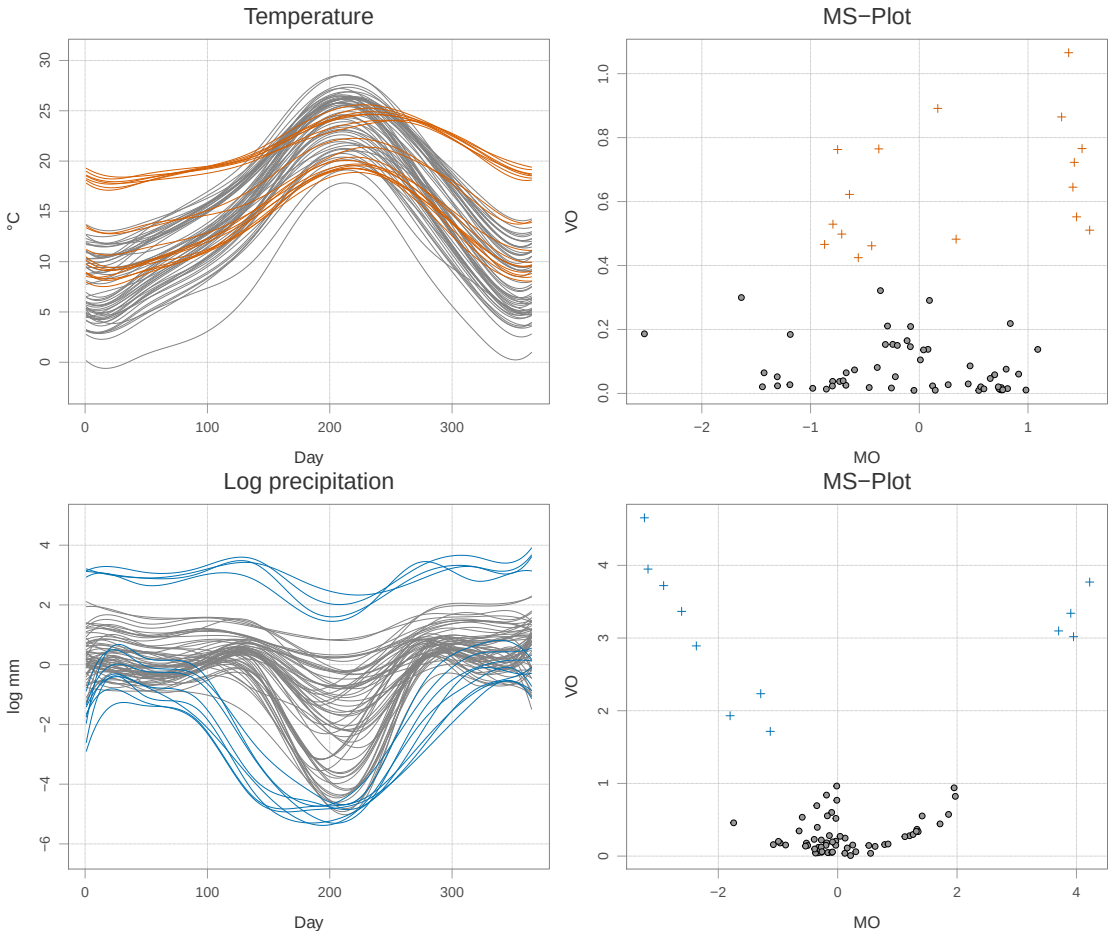

Figure 2.4: Plot of temperature and log precipitation and their MS-Plots. Lines and points in color are outliers.

```
R> joint_seq <- seq_transform(dts = joint_dt, sequence = "O",
+                 depth_method = "erld",
+                 erld_type = "one_sided_right")
R> joint_seq$outliers

$O
 [1] 33 34 35 36 39 44 45 55 56 57 58 60 66
```

As a second example, we consider the world population data analysis carried out in Dai et al. (2020). The data consists of the population of 105 countries as of July 1 between the years 1950-2010. These 105 countries have their populations between 1 million and 15 million on July 1, 1980. The preprocessed data is available in **fdaoutlier** under the name `world_population`. A plot of the data is shown in Figure 2.6. Using the `seq_transform()` function, we try to reproduce the results of the analysis carried

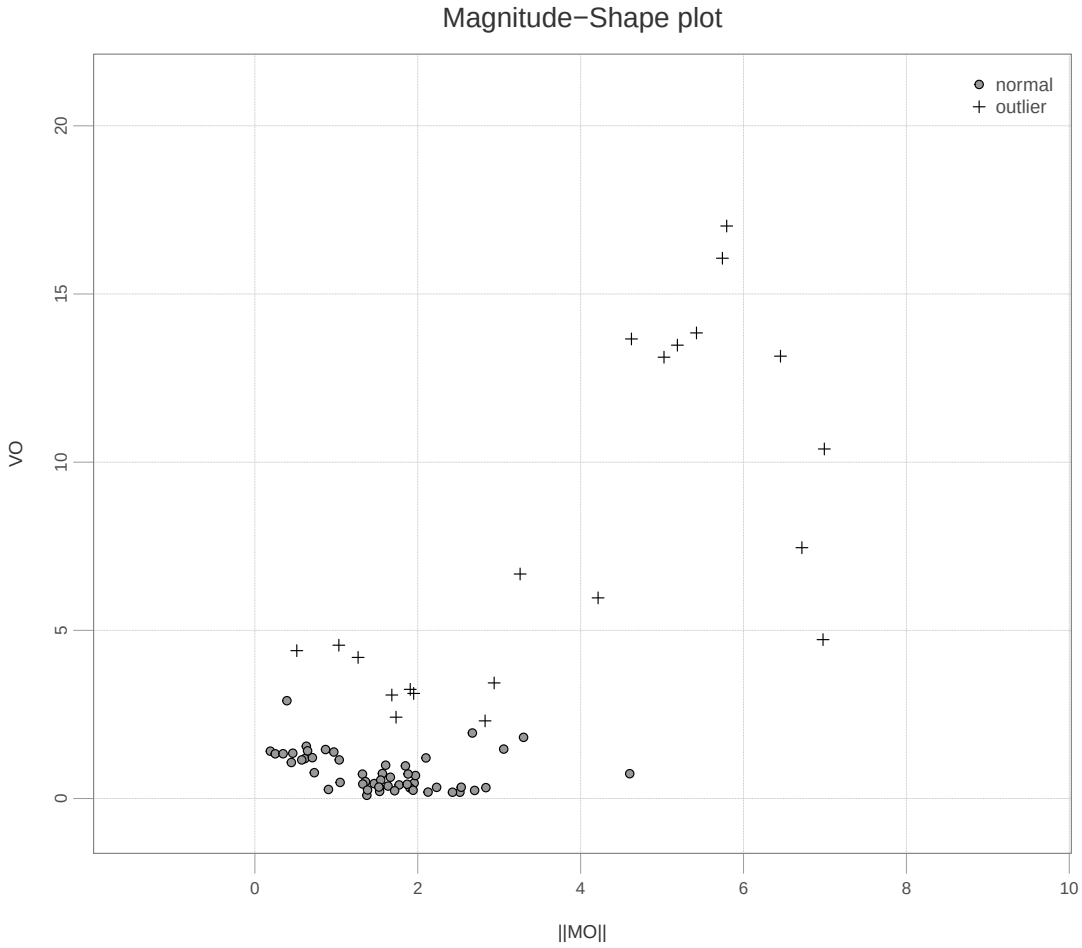

Figure 2.5: Plot of VO and $\|\mathbf{MO}\|$ for the joint multivariate functional data of temperature and log precipitation.

out in Dai et al. (2020) which identified different types of outliers in the data using the transformation $\mathcal{T}_2 \circ \mathcal{T}_1 \circ \mathcal{T}_0$ (and the L^∞ depth in the functional boxplots).

```
R> seq_pop <- seq_transform(dts = world_population,
+                           sequence = c("T0", "T1", "T2"),
+                           depth_method = "linfinity")
```

The magnitude outliers are then the $\mathcal{T}_0$ outliers which can be obtained with:

```
R> seq_pop$outliers$T0
```

```
[1]  5  9 18 25 40 41 44 49 55
```

```
R> (t0_outliers <- rownames(world_population)[seq_pop$outliers$T0])
```

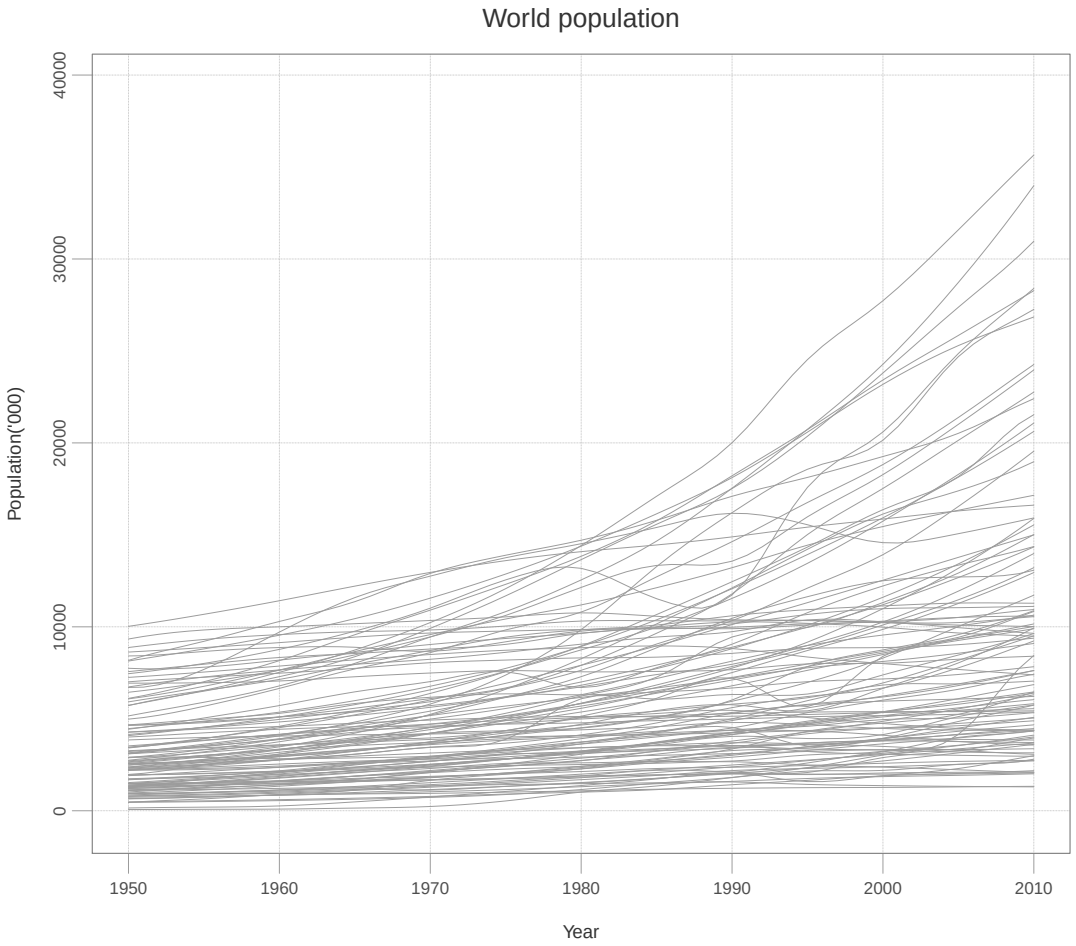

Figure 2.6: World population in thousands of 105 countries from 1950-2010.

```
[1] "Mozambique"   "Uganda"       "Sudan"        "Ghana"
[5] "Afghanistan"  "Nepal"        "Malaysia"     "Iraq"
[9] "Saudi Arabia"
```

and the $\mathcal{T}_1$ outliers:

```
R> seq_pop$outliers$T1

 [1]  3  5  9 12 13 18 24 25 36 40 41 44 49 55 57 59

R> (t1_outliers <- rownames(world_population)[seq_pop$outliers$T1])

 [1] "Madagascar"           "Mozambique"
 [3] "Uganda"               "Angola"
```

```
 [5] "Cameroon"              "Sudan"
 [7] "Cote d'Ivoire"         "Ghana"
 [9] "Kazakhstan"            "Afghanistan"
[11] "Nepal"                 "Malaysia"
[13] "Iraq"                  "Saudi Arabia"
[15] "Syrian Arab Republic" "Yemen"
```

In Dai et al. (2020), they considered the $\mathcal{T}_1$ outliers which are not $\mathcal{T}_0$ outliers as amplitude outliers for classification purposes. These can be obtained with:

```
R> amp_ind <- seq_pop$outliers$T1[!(seq_pop$outliers$T1
+                                   %in% seq_pop$outliers$T0)]
R> rownames(world_population)[amp_ind]
```

```
[1] "Madagascar"           "Angola"
[3] "Cameroon"             "Cote d'Ivoire"
[5] "Kazakhstan"           "Syrian Arab Republic"
[7] "Yemen"
```

Finally, the shape outliers are the $\mathcal{T}_2$ outliers that are neither $\mathcal{T}_0$ outliers nor $\mathcal{T}_1$ outliers.

```
R> shape_ind <- seq_pop$outliers$T2[!(seq_pop$outliers$T2
+                          %in% seq_pop$outliers$T0)]
R> shape_ind <- shape_ind[!(shape_ind %in% seq_pop$outliers$T1)]
R> rownames(world_population)[shape_ind]
```

```
 [1] "Rwanda"                 "Armenia"
 [3] "Georgia"                "Belarus"
 [5] "Bulgaria"               "Czech Republic"
 [7] "Hungary"                "Republic of Moldova"
 [9] "Estonia"                "Latvia"
[11] "Lithuania"              "Bosnia and Herzegovina"
[13] "Croatia"
```

The $\mathcal{T}_0$ and $\mathcal{T}_1$ outliers are mostly countries in Africa and the Middle East while the shape outliers ($\mathcal{T}_2$ outliers) are mostly Eastern and Central European countries. The outliers detected are visualised in Figure 2.7, and these results are consistent with Dai et al. (2020, Table 5). We can also use `tvdmss()` and `muod()` on the world population data:

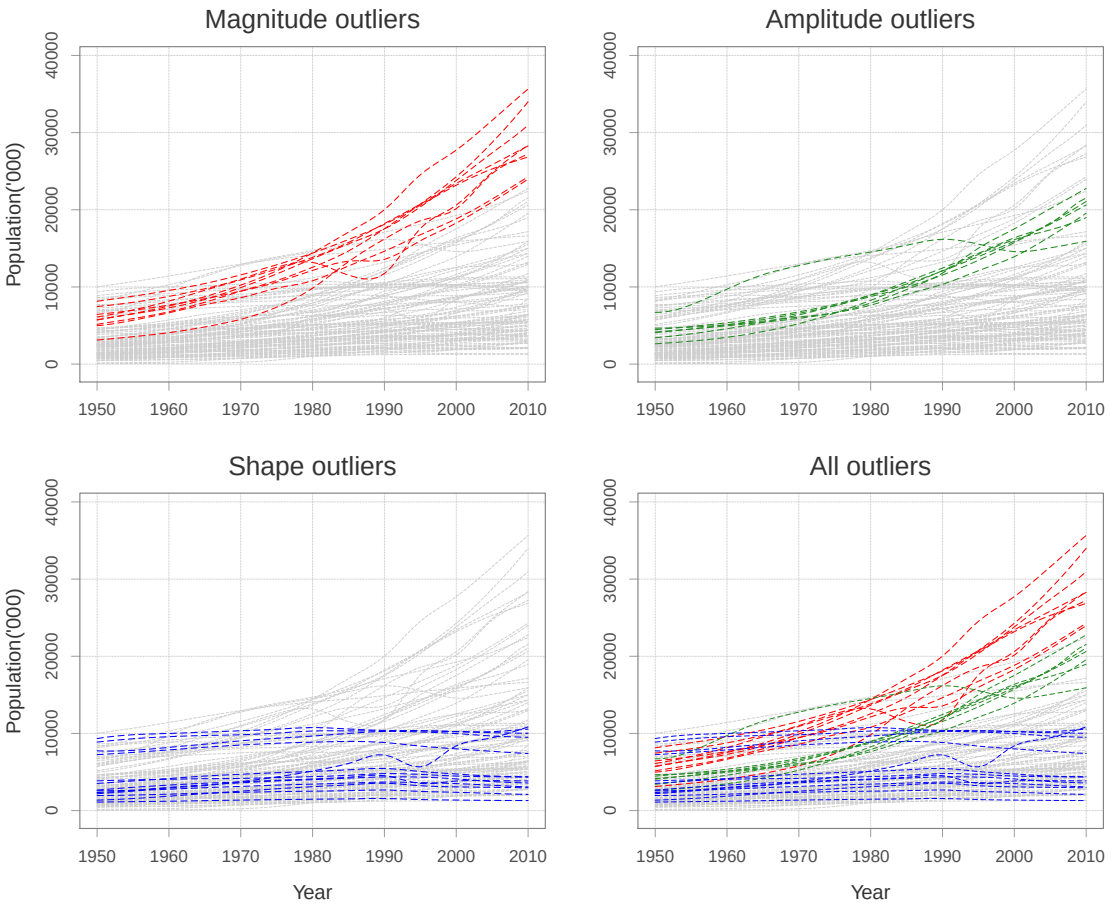

Figure 2.7: Outliers detected in world population data using sequential transformation. Curves in red are magnitude outliers, in blue are shape outliers, in green are amplitude outliers and in grey are normal observations.

```
R> wptvdout <- tvdmss(dts = world_population)
R> wpmout <- muod(dts = world_population, cut_method = "boxplot")
```

TVD did not detect any magnitude outlier but MSS does find a couple of "shape" outliers all of which are in Africa and the Middle East, and these countries are those classified as $\mathcal{T}_0$ and $\mathcal{T}_1$ outliers by sequential transformations above:

```
R> wptvdout$magnitude_outliers

NULL

R> rownames(world_population)[wptvdout$shape_outliers]

 [1] "Mozambique"            "Uganda"
 [3] "Sudan"                 "Cote d'Ivoire"
```

```
 [5] "Ghana"                "Kazakhstan"
 [7] "Afghanistan"          "Nepal"
 [9] "Malaysia"             "Iraq"
[11] "Saudi Arabia"         "Syrian Arab Republic"
[13] "United Arab Emirates" "Yemen"
```

The countries flagged as magnitude outliers by `muod()` are very similar to those flagged by sequential transformations as $\mathcal{T}_0$ outliers:

```
R> rownames(world_population)[wpmout$outliers$magnitude]

[1] "Sudan"         "Saudi Arabia"  "Uganda"        "Iraq"
[5] "Cote d'Ivoire" "Malaysia"
```

The amplitude outliers detected by `muod()` that are also not magnitude outliers are:

```
R> wpamp_ind <- wpmout$outliers$amplitude[!(wpmout$outliers$amplitude
+                                           %in% wpmout$outliers$magnitude)]
R> rownames(world_population)[wpamp_ind]

[1] "Nepal"                "Ghana"
[3] "Syrian Arab Republic" "Yemen"
[5] "Afghanistan"          "Mozambique"
[7] "Madagascar"
```

Finally, the shape outliers flagged by `muod()` that are neither amplitude nor magnitude outliers are mostly the Eastern and Central European countries flagged as $\mathcal{T}_2$ outliers by sequential tranformations:

```
R> wpshape_ind <- wpmout$outliers$shape[!(wpmout$outliers$shape
+                                         %in% wpmout$outliers$magnitude)]
R> wpshape_ind <- wpshape_ind[!(wpshape_ind %in% wpmout$outliers$amplitude)]
R> rownames(world_population)[wpshape_ind]

 [1] "Bulgaria"              "Hungary"
 [3] "Latvia"                "Georgia"
 [5] "Estonia"               "Bosnia and Herzegovina"
 [7] "Lithuania"             "Republic of Moldova"
 [9] "Croatia"               "Armenia"
[11] "Belarus"               "United Arab Emirates"
[13] "Kazakhstan"            "Czech Republic"
```

In conclusion, the outliers detected and the classification of such outliers may vary across different outlier detection methods as shown by this world population data example. While the results of the outlier detection and classification for MUOD and sequential transformations are quite similar, those of TVD and MSS are quite different and in particular, countries identified as shape outliers by TVD and MSS are those flagged as magnitude and amplitude outliers by MUOD and sequential tranformations.

2.4 Discussion

This chapter introduces the **fdaoutlier** package, which extends the available tools for outlier detection in functional data analysis in R. **fdaoutlier**'s focus so far has been the implementation of the latest state-of-the-art outlier detection methods in the FDA literature that are not yet implemented in R and that are especially useful for detecting shape outliers. These include the directional outlyingness and MS-plot, the total variation depth and modified shape similarity, and sequential transformations. These implementations will be especially useful to FDA researchers for testing and comparisons in the development of new outlier detection and exploratory methods. Likewise, **fdaoutlier** will be useful for practitioners in the exploratory analysis of their functional data.

We will continue adding more outlier detection methods for functional data to the **fdaoutlier** package, especially (future) outlier detection methods not yet implemented in R, and also those proposed in the subsequent chapters of this thesis. We will also continue adding useful tools for development and testing of new outlier detection methods for functional data, e.g., additional simulation models and functions for comparing outlier detection methods, and these will be directed by the trends in the FDA literature. Our long-term goal for the development of **fdaoutlier** is for it to be a helpful package for practitioners and researchers alike for conducting robust analysis and outlier detection for functional data.

Chapter 3

Detecting and Classifying Outliers in Big Functional Data

This chapter is a reprint of:

Ojo, O. T., Fernández Anta, A., Lillo, R. E., & Sguera, C. (2021). "Detecting and classifying outliers in big functional data". *Advances in Data Analysis and Classification*, 1-36. DOI: 10.1007/s11634-021-00460-9

Copyright:

Acknowledgements:

This research was funded in part by Agencia Estatal de Investigación (AEI) grant number AEI/10.13039/501100011033. This research was also partially supported by the Regional Government of Madrid (CM) grant EdgeData-CM (P2018/TCS4499, cofunded by FSE & FEDER) and Agencia Estatal de Investigación (AEI) grant PID2019-109805RB-I00/ AEI/10.13039/501100011033.

Abstract:

We propose two new outlier detection methods, for identifying and classifying different types of outliers in (big) functional data sets. The proposed methods are based on an existing method called Massive Unsupervised Outlier Detection (MUOD). MUOD detects and classifies outliers by computing for each curve, three indices, all based on the concept of linear regression and correlation, which measure outlyingness in terms of shape, magnitude and amplitude, relative to the other curves in the data. 'Semifast-MUOD', the first method, uses a sample of the observations in computing the indices, while 'Fast-MUOD', the second method, uses the point-wise or L_1 median in computing the indices. The classical boxplot is used to separate the indices of the outliers from those of the typical observations. Performance evaluation of the proposed methods using simulated data show significant improvements compared to MUOD, both in outlier detection and computational time. We show that Fast-MUOD is especially well suited to handling big and dense functional datasets with very small computational time compared to other methods. Further comparisons with some recent outlier detection methods for functional data also show superior or comparable outlier detection accuracy of the proposed methods. We apply the proposed methods on weather, population growth, and video data.

3.1 Introduction

Technological advances in the latest decades have allowed the observation of data samples that can be considered as functions or curves over a domain. These include temperature data over time, pixel values of images, frames of video, etc., and it is natural to assume that these observations have been generated by a stochastic function over a domain. Functional data analysis (FDA) deals with statistical analysis of these types of data. We refer the reader to Ramsay and Silverman (2006) for an overview of statistical methods for analysing functional data. Non-parametric methods for FDA have also been treated in Ferraty and Vieu (2006), while a survey of theory of statistics for FDA can be found in Cuevas (2014).

It is common practise to identify outliers before conducting statistical analyses. Outliers are of interest because they could significantly bias the results of statistical inference. Furthermore, an outlier, rather than being due to a measurement error, could be due to some interesting changes or behaviour in the data-generating process, and it is often of interest to investigate such changes. This is even more important in the analysis of weather, pollution, and geochemical data where identifying such changes is necessary to make important environmental policy decisions (e.g., Filzmoser et al. (2005)). In the context of FDA, identifying outliers becomes even more difficult because of the nature of functional observations. Such observations are realizations of functions over an interval and thus, outlying observations could have extreme values in a part of the interval or in all the interval. These (outlying) functional observations could exhibit different properties which make them anomalous. These include being significantly shifted from the rest of the data or having a shape that on the average is different from the rest of the data. Hubert et al. (2015) defined the former as magnitude outliers, the latter as shape outliers, and in addition defined amplitude outliers as curves or functions which may have the same shape as the mass of the data but with different amplitude.

Outliers in multivariate data are typically identified using notions of statistical depth, which provide a centre-outward ordering for observations. Statistical depths were generalized to the functional domain starting with the work of Fraiman and Muniz (2001). Since then, various depth notions for ordering functional data have been introduced, including band depth and modified band depth (López-Pintado and Romo 2009), extremal depth (Narisetty and Nair 2016), half-region depth (López-Pintado and Romo 2011), and total variation depth (Huang and Sun 2019), among others (see Nieto-Reyes and Battey (2016) for more details). A number of exploratory and outlier detection methods for functional data are based on functional depth notions. For instance, Febrero et al. (2008) proposed an outlier detection method using functional depths with cutoffs determined through a bootstrap, while Sguera et al. (2015) proposed to use a kernel-

ized functional spatial local depth (KFSD) for identifying outliers. The functional boxplot (Sun and Genton 2011) also uses the modified band depth to define a 50% central region on a functional data with curves outside 1.5 times the central region flagged as outliers, analogous to the classical boxplot. Likewise Arribas-Gil and Romo (2014) proposed the outliergram, which uses the quadratic relationship between the modified band depth and the modified epigraph index (López-Pintado and Romo 2011) to identify shape outliers. Other methods, like the functional bagplot or the functional highest density regions (Hyndman and Shang 2010), use the first two principal components of the functional data to construct a bagplot or a highest density region plot, respectively, to identify outliers. Hubert et al. (2015) also proposed using a bag distance and skewness adjusted projection depth to identify outliers.

More recent literature include the work of Dai and Genton (2018), in which they constructed a magnitude-shape plot (MS-Plot) for visualizing the centrality of multivariate functional observations and for identifying outliers, using a functional directional outlyingness measure. This functional directional outlyingness measure for multivariate functional data was further investigated in Dai and Genton (2019). Furthermore, Rousseeuw et al. (2018) introduced another measure of functional directional outlyingness for multivariate functional data and used it to construct the functional outlier map (FOM) for identifying outliers in multivariate functional data, while Huang and Sun (2019) defined the shape similarity index and the modified shape similarity index based on total variation depth to identify shape outliers. Dai et al. (2020) proposed to use some predefined sequence of transformations to identify and classify shape and amplitude outliers after first removing the magnitude outliers using a functional boxplot.

It is desirable to be able to identify all the different types of outliers in functional data. However, some outlier detection methods for functional data are specialized, in the sense that they are well suited to identifying outliers of a certain type; e.g., outliergram is well suited to identifying shape outliers, while functional boxplot is well suited to identifying magnitude outliers. While some methods are sensitive to different types of outliers, they do not automatically provide information on the type of outliers, unless the data is visualized. Thus, it might be difficult to understand why a particular curve is flagged as an outlier. This is especially important when the functional data is large and not easy to visualize. Classifying the types of outliers also allows for selectively targeting different types of outliers. For example, one might be interested only in shape outliers or only in magnitude outliers. Furthermore, some methods do not scale up to large functional datasets, which poses a challenge with the huge amounts of data that is being generated nowadays.

In this chapter, we introduce two new outlier detection methods for univariate func-

tional data: Fast-MUOD and Semifast-MUOD, which are based on the Massive Unsupervised Outlier Detection (MUOD) method proposed in Azcorra et al. (2018). These methods are capable of identifying and classifying magnitude, amplitude and shape outliers without the need for visualization. The proposed methods are based on the concepts of linear regression and correlation, making them quite intuitive and easy to compute. We also show that one of the proposed methods, Fast-MUOD, scales quite well and is thus suitable for detecting outliers in big functional data. We show that these methods have good outlier detection performance on a range of outlier types using simulation experiments and we also compare positively, their outlier detection performance and computation time to some existing outlier detection methods for functional data. The main contributions of this work are:

- Proposal of two new methods capable of identifying and classifying outliers in functional data.

- Simulation study comparing the proposed methods and some other recent outlier detection methods.

- Time benchmark (comparing the proposed methods and other outlier detection methods) showing the computational time of the proposed methods.

- Case studies showing how the proposed methods can be used in a real application and comparisons with some existing work with similar case studies.

- An implementation of the proposed methods available on Github.

The rest of the article is organized as follows: Section 3.2 provides an overview of MUOD. In Section 3.3, we present the proposed methods. These are followed by performance evaluation with some simulation studies in Section 3.4. We illustrate in Section 3.5, the use of the proposed methods on a variety of real datasets and use cases, including object detection in surveillance video, outlier detection in weather data, and discovering growth trends in population data. We end this chapter with some discussions and conclusions in Section 3.6.

3.2 The MUOD Method

In this section, we present a brief primer on MUOD as described in the supplementary material of Azcorra et al. (2018). MUOD identifies outliers by computing for each observation or curve, three indices, namely shape, magnitude and amplitude indices. These indices measure how outlying each observation is as regards its shape, magnitude and

amplitude, compared to the other observations. The definition of these indices as defined in Azcorra et al. (2018) is introduced in the following.

Consider a set of functional observations $\{Y_i\}_{i=1}^n \in \mathcal{C}(\mathcal{I})$, defined on d equidistant points of an interval $\mathcal{I} \in \mathbb{R}$, where $\mathcal{C}(\mathcal{I})$ is the space of real continuous functions defined on $\mathcal{I}$. We assume that Y_i follows a distribution F_Y also defined on $\mathcal{C}(\mathcal{I})$. We define the MUOD shape index of Y_i with respect to F_Y, denoted by $I_S(Y_i, F_Y)$, as

$$I_S(Y_i, F_Y) = \left| \frac{1}{n} \sum_{j=1}^{n} \hat{\rho}(Y_i, Y_j) - 1 \right|, \tag{3.1}$$

where $\hat{\rho}(Y_i, Y_j)$ is the estimated Pearson correlation coefficient between Y_i and Y_j, given by

$$\hat{\rho}(Y_i, Y_j) = \frac{\text{cov}(Y_i, Y_j)}{s_{Y_i} s_{Y_j}}, \qquad s_{Y_i}, s_{Y_j} \neq 0$$

The correlation coefficient is responsible for capturing the similarity between each pair of curves (Y_r, Y_s) in terms of shape. The intuition behind the MUOD shape index is as follows. Assume that the number of outlying curves n_o is much less than the number of non-outlying curves n_n, i.e., $(n_o << n_n)$. Let Y_i be a normal curve (in terms of shape) with respect to (w.r.t.) F_Y and let Y_k be a shape outlier w.r.t. F_Y. Also, denote by $\{Y_j\}_{j=1}^{n_n}$, the set of normal curves and by $\{Y_l\}_{l=1}^{n_o}$ the set of outlying curves. Since Y_i has a similar shape w.r.t. F_Y and Y_k is a shape outlier, the correlations between the pairs $(Y_i, Y_j)_{j=1}^{n_n}$ will be close to 1 and greater than the correlations between the pairs $(Y_k, Y_j)_{j=1}^{n_n}$. Also, the correlations between the pairs $(Y_i, Y_l)_{l=1}^{n_o}$ and $(Y_k, Y_l)_{l=1}^{n_o}$ could be any value between -1 and 1. However, $n_o << n_n$ ensures that the average of the correlations over all possible pairs $(Y_i, Y_m)_{m=1}^{n}$ is greater than the average of the correlation overall all possible pairs $(Y_k, Y_m)_{m=1}^{n}$. Consequently, subtracting these averages from 1 assigns Y_i a lesser shape index compared to the shape index of Y_k. We illustrate this behaviour of the MUOD shape index below. We generate 99 non-outlying curves from the model

$$Y(t) = a_1 \sin(t) + a_2 \cos(t), \tag{3.2}$$

where $t \in T$, with T made up of $d = 50$ equidistant domain points between 0 and 2π, and both a_1 and a_2 generated from independent uniform random variables between 0.75 and 1.25. Moreover, we generate a single shape outlier using the following different model:

$$Y(t) = b_1 \sin(t) + b_2 \cos(t) + \epsilon(t), \tag{3.3}$$

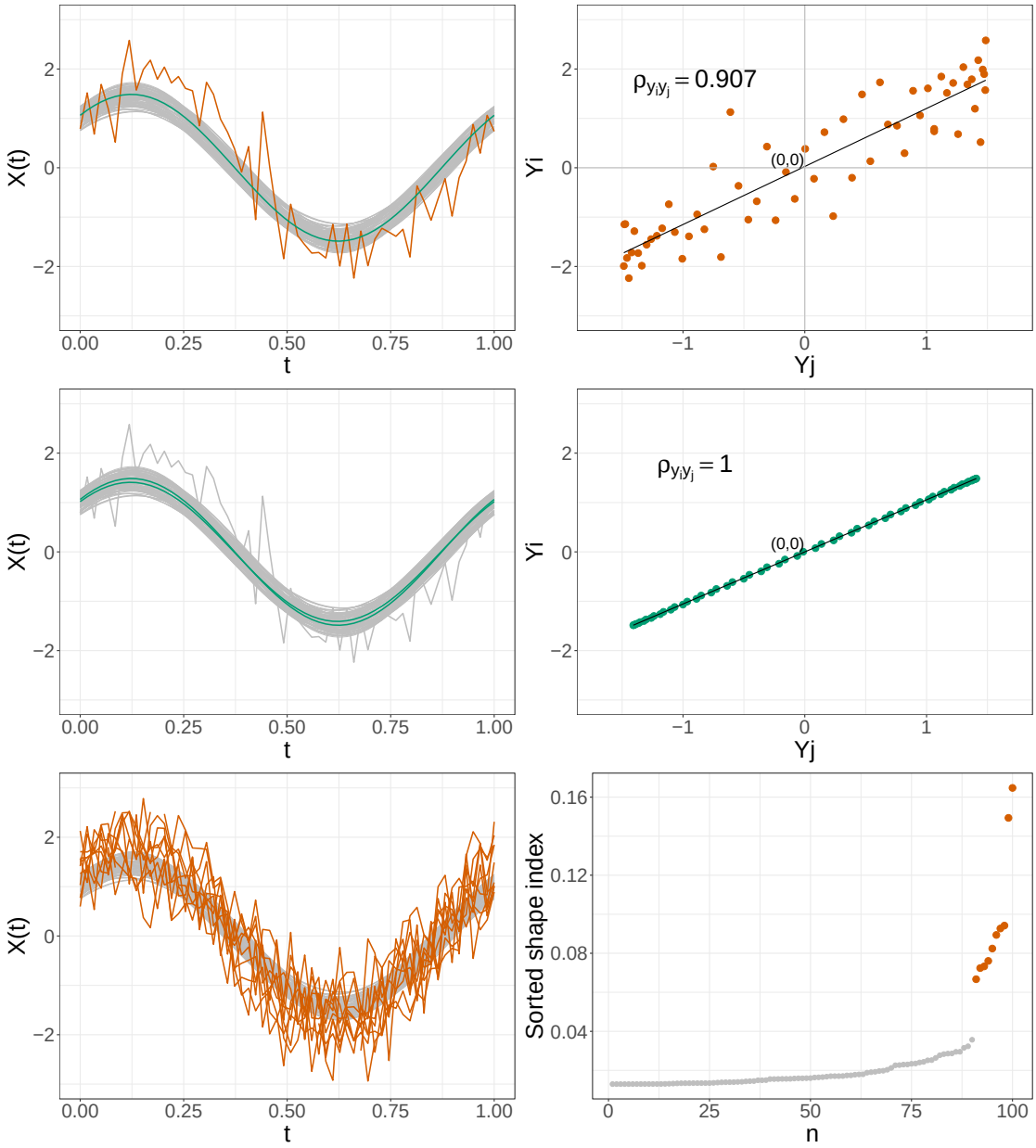

Figure 3.1: First Row Left: simulated data using Equation (3.2) (99 curves, 98 in gray, 1 in green) and Equation (3.3) (1 curve, in orange). First Row Right: estimated correlation coefficient between the observed points of the orange curve and the green curve. Second Row Left: same as First Row Left, highlighting two normal curves (green). Second Row Right: estimated correlation coefficient between the green curves. Third Low Left: Simulated data set using Equation (3.2) for normal curves (in gray) and Equation (3.3) for outliers (orange). Third Row Right: associated sorted MUOD shape indices.

where each $\epsilon(t)$ is drawn from a normal random variable with $\mu = 0$ and $\sigma^2 = 1/4$, and both b_1 and b_2 are realizations of independent uniform random variables with parameters 0.75 and 1.75. In the first row of Figure 3.1, we highlight a typical observation in green and the single shape outlier in orange (left) and show their estimated correlation coefficient (right). In the second row of Figure 3.1, we show the estimated correlation coefficient for two typical curves colored in green. We observe that the estimated correlation coefficients of the two typical curves is greater than that of the "typical-outlier" pair of curves. In the third row of Figure 3.1, we show the MUOD shape indices of 90 typical curves and 10 outliers generated from Equations (3.2) and (3.3) respectively. Clearly, the indices of the shape outliers (in orange) are greater than the indices of the typical observations (in grey). We note that the conditions $s_{Y_i} \neq 0$ and $s_{Y_j} \neq 0$ in Equation (3.1) can easily be broken if any of the curves Y_i or Y_j is a straight line. To avoid this, we ignore any curve Y_j in the data set with $s_{Y_y} = 0$ when computing the indices.

The magnitude and amplitude indices of an observation Y_i, denoted by $I_M(Y_i, F_Y)$ and $I_A(Y_i, F_Y)$ respectively, are based on the intercept and slope of a linear regression between the observed points of all possible pairs $(Y_i, Y_j)_{j=1}^n$. Let $\hat{\alpha}_j$ and $\hat{\beta}_j$ be the estimated coefficients (intercept and slope respectively) of the linear regression between the pair (Y_i, Y_j) with the observed points of the function Y_j being the independent variable, and the observed points of the function Y_i being the dependent variable. Then the magnitude index $I_M(Y_i, F_Y)$ of Y_i is defined as:

$$I_M(Y_i, F_Y) = \left| \frac{1}{n} \sum_{j=1}^{n} \hat{\alpha}_j \right|, \tag{3.4}$$

and the amplitude index $I_A(Y_i, F_Y)$ of Y_i is defined as:

$$I_A(Y_i, F_Y) = \left| \frac{1}{n} \sum_{j=1}^{n} \hat{\beta}_j - 1 \right|, \tag{3.5}$$

with

$$\hat{\beta}_j = \frac{\text{cov}(Y_i, Y_j)}{s^2_{Y_j}}, \qquad s^2_{Y_j} \neq 0,$$

and

$$\hat{\alpha}_j = \bar{x_i} - \hat{\beta}_j \bar{x}_j,$$

where

$$\bar{x_i} = \frac{\sum_{t \in \mathcal{I}} Y_i(t)}{d}.$$

The intuition behind the magnitude index is similar to that of the shape index. We

adapt the same notation used before for shape outliers to magnitude outliers. If Y_k is a magnitude outlier w.r.t. F_Y, and Y_i is a typical curve (in terms of magnitude) w.r.t. F_Y, then a linear regression between the d observed points of Y_k on those of any Y_j (the typical curves) will produce a large estimated intercept coefficient $\hat{\alpha}_{kj}$ compared to the esimated intercept $\hat{\alpha}_{ij}$ of the linear regression of Y_i on Y_j (since both Y_i and Y_j are not magnitude outliers). Provided that $n_o << n_n$, the average of the estimated $\hat{\alpha}_{km}$ values over all possible pairs $(Y_k, Y_m)_{m=1}^n$ will be greater than the average of the estimated $\hat{\alpha}_{im}$ values over all possible pairs $(Y_i, Y_m)_{m=1}^n$, which consequently assigns a larger magnitude index to Y_k, the magnitude outlier. To illustrate the magnitude index, we generate 99 observations using Equation (3.2) and a single magnitude outlier from the model below:

$$Y(t) = a_1 \sin(t) + a_2 \cos(t) + 1. \tag{3.6}$$

In the first row of Figure 3.2, we show the simulated data set (left) and the estimated linear regression model between a randomly selected non-outlying curve and the unique (magnitude) outlying function together with the value of their estimated intercept (right). In the second row of Figure 3.2, we show the same simulated data set (left) and the estimated linear regression model between two randomly selected non-outlying curves (right). A comparison of the estimated intercepts (of the former and the latter pairs of functions) shows that the estimated intercept for "normal-outlier" pair of curves is greater than that of the "normal-normal" pair of curves. Finally, in the third row of Figure 3.2, we show another simulated data set (left) where normal observations are generated using Equation (3.2), and 10 magnitude outliers are generated using Equation (3.7):

$$Y(t) = a_1 \sin(t) + a_2 \cos(t) + k, \tag{3.7}$$

where k takes either -1 or 1 with equal probability, and it controls whether an outlier is higher or lower in magnitude than the typical observations. On the right of the third row of Figure 3.2, we show the sorted MUOD magnitude indices. All the low and high magnitude outliers (in blue and orange respectively) have significantly larger indices than the typical observations.

Unlike the magnitude index which uses the intercept term, the amplitude index uses the slope term. The same intuition applies for the amplitude index because if both Y_i and Y_j are similar curves (in amplitude), increasing and decreasing in amplitude at a similar rate, then the linear regression between their d observed points will produce an estimated slope coefficient $\hat{\beta}_j$ close to 1. We illustrate the amplitude index in Figure 3.3. This figure resembles Figure 3.2, but with amplitude outliers, which we generate using

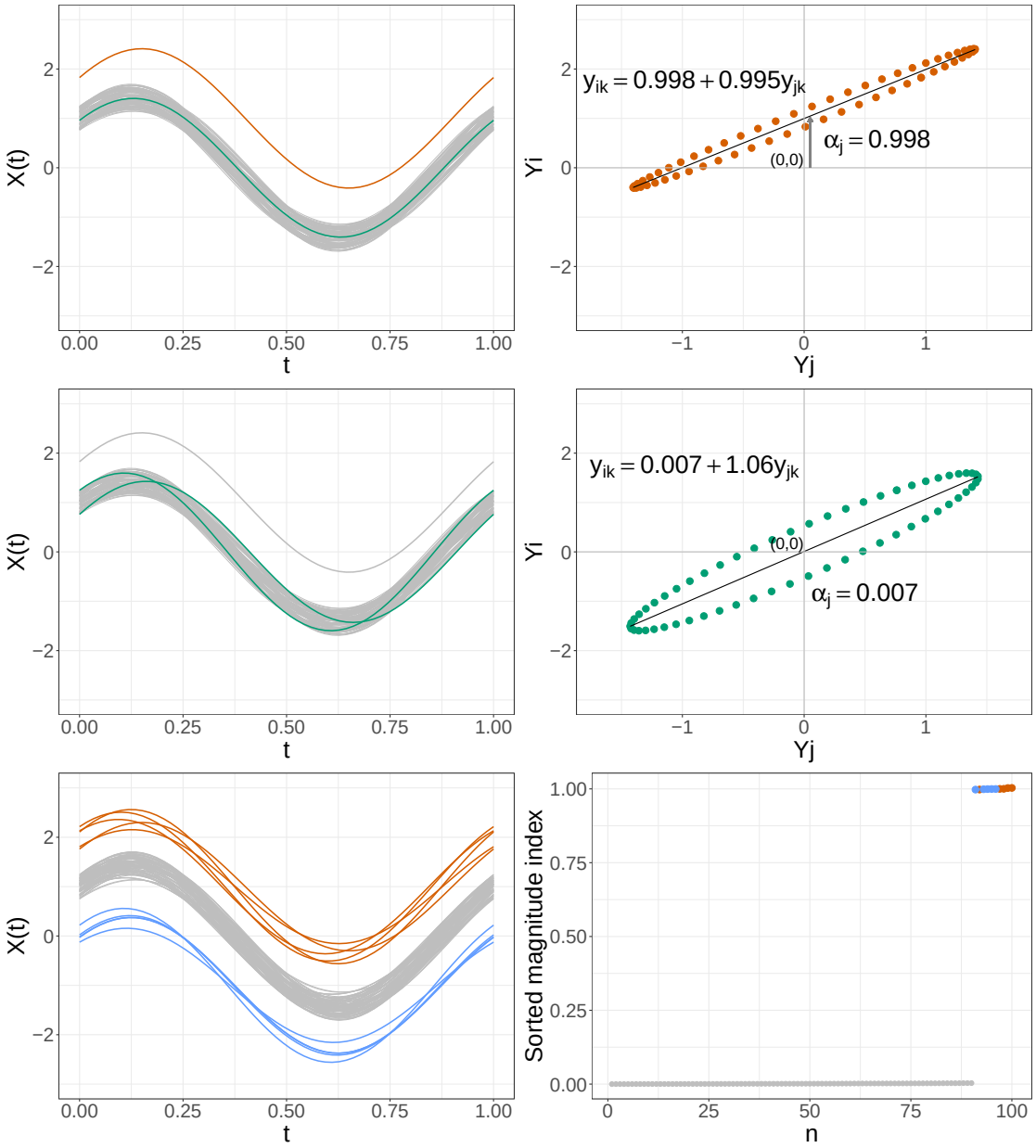

Figure 3.2: First Row Left: simulated data using Equation (3.2) (99 curves, 98 in gray, 1 in green) and Equation (3.7) (1 curve, in orange). First Row Right: estimated linear regression model of the orange curve on the green curve. Second Row Left: same as First Row Left, highlighting two normal curves (green). Second Row Right: estimated linear regression model between the green curves. Third Low Left: Simulated data set using Equation (3.2) for normal curves (in gray) and Equation (3.7) for outliers (in blue and orange). Third Row Right: associated sorted MUOD magnitude indices.

the model in Equation (3.8):

$$Y(t) = c_1 \sin(t) + c_2 \cos(t), \tag{3.8}$$

where c_1 and c_2 are independent uniform random variables between 1.7 and 2.0 for higher amplitude outliers; and between 0.2 and 0.4 for lower amplitude outliers. From Figure 3.3, the estimated slope coefficient between the amplitude outlier (in orange) and the typical observation (in green) is $\hat{\beta}_j = 1.855$ (top row), while the estimated slope coefficient between the two typical observations is $\hat{\beta}_j = 0.979$ (second row). Moreover, the sorted MUOD amplitude indices of the amplitude outliers are greater than those of the typical observations (third row).

After obtaining the MUOD indices as defined above, the next step in outlier detection is to differentiate the indices of the outliers from the indices of the typical observations. Azcorra et al. (2018) proposed two heuristic methods to perform this task. The first involves approximating the sorted indices with a curve and searching for a cutoff point on the curve where the first derivative of such point fulfills a certain condition (e.g., a point on the curve with first derivative greater than 2). The other method, named '*tangent method*', searches for the line tangent to the maximum index and then uses as threshold the point at which the tangent intercepts the x-axis. These methods are particularly prone to detecting normal observations as outliers (Vinue and Epifanio 2020b). Furthermore, there is no statistical motivation behind these two proposed heuristic methods since they were mainly used as a quick support for identifying outliers in the real data application in Azcorra et al. (2018). As part of our proposed improvements, we use a classical boxplot for separating the indices of the outliers from those of the typical curves.

3.3 Fast-MUOD and Semifast-MUOD

We discuss the proposed methods based on MUOD in this section. First we describe how Semifast-MUOD and Fast-MUOD compute their outlier indices. Then we present the use of the classical boxplot for identifying a cutoff for the indices. We also describe their implementations.

3.3.1 Semifast-MUOD

Due to the way MUOD indices are defined, MUOD is computationally intensive and by design the time complexity for MUOD to compute its three indices is in the order of $\Theta(n^2 d)$. This is because the three indices of each of the n functional observations are

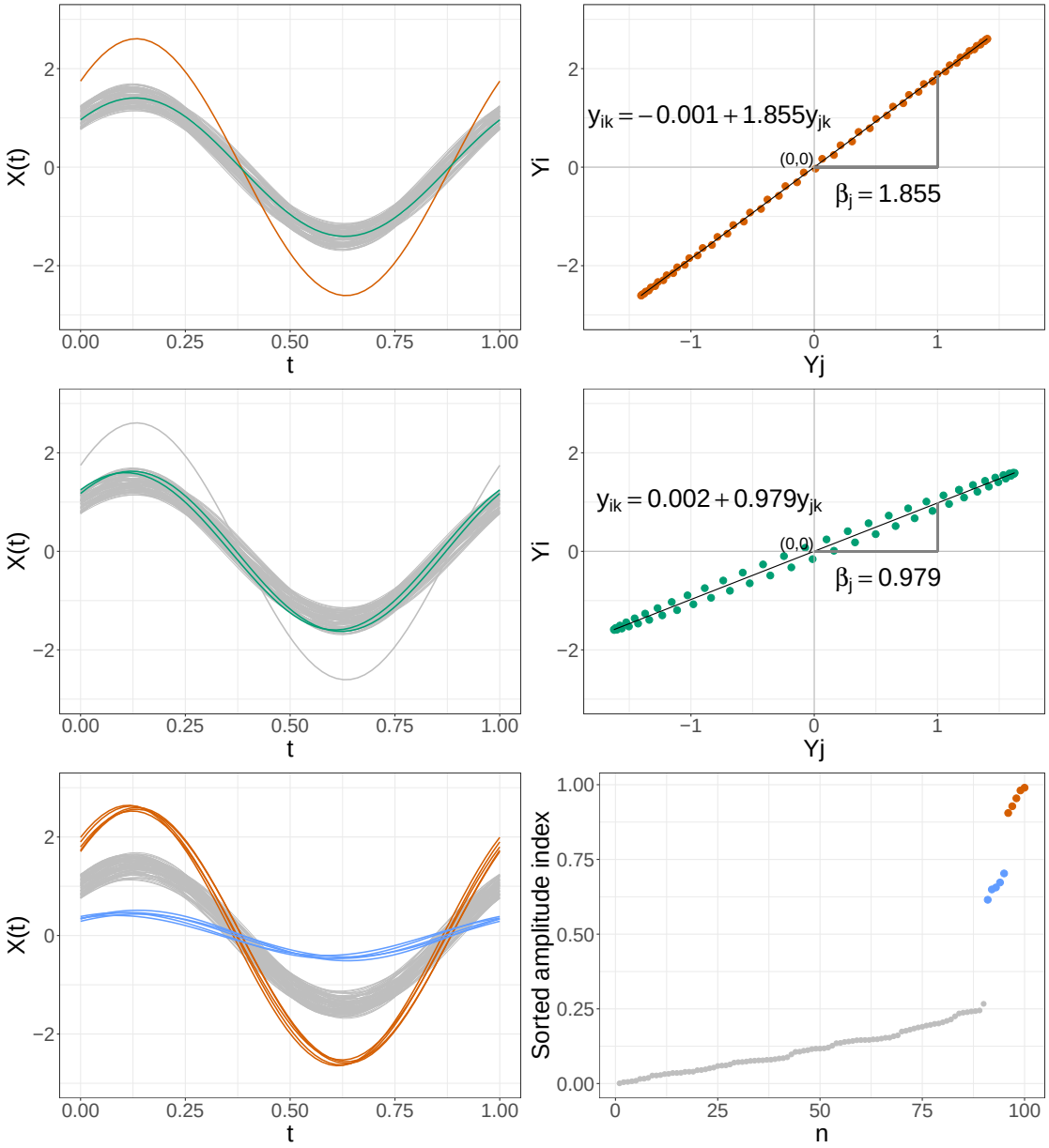

Figure 3.3: First Row Left: simulated data using Equation (3.2) (99 curves, 98 in gray, 1 in green) and Equation (3.8) (1 curve, in orange). First Row Right: estimated linear regression model of the orange curve on the green curve. Second Row Left: as First Row Left, highlighting two normal curves in green. Second Row Right: estimated linear regression model between the green curves. Third Row Left: Simulated data set using Equation (3.2) for normal curves (in gray) and Equation (3.8) for outliers (in blue and orange). Third Row Right: associated sorted MUOD amplitude indices.

computed by using all the n observations in the data. To reduce computational time, we propose to use a sample of the observations in the computation of the three indices. We pick a random sample (without replacement) of size n_X, from the set of observations $\{Y_i\}_{i=1}^n$ based on an appropriate sample proportion $p \in (0, 1]$. Denote this random sample by $\{X_i\}_{i=1}^{n_X}$ and its empirical distribution by F_X. Then, for each observation Y_i, the three indices for Y_i are computed using the n_X observations in $\{X_i\}_{i=1}^{n_X}$ rather than the n observations of Y. Formally, we define the shape index of any Y_i, now with respect to F_X, denoted by $I_S(Y_i, F_X)$ as

$$I_S(Y_i, F_X) = \left| \frac{1}{n_X} \sum_{j=1}^{n_X} \hat{\rho}(Y_i, X_j) - 1 \right|, \tag{3.9}$$

where $\hat{\rho}(Y_i, X_j)$ still remains the estimated Pearson correlation coefficient between Y_i and X_j for $i = 1. \ldots, n$ and $j = 1, \ldots, n_X$. Likewise, we define the new magnitude and amplitude indices, $I_M(Y_i, F_X)$ and $I_A(Y_i, F_X)$, computed w.r.t. F_X as

$$I_M(Y_i, F_X) = \left| \frac{1}{n_X} \sum_{j=1}^{n_X} \hat{\alpha}_j \right|, \tag{3.10}$$

$$I_A(Y_i, F_X) = \left| \frac{1}{n_X} \sum_{j=1}^{n_X} \hat{\beta}_j - 1 \right|, \tag{3.11}$$

where

$$\hat{\beta}_j = \frac{\text{cov}(Y_i, X_j)}{s^2_{X_j}}, \qquad s^2_{X_j} \neq 0$$

and

$$\hat{\alpha}_j = \bar{Y}_i - \hat{\beta}_j \bar{X}_j.$$

Semifast-MUOD has the advantage of reducing the computational time, since only a subsample of the functional data is used in computing the indices. Obviously, the gains in computational time is dependent on the sample size n_X, which is in turn dependent on the sample proportion p. Thus, the time complexity is reduced to an order of $\Theta(pn^2d)$.

3.3.2 Fast-MUOD

For Fast-MUOD, we propose to use only the point-wise median in the computation of the indices. Let $\tilde{Y}$ be the point-wise median of the observations in $\{Y\}_{i=1}^n$. Then, we compute the shape, magnitude and amplitude indices of any Y_i w.r.t. to $\tilde{Y}$, instead of

F_Y or F_X. We define the Fast-MUOD shape index of Y_i as

$$I_S(Y_i, \tilde{Y}) = \left| \hat{\rho}(Y_i, \tilde{Y}) - 1 \right|. \tag{3.12}$$

Likewise, the amplitude and magnitude indices of Y_i are given by

$$I_A(Y_i, \tilde{Y}) = \left| \hat{\beta}_i - 1 \right| \tag{3.13}$$

$$I_M(Y_i, \tilde{Y}) = |\hat{\alpha}_i| \tag{3.14}$$

where

$$\hat{\beta}_i = \frac{\text{cov}(Y_i, \tilde{Y})}{s^2_{\tilde{Y}}} \quad s^2_{\tilde{Y}} \neq 0,$$

and

$$\hat{\alpha}_i = \bar{Y}_i - \hat{\beta}_j \bar{\tilde{Y}}$$

Fast-MUOD is highly scalable since the time complexity has been reduced to an order of $\Theta(nd)$. These indices are more robust to outliers since they are computed with respect to only the point-wise median which corresponds to the depth median of the integrated Tukey halfspace depth (Nagy et al., 2016; Claeskens et al., 2014).

3.3.3 Alternative Medians and Correlation Coefficients

The point-wise median, in general, is not necessarily one of the observed curves, and its use (in Fast-MUOD) is to create a reference "typical" observation used for computing the indices, rather than identify a median observation of the functional data. Other median observations can be identified (and used in the computation of Fast-MUOD indices) using a functional depth measure. Although such a depth measure can also be used in detecting outliers (e.g., using a functional boxplot), our methods still provide the advantage of classifying the outliers. The point-wise median is desirable because it is fast and easy to compute, even for dense functional data.

As an alternative, the multivariate L_1 median can be used. However, we have found that this is difficult to compute for dense functional data observed on lots of domain points. Moreover, the use of the L_1 median in computing the indices does not show any significant gains in outliers detection performance in our simulation tests, despite being more computationally expensive (see Section A.1 of the Supplementary Material in Appendix A for comparison between Fast-MUOD using the L_1 median and the point-wise median). In general, we recommend the use of the point-wise median for dense and big functional data. For an overview of the computation of the L_1 median, we refer the reader to Fritz et al. (2011).

Likewise, other robust or non-parametric correlation coefficients like Kendall's Tau and Spearman's rank correlation coefficients have been considered in the formulation of the shape indices I_S. Results of our tests show that the Pearson correlation coefficient provides the best outlier detection performance. See Section A.4 of the Supplementary Material in Appendix A for a comparison of the performance of I_S for Fast-MUOD computed using different correlation coefficients.

3.3.4 Fast-MUOD and Semifast-MUOD Indices Cutoff

After obtaining the indices (using Fast-MUOD, or Semifast-MUOD), the next step in outlier detection is to determine a cutoff value for separating the outliers from the typical observations. The theoretical distributions of these indices are unknown, but simulations show that the distributions of these indices are right skewed and that the indices of the outliers appear on the right tail. Hence, a good cutoff method should be able to find a reasonable threshold in the right tails. We propose to use a classical boxplot on the indices. We declare Y_i a shape outlier if $I_S(Y_i, F) \geq Q_{3I_S} + 1.5 \times IQR_{I_S}$ where Q_{3I_S} and IQR_{I_S} are the third quartile and the inter-quartile range of I_S respectively, for $F \in \{F_X, \tilde{Y}\}$. We apply the same cutoff rule on the magnitude and amplitude indices $I_M(Y_i, F)$ and $I_A(Y_i, F)$. The identified outliers of each type are then returned (together with their type(s)), to give a clue why they are flagged as outliers.

We have also considered other cutting methods including the transformation of the indices and the use of more specialized boxplots (e.g., the adjusted boxplot for skewed distributions of Hubert and Vandervieren (2008) and the boxplot of Carling (2000)). We find that the adjusted boxplot is not sensitive enough to detect outliers and transformations of the indices usually worsen the separation between the indices of the outliers and typical observations. In our tests, the cutoff based on the classical boxplot performed well consistently acrosss the different types of outliers. Consequently, the results of the subsequent simulations and applications in this paper are obtained using this cutoff method for Semifast-MUOD and Fast-MUOD.

3.3.5 Implementation

MUOD was implemented in R, (R Core Team, 2022) with some of the computational intensive parts of the algorithm written in C++ using the Rcpp package (Eddelbuettel and Francois 2011). Fast-MUOD and Semifast-MUOD follow the same implementation. We provide an overview into the implementation of both methods in this section. For Semifast-MUOD, $I_A(Y, F_X)$, $I_S(Y, F_X)$, and $I_M(Y, F_X)$, are computed using Algorithm 2. The algorithm takes as input the row matrix $M_Y = [Y_1, \ldots Y_n]$ built from the obser-

Algorithm 2: SemiFastMUOD(M_Y)

1 M_X = sample(M_Y, p) : $\mathbb{R}^{d\times n} \to \mathbb{R}^{d\times n_X}$
2 means = colmean(M_Y): $\mathbb{R}^{d\times n} \to \mathbb{R}^{n}$
3 sds = colsd(M_Y): $\mathbb{R}^{d\times n} \to \mathbb{R}^{n}$
4 refmean = colmean(M_X): $\mathbb{R}^{d\times n_X} \to \mathbb{R}^{n_X}$
5 refvar = colvar(M_X): $\mathbb{R}^{d\times n_X} \to \mathbb{R}^{n_X}$
6 refsds = colsd(M_X): $\mathbb{R}^{d\times n_X} \to \mathbb{R}^{n_X}$
7 cov = covariance(M_X, M_Y): $\mathbb{R}^{d\times n_X} \times \mathbb{R}^{d\times n} \to \mathbb{R}^{n_X\times n}$
8 cor = cov/refsds/sds : $\mathbb{R}^{n_X\times n} \times \mathbb{R}^{n_X} \times \mathbb{R}^{n} \to \mathbb{R}^{n_X\times n}$
9 $I_S(Y, F_X) = |\text{colmean(cor)} - 1| : \mathbb{R}^{n_X\times n} \to \mathbb{R}^{n}$
10 β = cov/refvar : $\mathbb{R}^{n_X\times n} \times \mathbb{R}^{n_X} \to \mathbb{R}^{n_X\times n}$
11 $I_A(Y, F_X) = |\text{colmean}(\beta) - 1| : \mathbb{R}^{n_X\times n} \to \mathbb{R}^{n}$
12 $\beta x = \beta \times$ refmean: $\mathbb{R}^{n_X\times n} \times \mathbb{R}^{n_X} \to \mathbb{R}^{n_X\times n}$
13 α = means $- \beta x$: $\mathbb{R}^{n_X\times n} \times \mathbb{R}^{n} \to \mathbb{R}^{n_X\times n}$
14 $I_M(Y, F_X) = |\text{colmean}(\alpha)| : \mathbb{R}^{n_X\times n} \to \mathbb{R}^{n}$
15 Return $I_A(Y, F_X), I_M(Y, F_X), I_S(Y, F_X)$

vations in $\{Y_i\}_{i=1}^n$, with $|Y_i| = d$. Next, we randomly sample from the columns of M_Y to create the sample row matrix $M_X = [X_1, \ldots, X_{n_X}]$, the random sample to use for computing the indices. The rest of the computation follows as outlined in Algorithm 2. It is noteworthy that the covariance matrix in Line 7 of Algorithm 2 can become quite large easily. To manage memory, we implemented the computation of the values of this matrix and the rest of the indices sequentially in C++, so that we do not have to store the covariance matrix in memory. The implementation for Fast-MUOD is very similar and is outlined in Algorithm 3. The algorithm takes as input M_Y and then computes the point-wise median $\tilde{Y} \in \mathbb{R}^d$ which is used in the computation of the indices. The operations "colmean(·)", "colmedian(·)", "colsd(·)", and "colvar(·)" used in both algorithms indicate column-wise mean, median, standard deviation, and variance operations respectively.

3.4 Simulation Study

In this section, we evaluate the performance of the proposed methods using some simulation experiments.

3.4.1 Outlier Models

In our simulation study, we generate curves from different outlier models that have been studied in Dai and Genton (2018), Arribas-Gil and Romo (2014), Febrero et al.

Algorithm 3: FastMUOD(M_Y)

$\tilde{Y}$ = colmedian(M_Y): $\mathbb{R}^{d\times n} \to \mathbb{R}^d$
means = colmean(M_Y): $\mathbb{R}^{d\times n} \to \mathbb{R}^n$
sds = colsd(M_Y): $\mathbb{R}^{d\times n} \to \mathbb{R}^n$
refmean = mean($\tilde{Y}$): $\mathbb{R}^d \to \mathbb{R}$
refvar = var($\tilde{Y}$): $\mathbb{R}^d \to \mathbb{R}$
refsds = sd($\tilde{Y}$): $\mathbb{R}^d \to \mathbb{R}$
cov = covariance($\tilde{Y}$, M_Y): $\mathbb{R}^d \times \mathbb{R}^{d\times n} \to \mathbb{R}^n$
cor = cov/refsds/sds : $\mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \to \mathbb{R}^n$
$I_S(Y, \tilde{Y}) = |\text{cor} - 1| : \mathbb{R}^n \to \mathbb{R}^n$
β = cov/refvar : $\mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$
$I_A(Y, \tilde{Y}) = |\beta - 1| : \mathbb{R}^n \to \mathbb{R}^n$
$\beta x = \beta \times$ refmean: $\mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$
$\alpha = \text{means} - \beta \cdot x : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$
$I_M(Y, \tilde{Y}) = |\alpha| : \mathbb{R}^n \to \mathbb{R}^n$
Return $I_A(Y, \tilde{Y}), I_M(Y, \tilde{Y}), I_S(Y, \tilde{Y})$

(2008) and Sun and Genton (2011). In total, we consider eight models where the first model, Model 1, is a clean model with no outlier, while Models 2 – 8 contain outliers. The base models and the corresponding contamination models are specified below.

- **Model 1**: Main model $X_i(t) = 4t + e_i(t)$ with no contamination, for $i = 1, \ldots, n$. $e_i(t)$ is a Gaussian process with zero mean and covariance function $\gamma(s, t) = \exp\{-|t - s|\}$, where $s,\ t \in [0, 1]$.

- **Model 2**: Main model: same as Model 1; Contamination model: $X_i(t) = 4t + 8k_i + e_i(t)$, for $i = 1, \ldots, n$ and $k_i \in \{-1, 1\}$ with equal probability. $e_i(t)$ remains as defined above. This is a shifted model where the generated curves are magnitude outliers shifted from the main model.

- **Model 3**: Main model: same as Model 1; Contamination model: $X_i(t) = 4t + 8k_i I_{T_i \le t \le T_i + 0.05} + e_i(t)$, for $i = 1, \ldots, n$, $T_i \sim$ Unif$(0.1, 0.9)$, and I an indicator function. k_i and $e_i(t)$ remain as defined above. The outlying curves from this model are magnitude outliers for only a small portion of the domain, which produce spikes along the domain.

- **Model 4**: Main model: $X_i(t) = 30t(1-t)^{3/2} + \bar{e}_i(t)$; Contamination model: $X_i(t) = 30t^{3/2}(1 - t) + \bar{e}_i(t)$, for $i = 1, \ldots, n$; where $\bar{e}_i(t)$ is a Gaussian process with zero mean and covariance function $\bar{\gamma}(s, t) = 0.3 \exp\{-|s - t|/0.3\}$ with $s, t \in [0, 1]$. The outlying curves in this model produces outliers that are similar to typical observations but slightly shifted horizontally and reversed.

- **Model 5**: Main model: same as Model 1; Contamination model: $X_i(t) = 4t + e_{2_i}(t)$, for $i = 1, \ldots, n$; where $e_{2_i}(t)$ is a Gaussian process with zero mean and covariance function $\gamma_2(s,t) = 5\exp\{-2|t-s|^{0.5}\}$ with $s, t \in [0,1]$. The outlying curves generated are shape outliers with a different covariance function even though they follow the general trend of the normal observations.

- **Model 6**: Main model: Same as Model 1, Contamination model: $X_i(t) = 4t + 2\sin(4(t+\theta_i)\pi) + e_i(t)$, for $i = 1, \ldots, n$; where $\theta_i \sim Unif(.25, .75)$. $e_i(t)$ remains as defined above. Like Model 5 above, the generated outlying curves have the same trend as the normal observations but they are periodic in nature.

- **Model 7**: Main model $X_i(t) = a_i \sin\theta + b_i \cos\theta + e_i(t)$; Contamination model: $X_i(t) = (9\sin\theta + 9\cos\theta) \cdot (1 - u_i) + (p_i \sin\theta + q_i \cos\theta)u_i + e_i(t)$, for $i = 1, \ldots, n$; where $\theta \in [0, 2\pi]$, $a_i, b_i \sim$ Unif$(3, 8)$, $p_i, q_i \sim$ Unif$(1.5, 25.)$ and $u_i \in \{0, 1\}$ with equal probability. $e_i(t)$ remains as defined above. The contaminating curves are amplitude outliers with a similar periodic shape as the normal observations but with slightly increased or decreased amplitude.

- **Model 8**: Main model: same as Model 1; Contamination model: For each outlier to be generated, a contamination model is sampled from any of the following contamination models (with equal probability):

  (i) Contamination model of Model 2

  (ii) Contamination model of Model 3

  (iii) Contamination model of Model 5

  (iv) Contamination model of Model 6

  Thus, Model 8 is a mixture model containing different types of outliers.

Simulated data from these eight models will be a mixture of observations from the main model with outliers from the contamination model, where number of outliers is determined by the contamination rate alpha α. In the subsequent simulation results, we set the contamination rate $\alpha = 0.1$ for each Model 2 – 8, and we generate $n = 300$ curves on $d = 50$ equidistant points on the interval $[0, 1]$. Figure 3.4 shows a sample of the eight models with $\alpha = 0.1$, $n = 100$ and $d = 50$.

3.4.2 Outlier Detection Methods

We focus on comparing the outlier detection performance of the proposed methods to MUOD and to other recent outlier detection methods for functional data. Since the

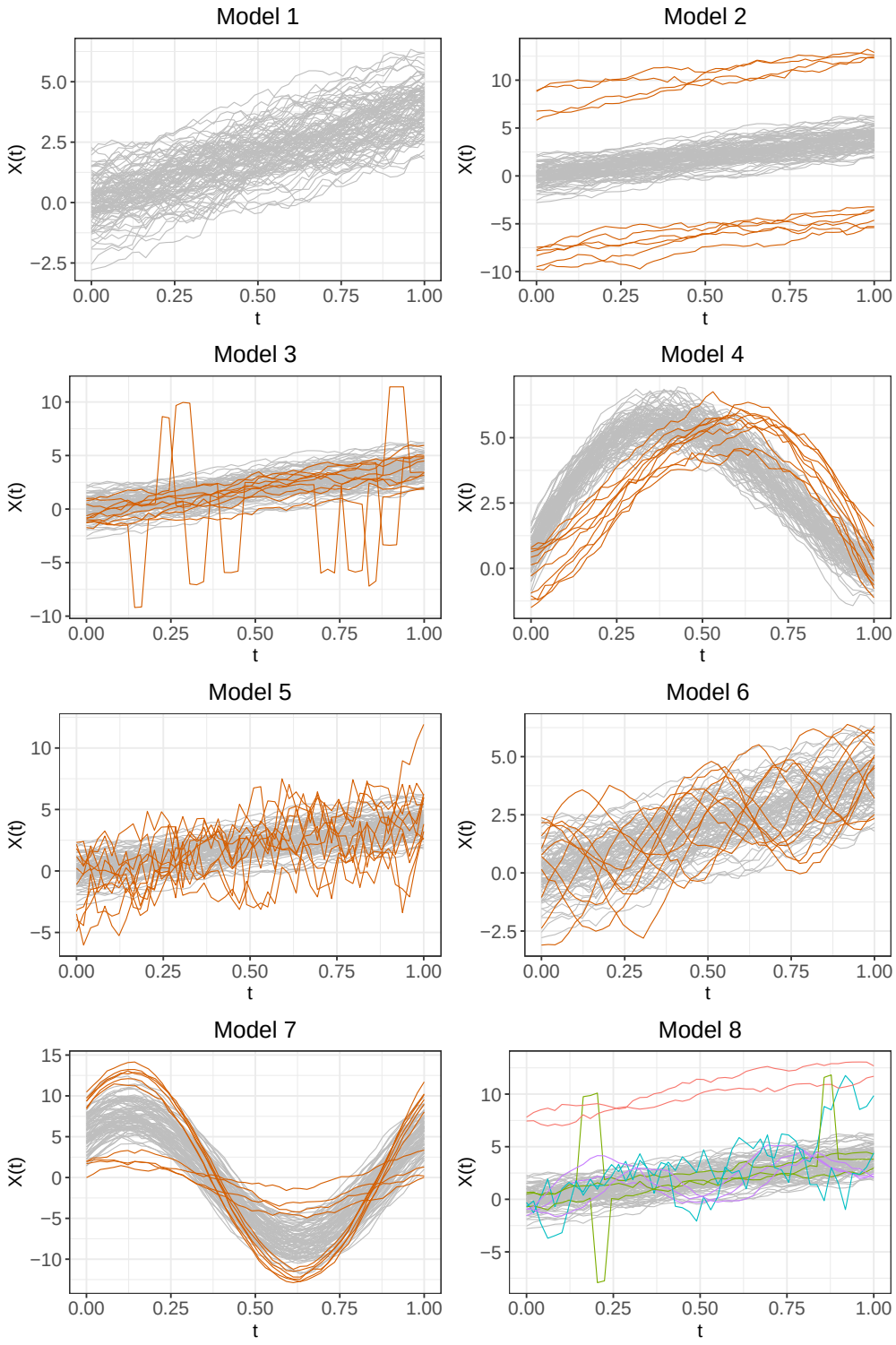

Figure 3.4: Sample data generated by the eight simulation models ($\alpha = 0.10$, $n = 100$ and $d = 50$). Outliers are in color.

proposed methods produce three types of outliers (magnitude, amplitude, and shape), there are different ways to study the performance of the methods. For instance, a user might decide to target only magnitude outliers and discard the other types of outliers (produced by the methods) based on practical background and use case scenario. On the other hand, one might decide to consider all the outliers provided by the proposed methods. Consequently, we will consider the following different possible sets in our comparison:

- FST: This is the union of the different types of outliers flagged by Fast-MUOD. Thus, an observation is an outlier under this scheme if it is either a shape, magnitude or amplitude outlier. In our simulation, we used the pointwise median for Fast-MUOD but the results obtained with the L_1 median are similar (see Section A.1 of Appendix A).
- FSTMG: This considers only the magnitude outliers flagged by Fast-MUOD. Consequently, an observation is an outlier only if it is flagged by Fast-MUOD as a magnitude outlier.
- FSTSH: This considers only the shape outliers flagged by Fast-MUOD. Thus, an observation is an outlier only if it is flagged by Fast-MUOD as a shape outlier.
- FSTAM: This is considers only the amplitude outliers flagged by Fast-MUOD. An observation is an outlier only if it is flagged by Fast-MUOD as an amplitude outlier.
- SF: This is the union of the different types of outliers flagged by Semifast-MUOD (using a random sample whose size is 50% of the size of the original data).
- SF25: This is the union of the different types of outliers flagged by Semifast-MUOD but using a random sample whose size is 25% of the size of the original data.
- MUOD: This is the union of the different types of outliers flagged by MUOD as proposed in Azcorra et al. (2018) (using the "*tangent method*" to determine a cutoff).

Considering the different types of outliers flagged by Fast-MUOD in our performance evaluation gives a clear picture of how the different types of outliers contribute to the overall performance of Fast-MUOD (FST). It is easy to do the same for SF and SF25. However, we do not include these results here but rather their overall performance since the results are quite similar to those of Fast-MUOD. We compare the methods above to the following outlier detection methods for functional data.

- OGMBD: The outliergram method, proposed in Arribas-Gil and Romo (2014), mainly targets shape outliers. It uses a scatter plot of the modified band depth (MBD) and the modified epigraph index (MEI). Outliers are identified by using a boxplot to find the most distant points that lie below the parabola generated by the plot of (MEI, MBD). In addition, outliergram uses the functional boxplot to detect magnitude outliers. Thus, in our evaluation, we consider outliers flagged by both outliergram and functional boxplot.

- MSPLT: MS-plot is based on a directional outlyingness for multivariate functional data proposed by Dai and Genton (2019) and it decomposes the "total directional outlyingness" of sample curves into "magnitude outlyingness" (represented by the "mean directional outlyingness", MO) and "shape outlyingness" (represented by the "variation of directional outlyingness", VO). The MS-Plot is then the scatter plot of $(\text{MO}, \text{VO})^\top$. Outlying curves are identified by computing the squared robust Mahalanobis distance of $(\text{MO}, \text{VO})^\top$ (using the minimum covariance determinant (MCD) algorithm of Rousseeuw and Driessen (1999)), and approximating the distribution of these distances using an F distribution according to Hardin and Rocke (2005). Curves with robust distance greater than a threshold obtained from the tails of the F distribution are flagged as outliers.

- TVD: This method uses the total variation depth (TVD) proposed by Huang and Sun (2019) to compute a "(modified) shape similarity" (MSS) index of the sample functions. A classical boxplot, with the $F_c \times IQR$ cutoff rule (where F_c is the factor), is then applied on the MSS index to detect shape outliers. After removing the shape outliers, a functional boxplot (using TVD to construct the central region) is then applied on the remaining observation in order to detect magnitude outliers. The magnitude outliers are functions outside of 1.5 times the 50% central region with respect to the original sample before the shape outliers were removed. In our simulation study, we used the default value of $F_c = 3$.

- FOM: The functional outlier map, proposed by Rousseeuw et al. (2018), uses a "directional outlyingness" (DO) measure. This measure is then extended to functional data to get the "functional directional outlyingnees" (fDO), computed at the observed points of each function's domain. The variability of the DO values (vDO), is then defined, and the FOM is the scatterplot of (fDO, vDO). To flag observations as outliers, the "combined functional outlyingness" (CFO), based on fDO and vDO, is computed, transformed to logarithm (LCFO), and standardized in a robust way (SLCFO). Any observation with $SLCFO > \Phi^{-1}(.995)$ is then flagged as an outlier, where $\Phi(\cdot)$ is the standard normal cumulative distribution.

- FAO: The functional adjusted outlyingness is similar to FOM above, but uses the "adjusted outlyingness" (AO) proposed by Brys et al. (2005) (see also Hubert and Van der Veeken (2008) and Hubert et al. (2015)) instead of the DO proposed in Rousseeuw et al. (2018). The AO can be extended to functional data to get a functional Adjusted Outlyingness (fAO). The variability (vAO) of the fAO and fAO itself can then be used in a scatterplot to build a functional outlier map as done in FOM above.

- FOM2 and FAO2: Since the functional directional outlyingness and functional adjusted outlyingness can be computed for multivariate functional data, we add the first derivatives of the simulated data as a second dimension to the original data and analyse the obtained bivariate functional data with functional outlier maps of fDO (for FOM2) and fAO (for FAO2).

- ED: The extremal depth notion proposed by Narisetty and Nair (2016) orders functions using a left-tail stochastic ordering of the depth distribution. This depth notion focuses mainly on "extreme outlyingness" just as the name implies, and thus tends to penalize functions with extreme values, even if these extreme values occur in small portions of the domain. In our simulation, we use the extremal depth to construct a central region which is used in a functional boxplot to detect outliers.

- SEQ1, SEQ2 and SEQ3: These methods detect outliers using some standard sets of sequential transformations proposed in Dai et al. (2020). The functional data are sequentially transformed and outliers are removed after each transformation using a functional boxplot based on some depth measure. The first transformation proposed is $\{\mathcal{T}_0(Y_i)\}_{i=1}^n$ and it indicates applying a functional boxplot to the raw data $\{Y_i\}_{i=1}^n$ (to get the $\mathcal{T}_0-$outliers). Other proposed transformation include shifting the curves $\{Y_i\}_{i=1}^n$ to their centers:

$$\mathcal{T}_1(Y_i(t)) = Y_i(t) - \lambda(\mathcal{I})^{-1} \int_{\mathcal{I}} Y_i(t)dt,$$

where $\lambda(\mathcal{I})$ is the Lebesgue measure of $\mathcal{I}$; and normalising the curves $\{Y_i\}_{i=1}^n$ with their L_2 norm:

$$\mathcal{T}_2(Y_i(t)) = Y_i(t)\|Y_i(t)\|_2^{-1},$$

with $\|Y_i(t)\|_2 = \left[\int_{\mathcal{I}}\{Y_i(t)\}^2 dt\right]^{1/2}$. Additional transformations involve taking the first order derivatives of the raw curves (denoted by $\mathcal{D}_1$) and further differentiating the first order derivatives (denoted by $\mathcal{D}_2$).

We consider the following sequence of transformations:

(i) SEQ1 $= \{\mathcal{D}_1 \circ \mathcal{T}_1 \circ \mathcal{T}_0(Y_i)\}_{i=1}^n$,

(ii) SEQ2 $= \{\mathcal{T}_2 \circ \mathcal{T}_1 \circ \mathcal{T}_0(Y_i)\}_{i=1}^n$,

(iii) SEQ3 $= \{\mathcal{D}_2 \circ \mathcal{D}_1 \circ \mathcal{T}_0(Y_i)\}_{i=1}^n$.

We use the distance based L^∞ functional depth (Long and Huang, 2015) for ordering the curves in the intermediate functional boxplots applied after each transformation. We selected L^∞ depth because it had one of the best performance in the simulation study conducted in Dai et al. (2020).

3.4.3 Simulation Results

For each of the Models 2 – 8, we evaluate the true positive rate (TPR), the percentage of correctly identified out of the true outliers, and the false positive positive rate (FPR), the percentage of false positives out of the number of non-outliers. Since Model 1 is a clean model, we present only the FPR under Model 1.

Table 3.1 shows the mean and standard deviation (in parenthesis) of the TPRs and FPRs for all the methods over 500 repetitions. In Model 1, where we have a clean model, MUOD has an exceptionally high FPR of 12.07% mainly because of the aggressive tangent cutoff method which it uses for detecting outliers. FST, SF, and SF25 which use the classical boxplot as a cutoff technique show better FPRs than MUOD. Compared to other functional outlier detection methods, the proposed MUOD-based methods (FST, SF and SF25) show higher FPRs. This is because FST, SF and SF25 are the unions of the three types of outliers flagged by Fast-MUOD, Semifast-MUOD, and Semifast-MUOD with 25% of the sample, respectively. For instance, FST is the union of FSTMG, FSTSH and FSTAM, and consequently, it inherits the FPRs of these individual methods (same applies for SF and SF25). However, considering the individual methods, FSTMG and FSTAM, we see low FPRs. In fact, the overall FPRs of the MUOD-based methods are mainly driven by the FPR of the shape outliers (as seen with FST and FSTSH). This is because the MUOD based methods use a simple Pearson correlation as an index to identify shape outliers and this might be affected to some extent by random noise. However, we find this not be too much of an issue in real life use cases, especially because it is typical for functions to be smoothed (or represented with some basis function with implicit smoothing effect) during the exploratory analysis process (for example, see Section 3.5).

All the methods have very high accuracy for Model 2 where we have magnitude outliers, except for FSTSH and FSTAM which target shape and amplitude outliers respectively. FST, SF and SF25 have higher FPRs for the same reasons explained above

Table 3.1: Mean and Standard Deviation (in parentheses) of the True Positive Rates (TPR) and False Positive Rate (FPR) over eight simulation models with 500 repetitions for each possible case. Each simulation is done with $n = 300$ and $d = 50$ and $\alpha = 0.1$. Comparatively high TPRs are in bold. Proposed methods in italics.

Method	Model 1		Model 2		Model 3		Model 4	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
FST	-	9.90(1.50)	**100.00(0.00)**	8.95(1.59)	**99.81(0.89)**	6.10(1.37)	**100.00(0.00)**	3.15(1.13)
FSTMG	-	1.74(0.92)	**99.99(0.15)**	0.36(0.40)	4.13(3.70)	1.52(0.89)	41.25(9.71)	0.63(0.54)
FSTSH	-	7.94(1.34)	7.79(4.86)	7.94(1.48)	**98.97(2.03)**	4.36(1.12)	**100.00(0.00)**	2.24(0.95)
FSTAM	-	1.70(0.83)	1.66(2.43)	1.69(0.93)	6.41(4.62)	1.38(0.80)	54.56(11.76)	0.45(0.45)
SF	-	9.58(1.54)	**100.00(0.00)**	8.66(1.53)	**99.49(1.49)**	5.60(1.32)	**99.94(0.44)**	2.65(1.02)
SF25	-	9.60(1.51)	**99.99(0.21)**	8.63(1.58)	**99.53(1.29)**	5.59(1.28)	**99.87(0.67)**	2.59(1.03)
MUOD	-	12.07(4.41)	**99.75(3.26)**	8.40(3.78)	56.32(24.44)	10.67(4.95)	95.39(10.02)	3.95(2.83)
OGMBD	-	4.77(1.25)	**100.00(0.00)**	4.65(1.35)	39.43(11.40)	3.45(1.16)	93.54(5.21)	1.22(0.72)
MSPLT	-	3.72(1.41)	**99.97(0.33)**	2.90(1.24)	**100.00(0.00)**	2.95(1.34)	**99.95(0.39)**	1.36(0.84)
TVD	-	0.00(0.03)	**100.00(0.00)**	0.00(0.03)	**100.00(0.00)**	0.00(0.00)	2.77(3.77)	0.00(0.00)
FOM	-	0.53(0.52)	**100.00(0.00)**	0.07(0.17)	47.11(18.59)	0.09(0.19)	48.23(17.44)	0.06(0.15)
FAO	-	0.21(0.34)	**100.00(0.00)**	0.02(0.08)	25.97(16.11)	0.02(0.09)	5.92(8.13)	0.02(0.09)
FOM2	-	3.96(1.13)	**100.00(0.00)**	1.26(0.73)	**100.00(0.00)**	2.02(0.95)	78.98(8.93)	0.78(0.58)
FAO2	-	3.39(1.17)	**100.00(0.00)**	1.41(0.83)	**100.00(0.00)**	1.59(0.89)	32.31(15.17)	0.80(0.62)
ED	-	0.00(0.00)	**99.99(0.15)**	0.00(0.02)	**99.09(1.73)**	0.00(0.00)	0.12(0.62)	0.00(0.00)
SEQ1	-	0.00(0.01)	**99.99(0.15)**	0.00(0.00)	**100.00(0.00)**	0.00(0.00)	10.49(7.41)	0.00(0.00)
SEQ2	-	0.65(0.46)	**99.99(0.15)**	0.68(0.49)	**100.00(0.00)**	0.61(0.48)	29.56(13.57)	0.00(0.00)
SEQ3	-	0.00(0.00)	**99.99(0.15)**	0.00(0.00)	**100.00(0.00)**	0.00(0.00)	5.58(4.72)	0.00(0.00)
Method	Model 5		Model 6		Model 7		Model 8	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
FST	**95.97(4.27)**	5.67(1.19)	**93.05(6.42)**	6.31(1.35)	**79.73(14.95)**	6.55(1.91)	**98.63(2.45)**	6.65(1.40)
FSTMG	15.94(6.70)	1.08(0.71)	0.83(1.70)	1.77(0.94)	1.65(2.36)	1.69(0.90)	30.65(8.10)	1.04(0.75)
FSTSH	86.35(6.74)	4.39(1.12)	**91.01(6.75)**	4.35(1.10)	4.21(3.68)	4.94(1.72)	71.77(7.58)	5.31(1.29)
FSTAM	22.99(7.93)	1.01(0.71)	3.54(3.78)	1.40(0.79)	**79.10(15.42)**	0.01(0.05)	10.74(5.69)	1.29(0.83)
SF	**94.05(5.00)**	5.27(1.22)	**92.46(6.31)**	5.87(1.32)	67.31(17.06)	6.54(1.86)	**98.11(2.69)**	6.19(1.35)
SF25	**93.70(5.44)**	5.26(1.25)	**91.95(6.74)**	5.84(1.25)	66.75(17.67)	6.63(1.92)	**97.85(2.97)**	6.15(1.33)
MUOD	50.16(14.96)	4.35(2.96)	48.60(23.31)	12.01(5.45)	**98.22(5.95)**	18.77(6.15)	64.23(16.18)	4.41(3.25)
OGMBD	**95.99(4.02)**	1.97(0.88)	**99.89(0.60)**	1.88(0.89)	15.40(14.65)	0.00(0.00)	82.99(7.39)	2.87(1.16)
MSPLT	**99.99(0.15)**	2.81(1.21)	**100.00(0.00)**	2.91(1.26)	66.39(16.04)	0.02(0.08)	**99.98(0.26)**	2.85(1.31)
TVD	**100.00(0.00)**	0.00(0.02)	84.25(12.13)	0.00(0.02)	40.88(12.82)	0.00(0.02)	**99.42(1.51)**	0.00(0.00)
FOM	10.84(7.76)	0.09(0.21)	0.02(0.26)	0.08(0.18)	0.64(1.72)	0.00(0.03)	40.85(10.30)	0.10(0.21)
FAO	7.11(6.55)	0.03(0.10)	0.02(0.26)	0.02(0.08)	0.57(1.67)	0.00(0.03)	33.20(9.53)	0.03(0.12)
FOM2	**100.00(0.00)**	2.06(0.94)	71.25(15.41)	1.90(0.86)	30.75(16.49)	0.17(0.25)	**99.01(2.05)**	1.92(0.89)
FAO2	**100.00(0.00)**	1.61(0.83)	54.90(17.75)	1.47(0.83)	12.88(11.47)	0.11(0.22)	**97.06(3.47)**	1.56(0.84)
ED	25.01(9.20)	0.00(0.00)	0.01(0.15)	0.00(0.00)	0.00(0.00)	0.00(0.00)	55.85(8.85)	0.00(0.00)
SEQ1	**100.00(0.00)**	0.00(0.00)	0.17(0.76)	0.00(0.00)	0.00(0.00)	0.00(0.00)	75.55(7.86)	0.00(0.02)
SEQ2	83.91(7.41)	0.57(0.46)	6.90(5.50)	0.60(0.46)	1.61(2.26)	0.00(0.03)	74.68(7.80)	0.60(0.45)
SEQ3	**100.00(0.00)**	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.00)	74.87(7.87)	0.00(0.02)

for Model 1. However, FSTMG has a very low FPR of 0.36%, a value comparable to or better than some other methods. For Models 3 - 8, which contain either shape, amplitude or a mixture of outliers, the proposed MUOD-based methods (FST, SF and SF25) show good outlier detection performance. Considering the individual outliers flagged by FST across these models, the performance of FSTMG, FSTSH and FSTAM vary depending on the type of outliers contained in the model, showing the effectiveness of these individual methods in targeting their specific types of outliers. However, FOM and FAO have very low TPRs for models with shape or amplitude outliers, indicating that they are only well suited to identifying magnitude outliers. FOM2 and FAO2 have high accuracy on Models 3 and 5 as they analyse a bivariate data which includes the first derivative of the simulated data. They however struggle with Models 4, 6 and 7 which contain shape and amplitude outliers. In Model 3, FST shows a very good TPR of 99.81%, mostly buoyed by the outliers detected by FSTSH with its TPR of 98.97%. Furthermore, MSPLT and TVD performed excellently on Model 3. The same applies to ED, with its emphasis on extreme outlyingness of functions, even if such outlyingness is within a small portion of the domain, a property that Model 3 clearly satisfies. OGMBD and MUOD, on the other hand, have very low TPRs for Model 3. For Model 4 however, TVD and ED fail with very low TPRs while MSPLT and FST showed excellent performance. The outliers detected by FSTSH (100.00% TPR) in this model contributed to the overall performance of FST (the same applies for SF and SF25). OGMBD and MUOD also show very good outlier detection performance on Model 4 with 93.54% and 95.39% TPRs respectively. In Model 6, TVD did not have quite as good TPR compared to MSLPT, OGMB and the MUOD-based methods even though this model contains pure shape outliers. The methods based on sequential transformations (SEQ1, SEQ2 and SEQ3) have good outlier detection performance on Models 2, 3 and 5 but they are ineffective on Models 6 and 7 which contain pure shape and amplitude outliers, respectively.

Only FST gives a satisfactory outlier detection performance for Model 7 with its TPR of 79.73%. This model contains pure amplitude outliers and it is especially challenging because the outliers are quite similar in shape and magnitude to the non-outliers. The amplitude outliers detected by FSTAM helped FST to have this satisfactory performance. SF and SF25 on the other hand have low TPRs. Since these two methods use a random sample of the data (50% and 25% of sample size for SF and SF25 respectively), they are not as sensitive as FST which uses only the point-wise (or L_1) median. This proves to be an advantage in detecting outliers that are very similar in shape and magnitude to non-outliers. MUOD, on the other hand has a high TPR but also a very high FPR of 18.34% (caused by the tangent cutoff method) which makes the overall perfor-

mance bad. Finally, the proposed MUOD-based methods show a good outlier detection performance on Model 8 which contains a mixture of pure shape and magnitude outliers. Likewise MSPLT and TVD show good outlier detection TPRs on Model 8, while MUOD and ED have very low TPRs. OGMB, on the other hand, did not have quite as high TPR compared to MSPLT, TVD and the MUOD-based methods.

While we do not claim that the proposed methods are capable of identifying every possible type of outlier, the results of the simulation study has shown that the proposed MUOD-based methods (and especially Fast-MUOD) have a good and well-balanced performance over a wide range of different outlier types, thanks to the fact that they target three different types of outliers simultaneously. Since the outliers identified are also classified into different types (magnitude, amplitude or shape), additional information is provided to the user as to possible reasons why an outlier is indeed flagged as such without the need for manual inspection or data visualization. This will prove valuable when exploring large functional datasets (where visualization is difficult) and also enables selective targeting of different outlier types based on practical background and use case.

3.4.4 Computational Time

A major advantage of the proposed methods is their simplicity despite their effectiveness. Since the indices for Fast-MUOD are very easy to compute, the computational overhead of Fast-MUOD is quite low compared to other existing outlier detection methods. For functional datasets of typical size, this does not matter much. However, a fast method is required in order to handle large (and dense) functional datasets and Fast-MUOD excels in this regard (e.g., see Subsection 3.5.2). While Semifast-MUOD is not as fast compared to Fast-MUOD, it is faster than the original MUOD as described in Azcorra et al. (2018), while achieving better outlier detection performance.

In this section, we focus on comparing the running times of the proposed methods to MUOD and to other existing outlier detection methods used in Subsection 3.4.1. We generated data from Model 2 (in Subsection 3.4.1) with the number of observations $n \in \{10^2, 3 \times 10^2, 10^3, 3 \times 10^3, 10^4, 3 \times 10^4, 10^5\}$. To generate the data, we set $d = 100$ and contamination rate $\alpha = 0.05$. We used the $tictoc$ package in R (Izrailev 2021) to get the running time of each method. For each n, we ran 20 iterations and took the median. The experiment was run on a computer with a Core i9 8950HK processor (6 cores, 12 threads, up to 4.8GHz) with 32GB RAM.

Figure 3.5 shows the results of the running time of the different methods with log-log axes. For small sample size, all the methods have relatively short running time. However, for larger sample sizes, FSTP and FSTL1 (representing Fast-MUOD with the

point-wise median and L_1 median respectively) have the shortest running time taking just about 0.8 and 2.9 seconds, respectively, to process 100,000 observations. ED, FOM, FAO and MSPLOT also show reasonable running times for large number of observations. TVD and OGMBD however are quite slow, requiring over 5 and 8 hours, respectively, to handle 100,000 observations. Consequently, these methods are only suitable for relatively small data (see Subsection 3.5.2 for example). SF and SF25 are much faster than MUOD taking about 7 and 3 minutes, respectively, for 100,000 observations (compared to the 18 minutes required by MUOD). The methods based on sequential transformation have similar running times and hence we show only the results for SEQ1 in Figure 3.5. These methods take about 6 minutes to process 30,000 observations (we could not run the tests up to 100,000 observations on these methods due to memory issues). Finally, FAO2 and FOM2 take about 36 minutes and 33 seconds, respectively, to handle 100,000 observations.

An alternative way to evaluate computational time is to evaluate the maximum number of observations a method can handle within a given set time. Using the same setup as before (simulated data from Model 2, with $d = 100$ and contamination rate $\alpha = 0.05$), we evaluate the maximum number of observation that each method can handle under 10 seconds. Starting from a sample size of 100 to 10,000, we increase the number of observations in steps of 100, while from 20,000 up to 2 million, we increase the sample size in steps of 10,000. Table 3.2 below shows the result. FSTP handles over 1 million observations in less than 10 seconds while FSTL1 handles 290,000 observations under 10 seconds. Given the significant difference in computational time between FSTP and FSTL1 (due to the computation of the L_1 median), we recommend to always use FSTP for large data since the outlier detection performance for both methods are quite similar as mentioned earlier. Compared to other methods, FOM can handle only 270,000 observations, while MSPLOT can handle only 50,000 observations under 10 seconds. As expected, OGMBD and TVD, with their slow running times, can only handle about 1,000 and 2,000 observations respectively under 10 seconds.

The codes for OGMBD and MSPLOT used in this experiment were obtained from the supplementary materials of Arribas-Gil and Romo (2014) and Dai and Genton (2018), respectively, while the implementation of TVD used is at github.com/hhuang90/TVD as stated in Huang and Sun (2019). FOM, FAO, FOM2 and FAO2 were based on codes obtained from wis.kuleuven.be/stat/robust/software while code for ED was obtained from the authors of Narisetty and Nair (2016). SEQ1, SEQ2 and SEQ3 are based on their respective implementations in the **fdaoutlier** R package (Ojo et al., 2021b). Worthy of note is that the implementations of these methods used in this experiment might not be the most optimal version available (or attainable) of the respective methods.

Table 3.2: Number of observations handled under 10 seconds. Simulated data from Model 2, with $d = 100$ and contamination rate $\alpha = 0.05$

Method	Sample size	Time (s)
FSTP	1,060,000	9.89
FSTL1	290,000	9.41
SF	10,000	7.36
SF25	20,000	9.25
MUOD	10,000	9.58
OGMBD	1,000	8.06
MSPLOT	50,000	9.37
TVD	2,000	9.97
FOM	270,000	9.99
FAO	90,000	9.86
FOM2	40000	9.32
FAO2	700	9.86
ED	40,000	9.45
SEQ1	5900	9.86
SEQ2	5900	9.79
SEQ3	5900	9.65

In conclusion, Fast-MUOD provides a huge time performance gain over the original MUOD and Semifast-MUOD, despite its comparable or better outlier detection performance. Semifast-MUOD also provides some gains in running time over the original MUOD but not as much as Fast-MUOD, and its running time still increases with a factor dependent on n^2. All variants of MUOD used in these experiments were run using a single core. Since the MUOD methods have parallel implementations, more performance gains can be obtained by running in parallel with more than one core, especially for Semifast-MUOD.

3.4.5 Sensitivity Analysis

We evaluate the performance of all the methods with increased contamination rate of $\alpha = 0.15$ and $\alpha = 0.2$. The results are presented in Tables A.2 and A.3 (in Section A.2) of the Supplementary Material in Appendix A and they show the outlier detection results on Models 2 - 8. When $\alpha = 0.15$, the proposed methods maintain their good performance on Models 2 and 4 with slight reductions in outlier detection accuracy on Models 3, 5, 6, and 8. The most notable difference is the reduction in average TPRs of the proposed methods on Model 7 (e.g., FST reduces from 79.73% to 41.90%) which is challenging as mentioned earlier (because the outliers are quite similar in shape and magnitude to the mass of the data). Other competing methods suffer some reduction in performance also, including OGMBD on Models 3, 4, 5, and Model 8. The perfor-

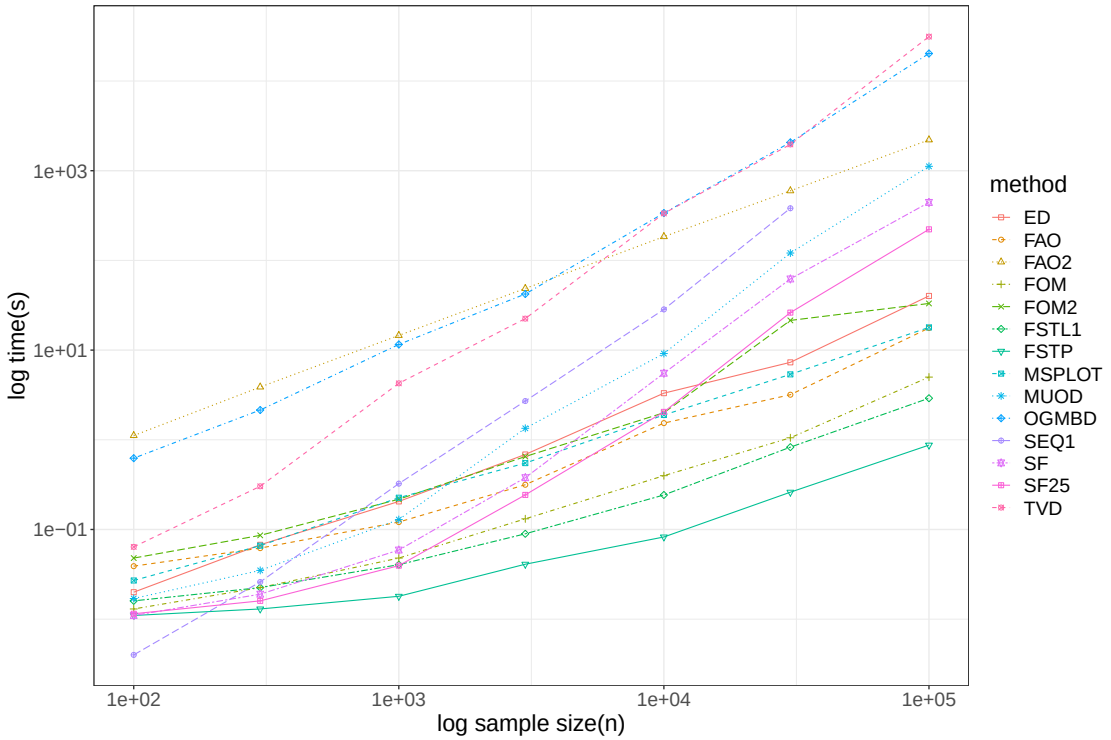

Figure 3.5: Plot of the median computational time of the different outlier detection methods in log-log axes. Each simulation is done with $d = 100$ and $\alpha = 0.05$ with data generated from Model 2. Legend: *FSTP*: Fast-MUOD computed with point-wise median, *FSTL1*: Fast-MUOD computed with the L_1 median.

mance of TVD, FOM2 and FAO2 also reduces on Model 6 which contains pure shape outliers. At $\alpha = 0.2$, the proposed methods still maintain their performance on Models 2 and 4. There is a reduction in performance on Model 8 but the average TPRs of the proposed methods are still quite high at around 90%. The reduction in performance of the methods on Models 3, 5 and 6 (which all contain some form of shape outliers) are more pronounced. The proposed methods break down on Model 7, although other methods also break down on this model. OGMBD, TVD, FOM2 and FAO2 also reduce in outlier detection performance especially on the models with shape outliers. Worthy of mention is MSPLOT which maintains its outlier detection accuracy across the different contamination rates, except on Model 7 on which it did not perform well, even at $\alpha = 0.10$.

In Section A.3 of Appendix A, we evaluate the performance of the proposed methods on lower sample size of $n = 100$ and evaluation points of $d = 25$. The results of this experiment can be found in Table A.4. Except in Model 7, the performance of the proposed models does not change much despite the reduction in sample size and num-

ber of evaluation points. In Model 7, there is a reduction in the accuracy of Fast-MUOD from an average TPR of 79.73% (when $n = 300$ and $d = 50$) to 73.42%. We also notice an increase in the standard deviation of the TPR from 14.95% to 26.24%. This is an indication that the amplitude indices I_A might be less sensitive to outliers with reduced sample size or evaluation points. It is also worthy of note that the TPR of other competing methods like MSPLOT and TVD decreased from 66.39% and 40.88%, respectively, to 48.42% and 27.38%, respectively. The standard deviations of their TPR also increased from 16.04% and 12.82%, respectively, to 27.14% and 24.90%, respectively, in this model.

Section A.5 of Appendix A shows the performance of the proposed methods when the signal-to-noise ratio in the simulated data is increased or decreased. We do this by increasing or decreasing the variance of the simulation models. Using Models 2, 3, 4, and 6, we change the covariance matrix in the base and contamination models to $\gamma(s,t) = \nu \cdot \exp -|t-s|$, where $s,t \in [0,1]$ and $\nu \in \{0.25, 0.5, 1.5, 5\}$. At lower variance levels ($\nu \in \{0.25, 0.5\}$), the TPRs of the proposed methods increased, especially on Model 6, because of the reduced noise in the data. However, at higher variance levels ($\nu \in 1.5, 5$), the proposed methods starts to break down due to the increased noise. This breakdown is also seen in other competing outlier detection methods. When $\nu = 1.5$, all the methods still perform well on Model 2 with magnitude outliers and our proposed methods still maintain a good performance for Model 4. When $\nu = 5$ though, all the methods break down with low TPRs on all the models considered except for TVD on Model 3, FOM2 and FAO2 on Models 2 and 3 and SEQ1 and SEQ3 on Model 3. We refer the reader to Tables A.6 and A.7 of Appendix A for detailed results of the experiment.

3.5 Applications

In this section, we apply the Fast-MUOD method to three scenarios: outlier detection in weather data, object recognition in surveillance video data, and population growth patterns of countries.

3.5.1 Spanish Weather Data

The Spanish weather data collected by the "Agencia de Estatal de Meteorologia" (AEMET) of Spain, contains daily average temperature, precipitation, and wind speed of 73 Spanish weather stations between the period 1980-2009. Geographical information about the location of these stations are also recorded in the data. This dataset is available in the **fda.usc** (Febrero-Bande and de la Fuente 2012) R package and it has been analysed in FDA literature, e.g., Dai and Genton (2018). For this analysis, we use the temperature and log precipitation data. As done in Dai and Genton (2018), we first smooth the data,

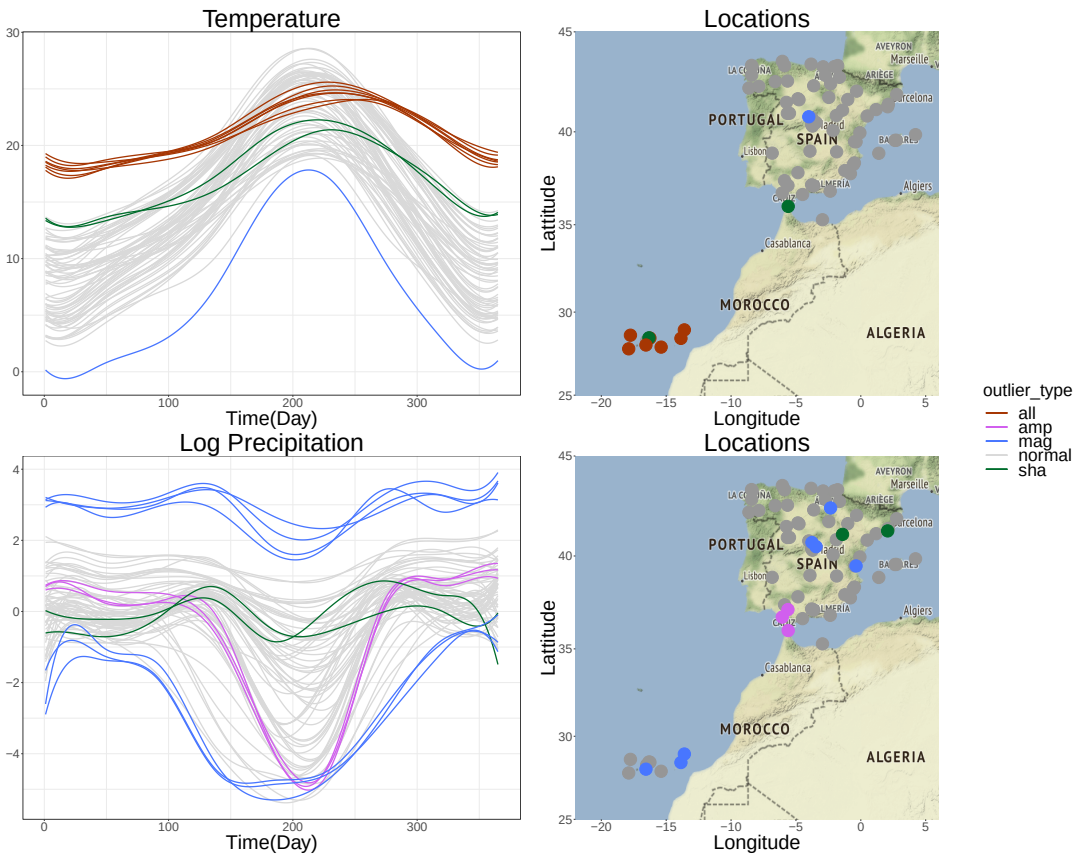

Figure 3.6: Curves flagged as outliers by Fast-MUOD. First Column: smoothed Temperature curves (top), and smoothed Log Precipitation curves (bottom). Second Column: geolocations of weather stations. Legend: curves flagged as magnitude, amplitude and shape outliers (all, in orange), curves flagged as magnitude outliers only (mag, in blue), curves flagged as shape outliers only (sha, in green), curves flagged as amplitude outliers only (amp, in purple), non-outlying curves (normal, in gray).

we then run Fast-MUOD on the smoothed data and collate the different types of outliers flagged for both temperature and log precipitation. The first column of Figure 3.6 shows the different outliers flagged while the second column shows the geographical locations of the flagged outliers.

For temperature, seven weather stations on the Canary Islands are flagged simultaneously as amplitude, shape and magnitude outliers because of the different prevailing weather conditions on this archipelago compared to the other stations located in mainland Spain. Furthermore, two pure shape outliers are flagged, one located on the Canary Islands and the other on the southern tip of Spain, close to the Strait of Gibraltar. The temperature in these regions changes more gradually over the year than in mainland Spain. Finally, a single magnitude outlier is flagged, albeit a lower magnitude one.

This weather station records lower temperatures all through the year compared to the other stations because it is located at a very high altitude in the "Puerto de Navacerrada" mountain pass in the north of Madrid. This station has the highest altitude of all the weather stations in mainland Spain and is known to experience cold temperatures.

For log precipitation, two groups of magnitude outliers are identified, with the first group (of four stations) recording higher precipitation on the average. The second group of three stations are located on the Canary Islands where it is dryer on the average all through the year. A group of pure amplitude outliers, containing 3 stations, is also flagged by Fast-MUOD. These stations experience a more abrupt decline in precipitation during the summer months compared to the more gradual decline in precipitation experienced in other stations located in Spain's interior. These three stations are located in the southern tip of Spain which is known to experience dry summer months. Finally a cluster of pure shape outliers made up of two stations is flagged. The curves of these two stations seem to vary more through the year. One of these station is located in Barcelona, on the eastern coast of Spain which is known to be humid and rainy. The other station is located in Zaragoza, with wet periods during the spring and autumn months.

We compared the results of our analysis to those of MS-plot obtained by Dai and Genton (2018). Even though MS-plot is for multivariate functional data visualization and outlier detection, we chose it because it also handles univariate outlier detection quite well as shown by the results of our simulation studies. In doing this comparison, we combined all the outliers of different types flagged by Fast-MUOD and compared them to those flagged by MS-plot. Figure 3.7 shows the results of both methods.

For temperature, Fast-MUOD and MS-plot both flag as outliers all the weather stations on the Canary Islands and the single station in the south tip of Spain by the Strait of Gibraltar. However, only MS-plot flags the stations in the north of Spain as outliers, while only Fast-MUOD flags as outlier the single lone station in Madrid where significantly lower temperatures are recorded all through the year. Likewise, for log precipitation, both methods flag as outliers the four stations with significantly higher precipitation than the remaining stations. MS-plot flags as outliers all the stations in the southern Canary Islands, while Fast-MUOD flags only some of them, specifically those with the lowest precipitation for most part of the year. Only Fast-MUOD flags as outliers the three stations in the southern tip of Spain where there is a sharper decline in precipitation during the summer months, because of its ability to detect amplitude outliers. Furthermore, two additional stations in Barcelona and Zaragoza, flagged by Fast-MUOD as shape outliers, were not flagged by MS-plot. Even though there is no ground truth as to which stations are outliers or not in this dataset, both Fast-MUOD

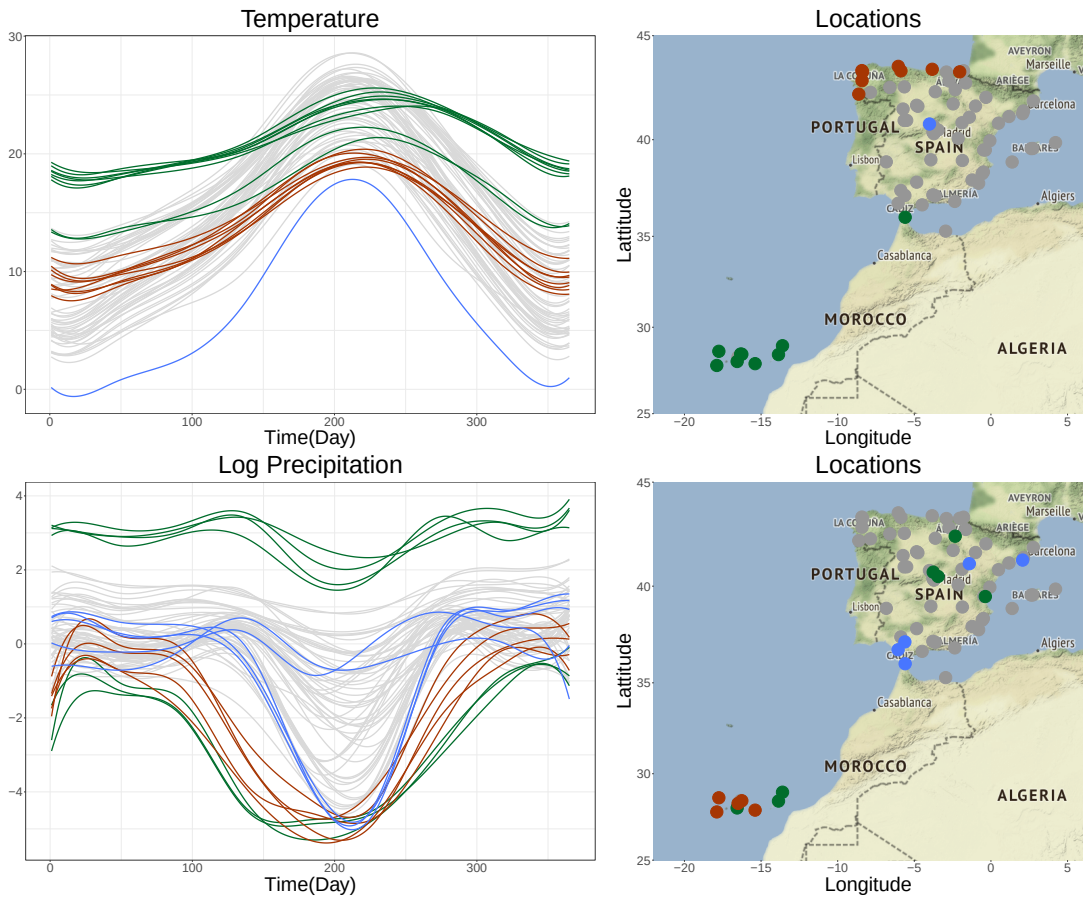

Figure 3.7: Curves flagged as outliers by Fast-MUOD and MS-plot. First column: smoothed Temperature curves (top), and smoothed Log Precipitation curves (bottom). Second column: Geolocations of weather stations. Color code: Curves flagged as outliers by Fast-MUOD and MS-plot (green), curves flagged as outliers by MS-plot only (orange), curves flagged as outliers by Fast-MUOD only (blue).

and MS-plot flagged reasonable outliers and the classification of outliers into types by Fast-MUOD could be an advantage since it is not necessary to visualize the data to know why an observation is an outlier.

3.5.2 Surveillance Video

Next we apply Fast-MUOD on a surveillance video data named *WalkByShop1front*. This video was filmed by a camera across the hallway in a shopping centre in Lisbon. The video is made available online by the CAVIAR project at the link homepages.inf.ed.ac.uk /rbf/CAVIARDATA1. The video shows the front of a clothing store with people walking through the corridor in front of the shop. The video is about 94 seconds long and at different times in the course of the video, five people passed by the front of the shop,

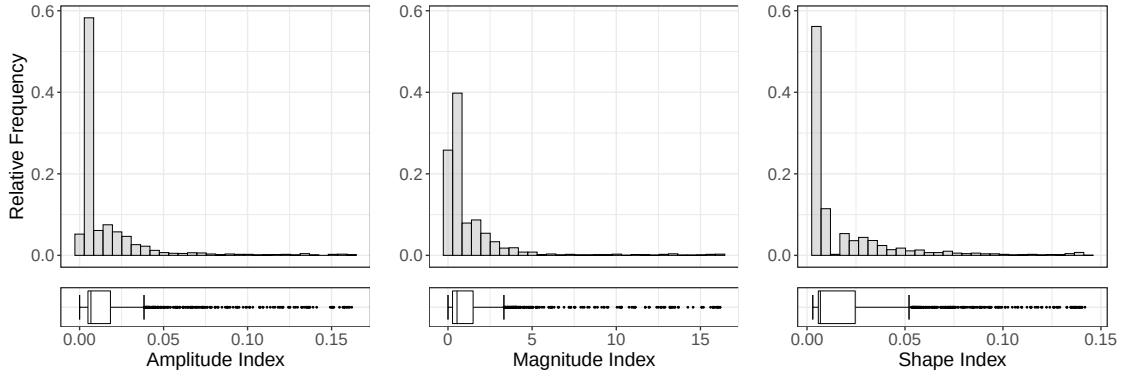

Figure 3.8: Distribution of the amplitude, shape and magnitude indices of the video data.

Figure 3.9: Some outliers detected by the Fast-MUOD from the video.

two of whom entered the store to check clothes in the store. The objective is to use Fast-MUOD to identify points in the video when people passed by the front of the shop.

With each second of the video consisting of 25 frames, the video contains a total of 2359 frames. Each of the frame is made up of $384 \times 288 = 110592$ pixels. We first convert the RGB values of the pixels of the each frame to gray scale. From the matrix of gray intensity values of each frame, we form a row vector of length 110592 by ordering the values of the matrix column-wise (we obtained the same result by arranging row-wise). The constructed functional data is made up of 2359 curves observed at 110592 points.

We then apply Fast-MUOD on the constructed functional data using the point-wise median to speed up computation. Figure 3.8 shows the histogram and the boxplot of magnitude, amplitude and shape indices from this data. We obtained 216 shape outliers, 206 amplitude outliers and 194 magnitude outliers. The three types of outliers flagged are not mutually exclusive as shape outliers for instance can also have partial magnitude outlyingness. There are 125 outliers that are outliers of the three types, i.e., they are flagged simultaneously as magnitude, amplitude and shape outliers. There are only 34 pure magnitude outliers, 15 pure amplitude outliers and 48 pure shape outliers. In total, there are 294 unique frames flagged as outliers.

All the 294 outlying frames correspond to time points in the video when people passed by the front of the store, thus there are no false positives. For instance, frames 831 - 846 and 885 - 887 correspond to the period when the first person in the video passed by the front of the store (see Figure 3.9, Frame 837). The same for frames 1614 - 1642 when the second person passed by and entered the shop (Figure 3.9, Frame 1625). Frames 1852 - 1983 correspond to when two women passed by together in front of the shop (Figure 3.9, Frame 1900) and frames 2112 - 2169 and 2296 - 2336 which correspond to the period when the last person passed by and entered the shop (Figure 3.9, Frame 2130). There are small pockets of time periods (frames) in the video (which are in between the frames flagged as outliers), that contain people but are not flagged as outliers. We notice that these usually corresponds to time periods where there is not a enough contrast (from the gray intensities) between the person in the video and the environment because we converted the frames to gray scale before analysis. This is seen in Frame 2110 (Figure 3.10) for instance, when a man wearing a dark blue and red shirt with dark trousers was standing entirely behind a dark pillar or in Frame 2295 (Figure 3.10) when the same man was standing beside dark clothes in the store.

Examining the outliers of each type also provide some additional insight. The 34 pure magnitude outliers are frames which contain the man wearing a dark clothes as he entered into the shop, e.g., Frame 2166 (Figure 3.10). The gray intensities (and hence contrast with the environment) at this time is very high due to the dark nature of his clothing and hence the reason why these frames are pure magnitude outliers. The 48 pure shape outliers correspond to frames that contain the two women passing by together in front of the shop, e.g., Frame 1914 in Figure 3.10. The 15 pure amplitude outliers are frames where people just entered or are about to exit the field of view of the camera (see Frames 887 and 2112). Thus, in addition to detecting outlying frames, the classification of the different frames also give some insight which might prove valuable in different use cases.

Huang and Sun (2019) and Rousseeuw et al. (2018) applied their methods on video data. For comparison, we run their methods (TVD and FOM) and MSPLT on the constructed functional data obtained from the surveilance video. While MSPLT did not produce any result due to some computational error, TVD flagged all the frames as outliers after running for over 9 hours (compared to about 42 seconds of Fast-MUOD). FOM however performed excellently, flagging only frames that include people passing by the shop and producing the result in a reasonable time of 193 seconds.

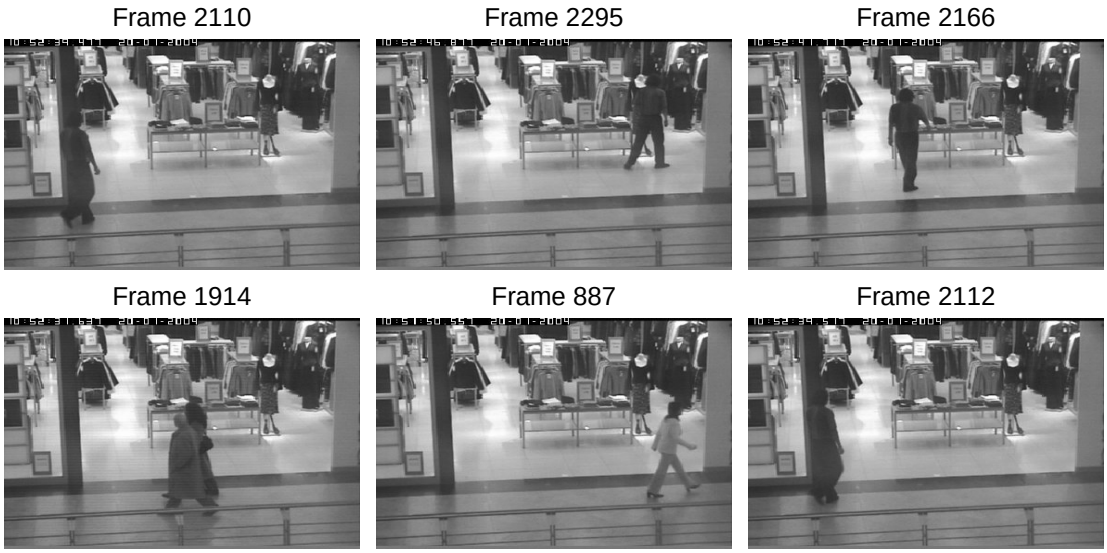

Figure 3.10: Selected frames from the video in gray scale. Frames 2110 and 2295: frames not detected as outliers. Frame 2166: sample pure magnitude outlier. Frame 1914: sample pure shape outlier. Frame 887 and 2112: sample pure amplitude outliers.

3.5.3 Population Data

Finally, we analyse the world population data from the United Nations, also analysed by Nagy et al. (2017) and Dai et al. (2020). This data contains the yearly total population of 233 countries (and autonomous regions) recorded in the month of July, 1950 to 2015. Following Nagy et al. (2017) and Dai et al. (2020), we select only countries with population between one million and fifteen million in 1980 which leaves us with 105 countries out of the total 233 countries. The constructed functional data is then made up of 105 curves observed at 65 points. We apply Fast-MUOD and the countries/regions detected as outliers are shown in Table 3.3.

In total there are 33 unique countries detected as outliers; 3 of them magnitude outliers, 15 of them amplitude outliers and 18 of them shape outliers. Again, the types of outliers are not mutually exclusive as all magnitude outliers are also amplitude outliers. Saudi Arabia, Sudan and Uganda are flagged as magnitude outliers because they had highest population values toward the end of the period of the data (2015), as can be seen in Figure 3.11. Sudan, despite having a population of about 5 million in 1950, had the highest population value of 40 million 2015. The same trend is observed for Uganda and Saudi Arabia, with population of 39 million and 31 million respectively in 2015. The amplitude outliers are shown in the top-right panel of Figure 3.11. These are countries with very high population growth rate in the period of the data. Among these are Sudan, Uganda and Iraq, as they had an increase of 34 million, 33 million and

Table 3.3: Countries detected as outliers by Fast-MUOD

Magnitude outliers	Amplitude outliers	Shape outliers
Saudi Arabia, Sudan, Uganda	Sudan, Uganda, Saudi Arabia, Iraq, Malaysia, Yemen, Afghanistan, Ghana, Nepal, Côte d'Ivoire, Mozambique, Madagascar, Angola, Syrian Arab Republic, Cameroon	Bulgaria, Latvia, Hungary, Georgia, Croatia, Estonia, Lithuania, Bosnia and Herzegovina, Belarus, Armenia, Serbia, Republic of Moldova, Kazakhstan, Albania, Czech Republic, United Arab Emirates, TFYR Macedonia, Slovakia

30 million respectively between 1950 and 2015. Other countries include Saudi Arabia, Afghanistan and Malaysia. All the countries flagged as amplitude outliers are either located in the Middle East or Africa.

Finally, the shape outliers are shown in the bottom-left panel of Figure 3.11. The curves of these countries show a different shape and trend compared to the other countries. One observation about these countries is their peculiar pattern of a slight increase in population growth till 1980 followed by a plateau or slight decrease in the population till the end of the study period. There are also few countries with a sharp increase or decrease in population. Furthermore, all these countries except for United Arab Emirates (UAE) are located in Central and Eastern Europe with similar demographics, geographical, economic and political environment.

Compared to the results obtained in Nagy et al. (2017), our method identifies more outliers. For instance, the first order outliers (which are equivalent to magnitude outliers) identified in Nagy et al. (2017) did not include Sudan which had the highest population by the end of the investigated period (2010). Though Sudan was flagged as a second order outliers, a lot of countries in Eastern Europe flagged as shape outliers were not flagged as outliers. Except for Netherlands, all the second and third order outliers flagged by Nagy et al. (2017) are also flagged as either shape outliers or amplitude outliers by Fast-MUOD. Furthermore, our classification of the different outliers provide additional information and consistent interpretation on why observations are flagged as outliers.

Dai et al. (2020) also analysed this data and classified outliers found using sequential transformations. All the "pattern" outliers found are countries in Eastern Europe except for Rwanda just like the shape outliers flagged by Fast-MUOD. In fact, our me-

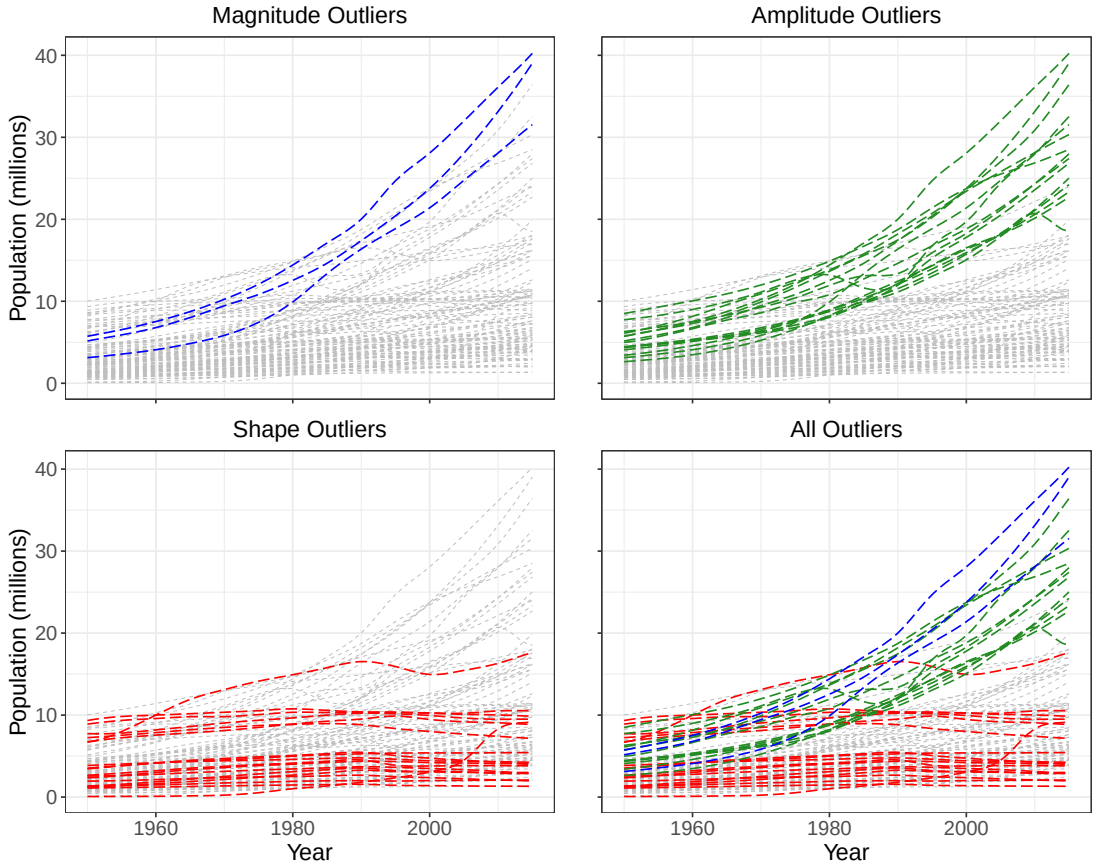

Figure 3.11: Outliers detected by Fast-MUOD from the population data. Top-left: Magnitude outliers. Top-right: Amplitude outliers. Bottom-left: Shape outliers. Bottom-right: All the outliers.

thod flagged all the pattern outliers found by Dai et al. (2020) as shape outliers except for Rwanda. However, Fast-MUOD flags five additional shape outliers including Macedonia, Serbia, Albania, and Slovakia, all located in Central and Eastern Europe. The amplitude outliers flagged by Fast-MUOD also include all the amplitude outliers flagged by the method described in Dai et al. (2020) and all these countries are located in the Middle East and Africa too. While Fast-MUOD flagged only three magnitude outliers compared to nine magnitude outliers flagged by Dai et al. (2020), the remaining six magnitude outliers were flagged by Fast-MUOD as amplitude outliers. In fact, some of these six magnitude outliers were flagged as both magnitude and amplitude outliers by Dai et al. (2020), but they were grouped as magnitude outliers in order to maintain a mutually exclusive classification of outliers (see Table 5 in Dai et al. (2020) for details).

Overall, our results are quite consistent with those obtained by Dai et al. (2020) even though there are slight differences in classification of some outliers. Also worthy of note

is the slight difference in investigated period (1950 - 2010) in the analysis by Dai et al. (2020) and Nagy et al. (2017).

3.6 Discussion

In this chapter, we have proposed two methods based on the MUOD outlier detection method. These methods use a sample of the data to compute the indices for Semifast-MOUD, or a median (L_1 or point-wise) in the case of Fast-MUOD and they improve on the scalability and outlier detection performance of MUOD. In separating the outlier indices from the indices of the typical observations, we use the classical boxplot. All these put together make the proposed methods intuitive and based on simple statistical concepts, consequently making them less computationally intensive. Different types of outliers are identified and classified directly, giving an intuition as to why a curve is flagged as an outlier without the need for visualization or manual inspection. This is valuable in cases where manual inspection or visualizing the data is difficult.

Using both simulated and real data, we have shown the performance benefits of these methods over MUOD. Further comparisons to existing univariate functional outlier detection tools shows comparable or superior results in correctly identifying potential outliers of different types. Implementation is done in R and the code is made available at https://github.com/otsegun/fastmuod.

Possible further improvement is extension of the methods to multivariate functional data (see Chapter 5). The use of orthorgonal regression in the computation of the indices is interesting to study. Exploring the theoretical properties of the MUOD indices is also a possible next line of investigation (see Chapter 4).

Chapter 4

Properties of the Fast-MUOD Indices

This chapter is based on:

Ojo, O. T., Fernández Anta, A., Genton, M. G., & Lillo, R. E. (2022). "Multivariate Functional Outlier Detection using the FastMUOD Indices". arXiv:2207.12803

4.1 Introduction

We consider the problem of detecting outliers in a collection of multivariate functional observations. In particular, we consider observations of the form: $\{\boldsymbol{Y}_i(t), t \in \mathcal{I}\}_{i=1}^n$, wherein a vector $\boldsymbol{Y}_i(t) \in \mathbb{R}^d$, $d \in \mathbb{N}$, is observed at a domain point t in the interval $\mathcal{I}$. Such vector-valued functional observations are increasingly observed in real-life studies and various physical and environmental applications. Thus, exploratory methods for multivariate functional data have been recently garnering considerable interest.

Outlier detection (OD), a part of the exploratory data analysis process, involves identifying observations that differ from the bulk of the data, either because they come from a different distribution compared with the bulk or because they lie at the extremes of the distribution of the data. However, identifying outliers is more complicated when observations are functions observed on a domain, i.e., functional data. Functional observations demonstrate different outlying behaviours, e.g., a vertical shift, compared to the bulk of the data (magnitude outliers) or a horizontal shift, in which case the outlying function is not well aligned with the bulk of the data. Functional outliers can also have different shapes or follow different paths compared to the bulk of the data. Hubert et al. (2015) proposed a taxonomy for different types of functional outliers based on the dif-

ferent outlying behaviours they exhibit, and whether such behaviours can be observed in a small part of the domain or throughout the domain; (see also Dai et al., 2020).

To identify outliers among multivariate (non-functional) observations (i.e., vector observations $\boldsymbol{X} \in \mathbb{R}^d$), it is typical to order the observations, from the center outward using a notion of *statistical depth*. Then, the observations having the lowest depth values can be closely examined for outlying behaviours. This procedure is convenient because most depth notions are non-parametric and they do not require any assumption concerning the underlying data distribution.

The approach mentioned above has also caught on in the analysis of functional observations, where several OD methods are based on notions of functional depths. For example, the *functional boxplot* (Sun and Genton, 2011) uses the modified band depth (López-Pintado and Romo, 2009) to order functional observations and define a 50% central region. Then, the outliers are functions that lie outside of the central region inflated by 1.5, similar to the classical boxplot. Other proposals around this theme include Sguera et al. (2015) and Febrero et al. (2008), where functional depth measures were used for OD.

On the other hand, several functional OD methods are based on "custom-built" outlyingness indices, metrics, or pseudo-depths directly targeted toward OD, instead of ordering (as with functional depth notions). Examples along this line include the *magnitude-shape plot* (MS-plot) (Dai and Genton, 2018), based on the directional outlyingness proposed by Dai and Genton (2019); the *functional outlier map* (FOM), based on another (functional) directional outlyingness proposed by Rousseeuw et al. (2018); the *modified shape similarity* index (MSS) proposed in Huang and Sun (2019); the (robustified) *functional tangential angle* (rFUNTA) proposed in Kuhnt and Rehage (2016), and an earlier proposal of Hubert et al. (2015) in which the *bag distance* and *skewness adjusted projection depth* were proposed for functional OD.

Finally, certain functional OD procedures are based on either a combination of depth notions and outlyingness indices, or the use of more primitive methods (such as dimension reduction or transformation). Some of these include the *outliergram*, based on the modified epigraph index (López-Pintado and Romo, 2011) and the modified band depth; the *functional bagplots*, and the *functional highest density regions* (Hyndman and Shang, 2010), both using the first two robust principal components of the functional data to construct plots used for detecting functional outliers. Likewise, Dai et al. (2020) proposed detecting functional outliers using a sequence of (functional) data transformations, each followed by a functional boxplot to detect different types of outliers. Recently, Herrmann and Scheipl (2021) proposed using multidimensional scaling (Cox and Cox, 2008) to reduce functional data to lower dimensional embeddings. Then, an

OD method such as the local outlier factors (Breunig et al., 2000) was applied on the embeddings to detect outlying curves.

Fast Massive Unsupervised Outlier Detection (Fast-MUOD), introduced by Ojo et al. (2021a), belongs to the second group of functional OD methods (outlined above) because it uses three indices, each targeting different outlying behaviours that functional outliers may exhibit. The Fast-MUOD indices are the *magnitude index*, which targets magnitude outliers; the *shape index*, which targets shape outliers; and the *amplitude index*, which targets amplitude outliers. Because these indices target different outlier types, the outliers identified are also classified as per their types, unsupervised, without the need for inspection or visualisation of the data. The method is fast and simple, making it scalable to (and suitable for) "big" functional data analysis.

Nevertheless, despite its advantages, Fast-MUOD has its limitations. First, its indices are designed for univariate functional data. Second, it is not exactly clear from Ojo et al. (2021a) why the Fast-MUOD indices are suitable for OD from a theoretical perspective, despite the good and scalable performance observed on simulated and real datasets. In this chapter, we aim to we explore the properties of the Fast-MUOD indices rigorously; and in Chapter 5, we extend the Fast-MUOD indices to outlier detection in multivariate functional data.

4.2 Definitions and Properties of the Fast-MUOD Indices

We present the sample and population definitions of the Fast-MUOD indices and explore their properties. First, we describe the notations used in this article. We assume that functions are defined on the unit interval $[0,1]$ and denote by $L^2([0,1])$, the space of all square-integrable functions defined over $[0,1]$. We denote by $\langle f, g\rangle$ (unless otherwise specified), the inner product of two functions $f, g \in L^2([0,1])$. The norm of $f \in L^2([0,1])$ induced by this inner product is denoted by $\|f\|$.

4.2.1 Definitions of the Univariate Fast-MUOD Indices

Definition 4.1 (Definitions of Fast-MUOD indices)**.** Let X be a stochastic process in $L^2([0,1])$ with distribution F_X and $\mu(t) = \mathbb{E}[X(t)]$ be its population mean function. We define the **shape index** of a function $y \in L^2([0,1])$ (which may be a realization of X) with respect to (w.r.t.) F_X as

$$I_S(y, F_X) := 1 - \frac{\int \tilde{y}(t)\tilde{\mu}(t)dt}{\left[\int \tilde{y}(t)^2 dt\right]^{1/2}\left[\int \tilde{\mu}(t)^2 dt\right]^{1/2}} = 1 - \frac{\langle \tilde{y}, \tilde{\mu}\rangle}{\|\tilde{y}\| \cdot \|\tilde{\mu}\|},$$

where $\tilde{y}(t)$ and $\tilde{\mu}(t)$ denote the centered curves given by: $\tilde{y}(t) \coloneqq y(t) - \int y(r)dr$, and $\tilde{\mu}(t) \coloneqq \mu(t) - \int \mu(r)dr$, respectively. We define the **amplitude index** of y w.r.t. F_X as

$$I_A(y, F_X) \coloneqq \frac{\int \tilde{y}(t)\tilde{\mu}(t)dt}{\int \tilde{\mu}(t)^2 dt} - 1 = \frac{\langle \tilde{y}, \tilde{\mu} \rangle}{\|\tilde{\mu}\|^2} - 1.$$

Finally, we define the **magnitude index** of a function y w.r.t. F_X as

$$I_M(y, F_X) \coloneqq \int y(t)dt - \beta(y) \int \mu(t)dt,$$

where $\beta(y) = I_A(y, F_X) + 1$.

In practice, functions are usually observed on a finite number of points in the domain. In this case, an approximation to the Fast-MUOD indices can be obtained by replacing the integral with a summation, yielding the following trivial definitions, which we include for completeness.

Definition 4.2 (Finite-dimensional approximation of Fast-MUOD indices)**.** Suppose the function y is observed on the finite points $T = \{t_1 = 0, t_2, \ldots, t_k = 1\} \subset [0, 1]$ with $t_j - t_{j-1} = \Delta$, a constant. Moreover, let $\mu(t_j) = \mathbb{E}[X(t_j)]$ for all $t_j \in T$. Then, we define the finite-dimensional version of the shape index of y w.r.t F_X as:

$$I_{S_k}(y, F_X) \coloneqq 1 - \frac{\sum_{j=1}^k \tilde{y}(t_j)\tilde{\mu}(t_j)}{\left[\sum_{j=1}^k \tilde{y}(t_j)^2\right]^{1/2} \left[\sum_{j=1}^k \tilde{\mu}(t_j)^2\right]^{1/2}},$$

where $\tilde{y}(t_j)$ and $\tilde{\mu}(t_j)$ denote the centered functions: $\tilde{y}(t_j) \coloneqq y(t_j) - \frac{1}{k}\sum_{j=1}^k y(t_j)$, and $\tilde{\mu}(t_j) \coloneqq \mu(t_j) - \frac{1}{k}\sum_{j=1}^k \mu(t_j)$, respectively. The finite-dimensional versions of the amplitude and magnitude indices are, respectively, defined as:

$$I_{A_k}(y, F_X) \coloneqq \frac{\sum_{j=1}^k \tilde{y}(t_j)\tilde{\mu}(t_j)}{\sum_{j=1}^k \tilde{\mu}(t_j)^2} - 1,$$

and

$$I_{M_k}(y, F_X) \coloneqq \left(\frac{1}{k}\sum_{j=1}^k y(t_j)\right) - \beta_k(y) \left(\frac{1}{k}\sum_{j=1}^k \mu(t_j)\right),$$

with $\beta_k(y) = I_{A_k}(y, F_X) + 1$.

Proposition 4.1 (Convergence of the finite-dimensional approximation of Fast-MUOD indices)**.** *For a stochastic process $X(t) \in L^2([0, 1])$ and a function y observed on the finite points $T = \{t_1, t_2, \ldots, t_k\} \subset [0, 1]$ with $t_j - t_{j-1} = \Delta$, a constant, the indices $I_{S_k}(y, F_X)$,*

$I_{A_k}(y, F_X)$, and $I_{M_k}(y, F_X)$ converge (in limit) to $I_S(y, F_X)$, $I_A(y, F_X)$, and $I_M(y, F_X)$ respectively, as $k \longrightarrow \infty$ (and $\Delta \longrightarrow 0$).

Proof. The proof follows from the definition of the Riemann integral. □

Depending on whether the realizations of X are continuously or discretely sampled in time, the sample versions of the indices can be defined by replacing the mean function, μ, of X with an appropriate empirical estimate. Suppose that $X_1(t), \ldots, X_n(t)$ are independent and identically distributed (iid) realizations from $X(t)$, the (pointwise) sample mean function, which we will denote by $\bar{X}(t)$, and given by $\bar{X}(t) = n^{-1} \sum_{i=1}^{n} X_i(t)$, is an estimate of the mean function $\mu(t)$. This leads to a direct proposal of the following definitions for the sample versions of the Fast-MUOD indices.

Definition 4.3 (Sample version of Fast-MUOD indices)**.** Let $X_1(t), \ldots, X_n(t)$ be iid realizations of the stochastic process $X \in L^2([0,1])$, with empirical distribution F_{X_n}. We define the sample shape index as:

$$I_{S_n}(y, F_{X_n}) \coloneqq 1 - \frac{\int \tilde{y}(t)\tilde{\bar{X}}(t)dt}{\left[\int \tilde{y}(t)^2 dt\right]^{1/2} \left[\int \tilde{\bar{X}}(t)^2 dt\right]^{1/2}} = 1 - \frac{\langle \tilde{y}, \tilde{\bar{X}} \rangle}{\|\tilde{y}\| \cdot \|\tilde{\bar{X}}\|},$$

where $\tilde{\bar{X}}(t)$ denotes the centered sample mean function: $\tilde{\bar{X}}(t) \coloneqq \bar{X}(t) - \int \bar{X}(t)dt$. The sample amplitude and magnitude indices are then defined as:

$$I_{A_n}(y, F_{X_n}) \coloneqq \frac{\int \tilde{y}(t)\tilde{\bar{X}}(t)dt}{\int \tilde{\bar{X}}(t)^2 dt} - 1 = \frac{\langle \tilde{y}, \tilde{\bar{X}} \rangle}{\|\tilde{\bar{X}}\|^2} - 1,$$

and

$$I_{M_n}(y) \coloneqq \int y(t)dt - \beta_n(y) \int \bar{X}(t)dt,$$

with $\beta_n(y) = I_{A_n}(y, F_{X_n}) + 1$.

Definition 4.4 (Finite sample version of Fast-MUOD indices)**.** Let $X_1(t), \ldots, X_n(t)$ be iid realizations of X and let each X_i be observed on finite points $T = \{t_1, t_2, \ldots, t_k\} \subset [0,1]$ where $t_j - t_{j-1} = \Delta$, a constant. For a function y observed on the same set of domain points T, we define an approximation to the sample version of the shape index as:

$$I_{S_{n,k}}(y, F_{X_{n,k}}) \coloneqq 1 - \frac{\sum_{j=1}^{k} \tilde{y}(t_j)\tilde{\bar{X}}(t_j)}{\left[\sum_{j=1}^{k} \tilde{y}(t_j)^2\right]^{1/2} \left[\sum_{j=1}^{k} \tilde{\bar{X}}(t_j)^2\right]^{1/2}} = 1 - \hat{\rho}(y, \bar{X}),$$

where $\hat{\rho}(y, \bar{X})$ denotes the sample Pearson correlation coefficient between the observed points of the sample mean function $\bar{X}$ and y. Approximations of the sample amplitude and magnitude indices are defined as:

$$I_{A_{n,k}}(y, F_{X_{n,k}}) \coloneqq \frac{\sum_{j=1}^{k} \tilde{y}(t_j)\tilde{\bar{X}}(t_j)}{\sum_{j=1}^{k} \tilde{\bar{X}}(t_j)^2} - 1,$$

and

$$I_{M_{n,k}}(y, F_{X_{n,k}}) \coloneqq \left(\frac{1}{k}\sum_{j=1}^{k} y(t_j)\right) - \beta_{n,k}(y)\left(\frac{1}{k}\sum_{j=1}^{k} \bar{X}(t_j)\right),$$

where $\beta_{n,k}(y) = I_{A_{n,k}}(y, F_{X_{n,k}}) + 1$.

Proposition 4.2 (L^2-consistency of the sample Fast-MUOD indices)**.** *Let $X \in L^2([0,1])$ be a stochastic process. Then for another function $y \in L^2([0,1])$, $I_{S_n}(y, F_{X_n})$, $I_{A_n}(y, F_{X_n})$, and $I_{M_n}(y, F_{X_n})$, are L^2-consistent estimators of $I_S(y, F_X)$, $I_A(y, F_X)$, and $I_M(y, F_X)$, respectively.*

Proof. The pointwise sample mean function $\bar{X}(t)$ is an L^2-consistent estimator of the mean function $\mu = \mathbb{E}[X(t)]$, i.e., $\|\mu - \bar{X}\| \xrightarrow{P} 0$, where $\|X\| = \left(\int X(t)^2 dt\right)^{1/2}$ is the L^2-norm, and $\xrightarrow{P}$ indicates convergence in probability (Kokoszka and Reimherr, 2017). Thus, $\bar{X}(t) \xrightarrow{P} \mu(t)$, as $n \to \infty$. By the continuous mapping theorem, then:

$$\int_0^1 \bar{X}(t)dt \xrightarrow{P} \int_0^1 \mu(t)dt$$

and

$$\left[\bar{X}(t) - \int_0^1 \bar{X}(s)ds\right] \xrightarrow{P} \left[\mu(t) - \int_0^1 \mu(s)ds\right].$$

Now for a given $y \in L^2([0,1])$, which may or may not be a realization of X, define:

$$\tilde{y}(t) \coloneqq y(t) - \int y(s)ds.$$

By the continuous mapping theorem again,

$$\tilde{y}(t)\left[\bar{X}(t) - \int_0^1 \bar{X}(s)ds\right] \xrightarrow{P} \tilde{y}(t)\left[\mu(t) - \int_0^1 \mu(s)ds\right],$$

$$\int \tilde{y}(t)\left[\bar{X}(t) - \int_0^1 \bar{X}(s)ds\right] dt \xrightarrow{P} \int \tilde{y}(t)\left[\mu(t) - \int_0^1 \mu(s)ds\right] dt. \tag{4.1}$$

Using similar arguments, since $\bar{X}(t) \xrightarrow{P} \mu(t)$ as $n \to \infty$,

$$\left[\bar{X}(t) - \int_0^1 \bar{X}(s)ds\right]^2 \xrightarrow{P} \left[\mu(t) - \int_0^1 \mu(s)ds\right]^2,$$

and thus,

$$\left(\int \left[\bar{X}(t) - \int_0^1 \bar{X}(s)ds\right]^2 dt\right)^{1/2} \xrightarrow{P} \left(\int \left[\mu(t) - \int_0^1 \mu(s)ds\right]^2 dt\right)^{1/2}.$$

So,

$$\left(\int \tilde{y}(t)^2 dt\right)^{1/2} \left(\int \tilde{\bar{X}}(t)^2 dt\right)^{1/2} \xrightarrow{P} \left(\int \tilde{y}(t)^2 dt\right)^{1/2} \left(\int \tilde{\mu}(t)^2 dt\right)^{1/2}, \tag{4.2}$$

where:

$$\tilde{\bar{X}}(t) = \bar{X}(t) - \int_0^1 \bar{X}(s)ds \quad \text{and} \quad \tilde{\mu}(t) = \mu(t) - \int_0^1 \mu(s)ds.$$

If $\tilde{y}(t) \neq 0$ and $\tilde{\mu}(t) \neq 0$ almost surely, for all $t \in [0,1]$, then from Equation (4.1) and (4.2):

$$I_{S_n}(y) = 1 - \frac{\int \tilde{y}(t)\tilde{\bar{X}}(t)dt}{\left(\int \tilde{y}(t)^2 dt\right)^{1/2} \left(\int \tilde{\bar{X}}(t)^2 dt\right)^{1/2}} \xrightarrow{P} 1 - \frac{\int \tilde{y}(t)\tilde{\mu}(t)dt}{\left(\int \tilde{y}(t)^2 dt\right)^{1/2} \left(\int \tilde{\mu}(t)^2 dt\right)^{1/2}} = I_S(y)$$

Using similar arguments, we see that

$$I_{A_n}(y) \xrightarrow{P} I_A(y),$$

and

$$I_{M_n}(y) \xrightarrow{P} I_M(y).$$

□

4.2.2 Properties of the Univariate Fast-MUOD Indices

In this subsection, we present some results showing how the Fast-MUOD indices behave under certain simple transformations and why this behaviour makes them ideal for detecting outliers.

Proposition 4.3 (Properties of Fast-MUOD indices)**.** *Let X be a stochastic process in $L^2([0,1])$ with distribution F_X and mean function $\mu(t)$. Let y and z be other functions in $L^2([0,1])$ (which may be realizations of X) and let $a, b \in \mathbb{R}$. Then, the following statements hold.*

(i) For a new function $y'(t) = ay(t) + b$*, we have*

(a) $I_M(y', F_X) = aI_M(y, F_X) + b$*;*

(b) $I_A(y', F_X) = aI_A(y, F_X) + a - 1$*;*

(c) and if $a \neq 0$*, then* $I_S(y, F_X) = I_S(y', F_X)$*.*

(ii) For a new function $y'(t) = y(t) + z(t)$ *we have*

(a) $I_M(y', F_X) = I_M(y, F_X) + I_M(z, F_X)$*;*

(b) $I_A(y', F_X) = I_A(y, F_X) \iff \langle \tilde{z}, \tilde{\mu} \rangle = 0$*;*

(c) and $I_S(y, F_X) = I_S(y', F_X) \iff \frac{\langle \tilde{y}, \tilde{\mu} \rangle}{\|\tilde{y}\|} = \frac{\langle \tilde{y}, \tilde{\mu} \rangle + \langle \tilde{z}, \tilde{\mu} \rangle}{\|\tilde{y} + \tilde{z}\|}$*.*

(iii) For a new function $y'(t) = z(t)y(t)$*, we have*

(a) $I_A(y, F_X) = I_A(y', F_X) \iff \langle \tilde{y}, \tilde{\mu} \rangle = \langle \tilde{zy}, \tilde{\mu} \rangle$*;*

(b) and $I_S(y, F_X) = I_S(y', F_X) \iff \frac{\langle \tilde{y}, \tilde{\mu} \rangle}{\|\tilde{y}\|} = \frac{\langle \tilde{zy}, \tilde{\mu} \rangle}{\|\tilde{zy}\|}$*.*

Proof. See Section B.1 of Appendix B. □

Proposition 4.3 provides insights into how the different Fast-MUOD indices behave under transformations and hence, why they are useful for targeting the corresponding types of outliers. The first property demonstrates that I_M is sensitive to the translation and scaling of a function (by real numbers), which is a desirable property, because I_M is intended to be a measure of magnitude outlyingness; it should consequently capture any magnitude shift to be such a worthy measure (of magnitude outlyingness). This property is shown in the first row of Figure 4.1 where 100 realizations of the process $X_i(t) = 4t$ are obtained by adding some noise generated from a Gaussian process $e_i(t)$ with zero mean and covariance function $\gamma(s,t) = \exp\{-0.3|t-s|\}$, for $s, t \in [0,1]$ and $i = 1, \ldots, 100$. One of these realizations is then transformed by scaling ($a = 2$) and shifting it ($b = 3$). The second plot on the first row of Figure 4.1 shows that the index of the transformed function ($I_M(y'', F_X)$) is equal to that of the original realization of $X(t)$ scaled and shifted with the same values ($aI_M(y, F_X) + b$).

The second property, illustrated in the second row of Figure 4.1, demonstrates that the magnitude index preserves the functional addition operation, which is a desirable property because functional addition causes a shift in magnitude; this shift is captured by the magnitude index. Thus, for a new function $y'(t)$ obtained by adding another function $z(t)$ to $y(t)$, $I_M(y') = I_M(y) \iff I_M(z) = 0$.

Unlike the magnitude index, the amplitude index is not sensitive to translation by a scalar. Because shifting a (periodic) function in magnitude does not inherently change

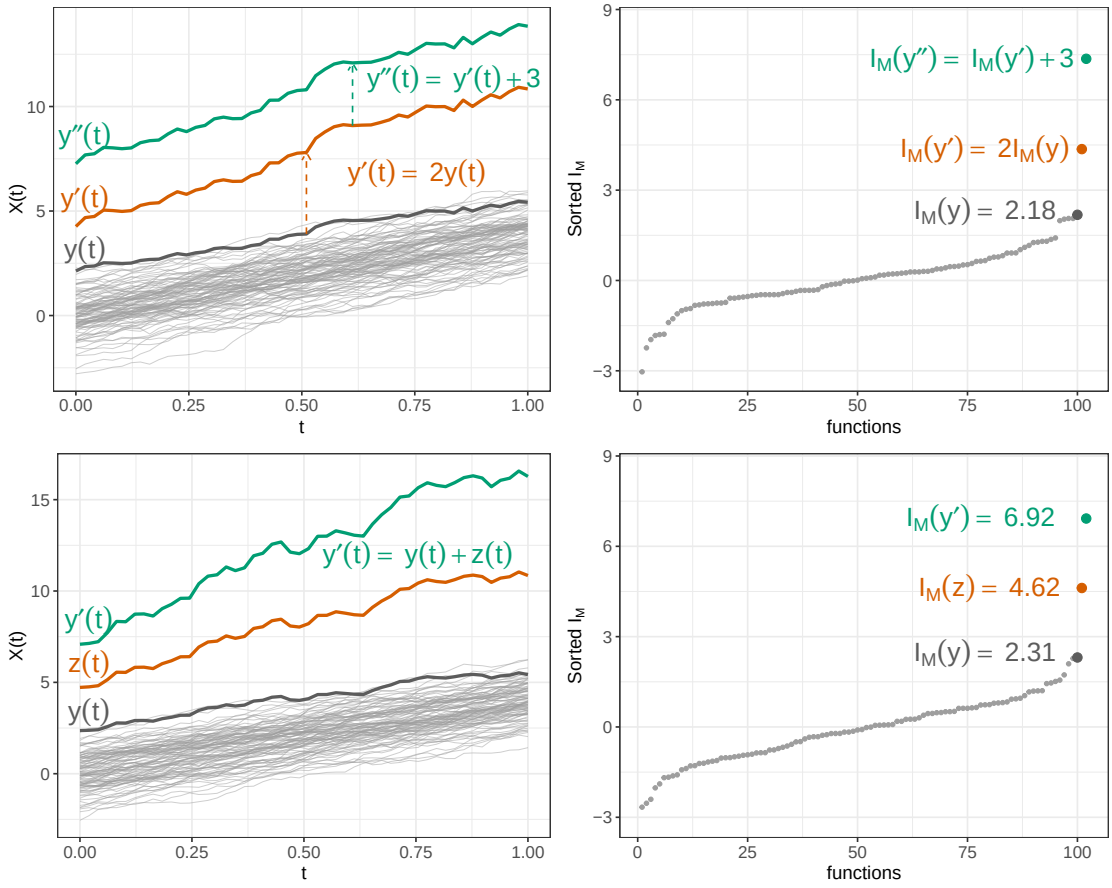

Figure 4.1: Illustration of the magnitude indices under scaling and translation. Functions and their sorted magnitude indices are shown in the first and second columns, respectively. Functions in grey are the bulk of the data. The function in black is $y(t)$. Functions in orange and green are transformed functions. The same colour code applies to points representing the indices.

its amplitude, a good measure of amplitude outlyingness should ignore such a transformation. However, the index I_A is sensitive to scaling by a scalar because this transformation changes the amplitude of a function. In fact, Proposition 3.1 indicates that for a transformed function $y'(t) = ay(t), \quad a \in \mathbb{R}$, $I_A(y') = I_A(y) \iff a = 1$. This property of the amplitude index is illustrated in the first and second rows of Figure 4.2. Proposition 4.3 further shows that I_S is neither sensitive to scaling nor translation by scalar values (Figure 4.3). The remaining properties in Proposition 4.3 establish conditions under which the amplitude and shape indices of a transformed function remain the same.

The Fast-MUOD indices defined above slightly differ from those used in Ojo et al. (2021a) (presented in Chapter 3, Equations (3.12) - (3.14)). The original amplitude and

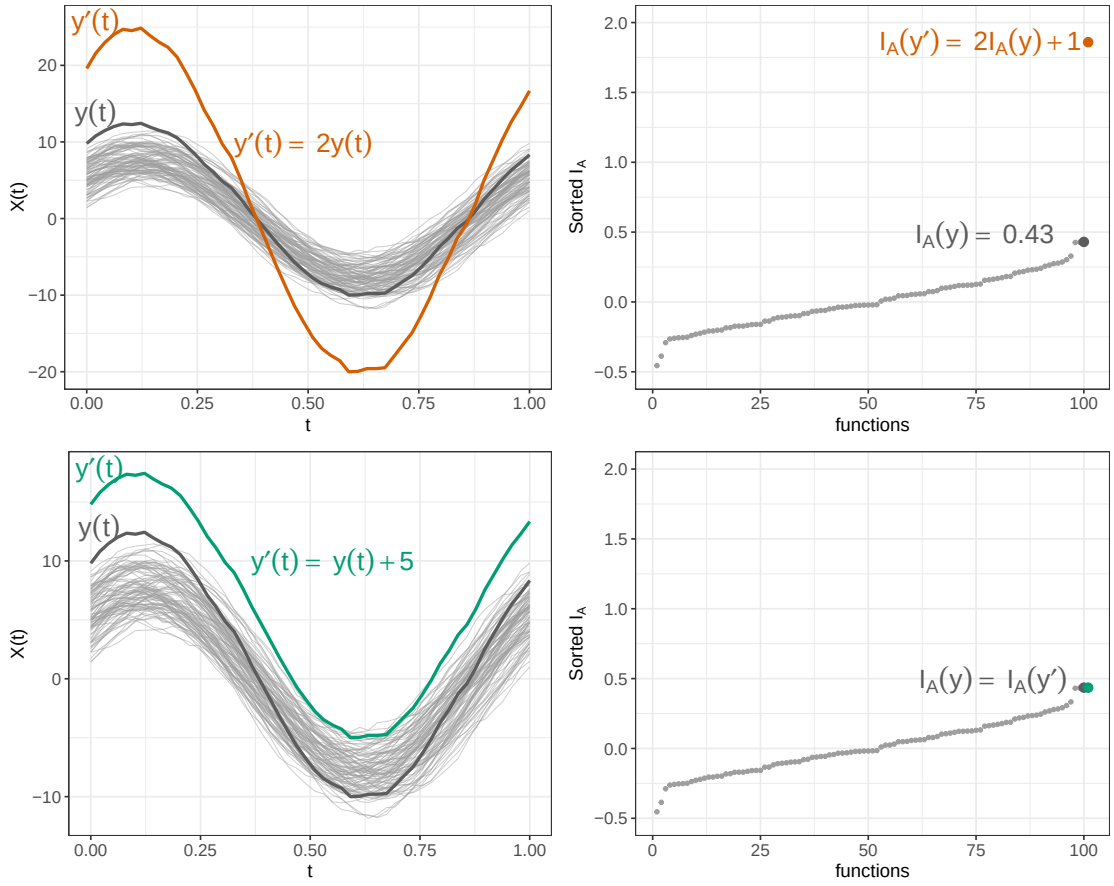

Figure 4.2: Illustration of the amplitude indices under simple transformation. Functions and their sorted amplitude indices are shown in the first and second columns, respectively. The functions in grey are the bulk of the data. The function in black is $y(t)$. The functions in orange and green are transformed functions. The same colour code applies to the points representing the indices.

magnitude indices (which we will denote by I_{A_v} and I_{M_v} in this chapter) used by Ojo et al. (2021a) had absolute values that guaranteed these indices were positive. These indices also exhibit slightly different properties (because of the use of the absolute value function). For completeness, we provide definitions and properties of the original indices, I_{A_v} and I_{M_v}, used by Ojo et al. (2021a) in the next subsection.

4.2.3 Original Fast-MUOD Magnitude and Amplitude Indices

Definition 4.5 (Original Fast-MUOD indices)**.** Let X be a stochastic process in $L^2([0,1])$ with distribution F_X and let $\mu(t) = \mathbb{E}(X(t))$ be its population mean function. An alter-

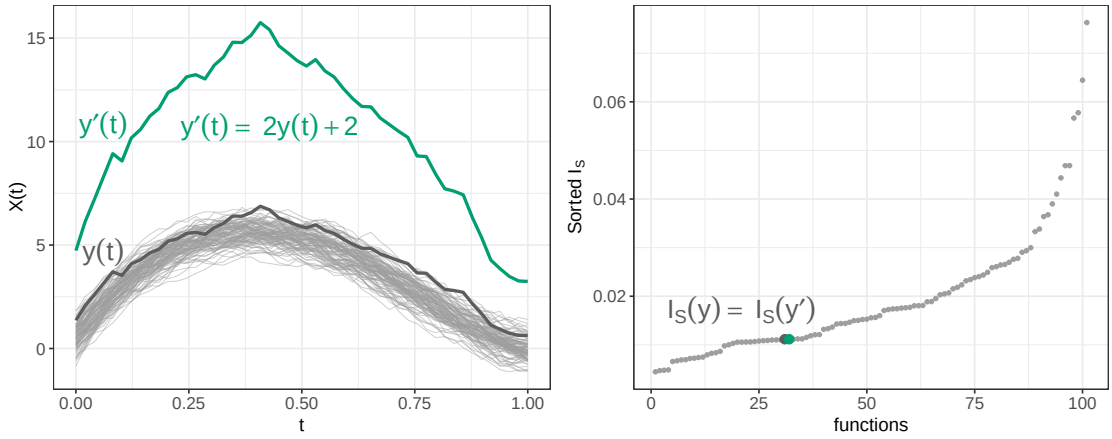

Figure 4.3: Illustration of the shape indices under simple transformation. Functions and their sorted shape indices are shown in the first and second columns, respectively. The functions in grey are the bulk of the data. The function in black is $y(t)$. The function in green is the transformed function $(y'(t))$. The same colour code applies to the points representing the indices.

native definition of the amplitude index of a function $y \in L^2([0,1])$ w.r.t. F_X is:

$$I_{A_v}(y, F_X) := \left| \frac{\int \tilde{y}(t)\tilde{\mu}(t)dt}{\int \tilde{\mu}(t)^2 dt} - 1 \right| = \left| \frac{\langle \tilde{y}, \tilde{\mu} \rangle}{\|\tilde{\mu}\|^2} - 1 \right| = |I_A(y, F_X)| \,. \tag{4.3}$$

Furthermore, an alternative definition of the magnitude index of y w.r.t. F_X is:

$$I_{M_v}(y, F_X) := \left| \int y(t)dt - \beta(y) \int \mu(t)dt \right| = |I_M(y, F_X)| \,. \tag{4.4}$$

The primary difference between $I_A(y, F_X)$ and $I_{A_v}(y, F_X)$ is the use of the absolute value function in the latter, guaranteeing that $I_{A_v}(y, F_X)$ is always positive (the same applies to $I_M(y, F_X)$ and $I_{M_v}(y, F_X)$). However, this makes the distributions of $I_{A_v}(y, F_X)$ and $I_{M_v}(y, F_X)$ right-skewed, compared to normal distribution for $I_A(y, F_X)$ and $I_M(y, F_X)$ (Figure 4.4). Note that additional information about the nature of magnitude and amplitude outliers is lost because $I_M(y, F_X)$ ($I_A(y, F_X)$) assigns larger index values to higher magnitude (amplitude) outliers and smaller index values to lower magnitude (amplitude) outliers, which is not the case with $I_{A_v}(y, F_X)$ and $I_{M_v}(y, F_X)$. For these reasons, we recommend to use $I_A(y, F_X)$ and $I_M(y, F_X)$ whenever possible.

It is straightforward to construct the respective sample and finite dimensional versions of $I_{A_v}(y, F_X)$ and $I_{M_v}(y, F_X)$ following the ideas in Subsection 4.2.1. However, their properties slightly differ under simple transformations.

Proposition 4.4 (Properties of original Fast-MUOD indices)**.** *Suppose that X is a stochastic*

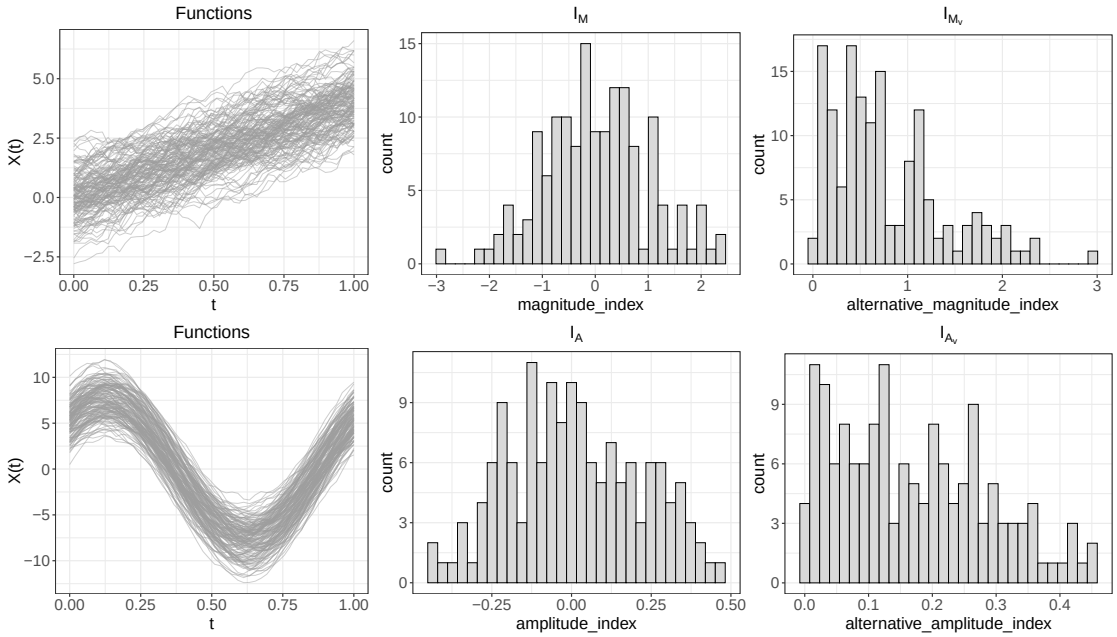

Figure 4.4: Fast-MUOD and alternative Fast-MUOD amplitude and magnitude indices. The first row shows I_M and I_{M_v} with an approximately normal and right-skewed distribution respectively. The second row shows the same for I_A and I_{A_v}.

process in $L^2([0,1])$ with distribution F_X and mean function $\mu(t)$. Let y and z other functions in $L^2([0,1])$ (which may be realizations of X) and let $a, b \in \mathbb{R}$. Then the following statements hold.

(i) For a new function $y'(t) = ay(t) + b$: $I_{M_v}(y', F_X) = |aI_M(y, F_X) + b|$.

(ii) For a new function $y'(t) = y(t) + z(t)$: $I_{M_v}(y', F_X) = |I_M(y, F_X) + I_M(z, F_X)|$.

(iii) For a new function $y'(t) = ay(t) + b$: $I_{A_v}(y', F_X) = |aI_A(y, F_X) + a - 1|$.

(iv) For a new function $y'(t) = y(t) + z(t)$: if $\langle \tilde{z}, \tilde{\mu} \rangle = 0$ then $I_{A_v}(y', F_X) = I_{A_v}(y, F_X)$.

(v) For a new function $y'(t) = z(t)y(t)$: if $\langle \tilde{zy}, \tilde{\mu} \rangle = \langle \tilde{y}, \tilde{\mu} \rangle$ then $I_{A_v}(y', F_X) = I_{A_v}(y, F_X)$.

Proof. Proofs of the statements follows directly from the definition of I_{M_v} and I_{A_v}, and application of Proposition 4.3. □

The following corollary establishes conditions under which the original magnitude and amplitude indices of a function remain the same after they have been scaled and/or translated.

Corollary 4.1. *Suppose that X is a stochastic process in $L^2([0,1])$ with distribution F_X and mean function $\mu(t)$. Let y be another function in $L^2([0,1])$ and let $a, b \in \mathbb{R}$. Then the following statements hold,*

(i) For a new function $y'(t) = ay(t) + b$: $I_{M_v}(y', F_X) = I_{M_v}(y, F_X)$ iff $b = (-a \pm 1)I_M(y, F_X)$

(ii) For a new function $y'(t) = ay(t) + b$: $I_{A_v}(y', F_X) = I_{A_v}(y, F_X)$ iff $a = 1$ or $a = \frac{1-I_A(y,F_X)}{1+I_A(y,F_X)}$

Proof. See Section B.2 of Appendix B □

4.2.4 Implementation and Cutoffs for Fast-MUOD Indices

The sample versions of I_A and I_M were implemented in R (R Core Team, 2022) using an algorithm similar to the one presented in Ojo et al. (2021a) (Chapter 3, Algorithm 3). Furthermore, the point-wise median is used in the implementation of I_{A_n} and I_{M_n} instead of the point-wise sample mean as the former is much more robust to outliers. In Ojo et al. (2021a), where the sample versions of I_{A_v}, I_{M_v} and I_S were proposed, a classical boxplot was proposed to determine a cutoff for I_{A_v}, I_{M_v} and I_S by considering only the upper whisker of the boxplot (the third quartile extended by 1.5 times the inter-quartile range of I_{A_v}, I_{M_v} and I_S) as a cutoff. This worked because I_{A_v}, I_{M_v} and I_S are always non-negative and their distributions are right-skewed. For the sample versions of I_A and I_M, both the upper and lower whiskers of the boxplot have to be considered for a cutoff because outliers have indices that occur on both tails of the distribution of these indices. Thus we propose to consider as amplitude outliers observations with I_{A_n} value greater than $Q_{3I_{A_n}} + 1.5 \times IQR_{I_{A_n}}$ or less than $Q_{1I_{A_n}} - 1.5 \times IQR_{I_{A_n}}$. We also propose the same rule for flagging magnitude outliers with I_{M_n}.

4.3 Discussion

This chapter examined the theoretical properties of the Fast-MUOD indices. First, we presented definitions and sample approximations of the indices. Then, we provided provided a convergence proof and explored the behaviour of these indices under simple transformations. The Fast-MUOD magnitude index of a curve, w.r.t. to the sample, changes when the curve is scaled or shifted by a scalar value. In addition, the magnitude index preserves the functional addition operation. On the other hand, the amplitude index is sensitive to scaling by a scalar but not to a shift. The shape index is neither sensitive to scaling or translation. These properties make the Fast-MUOD indices well

suited to targeting and identifying different types of outliers. In the next chapter, we present three techniques for detecting outliers in multivariate functional data using the Fast-MUOD indices.

Chapter 5

Multivariate Functional Outlier Detection with the Fast-MOUD Indices

This chapter is based on:

Ojo, O. T., Fernández Anta, A., Genton, M. G., & Lillo, R. E. (2022). "Multivariate Functional Outlier Detection using the FastMUOD Indices". arXiv:2207.12803

5.1 Fast-MUOD Extensions to Multivariate Functional Data

Fast-MUOD was proposed for univariate functional data; however, many real functional data are multivariate in nature. Consequently, we present some techniques for detecting outliers in multivariate functional data using Fast-MUOD indices. The proposed techniques all involve applying the univariate Fast-MUOD indices on univariate functional datasets obtained from the multivariate functional data of interest; hence, for our multivariate applications, the definitions and properties presented in Section 4.2 are relevant.

5.1.1 Marginal Outlier Detection with Fast-MUOD Indices

Suppose $\{\boldsymbol{Y}(t), t \in \mathcal{I}\}$ is a stochastic process defined on $\mathcal{I} = [0,1]$ taking values in $\mathbb{R}^d$. Let the distribution of $\boldsymbol{Y}(t)$ be $F_{\boldsymbol{Y}(t)}$ and let $F_{Y^j(t)}$ be the distribution of the j^{th} marginal component of $\boldsymbol{Y}(t)$, with $j = 1, \ldots, d$. Consider a set of n realizations of $\boldsymbol{Y}$: $\{\boldsymbol{Y}_i(t)\}_{i=1}^n$. To identify outliers in $\{\boldsymbol{Y}_i(t)\}_{i=1}^n$, a first option is to apply Fast-MUOD to the d marginals of the observed curves, (i.e., $Y_i^j(t)$) and identify $\boldsymbol{Y}_i$ as an outlier if it

is an outlier (of any type: shape, amplitude, or magnitude) in any of the d margins. However, this technique has the obvious limitation of not detecting "joint-outliers", i.e., observations that are not outliers in any of the marginal distributions but are outlying compared to the joint distribution of the data. It is also quite prone to false positives (FPs) because the final FPs of the procedure is the union of the FPs of the three indices for each margin of the multivariate functional data to which Fast-MUOD is applied.

5.1.2 Stringing Marginal Functions into Univariate Functional Data

For a multivariate functional observation, $\boldsymbol{Y}_i(t)$, we can concatenate or "string" its d univariate dimensions (i.e., $Y_i^1(t), Y_i^2(t), \ldots, Y_i^d(t)$) together into a single univariate function. Thus, we can obtain univariate functional curves $Z_i(t')$ defined on the interval $[0, d]$ from the original multivariate observations $\{\boldsymbol{Y}_i(t)\}_{i=1}^n$ given by $Z_i(t') := Y_i^j(t' - j + 1)$, whenever $t' \in (j-1, j]$, for $j = 1, \ldots, d$ (set $Z_i(0) := Y_i^1(0)$). Then, Fast-MUOD can be applied on $\{Z_i(t')\}_{i=1}^n$ by estimating for each Z_i, the indices $I_{S_n}(Z_i, F_{Z_n})$, $I_{A_n}(Z_i, F_{Z_n})$ and $I_{M_n}(Z_i, F_{Z_n})$ and applying the cutoffs described in Subsection 4.2.4. There are two potential issues with this technique. First, the d univariate functions $(Y_i^1(t), Y_i^2(t), \ldots, Y_i^d(t))$ may have different ranges in which case it might be convenient to scale the dimensions of $\boldsymbol{Y}_i(t)$ into the same range (e.g., using a min-max scaling). Furthermore, changing the order of stringing of the marginal functions into univariate functions might have an effect on which observations are detected as outliers or not, e.g., for a set of multivariate functional curves $\{\boldsymbol{R}(t)\}_{i=1}^n$, the set of stringed functions $\{A_i(t') = R_i^j(t' - j + 1)\}_{i=1}^n$ and $\{B_i(t') = Y_i^{d+1-j}(t' - j + 1)\}_{i=1}^n$ might produce different outliers.

5.1.3 Random Projections

Owing to the potential limitations of the two techniques presented in Subsections 5.1.1 and 5.1.2, we introduce a new technique based on random projections. For a multivariate functional dataset $\{\boldsymbol{Y}_i(t)\}_{i=1}^n$ taking values in $\mathbb{R}^d$, we generate L random unit vectors $\{\hat{\boldsymbol{a}}_l \in \mathbb{R}^d : l = 1, \ldots, L\}$. Then, we compute the projection of $\boldsymbol{Y}_i(t)$ in the direction of $\hat{\boldsymbol{a}}_l$:

$$Y_{i,l}(t) = \hat{\boldsymbol{a}}_l^\top \boldsymbol{Y}_i(t) = \sum_{j=1}^{d} a_l^j Y_i^j(t) \in \mathbb{R},$$

where $Y_i^j(t)$ is the j^{th} component of $\boldsymbol{Y}_i$ evaluated at t and a_l^j is the j^{th} component of $\hat{\boldsymbol{a}}_l$. Then, Fast-MUOD can be applied on the univariate functional data $\{Y_{i,l}(t)\}_{i=1}^n$ by estimating for each $Y_{i,l}(t)$, the indices $I_{S_n}(Y_{i,l}, F_{Y_{n,l}})$, $I_{A_n}(Y_{i,l}, F_{Y_{n,l}})$, and $I_{M_n}(Y_{i,l}, F_{Y_{n,l}})$, where $F_{Y_{n,l}}$ is the empirical distribution of $\{\boldsymbol{Y}_i(t)\}_{i=1}^n$ projected on $\hat{\boldsymbol{a}}_l$. Applying the cut-

off described in Subsection 4.2.4 on the sets $\{I_{S_n}(Y_{i,l}, F_{Y_{n,l}})\}_{i=1}^n$, $\{I_{A_n}(Y_{i,l}, F_{Y_{n,l}})\}_{i=1}^n$, and $\{I_{M_n}(Y_{i,l}, F_{Y_{n,l}})\}_{i=1}^n$ reveals whether $\boldsymbol{Y}_i(t)$ is a shape, amplitude, or magnitude outlier, respectively, when projected in the direction of $\hat{\boldsymbol{a}}_l$.

Threshold for the Random Projections

To combine all information from the L projections, we adopt a "voting system" in which a multivariate function is flagged as an outlier of a specific type if it is an outlier of that type in more than a fixed proportion of the projection directions. To this end, we define the following indicator functions:

$$\begin{aligned} O_{S,l}(\boldsymbol{Y}_i) &\coloneqq \mathbb{1}\{\text{if } Y_{i,l}(t) \text{ is a shape outlier}\}, \\ O_{A,l}(\boldsymbol{Y}_i) &\coloneqq \mathbb{1}\{\text{if } Y_{i,l}(t) \text{ is an amplitude outlier}\}, \\ O_{M,l}(\boldsymbol{Y}_i) &\coloneqq \mathbb{1}\{\text{if } Y_{i,l}(t) \text{ is a magnitude outlier}\}. \end{aligned} \tag{5.1}$$

These indicator functions indicate whether $\boldsymbol{Y}_i$ is a shape, amplitude, or magnitude outlier when $\boldsymbol{Y}_i(t)$ is projected in the direction of $\hat{\boldsymbol{a}}_l$. Then, we fix the threshold triple $Q = (\tau_S, \tau_A, \tau_M)$, where $\tau_S, \tau_A, \tau_M \in [0, 1]$, and declare $\boldsymbol{Y}_i(t)$ a "shape" outlier if the $\mathbb{E}_l[O_{S,l}(\boldsymbol{Y}_i)] \geq \tau_S$. Similarly, we declare $\boldsymbol{Y}_i(t)$ an "amplitude" outlier if $\mathbb{E}_l[O_{A,l}(\boldsymbol{Y}_i)] \geq \tau_A$ and a "magnitude" outlier if $\mathbb{E}_l[O_{M,l}(\boldsymbol{Y}_i)] \geq \tau_M$. Note that the classification of a multivariate functional outlier into a specific type ("amplitude", "magnitude" or "shape") now indicates that the function is an outlier of that type in at least τ proportion of the projections (for $\tau \in Q$). This classification is also not necessarily disjoint since an observation can be flagged as an outlier of more than one type in each projection (for examples, see Subsections 5.2.3 and 5.3.1).

The threshold triple Q helps to control the false positive rate (FPR) of the procedure. The lower the value of $\tau \in Q$, the more aggressive the procedure is in flagging an observation as an outlier (because flagging an outlier requires less number of "votes" from the random projections). Higher values of τ, on the other hand, make the procedure more conservative in detecting outliers. For the amplitude and magnitude indices, we find (in our simulation tests) that limiting the value of both τ_A and τ_M to the interval $[0.3, 0.7]$ works well for most applications (see Section B.5 in the Supplementary Material). In the case when there are magnitude (or amplitude) outliers in the projected data, τ_M (or τ_A) should be close to the lower bound of 0.3, which is sufficiently low to allow for flagging the outliers without introducing many FPs. When there are no magnitude (amplitude) outliers, τ_M (or τ_A) should be close to 0.7, which is a sufficiently high proportion to prevent FPs. When there are magnitude (or amplitude) outliers, some random projections of the data will not detect all the true outliers, and therefore

it is imperative not to set τ_M (or τ_A) to a high proportion in this case. However, setting τ_M (or τ_A) to a very low proportion, even when there are magnitude (amplitude) outliers, will yield many FPs because some non-outliers will be erroneously flagged as outliers in some of the projections. For the shape index, we suggest limiting τ_S to the interval $[0.4, 0.7]$ because we know from previous studies that the shape index is more prone to FPs than the magnitude and amplitude indices (partly because of its skewed distribution and sometimes because of random noise in the data, see Ojo et al. (2021a)).

Selecting the Thresholds Q

It is possible to select the threshold values in Q (within the suggested intervals $[0.3, 0.7]$ for τ_A/τ_M and $[0.4, 0.7]$ for τ_S) in a data-driven way if the distribution or model from which the functional data come from is known. Consider, for example, the following model for $T \in \{S, A, M\}$ (where S, A, and T denote shape, amplitude, and magnitude, respectively):

$$\tau_T \coloneqq \begin{cases} \gamma_T - \eta_T \frac{\Delta_{PT}}{\Delta_C} & \text{if } \frac{\Delta_{PT}}{\Delta_C} \in [0,1], \\ \gamma_T - \eta_T & \text{if } \frac{\Delta_{PT}}{\Delta_C} > 1, \\ \gamma_T & \text{otherwise,} \end{cases} \tag{5.2}$$

where γ_T, $\eta_T \in [0, 1]$. The term Δ_{PT} is an estimate of the proportion of outliers of type T present in the data computed by subtracting the expected proportion of FP of type T under the null model (a model where there are no outliers) from the average proportions of outliers of type T found over all L projections:

$$\Delta_{PT} = \frac{\sum_{l=1}^{L} \sum_{i=1}^{n} \hat{O}_{T,l}(\boldsymbol{Y}_i)}{n \times L} - \hat{B}_T, \tag{5.3}$$

where $\hat{B}_T$ is an estimate of the "baseline" expected proportion of FP of type T under the null model and $\hat{O}_{T,l}$ is an estimate of the indicator functions in Equation (5.1). Likewise, Δ_C is an estimate of the proportion of all unique outliers (regardless of their type) present in the data computed by subtracting the expected proportion of total FPs under the null model from the average proportions of total unique outliers found over all L projections:

$$\Delta_C = \frac{\sum_{l=1}^{L} \sum_{i=1}^{n} \hat{O}_{l}(\boldsymbol{Y}_i)}{n \times L} - \hat{B}_C, \tag{5.4}$$

where $\hat{O}_l$ is an estimate of the indicator function $O_l(\boldsymbol{Y}_i) \coloneqq \mathbb{1}\{\text{if any } O_{T,l}(Y_i) = 1 \text{ for } T \in \{S, A, M\}\}$ and $\hat{B}_C$ is an estimate of the "baseline" expected proportion of total FPs (of

any type) under the null model. To ensure that τ_T is within an interval $[a, b] \subset [0, 1]$ of interest in Equation (5.2), it suffices to set $\gamma_T = b$ and $\eta_T = b - a$. For example, to ensure that $\tau_S \in [0.4, 0.7]$, we can set $\gamma_S = 0.7$ and $\eta_S = 0.3$ in Equation (5.2). The intuition is that if there are only shape outliers in the data, the proportion $\frac{\Delta_{PS}}{\Delta_C}$ will be close to 1, resulting in $\tau_S \approx 0.4$, which is the lower bound of the suggested interval $[0.4, 0.7]$ for shape outliers. On the other hand, if there are no shape outliers, $\frac{\Delta_{PS}}{\Delta_C}$ will be close to 0 so that $\tau_S \approx 0.7$, which is the upper bound of the suggested interval $[0.4, 0.7]$, thereby controlling for FPs. However, to estimate the proportion $\frac{\Delta_{PS}}{\Delta_C}$, it is necessary to have an estimate of the baseline values $\hat{B}_S$ and $\hat{B}_C$ in Equations (5.3) and (5.4), respectively. If the model or distribution of the data is known, it is possible to estimate these baseline values by simulating the null model (observation from the model without outliers) and estimating the proportion of FP of type S ($\hat{B}_S$) and the proportion of all FPs ($\hat{B}_C$).

However, for real applications, the distribution or model which the data come from is unknown, and therefore it is impossible to estimate the baselines B_T and B_C; hence, the model in Equation (5.2) cannot be used to fix the threshold values in Q. An obvious option is to consider as an outlier of type T any observation that is flagged as an outlier of type T in at least one projection, i.e., flag $\boldsymbol{Y}_i(t)$ as an outlier of type T if $\mathbb{E}_l[O_{T,l}(\boldsymbol{Y}_i)] > 0$. This has the downside of being prone to FPs since it does not control for any FP due to the projection directions and the Fast-MUOD indices. Another option, which we recommend, is to use the threshold triple $Q = (\tau_S, \tau_A, \tau_M) = (0.4, 0.3, 0.3)$ that we have found to have a well-balanced performance across various scenarios in our simulation studies (see Section B.5 of the Supplementary Material).

5.2 Simulation Study

We performed a simulation study to compare the various techniques discussed in Section 5.1 with state-of-the-art methods.

5.2.1 Simulation Models

We simulated trivariate ($d = 3$) functional datasets from models based on the truncated Karhunen–Loève expansion for multivariate functional data (Happ and Greven, 2018):

$$\boldsymbol{Y}_i(t) = \boldsymbol{\mu}(t) + \sum_{m=1}^{M} \rho_{i,m} \boldsymbol{\psi}_m(t) + \boldsymbol{\epsilon}(t), \quad i = 1, \ldots, n, \; M \in \mathbb{N},$$

where $\boldsymbol{Y}_i(t) \in \mathbb{R}^3$, $\boldsymbol{\mu}(t) \in \mathbb{R}^3$ is the multivariate mean function, and $\boldsymbol{\psi}_m(t) \in \mathbb{R}^3$, for $m = 1, \ldots, M$ are multivariate eigenfunctions. The scores $\rho_{i,m} \overset{\text{iid}}{\sim} \mathcal{N}(0, \nu_m)$ for eigen-

values ν_m that are linearly decreasing ($\nu_m = \frac{M+1-m}{M}$). The errors $\boldsymbol{\epsilon}(t) \in \mathbb{R}^3$, and $\boldsymbol{\epsilon}(t) \overset{\text{iid}}{\sim} \mathcal{N}_3(\mathbf{0}, \boldsymbol{\Sigma})$, with $\boldsymbol{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$, where $\sigma_i \overset{\text{iid}}{\sim} U[0.1, 0.3]$, for $i = 1, 2, 3$. The eigenfunctions $\boldsymbol{\psi}_m(t)$ were constructed by splitting orthonormal Fourier functions into $d = 3$ pieces and shifting them to the required domain (Happ-Kurz, 2020). We set $M = 9$ basis functions. The sample size for each dataset is $n = 100$ and we considered a contamination rate of 10%. The simulated functions are evaluated at 50 equidistant points in $[0, 1]$. For each simulation model considered, the non-outliers were generated from a main model while the outliers were generated from a contaminated model, both listed below:

(i) Simulation Model 0 (No outliers):

Main Model: $\boldsymbol{Y}_i(t) = \boldsymbol{\mu}(t) + \sum_{m=1}^{M} \rho_{i,m}\boldsymbol{\psi}_m(t) + \boldsymbol{\epsilon}(t)$; where $\boldsymbol{\mu}(t) = (4t, 30t(1-t)^{\frac{3}{2}}, 5(t-1)^2)^\top$.

(ii) Simulation Model 1 (Persistent magnitude outliers):

Main Model: The same as Model 0.

Contamination Model: $\boldsymbol{Y}_i(t) = \boldsymbol{\mu}(t) + \boldsymbol{u}(t) + \sum_{m=1}^{M} \rho_{i,m}\boldsymbol{\psi}_m(t) + \boldsymbol{\epsilon}(t)$; where $\boldsymbol{\mu}(t)$ is the same as in Model 0 and $\boldsymbol{u}(t)$ is given by: $u_i^j(t) = 8W_j$ for $j = 1, 2, 3$. W_j is sampled from $\{-1, 1\}$ with equal probability.

(iii) Simulation Model 2 (Non-persistent magnitude outliers):

Main Model: The same as Model 0 but with $\boldsymbol{\mu}(t) = (5\sin(2\pi t), 5\cos(2\pi t), 5(t-1)^2)^\top$.

Contamination Model: $\boldsymbol{Y}_i(t) = \boldsymbol{\mu}(t) + \boldsymbol{u}(t) + \sum_{m=1}^{M} \rho_{i,m}\boldsymbol{\psi}_m(t) + \boldsymbol{\epsilon}(t)$ where $\boldsymbol{\mu}(t)$ is the same as the Main Model above, and $\boldsymbol{u}(t)$ is given by:

$$u_i^j(t) = 8W_j \cdot \mathbb{1}\{\text{if } t \in [T_q, T_q + 0.1]\}.$$

W_j is the same as in Model 1 and $T_q \sim U[0, 0.9]$.

(iv) Simulation Model 3 (Shape outlier I):

Main Model: The same as Model 0 but with $\boldsymbol{\mu}(t) = (5\sin(2\pi t), 5\cos(2\pi t), 5(t-1)^2)^\top$.

Contamination Model: The same as Model 0 but with $\boldsymbol{\mu}(t)$ changed to:

$$\boldsymbol{\mu}(t) = (5\sin(2\pi(t-0.3)), 5\cos(2\pi(t-0.2)), 5(0.1-t)^2)^\top.$$

(v) Simulation Model 4 (Shape outlier II):

Main Model: $\boldsymbol{Y}_i(t) = \boldsymbol{\mu}(t) + \boldsymbol{u}(t) + \sum_{m=1}^{M} \rho_{i,m}\boldsymbol{\psi}_m(t) + \boldsymbol{\epsilon}(t)$; where $\boldsymbol{\mu}(t) = (5\sin(2\pi t), 5\cos(2\pi t), 5(t-1)^2)^\top$, and $\boldsymbol{u}(t)$ is given by $u_i^j(t) = \varrho_j$, with $\varrho_j \overset{\text{iid}}{\sim} U[-2.1, 2.1]$.

Contamination Model: Same as the Main Model above but with $\boldsymbol{u}(t)$ changed to:

$$\boldsymbol{u}(t) = (2\sin(4\pi t), 2\cos(4\pi t), 2\cos(8\pi t))^\top.$$

(vi) Simulation Model 5 (Amplitude outliers):

Main Model: The same as Model 0 but with $\boldsymbol{\mu}(t) = (5\sin(2\pi t), 5\cos(2\pi t), 5(t-1)^2)^\top$.

Contamination Model: $\boldsymbol{Y}_i(t) = \boldsymbol{\mu}(t) + \boldsymbol{u}(t) + \sum_{m=1}^{M} \rho_{i,m}\boldsymbol{\psi}_m(t) + \boldsymbol{\epsilon}(t)$; where $\boldsymbol{\mu}(t)$ is the same as the Main Model above and $\boldsymbol{u}(t)$ is given by:

$$\boldsymbol{u}_i(t) = ((2 + R_i^1)\mu^1(t),\ (2 + R_i^2)\mu^2(t), (2 + R_i^3)\mu^3(t) - 6)^\top.$$

$\mu^j(t)$ are the components of $\boldsymbol{\mu}(t)$ in the Main Model above and $R_i^j \overset{\text{iid}}{\sim} \text{Exp}(2)$, for $j = 1, 2, 3$.

(vii) Simulation Model 6 (Shape outlier III):

Main Model: $\boldsymbol{Y}_i(t) = \boldsymbol{\mu}(t) + \boldsymbol{u}(t) + \sum_{m=1}^{M} \rho_{i,m}\boldsymbol{\psi}_m(t) + \boldsymbol{\epsilon}(t)$; where $\boldsymbol{\mu}(t) = (5\sin(2\pi t), 5\cos(2\pi t), 5(t-1)^2)^\top$, and $\boldsymbol{u}(t) = (8t\sin(\pi t), t\cos(\pi t), 6\sin(2\pi t) - 3)^\top$.

Contamination Model: Same as the Main Model above but with $\boldsymbol{u}(t)$ changed to:

$$\boldsymbol{u}(t) = (10t\sin(\pi t), 11t\cos(\pi t), 10\sin(2\pi t) - 6)^\top.$$

Some sample data from these models are shown in Figures 5.1 and 5.2.

5.2.2 Outlier Detection Methods

Data were simulated from the seven models presented in Subsection 5.2.1. For each model, we compared the OD performance of the proposed extensions in Section 5.1. We also compared our proposals to other multivariate OD methods such as MS-plot (Dai and Genton, 2018), FOM, and functional adjusted outlyingness (FAO) (Rousseeuw et al., 2018). Since our Fast-MUOD-based proposals use indices that target different types of outliers, we can consider the OD performance of each index or consider a union

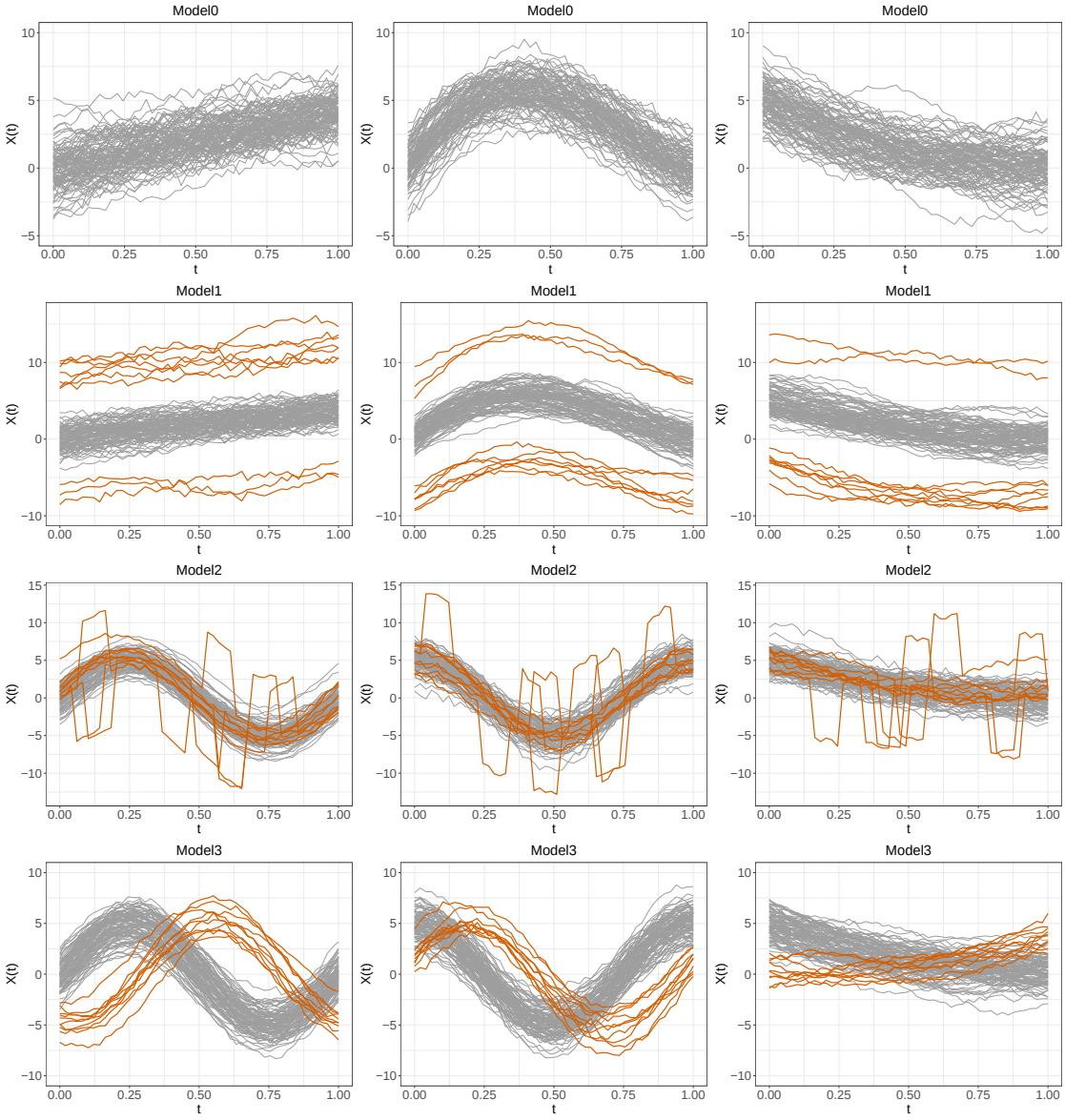

Figure 5.1: Sample data generated by Models 0 – 3 with contamination rate $\alpha = 0.10$, sample size $n = 100$, and evaluation point $d = 50$. Each row corresponds to a simulation model, and each column corresponds to a marginal component of the multivariate functional data. Outliers are shown in colour.

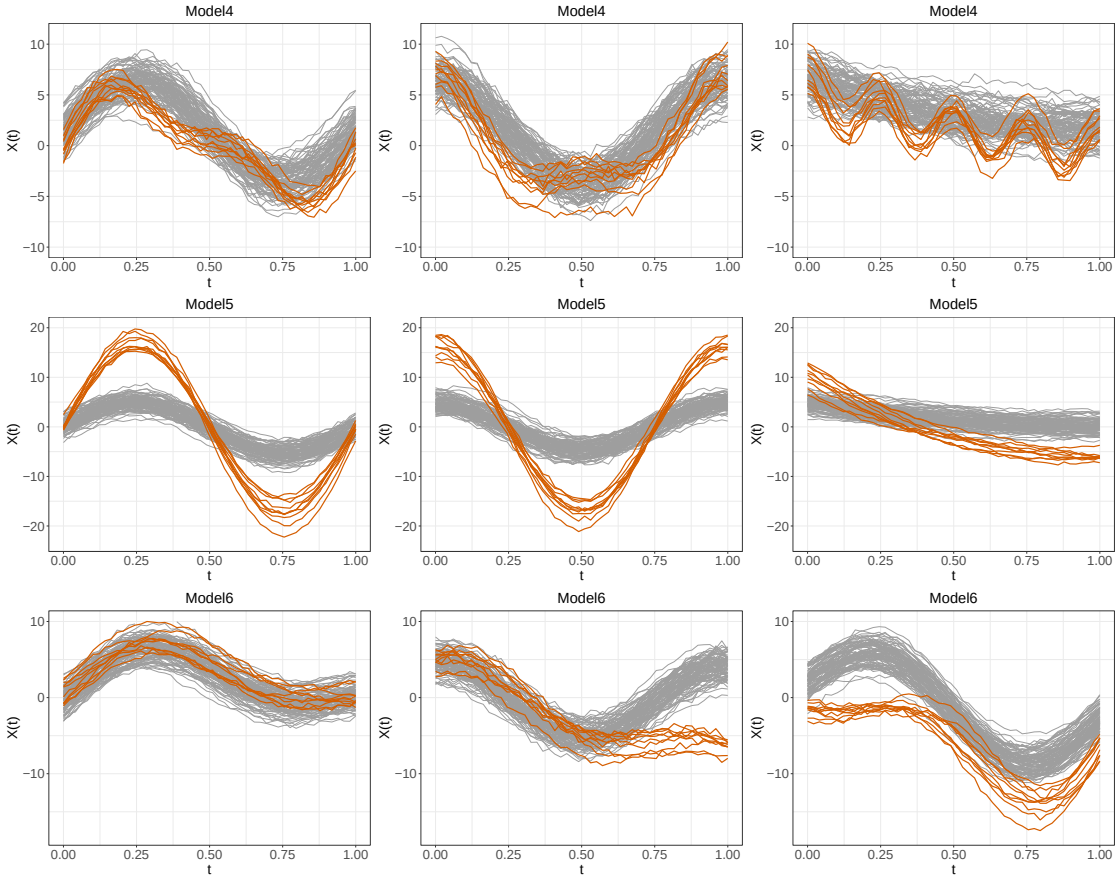

Figure 5.2: Sample data generated by Models 4 – 6 with contamination rate $\alpha = 0.10$, sample size $n = 100$, and evaluation point $d = 50$. Each row corresponds to a simulation model and each column corresponds to a marginal component of the multivariate functional data. Outliers are shown in colour.

of outliers flagged by the three indices. Thus, we considered the following methods in our comparison.

- FST-MAR: This is the union of all outlier types detected by applying Fast-MUOD to the marginal distributions of the multivariate functional data (described in Subsection 5.1.1). Consequently, an observation is an outlier if it is flagged as an outlier, of any type, in any of the three dimensions of the simulated multivariate functional data.

- FST-STR: This is the union of all outlier types detected by applying Fast-MUOD to the univariate functional data obtained by stringing marginal functions together (described in Subsection 5.1.2).

- FST-PRJ: This is the union of all outlier types detected by applying Fast-MUOD us-

ing random projections (described in Subsection 5.1.3). For our simulation tests, 60 random directions were used for projection. To generate each random direction, the components of the vector were simulated from the standard normal distribution and the resulting vector was then normalised to have a unit norm. Generating the random directions in this manner is straightforward and fast with limited computational burden.

We used Equation (5.2) to determine the threshold triple $Q = (\tau_S, \tau_A, \tau_M)$. Because we knew the base model (Model 0) from which the simulated data were generated, we could estimate the values of B_T and B_C used in Equations (5.3) and (5.4), respectively. For this purpose, we simulated data from Model 0 (null model without outliers) and computed the total FPR (of all outlier types) and the FPR of each type of outliers (shape, magnitude, and amplitude). Then, we used the computed FPR of all outliers as an estimate of B_C and the computed FPRs of outliers of each type as an estimate of B_T, for $T \in \{S, A, M\}$. Our simulation results yielded the following baseline values: $B_A = B_M = 0.009$, $B_S = 0.075$, and $B_C = 0.09$, which we then used in the estimation of $\frac{\Delta_{PT}}{\Delta_C}$ in Equation (5.2). Finally, we set the parameters γ_T and η_T in Equation (5.2) to $\gamma_T = 0.7$ for $T \in \{S, A, M\}$, $\eta_S = 0.3$, and $\eta_A, \eta_M = 0.4$ so that $\tau_S \in [0.4, 0.7]$ and $\tau_A, \tau_M \in [0.3, 0.7]$, as reported in Subsection 5.1.3.

- FST-PRJ-MG: This is the set of ONLY the "magnitude" outliers detected using FST-PRJ.

- FST-PRJ-AM: This is the set of ONLY the "amplitude" outliers detected using FST-PRJ.

- FST-PRJ-SH: This is the set of ONLY the "shape" outliers detected using FST-PRJ.

- FST-PRJ1: This is similar to FST-PRJ, but uses the threshold triple $Q = (\tau_S, \tau_A, \tau_M) = (0.4, 0.3, 0.3)$, which we recommend in real application when it is impossible to use Equation (5.2) to determine the values of Q.

- FST-PRJ1-MG: This is the set of ONLY the "magnitude" outliers detected using FST-PRJ1.

- FST-PRJ1-AM: This is the set of ONLY the "amplitude" outliers detected using FST-PRJ1.

- FST-PRJ1-SH: This is the set of ONLY the "shape" outliers detected using FST-PRJ1.

- FST-PRJ2: This is similar to FST-PRJ, but rather than using Equation (5.2) to select the threshold Q, we consider an observation as an outlier of type T if it is flagged as an outlier of that type in ANY projection, i.e., an observation is an outlier of type T if $\mathbb{E}_l[O_{T,l}(\boldsymbol{Y}_i)] > 0$ for $T \in \{S, A, M\}$.

- FST-PRJ2-MG: This is the set of ONLY the "magnitude" outliers detected using FST-PRJ2.

- FST-PRJ2-AM: This is the set of ONLY the "amplitude" outliers detected using FST-PRJ2.

- FST-PRJ2-SH: This is the set of ONLY the "shape" outliers detected using FST-PRJ2.

- MSPLOT: This is a multivariate functional outlier detection and visualisation method based on the directional outlyingness proposed in Dai and Genton (2018).

- FOM: The functional outlier map proposed in Rousseeuw et al. (2018) is similar to MSPLOT in that it is based on another type of directional outlyingess (DO) for multivariate data given by:

$$DO(\boldsymbol{Y}(t), F_{\boldsymbol{Y}(t)}) = \sup_{\boldsymbol{v}} \text{uDO}(\boldsymbol{v}^\top \boldsymbol{Y}(t), F_{\boldsymbol{v}^\top \boldsymbol{Y}(t)}),$$

where $\boldsymbol{v}$ is a random direction and uDO is the univariate directional outlyingness in Rousseeuw et al. (2018). The functional directional outylingess (fDO) is then defined as the integral of the DO over the domain:

$$\text{fDO}(\boldsymbol{Y}, F_{\boldsymbol{Y}}) = \int_{\mathcal{I}} DO(\boldsymbol{Y}(t), F_{\boldsymbol{Y}(t)}) w(t) dt,$$

where $w(t)$ is a weight function. The variation of DO values is defined as:

$$\text{vDO}(\boldsymbol{Y}, F_{\boldsymbol{Y}}) = \frac{\text{stdev}(DO(\boldsymbol{Y}(t), F_{\boldsymbol{Y}(t)})}{1 + \text{fDO}(\boldsymbol{Y}, F_{\boldsymbol{Y}})}.$$

The functional outlier map is then a scatter plot of the points $(\text{fDO}(\boldsymbol{Y}, F_{\boldsymbol{Y}}), \text{vDO}(\boldsymbol{Y}, F_{\boldsymbol{Y}}))$.

To flag functions as outliers, the combined functional outlyingness (CFO) is first computed as:

$$\text{CFO}(\boldsymbol{Y}, F_{\boldsymbol{Y}}) = \sqrt{\left(\frac{\text{fDO}(\boldsymbol{Y}, F_{\boldsymbol{Y}})}{\text{med}(\text{fDO}(\boldsymbol{Y}, F_{\boldsymbol{Y}})}\right)^2 + \left(\frac{\text{vDO}(\boldsymbol{Y}, F_{\boldsymbol{Y}})}{\text{med}(\text{vDO}(\boldsymbol{Y}, F_{\boldsymbol{Y}})}\right)^2}.$$

The log CFO is then computed and an observation is flagged as an outlier if $\frac{LCFO - \text{med}(LCFO)}{MAD(LCFO)} \geq \Phi(0.995)$, where $\Phi(\cdot)$ is the standard normal cumulative distribution.

- FAO: The functional adjusted outlyingness uses the adjusted outlyingness (AO) and its functional extension (fAO) proposed in Brys et al. (2005) and Hubert et al. (2015) respectively (instead of the DO and fDO) in a functional outlier map. The functional adjusted outlyingness of a function $\boldsymbol{Y}(t)$ w.r.t. $F_{\boldsymbol{Y}(t)}$ is the (weighted) integral of its pointwise AO values over the domain:

$$\text{fAO}(\boldsymbol{Y}, F_{\boldsymbol{Y}}) = \int_I AO(\boldsymbol{Y}(t), F_{\boldsymbol{Y}(t)}) w(t) dt.$$

The fAO and its variation (vAO) can then be used in a functional outlier map as in FOM above.

5.2.3 Simulation Results

For each of the models in Subsection 5.2.1, we tested the methods described Section 5.2.2. We set the contamination rate to 10% and performed 200 repetitions for each possible model. Table 5.1 shows the performance of the proposed techniques on the different simulation models. Because Model 0 is a null model without outliers, we only show the FPRs of the techniques. For Models 1-6, we show the true positive rate (TPR) and the FPR together with their respective standard deviations in parentheses.

The results of Model 0 show that FST-MAR and FST-PRJ2 both have very high FPRs. For FST-MAR, Fast-MUOD was independently applied on each of the three dimensions of the simulated trivariate functional data. For each dimension, the three Fast-MUOD indices contributed some FPs and the union of all these FPs over the three dimensions of the dataset yielded the overall FPR of about 26% for the null model in Model 0. Moreover, the extremely high FPR of FST-PRJ2 justifies the need to impose the threshold $Q = (\tau_S, \tau_A, \tau_M)$ used for determining if an observation is an outlier. Simply flagging an observation as an outlier if it is detected as an outlier in any random projection does not work for the Fast-MUOD indices. First, because a non-outlier might sometimes appear to be an outlier in the projected direction, and second because the Fast-MUOD indices (especially the shape index) and the boxplot cutoff procedure described in Subsection 4.2.4 also produce some FPs. These high FPRs can be observed for both methods (FST-MAR and FST-PRJ2) across all tested models.

For Model 1 where the outliers are clear magnitude outliers in all dimensions, all methods performed well, except for FST-MAR and FST-PRJ2 because of their high FPRs.

Table 5.1: Mean and Standard Deviation (in parentheses) of the true positive rate (TPR) and the false positive rate (FPR) (in percentage) over 200 repetitions for each model. Sample size $n = 100$, evaluation points $t_j = 50$, and contamination rate is 10%. Comparatively high TPRs ($\geq 95\%$) and low FPRs ($\leq 1\%$) are marked in bold. The proposed techniques are in italics.

Method	Model 0	Model 1		Model 2		Model 3	
	FPR	TPR	FPR	TPR	FPR	TPR	FPR
FST-MAR	26.2(4.2)	**100.0(0.0)**	25.2(4.8)	**99.9(1.0)**	13.9(3.2)	**100.0(0.0)**	13.4(3.3)
FST-STR	4.6(2.5)	**100.0(0.0)**	2.3(1.6)	90.4(10.9)	2.3(1.8)	**100.0(0.0)**	1.9(1.8)
FST-PRJ	1.7(2.6)	**100.0(0.0)**	**0.2(0.5)**	**98.7(4.3)**	**0.7(0.9)**	**100.0(0.0)**	**0.8(1.0)**
FST-PRJ-SH	1.7(2.6)	0.2(1.4)	**0.2(0.5)**	**98.7(4.3)**	**0.7(0.9)**	**100.0(0.0)**	**0.8(1.0)**
FST-PRJ-AM	**0.0(0.3)**	0.0(0.0)	**0.0(0.0)**	0.0(0.7)	**0.0(0.0)**	**100.0(0.0)**	**0.0(0.2)**
FST-PRJ-MG	**0.0(0.2)**	**100.0(0.0)**	**0.0(0.1)**	0.0(0.0)	**0.0(0.0)**	0.0(0.0)	**0.0(0.0)**
FST-PRJ1	3.7(1.9)	**100.0(0.0)**	3.4(1.9)	**99.2(2.8)**	1.1(1.2)	**100.0(0.0)**	**1.0(1.1)**
FST-PRJ1-SH	3.5(1.8)	4.2(6.5)	3.3(1.9)	**98.9(3.6)**	**0.8(1.0)**	**100.0(0.0)**	**0.8(0.9)**
FST-PRJ1-AM	**0.2(0.4)**	0.1(1.0)	**0.2(0.5)**	4.0(6.6)	**0.1(0.4)**	**100.0(0.0)**	**0.0(0.2)**
FST-PRJ1-MG	**0.1(0.4)**	**100.0(0.0)**	**0.0(0.2)**	2.9(5.3)	**0.2(0.6)**	2.1(4.7)	**0.2(0.5)**
FST-PRJ2	52.2(3.6)	**100.0(0.0)**	51.0(4.0)	**100.0(0.0)**	35.7(3.8)	**100.0(0.0)**	34.8(4.1)
FST-PRJ2-SH	47.0(3.3)	47.8(14.7)	47.1(4.0)	**100.0(0.0)**	28.2(3.0)	**100.0(0.0)**	27.4(3.3)
FST-PRJ2-AM	12.4(3.6)	12.6(10.5)	12.4(3.8)	64.3(14.6)	10.1(3.5)	**100.0(0.0)**	6.6(3.2)
FST-PRJ2-MG	14.0(4.1)	**100.0(0.0)**	7.2(3.1)	37.7(16.6)	8.9(3.6)	44.1(18.7)	9.0(3.6)
MSPLOT	1.6(1.7)	**100.0(0.0)**	**0.5(0.8)**	**100.0(0.0)**	**0.9(1.2)**	**100.0(0.0)**	1.1(1.4)
FOM	**0.3(0.6)**	**100.0(0.0)**	**0.1(0.3)**	**96.9(7.5)**	**0.0(0.3)**	70.4(29.0)	**0.1(0.3)**
FAO	**0.3(0.6)**	**100.0(0.0)**	**0.0(0.2)**	84.9(17.1)	**0.1(0.3)**	43.5(35.3)	**0.1(0.3)**

Method	Model 4		Model 5		Model 6	
	TPR	FPR	TPR	FPR	TPR	FPR
FST-MAR	76.3(17.0)	15.7(3.6)	**100.0(0.0)**	25.4(4.0)	**100.0(0.0)**	14.0(3.5)
FST-STR	49.7(27.1)	2.4(1.5)	**100.0(0.0)**	4.7(2.4)	**99.9(1.0)**	2.3(1.6)
FST-PRJ	17.0(23.7)	**0.3(0.8)**	**100.0(0.0)**	**0.2(0.4)**	**99.6(2.0)**	**0.6(0.8)**
FST-PRJ-SH	15.4(22.8)	**0.2(0.7)**	0.0(0.0)	**0.1(0.4)**	93.1(13.2)	**0.5(0.8)**
FST-PRJ-AM	0.0(0.0)	**0.0(0.2)**	**100.0(0.0)**	**0.0(0.2)**	1.3(6.3)	**0.0(0.1)**
FST-PRJ-MG	2.1(6.7)	**0.1(0.3)**	45.5(30.3)	**0.0(0.1)**	90.1(15.8)	**0.1(0.3)**
FST-PRJ1	45.2(19.2)	1.2(1.2)	**100.0(0.0)**	3.9(1.9)	**99.7(2.2)**	**0.9(0.9)**
FST-PRJ1-SH	41.8(18.7)	**1.0(1.1)**	0.0(0.0)	3.8(1.9)	**98.2(5.6)**	**0.8(0.9)**
FST-PRJ1-AM	0.1(1.2)	**0.2(0.5)**	**100.0(0.0)**	**0.0(0.1)**	18.4(16.4)	**0.1(0.3)**
FST-PRJ1-MG	5.6(8.5)	**0.1(0.4)**	86.6(12.3)	**0.1(0.4)**	90.6(11.9)	**0.0(0.2)**
FST-PRJ2	**96.9(6.7)**	37.6(3.9)	**100.0(0.0)**	51.8(3.9)	**100.0(0.0)**	39.1(3.9)
FST-PRJ2-SH	94.1(8.2)	29.9(3.6)	0.7(2.5)	49.3(3.7)	**100.0(0.0)**	34.1(3.9)
FST-PRJ2-AM	16.8(12.4)	13.2(3.7)	**100.0(0.0)**	5.9(2.9)	**95.4(7.6)**	9.5(3.3)
FST-PRJ2-MG	41.6(22.3)	8.3(3.2)	**99.4(2.5)**	6.0(2.8)	**99.8(1.9)**	7.5(3.1)
MSPLOT	34.8(21.9)	1.1(1.4)	**100.0(0.0)**	**0.9(1.1)**	**95.6(7.2)**	**1.0(1.2)**
FOM	1.9(4.6)	**0.1(0.3)**	**100.0(0.0)**	**0.1(0.3)**	51.4(33.9)	**0.1(0.3)**
FAO	1.1(3.6)	**0.1(0.3)**	**99.9(1.0)**	**0.1(0.3)**	25.6(29.9)	**0.0(0.1)**

The high TPRs of FST-PRJ1-MG and FST-PRJ-MG demonstrate that the magnitude indices detect the magnitude outliers while the other indices (FST-PRJ-AM, FST-PRJ1-AM, FST-PRJ-SH, and FST-PRJ1-SH) do not contribute significantly to the FPs, thus yielding good overall results for FST-PRJ and FST-PRJ1. This also shows that the multivariate magnitude outliers in Model 1 remained magnitude outliers after the projection procedure since only the magnitude indices were activated. FST-STR, which used the stringing procedure described in Subsection 5.1.2, also showed a very high TPR with low FPR on this magnitude model. In Model 2, which contained non-persistent magnitude outliers, FST-PRJ and FST-PRJ1 show very good OD performance but this time powered by their shape indices (FST-PRJ-SH and FST-PRJ1-SH). FAO however struggled on this model with less than 90% TPR and high standard deviation. Both FOM and FAO did not perform well in Model 3, while FST-PRJ and FST-PRJ1 showed excellent OD performance on this model, helped by their amplitude and shape indices (FST-PRJ-SH, FST-PRJ-AM, FST-PRJ1-SH, and FST-PRJ1-AM). This reiterates that outlier classification is not necessarily disjoint because on the average, all the outliers in Model 3 were flagged as both amplitude and shape outliers. The "non-disjoint" classification of outliers can also be observed in the results of FST-PRJ and FST-PRJ1 on Models 5 and 6. All the methods showed poor TPRs (and FPRs) in Model 4, because Model 4 contains pure shape outliers that follow the overall trend of the data and are hidden within the bulk of the data. Apart from Model 4, MSPLOT maintains an excellent OD performance across all other models, except for Model 6 where it did not perform quite as well with a TPR of 95% compared to 100% for FST-PRJ1 and FST-PRJ.

Most of the simulation models used in this study have outliers outlying in all three dimensions (except for Model 6). In the Supplementary Material (Section B.3), we show the performance of the presented methods with similar simulation models but with the outliers only outlying in one or two of the three dimensions of the functional data. In addition, more simulation results for different contamination rates are presented in Section B.4 of the Supplementary Material.

To summarise, FST-PRJ and FST-PRJ1 showed the best performance across all tested simulation models. We recommend FST-PRJ1 in most usual applications because the underlying data distribution will be unknown, and it will consequently be impossible to compute the threshold Q for FST-PRJ.

5.3 Data Examples

In this section, we apply the Fast-MUOD extension using random projections described in Subsection 5.1.3 to detect outliers in two multivariate functional datasets: character

and video datasets.

5.3.1 Characters Dataset

The character dataset comprises bivariate functional data of trajectories of a pen tip along the x and y axes while a subject repeatedly writes various letters of the English alphabet. The original data were provided as part of the Character Trajectories dataset on the UCI machine learning repository (Williams et al., 2006). The versions of the dataset used in this study are for the letters "i" (without the dot) and "a" provided in the **mrfDepth** R package (Segaert et al., 2020). For the letter "i", the dataset consists of $n_i = 174$ bivariate functions, observed at 100 times points, whereas for letter the "a", there are $n_a = 171$ bivariate functions observed at the same number of time points. The bivariate functions in both datasets are the vertical and horizontal coordinates of the pen tip while the subject wrote n_i or n_a copies of each corresponding letter. The first row of Figures 5.3 and 5.6 show the bivariate functions for letters "i" and "a", respectively.

Plotting the vertical coordinates against the horizontal coordinates in both datasets reveals the handwritten characters (Figures 5.3 and 5.6). The aim is to use the Fast-MUOD via projections (FST-PRJ1) to detect outliers in both datasets. For each dataset, we generated 60 random unit vectors in $\mathbb{R}^2$ (entries of the vectors follow the standard normal distribution and each vector is normalised to have a unit norm) and projected the data. Fast-MUOD was then applied on the projections and we used a threshold triple of $Q = (\tau_S, \tau_A, \tau_M) = (0.4, 0.3, 0.3)$ to determine which observations were flagged as outliers of the various types.

The Letter "i"

For the letter "i", curves 41 and 46 were flagged as magnitude outliers; curves 39, 46, and 67 were flagged as amplitude outliers and curves $3, 5, 6, 9, 35, 39, 40, 41, 46, 64, 90$, and 102 were flagged as shape outliers. Thus, 13 unique outliers were flagged in total. All magnitude outliers are also shape outliers, and curve 46 is an outlier of all types. The "magnitude" and "amplitude" outliers are shown in the bottom left plot of Figure 5.3 (and Figure B.10 of the Supplementary Material shows plots of their horizontal and vertical coordinates). Curve 41 deviates from the overall trend of the data while curve 46 does not have enough "follow through" compared to other curves. Some of the shape outliers are shown on the bottom right plot of Figure 5.3. Like curve 41, curves 40 and 64 deviate from the overall trend of the data, while the curve 102 looks rather similar to a "v" instead of an "i". Although curve 9 seems to follow the overall trend of the data and does not appear to be an outlier, a closer look at its horizontal and vertical coordinate

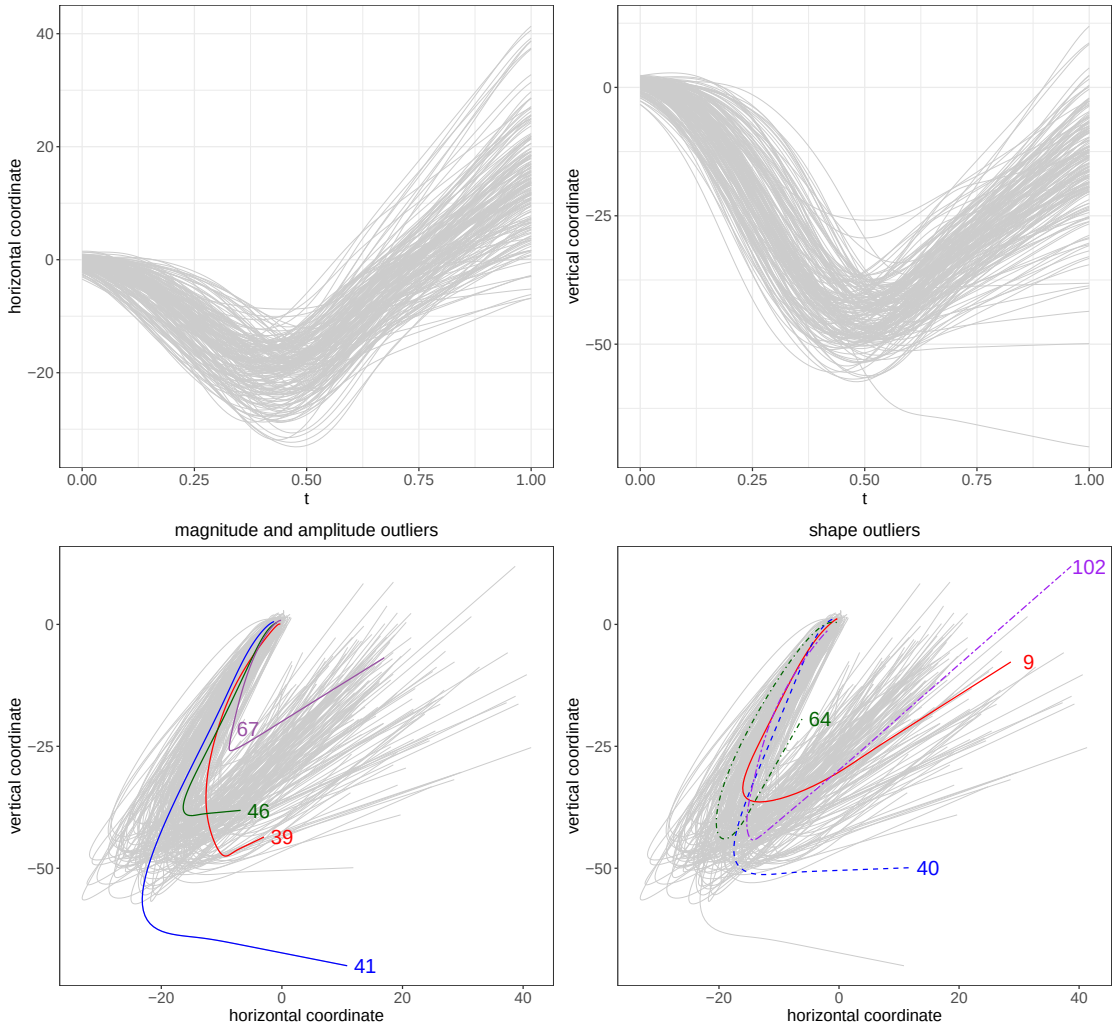

Figure 5.3: First Row: Horizontal and vertical trajectories for letter "i" data. Second Row: All magnitude outliers and some shape outliers detected in letter "i" data.

curves (see Figure B.11 of the Supplementary Material) reveals that the minimum points of both curves are horizontally shifted (to the right) compared to other curves; hence, it was flagged as a shape outlier. Curves 3, 5, 6, and 90 (shown in Figure B.11 of the Supplementary Material) were also flagged as outliers for this same reason.

For comparison, we applied MSPLOT on the character dataset for letter "i". MSPLOT discovered a total of 18 unique outliers. The curves flagged by MSPLOT were: 3, 5, 6, 9, 11, 12, 14, 39, 40, 41, 67, 73, 90, 102, 109, 110, 111, and 141. Among the 13 unique outliers flagged by Fast-MUOD, 10 were flagged by MSPLOT. The functions flagged as outliers by only Fast-MUOD are curves: 35, 46, and 64; while those flagged by only MSPLOT are curves: 11, 12, 14, 73, 109, 110, 111, and 141. These curves are shown in Figure 5.4.

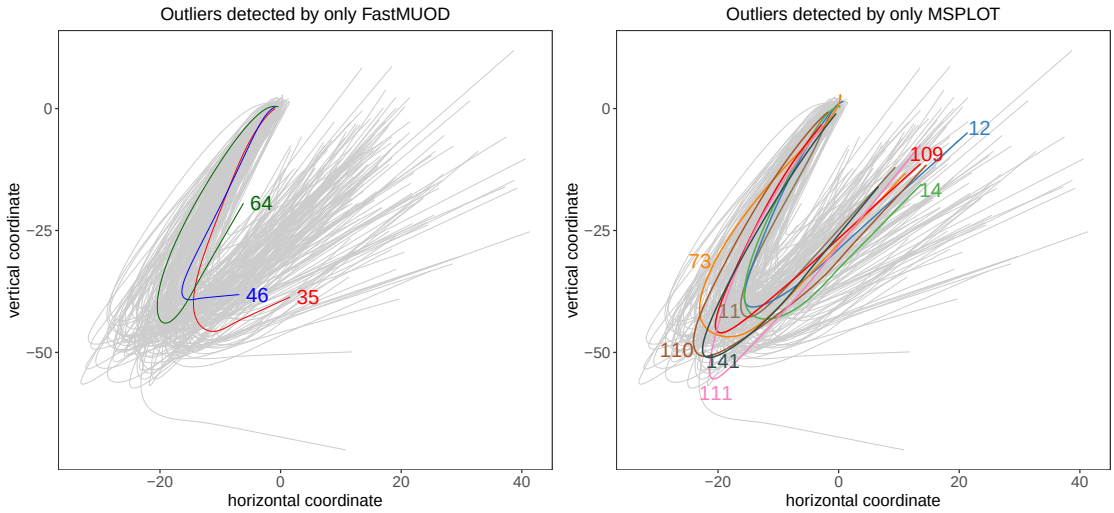

Figure 5.4: Outliers detected by only Fast-MUOD and only MSPLOT.

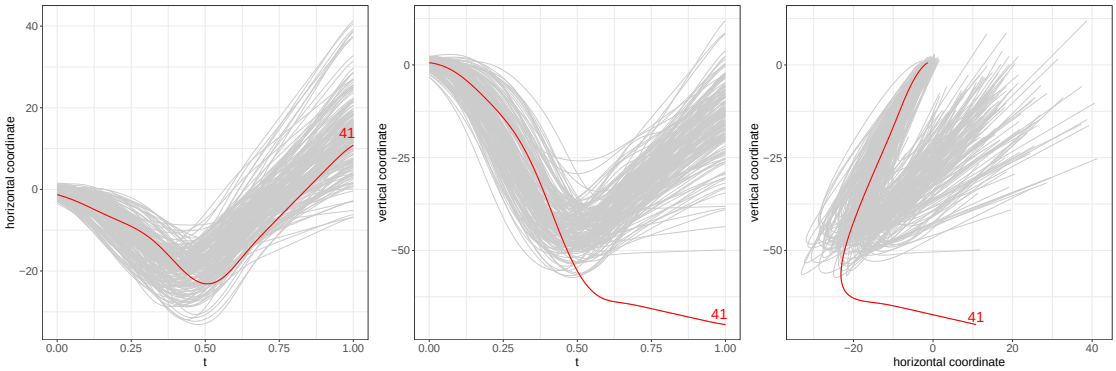

Figure 5.5: Curve 41, the only outlier detected by FOM.

Considering the curves detected by only Fast-MUOD and MSPLOT reveals certain interesting features in the data. Curves 35, 46 and 64, detected by only Fast-MUOD, clearly show some deviation from the trend of the data, especially in their follow-throughs. On the other hand, curve 111, flagged by only MSPLOT, seem to resemble a slanted "v" rather than an "i". To summarise, MSPLOT seems to be more aggressive in declaring curves as outliers compared to Fast-MUOD.

Finally, we compared the results obtained to those of FOM, which only flagged curve 41 (shown in the bottom left plot of Figure 5.3) as an outlier, probably because FOM is more suited to detecting magnitude outliers rather than shape outliers. Curve 41 demonstrates a clear magnitude deviation in its vertical axis (Figure 5.5).

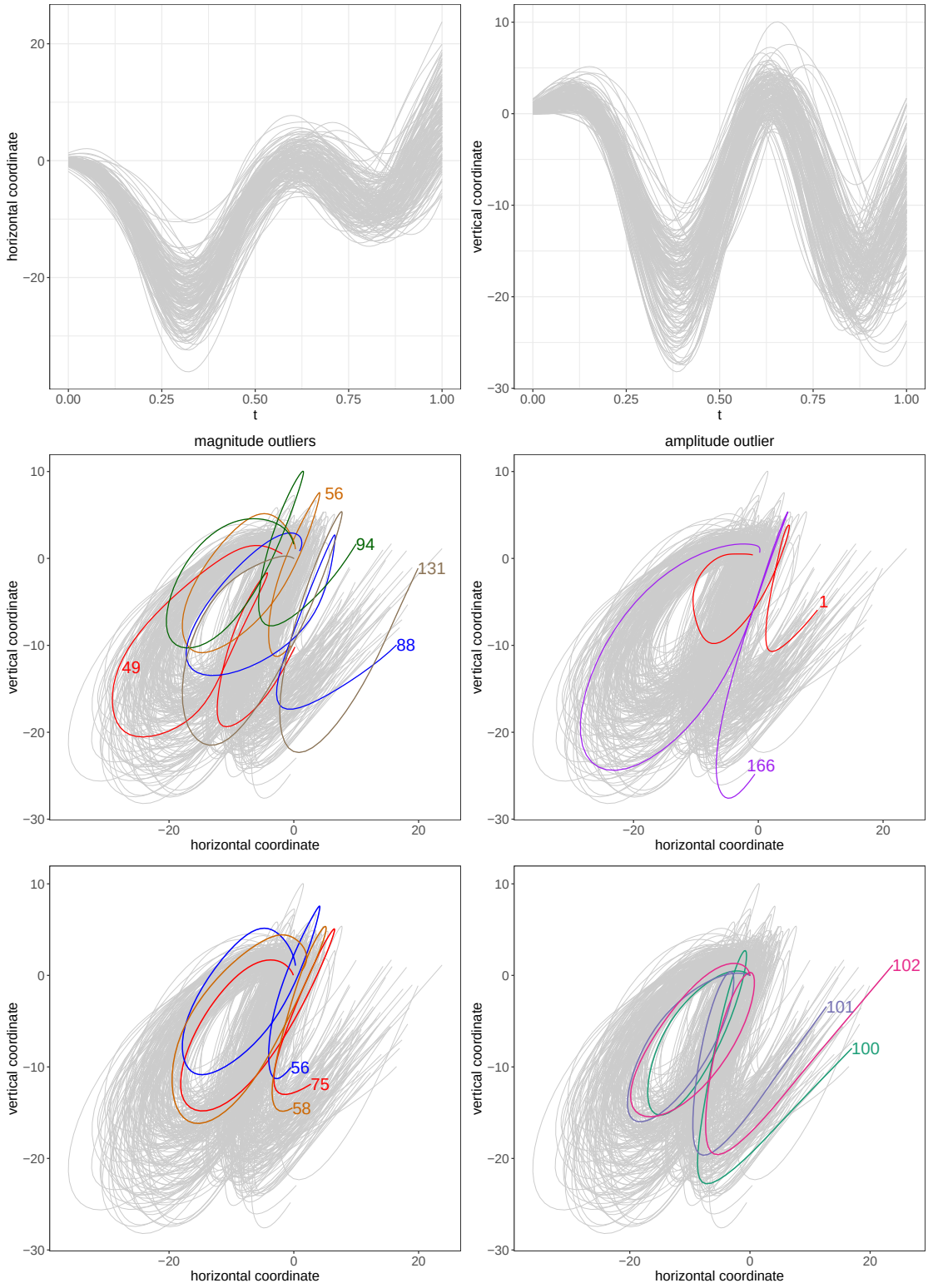

Figure 5.6: First Row: Horizontal and vertical trajectories for letter "a" data. Second Row: Magnitude and amplitude outliers detected by Fast-MUOD (FST-PRJ1). Third Row: Shape outliers with short (left) and long (right) "follow-throughs" respectively.

Letter "a"

For the letter "a", FST-PRJ1 flagged 17 curves as outliers. Curves: 49, 56, 88, 94, and 131 were flagged as magnitude outliers. Curves 1 and 166 were flagged as amplitude outliers, while curves 21, 56, 58, 75, 100, 101, 102, 114, 117, 125, and 136 were flagged as shape outliers. The curves flagged as magnitude and amplitude outliers (shown in the middle row of Figure 5.6) show a shift, either in the vertical, horizontal, or both axes.

Some of the flagged shape outliers have their peaks (or turning point) shifted to the right, particularly in the vertical axis, compared to the bulk of the data, resulting in letter "a"s with very small follow-through (when both axes are plotted) compared to the bulk of the data. Some of these functions are shown in the bottom row of Figure 5.6. On the other hand, some of the shape outliers have their peaks shifted to the left, which results in letter "a"s with a long follow through compared to the bulk of the data, thereby making the corresponding letters look like a "q" rather than an "a". These functions are also shown in Figure 5.6 (see Figures B.12 and B.13 of the Supplementary Material for the plots of the horizontal and vertical coordinates of these curves).

We applied MSPLOT on the data for letter "a" and MSPLOT flagged 12 unique outliers compared to the 17 outliers flagged by Fast-MUOD (FST-PRJ1). The outliers flagged by MSPLOT are the curves: 1, 21, 23, 49, 56, 58, 75, 102, 114, 125, 131,and 158. Among these 12 unique outliers, 10 were also flagged by FST-PRJ1, indicating a good overlap between the outliers flagged by MSPLOT and Fast-MUOD. Curves 23 and 158 were flagged as outliers by MSPLOT but not by Fast-MUOD, while the curves 88, 94, 100, 101, 117, 136, 166 were flagged by Fast-MUOD but not by MSPLOT. Some of these curves are shown in Figure B.14 of the Supplementary Material. FOM detected only 3 unique outliers. These are curves 23, 56, and 58, which can be seen in Figure 5.6 (and Figures B.12 and B.14 of the Supplementary Material).

5.3.2 Video Data

In the second application, we applied FST-PRJ1 on a surveillance video data named "WalkByShop1front" (made available by the EC Funded CAVIAR project/IST 2001 37540 at: homepages.inf.ed.ac.uk/rbf/CAVIAR/). The video consists of a 94 seconds long recording of a surveillance camera in front of a clothing shop in a shopping mall in Lisbon. At various time stamps in the course of the video clip, people passed by the front of the shop; sometimes they entered the shop to explore the products too. The aim is to identify time frames during which people passed by or entered the shop. This video dataset was analysed in Ojo et al. (2021a); they converted the video to greyscale, and used the original Fast-MUOD to analyse the resulting univariate functional data (with

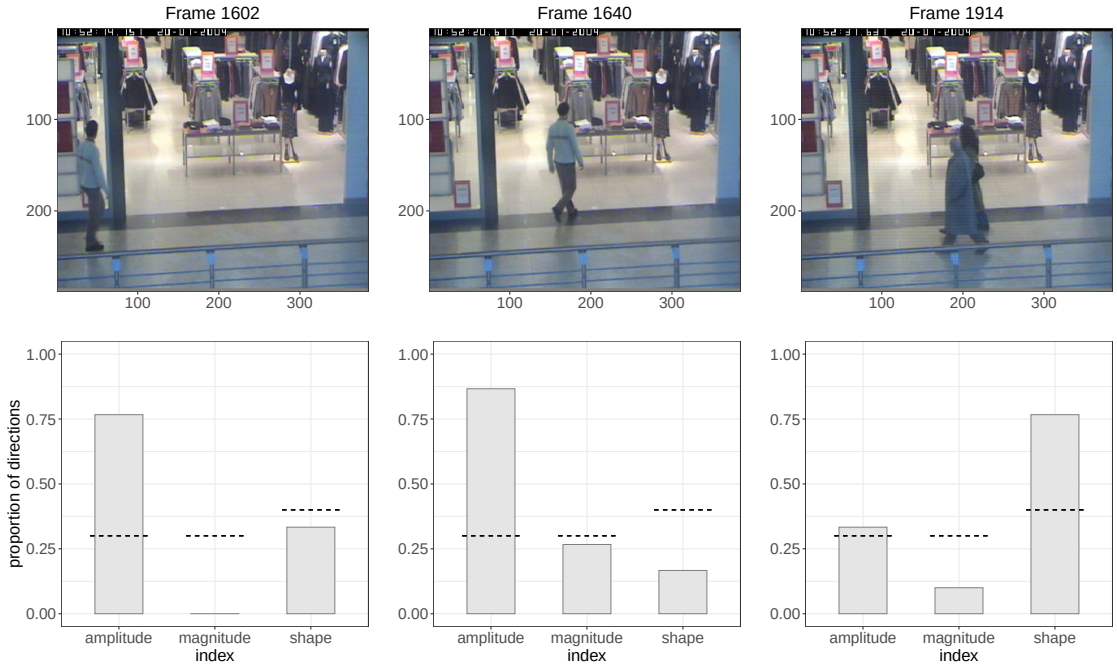

Figure 5.7: Some selected frames detected as outliers by FST-PRJ1. The bar charts below each frame show the proportion of projections in which that corresponding frame was flagged as an outlier of a particular type. The dotted lines indicate threshold values in Q.

each frame represented as a function and each pixel being an evaluation point on the curve). Because the original video is coloured, some information is lost in the conversion to grayscale. We represent the coloured video as trivariate functional data with each dimension being the RBG values of each pixel. We aim to apply FST-PRJ1 to the trivariate functional data and compare the performance to the unviariate analysis of the greyscale values done in Ojo et al. (2021a).

The video clip is provided at 25 frames/seconds and there are a total of 2,359 frames. The resolution of the video is 384×288, and therefore each frame contains $384 \times 288 = 110{,}592$ pixels. For each frame, we arranged the RBG pixel values into an array of size $110{,}592 \times 3$. Thus, the trivariate functional dataset is of dimension $2{,}359 \times 110{,}592 \times 3$ representing 2,359 functions (the frames) evaluated at 110,592 points (the pixels) where the value of each point is a vector in $\mathbb{R}^3$ (the RGB pixels intensity). Then, we projected the constructed trivariate functional data on 30 random unit vectors in $\mathbb{R}^3$ and applied Fast-MUOD (FST-PRJ1) on the 30 univariate functional data of size $2{,}359 \times 110{,}592$. We then set the threshold triple to $Q = (0.4, 0.3, 0.3)$.

In total, 356 unique frames were flagged as outliers with 213, 270, and 226 frames flagged as shape, amplitude, and magnitude outliers, respectively (Figure 5.7). A total

of 143 frames were flagged as outliers of all types. The 356 unique outliers flagged are an improvement over the 294 unique frames detected as outliers in the previous analysis of the greyscale pixel values performed by Ojo et al. (2021a). Similar to the analysis in Ojo et al. (2021a), all the frames flagged as outliers correspond to frames during which people pass by or enter the shop. This improvement underlines the advantage of using the multivariate data of the video data compared to performing a univariate analysis of the greyscale values as done in Ojo et al. (2021a).

To evaluate the performance of Fast-MUOD (with projections) in detecting the video frames of interest, it is necessary to understand the distribution of the outlying video frames. The video itself contains three major segments during which various people passed by or entered the front of the shop. The first segment contains frames 804 – 908, during which a woman passed by the front of the shop. The second segment contains frames 1,588 – 2,000, when a man entered the shop (to check the products on sale) and two other women passed by the shop. The third segment contains frames 2,073 – 2,359, which show another man entering the shop. All frames detected as outliers are within the frames of the three main segments, and so there are no FPs. However, similar to the results obtained by Ojo et al. (2021a), there are pockets of timestamps in these segments not flagged as outliers. For instance, the first frames detected as outliers are in the frames of the first segment (frames 803 – 908) with Fast-MUOD flagging frames 815, 830 – 851, 855 – 857, 864, and 881 – 903 as outliers while missing some frames at the beginning (frames 804–814 and 816 - 830), middle (frames 858 – 863 and 865 – 880), and end (frames 904 – 908) of this segment. We observed the same behaviour for the second and third segments, with certain pockets of a few frames in the beginning, middle and end of the segments not flagged as outliers. Usually, the pocket of frames not flagged as outliers in the middle of the segments correspond to timestamps when someone enters the shop and stands beside the products on display, yielding insufficient contrast in the pixel values of the person and the products on display in the shop. This is shown in Figure 5.8, which shows some frames in the second segment of outlying frames that were not flagged as outliers. The first frame, frame 1,597, shows when a man just entered the camera view. The second frame, frame 1,700, shows the same man in the store checking out the products. Figure 5.7, on the other hand, shows some selected frames in the second segment that were flagged as outliers. In addition to frames shown in Figures 5.7 and 5.8, the proportion of directions in which the frames are detected as outliers of each type are also shown.

Since the frames detected as outliers depend on the threshold triple Q, it is useful to visualise the frames of the video together with an animation of the proportion of directions in which the frames are flagged as outliers of different types. Such an animation

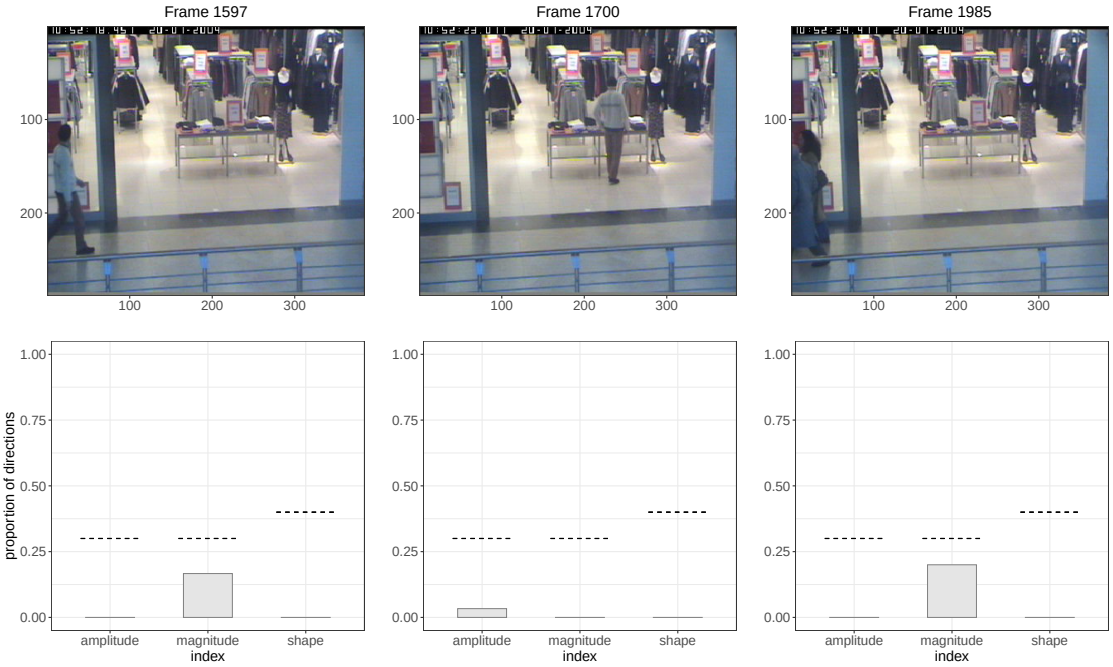

Figure 5.8: Some selected frames not detected as outliers by FST-PRJ1. The bar charts below each frame show the proportion of projections in which that corresponding frame was flagged as an outlier of a particular type. The dotted lines indicate threshold values in Q.

can be seen by clicking on this [link] and it shows the variation in the proportion of directions in which frames are outlying as people pass by or enter the shop.

For comparison, we applied MSPLOT on the same video data. MSPLOT flagged 1,001 frames (out of the 2,359 frames in the data) as outliers; although most frames in the three segments of interest were flagged as outliers, about 200 additional frames that are clearly out of the outlying segments were detected as outliers. However, FOM performed very well on the data, flagging 774 outliers with all the flagged frames coming from the three outlying segments in the video data. FOM excels in the analysis of image and video data because it computes a directional outlyingness at each grid point of the functional data, and image and video data usually consist of thousands of gridpoints (or pixels) per observation (image or frame). Only a few frames from the beginning and end of the outlying segments were not flagged by FOM.

To briefly examine the computational burden of the three methods, Table 5.2 shows the computational time for each method to analyse the video data. FOM is the fastest requiring about 47 minutes to complete the analysis. Fast-MUOD with 30 projections used about 51 minutes, while MSPLOT required over 679 minutes (>11 hours) to complete the analysis. Although FOM had the fastest running time, it also required the largest

amount of random access memory (RAM) for the analysis. MSPLOT and Fast-MUOD ran in a computer with 64 GB memory, while FOM required $\geq$ 64 GB memory to complete the analysis. The running time experiment was performed on an Ubuntu Server containing an AMD Opteron CPU containing 64 Cores (each running at 2.3 GHz) with 512 GB of RAM. The codes used in this experiment are those provided with the published papers on the methods or those provided on the website of the authors without any prior optimization.

Table 5.2: Computational time in minutes for the video data

Method	Times (Minutes)
FOM	47.4
Fast-MUOD	51.1
MSPLOT	679.7

5.4 Discussion

The Fast-MUOD indices, introduced by Ojo et al. (2021a) are useful and scalable tools for OD for functional data. However, their use presented in Ojo et al. (2021a) was limited to univariate functional data. We sought to address that in this work by presenting three techniques for using these indices for outlier detection in multivariate functional data settings.

Among the various proposed techniques, using random projections showed the most effective results. This involves projecting the multivariate functional data of interest on different unit vectors and then applying Fast-MUOD indices on the resulting projected univariate functional data. Then, an observation is flagged as an outlier if it is detected as an outlier in at least a fixed proportion of the projection directions.

We demonstrated the proposed methods on various simulated and real datasets and compared their performance to other multivariate functional OD methods. Our simulation results show the need for adequate selection of a threshold triple $Q = (\tau_S, \tau_A, \tau_M)$ of the proportion of projection directions used in determining whether an observation is an outlier (of a particular type). Instead of declaring an observation as an outlier if it is detected as an outlier in any direction (possibly resulting in a high FPR), carefully selecting the threshold helps to control the FPR. A possible direction for improvement is to develop a method to select these threshold values even when the distribution and base model of the data are unknown. With the proposed techniques, the Fast-MUOD indices add to the available options of OD tools for multivariate functional data.

Chapter 6

Concluding Remarks and Future Work

This thesis extends the outlier detection in functional data literature with three main contributions. The second chapter of the thesis presents the **fdaoutlier** R package. The package implements some of the state-of-the-art methods for outlier detection in functional data like directional outlyingness and MS-Plot, total variation depth and modified shape similarity index (TVD and MSS), sequential transformations and the massive unsupervised outlier detection (MUOD). A thorough review of various methods implemented were presented coupled with clear usage examples and use cases.

The third chapter then builds on Chapter 2 by proposing Semifast-MUOD and Fast-MUOD for outlier detection in univariate functional data. These methods detect outliers by computing for each functional observation, a magnitude, amplitude, and shape index, which target the corresponding types of outliers. Functional observations with any extremely high index are then identified using a boxplot cutoff method (applied on the indices), such observation is classified as an outlier. Because the three indices are independently checked for outliers, identified functional observations are also classified, unsupervised, as a by-product of the outlier detection process, without the need for visualization. This is useful when it is difficult to visualize the data. We then explored the theoretical properties of the Fast-MUOD indices under simple transformations in Chapter 4, and extended Fast-MUOD for outlier detection in multivariate functional data in Chapter 5. The proposed methods were applied to simulated and real data examples including video, handwriting, weather and population growth data.

Some interesting further research work arising from the contributions of this thesis are:

- Further development of **fdaoutlier** to include more recent functional outlier detec-

tion methods that are not yet implemented. For instance, we are already working on including Fast-MUOD and Semifast-MUOD in **fdaoutlier**.

- In our work, we focused mainly on detecting outliers with the Fast-MUOD (and Semifast-MUOD) indices. It will be interesting to use these indices for further statistical analysis of functional data, e.g., hypothesis testing, classification, and clustering. Initial findings on the use of the Fast-MUOD indices for classification tasks in functional data show promising results

- Our current analysis has focused on univariate and multivariate functional data defined on an interval $[a, b] \subset \mathbb{R}$. It is of interest to explore generalization of the indices to functional data defined over more complex structures (e.g., subsets of $\mathbb{R}^p$, $p \in \mathbb{N}$, or manifolds).

- Delve into more theoretical properties of both the indices and the methods developed in the thesis.

- Explore more complex datasets that can be considered as functional data and work on the interpretation of the outliers found.

Bibliography

Arribas-Gil, A. and Romo, J. (2014). Shape outlier detection and visualization for functional data: the outliergram. *Biostatistics*, 15(4):603–619.

Azcorra, A., Chiroque, L. F., Cuevas, R., Anta, A. F., Laniado, H., Lillo, R. E., Romo, J., and Sguera, C. (2018). Unsupervised scalable statistical method for identifying influential users in online social networks. *Scientific Reports*, 8(1).

Barnett, V. and Lewis, T. (1994). *Outliers in Statistical Data*. John Wiley & Sons.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104.

Brys, G., Hubert, M., and Rousseeuw, P. J. (2005). A robustification of independent component analysis. *Journal of Chemometrics*, 19(5-7):364–375.

Carling, K. (2000). Resistant outlier rules and the non-gaussian case. *Computational Statistics & Data Analysis*, 33(3):249–258.

Claeskens, G., Hubert, M., Slaets, L., and Vakili, K. (2014). Multivariate functional half-space depth. *Journal of the American Statistical Association*, 109(505):411–423.

Cox, M. A. A. and Cox, T. F. (2008). Multidimensional scaling. In *Handbook of Data Visualization*, pages 315–347. Springer Berlin Heidelberg.

Cuevas, A. (2014). A partial overview of the theory of statistics with functional data. *Journal of Statistical Planning and Inference*, 147:1–23.

Dai, W. and Genton, M. G. (2018). Multivariate functional data visualization and outlier detection. *Journal of Computational and Graphical Statistics*, 27(4):923–934.

Dai, W. and Genton, M. G. (2019). Directional outlyingness for multivariate functional data. *Computational Statistics & Data Analysis*, 131:50–65.

Dai, W., Mrkvička, T., Sun, Y., and Genton, M. G. (2020). Functional outlier detection and taxonomy by sequential transformations. *Computational Statistics & Data Analysis*, 149:106960.

Eddelbuettel, D. and Francois, R. (2011). **Rcpp**: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18.

Febrero, M., Galeano, P., and González-Manteiga, W. (2007). A functional analysis of NOx levels: location and scale estimation and outlier detection. *Computational Statistics*, 22(3):411–427.

Febrero, M., Galeano, P., and González-Manteiga, W. (2008). Outlier detection in functional data by depth measures, with application to identify abnormal NOx levels. *Environmetrics*, 19(4):331–345.

Febrero-Bande, M. and de la Fuente, M. O. (2012). Statistical computing in functional data analysis: The R package **fda.usc**. *Journal of Statistical Software*, 51(4):1–28.

Ferraty, F. and Vieu, P. (2006). *Nonparametric Functional Data Analysis*. Springer-Verlag GmbH.

Filzmoser, P., Garrett, R. G., and Reimann, C. (2005). Multivariate outlier detection in exploration geochemistry. *Computers & Geosciences*, 31(5):579–587.

Fraiman, R. and Muniz, G. (2001). Trimmed means for functional data. *Test*, 10(2):419–440.

Fritz, H., Filzmoser, P., and Croux, C. (2011). A comparison of algorithms for the multivariate l 1-median. *Computational Statistics*, 27(3):393–410.

Han, L. S. (2011). **rainbow**: An R package for visualizing functional time series. *The R Journal*, 3(2):54.

Happ, C. and Greven, S. (2018). Multivariate functional principal component analysis for data observed on different (dimensional) domains. *Journal of the American Statistical Association*, 113(522):649–659.

Happ-Kurz, C. (2020). Object-oriented software for functional data. *Journal of Statistical Software*, 93(5).

Hardin, J. and Rocke, D. M. (2005). The distribution of robust distances. *Journal of Computational and Graphical Statistics*, 14(4):928–946.

Hawkins, D. M. (1980). *Identification of Outliers*. Springer Netherlands.

Herrmann, M. and Scheipl, F. (2021). A geometric perspective on functional outlier detection. *Stats*, 4(4):971–1011.

Hsing, T. and Eubank, R. (2015). *Theoretical Foundations of Functional Data Analysis, with an Introduction to Linear Operators*. John Wiley & Sons, Ltd.

Huang, H. and Sun, Y. (2019). A decomposition of total variation depth for understanding functional outliers. *Technometrics*, 61(4):445–458.

Hubert, M., Rousseeuw, P. J., and Segaert, P. (2015). Multivariate functional outlier detection. *Statistical Methods & Applications*, 24(2):177–202.

Hubert, M. and Van der Veeken, S. (2008). Outlier detection for skewed data. *Journal of Chemometrics*, 22(3-4):235–246.

Hubert, M. and Vandervieren, E. (2008). An adjusted boxplot for skewed distributions. *Computational Statistics & Data Analysis*, 52(12):5186–5201.

Hyndman, R. J. and Shang, H. L. (2010). Rainbow plots, bagplots, and boxplots for functional data. *Journal of Computational and Graphical Statistics*, 19(1):29–45.

Hyndman, R. J. and Ullah, M. S. (2007). Robust forecasting of mortality and fertility rates: A functional data approach. *Computational Statistics & Data Analysis*, 51(10):4942–4956.

Ieva, F., Anna, M. P., Romo, J., and Tarabelloni, N. (2019). roahd package: Robust analysis of high dimensional data. *The R Journal*, 11(2):291.

Ieva, F. and Paganoni, A. M. (2020). Component-wise outlier detection methods for robustifying multivariate functional samples. *Statistical Papers*, 61(2):595–614.

Izrailev, S. (2021). ***tictoc**: Functions for Timing R Scripts, as Well as Implementations of Stack and List Structures*. R package version 1.0.1.

Kokoszka, P. and Reimherr, M. (2017). *Introduction to Functional Data Analysis*. Chapman and Hall/CRC.

Kuhnt, S. and Rehage, A. (2016). An angle-based multivariate functional pseudo-depth for shape outlier detection. *Journal of Multivariate Analysis*, 146:325–340.

Long, J. P. and Huang, J. Z. (2015). A study of functional depths.

López-Pintado, S. and Romo, J. (2009). On the concept of depth for functional data. *Journal of the American Statistical Association*, 104(486):718–734.

López-Pintado, S. and Romo, J. (2011). A half-region depth for functional data. *Computational Statistics & Data Analysis*, 55(4):1679–1695.

Myllymäki, M., Mrkvička, T., Grabarnik, P., Seijo, H., and Hahn, U. (2017). Global envelope tests for spatial processes. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 79(2):381–404.

Nagy, S., Gijbels, I., and Hlubinka, D. (2017). Depth-based recognition of shape outlying functions. *Journal of Computational and Graphical Statistics*, 26(4):883–893.

Nagy, S., Gijbels, I., Omelka, M., and Hlubinka, D. (2016). Integrated depth for functional data: statistical properties and consistency. *ESAIM: Probability and Statistics*, 20:95–130.

Narisetty, N. N. and Nair, V. N. (2016). Extremal depth for functional data and applications. *Journal of the American Statistical Association*, 111(516):1705–1714.

Nieto-Reyes, A. and Battey, H. (2016). A topologically valid definition of depth for functional data. *Statistical Science*, 31(1).

Ojo, O. T., Anta, A. F., Lillo, R. E., and Sguera, C. (2021a). Detecting and classifying outliers in big functional data. *Advances in Data Analysis and Classification*.

Ojo, O. T., Lillo, R. E., and Fernandez Anta, A. (2021b). ***fdaoutlier**: Outlier Detection Tools for Functional Data Analysis*. R package version 0.2.0.

Pokotylo, O., Mozharovskyi, P., and Dyckerhoff, R. (2019). Depth and depth-based classification with R package **ddalpha**. *Journal of Statistical Software*, 91(5).

R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ramsay, J. and Silverman, B. W. (2006). *Functional Data Analysis*. Springer-Verlag GmbH.

Ramsay, J. O. (1982). When the data are functions. *Psychometrika*, 47(4):379–396.

Ramsay, J. O., Graves, S., and Hooker, G. (2022). ***fda**: Functional Data Analysis*. R package version 6.0.3.

Rousseeuw, P. J. and Driessen, K. V. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223.

Rousseeuw, P. J., Raymaekers, J., and Hubert, M. (2018). A measure of directional outlyingness with applications to image data and video. *Journal of Computational and Graphical Statistics*, 27(2):345–359.

Segaert, P., Hubert, M., Rousseeuw, P., and Raymaekers, J. (2020). ***mrfDepth**: Depth Measures in Multivariate, Regression and Functional Settings*. R package version 1.0.13.

Sguera, C., Galeano, P., and Lillo, R. E. (2015). Functional outlier detection by a local depth with application to NO x levels. *Stochastic Environmental Research and Risk Assessment*, 30(4):1115–1130.

Sun, Y. and Genton, M. G. (2011). Functional boxplots. *Journal of Computational and Graphical Statistics*, 20(2):316–334.

Sun, Y., Genton, M. G., and Nychka, D. W. (2012). Exact fast computation of band depth for large functional datasets: How quickly can one million curves be ranked? *Stat*, 1(1):68–74.

Tukey, J. W. (1975). Mathematics and the picturing of data. In *Proceedings of the International Congress of Mathematicians, Vancouver, 1975*, volume 2, pages 523–531.

Vinue, G. and Epifanio, I. (2020a). ***adamethods**: Archetypoid Algorithms and Anomaly Detection*. R package version 1.2.1.

Vinue, G. and Epifanio, I. (2020b). Robust archetypoids for anomaly detection in big functional data. *Advances in Data Analysis and Classification*, 15(2):437–462.

Williams, B. H., Toussaint, M., and Storkey, A. J. (2006). Extracting motion primitives from natural handwriting data. In *Artificial Neural Networks – ICANN 2006*, pages 634–643. Springer Berlin Heidelberg.

Appendix A

Supplementary Material: Detecting and Classifying Outliers in Big Functional Data

A.1 Comparison between L_1 median and Point-wise median for Fast-MUOD

In this section, we present simulation results showing that the performance of Fast-MUOD using the point-wise median (FSTP) and the L_1 median (FSTL1) are similar. Note that FSTL1MAG considers magnitude outliers only, flagged by the magnitude index of FSTL1. The same applies to FSTL1SHA and FSTL1AMP (considering shape outliers and amplitude outliers only of FSTL1 respectively). Thus, FSTL1 is the union of outliers flagged by FSTL1MAG, FSTL1SHA, and FSTL1AMP. The same notation system is used for FSTP. The results can be found in Table A.1.

A.2 Contamination rate

Here, we present simulation results showing that the performance of the proposed methods as the contamination rate is increased up to 20%, Results are presented in Tables A.2 and A.3.

Table A.1: Mean and Standard Deviation (in parentheses) of the True Positive Rate (TPR) and the False Positive Rate (FPR) over eight simulation models comparing the point-wise median and the L_1 median for computing the Fast-MUOD Indices. Experiment setup include 500 repetitions with $n = 300$, $d = 50$, and $\alpha = 0.1$.

Method	Model1		Model 2		Model 3		Model 4	
	FPR	TPR	FPR	TPR	FPR	TPR	FPR	
FSTL1	9.91(1.53)	100.00(0.00)	8.94(1.53)	99.79(0.98)	6.10(1.37)	100.00(0.00)	3.15(1.10)	
FSTL1MAG	1.72(0.92)	99.99(0.15)	0.36(0.39)	4.23(3.66)	1.52(0.89)	41.99(9.76)	0.62(0.53)	
FSTL1SHA	7.95(1.37)	7.74(4.85)	7.94(1.43)	98.98(2.01)	4.35(1.14)	100.00(0.00)	2.25(0.94)	
FSTL1AMP	1.71(0.85)	1.67(2.39)	1.69(0.93)	6.39(4.70)	1.39(0.79)	55.74(11.72)	0.45(0.45)	
FSTP	9.90(1.50)	100.00(0.00)	8.95(1.59)	99.81(0.89)	6.10(1.37)	100.00(0.00)	3.15(1.13)	
FSTPMAG	1.74(0.92)	99.99(0.15)	0.36(0.40)	4.13(3.70)	1.52(0.89)	41.25(9.71)	0.63(0.54)	
FSTPSHA	7.94(1.34)	7.79(4.86)	7.94(1.48)	98.97(2.03)	4.36(1.12)	100.00(0.00)	2.24(0.95)	
FSTPAMP	1.70(0.83)	1.66(2.43)	1.69(0.93)	6.41(4.62)	1.38(0.80)	54.56(11.76)	0.45(0.45)	

Method	Model 5		Model 6		Model 7		Model 8	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
FSTL1	96.11(4.22)	5.67(1.18)	93.31(6.25)	6.33(1.37)	79.69(15.05)	6.56(1.89)	98.75(2.28)	6.64(1.41)
FSTL1MAG	16.02(6.73)	1.08(0.71)	0.81(1.65)	1.76(0.96)	1.64(2.34)	1.69(0.89)	30.65(8.12)	1.05(0.75)
FSTL1SHA	86.50(6.79)	4.40(1.11)	91.26(6.67)	4.37(1.13)	4.23(3.68)	4.95(1.69)	71.92(7.60)	5.31(1.29)
FSTL1AMP	23.10(7.96)	1.02(0.72)	3.53(3.60)	1.42(0.79)	79.03(15.59)	0.01(0.05)	10.71(5.72)	1.29(0.84)
FSTP	95.97(4.27)	5.67(1.19)	93.05(6.42)	6.31(1.35)	79.73(14.95)	6.55(1.91)	98.63(2.45)	6.65(1.40)
FSTPMAG	15.94(6.70)	1.08(0.71)	0.83(1.70)	1.77(0.94)	1.65(2.36)	1.69(0.90)	30.65(8.10)	1.04(0.75)
FSTPSHA	86.35(6.74)	4.39(1.12)	91.01(6.75)	4.35(1.10)	4.21(3.68)	4.94(1.72)	71.77(7.58)	5.31(1.29)
FSTPAMP	22.99(7.93)	1.01(0.71)	3.54(3.78)	1.40(0.79)	79.10(15.42)	0.01(0.05)	10.74(5.69)	1.29(0.83)

Table A.2: Mean and Standard Deviation (in parentheses) of the True Positive Rates (TPR) and False Positive Rate (FPR) over eight simulation models with 500 repetitions for each possible case. Each simulation is done with $n = 300$ and $d = 50$ and $\alpha = 0.15$. Comparatively high TPRs are in bold. Proposed methods in italics.

Method	Model 2		Model 3		Model 4		Model 5	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
FST	**100.00(0.00)**	8.89(1.58)	**96.48(4.37)**	4.31(1.14)	**99.98(0.20)**	1.38(0.77)	**88.65(7.05)**	3.82(1.13)
FSTMG	**100.00(0.00)**	0.09(0.20)	3.97(2.95)	1.40(0.84)	30.94(7.84)	0.34(0.40)	14.82(5.42)	0.91(0.66)
FSTSH	7.98(4.20)	8.02(1.55)	**94.01(5.36)**	2.65(0.87)	**99.98(0.22)**	0.94(0.63)	77.31(7.95)	2.78(0.95)
FSTAM	1.79(1.92)	1.69(0.97)	6.27(3.55)	1.22(0.76)	31.72(10.31)	0.14(0.24)	21.01(6.41)	0.72(0.60)
SF	**99.99(0.17)**	8.57(1.60)	**95.25(4.98)**	3.91(1.16)	**98.91(1.85)**	0.95(0.66)	84.76(7.43)	3.52(1.08)
SF25	**99.96(0.42)**	8.55(1.60)	**95.14(4.97)**	3.86(1.11)	**98.26(3.19)**	0.94(0.66)	84.57(7.52)	3.52(1.06)
MUOD	**99.62(4.90)**	8.18(3.92)	48.67(23.19)	9.87(4.57)	88.41(15.03)	3.77(3.50)	42.63(12.84)	3.52(2.70)
OGMBD	**100.00(0.10)**	4.72(1.46)	29.70(9.91)	2.97(1.06)	58.88(12.56)	0.50(0.45)	**89.67(5.40)**	0.94(0.66)
MSPLT	**99.93(0.39)**	2.39(1.21)	**100.00(0.00)**	2.47(1.16)	**99.80(0.70)**	1.15(0.77)	**99.98(0.20)**	2.39(1.17)
TVD	**99.99(0.14)**	0.00(0.02)	**100.00(0.00)**	0.00(0.00)	1.48(2.09)	0.00(0.00)	**100.00(0.00)**	0.00(0.02)
FOM	**100.00(0.00)**	0.01(0.07)	21.02(14.78)	0.02(0.09)	6.06(5.86)	0.01(0.05)	4.70(4.43)	0.01(0.07)
FAO	**100.00(0.10)**	0.00(0.03)	8.56(8.91)	0.00(0.04)	0.07(0.38)	0.00(0.04)	2.52(3.14)	0.00(0.03)
FOM2	**100.00(0.00)**	0.51(0.47)	**100.00(0.00)**	1.23(0.74)	34.47(10.73)	0.49(0.47)	**100.00(0.00)**	1.16(0.73)
FAO2	**100.00(0.00)**	0.70(0.56)	**100.00(0.00)**	0.89(0.67)	4.27(3.47)	1.39(0.87)	**100.00(0.00)**	0.89(0.67)
ED	**99.97(0.24)**	0.00(0.00)	**98.71(1.80)**	0.00(0.00)	0.00(0.00)	0.00(0.00)	21.52(7.10)	0.00(0.00)
SEQ1	**99.98(0.22)**	0.00(0.00)	**100.00(0.00)**	0.00(0.02)	6.35(4.97)	0.00(0.00)	**100.00(0.00)**	0.00(0.00)
SEQ2	**99.98(0.22)**	0.68(0.52)	**100.00(0.00)**	0.51(0.46)	24.55(11.66)	0.00(0.00)	81.52(6.46)	0.48(0.43)
SEQ3	**99.98(0.22)**	0.00(0.00)	**100.00(0.00)**	0.00(0.02)	3.46(3.21)	0.00(0.00)	**100.00(0.00)**	0.00(0.00)

Method	Model 6		Model 7		Model 8	
	TPR	FPR	TPR	FPR	TPR	FPR
FST	**79.53(9.87)**	4.58(1.32)	41.90(20.48)	6.75(2.03)	**96.45(3.47)**	5.08(1.28)
FSTMG	0.98(1.50)	1.74(1.04)	1.67(1.97)	1.68(0.93)	30.07(6.58)	0.81(0.63)
FSTSH	**77.73(9.75)**	2.67(0.89)	4.06(3.04)	5.16(1.82)	68.58(6.13)	3.95(1.15)
FSTAM	2.98(2.84)	1.18(0.80)	39.66(21.37)	0.00(0.00)	9.31(4.30)	1.20(0.80)
SF	**79.25(9.73)**	4.24(1.21)	34.30(15.56)	6.77(2.09)	**95.18(4.01)**	4.65(1.18)
SF25	**79.05(10.23)**	4.20(1.24)	33.84(16.75)	6.83(2.04)	**95.04(4.18)**	4.63(1.23)
MUOD	39.41(20.18)	11.68(4.77)	92.22(12.04)	18.09(6.89)	56.48(15.12)	3.75(2.44)
OGMBD	**98.72(2.08)**	0.75(0.58)	3.33(5.64)	0.00(0.00)	80.19(6.37)	2.15(0.96)
MSPLT	**100.00(0.00)**	2.44(1.24)	52.31(17.11)	0.02(0.10)	**100.00(0.10)**	2.43(1.24)
TVD	55.78(19.38)	0.00(0.00)	20.12(12.48)	0.00(0.00)	**98.59(2.12)**	0.00(0.02)
FOM	0.00(0.10)	0.02(0.09)	0.01(0.17)	0.00(0.00)	33.43(8.57)	0.02(0.09)
FAO	0.00(0.00)	0.00(0.04)	0.03(0.30)	0.00(0.00)	28.36(7.21)	0.00(0.04)
FOM2	39.80(16.33)	1.09(0.66)	5.30(5.49)	0.06(0.15)	**97.17(3.27)**	1.12(0.70)
FAO2	28.24(15.09)	0.78(0.62)	1.53(3.10)	0.05(0.15)	**92.58(4.88)**	0.89(0.66)
ED	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.00)	55.08(7.47)	0.00(0.00)
SEQ1	0.10(0.46)	0.00(0.00)	0.00(0.00)	0.00(0.00)	75.32(6.27)	0.00(0.00)
SEQ2	5.58(4.28)	0.48(0.43)	1.61(1.96)	0.01(0.05)	74.28(6.67)	0.55(0.48)
SEQ3	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.00)	74.78(6.30)	0.00(0.00)

Table A.3: Mean and Standard Deviation (in parentheses) of the True Positive Rates (TPR) and False Positive Rate (FPR) over eight simulation models with 500 repetitions for each possible case. Each simulation is done with $n = 300$ and $d = 50$ and $\alpha = 0.2$. Comparatively high TPRs are in bold. Proposed methods in italics.

Method	Model 2		Model 3		Model 4		Model 5	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
FST	**99.96(0.26)**	8.84(1.65)	73.15(13.34)	2.91(1.14)	**99.69(0.74)**	0.44(0.46)	75.13(8.18)	2.35(0.91)
FSTMG	**99.92(0.36)**	0.00(0.03)	3.72(2.59)	1.40(0.86)	22.86(6.32)	0.21(0.31)	13.92(4.77)	0.68(0.58)
FSTSH	7.89(3.46)	7.98(1.59)	69.49(13.42)	1.16(0.63)	**99.67(0.76)**	0.19(0.30)	62.95(7.98)	1.54(0.71)
FSTAM	1.62(1.70)	1.70(0.94)	5.41(3.05)	1.14(0.79)	13.12(6.18)	0.06(0.15)	19.26(5.40)	0.52(0.52)
SF	**99.74(1.14)**	8.56(1.60)	71.60(13.23)	2.55(1.07)	86.94(8.61)	0.20(0.29)	69.35(8.24)	2.15(0.88)
SF25	**99.46(1.95)**	8.54(1.67)	71.24(12.98)	2.54(1.10)	84.93(13.91)	0.20(0.31)	69.18(8.11)	2.14(0.88)
MUOD	**100.00(0.00)**	8.31(3.92)	40.59(21.34)	10.16(4.67)	74.23(18.99)	3.42(4.24)	37.97(11.39)	3.24(2.43)
OGMBD	**99.95(0.33)**	4.52(1.48)	22.58(8.24)	2.62(1.08)	7.21(5.31)	0.07(0.19)	75.17(8.95)	0.30(0.36)
MSPLT	**99.90(0.44)**	2.01(1.16)	**100.00(0.00)**	2.06(1.16)	**99.34(1.12)**	0.97(0.73)	**99.98(0.17)**	2.05(1.04)
TVD	**99.95(0.33)**	0.00(0.03)	**100.00(0.00)**	0.00(0.00)	0.75(1.26)	0.00(0.00)	**100.00(0.00)**	0.00(0.00)
FOM	**99.99(0.13)**	0.00(0.00)	4.63(7.73)	0.00(0.04)	0.51(1.06)	0.00(0.03)	1.39(1.87)	0.00(0.02)
FAO	**99.88(0.67)**	0.00(0.00)	1.30(3.08)	0.00(0.00)	0.05(0.28)	0.05(0.16)	0.55(1.17)	0.00(0.00)
FOM2	**100.00(0.00)**	0.14(0.26)	**100.00(0.00)**	0.72(0.60)	12.28(4.90)	0.50(0.51)	**100.00(0.00)**	0.53(0.51)
FAO2	**100.00(0.00)**	0.26(0.35)	**100.00(0.00)**	0.44(0.46)	1.39(1.72)	2.25(1.20)	**100.00(0.00)**	0.39(0.46)
ED	**99.93(0.53)**	0.00(0.00)	**97.60(2.48)**	0.00(0.00)	0.00(0.00)	0.00(0.00)	17.78(6.05)	0.00(0.00)
SEQ1	**99.96(0.27)**	0.00(0.00)	**100.00(0.00)**	0.00(0.00)	4.12(3.56)	0.00(0.00)	**100.00(0.00)**	0.00(0.00)
SEQ2	**99.96(0.27)**	0.67(0.52)	**100.00(0.00)**	0.44(0.41)	16.55(9.34)	0.00(0.00)	78.28(6.15)	0.46(0.41)
SEQ3	**99.96(0.27)**	0.00(0.00)	**100.00(0.00)**	0.00(0.00)	2.40(2.58)	0.00(0.00)	**100.00(0.00)**	0.00(0.00)

Method	Model 6		Model 7		Model 8	
	TPR	FPR	TPR	FPR	TPR	FPR
FST	55.81(9.97)	3.30(1.12)	9.32(6.13)	7.18(2.03)	**91.60(5.80)**	3.66(1.13)
FSTMG	0.91(1.21)	1.79(0.91)	1.76(1.75)	1.70(0.92)	28.88(5.48)	0.62(0.55)
FSTSH	54.20(9.69)	1.29(0.70)	4.32(2.62)	5.55(1.84)	63.16(6.13)	2.70(1.01)
FSTAM	2.62(2.30)	1.05(0.74)	4.20(6.36)	0.00(0.00)	9.20(3.87)	0.96(0.70)
SF	54.30(10.28)	3.07(1.08)	7.99(5.07)	7.16(2.03)	**89.58(6.17)**	3.21(1.09)
SF25	54.55(10.22)	3.05(1.06)	9.08(6.27)	7.20(2.09)	**89.59(6.21)**	3.23(1.08)
MUOD	33.77(19.12)	12.08(5.16)	85.97(14.26)	16.82(7.26)	51.12(13.75)	3.53(2.97)
OGMBD	**84.93(14.27)**	0.13(0.23)	0.13(0.65)	0.00(0.00)	76.86(6.27)	1.51(0.88)
MSPLT	100.00(0.00)	1.96(1.16)	38.82(16.58)	0.02(0.10)	**99.96(0.24)**	2.03(1.08)
TVD	12.77(12.73)	0.00(0.00)	2.89(5.08)	0.00(0.00)	**95.57(4.18)**	0.00(0.02)
FOM	0.00(0.00)	0.01(0.06)	0.00(0.00)	0.00(0.00)	27.57(6.55)	0.00(0.00)
FAO	0.00(0.00)	0.00(0.03)	0.00(0.00)	0.00(0.00)	25.33(5.84)	0.00(0.00)
FOM2	13.39(9.51)	0.50(0.48)	0.33(0.80)	0.01(0.08)	**92.76(5.06)**	0.56(0.53)
FAO2	8.26(6.89)	0.33(0.40)	0.14(0.85)	0.03(0.13)	86.48(5.47)	0.43(0.47)
ED	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.00)	53.28(6.71)	0.00(0.00)
SEQ1	0.10(0.48)	0.00(0.00)	0.00(0.00)	0.00(0.00)	75.53(5.80)	0.00(0.00)
SEQ2	4.49(3.51)	0.45(0.41)	1.65(1.61)	0.00(0.03)	73.00(6.24)	0.46(0.42)
SEQ3	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.00)	75.14(5.80)	0.00(0.00)

A.3 Sample Size and Evaluation Points

Here, we present simulation results showing that the performance of the proposed methods, Semifast-MUOD and Fast-MUOD, remain mostly similar, except for Model 7, even with lower sample size and evaluation points of $n = 100$ and $d = 25$ respectively. See Subsection 3.4.2 of the main manuscript for a description of the methods compared. Results are presented in Table A.4.

A.4 Correlation coefficients

We compare the effectiveness of different correlation coefficients that can be used in computing the shape index $I_S(Y_i, \tilde{Y})$ for Fast-MUOD. We considered Pearson, Spearman's rank, and Kendall's Tau correlation coefficients, in addition to the Cosine similarity index. We used Models 3, 4, and 6 all which have some form of shape outliers. We then recorded the True Positive Rate (TPR) and the False Positive Rate (FPR) (together with their standard deviations) of the outliers flagged by only the shape indices. Specifically, the methods considered are:

- FSTSH_PEARSON: An observation is an outlier only if it is flagged by Fast-MUOD as a shape outlier using $I_S(Y_i, \tilde{Y})$ computed with the Pearson's correlation coefficient.
- FSTSH_KENDALL: An observation is an outlier only if it is flagged by Fast-MUOD as a shape outlier using $I_S(Y_i, \tilde{Y})$ computed with the Kendall's Tau correlation coefficient.
- FSTSH_SPEARMAN: An observation is an outlier only if it is flagged by Fast-MUOD as a shape outlier using $I_S(Y_i, \tilde{Y})$ computed with Spearman's rank correlation coefficient.
- FSTSH_COSINE: An observation is an outlier only if it is flagged by Fast-MUOD as a shape outlier using $I_S(Y_i, \tilde{Y})$ computed with the Cosine similarity index.

Table A.5 shows the results of our simulation. The shape index $I_S(Y_i, \tilde{Y})$ computed using the Pearson correlation coefficient identifies more shape outliers, especially in Model 3 and Model 6.

A.5 Signal to Noise Ratio

Here we study the changes in the True Positive Rate (TPR) and the False Positive Rate (FPR) while considering different level of noise in the simulated data. To do this, we

Table A.4: Mean and Standard Deviation (in parentheses) of the True Positive Rate (TPR) and the False Positive Rate (FPR) over eight simulation models with 500 repetitions for each possible case with sample size $n = 100$, evaluation points $d = 25$, and contamination rate $\alpha = 0.1$. Comparatively high TPRs are marked in bold.

Method	Model1		Model 2		Model 3		Model 4	
	FPR	TPR	FPR	TPR	FPR	TPR	FPR	
FST	10.07(2.87)	**100.00(0.00)**	8.99(2.81)	**99.04(3.51)**	6.16(2.37)	**99.98(0.45)**	3.15(1.85)	
FSTMG	1.94(1.62)	**100.00(0.00)**	0.45(0.78)	4.66(6.59)	1.64(1.49)	39.70(16.82)	0.69(0.93)	
FSTSH	7.93(2.48)	7.70(8.73)	7.86(2.63)	**98.08(4.85)**	4.34(1.95)	**99.98(0.45)**	2.17(1.51)	
FSTAM	1.97(1.73)	2.04(4.63)	1.83(1.62)	6.70(8.33)	1.43(1.43)	50.26(19.82)	0.50(0.76)	
SF	9.76(2.83)	**99.96(0.63)**	8.62(2.81)	**98.50(4.43)**	5.77(2.24)	**99.90(1.00)**	2.68(1.69)	
SF25	9.72(2.76)	**99.58(3.85)**	8.77(2.87)	**98.30(4.92)**	5.62(2.22)	**99.34(3.37)**	2.66(1.72)	
MUOD	29.10(12.38)	99.64(3.89)	18.92(10.08)	85.70(19.89)	25.50(14.10)	**99.00(4.36)**	11.16(6.35)	
OGMBD	5.25(2.31)	**100.00(0.00)**	4.66(2.35)	51.96(17.84)	3.85(2.13)	**94.68(7.89)**	1.33(1.18)	
MSPLOT	2.97(2.37)	**99.28(2.88)**	2.12(1.86)	**100.00(0.00)**	2.11(1.98)	**99.82(1.33)**	0.90(1.16)	
TVD	0.04(0.22)	**100.00(0.00)**	0.02(0.16)	**100.00(0.00)**	0.02(0.16)	15.66(14.62)	0.00(0.00)	
FOM	0.69(1.01)	**100.00(0.00)**	0.09(0.39)	40.30(26.34)	0.14(0.46)	49.16(26.10)	0.12(0.43)	
FAO	0.32(0.68)	**100.00(0.00)**	0.06(0.32)	22.58(20.10)	0.04(0.22)	12.20(19.55)	0.03(0.20)	
FOM2	2.06(1.62)	**100.00(0.00)**	0.45(0.76)	**100.00(0.00)**	0.92(1.10)	66.64(20.43)	0.36(0.71)	
FAO2	1.68(1.59)	**100.00(0.00)**	0.64(0.99)	**100.00(0.00)**	0.66(0.99)	27.54(24.54)	0.29(0.63)	
ED	0.01(0.11)	**100.00(0.00)**	0.01(0.12)	**99.26(3.04)**	0.02(0.13)	5.22(8.74)	0.00(0.00)	
SEQ1	0.03(0.19)	**100.00(0.00)**	0.02(0.16)	**100.00(0.00)**	0.03(0.17)	38.96(20.61)	0.00(0.00)	
SEQ2	1.27(1.17)	**100.00(0.00)**	1.24(1.18)	**100.00(0.00)**	1.08(1.09)	69.76(18.96)	0.00(0.00)	
SEQ3	0.03(0.18)	**100.00(0.00)**	0.02(0.15)	**100.00(0.00)**	0.02(0.16)	18.76(14.49)	0.00(0.00)	

Method	Model 5		Model 6		Model 7		Model 8	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
FST	**94.18(8.37)**	5.93(2.26)	**91.92(11.89)**	6.37(2.32)	**73.42(26.24)**	6.96(3.17)	**97.56(5.52)**	6.80(2.38)
FSTMG	17.14(12.05)	1.30(1.31)	0.88(2.91)	1.77(1.63)	1.62(4.10)	1.83(1.67)	30.86(14.54)	1.18(1.30)
FSTSH	83.54(13.01)	4.47(1.90)	**90.12(12.44)**	4.38(1.82)	4.48(6.63)	5.20(2.87)	70.62(14.58)	5.24(2.14)
FSTAM	23.28(13.22)	1.12(1.23)	3.96(6.39)	1.50(1.38)	**72.56(26.80)**	0.03(0.21)	9.70(9.31)	1.42(1.38)
SF	**91.94(9.73)**	5.43(2.13)	**92.14(11.36)**	5.91(2.21)	62.74(26.88)	7.00(3.00)	**97.30(5.81)**	6.30(2.25)
SF25	**91.80(9.93)**	5.38(2.08)	**90.46(12.44)**	5.87(2.28)	60.36(27.69)	6.97(3.01)	**96.86(6.39)**	6.24(2.35)
MUOD	80.92(17.19)	16.44(10.97)	80.26(21.09)	26.72(12.57)	98.32(6.76)	33.58(12.18)	88.32(14.55)	15.26(11.95)
OGMBD	**96.68(5.89)**	2.18(1.62)	**99.88(1.09)**	2.05(1.57)	22.72(25.78)	0.00(0.07)	86.22(11.18)	2.85(1.86)
MSPLT	**99.42(2.42)**	2.04(1.75)	**100.00(0.00)**	2.16(1.95)	48.42(27.14)	0.03(0.20)	**99.72(1.65)**	2.09(1.86)
TVD	**100.00(0.00)**	0.03(0.17)	86.00(20.49)	0.02(0.15)	27.38(24.90)	0.00(0.05)	**97.98(5.23)**	0.02(0.16)
FOM	13.12(14.69)	0.12(0.39)	0.16(1.26)	0.16(0.45)	2.78(8.55)	0.01(0.09)	39.06(17.63)	0.15(0.47)
FAO	8.04(11.38)	0.06(0.27)	0.14(1.34)	0.06(0.24)	2.58(7.30)	0.01(0.11)	33.42(16.24)	0.06(0.27)
FOM2	**100.00(0.00)**	0.94(1.07)	**89.54(16.99)**	0.88(1.06)	17.40(19.86)	0.16(0.44)	**99.46(2.51)**	0.80(0.98)
FAO2	**100.00(0.00)**	0.63(0.93)	70.16(27.97)	0.65(0.92)	12.90(17.81)	0.15(0.47)	**95.94(7.20)**	0.62(0.91)
ED	31.16(16.07)	0.00(0.05)	0.14(1.34)	0.02(0.16)	0.00(0.00)	0.00(0.00)	57.62(15.55)	0.01(0.09)
SEQ1	**99.86(1.18)**	0.01(0.10)	7.00(9.80)	0.02(0.16)	0.04(0.63)	0.00(0.00)	79.36(12.13)	0.01(0.11)
SEQ2	87.98(10.79)	1.05(1.05)	28.34(17.62)	1.14(1.10)	3.24(5.51)	0.06(0.26)	82.66(11.76)	1.07(1.10)
SEQ3	**99.94(0.77)**	0.01(0.09)	0.20(1.40)	0.02(0.16)	0.04(0.63)	0.00(0.00)	74.98(13.09)	0.01(0.11)

Table A.5: Mean and Standard Deviation (in parentheses) of the TPR and FPR over three models with 500 repetitions for each possible case with $n = 300$, $d = 50$, $\alpha = 0.1$. Comparatively high TPRs are marked in bold.

Method	Model 3		Model 4		Model 6	
	TPR	FPR	TPR	FPR	TPR	FPR
FSTSH_PEARSON	**99.03(2.17)**	4.36(1.16)	**99.99(0.15)**	2.12(0.89)	**89.80(6.86)**	4.38(1.12)
FSTSH_KENDALL	8.73(5.36)	4.48(1.26)	**99.89(0.59)**	0.64(0.52)	76.59(11.80)	2.66(0.97)
FSTSH_SPEARMAN	29.67(9.97)	6.67(1.41)	**99.97(0.33)**	1.79(0.82)	**87.25(8.73)**	4.87(1.23)
FSTSH_COSINE	71.09(11.17)	6.58(1.17)	**99.95(0.42)**	2.30(0.97)	54.07(11.80)	6.78(1.27)

change the covariance matrix in the base and contamination models for Models 2, 3, 4, and 6 to $\gamma(s,t) = \nu \cdot \exp\{-|t-s|\}$, where $s,\ t \in [0,1]$ and $\nu \in \{0.25, 0.5, 1.5, 5\}$. These results can be seen in Tables A.6 and A.7.

Table A.6: Mean and Standard Deviation (in parentheses) of the TPR and FPR over four models with 500 repetitions for each possible case with $n = 300$, $d = 50$, $\alpha = 0.1$ and $\nu \in \{0.25, 0.5\}$. Comparatively high TPRs are marked in bold.

Method	Model 2		Model 3		Model 4		Model 6	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
				$\nu = 0.25$				
FST	**100.00(0.00)**	6.92(1.59)	**100.00(0.00)**	4.29(1.21)	**100.00(0.00)**	3.48(1.16)	**100.00(0.00)**	4.01(1.21)
FSTMG	**100.00(0.00)**	0.39(0.41)	12.06(6.40)	1.22(0.75)	38.07(10.76)	0.53(0.46)	11.65(6.27)	1.09(0.73)
FSTSH	5.88(4.44)	5.49(1.45)	**100.00(0.00)**	2.63(0.97)	**100.00(0.00)**	2.68(1.01)	**100.00(0.00)**	2.58(0.96)
FSTAM	1.70(2.44)	1.71(0.97)	21.58(7.67)	0.99(0.70)	**95.71(3.99)**	0.38(0.39)	30.75(8.73)	0.73(0.58)
SF	**100.00(0.00)**	6.92(1.58)	**100.00(0.00)**	4.08(1.17)	**100.00(0.00)**	2.73(1.09)	**100.00(0.00)**	3.85(1.18)
SF25	**100.00(0.00)**	6.93(1.65)	**100.00(0.00)**	4.10(1.21)	**100.00(0.00)**	2.65(1.07)	**100.00(0.00)**	3.81(1.17)
MUOD	**100.00(0.00)**	10.11(4.09)	**99.11(6.77)**	6.68(3.35)	**99.98(0.45)**	5.52(2.93)	94.50(14.33)	6.76(3.55)
OGMBD	**100.00(0.00)**	4.76(1.42)	38.89(11.78)	3.54(1.23)	100.00(0.00)	2.08(0.87)	**100.00(0.00)**	1.71(0.86)
MSPLT	**100.00(0.00)**	2.95(1.32)	**100.00(0.00)**	2.99(1.31)	100.00(0.00)	2.91(1.38)	**100.00(0.00)**	2.81(1.33)
TVD	**100.00(0.00)**	0.00(0.02)	**100.00(0.00)**	0.00(0.00)	**99.26(1.89)**	0.00(0.00)	**100.00(0.00)**	0.00(0.03)
FOM	**100.00(0.00)**	0.06(0.16)	**99.97(0.33)**	0.11(0.22)	7.19(10.03)	0.05(0.15)	44.57(31.21)	0.08(0.20)
FAO	**100.00(0.00)**	0.02(0.09)	**99.55(1.45)**	0.03(0.10)	0.07(0.51)	0.00(0.04)	19.87(19.94)	0.03(0.12)
FOM2	**100.00(0.00)**	1.26(0.71)	**100.00(0.00)**	2.06(0.99)	**99.49(1.32)**	1.75(0.86)	**100.00(0.00)**	1.87(0.86)
FAO2	**100.00(0.00)**	1.41(0.79)	**100.00(0.00)**	1.61(0.91)	39.20(23.27)	1.26(0.81)	**100.00(0.00)**	1.53(0.81)
ED	**100.00(0.00)**	0.00(0.00)	**100.00(0.00)**	0.00(0.00)	11.15(9.54)	0.00(0.00)	26.19(11.04)	0.00(0.02)
SEQ1	**100.00(0.00)**	0.00(0.00)	**100.00(0.00)**	0.00(0.00)	**98.80(2.17)**	0.00(0.00)	**99.99(0.15)**	0.00(0.02)
SEQ2	**100.00(0.00)**	0.00(0.02)	**100.00(0.00)**	0.01(0.04)	**99.95(0.39)**	0.00(0.02)	**100.00(0.00)**	0.01(0.04)
SEQ3	**100.00(0.00)**	0.00(0.00)	**100.00(0.00)**	0.00(0.00)	31.67(13.54)	0.00(0.00)	26.29(11.12)	0.00(0.02)
				$\nu = 0.5$				
FST	**100.00(0.00)**	7.62(1.54)	**100.00(0.00)**	5.03(1.27)	**99.99(0.21)**	3.64(1.17)	**100.00(0.00)**	4.82(1.23)
FSTMG	**100.00(0.00)**	0.38(0.38)	6.62(4.57)	1.40(0.82)	17.77(8.22)	0.82(0.58)	6.08(4.48)	1.31(0.81)
FSTSH	6.54(4.52)	6.44(1.41)	**100.00(0.00)**	3.26(1.01)	**99.99(0.21)**	2.58(1.01)	**100.00(0.00)**	3.23(0.97)
FSTAM	1.63(2.38)	1.73(0.97)	11.67(6.13)	1.20(0.82)	60.70(11.91)	0.36(0.38)	15.65(6.69)	1.01(0.70)
SF	**100.00(0.00)**	7.58(1.60)	**100.00(0.00)**	4.73(1.24)	**99.96(0.36)**	3.11(1.11)	**100.00(0.00)**	4.58(1.19)
SF25	**100.00(0.00)**	7.55(1.54)	**100.00(0.00)**	4.71(1.19)	**99.94(0.44)**	3.15(1.14)	**100.00(0.00)**	4.59(1.21)
MUOD	**100.00(0.00)**	9.50(3.90)	87.49(20.59)	9.51(4.31)	**96.41(8.80)**	6.15(3.63)	69.81(22.80)	8.45(4.16)
OGMBD	**100.00(0.00)**	4.84(1.45)	38.75(11.62)	3.48(1.21)	**98.57(2.28)**	2.02(0.88)	**100.00(0.00)**	1.78(0.82)
MSPLT	**100.00(0.00)**	2.97(1.39)	**100.00(0.00)**	2.87(1.35)	**99.97(0.30)**	2.83(1.32)	**100.00(0.00)**	2.94(1.33)
TVD	**100.00(0.00)**	0.00(0.02)	**100.00(0.00)**	0.00(0.00)	38.87(18.45)	0.00(0.00)	**100.00(0.00)**	0.00(0.02)
FOM	**100.00(0.00)**	0.07(0.17)	**93.29(6.72)**	0.09(0.19)	0.65(1.69)	0.06(0.15)	2.85(4.28)	0.06(0.17)
FAO	**100.00(0.00)**	0.02(0.11)	82.73(12.58)	0.03(0.10)	0.11(0.64)	0.00(0.04)	2.01(3.48)	0.01(0.09)
FOM2	**100.00(0.00)**	1.30(0.72)	**100.00(0.00)**	1.95(0.88)	66.22(13.00)	1.80(0.87)	**100.00(0.00)**	1.89(0.87)
FAO2	**100.00(0.00)**	1.42(0.82)	**100.00(0.00)**	1.49(0.74)	17.43(13.20)	1.50(0.85)	**100.00(0.00)**	1.53(0.84)
ED	**100.00(0.00)**	0.00(0.00)	**100.00(0.00)**	0.00(0.00)	0.79(1.77)	0.00(0.00)	2.22(2.82)	0.00(0.00)
SEQ1	**100.00(0.00)**	0.00(0.00)	**100.00(0.00)**	0.00(0.00)	41.70(14.60)	0.00(0.00)	49.93(13.31)	0.00(0.00)
SEQ2	**100.00(0.00)**	0.09(0.18)	**100.00(0.00)**	0.07(0.17)	80.37(10.31)	0.00(0.03)	72.23(10.45)	0.07(0.16)
SEQ3	**100.00(0.00)**	0.00(0.00)	**100.00(0.00)**	0.00(0.00)	2.81(3.44)	0.00(0.00)	2.39(2.93)	0.00(0.00)

Table A.7: Mean and Standard Deviation (in parentheses) of the TPR and FPR over four models with 500 repetitions for each possible case with $n = 300$, $d = 50$, $\alpha = 0.1$ and $\nu \in \{1.5, 5\}$. Comparatively high TPRs are marked in bold.

Method	Model 2		Model 3		Model 4		Model 6	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
				$\nu = 1.5$				
FST	**99.93(0.49)**	10.06(1.55)	80.99(9.75)	6.86(1.35)	**90.73(6.01)**	4.43(1.39)	65.28(11.38)	7.10(1.37)
FSTMG	**99.89(0.60)**	0.35(0.38)	3.23(3.33)	1.53(0.86)	5.97(4.60)	1.38(0.87)	3.09(3.16)	1.59(0.85)
FSTSH	8.93(5.00)	9.03(1.45)	78.13(9.77)	5.13(1.21)	**90.65(6.08)**	2.52(0.97)	61.44(10.96)	5.32(1.18)
FSTAM	1.71(2.38)	1.69(0.91)	4.77(3.94)	1.43(0.84)	18.06(8.08)	0.81(0.61)	5.66(4.45)	1.37(0.79)
SF	**99.94(0.44)**	9.46(1.51)	79.93(9.68)	6.30(1.31)	**87.32(6.69)**	3.64(1.19)	63.53(11.06)	6.47(1.28)
SF25	**99.91(0.57)**	9.49(1.49)	79.45(9.82)	6.28(1.38)	**87.04(8.02)**	3.72(1.24)	63.49(10.80)	6.48(1.26)
MUOD	**97.78(8.96)**	8.65(4.38)	34.67(17.21)	11.26(4.68)	68.05(15.45)	8.30(4.04)	31.61(14.09)	11.05(4.57)
OGMBD	**98.25(2.51)**	4.72(1.39)	32.25(10.39)	3.42(1.13)	67.01(9.08)	2.24(0.95)	**88.95(7.15)**	1.93(0.85)
MSPLT	**94.21(5.34)**	2.92(1.34)	**100.00(0.00)**	2.87(1.34)	**80.71(8.70)**	2.92(1.34)	**99.12(2.00)**	2.79(1.27)
TVD	**98.24(2.56)**	0.00(0.02)	**100.00(0.00)**	0.00(0.00)	0.44(1.26)	0.00(0.02)	54.42(15.48)	0.00(0.00)
FOM	**99.99(0.15)**	0.06(0.16)	13.05(12.26)	0.11(0.23)	0.58(1.52)	0.17(0.26)	0.43(1.22)	0.12(0.22)
FAO	**99.93(0.49)**	0.02(0.09)	5.25(7.10)	0.03(0.10)	0.19(0.80)	0.06(0.16)	0.19(0.81)	0.03(0.11)
FOM2	**100.00(0.00)**	1.23(0.74)	**100.00(0.00)**	2.02(0.94)	19.60(8.27)	2.32(0.92)	59.50(14.36)	2.02(0.92)
FAO2	**100.00(0.00)**	1.45(0.86)	**100.00(0.00)**	1.56(0.87)	12.49(7.43)	2.05(0.97)	48.37(15.46)	1.58(0.86)
ED	**98.43(2.45)**	0.00(0.00)	83.27(7.70)	0.00(0.00)	0.03(0.33)	0.00(0.00)	0.07(0.47)	0.00(0.00)
SEQ1	**98.29(2.45)**	0.00(0.00)	**100.00(0.00)**	0.00(0.00)	0.55(1.47)	0.00(0.00)	0.38(1.08)	0.00(0.00)
SEQ2	**98.29(2.45)**	1.40(0.70)	**100.00(0.00)**	1.13(0.67)	6.78(6.46)	0.01(0.05)	5.39(4.65)	1.10(0.68)
SEQ3	**98.29(2.45)**	0.00(0.00)	**100.00(0.00)**	0.00(0.00)	0.06(0.44)	0.00(0.00)	0.07(0.47)	0.00(0.00)
				$\nu = 5$				
FST	**70.97(10.80)**	11.17(1.68)	11.85(6.49)	9.80(1.62)	42.21(9.51)	5.77(1.48)	14.59(6.90)	10.49(1.69)
FSTMG	66.31(11.31)	0.41(0.42)	1.88(2.54)	1.69(0.89)	2.86(3.14)	1.59(0.92)	2.13(2.79)	1.76(0.89)
FSTSHA	10.29(5.33)	10.15(1.68)	9.61(6.09)	7.97(1.51)	41.25(9.54)	3.55(1.10)	12.10(6.53)	8.64(1.61)
FSTAM	1.72(2.31)	1.66(0.88)	2.47(2.77)	1.64(0.91)	5.55(4.64)	1.31(0.78)	2.91(3.18)	1.65(0.92)
SF	**76.50(9.30)**	10.47(1.67)	10.51(6.24)	9.05(1.67)	38.99(9.89)	4.62(1.30)	13.43(6.69)	9.74(1.70)
SF25	**75.65(9.56)**	10.41(1.64)	10.39(6.13)	8.99(1.70)	38.69(10.57)	4.65(1.34)	13.51(6.89)	9.76(1.72)
MUOD	74.78(18.66)	14.67(7.06)	26.25(14.44)	18.46(10.50)	44.07(12.90)	12.76(4.74)	24.81(12.34)	17.97(7.90)
OGMBD	18.68(9.64)	4.72(1.40)	11.17(6.43)	3.50(1.15)	26.32(8.14)	3.35(1.17)	19.69(8.67)	2.83(1.12)
MSPLT	9.15(6.47)	2.77(1.27)	48.01(14.93)	2.89(1.27)	27.09(9.30)	2.80(1.31)	21.75(10.97)	2.65(1.25)
TVD	18.51(9.74)	0.00(0.02)	**100.00(0.00)**	0.00(0.02)	0.03(0.30)	0.00(0.02)	0.33(1.09)	0.00(0.02)
FOM	**71.67(12.59)**	0.07(0.17)	0.30(1.00)	0.15(0.26)	0.67(1.59)	0.41(0.43)	0.56(1.46)	0.43(0.48)
FAO	49.99(16.10)	0.03(0.10)	0.15(0.76)	0.07(0.16)	0.28(1.02)	0.17(0.26)	0.28(0.97)	0.17(0.29)
FOM2	**93.87(4.73)**	1.25(0.70)	**100.00(0.00)**	2.14(0.93)	8.59(5.64)	3.14(1.18)	10.33(6.27)	2.56(1.02)
FAO2	**93.01(5.61)**	1.51(0.83)	**100.00(0.00)**	1.70(0.91)	7.07(5.10)	2.74(1.14)	8.21(5.57)	2.09(0.98)
ED	16.40(8.54)	0.00(0.02)	3.55(3.47)	0.00(0.02)	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.00)
SEQ1	15.25(8.06)	0.00(0.02)	**100.00(0.00)**	0.00(0.02)	0.01(0.15)	0.00(0.00)	0.00(0.00)	0.00(0.02)
SEQ2	15.68(8.17)	0.47(0.52)	60.53(10.52)	0.22(0.36)	0.14(0.67)	0.04(0.12)	0.13(0.76)	0.28(0.37)
SEQ3	15.25(8.06)	0.00(0.02)	**100.00(0.00)**	0.00(0.02)	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.02)

Appendix B

Supplementary Material: Multivariate Functional Outlier Detection with the Fast-MOUD Indices

B.1 Proof of Proposition 4.3

(i) (a) By definition,

$$I_M(y', F_X) = \int y'(t)dt - \beta(y') \int \mu(r)dr \tag{B.1}$$
$$= \int (ay(t) + b)dt - \beta(ay(t) + b) \int \mu(r)dr.$$

But,

$$\beta(ay(t) + b) = \frac{\int \left[ay(t) + b - \int (ay(r) + b)dr\right] \tilde{\mu}(t)dt}{\int \tilde{\mu}(t)^2 dt}$$
$$= \frac{\int \left[ay(t) - \int ay(r)dr\right] \tilde{\mu}(t)dt}{\int \tilde{\mu}(t)^2 dt}$$
$$= \frac{a \int \tilde{y}(t)\tilde{\mu}(t)dt}{\int \tilde{\mu}(t)^2 dt} = a\beta(y).$$

So Equation (B.1) becomes

$$\begin{aligned} I_M(y', F_X) &= b + a\int y(t)dt - a\beta(y)\int \mu(r)dr \\ &= aI_M(y, F_X) + b. \end{aligned}$$

(b) For the amplitude index, by definition, we have that

$$\begin{aligned} I_A(y', F_X) &= \frac{\int \left[ay(t) + b - \int (ay(r) + b)dr\right]\left[\mu(t) - \int \mu(r)dr\right]dt}{\int \left[\mu(t) - \int \mu(r)dr\right]^2 dt} - 1 \\ &= \frac{a\int \left[y(t) - \int y(r)dr\right]\left[\mu(t) - \int \mu(r)dr\right]dt}{\int \left[\mu(t) - \int \mu(r)dr\right]^2 dt} - 1 \\ &= \frac{a\int \tilde{y}(t)\tilde{\mu}(t)dt}{\int \tilde{\mu}(t)^2 dt} - 1 \\ &= a(I_A(y, F_X) + 1) - 1 \\ &= aI_A(y, F_X) + a - 1. \end{aligned}$$

(c) For the shape index, since $a \neq 0$, we have by definition:

$$\begin{aligned} I_S(y', F_X) &= 1 - \frac{\int \left[ay(t) + b - \int (ay(r) + b)dr\right]\left[\mu(t) - \int \mu(r)dr\right]dt}{\left(\int \left[ay(t) + b - \int (ay(r) + b)dr\right]^2 dt\right)^{1/2}\left(\int \left[\mu(t) - \int \mu(r)dr\right]^2 dt\right)^{1/2}} \\ &= 1 - \frac{a\int \left[y(t) - \int y(r)dr\right]\left[\mu(t) - \int \mu(r)dr\right]dt}{a\left(\int \left[y(t) - \int y(r)dr\right]^2 dt\right)^{1/2}\left(\int \left[\mu(t) - \int \mu(r)dr\right]^2 dt\right)^{1/2}} \\ &= 1 - \frac{\int \tilde{y}(t)\tilde{\mu}(t)dt}{\left(\int \tilde{y}(t)^2 dt\right)^{1/2}\left(\int \tilde{\mu}(t)^2 dt\right)^{1/2}} = I_S(y, F_X). \end{aligned}$$

Thus, for any $a, b \in \mathbb{R}$, $a \neq 0$, we have that $I_S(y, F_X) = I_S(y', F_X)$.

(ii) (a) Since

$$\begin{aligned} \beta(y') &= \beta(y(t) + z(t)) \\ &= \frac{\int \left[y(t) + z(t) - \int (y(r) + z(r))dr\right]\tilde{\mu}(t)dt}{\int \tilde{\mu}(t)^2 dt} \\ &= \frac{\int \left[y(t) - \int y(r)dr + z(t) - \int z(r)dr\right]\tilde{\mu}(t)dt}{\int \tilde{\mu}(t)^2 dt} \\ &= \frac{\int \left[\tilde{y}(t) + \tilde{z}(t)\right]\tilde{\mu}(t)dt}{\int \tilde{\mu}(t)^2 dt} \\ &= \beta(y) + \beta(z), \end{aligned}$$

we have (by definition) that

$$\begin{aligned} I_M(y', F_X) &= \int y'(t)dt - \beta(y') \int \mu(r)dr \\ &= \int (y(t) + z(t))dt - [\beta(y) + \beta(z)] \int \mu(r)dr \\ &= \int (y(t) + z(t))dt - \beta(y) \int \mu(r)dr - \beta(z) \int \mu(r)dr \\ &= I_M(y, F_X) + I_M(z, F_X). \end{aligned}$$

(b) For the amplitude index, assume that for some $z \in L^2([0,1])$, $\langle \tilde{z}, \tilde{\mu} \rangle = 0$, we have that

$$\begin{aligned} I_A(y', F_X) &= \frac{\int \tilde{y}'(t)\tilde{\mu}(t)}{\int \tilde{\mu}(t)^2 dt} - 1 \\ &= \frac{\int \left[y(t) + z(t) - \int (y(r) + z(r))dr \right] \tilde{\mu}(t)dt}{\int \tilde{\mu}(t)^2 dt} - 1 \\ &= \frac{\int \left[y(t) - \int y(r)dr + z(t) - \int z(r)dr \right] \tilde{\mu}(t)dt}{\int \tilde{\mu}(t)^2 dt} - 1 \\ &= \frac{\int [\tilde{y}(t) + \tilde{z}(t)] \tilde{\mu}(t)dt}{\int \tilde{\mu}(t)^2 dt} - 1 \\ &= \frac{\int \tilde{y}(t)\tilde{\mu}(t)dt}{\int \tilde{\mu}(t)^2 dt} + \frac{\int \tilde{z}(t)\tilde{\mu}(t)dt}{\int \tilde{\mu}(t)^2 dt} - 1 \\ &= \frac{\langle \tilde{y}, \tilde{\mu} \rangle}{\|\tilde{\mu}\|^2} + \frac{\langle \tilde{z}, \tilde{\mu} \rangle}{\|\tilde{\mu}\|^2} - 1 \\ &= \frac{\langle \tilde{y}, \tilde{\mu} \rangle}{\|\tilde{\mu}\|^2} - 1 \\ &= I_A(y, F_X). \end{aligned}$$

Now, assume that $I_A(y, F_X) = I_A(y', F_X)$, we have (by definition of I_A) that

$$\begin{aligned} \int \tilde{y}(t)\tilde{\mu}(t)dt &= \int \tilde{y}'(t)\tilde{\mu}(t)dt \\ &= \int \left[y(t) + z(t) - \int (y(r) + z(r))dr \right] \tilde{\mu}(t)dt \\ &= \int \left[y(t) - \int y(r)dr + z(t) - \int z(r)dr \right] \tilde{\mu}(t)dt \\ &= \int [\tilde{y}(t) + \tilde{z}(t)] \tilde{\mu}(t)dt \\ &= \int \tilde{y}(t)\tilde{\mu}(t)dt + \int \tilde{z}(t)\tilde{\mu}(t)dt, \end{aligned}$$

which implies that:

$$\int \tilde{z}(t)\tilde{\mu}(t)dt = \langle \tilde{z}, \tilde{\mu} \rangle = 0.$$

(c) For the shape index, assume that $I_S(y, F_X) = I_S(y', F_X)$. By definition, $I_S(y, F_X) = I_S(y', F_X)$ implies that

$$\begin{aligned}
\frac{\int \tilde{y}(t)\tilde{\mu}(t)dt}{[\int \tilde{y}(t)^2 dt]^{1/2}} &= \frac{\int \tilde{y}'(t)\tilde{\mu}(t)dt}{[\int \tilde{y}'(t)^2 dt]^{1/2}} \\
&= \frac{\int \tilde{y}(t)\tilde{\mu}(t)dt + \int \tilde{z}(t)\tilde{\mu}(t)dt}{[\int (\tilde{y}(t) + \tilde{z}(t))^2 dt]^{1/2}} \\
&= \frac{\langle \tilde{y}, \tilde{\mu} \rangle + \langle \tilde{b}, \tilde{\mu} \rangle}{\|\tilde{y} + \tilde{z}\|}.
\end{aligned}$$

Now suppose that

$$\frac{\langle \tilde{y}, \tilde{\mu} \rangle}{\|\tilde{y}\|} = \frac{\langle \tilde{y}, \tilde{\mu} \rangle + \langle \tilde{z}, \tilde{\mu} \rangle}{\|\tilde{y} + \tilde{z}\|},$$

then

$$\begin{aligned}
I_S(y', F_X) &= 1 - \frac{\langle \tilde{y}', \tilde{\mu} \rangle}{\|\tilde{y}'\| \cdot \|\tilde{\mu}\|} \\
&= 1 - \frac{\langle \tilde{y}, \tilde{\mu} \rangle + \langle \tilde{z}, \tilde{\mu} \rangle}{\|\tilde{y} + \tilde{z}\| \cdot \|\tilde{\mu}\|} \\
&= 1 - \frac{\langle \tilde{y}, \tilde{\mu} \rangle}{\|\tilde{y}\| \cdot \|\tilde{\mu}\|} \\
&= I_S(y, F_X).
\end{aligned}$$

(iii) The proofs follow from the definitions of $I_A(y, F_X)$, $I_A(y', F_X)$, $I_S(y, F_X)$ and $I_S(y', F_X)$.

□

B.2 Proof of Corollary 4.1

Proof. The proofs of the statements follow from the definition.

(i) Suppose that $I_{M_v}(y', F_X) = I_{M_v}(y, F_X)$, then by definition (of I_{M_v}) and Proposition 4.3 we have that:

$$|aI_M(y, F_X) + b| = |I_M(y, F_X)|.$$

If both $aI_M(y, F_X) + b$ and $I_M(y, F_X)$ have the same sign, then

$$aI_M(y, F_X) + b = I_M(y, F_X),$$

which implies that $b = (-a + 1)I_M(y, F_X)$. However, if $aI_M(y, F_X) + b$ and $I_M(y, F_X)$ have different signs,

$$aI_M(y, F_X) + b = -I_M(y, F_X),$$

which implies that $b = (-a - 1)I_M(y, F_X)$.

To prove the reverse direction, we have to show that whenever $b = (-a \pm 1)I_M(y, F_X)$, $I_{M_v}(y', F_X) = I_{M_v}(y, F_X)$. For the first case, we assume that $b = (-a+1)I_M(y, F_X)$, then

$$\begin{aligned} I_{M_v}(y', F_X) &= |I_M(y', F_X)| \\ &= |aI_M(y, F_X) + b| \\ &= |aI_M(y, F_X) + (1 - a)I_M(y, F_X)| \\ &= |I_M(y, F_X)| \\ &= I_{M_v}(y, F_X). \end{aligned}$$

For the second case, suppose that $b = (-a - 1)I_M(y, F_X)$, then

$$\begin{aligned} I_{M_v}(y', F_X) &= |I_M(y', F_X)| \\ &= |aI_M(y, F_X) + b| \\ &= |aI_M(y, F_X) + (-a - 1)I_M(y, F_X)| \\ &= | - I_M(y, F_X)| \\ &= |I_M(y, F_X)| \\ &= I_{M_v}(y, F_X). \end{aligned}$$

So in both cases, we have that $I_{M_v}(y', F_X) = I_{M_v}(y, F_X)$, which completes the proof.

(ii) Suppose that $I_{A_v}(y', F_X) = I_{A_v}(y, F_X)$, by definition (of I_{A_v}) and Proposition 4.3 we have

$$|aI_A(y, F_X) + a - 1| = |I_A(y, F_X)|.$$

If both $aI_A(y, F_X) + a - 1$ and $I_A(y, F_X)$ have the same sign,

$$aI_A(y, F_X) + a - 1 = I_A(y, F_X),$$

which indicates that

$$a = \frac{I_A(y, F_X) + 1}{I_A(y, F_X) + 1} = 1.$$

Nevertheless, if $aI_A(y, F_X) + a - 1$ and $I_A(y, F_X)$ have different signs,

$$aI_A(y, F_X) + a - 1 = -I_A(y, F_X),$$

which indicates that

$$a = \frac{1 - I_A(y, F_X)}{1 + I_A(y, F_X)}.$$

To prove the reverse case, we have to show that whenever $a = 1$ or $a = \frac{1-I_A(y,F_X)}{1+I_A(y,F_X)}$, $I_{A_v}(y', F_X) = I_{A_v}(y, F_X)$. For the first case, assume that $a = 1$, then:

$$\begin{aligned} I_{A_v}(y', F_X) &= |I_A(y', F_X)| \\ &= |aI_A(y, F_X) + a - 1| \\ &= |I_A(y, F_X)| \\ &= I_{A_v}(y, F_X). \end{aligned}$$

For the second case, assume that $a = \frac{1-I_A(y,F_X)}{1+I_A(y,F_X)}$, then:

$$\begin{aligned} I_{A_v}(y', F_X) &= |I_A(y', F_X)| \\ &= |aI_A(y, F_X) + a - 1| \\ &= \left| \frac{(1 + I_A(y, F_X)(1 - I_A(y, F_X))}{1 + I_A(y, F_X)} - 1 \right| \\ &= | - I_A(y, F_X)| \\ &= |I_A(y, F_X)| \\ &= I_{A_v}(y, F_X). \end{aligned}$$

Thus, in both cases, $I_{A_v}(y', F_X) = I_{A_v}(y, F_X)$, which completes the proof.

□

B.3 Additional Simulation Results on Multivariate Functional Data

In this section we show the results of the methods outlined in Subsection 5.2.2 (of the thesis) on more simulation models. The models considered are variants of the simulation models in Subsection 5.2.1. The outliers are outlying only in one or two dimensions of the trivariate functional dataset. Figures B.1 and B.2 show the simulation models, and the results are shown in Table B.1 - B.4 with different contamination rates.

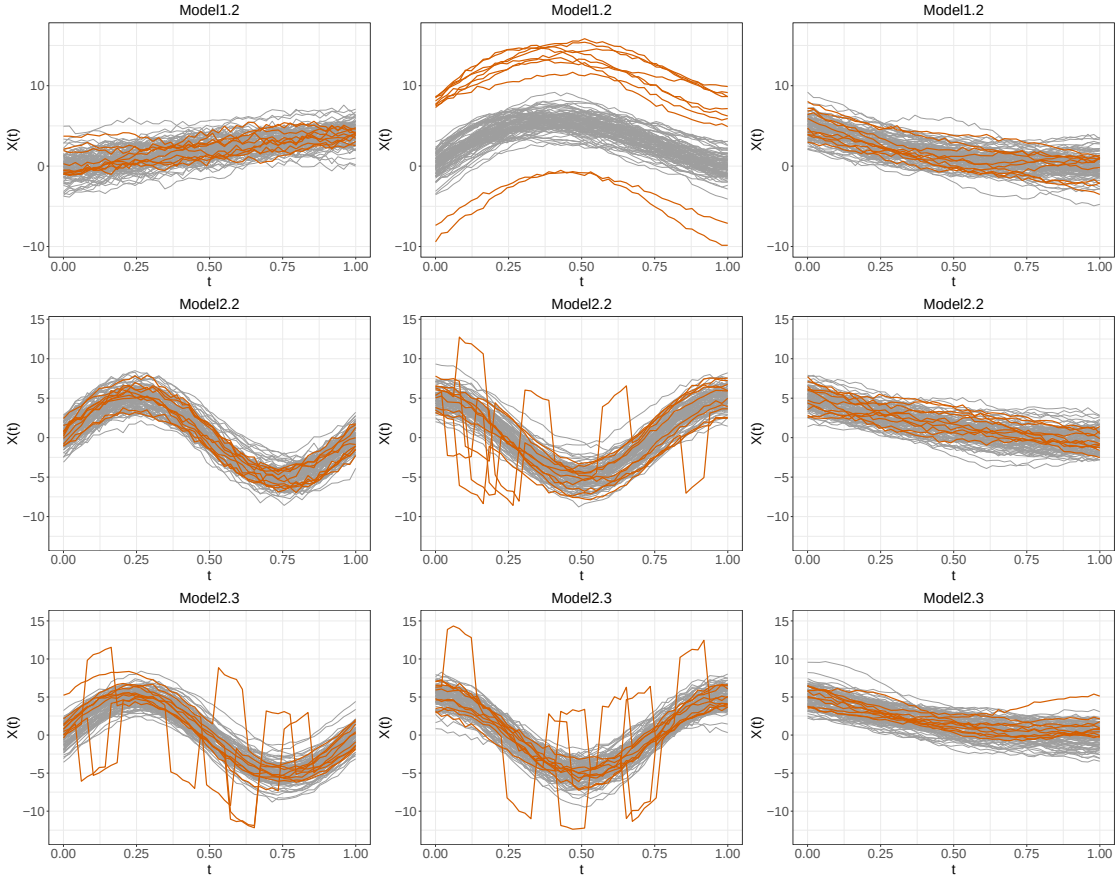

Figure B.1: Sample data generated by variants of Models 1 and 2 with contamination rate $\alpha = 0.10$, sample size $n = 100$, and evaluation point $d = 50$. Each row corresponds to a simulation model and each column corresponds to the margins of the multivariate functional data. Outliers are shown in colour.

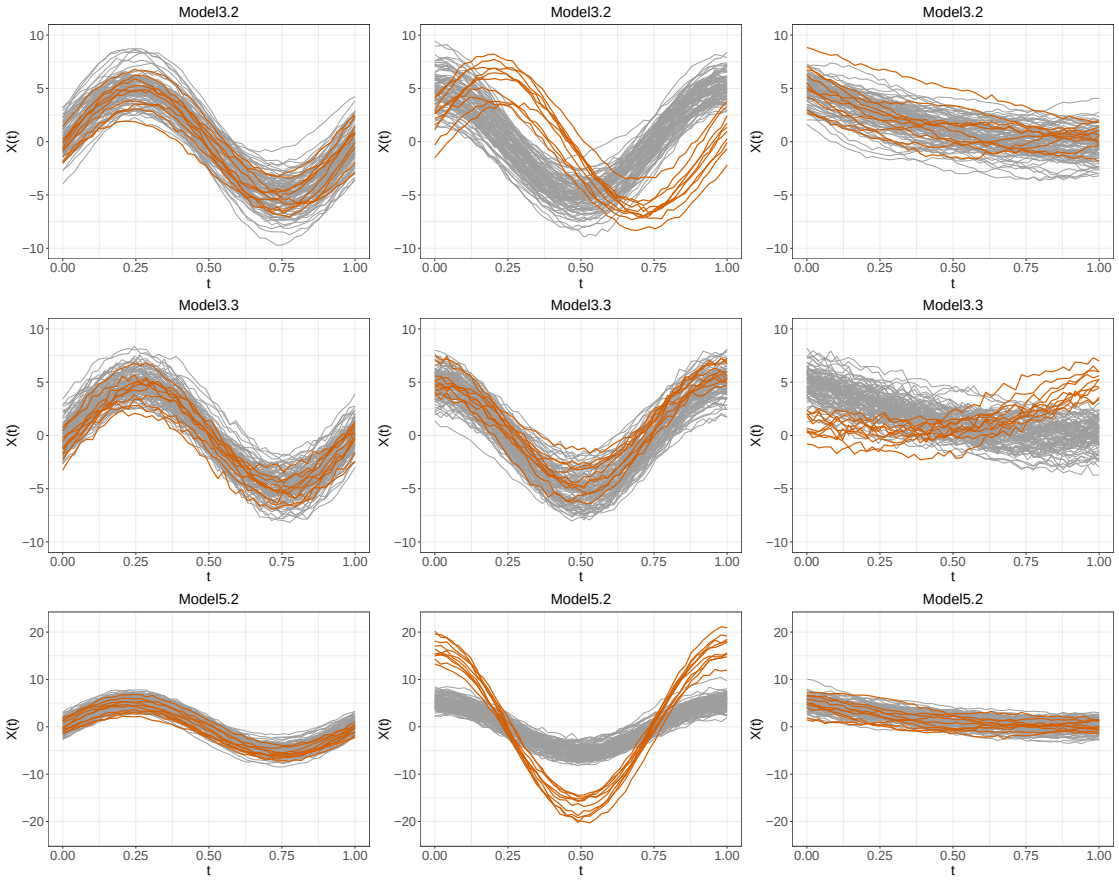

Figure B.2: Sample data generated by variants of Models 3 and 5 with contamination rate $\alpha = 0.10$, sample size $n = 100$, and evaluation point $d = 50$. Each row corresponds to a simulation model and each column corresponds to the margins of the multivariate functional data. Outliers are shown in colour.

Table B.1: Mean and Standard Deviation (in parentheses) of the TPR and FPR (in percentage) over 200 repetitions for each model. Sample size $n = 100$, evaluation points $t_j = 50$, and contamination rate is 5%. The proposed methods are in italics.

Method	Model 1.2		Model 2.2		Model 2.3	
	TPR	FPR	TPR	FPR	TPR	FPR
FST-MAR	100.0(0.0)	26.5(4.1)	99.0(4.8)	22.2(3.7)	100.0(0.0)	21.1(3.9)
FST-STR	92.9(12.2)	3.2(2.0)	34.2(22.8)	3.6(2.1)	76.3(21.1)	3.6(2.2)
FST-PRJ	99.2(3.9)	0.5(0.8)	36.6(36.9)	1.2(1.5)	95.1(14.1)	1.7(1.3)
FST-PRJ-SH	0.5(3.1)	0.4(0.8)	36.3(36.9)	1.1(1.6)	94.4(15.4)	1.6(1.4)
FST-PRJ-AM	0.0(0.0)	0.0(0.0)	0.2(2.0)	0.0(0.1)	0.8(3.9)	0.0(0.2)
FST-PRJ-MG	99.2(3.9)	0.1(0.3)	0.2(2.0)	0.1(0.3)	0.0(0.0)	0.0(0.2)
FST-PRJ1	99.8(2.0)	4.1(1.9)	63.5(25.3)	2.5(1.4)	98.9(4.6)	2.3(1.6)
FST-PRJ1-SH	4.1(8.8)	3.9(1.9)	63.0(25.2)	2.1(1.3)	98.0(6.0)	2.0(1.5)
FST-PRJ1-AM	0.5(3.1)	0.2(0.5)	0.6(3.4)	0.1(0.4)	2.6(6.7)	0.1(0.3)
FST-PRJ1-MG	99.8(2.0)	0.1(0.4)	1.0(4.4)	0.3(0.6)	2.6(7.3)	0.3(0.6)
FST-PRJ2	100.0(0.0)	51.9(3.7)	100.0(0.0)	47.3(3.7)	100.0(0.0)	45.4(3.7)
FST-PRJ2-SH	43.6(22.2)	47.0(3.8)	99.8(2.0)	40.9(3.6)	100.0(0.0)	38.8(3.3)
FST-PRJ2-AM	12.9(15.3)	13.3(3.6)	33.4(22.7)	12.8(4.1)	49.6(24.0)	12.1(3.6)
FST-PRJ2-MG	100.0(0.0)	11.2(3.4)	18.5(16.5)	10.0(3.4)	27.0(19.8)	10.1(3.3)
MSPLOT	100.0(0.0)	0.9(1.1)	80.8(21.1)	1.5(1.5)	99.8(2.0)	1.4(1.5)
FOM	99.2(3.9)	0.1(0.4)	66.9(26.0)	0.2(0.5)	93.1(13.7)	0.2(0.4)
FAO	99.2(4.4)	0.1(0.4)	41.8(27.2)	0.1(0.3)	78.9(24.2)	0.1(0.4)

Method	Model 3.2		Model 3.3		Model 5.2	
	TPR	FPR	TPR	FPR	TPR	FPR
FST-MAR	100.0(0.0)	22.6(4.2)	99.3(4.2)	22.5(4.1)	100.0(0.0)	24.5(4.4)
FST-STR	99.6(2.8)	3.2(1.9)	61.0(22.3)	3.8(2.2)	100.0(0.0)	3.5(1.9)
FST-PRJ	90.4(24.6)	1.8(1.4)	35.1(34.9)	1.5(1.6)	99.1(8.3)	0.3(0.7)
FST-PRJ-SH	88.8(26.9)	1.7(1.4)	34.4(35.3)	1.3(1.7)	4.1(12.9)	0.2(0.6)
FST-PRJ-AM	43.5(36.3)	0.1(0.4)	1.9(7.4)	0.1(0.3)	99.1(8.3)	0.1(0.3)
FST-PRJ-MG	0.0(0.0)	0.0(0.2)	0.1(1.4)	0.1(0.4)	0.9(4.6)	0.0(0.2)
FST-PRJ1	99.5(3.1)	2.2(1.6)	60.5(25.7)	2.6(1.5)	100.0(0.0)	2.6(1.5)
FST-PRJ1-SH	99.4(3.4)	1.9(1.4)	60.4(25.9)	2.3(1.5)	45.8(26.6)	2.3(1.4)
FST-PRJ1-AM	61.7(30.4)	0.1(0.3)	5.8(11.6)	0.1(0.3)	100.0(0.0)	0.1(0.3)
FST-PRJ1-MG	0.2(2.0)	0.3(0.6)	0.4(2.8)	0.3(0.5)	4.9(9.3)	0.3(0.5)
FST-PRJ2	100.0(0.0)	46.3(4.1)	100.0(0.0)	47.9(3.8)	100.0(0.0)	47.9(3.5)
FST-PRJ2-SH	100.0(0.0)	39.9(3.6)	99.9(1.4)	41.2(3.4)	99.9(1.4)	42.2(3.3)
FST-PRJ2-AM	99.8(2.0)	11.5(3.6)	89.2(15.5)	12.3(3.6)	100.0(0.0)	10.4(3.5)
FST-PRJ2-MG	23.4(19.3)	10.1(3.5)	36.9(25.0)	10.3(3.6)	75.8(22.0)	10.0(3.5)
MSPLOT	99.0(4.4)	1.3(1.5)	56.4(22.0)	1.2(1.3)	100.0(0.0)	1.0(1.1)
FOM	27.0(28.3)	0.1(0.4)	6.7(12.6)	0.3(0.5)	99.9(1.4)	0.1(0.5)
FAO	19.2(24.6)	0.1(0.4)	4.0(9.2)	0.1(0.5)	97.9(9.9)	0.1(0.4)

Table B.2: Mean and Standard Deviation (in parentheses) of the TPR and FPR (in percentage) over 200 repetitions for each model. Sample size $n = 100$, evaluation points $t_j = 50$, and contamination rate is 10%. The proposed methods are in italics.

Method	Model 1.2		Model 2.2		Model 2.3	
	TPR	FPR	TPR	FPR	TPR	FPR
FST-MAR	100.0(0.0)	26.3(4.0)	95.5(8.1)	21.1(4.0)	99.8(1.4)	17.6(3.6)
FST-STR	87.5(12.2)	2.1(1.6)	26.9(14.7)	2.9(2.0)	66.5(19.4)	2.6(2.0)
FST-PRJ	98.2(6.6)	0.3(0.7)	34.2(27.9)	0.9(1.1)	90.7(14.9)	0.8(1.0)
FST-PRJ-SH	0.3(1.7)	0.3(0.7)	34.1(28.0)	0.8(1.0)	90.6(15.0)	0.8(1.0)
FST-PRJ-AM	0.0(0.7)	0.0(0.0)	0.1(1.0)	0.0(0.1)	0.0(0.7)	0.0(0.0)
FST-PRJ-MG	98.2(6.6)	0.0(0.2)	0.2(1.2)	0.1(0.3)	0.0(0.0)	0.0(0.2)
FST-PRJ1	99.4(2.8)	3.7(2.2)	50.1(19.9)	1.8(1.4)	93.8(8.1)	1.3(1.2)
FST-PRJ1-SH	4.5(6.1)	3.6(2.2)	49.2(20.0)	1.4(1.2)	92.7(8.6)	0.9(1.0)
FST-PRJ1-AM	0.4(1.8)	0.2(0.4)	0.8(3.0)	0.1(0.4)	3.0(5.4)	0.1(0.3)
FST-PRJ1-MG	99.2(3.0)	0.0(0.2)	1.0(2.9)	0.4(0.7)	2.1(4.3)	0.3(0.6)
FST-PRJ2	100.0(0.0)	51.3(4.3)	99.7(2.3)	44.6(4.1)	100.0(0.0)	38.9(4.1)
FST-PRJ2-SH	47.9(14.9)	47.1(4.2)	99.2(3.4)	37.3(3.7)	100.0(0.0)	31.7(3.6)
FST-PRJ2-AM	13.6(10.9)	12.3(3.7)	28.1(14.8)	11.9(4.1)	46.7(17.5)	10.5(3.2)
FST-PRJ2-MG	100.0(0.0)	9.0(3.5)	16.1(11.4)	10.7(3.6)	24.3(13.7)	9.7(3.7)
MSPLOT	99.9(1.0)	0.5(0.8)	75.9(18.8)	1.1(1.3)	99.5(2.5)	1.0(1.2)
FOM	87.5(17.8)	0.1(0.3)	57.6(21.4)	0.0(0.2)	89.3(12.9)	0.1(0.4)
FAO	91.5(16.1)	0.0(0.2)	31.2(20.2)	0.0(0.2)	65.8(23.5)	0.0(0.1)

Method	Model 3.2		Model 3.3		Model 5.2	
	TPR	FPR	TPR	FPR	TPR	FPR
FST-MAR	100.0(0.0)	21.4(3.9)	98.7(3.6)	20.3(3.9)	100.0(0.0)	24.3(4.3)
FST-STR	97.9(4.8)	2.6(1.8)	45.1(19.1)	2.5(1.9)	100.0(0.0)	2.1(1.6)
FST-PRJ	94.1(10.9)	1.1(1.2)	33.8(26.6)	1.2(1.2)	100.0(0.0)	0.1(0.4)
FST-PRJ-SH	94.0(11.1)	1.1(1.2)	33.7(26.7)	1.1(1.2)	0.2(1.4)	0.1(0.3)
FST-PRJ-AM	25.8(27.1)	0.0(0.2)	1.6(4.4)	0.0(0.2)	100.0(0.0)	0.1(0.2)
FST-PRJ-MG	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.1(0.3)	0.0(0.0)	0.0(0.0)
FST-PRJ1	96.4(7.2)	1.5(1.3)	45.4(20.2)	1.8(1.3)	100.0(0.0)	1.8(1.3)
FST-PRJ1-SH	96.2(7.5)	1.2(1.1)	45.2(20.2)	1.5(1.2)	22.9(18.5)	1.6(1.2)
FST-PRJ1-AM	41.3(25.3)	0.0(0.2)	4.0(6.5)	0.1(0.3)	100.0(0.0)	0.0(0.3)
FST-PRJ1-MG	0.4(1.8)	0.3(0.6)	0.3(1.7)	0.3(0.6)	4.2(6.7)	0.2(0.5)
FST-PRJ2	100.0(0.0)	43.4(4.3)	99.8(1.4)	44.5(4.0)	100.0(0.0)	45.7(3.6)
FST-PRJ2-SH	100.0(0.0)	36.2(3.7)	99.8(1.6)	37.4(3.4)	98.8(3.7)	40.8(3.7)
FST-PRJ2-AM	99.4(2.5)	10.4(3.5)	83.3(14.0)	10.4(3.4)	100.0(0.0)	8.3(3.0)
FST-PRJ2-MG	20.4(13.8)	10.5(3.6)	28.6(16.6)	10.0(3.5)	65.7(20.8)	8.9(3.5)
MSPLOT	94.1(8.3)	1.1(1.2)	39.8(17.4)	1.0(1.3)	100.0(0.0)	0.9(1.1)
FOM	2.6(6.8)	0.1(0.4)	1.2(3.8)	0.2(0.5)	95.0(11.6)	0.1(0.3)
FAO	1.9(6.3)	0.1(0.2)	0.6(2.6)	0.1(0.4)	80.3(24.0)	0.0(0.2)

Table B.3: Mean and Standard Deviation (in parentheses) of the TPR and FPR (in percentage) over 200 repetitions for each model. Sample size $n = 100$, evaluation points $t_j = 50$, and contamination rate is 15%. The proposed methods are in italics.

Method	Model 1.2		Model 2.2		Model 2.3	
	TPR	FPR	TPR	FPR	TPR	FPR
FST-MAR	100.0(0.0)	26.7(4.2)	86.2(13.0)	19.0(3.8)	98.1(4.0)	14.4(3.4)
FST-STR	85.1(13.7)	1.5(1.5)	18.9(11.9)	2.3(1.7)	50.9(18.6)	1.8(1.6)
FST-PRJ	97.7(6.1)	0.2(0.5)	19.5(20.6)	0.5(0.9)	75.7(16.9)	0.4(0.7)
FST-PRJ-SH	0.2(1.1)	0.2(0.5)	19.3(20.5)	0.4(0.9)	75.7(16.9)	0.4(0.6)
FST-PRJ-AM	0.0(0.0)	0.0(0.0)	0.2(1.4)	0.0(0.2)	0.1(0.8)	0.0(0.0)
FST-PRJ-MG	97.7(6.1)	0.0(0.2)	0.1(0.9)	0.0(0.3)	0.0(0.0)	0.0(0.2)
FST-PRJ1	98.9(3.8)	3.6(2.1)	33.5(14.7)	1.2(1.2)	78.6(13.5)	0.6(0.9)
FST-PRJ1-SH	3.1(4.7)	3.5(2.1)	32.8(14.9)	0.9(1.0)	77.3(13.7)	0.4(0.7)
FST-PRJ1-AM	0.2(1.1)	0.2(0.5)	0.6(2.1)	0.1(0.3)	2.2(4.1)	0.1(0.3)
FST-PRJ1-MG	98.7(4.0)	0.0(0.1)	0.8(2.6)	0.2(0.5)	1.3(3.3)	0.2(0.5)
FST-PRJ2	100.0(0.0)	51.0(3.9)	97.0(5.8)	40.6(4.0)	99.9(0.8)	32.7(4.1)
FST-PRJ2-SH	46.8(11.5)	47.5(3.8)	95.1(6.9)	33.3(3.7)	99.5(1.9)	24.8(3.2)
FST-PRJ2-AM	12.5(9.0)	12.5(4.1)	27.1(12.4)	10.9(3.3)	43.0(15.0)	9.8(3.5)
FST-PRJ2-MG	100.0(0.0)	7.0(3.3)	15.6(9.9)	10.0(3.3)	24.1(11.6)	8.4(3.3)
MSPLOT	99.2(3.0)	0.4(0.7)	70.6(19.8)	1.1(1.3)	99.0(2.4)	0.8(1.0)
FOM	45.7(28.5)	0.0(0.1)	42.5(22.7)	0.0(0.2)	78.0(17.7)	0.0(0.1)
FAO	70.2(28.7)	0.0(0.0)	20.1(16.7)	0.0(0.1)	45.7(25.9)	0.0(0.1)

Method	Model 3.2		Model 3.3		Model 5.2	
	TPR	FPR	TPR	FPR	TPR	FPR
FST-MAR	100.0(0.0)	19.6(4.1)	97.8(3.8)	18.7(4.2)	100.0(0.0)	24.6(4.5)
FST-STR	93.5(7.9)	1.6(1.5)	37.0(14.5)	2.3(1.7)	100.0(0.0)	1.3(1.3)
FST-PRJ	85.7(15.7)	0.6(0.9)	25.7(19.6)	0.9(1.1)	100.0(0.5)	0.0(0.3)
FST-PRJ-SH	85.6(15.9)	0.6(0.9)	25.6(19.7)	0.8(1.1)	0.0(0.0)	0.0(0.2)
FST-PRJ-AM	11.7(19.0)	0.0(0.0)	0.7(2.5)	0.0(0.2)	100.0(0.5)	0.0(0.2)
FST-PRJ-MG	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.1)	0.0(0.0)	0.0(0.0)
FST-PRJ1	88.9(13.3)	1.0(1.1)	32.2(17.2)	1.3(1.2)	100.0(0.0)	1.3(1.2)
FST-PRJ1-SH	88.8(13.5)	0.6(0.9)	31.7(17.4)	1.1(1.1)	10.6(10.2)	1.2(1.1)
FST-PRJ1-AM	19.5(19.2)	0.0(0.2)	1.8(3.7)	0.1(0.3)	100.0(0.0)	0.0(0.2)
FST-PRJ1-MG	0.3(1.4)	0.3(0.6)	0.5(1.8)	0.2(0.5)	2.3(4.1)	0.1(0.4)
FST-PRJ2	100.0(0.0)	39.8(4.4)	99.0(2.6)	42.5(4.5)	100.0(0.0)	46.2(4.3)
FST-PRJ2-SH	100.0(0.0)	32.4(4.1)	98.8(2.8)	35.3(4.1)	91.6(9.3)	41.8(4.2)
FST-PRJ2-AM	97.5(5.2)	9.1(3.0)	70.3(18.3)	9.9(3.5)	100.0(0.0)	6.6(3.2)
FST-PRJ2-MG	17.5(9.9)	10.1(3.7)	23.8(13.5)	9.5(3.6)	51.2(18.5)	8.7(3.4)
MSPLOT	78.4(12.3)	0.9(1.2)	26.8(13.0)	0.8(1.0)	99.8(1.0)	1.0(1.3)
FOM	0.1(0.8)	0.1(0.3)	0.5(1.9)	0.2(0.6)	43.0(25.5)	0.0(0.2)
FAO	0.2(1.2)	0.1(0.4)	0.5(2.4)	0.2(0.6)	35.1(26.5)	0.0(0.1)

Table B.4: Mean and Standard Deviation (in parentheses) of the TPR and FPR (in percentage) over 200 repetitions for each model. Sample size $n = 100$, evaluation points $t_j = 50$, and contamination rate is 20%. The proposed methods are in italics.

Method	Model 1.2		Model 2.2		Model 2.3	
	TPR	FPR	TPR	FPR	TPR	FPR
FST-MAR	100.0(0.5)	26.3(4.5)	68.0(14.7)	18.4(4.1)	89.1(9.8)	12.6(3.8)
FST-STR	79.2(14.3)	1.2(1.4)	15.0(9.4)	1.9(1.7)	32.5(15.4)	1.3(1.4)
FST-PRJ	95.6(7.2)	0.2(0.5)	7.0(12.7)	0.1(0.5)	45.5(21.4)	0.1(0.4)
FST-PRJ-SH	0.2(1.0)	0.2(0.5)	6.9(12.7)	0.1(0.3)	45.2(21.7)	0.1(0.3)
FST-PRJ-AM	0.0(0.0)	0.0(0.0)	0.1(0.7)	0.0(0.0)	0.3(1.2)	0.0(0.0)
FST-PRJ-MG	95.6(7.2)	0.0(0.0)	0.2(0.9)	0.1(0.4)	0.1(0.6)	0.0(0.2)
FST-PRJ1	96.0(8.4)	3.7(2.4)	22.1(10.2)	0.8(1.0)	55.9(13.4)	0.4(0.7)
FST-PRJ1-SH	3.8(4.6)	3.6(2.3)	21.1(10.0)	0.6(0.8)	54.3(13.2)	0.1(0.4)
FST-PRJ1-AM	0.2(1.0)	0.2(0.6)	0.5(1.5)	0.1(0.3)	2.0(3.1)	0.0(0.2)
FST-PRJ1-MG	95.8(8.6)	0.0(0.0)	0.9(2.1)	0.2(0.6)	1.4(3.0)	0.2(0.5)
FST-PRJ2	100.0(0.0)	50.6(4.7)	88.7(8.0)	38.2(4.6)	98.6(3.2)	27.8(4.9)
FST-PRJ2-SH	46.9(10.7)	47.1(4.5)	83.9(9.8)	30.8(4.2)	96.0(4.9)	19.5(3.8)
FST-PRJ2-AM	12.4(7.8)	12.8(4.2)	27.2(10.9)	10.5(4.0)	40.3(11.8)	9.2(3.6)
FST-PRJ2-MG	100.0(0.0)	6.4(3.1)	14.8(8.1)	9.5(3.2)	22.7(9.5)	8.2(3.5)
MSPLOT	98.2(3.3)	0.2(0.6)	65.2(20.6)	0.7(1.1)	97.9(3.6)	0.8(1.1)
FOM	7.4(12.2)	0.0(0.1)	24.8(18.0)	0.0(0.0)	56.3(24.4)	0.0(0.0)
FAO	33.7(26.0)	0.0(0.0)	9.9(12.0)	0.0(0.0)	23.8(21.9)	0.0(0.0)
Method	Model 3.2		Model 3.3		Model 5.2	
	TPR	FPR	TPR	FPR	TPR	FPR
FST-MAR	99.9(0.7)	18.1(4.2)	92.0(11.3)	16.8(4.0)	100.0(0.0)	25.8(4.6)
FST-STR	69.5(17.7)	1.1(1.2)	23.5(10.9)	1.9(1.5)	99.9(0.7)	1.2(1.4)
FST-PRJ	66.9(18.2)	0.3(0.7)	16.3(13.0)	0.7(0.9)	99.8(2.3)	0.0(0.1)
FST-PRJ-SH	66.9(18.2)	0.3(0.7)	16.3(13.0)	0.7(0.9)	0.0(0.0)	0.0(0.1)
FST-PRJ-AM	2.1(6.6)	0.0(0.1)	0.2(0.9)	0.0(0.2)	99.8(2.3)	0.0(0.1)
FST-PRJ-MG	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.2)	0.0(0.0)	0.0(0.0)
FST-PRJ1	67.2(17.5)	0.5(0.8)	21.1(12.2)	1.1(1.1)	99.9(1.5)	1.1(1.1)
FST-PRJ1-SH	67.2(17.5)	0.3(0.6)	20.7(12.3)	0.9(1.0)	3.2(4.5)	0.9(1.0)
FST-PRJ1-AM	7.2(9.3)	0.0(0.2)	0.7(1.9)	0.0(0.2)	99.9(1.5)	0.0(0.1)
FST-PRJ1-MG	0.1(0.6)	0.2(0.5)	0.5(1.7)	0.2(0.5)	1.2(2.5)	0.2(0.6)
FST-PRJ2	100.0(0.0)	37.8(4.8)	97.6(3.6)	40.6(4.3)	100.0(0.0)	47.6(4.4)
FST-PRJ2-SH	100.0(0.0)	30.2(4.2)	97.1(4.0)	32.9(3.9)	71.7(11.7)	43.5(4.4)
FST-PRJ2-AM	89.1(10.2)	8.9(3.7)	49.1(20.2)	9.6(3.7)	100.0(0.0)	5.4(2.9)
FST-PRJ2-MG	14.8(8.4)	9.9(3.8)	19.4(10.1)	9.8(3.6)	32.7(13.8)	8.6(3.4)
MSPLOT	51.3(16.9)	0.7(1.1)	16.4(10.6)	0.6(1.0)	92.8(7.2)	0.9(1.2)
FOM	0.2(0.9)	0.2(0.6)	0.4(1.4)	0.2(0.6)	3.4(5.3)	0.0(0.2)
FAO	0.2(0.9)	0.3(0.8)	0.4(1.4)	0.2(0.6)	2.2(6.2)	0.2(0.6)

B.4 Additional Simulation Results on Contamination Rates

Tables B.5 - B.7 show the results of the methods outlined in Subsection 5.2.2 on the simulation models in Subsection 5.2.1, for contamination rates 5%, 15%, and 20%.

Table B.5: Mean and Standard Deviation (in parentheses) of the true positive rate (TPR) and the false positive rate (FPR) (in percentage) over 200 repetitions for each model. Sample size $n = 100$, evaluation points $t_j = 50$, and contamination rate is 5%.

Method	Model 0	Model 1		Model 2		Model 3	
	FPR	TPR	FPR	TPR	FPR	TPR	FPR
FST-MAR	26.2(4.2)	100.0(0.0)	25.5(3.9)	100.0(0.0)	19.0(3.8)	100.0(0.0)	18.4(4.4)
FST-STR	4.6(2.5)	100.0(0.0)	3.4(2.0)	95.5(9.5)	3.6(1.9)	100.0(0.0)	3.3(2.1)
FST-PRJ	1.7(2.6)	100.0(0.0)	0.4(0.8)	99.7(2.4)	1.5(1.3)	100.0(0.0)	1.5(1.2)
FST-PRJ-SH	1.7(2.6)	0.4(2.8)	0.4(0.8)	99.7(2.4)	1.5(1.3)	100.0(0.0)	1.4(1.2)
FST-PRJ-AM	0.0(0.3)	0.0(0.0)	0.0(0.0)	0.9(5.0)	0.0(0.1)	100.0(0.0)	0.1(0.2)
FST-PRJ-MG	0.0(0.2)	100.0(0.0)	0.1(0.2)	0.2(2.8)	0.0(0.2)	0.2(2.0)	0.0(0.0)
FST-PRJ1	3.7(1.9)	100.0(0.0)	4.1(2.2)	100.0(0.0)	2.0(1.4)	100.0(0.0)	2.0(1.4)
FST-PRJ1-SH	3.5(1.8)	3.1(7.3)	3.9(2.1)	100.0(0.0)	1.6(1.3)	100.0(0.0)	1.7(1.3)
FST-PRJ1-AM	0.2(0.4)	0.1(1.4)	0.2(0.6)	5.6(10.4)	0.1(0.4)	100.0(0.0)	0.1(0.2)
FST-PRJ1-MG	0.1(0.4)	100.0(0.0)	0.1(0.3)	2.8(8.0)	0.3(0.6)	3.4(8.0)	0.2(0.5)
FST-PRJ2	52.2(3.6)	100.0(0.0)	52.1(3.6)	100.0(0.0)	44.5(4.2)	100.0(0.0)	43.8(3.8)
FST-PRJ2-SH	47.0(3.3)	45.4(22.7)	47.4(3.7)	100.0(0.0)	37.6(3.6)	100.0(0.0)	37.4(3.5)
FST-PRJ2-AM	12.4(3.6)	11.8(13.2)	12.7(3.9)	70.3(21.4)	11.9(3.6)	100.0(0.0)	9.5(3.2)
FST-PRJ2-MG	14.0(4.1)	100.0(0.0)	10.8(3.4)	37.3(22.3)	9.9(3.0)	56.2(23.8)	9.6(3.5)
MSPLOT	1.6(1.7)	100.0(0.0)	0.8(1.1)	100.0(0.0)	1.3(1.3)	100.0(0.0)	1.1(1.3)
FOM	0.3(0.6)	100.0(0.0)	0.2(0.5)	99.1(5.0)	0.1(0.4)	99.2(4.4)	0.2(0.5)
FAO	0.3(0.6)	100.0(0.0)	0.1(0.3)	93.2(14.2)	0.1(0.3)	96.1(12.3)	0.2(0.5)

Method	Model 4		Model 5		Model 6	
	TPR	FPR	TPR	FPR	TPR	FPR
FST-MAR	92.1(13.4)	19.9(3.9)	100.0(0.0)	24.4(4.1)	100.0(0.0)	17.9(4.0)
FST-STR	63.4(29.4)	3.5(1.9)	100.0(0.0)	4.6(2.3)	100.0(0.0)	3.4(1.8)
FST-PRJ	51.0(37.7)	0.8(1.3)	100.0(0.0)	0.2(0.5)	98.4(9.7)	1.2(1.1)
FST-PRJ-SH	49.2(37.2)	0.8(1.3)	0.0(0.0)	0.1(0.4)	93.5(17.8)	1.0(1.1)
FST-PRJ-AM	0.1(1.4)	0.0(0.1)	100.0(0.0)	0.1(0.3)	15.7(24.5)	0.1(0.3)
FST-PRJ-MG	3.9(12.5)	0.1(0.3)	71.1(29.0)	0.0(0.1)	94.3(18.0)	0.1(0.3)
FST-PRJ1	77.1(20.9)	2.0(1.3)	100.0(0.0)	3.5(1.8)	100.0(0.0)	1.8(1.3)
FST-PRJ1-SH	74.8(21.9)	1.7(1.3)	0.0(0.0)	3.3(1.7)	99.7(2.4)	1.6(1.2)
FST-PRJ1-AM	0.3(2.4)	0.1(0.4)	100.0(0.0)	0.1(0.3)	27.7(24.9)	0.1(0.4)
FST-PRJ1-MG	9.4(13.6)	0.2(0.4)	95.4(9.8)	0.2(0.5)	97.1(7.6)	0.1(0.4)
FST-PRJ2	99.7(2.4)	44.4(3.8)	100.0(0.0)	51.7(3.8)	100.0(0.0)	45.5(3.9)
FST-PRJ2-SH	99.1(4.2)	37.7(3.5)	0.9(4.2)	47.9(3.7)	100.0(0.0)	40.3(3.5)
FST-PRJ2-AM	14.4(15.0)	13.1(4.1)	100.0(0.0)	9.2(3.5)	98.2(6.1)	11.7(3.8)
FST-PRJ2-MG	50.6(27.2)	9.5(3.5)	100.0(0.0)	7.9(3.1)	100.0(0.0)	9.2(3.4)
MSPLOT	53.0(26.7)	1.3(1.5)	100.0(0.0)	1.1(1.3)	99.1(4.2)	1.0(1.2)
FOM	7.3(13.3)	0.2(0.5)	100.0(0.0)	0.2(0.4)	94.7(13.1)	0.2(0.6)
FAO	6.8(14.5)	0.1(0.4)	100.0(0.0)	0.1(0.4)	84.6(26.0)	0.2(0.5)

Table B.6: Mean and Standard Deviation (in parentheses) of the true positive rate (TPR) and the false positive rate (FPR) (in percentage) over 200 repetitions for each model. Sample size $n = 100$, evaluation points $t_j = 50$, and contamination rate is 15%.

Method	Model 0	Model 1		Model 2		Model 3	
	FPR	TPR	FPR	TPR	FPR	TPR	FPR
FST-MAR	26.2(4.2)	100.0(0.0)	25.0(4.4)	99.1(2.6)	9.3(3.1)	100.0(0.0)	8.3(3.0)
FST-STR	4.6(2.5)	99.6(1.7)	1.5(1.5)	79.4(15.4)	1.7(1.7)	100.0(0.0)	1.3(1.3)
FST-PRJ	1.7(2.6)	100.0(0.0)	0.1(0.3)	92.7(7.6)	0.2(0.5)	100.0(0.0)	0.2(0.6)
FST-PRJ-SH	1.7(2.6)	0.2(1.0)	0.1(0.3)	92.7(7.6)	0.2(0.5)	100.0(0.0)	0.2(0.6)
FST-PRJ-AM	0.0(0.3)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	99.9(0.9)	0.0(0.0)
FST-PRJ-MG	0.0(0.2)	100.0(0.0)	0.0(0.0)	0.0(0.5)	0.0(0.0)	0.0(0.0)	0.0(0.0)
FST-PRJ1	3.7(1.9)	100.0(0.0)	3.7(2.2)	94.3(7.2)	0.5(0.7)	100.0(0.0)	0.5(0.8)
FST-PRJ1-SH	3.5(1.8)	4.1(5.2)	3.7(2.2)	93.2(7.3)	0.3(0.6)	100.0(0.0)	0.3(0.5)
FST-PRJ1-AM	0.2(0.4)	0.2(1.4)	0.1(0.4)	4.0(5.0)	0.1(0.3)	99.8(1.2)	0.0(0.0)
FST-PRJ1-MG	0.1(0.4)	100.0(0.0)	0.0(0.0)	2.6(3.9)	0.2(0.4)	1.2(2.8)	0.2(0.6)
FST-PRJ2	52.2(3.6)	100.0(0.0)	50.8(4.3)	100.0(0.0)	27.3(4.1)	100.0(0.0)	25.0(3.6)
FST-PRJ2-SH	47.0(3.3)	48.2(13.0)	47.3(4.3)	99.9(0.7)	19.1(3.3)	100.0(0.0)	16.9(2.8)
FST-PRJ2-AM	12.4(3.6)	12.0(7.5)	13.0(4.2)	61.8(13.5)	9.0(3.3)	100.0(0.0)	3.6(2.2)
FST-PRJ2-MG	14.0(4.1)	100.0(0.0)	5.1(2.6)	35.7(12.6)	7.8(3.4)	37.0(15.3)	9.1(3.3)
MSPLOT	1.6(1.7)	100.0(0.0)	0.2(0.5)	100.0(0.0)	0.8(1.2)	100.0(0.0)	1.0(1.3)
FOM	0.3(0.6)	100.0(0.0)	0.0(0.1)	90.8(14.5)	0.0(0.1)	6.6(14.0)	0.0(0.1)
FAO	0.3(0.6)	100.0(0.0)	0.0(0.1)	70.3(24.0)	0.0(0.1)	2.2(8.4)	0.0(0.2)

Method	Model 4		Model 5		Model 6	
	TPR	FPR	TPR	FPR	TPR	FPR
FST-MAR	54.0(16.6)	12.9(3.5)	100.0(0.0)	27.2(4.2)	100.0(0.0)	11.3(3.2)
FST-STR	33.6(23.3)	1.9(1.9)	100.0(0.0)	4.2(2.6)	98.6(3.9)	1.3(1.5)
FST-PRJ	1.7(4.0)	0.0(0.2)	100.0(0.0)	0.2(0.5)	94.7(8.0)	0.2(0.5)
FST-PRJ-SH	1.7(4.0)	0.0(0.1)	0.0(0.0)	0.2(0.5)	79.7(20.9)	0.2(0.5)
FST-PRJ-AM	0.0(0.0)	0.0(0.2)	100.0(0.0)	0.0(0.1)	0.1(0.7)	0.0(0.0)
FST-PRJ-MG	0.0(0.5)	0.0(0.0)	18.0(20.4)	0.0(0.0)	69.5(28.4)	0.0(0.2)
FST-PRJ1	18.9(12.0)	0.7(1.0)	100.0(0.0)	4.3(2.2)	98.0(4.5)	0.4(0.7)
FST-PRJ1-SH	15.5(10.8)	0.5(0.8)	0.0(0.0)	4.3(2.2)	93.8(8.1)	0.4(0.6)
FST-PRJ1-AM	0.2(1.0)	0.1(0.3)	100.0(0.0)	0.0(0.1)	8.3(10.0)	0.0(0.3)
FST-PRJ1-MG	3.6(6.3)	0.1(0.4)	73.5(15.0)	0.1(0.2)	76.6(18.3)	0.0(0.3)
FST-PRJ2	86.2(10.8)	32.3(4.3)	100.0(0.0)	53.9(3.8)	100.0(0.0)	32.9(4.1)
FST-PRJ2-SH	78.3(12.7)	24.0(3.7)	0.9(2.4)	52.1(3.7)	99.9(0.7)	27.8(3.8)
FST-PRJ2-AM	13.9(9.2)	13.0(4.2)	100.0(0.0)	3.1(2.2)	88.6(10.7)	8.7(3.5)
FST-PRJ2-MG	34.9(18.5)	7.9(3.6)	98.1(4.0)	5.0(2.4)	98.6(3.3)	5.8(2.9)
MSPLOT	13.9(13.7)	0.6(1.0)	100.0(0.0)	0.9(1.2)	77.6(14.2)	0.8(1.3)
FOM	0.3(1.5)	0.1(0.4)	95.1(12.1)	0.0(0.1)	2.6(7.0)	0.0(0.1)
FAO	0.6(2.4)	0.2(0.6)	84.5(24.4)	0.0(0.0)	2.1(7.7)	0.0(0.1)

Table B.7: Mean and Standard Deviation (in parentheses) of the true positive rate (TPR) and the false positive rate (FPR) (in percentage) over 200 repetitions for each model. Sample size $n = 100$, evaluation points $t_j = 50$, and contamination rate is 20%.

Method	Model 0	Model 1		Model 2		Model 3	
	FPR	TPR	FPR	TPR	FPR	TPR	FPR
FST-MAR	26.2(4.2)	100.0(0.0)	24.9(4.3)	91.0(8.4)	6.0(2.6)	100.0(0.0)	3.9(2.4)
FST-STR	4.6(2.5)	97.5(5.3)	0.8(1.2)	52.5(17.9)	1.1(1.2)	100.0(0.0)	0.9(1.3)
FST-PRJ	1.7(2.6)	100.0(0.0)	0.1(0.4)	66.3(16.7)	0.0(0.2)	100.0(0.0)	0.0(0.2)
FST-PRJ-SH	1.7(2.6)	0.2(1.1)	0.1(0.4)	66.3(16.8)	0.0(0.2)	100.0(0.0)	0.0(0.2)
FST-PRJ-AM	0.0(0.3)	0.0(0.0)	0.0(0.0)	0.2(0.9)	0.0(0.0)	89.8(15.9)	0.0(0.0)
FST-PRJ-MG	0.0(0.2)	100.0(0.0)	0.0(0.1)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
FST-PRJ1	3.7(1.9)	100.0(0.0)	3.9(2.3)	73.4(13.5)	0.2(0.5)	100.0(0.0)	0.2(0.5)
FST-PRJ1-SH	3.5(1.8)	3.7(4.6)	3.8(2.3)	71.8(13.4)	0.0(0.2)	100.0(0.0)	0.0(0.2)
FST-PRJ1-AM	0.2(0.4)	0.2(1.0)	0.1(0.4)	2.8(3.8)	0.1(0.3)	95.4(6.9)	0.0(0.0)
FST-PRJ1-MG	0.1(0.4)	100.0(0.0)	0.0(0.0)	1.9(3.3)	0.1(0.4)	0.8(2.0)	0.2(0.5)
FST-PRJ2	52.2(3.6)	100.0(0.0)	49.3(4.3)	99.7(1.2)	19.9(4.0)	100.0(0.0)	15.4(3.9)
FST-PRJ2-SH	47.0(3.3)	49.6(11.2)	46.3(4.6)	98.8(2.6)	10.8(2.6)	100.0(0.0)	6.5(2.0)
FST-PRJ2-AM	12.4(3.6)	12.8(8.3)	12.4(4.1)	60.9(12.8)	7.9(3.4)	100.0(0.0)	2.5(2.1)
FST-PRJ2-MG	14.0(4.1)	100.0(0.0)	3.5(2.3)	33.4(11.3)	7.2(3.0)	24.7(12.1)	8.6(3.6)
MSPLOT	1.6(1.7)	100.0(0.0)	0.1(0.4)	100.0(0.0)	0.7(1.0)	98.7(2.8)	0.9(1.4)
FOM	0.3(0.6)	99.9(1.1)	0.0(0.0)	76.2(24.1)	0.0(0.0)	0.0(0.0)	0.0(0.2)
FAO	0.3(0.6)	99.8(1.4)	0.0(0.1)	41.7(27.7)	0.0(0.0)	0.0(0.0)	0.5(1.0)

Method	Model 4		Model 5		Model 6	
	TPR	FPR	TPR	FPR	TPR	FPR
FST-MAR	38.5(12.2)	12.0(3.5)	100.0(0.0)	28.4(4.7)	99.6(1.7)	9.5(3.4)
FST-STR	20.3(15.0)	1.7(1.6)	100.0(0.0)	4.4(2.5)	85.8(14.2)	0.8(1.1)
FST-PRJ	0.3(1.2)	0.0(0.1)	100.0(0.0)	0.2(0.5)	69.1(18.6)	0.0(0.2)
FST-PRJ-SH	0.3(1.2)	0.0(0.1)	0.0(0.0)	0.2(0.5)	57.0(21.4)	0.0(0.2)
FST-PRJ-AM	0.0(0.0)	0.0(0.0)	100.0(0.0)	0.0(0.0)	0.0(0.4)	0.0(0.0)
FST-PRJ-MG	0.0(0.0)	0.0(0.0)	3.4(6.4)	0.0(0.0)	24.8(25.1)	0.0(0.0)
FST-PRJ1	8.6(6.6)	0.6(0.9)	100.0(0.0)	4.9(2.5)	87.9(10.7)	0.1(0.4)
FST-PRJ1-SH	7.1(6.1)	0.4(0.7)	0.0(0.0)	4.8(2.4)	77.8(13.4)	0.1(0.3)
FST-PRJ1-AM	0.2(1.0)	0.1(0.4)	100.0(0.0)	0.0(0.0)	2.6(3.9)	0.0(0.2)
FST-PRJ1-MG	1.5(3.0)	0.1(0.4)	43.5(16.9)	0.0(0.2)	43.6(21.4)	0.0(0.1)
FST-PRJ2	72.3(10.9)	30.0(4.8)	100.0(0.0)	56.3(4.1)	100.0(0.0)	28.7(4.6)
FST-PRJ2-SH	60.5(10.9)	21.8(3.8)	0.9(1.9)	54.9(4.0)	99.8(1.2)	23.4(4.1)
FST-PRJ2-AM	14.2(8.1)	12.4(4.3)	100.0(0.0)	1.1(1.2)	67.5(16.2)	8.6(3.9)
FST-PRJ2-MG	23.7(12.1)	7.1(3.3)	90.5(8.9)	4.3(2.4)	94.0(8.4)	4.7(2.6)
MSPLOT	7.2(8.5)	0.6(1.0)	99.8(1.1)	0.9(1.3)	55.5(17.8)	0.8(1.1)
FOM	0.3(1.2)	0.1(0.4)	26.8(21.2)	0.0(0.0)	0.0(0.4)	0.0(0.3)
FAO	0.5(1.7)	0.2(0.6)	29.0(33.1)	0.0(0.0)	0.0(0.4)	0.3(0.7)

B.5 Comparison of Various Thresholds Q

Now, we show the distribution of the F1 scores when Fast-MUOD with projections (FST-PRJ1) is used on the models presented in Subsection 5.2.1 with different threshold values of $Q = (\tau_S, \tau_A, \tau_M)$ ranging from $Q = (0.2, 0.2, 0.2)$ to $Q = (0.7, 0.7, 0.7)$. For Model 0 with no outliers, we show only the FPRs.

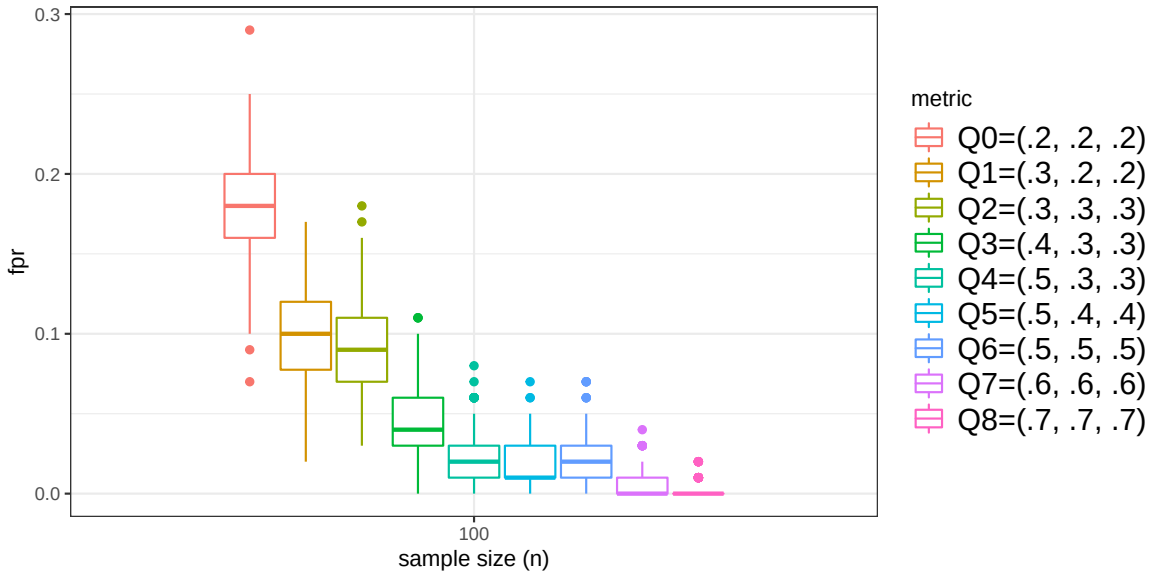

Figure B.3: The FPRs of Fast-MUOD with projections (FST-PRJ1) using different threshold values $Q = (\tau_S, \tau_A, \tau_M)$ on Model 0.

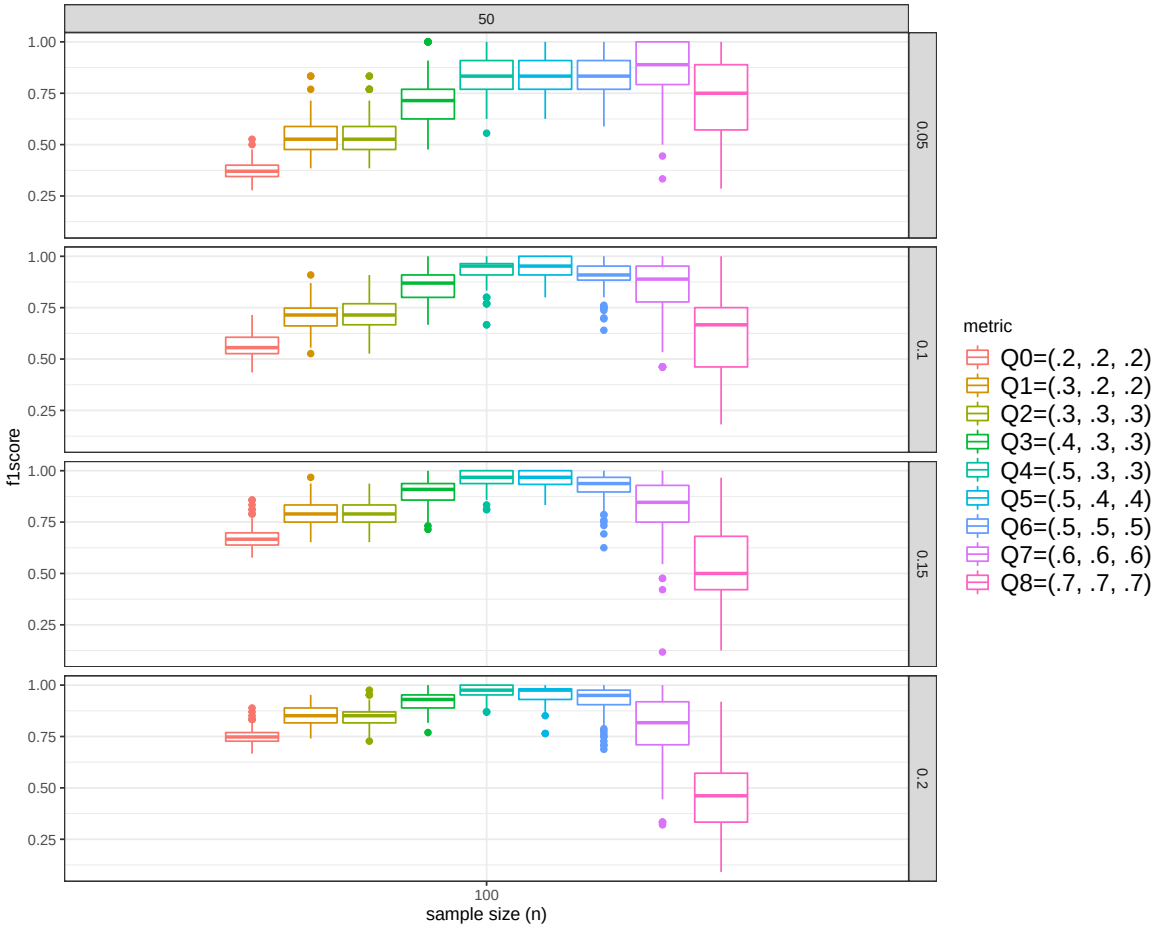

Figure B.4: F1 scores of Fast-MUOD with projections (FST-PRJ1) using different threshold values $Q = (\tau_S, \tau_A, \tau_M)$ on Model 1. The horizontal facets indicate the different contamination rates considered $(0.05, 0.1, 0.15, 0.2)$.

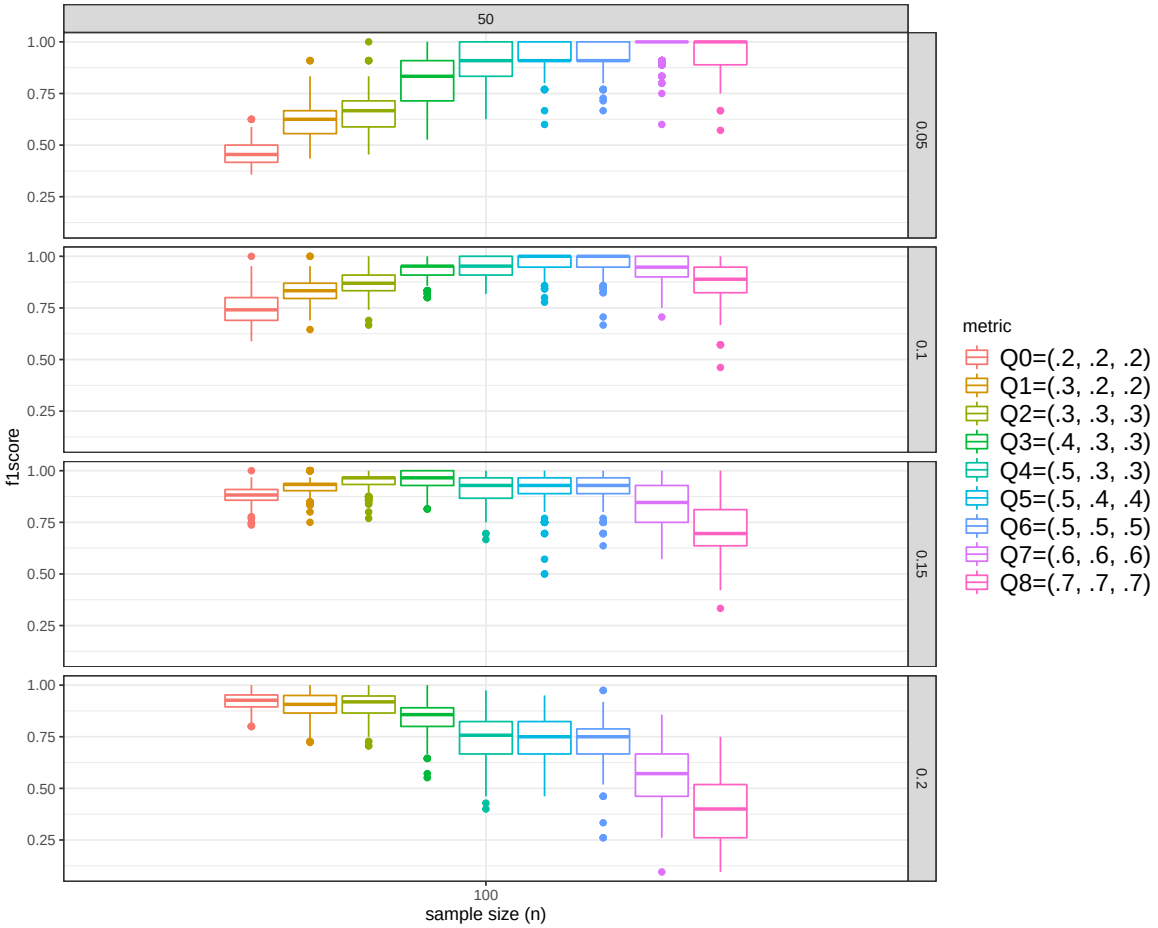

Figure B.5: F1 scores of Fast-MUOD with projections (FST-PRJ1) using different threshold values $Q = (\tau_S, \tau_A, \tau_M)$ on Model 2. The horizontal facets indicate the different contamination rates considered $(0.05, 0.1, 0.15, 0.2)$.

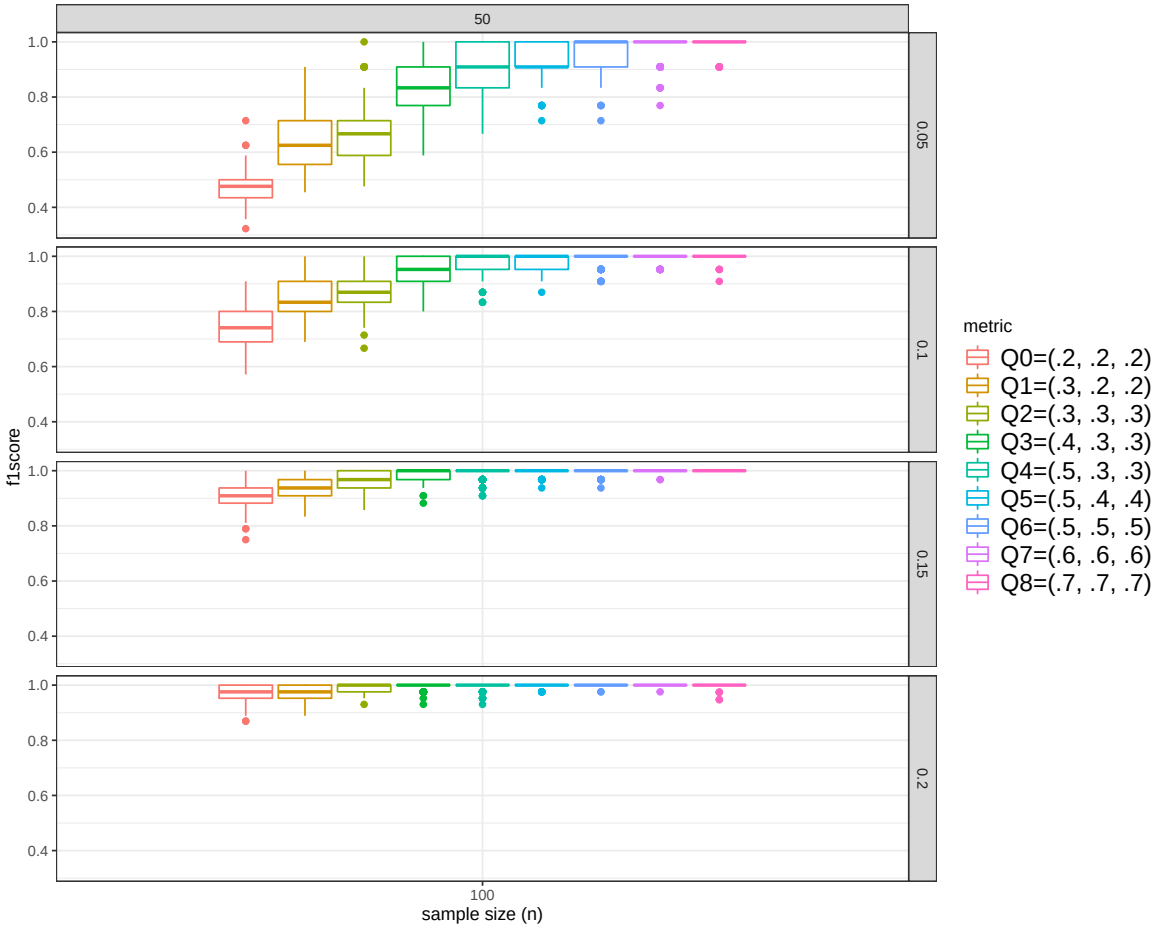

Figure B.6: F1 scores of Fast-MUOD with projections (FST-PRJ1) using different threshold values $Q = (\tau_S, \tau_A, \tau_M)$ on Model 3. The horizontal facets indicate the different contamination rates considered $(0.05, 0.1, 0.15, 0.2)$.

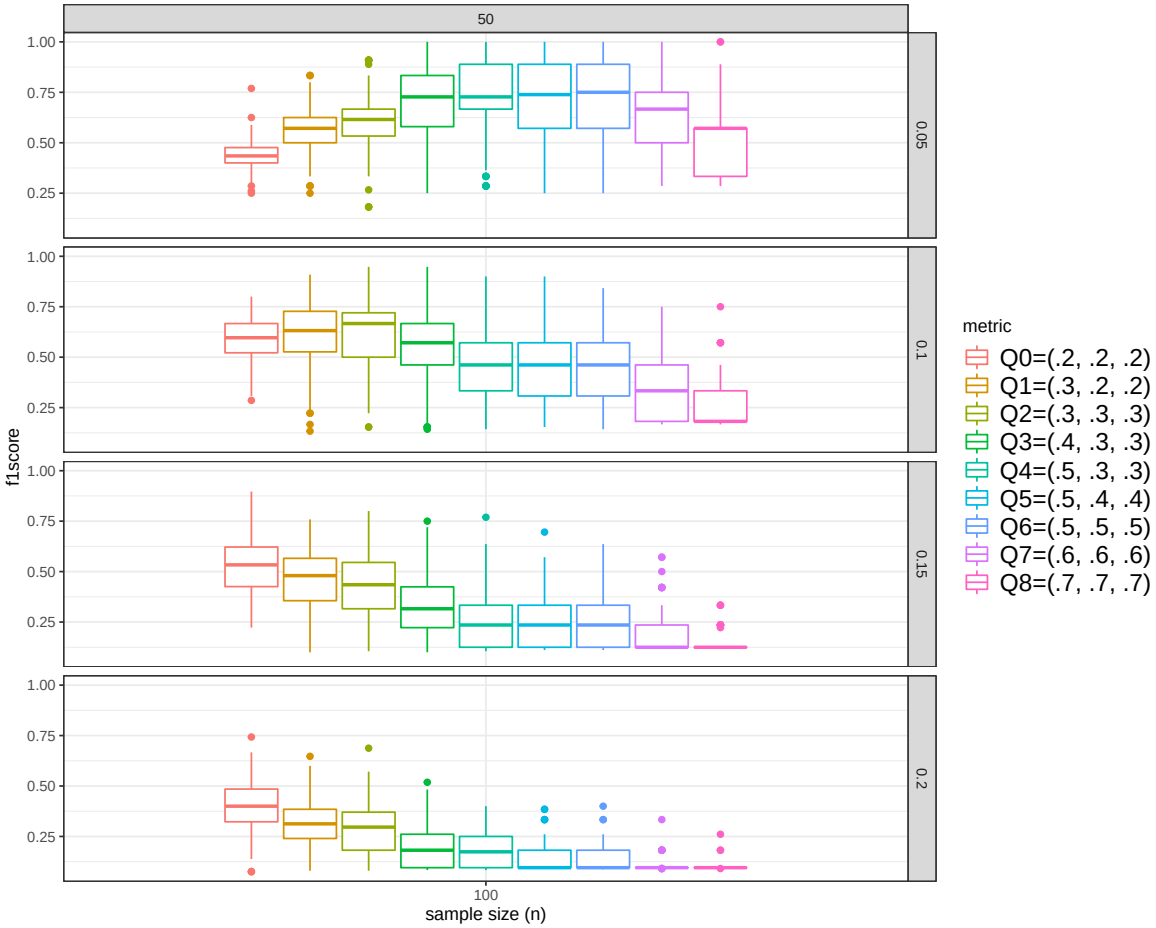

Figure B.7: F1 scores of Fast-MUOD with projections (FST-PRJ1) using different threshold values $Q = (\tau_S, \tau_A, \tau_M)$ on Model 4. The horizontal facets indicate the different contamination rates considered $(0.05, 0.1, 0.15, 0.2)$.

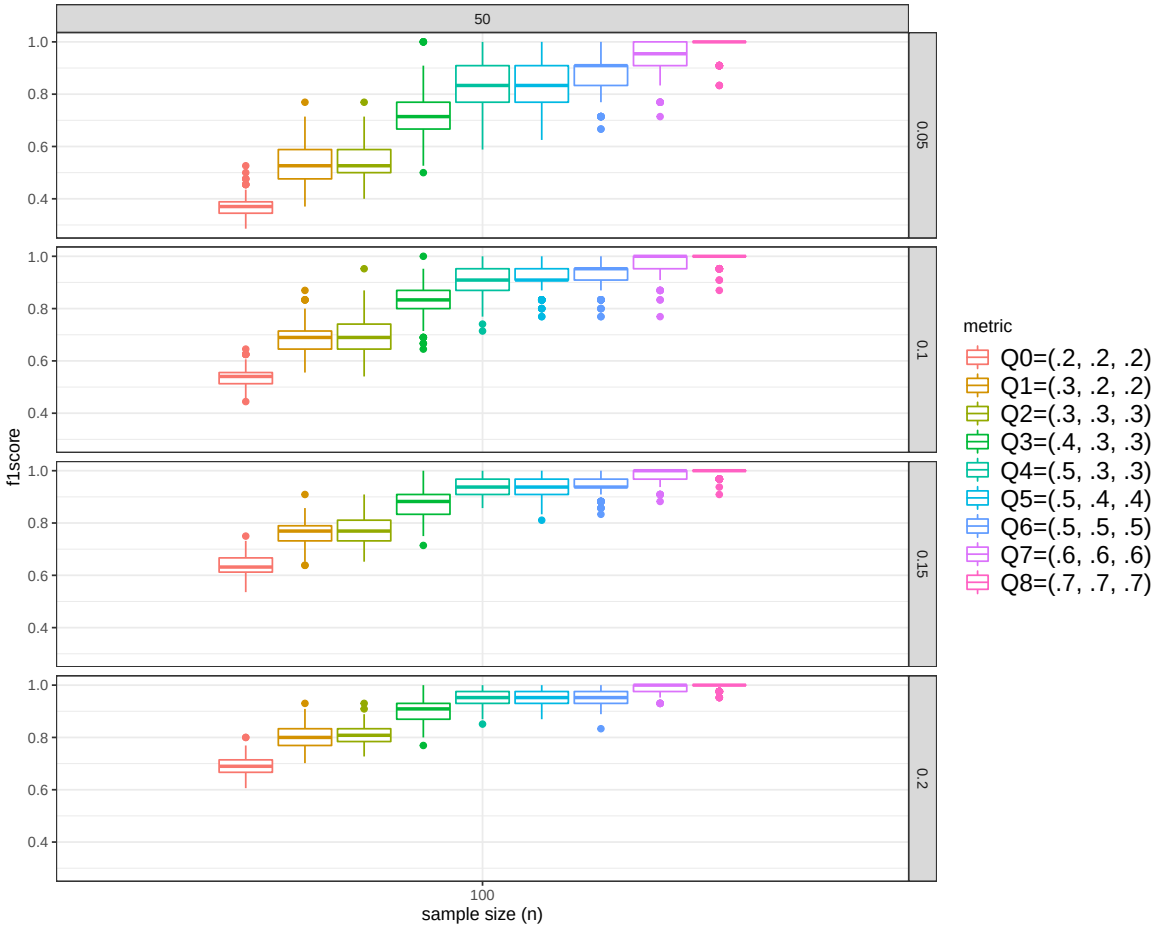

Figure B.8: F1 scores of Fast-MUOD with projections (FST-PRJ1) using different threshold values $Q = (\tau_S, \tau_A, \tau_M)$ on Model 5. The horizontal facets indicate the different contamination rates considered $(0.05, 0.1, 0.15, 0.2)$.

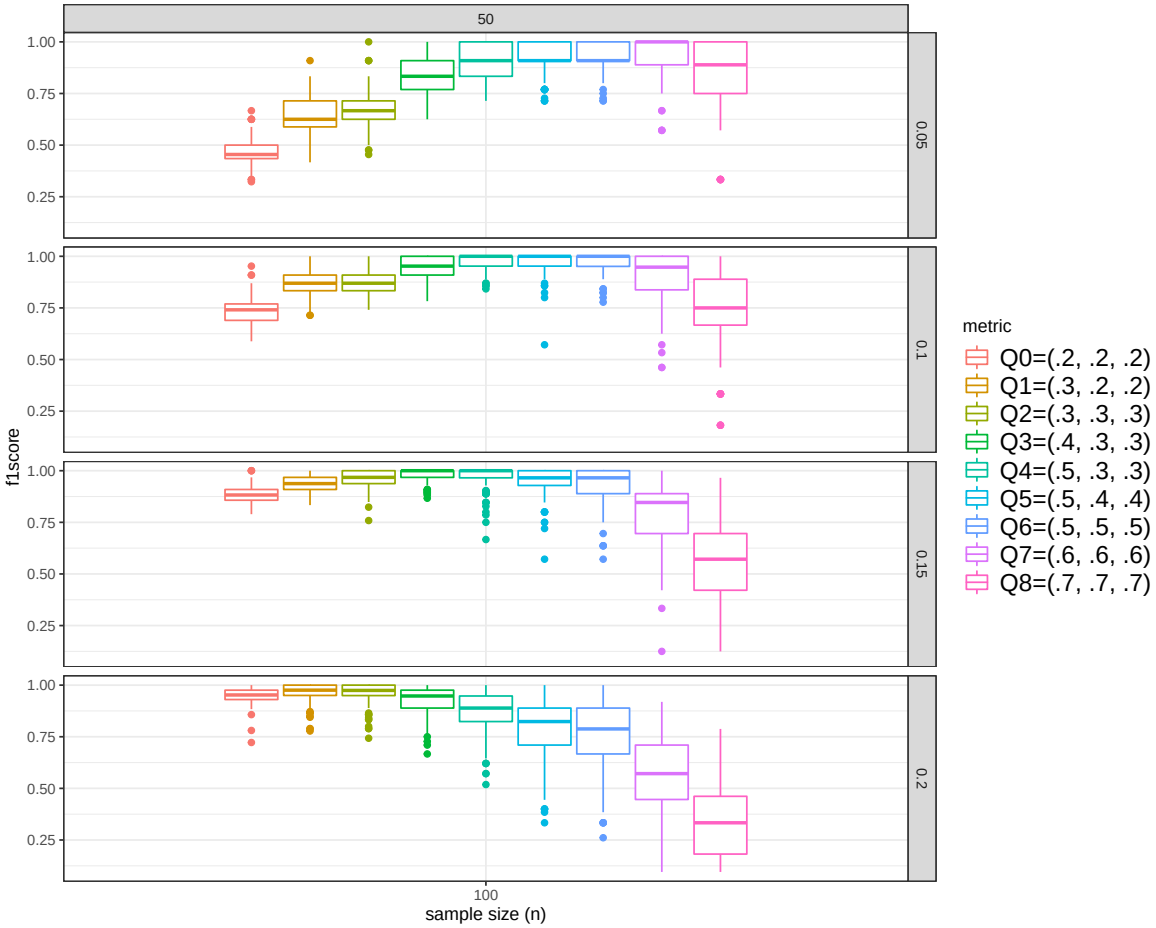

Figure B.9: F1 scores of Fast-MUOD with projections (FST-PRJ1) using different threshold values $Q = (\tau_S, \tau_A, \tau_M)$ on Model 6. The horizontal facets indicate the different contamination rates considered $(0.05, 0.1, 0.15, 0.2)$.

B.6 Character Data: Letter “i"

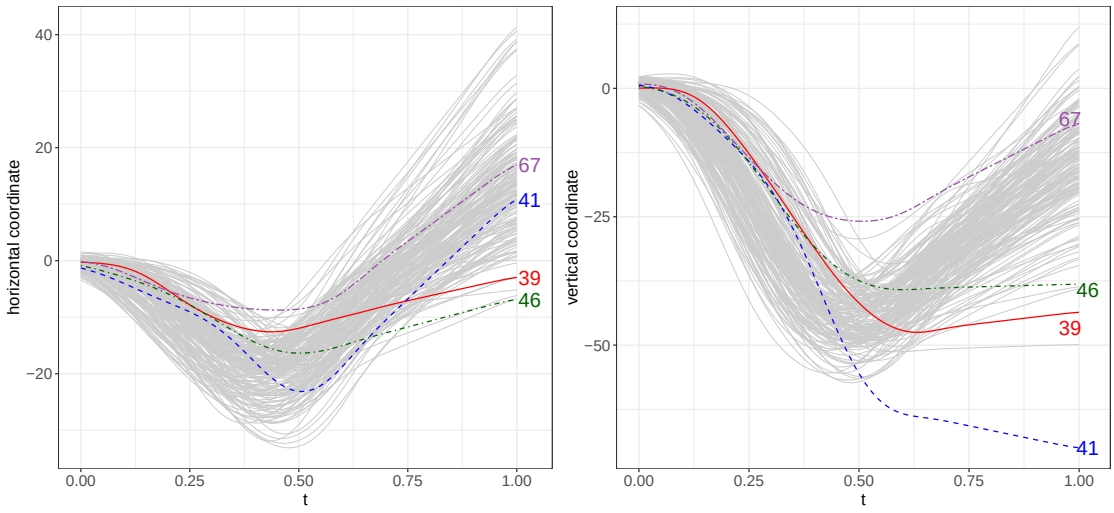

Figure B.10: The horizontal and vertical coordinates of the magnitude and amplitude outliers.

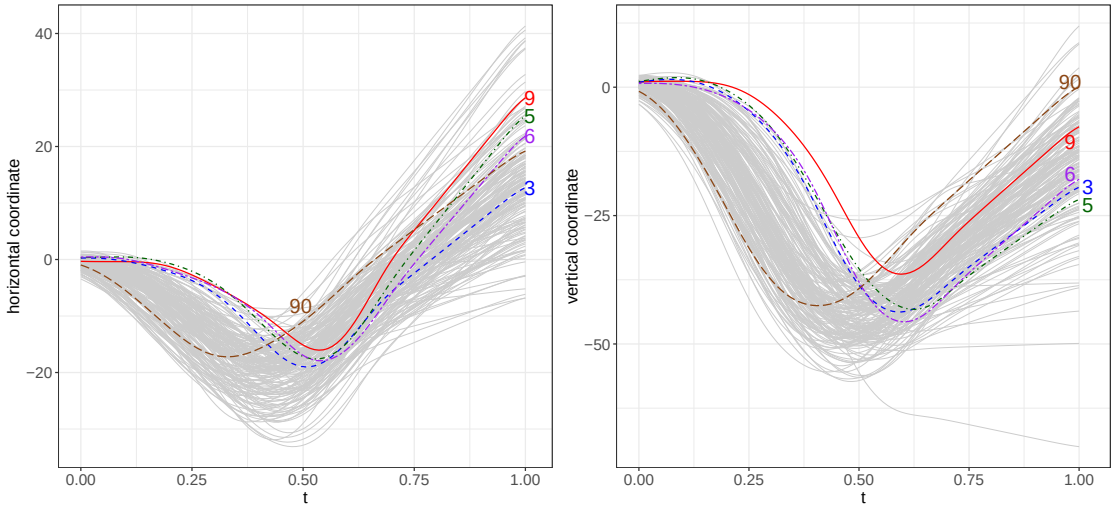

Figure B.11: Some shape outliers: curves 3, 5, 6, 9 and 90 with horizontal shift.

B.7 Character Data: Letter “a"

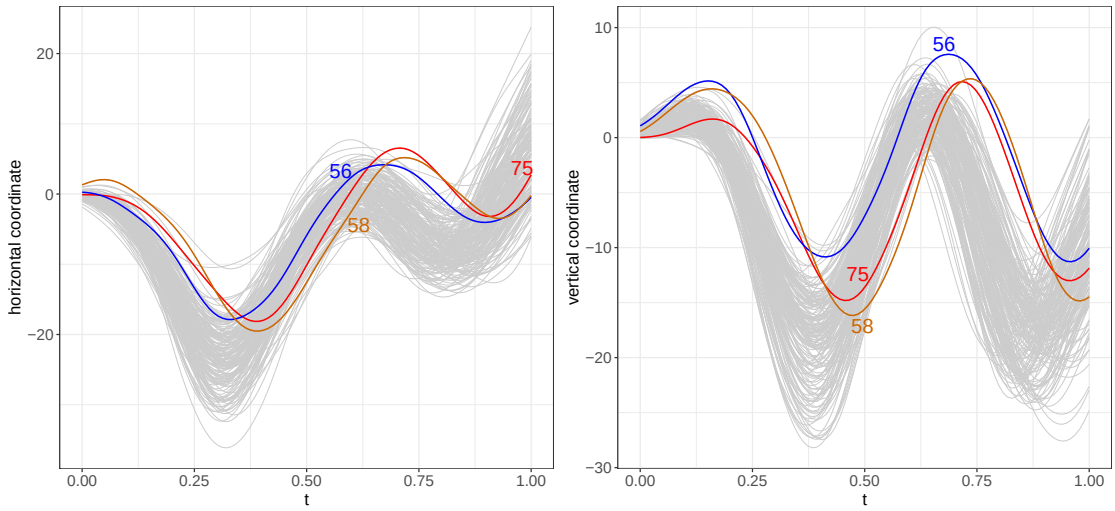

Figure B.12: Some shape outliers detected by Fast-MUOD with a shift to the right in peaks resulting in “short follow-throughs". See Figure 5.6 of the thesis.

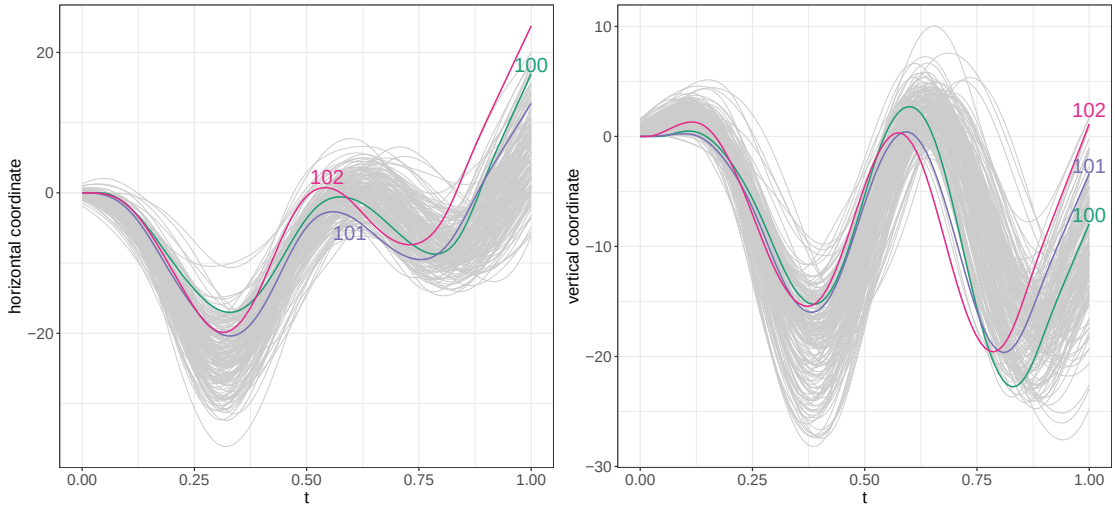

Figure B.13: Some shape outliers detected by Fast-MUOD with a shift to the left in peaks resulting in “long follow-throughs". See Figure 5.6 of the thesis.

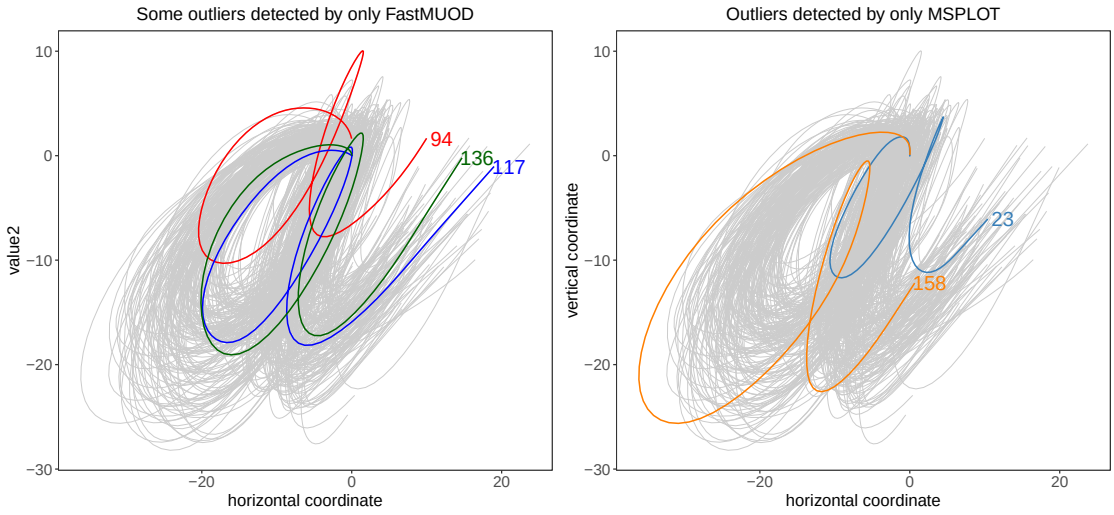

Figure B.14: Outliers detected by only Fast-MUOD and only MSPLOT.