Detecting demand outliers in transport systems

Nicola Rennie, B.Sc.(Hons.), M.Res

Submitted for the degree of Doctor of

Philosophy at Lancaster University.

August 2021

STOR-i

excellence with impact

Abstract

Optimisation routines used for demand management in transport systems strongly depend on accurate forecasts. Outliers caused by systematic shifts in demand cause erroneous forecasts for both current services and future services whose forecasts are based on historic demand. Transport service providers often rely on analysts to identify outlier demand and make adjustments accordingly. However, previous research on judgemental forecasting shows that such adjustments can be biased and even superfluous. Literature on automated detection and evaluation of outlier demand in this context is scarce.

To date, most literature on forecasting and optimisation in transport planning does not account for demand outliers despite the negative impacts it can have. This thesis presents a novel methodology, which combines network clustering with functional data analysis and time series forecasting, to detect outliers in demand for transport systems. This thesis also contributes a simulation framework for evaluating the performance of the proposed outlier detection procedure and for quantifying the effects of outlier demand on different optimisation routines. The use of such a method as a decision support tool for analyst adjusted forecasts, and how the outlier alerts may

be best communicated, is also considered. Computational studies highlight the benefits of different adjustments that analysts may take after the identification of outlier demand. Multiple empirical studies will demonstrate how the method can be applied in practice to different types of transport systems, with analyses of Deutsche Bahn railway booking data and Capital Bikeshare usage data.

Acknowledgements

I would first like to thank STOR-i Centre for Doctoral Training for all the opportunities that they have given me throughout my MRes and PhD. The financial support from EPSRC is also gratefully acknowledged. Over the last four years, I have had the pleasure of meeting many wonderful people through STOR-i and I am thankful for each and every one of them.

I am also exceptionally grateful for my incredible supervisors. This work would not have happened without your support, and I have learnt so much from all of you. Dr. Catherine Cleophas – thank you for being a wonderful mentor and someone I will always look up to. Despite never quite making it to Germany for an extended visit, I still feel a part of the research community in Kiel. Dr. Adam Sykulski – I couldn't think of a better person to have added to my supervisory team. Thanks for always being around to chat, and for reminding me to take breaks and explore the world outside of my PhD. Dr. Florian Dost – your never-ending support and encouragement have always brightened up my day.

Thank you to Deutsche Bahn for providing data and an outlet for my work in the real world. I'm particularly grateful to Philipp Bartke and Valentin Wagner for many

helpful conversations, and your continuing enthusiasm for my work.

Finally, I am eternally grateful to everyone who has supported me along my PhD journey. From the people who have listened to me vent my frustrations, to the ones I turn to to celebrate with, and everyone in between – you have made the last few years possible.

Declaration

I declare that the work in this thesis has been done by myself and has not been submitted elsewhere for the award of any other degree.

This thesis is constructed as a series of papers. Chapters 2, 3, and 4 should therefore be read as separate entities. The appendices relating to each of these papers are included as separate appendices at the end of the thesis.

Chapter 2 has been published as N. Rennie, C. Cleophas, A. M. Sykulski, and F. Dost. Identifying and responding to outlier demand in revenue management. *European Journal of Operational Research*, 293:1015–1030, 2021.

Chapter 3 has been submitted for publication as N. Rennie, C. Cleophas, A. M. Sykulski, and F. Dost (2021). Detecting outlying demand in multi-leg bookings for transportation networks. *European Journal of Operational Research*. This publication is currently under revision.

Chapter 4 has been submitted for publication as N. Rennie, C. Cleophas, A. M. Sykulski, and F. Dost (2021). Analysing and visualising bike sharing demand with outliers. *Transportation Research Part D: Transport and Environment*.

The word count for this thesis is 36,846.

Nicola Rennie

Contents

List of Figures

List of Tables

List of Abbreviations

AIC	Akaike Information Criterion
ANOVA	Analysis of variance
ARIMA	Autoregressive Integrated Moving Average
BCR	Balanced Classification Rate
BL	Booking Limit
CDF	Cumulative Distribution Function
CV-MSE	Cross-Validated Mean Integrated Squared Error
CV-SSE	Cross-Validated Sum of Integrated Squared Errors
EMSRb	Expected Marginal Seat Revenue - b
EMSRb-MR	Expected Marginal Seat Revenue - b with Marginal Revenue
EVT	Extreme Value Theory
FCFS	First-Come-First-Served
FDR	False Discovery Rate
FN	False Negative
FN	False Negative Rate
FP	False Positive

FPR	False Positive Rate
GPD	Generalised Pareto Distribution
HD	Halfspace Depth
IGARCH	Integrated Generalised Autoregressive Conditional Heteroskedasticity
MAD	Median Absolute Deviation
MFHD	Multivariate Functional Halfspace Depth
MST	Minimum Spanning Tree
NMI	Normalised Mutual Information
OD	Origin Destination
PCA	Principal Component Analysis
PL	Protection Level
ROC	Receiver Operating Characteristic
RMSE	Revenue Management
RM	Root Mean Square Error
SES	Simple Exponential Smoothing
SDCS	Standard Deviation of Cluster Size
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate

Chapter 1

Introduction

1.1 Motivation

The rising number of public transport passengers (Buehler and Pucher, 2012) makes accurate demand management and planning more important than ever. Transport service providers often utilise a combination of demand forecasting and optimisation to make decisions on issues such as fare levels, ticket availability, scheduling, and resource distribution. However, the resulting outputs from the optimisation routines are only optimal if the forecast is not erroneous. Forecasts can be inaccurate for several reasons (Cleophas et al., 2017): (i) the unavoidable variance of demand overtime prohibits perfect forecasts; (ii) Flaws in the chosen forecast model (such as the predictive time series component or the customer choice model) cause model-based forecast errors; (iii) Systemic changes in the market can cause short-term, shifts in demand. We term this final phenomenon as *outlier demand*.

When outlier demand occurs, it affects the planning process in two ways: (i) in

the forecasting component, as historic outlier demand contaminates future forecasts; and (ii) current outlier demand causes the output of the optimisation to be non-optimal. Although approaches such as robust optimisation may help to implicitly protect against effects from outlier demand, these approaches are often intractable and not often used in practice. Instead, we propose that outliers be explicitly identified as they occur and the forecasts corrected.

The use of analysts' expert judgement to make manual adjustments is an important aspect of demand forecasting (Banerjee et al., 2020; Currie and Rowley, 2010; Schütze et al., 2020). Manual adjustments to forecasts can contribute significant improvements to forecasting, and therefore optimisation Zeni (2003). However, previous research (Lawrence et al., 2006; De Baets and Harvey, 2020) has shown that such adjustments can be biased and even unnecessary. To avoid superfluous adjustments, and to make better use of resources, we propose the use of automated outlier detection as a tool to aid in identifying when a forecast adjustment is necessary. Such decision support tools can improve analyst judgement by reducing the complexity of the decisions, as discussed by Perera et al. (2019).

1.1.1 Outlier detection for transport demand planning

Literature on the detecting demand outliers in transport planning is limited. Talvitie and Kirshner (1978) consider the effects of outliers on the forecasting of demand for different modes of urban transport, but implement a simplistic trimming method to identify outliers. Guo et al. (2015) propose a procedure for identifying road traffic level outliers in an online setting based on the conditional variance of predictions.

They find that accounting for such outliers in future forecasts increases the systems performance. Neumann-Saavedra et al. (2021) show that service levels can be improved when adjustments are made to the optimal redistribution plan in bike-sharing systems if demand differs from that forecasted. Rashed et al. (2017) discuss exploiting knowledge of detected outliers to convey information about the data generating process to policy makers, when forecasting container throughput at ports.

1.1.2 Outlier detection for revenue management

Revenue management (RM) solves an optimisation problem (usually revenue maximisation), where firms decide on prices or availability of perishable products based on a demand forecast. Although the practice of revenue management is not restricted solely to the transport industry, given the origins of modern RM practices in the airline industry, much existing RM literature does focus on transport applications. In this thesis, we focus on applications of RM by transport providers, specifically railway service providers. However, the methods and results discussed are generalisable both to other transport providers (e.g. bike-sharing as discussed in Chapter 4), and other applications of revenue management (e.g. hospitality).

Demand forecasting for RM is well-documented in the literature Pereira (2016). Many studies point out the importance of accurate demand forecasts, both from the perspective of optimising revenue and to improve planning (Banerjee et al., 2020; Weatherford and Belobaba, 2002). Very little of this literature on forecasting considers the problem of outliers even though they can have a substantial impact on the outcome of the RM. The literature also tends to focus on removing outliers from historic data to

decontaminate future forecasts, rather than detecting outliers in an online setting to improve optimisation. Weatherford and Kimes (2003) implement a simple trimming approach to remove outliers caused by atypical events, such as holidays and special conventions, to improve forecasting. Outside of the transport industry, Liang and Cao (2018) fit a Normal distribution to detect anomalous observations in hotel booking data for hospitality RM.

Detecting outliers in demand for transport systems, and quantifying their impact, is an open problem. Several issues complicate the problem further: (i) capacity restrictions mean that observed bookings do not necessarily reflect actual demand. This can cause bookings for different services to follow similar patterns, even when the underlying demand is different, complicating the process of outlier detection. (ii) Outlier demand cannot be assumed to be temporally or spatially homogeneous and uncorrelated - some areas of the transport network are more prone to outlier demand than others. (iii) The large scale of many transport networks makes investigating all possible outliers infeasible, and so any detection method is required to prioritise which outliers are the most critical for further investigation. (iv) There are multiple decisions that an analyst may take after an outlier has been identified, and it is not clear which decisions are better than others.

Overall, the significant impacts that outlier demand can have on both forecasting and optimisation, and the complicated nature of the problem which makes judgemental decisions difficult, motivates the need for a semi-automated system which highlights outliers to support analysts.

1.2 Contributions

This thesis firstly contributes a novel outlier detection method, which combines functional data analysis and time series forecasting, for use as an automated tool to detect outlier demand in transport revenue management systems. This includes an extrapolation step which allows the method to be applied in the online setting. We provide a simulation-based framework to evaluate the performance of the method in a controlled environment. Analysis of the impact on revenue from correctly accounting for such outliers is also provided.

This thesis further contributes a method for extending the application to the significantly more complex *network* revenue management setting. Our proposed two-step approach generalises to any transport setting, and we provide a case study using Deutsche Bahn booking data. We also provide a simulation study showing improved performance against various benchmarks, and the revenue benefits under different actions that may be taken in response.

The third main contribution of this thesis is an extended study of empirical data from a bike-sharing system in Washington DC, where we evidence how the previously introduced methods generalise to other transport systems besides railway networks. This section includes an extended discussion of the temporal and spatial patterns of detected outliers, and how these may be best communicated to analysts.

1.3 Outline

In Chapter 2 we introduce a novel method for identifying outlying demand in transport revenue management systems in the single-leg setting, and include an adaptation for its application in the online setting. In Chapter 3 we approach the problem of demand changes in a network setting, and demonstrate a two-stage approach (comprised of clustering and outlier detection) on railway booking data from Deutsche Bahn. In Chapter 4, we consider how such a method may be adapted for alternative transport systems and exemplify the procedure on Capital Bikeshare data. In Chapter 5, the thesis is concluded with a summary of the contributions made and suggestions of future research in the area of detecting outlier demand in transport networks.

Chapter 2

Identifying and responding to outlier demand in revenue management

Revenue management strongly relies on accurate forecasts. Thus, when extraordinary events cause outlier demand, revenue management systems need to recognise this and adapt both forecast and controls. Many passenger transport service providers, such as railways and airlines, control the sale of tickets through revenue management. State-of-the-art systems in these industries rely on analyst expertise to identify outlier demand both online (within the booking horizon) and offline (in hindsight). So far, little research focuses on automating and evaluating the detection of outlier demand in this context. To remedy this, we propose a novel approach, which detects outliers using functional data analysis in combination with time series extrapolation. We evaluate the approach in a simulation framework, which generates outliers by

varying the demand model. The results show that functional outlier detection yields better detection rates than alternative approaches for both online and offline analyses. Depending on the category of outliers, extrapolation further increases online detection performance. We also apply the procedure to a set of empirical data to demonstrate its practical implications. By evaluating the full feedback-driven system of forecast and optimisation, we generate insight on the asymmetric effects of positive and negative demand outliers. We show that identifying instances of outlier demand and adjusting the forecast in a timely fashion substantially increases revenue compared to what is earned when ignoring outliers.

2.1 Introduction

In the last 40 years, *revenue management (RM)* has become an indispensable business practice, particularly for transport service providers such as airlines and railways (Weatherford, 2016b). RM solves an optimisation problem, where firms decide on offers for perishable products, usually with the objective of maximising revenue. This optimisation assumes a fixed capacity, low marginal cost, and a given *demand forecast*. In that regard, Weatherford and Belobaba (2002) highlight that inaccurate demand forecasts can significantly diminish the achieved revenue. Banerjee et al. (2020) point out that detailed demand forecasts also support in further planning steps, such as network resource and fuel planning.

Cleophas et al. (2017) list several causes for *forecast inaccuracies*: on the one hand, the unavoidable variance of day-to-day demand prohibits perfectly accurate

forecasts. On the other hand, any flaw in the forecast model, including both the predictive time series component and the customer choice model naturally causes model-based forecast errors. Finally, sudden shifts in the market may cause short-term, temporal *outliers*. For example, when the system does not account for special events such as a sports championship or a trade fair, these will cause observed demand to systematically deviate from predictions.

We focus on such *demand outliers* in the domain of revenue management for passenger transport, specifically railways and airlines. In this domain, RM via capacity controls optimises *booking limits*, which specify the number of units that can be sold per fare class and time in a fixed *booking horizon*. Accordingly, sold units are also termed *bookings*. The distribution of bookings over intervals of the booking horizon constitutes a *booking pattern*. Booking patterns may be aggregated across fare classes and are reported either for single resources, such as flight legs, or for complementary combinations of resources, such as network itineraries. Here, we focus on aggregated booking patterns as reported for single resources, such as a single flight or a railway connection.

Common RM demand forecasting techniques estimate demand from historical booking patterns and booking limits (Weatherford, 2016b). Accordingly, we let outlier detection rely on the same data. We follow the definition by Hawkins (1980) and define an outlier as 'an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism.' Detection can either apply *online*, within the booking horizon and considering partial booking patterns, or *offline*, after a booking horizon, when the complete pattern can be analysed.

Demand outliers affect revenue management systems in two ways: (i) in *foresight*, the flawed forecast results in non-optimal capacity allocations; and (ii) in *hindsight*, the outlier can contaminate the data that underlies future forecasts. Accordingly, online detection can improve foresight, whereas offline detection can improve hindsight. To detect outliers, functional data analysis, where each booking pattern is treated as an observation of a function over time, is a natural place to turn to. Functional approaches can detect outliers in both magnitude and shape of an observed booking pattern. In other words, they can detect outliers that deviate across the entire booking horizon and those that deviate in only part of the booking horizon. Effective detection in online and offline settings has to be capable of identifying both types of outliers.

By investigating practical RM implementations in the airline and railway industry, we find that the current process relies on analysts, who manually examine booking patterns. When analysts perceive demand outliers, they attempt to compensate by adjusting the reported data, the forecast, or the booking limits. The decision of whether an adjustment is necessary and in what form depends on the analysts' intuition. As noted by Cleophas et al. (2017) and Banerjee et al. (2020), little existing work systematically measures the effect of such interventions. There is even less consideration of providing systematic analytics support for the related decisions. However, research on human decision making in general, and judgemental forecasting in particular, clearly demonstrates fallibility and bias (O'Connor et al., 1993; Lawrence et al., 2000, 2006). This motivates the need for automated alerts to highlight outliers and thereby support analysts.

To our knowledge, we are the first to propose an automated methodology for outlier detection in the RM domain. Specifically, this chapter makes the following contributions: (i) proposing a novel outlier detection approach, combining functional data analysis and time series extrapolation, which improves overall detection performance; (ii) providing a simulation-based framework for generating regular and outlier booking patterns, and evaluating their effect throughout the RM process; (iii) demonstrating the asymmetric effects of outliers on RM performance; (iv) quantifying the benefits from successful online or offline outlier detection for RM; and (v) demonstrating the use of such outlier detection in an application to empirical railway booking data.

2.2 RM forecasts and forecast evaluation

The importance of accurate forecasts as input to revenue optimisation is well-documented in the literature. Authors are largely concerned with forecasting customer demand (Pereira (2016), Weatherford and Belobaba (2002), Talluri and Van Ryzin (2004)), although forecasting cancellations and no-shows has also been explored (Morales and Wang, 2010). Weatherford and Belobaba (2002) confirm previous findings that inaccurate demand estimates can significantly impact revenue. Under the use of optimisation heuristics such as Expected Marginal Seat Revenue (EMSRb) (Belobaba, 1989), under- or over-forecasting can even be beneficial. As described by Mukhopadhyay et al. (2007), most RM systems require forecasts of the *actual* demand, rather than the *observed* demand. The actual demand consists of both observed demand and

customer requests that were denied due to restrictive booking limits. Actual demand is difficult to observe in practice, and so must be estimated. To this end, Weatherford and Belobaba (2002) survey various techniques.

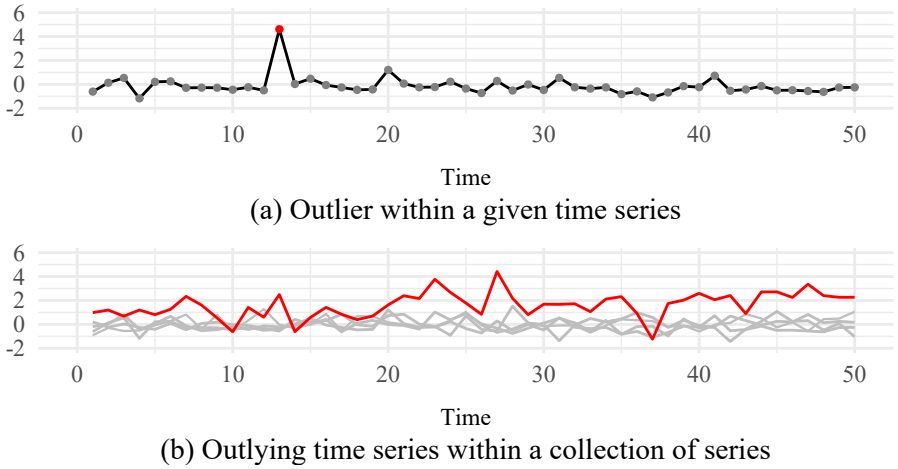
When allowing for inaccurate demand forecasts, much RM research focuses on rendering the optimisation component more robust or forecast-independent, as detailed in the contributions reviewed in Gönsch (2017). In another review, Cleophas et al. (2017) point out that there is little research into the effects of manually adjusted forecasts in RM. Mukhopadhyay et al. (2007) propose a method for measuring the performance of adjusted and unadjusted forecasts. They find that if analysts can reliably improve demand forecasts on critical flights, significantly more revenue can be generated. Zeni (2003) describe a study at US Airways, which aimed to isolate and estimate the value of analyst interactions. According to that study, around 3% of the additional revenue generated within the duration of the study could be attributed to analyst input.

Given that experiments in a live RM system carry significant risks, the use of simulation for evaluation is common. Additionally, simulation studies enable *a priori* knowledge about the true demand generation process, which can never be known in a real-world setting. Frank et al. (2008) discuss the use of simulation for RM and provide guidelines; in a related effort, Kimms and Müller-Bungart (2007) consider demand modelling for RM simulations. The chapter at hand follows these contributions in establishing a simulation-based framework to generate outlier observations. Doreswamy et al. (2015) employ simulation as a tool to analyse the effects of different RM techniques for different airlines, when switching from leg-based controls to net-

work controls. Cleophas et al. (2009) focus on an approach to evaluating the quality of RM forecasts both in terms of revenue and common forecast error measurements. Another example of using simulation to evaluate the performance of forecast components is given in Bartke et al. (2018). Temath et al. (2010) used a simulation-based approach to evaluate the robustness of a network-based revenue opportunity model when input data is flawed. In the broader context of demand forecasting, Petropoulos et al. (2014) evaluate fitting time series forecasts for particular patterns of demand evaluation by manipulating these patterns in a simulation framework.

2.3 Existing work on outlier detection

To assess the existing methodological contributions to outlier detection, we distinguish between identifying outlying observations within a time series (Figure 2.3.1a), and identifying an entire outlying time series (in our case, booking pattern) (Figure 2.3.1b). In this chapter, we aim for the latter.



(a) Outlier within a given time series

(b) Outlying time series within a collection of series

Figure 2.3.1: Different types of outliers in time series data

Literature on handling outliers in the RM process is scarce, though there is some discussion in Weatherford and Kimes (2003): the authors consider removing outliers caused by atypical events, such as holidays and special conventions, to improve future forecasting. However, they propose only to remove observations outside of the mean $\pm$ 3 standard deviations and do not seek to identify outliers online within the booking horizon.

Beyond RM, a wealth of literature studies outliers (also referred to as anomalies) in time series, as reviewed by Chandola et al. (2009) and Pimentel et al. (2014). For example, Hubert et al. (2015) survey various functional outlier detection techniques for time series data, and apply their methods to multiple real data sets. Barrow and Kourentzes (2018) consider the effect of functional outliers for call centre workload management and recommend an artificial neural network to model them as part of the forecast rather than identifying them. Talagala et al. (2019) propose a sliding window approach for detecting outlying time series within a set of (nonstationary) time series, based on the use of extreme value theory for outlier detection. The authors also distinguish identifying outliers within a time series, and identifying an outlying series from a set. The remainder of this chapter distinguishes three classes of approaches to outlier detection: (i) univariate, (ii) multivariate, or (iii) functional. Further technical details of all outlier detection methods described here are available in Appendix A.1.

Univariate approaches

Univariate outlier detection techniques identify anomalous observations of a single

variable, and so can be applied independently at different time points in a time series, e.g., to the cumulative number of bookings per interval in a booking horizon.

- **Nonparametric percentiles**: This class of approaches uses lower and upper percentiles of the observed empirical distribution at each time point as limits for what constitutes a regular observation as opposed to an outlier. This type of percentile-based approach is discussed by Pincus et al. (1995). It can be used as a basic way to estimate statistics in a more robust manner, by *trimming* or *winsorising* the data (see Dixon and Yuen (1974)). The downside of this approach is that a fixed percentage of the data will always be classified as outliers, even when there are fewer or more actual outliers in the data.

- **Tolerance intervals**: Statistical tolerance intervals contain at least a specified proportion of observations with a specific confidence level (Hahn and Chandra, 1981). They require two parameters: the coverage proportion, β, and confidence level, $1 - \alpha$. For booking patterns, at each interval of the booking horizon, these approaches define a tolerance interval for the cumulative number of bookings by that time. If the number of observed bookings lies outside of this tolerance interval, the pattern is deemed an outlier. Nonparametric tolerance intervals do not assume an underlying distribution, and instead are based on the order statistics of the data (Wilks, 1941). Parametric tolerance intervals assume an underlying distribution (Hahn and Chandra, 1981). The choice of distribution is not arbitrary, and a bad choice of distribution will perform poorly. Liang and Cao (2018) choose to fit a Normal distribution to hotel booking data to detect anomalous observations.

- **Robust Z-score**: The Z-score measures where an observation lies in relation to the mean and standard deviation of the overall data (Iglewicz and Hoaglin, 1993). The robust Z-score uses the median and the median absolute deviation to provide a similar measurement. As such, an observation with a robust Z-score above some threshold is classified as an outlier. This score-based method assumes that the observations in a given booking interval are approximately normally distributed based on two justifications: (i) a large proportion of univariate outlier detection methods rely on distributional assumptions (often normality); and (ii) although the discrete, non-negative integer nature of booking data suggests the use of a Poisson distribution, in the presence of trend or seasonal adjustments, the data may no longer have these properties.

Multivariate approaches

Univariate outlier detection approaches ignore the dependence both within and between time series. We next turn to multivariate approaches as potential methods for capturing within (but not between) time series dependence. In this setting, a time series of length τ, that is, a booking pattern observed over τ intervals, is considered as a point in a τ-dimensional space. This allows the multivariate approaches to compute the distance between any two booking patterns, but ignores the time ordering of observations.

- **Distance**: Each booking pattern (observed over τ intervals) can be characterised by its τ-dimensional distance to every other booking pattern. Aggregating these distances transforms the problem into a univariate outlier detection problem, based

on the mean distances. Depending on the length of the booking pattern, issues relating to sparsity due to high dimensionality may arise. As discussed by Aggarwal et al. (2001), some distance metrics perform better than others in a high dimensional space. However, in relation to distance metrics, high dimensionality often refers to at least hundreds of dimensions. The number of booking intervals in RM applications is often fewer than this, ranging from 20 to 50 in examples known to the authors. Therefore, we consider the classical Euclidean and Manhattan distance metrics in our comparative evaluation.

- **K-Means clustering**: K-means clustering splits the observed booking patterns into K groups by iteratively minimising the (τ-dimensional) distance between each booking pattern and the centre of its assigned cluster (see e.g. MacQueen (1967)). This approach uses a distance threshold to identify booking patterns as outliers based on their distance to the centre of their cluster (Deb and Dey, 2017). As in the distance-based approaches, the choice of distance metric is highly relevant for clustering. Once more, this chapter compares Euclidean and Manhattan distance metrics. The approach requires as its input parameter a given K to indicate the number of clusters. Information on the methodology used to determine K is available in Appendix A.1, including a comparison of performance under different choices of K, and the distribution of genuine outliers across such clusters.

Functional approaches

There are two main issues with the use of multivariate outlier detection approaches in this application: (i) the effects of high-dimensionality on distance metrics when

considering a large number of booking intervals; and (ii) the lack of accounting for the consecutive, time-ordered, nature of the observations. For such issues, functional data analysis is an intuitive place to turn. Functional data analysis addresses both issues by (i) treating booking patterns as functions observed τ times rather than points in a τ-dimensional space, and (ii) explicitly taking into account the time-ordering of the observations. We provide further analysis of the importance of time-ordering in Appendix A.3.4.

The functional analysis setting, as discussed by Febrero et al. (2008), lacks a rigorous definition of an outlier. We use the same definition as Febrero et al. (2008): 'a curve is an outlier if it has been generated by a stochastic process with a different distribution than the rest of the curves, which are assumed to be identically distributed'. We view this as a more specific version of the definition by Hawkins (1980) provided in our introduction.

A *depth* function attributes a sensible ordering to observations, such that observations near the centre should have higher depth and those far from the centre should have lower depth. In the functional data setting, this idea provides an ordering to a set of smooth functions observed over discrete time-intervals, with the most central curve trajectory having highest depth. Functional depth not only accounts for the magnitude of the observations, but also for the variability in amplitude i.e. the shape of the curve (Febrero et al., 2008). Given this definition of functional depth, the degree of abnormality of a curve can be characterised by its functional depth, if its depth is particularly low (Hubert et al., 2015). Depth-based approaches for detecting outlying curves are discussed in detail by Hubert et al. (2012). In this chapter, we

focus on the multivariate halfspace depth described by Claeskens et al. (2014). We state and explain the mathematical definition of the multivariate halfspace depth in Appendix A.1.

2.4 Proposed methodology: functional outlier detection with extrapolation

To improve foresight, RM systems need to identify demand outliers online and as early as possible in the booking horizon. This enables the RM system to update controls for the remainder of the horizon. We term this problem *online outlier detection*. When tasked with online detection at time t_τ in the booking horizon, all approaches discussed in the previous section would exclusively consider the first τ observation intervals only.

Therefore, in the online setting, only a partial booking pattern is available for analysis. We propose to supplement the outlier detection by extrapolating the expected bookings from the current time t_τ up to the end of the booking horizon, t_T. In the computational study, we compare simple exponential smoothing (SES, Chatfield (1975)), autoregressive integrated moving average models (ARIMA, Box and Jenkins (1970)), and integrated generalised autoregressive conditional heteroskedasticity models (IGARCH, Tsay (2002)). Appendix A.1.2 provides a detailed list of univariate forecasting methods that can be used for extrapolation.

Algorithm 1 outlines the procedure on a set of N booking patterns observed until time t_τ, where given an entire booking horizon of length t_T with $t_1, \ldots, t_\tau, \ldots, t_T$,

then $\boldsymbol{y}_n(t_\tau)$ is a time series describing the bookings for pattern n up to time t_τ: $\boldsymbol{y}_n(t_\tau) = \{y_n(t_1), y_n(t_2), \ldots, y_n(t_\tau)\}$.

Algorithm 1: Using extrapolation to improve functional outlier detection

At time t_τ forecast the accumulation of bookings at each time $t_{\tau+1}, \ldots, t_T$, denoted $\hat{y}_n(t_{\tau+1}), \ldots, \hat{y}_n(t_T)$, for each booking pattern n;

Calculate $\mathcal{D}_n(\hat{\boldsymbol{y}}_n(t_\tau))$, the functional depth of the observed and extrapolated booking pattern $\hat{\boldsymbol{y}}_n(t_\tau) = \{y_n(t_1), y_n(t_2), \ldots, y_n(t_\tau), \hat{y}_n(t_{\tau+1}), \ldots, \hat{y}_n(t_T)\}$, for each booking pattern n at time t_τ;

Calculate a threshold, C, for the functional depth;

    Bootstrap the original booking patterns, with probability proportional to their functional depths;

    Smooth the bootstrap samples;

    Let C^b be the 1^{st} percentile of the depths of the b^{th} bootstrapped sample;

    Set C as the median value of the C^b;

if $\mathcal{D}_n(\hat{\boldsymbol{y}}_n(t_\tau)) \leq C$ **then**

    Define booking pattern n as an outlier. Delete booking pattern n from the sample of N patterns.

end

while $\exists\ n\ s.t.\ \mathcal{D}_n(\hat{\boldsymbol{y}}_n(t_\tau)) \leq C$ **do**

    Recalculate functional depths on the new sample, and remove further outliers.

end

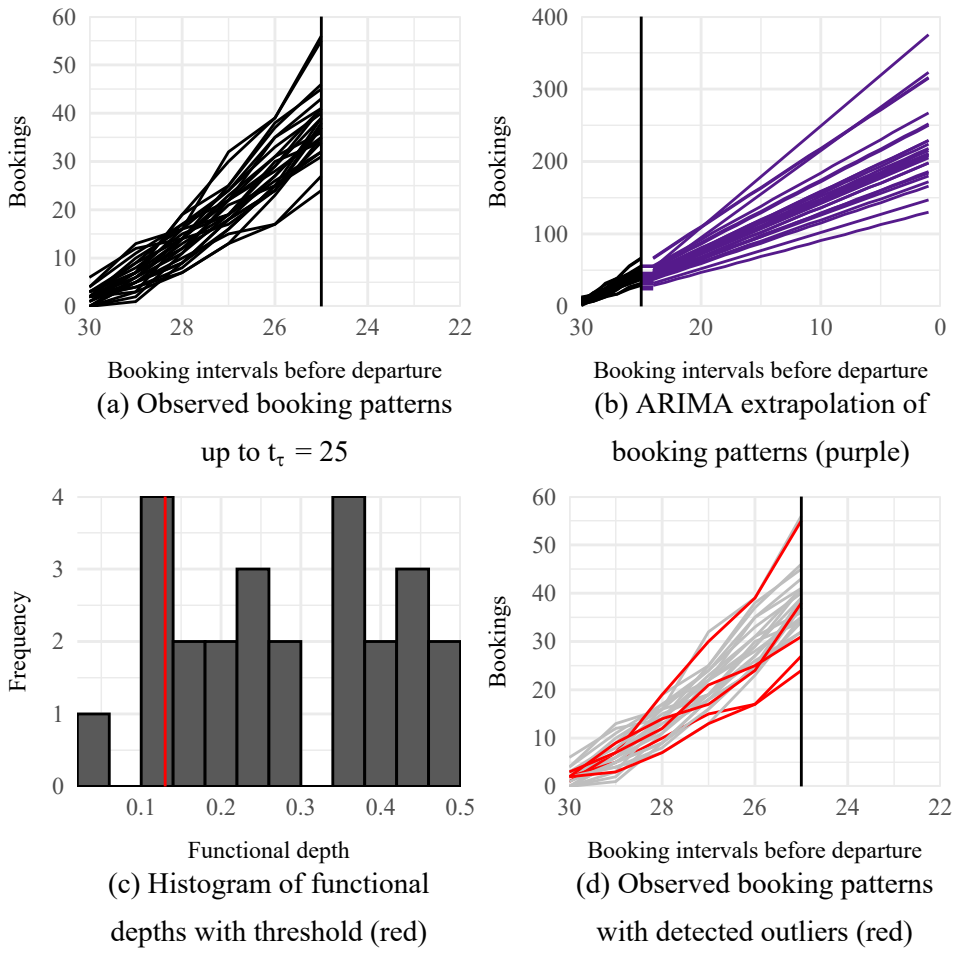


Figure 2.4.1: Example: functional halfspace depth with ARIMA extrapolation outlier detection

Figure 2.4.1 demonstrates the algorithmic approach; in the extensive simulation analysis, we apply it to a variety of booking patterns and outliers. Figure 2.4.1a shows 25 booking patterns that have been observed during the first five of thirty intervals of the booking horizon. The extrapolation step is shown in Figure 2.4.1b, where the purple lines depict the ARIMA extrapolation of accumulated bookings until the end of the horizon. The empirical distribution of the functional depths of the extrapolated sample are shown in Figure 2.4.1c, with the threshold shown in red (computed via

the bootstrapping routine described in Algorithm 1, lines 3-7). The booking patterns classified by the algorithm as outliers are highlighted in red in Figure 2.4.1d.

The input parameters relating to the calculation of the threshold includes the number of bootstrap samples (line 4), the smoothing method (line 5), and the choice of percentile (line 6). In this chapter, we select parameters as per Febrero et al. (2008) as these perform well in a wide range of settings. Further details of the threshold calculation are available in Appendix A.1. The proposed approach could alternatively feature any of the multivariate or functional approaches reviewed in Section 2.3.[1] However, a functional approach provides more scope for extensions, such as considering seasonality and increasing the frequency of outlier detection. In addition, the approach can utilise a variety of methods for extrapolating. Note that the methodology employed for this extrapolation step is independent of the forecasting methodology to predict demand for RM.

2.5 Simulation-based framework

To quantify effects from demand outliers and evaluate outlier detection approaches, we simulate a basic RM system with capacity controls. Such systems are common in the transport industry, but not limited to that domain (see Talluri and van Ryzin, 2004, Chapter 2.1). The system implemented here is minimal and general and does not fully mirror a real-world application system. However, the booking patterns our

[1]This approach is not applicable for univariate outlier detection methods as, in this setting, the number of bookings at each point in time is considered independently of past or future bookings.

simulation generates are comparable with those observed in real-world RM systems – see Appendix A.3.10. Since the simulation renders the process of demand generation to be explicit, computational experiments can yield truthful detection rates. This is impossible in empirical data analysis, where the true demand and the distinction of *regular* versus outlier demand is never fully certain. Therefore simulation modelling provides an alternative to the problem of creating reproducible forecasting research, highlighted for instance by Boylan et al. (2015).

The simulation implements the following steps:

1. Parameterise a demand model to specify both regular and outlier demand.

2. Generate multiple instances of regular and outlier demand from Step 1 in terms of customer requests (e.g. customers that intend to book a seat on a particular railway connection) arriving across the booking horizon.

3. From the demand model of regular demand (Step 2), compute the forecast in terms of the number of expected requests per fare class and time in the booking horizon.

4. Compute booking limits that maximise expected revenue from bookings based on the demand forecast from Step 3.

5. Use the booking limits (Step 4) to transform arriving requests (from Step 2) into booking patterns over the course of multiple consecutive simulated booking horizons.

6. Analyse booking patterns (from Step 5) to identify booking horizons with outlier

demand.

7. Compare knowledge of the underlying demand model (Step 2) to identified outliers (Step 6) to compute detection rates.

	Symbol	Definition	Regular Demand Value
Fixed Input	$\mathcal{I}$	The set of customer types	{1 = Business, 2 = Tourist}
	$\mathcal{J}$	The set of fare classes	{A, O, J, P, R, S, M}
	α, β	Parameters of Gamma distribution for number arrivals	$\alpha = 240, \beta = 1$
	a_i, b_i	Parameters of Beta distribution, $\acute{\lambda}_i(t)$	$a_1 = 5, b_1 = 2, a_2 = 2, b_2 = 5$
	ϕ_i	Proportion of total customer arrivals stemming from type i	$\phi_1 = 0.5, \phi_2 = 0.5$
	p_{ij}	Probability of type i being willing-to-pay at most fare class j	$p_{1j} = \{0.35, 0.1, 0.25, 0.15, 0.05, 0, 0\}$ $p_{2j} = \{0.05, 0.1, 0, 0.05, 0.1, 0.15, 0.5\}$
	r_j	Average fare for fare class j	$\{400, 300, 280, 240, 200, 185, 175\}$
	C	Capacity	200
	N_S	Number of runs of simulation used to compute forecasts $\hat{\mu}_j$ and $\hat{\sigma^2}_j$	100
Random Input	D	Total customer arrivals $\sim Gamma(\alpha, \beta)$	
	$\lambda_i(t)$	Time-dependent rate of the Poisson process of type i customer arrivals	
Output	$x_{n,i,j}(t)$	n^{th} realisation of Poisson process of type i customers purchasing fare class j at time t	
	$\hat{\mu}_j$	Forecast of mean of fare class j demand	
	$\hat{\sigma^2}_j$	Forecast of variance of fare class j demand	
	$y_{n,j}(t)$	n^{th} realisation of cumulative bookings in fare class j at time t	

Table 2.5.1: Table of notation and parameter values used for simulation

Table 2.5.1 sets out the notation used in the remainder of this section to detail the demand model, demand forecasting, revenue maximisation heuristics, and booking limits. In this section, we detail both the models and algorithms, and the parameter

settings implemented in the computational study.

2.5.1 Generating demand in terms of customer requests

Heterogeneous demand is a frequently stated RM precondition, assuming that customer segments differ in value and can be identified through their idiosyncratic booking behaviour. To model this parsimoniously, the simulation features two customer types but can be easily extended to feature more. We index any parameter that characterises high-value customers with index 1 and any parameter that characterises low-value customers with index 2. Classical RM assumes that requests from high-value customers typically arrive later in the booking horizon than those from low-value customers. High-value customers are more likely to book expensive fare classes when cheap fare classes are not offered. We follow Weatherford et al. (1993) in modelling requests from either customer type as arriving according to a non-homogeneous Poisson-Gamma process. Requests from customer type 1 arrive according to a Poisson($\lambda_1(t)$) distribution; those from customer type 2 arrive according to a Poisson($\lambda_2(t)$) distribution. The total number of customer arrivals D is split between the two segments, such that

$$\lambda_1(t)|(D=d) = d \times \phi_1 \frac{t^{a_1-1}(1-t)^{b_1-1}}{B(a_1,b_1)}, \tag{2.5.1}$$

$$\lambda_2(t)|(D=d) = d \times \phi_2 \frac{t^{a_2-1}(1-t)^{b_2-1}}{B(a_2,b_2)}, \tag{2.5.2}$$

where $D \sim \text{Gamma}(\alpha, \beta)$ with probability density function:

$$f(d|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha) d^{\alpha-1} e^{\beta d}}. \tag{2.5.3}$$

The constraint $\phi_1+\phi_2 = 1$ ensures that all requests belong to exactly one customer type. Additionally, we set parameters a_1, b_1, a_2 and b_2 such that they follow the assumption that valuable customers are more likely to request at later stages of the booking horizon:

$$\frac{a_1 - 1}{a_1 + b_1 - 2} > \frac{a_2 - 1}{a_2 + b_2 - 2}. \tag{2.5.4}$$

Figure 2.5.1a illustrates arrival rates $\lambda_1(t)$ and $\lambda_2(t)$ across the booking horizon, with Figure 2.5.1c showing one realisation of request arrivals in a specific horizon.

A set of fare classes, $1, \ldots, |\mathcal{J}|$, differentiates discount levels, $r_1 \geq r_2 \ldots \geq r_{|\mathcal{J}|}$. The simulation implements a random choice model to let customers choose from the set of currently offered classes. The model assumes all customers book the cheapest available fare class. At the same time, not all customers can afford to book any fare class. For every fare class k, the probability that a customer of type i is willing to pay *at most* fare class k is p_{ik}, as shown in Figure 2.5.1d. Each customer has a single fare class threshold, which is the most they are willing to pay, such that:

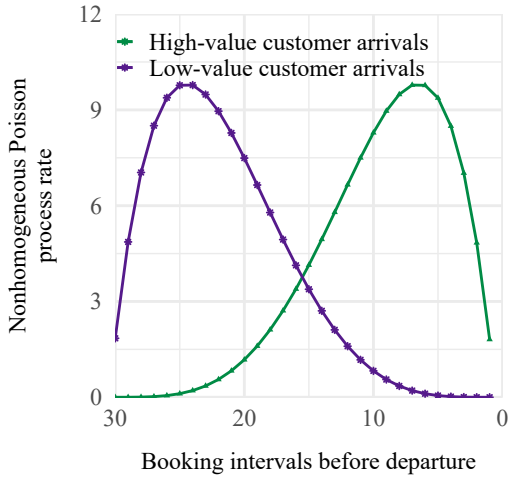
$$\sum_{k=1}^{|\mathcal{J}|} p_{ik} + p_{i0} = 1, \tag{2.5.5}$$

where p_{i0} is the the probability of a type i customer arriving and choosing not to book based on the classes on offer. Hence, the probability of booking fare class j is:

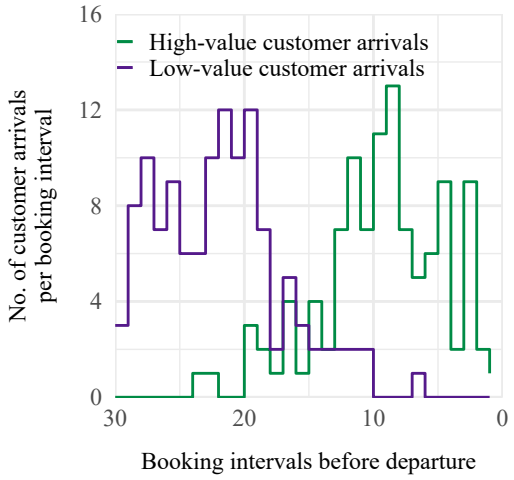
$$\mathbb{P}\left(\text{Book fare class } j | \text{No availability in classes } j+1, \ldots, |\mathcal{J}|\right) = \sum_{k=1}^{j} p_k, \tag{2.5.6}$$

$$\mathbb{P}\left(\text{Book fare class } j | \text{Availability in classes } j+1, \ldots, |\mathcal{J}|\right) = 0, \tag{2.5.7}$$

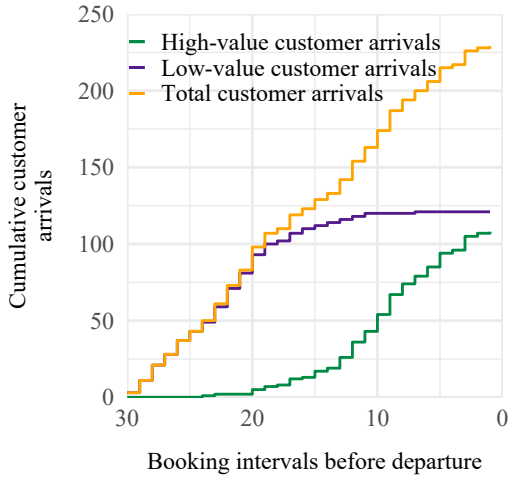
where p_k is the weighted average of probabilities of each customer type i being willing



(a) $\lambda_i(t)$, arrival rates per customer type



(b) $x_i(t)$, realisation of the Poisson process with rate $\lambda_i(t)$ per customer type



(c) $\Sigma_{k \geq t} x_i(k)$, cumulative number of customer arrivals per customer type

(d) p_{ij}, probability of a customer of type i being willing to pay up to fare class j

Figure 2.5.1: Customer arrivals generated by a nonhomogeneous Poisson-Gamma process

$D \sim Gamma(240, 1)$, $\phi_1 = \phi_2 = 0.5$, $a_1 = 5$, $b_1 = 2$, $a_2 = 2$, $b_2 = 5$

to pay up to fare class k:

$$p_k = \sum_{i \in \mathcal{I}} \phi_i p_{ik}, \tag{2.5.8}$$

and ϕ_i is the proportion of total customer arrivals stemming from type i.

While demand arrival rates vary across the booking horizon, the simulation models arrival rates and choice probabilities as stationary between booking horizons. While, in real-world markets, demand shifts in seasonal patterns and trends, we rely on random draws from distributions with stationary parameters as when introducing and detecting outliers, the simplest case lets all *regular* demand behaviour derive from the same distribution. When an approach cannot correctly detect abnormal demand when all regular demand comes from this same distribution, it is highly unlikely that it will perform better when regular demand is non-stationary.

2.5.2 Outlier generation

We generate outlier demand by parameterising demand generation in a way that deviates from the regular setting. Combining outlier demand with booking limits (optimised based on forecasts of regular demand) creates an outlier booking pattern. Outliers can result from three approaches to adjusting the parameters in Equations (2.5.1) and (2.5.2), and the probabilities p_{ij}:

1. *Demand-volume outliers*: Increasing or decreasing the volume of demand across the whole (or partial) booking horizon, by adjusting the parameters α and β in the Gamma distribution for D, the total demand.

2. *Willingness-to-pay outliers*: Shifting the proportions of demand across fare classes, by either adjusting the choice probabilities per customer type or to the ratio of customer types, ϕ_1, ϕ_2.

3. *Arrival-time outliers*: Shifting the arrival pattern of customer requests (from a

subset of customer types) over time by adjusting parameters a_1, b_1, a_2, b_2, which control the time at which requests from each customer type arrive.

2.5.3 Forecasting demand

Most RM approaches to capacity control rely on knowing the number of expected customer requests per offered product, potentially per set of offered products. The simulation implements heuristics that rely on the mean and the variance of expected requests per fare class (see Section 2.5.4).

To avoid interference from arbitrary forecasting errors, we exploit knowledge of the demand model given in the simulation setting when creating the forecast. We first draw N_S sets of customer arrivals from Equations (2.5.1) and (2.5.2). Let $x_{n,i,j}(t)$ define the n^{th} realisation of type i customers who booked in fare class j at time t as drawn from the Poisson arrival process with rate $\lambda_i(t)$ and probability p_{ij}. Then, we set the forecast to be the mean demand across all customer types upon departure from N_S simulations for fare class j, $\hat{\mu_j}$:

$$\hat{\mu_j} = \frac{1}{N_S} \sum_{n=1}^{N_S} \left(\sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}} x_{n,i,j}(t) \right). \tag{2.5.9}$$

Similarly, the simulation forecasts the variance of the demand for fare class j as:

$$\hat{\sigma_j}^2 = \frac{1}{N_S - 1} \sum_{n=1}^{N_S} \left\{ \left[\left(\sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}} x_{n,i,j}(t) \right) - \hat{\mu_j} \right]^2 \right\}. \tag{2.5.10}$$

Here we aggregate across the booking horizon in order to obtain forecasts for the final demand for each fare class. The resulting sum of customer requests per fare class across customer types gives the total expected demand per fare class. The mean

and variance of these N_S realisations are taken to be the forecasted parameters of a Normal distribution for each fare class demand.

Note that this aggregated forecast deliberately prepares the heuristic applied for revenue optimisation in this case. Applying, for instance, a dynamic program to optimally control arriving customer requests, would require a forecast of customer arrival rates and choice probabilities. The consequence of outliers, however, would be the same, as the arrival rates and choice probabilities deviate for demand outliers.

Last but not least, the simulation forecast assumes stationarity of demand, which is correct with regard to the demand setting simulated here. Therefore, a single forecast value is predicted for all future booking periods. Naturally, in a real-world setting, this stationarity is not given, but instead trends and seasonality complicate forecasting. Future research featuring such forecast aspects would open the path to further differentiation with regard to the effects of different types of outliers given different parameterisations of the non-stationary components.

2.5.4 Heuristic revenue optimisation

The simulation implements two well-known heuristic methods for obtaining booking controls for a single resource: EMSRb and EMSRb-MR. We pick these heuristics for their wide acceptance and pervasive use in practice. Furthermore, as opposed to e.g. exact dynamic programming formulations, these heuristics yield the booking limits widely implemented in current practice. We expect the nature of these booking limits and their updates to be a relevant factor for the recognition and compensation of demand outliers.

- *EMSRb*, Expected Marginal Seat Revenue-b, was introduced by Belobaba (1992). EMSRb calculates joint protection levels for all more expensive classes relative to the next cheaper fare class, based on the mean expected demand and its variance.

- *EMSRb-MR*: To make the EMSRb heuristic applicable when demand depends on the set of offered fare classes, e.g. when customers choose the cheapest available class, Fiig et al. (2010) introduce this variant. It applies a marginal revenue transformation to demand and fares before calculating the EMSRb protection levels based on transformed fares and predicted demand.

Booking limits can be implemented in either a *partitioned* or *nested* way (Brumelle and McGill (1993), and Talluri and Van Ryzin (2004), Chapter 2). Partitioned controls assign capacity such that each unit can only be sold in one specific fare class. Conversely, nested controls let assignments overlap in a hierarchical manner; i.e. units of capacity assigned to one fare class can also be sold in any more expensive fare class. Thus, nested booking limits ensure that for any offered class, all more expensive classes are also offered—as this seems an intuitive goal these booking limits are much more commonly used. Therefore, our simulations implement nested controls.

2.5.5 Evaluation of outlier detection

We regard outlier detection as a binary classification problem, where the two classes are *regular booking patterns* and *outlier booking patterns*. By definition, for any pattern generated in the simulation, we know the true class, as we know the underlying demand model.

Several indicators evaluate the performance of binary classification outcomes, as surveyed by Tharwat (2018). Each outcome falls into one of four categories: (i) if a genuine outlier is correctly classified, it is a *true positive (TP)*; (ii) if a regular observation is correctly classified, it is a *true negative (TN)*; (iii) if a regular observation is wrongly classified as an outlier, it is a *false positive (FP)*; and (iv) if a genuine outlier is wrongly classified as regular, it is a *false negative (FN)*.

To analyse results in this chapter, we implement the **Balanced Classification Rate (BCR)** as suggested by Tharwat (2018). This indicator accounts for both the average of the true positive rate and true negative rate:

$$BCR = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right). \tag{2.5.11}$$

The notions of high detection rates (fraction of genuine outliers which are correctly detected) and low false positive rates (fraction of regular observations which are incorrectly labelled as outliers) create conflicting objectives. For example, a high true positive rate does not necessarily indicate a high performing algorithm, if it is accompanied by a high false positive rate. Therefore, combining both into a single figure is useful. Nonetheless, additional results on true positive rates, false positive rates, and positive likelihood ratios (Habibzadeh and Habibzadeh, 2019) are included in Appendices C.4 and C.5. Typically, the number of outliers is outweighed by the number of normal observations. This leads to one class being significantly larger than the other. BCR is robust to this imbalance.

Additionally, we generate a **receiver operating characteristic (ROC)** curve by plotting the true positive rate against the false positive rate (McNeil and Hanley,

1984). This provides an additional diagnostic for binary classifiers. The ROC curve compares the true positive to false positive ratio as the threshold (at which an outlier is classified) varies. The optimal ROC curve is that with the combination of highest true positive rate and the the lowest false positive rate, i.e. with an area under the ROC curve closest to 1.

2.5.6 Experimental setup

We vary two main elements of the experimental setup for experimental analysis. The first is the parameter settings used to generate regular and outlier demand. The second are the settings of outlier detection.

We generate regular demand according to the parameters in Table 2.5.1, which results in regular total demand with a mean of 240, and a standard deviation of 15.492. We benchmark detection performance on outlier demand generated in various ways. Our main focus is on analysing different magnitudes of demand-volume outliers. Our choice of parameter changes for outlier generation follows Weatherford and Belobaba (2002), who investigate the effects of inaccurate demand forecasts on revenue. In particular, they consider cases where forecasts are 12.5% and 25% higher or lower than the actual demand. We perform a similar analysis on the benefits of detecting outliers where the overall number of customers deviates from regular demand by $\pm$ 12.5% and $\pm$ 25%. These four types of demand-volume outliers we consider are generated by varying the parameters α and β as described in Table 2.5.2. This results in a change in mean of the desired magnitude and direction, but no change in variance. In addition, we consider other types of outliers, as outlined in Section 2.6.3.

	Mean	**Std. Dev**	α	β
Regular Demand	240	15.492	240	1
25% Increase	300	15.492	375	1.25
12.5% Increase	270	15.492	303.75	1.125
12.5% Decrease	210	15.492	183.75	0.875
25% Decrease	180	15.492	135	0.75

Table 2.5.2: Parameter choices used to generate demand-volume outliers

In a wide-ranging computational study, we compared the performance of all outlier detection methods described in Section 2.3. Appendix A.2, Table A.2.3 lists the aggregated results from all experiments carried out. For conciseness, the results discussed in the next section focus on the *best* univariate method, *parametric (Poisson) tolerance intervals*; the best multivariate method, *K-means clustering with Euclidean distance*; the best functional method, *functional depth*; and the best extrapolation method, *ARIMA extrapolation combined with functional depth*.

The settings used for these four methods are as follows:

- *Parametric tolerance intervals*: The distribution chosen is Poisson, see Appendix A.1 for details. The coverage proportion is chosen to be $\beta = 0.95$, and the confidence level is $\alpha = 0.05$ by default.
- *K-means clustering*: The number of clusters, K, is chosen to be 2, see Appendix A.1 for reasoning. The default threshold for classifying a booking pattern as an outlier is half the sum of the maximum and minimum distances of the patterns

from their cluster centres (Deb and Dey, 2017).

- *Functional depth*: The number of bootstrap samples for the threshold is chosen to be 1000. The smoothing method is as suggested by Febrero et al. (2008). Similarly, the percentile chosen for this analysis is the 1^{st} percentile, as suggested by Febrero et al. (2008).
- *Functional depth + ARIMA extrapolation*: Thresholds are calculated as in functional depth. The orders of the ARIMA extrapolation are selected using `auto.arima` in R, based on AICc, with the augmented Dickey-Fuller test used to choose the order of differencing. The parameters are estimated using maximum likelihood with starting values chosen by conditional-sum-of-squares.

We provide further details on the extent of the computational study, including aggregated results, in Appendix A.2.

2.6 Simulation results

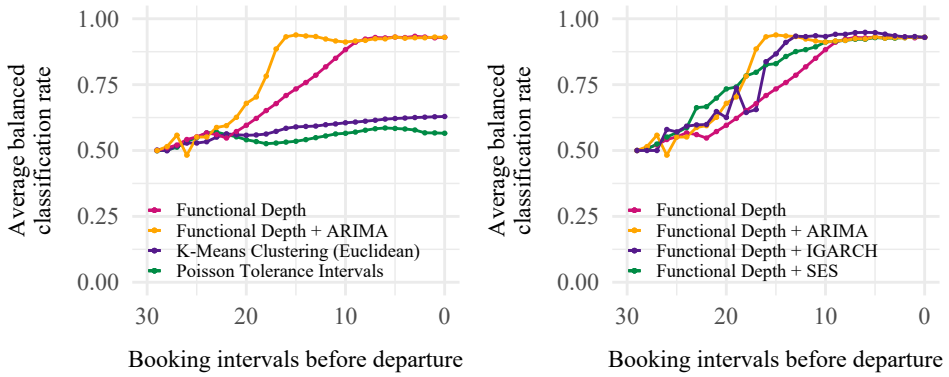
To investigate different outlier simulation and detection techniques, we follow a four-step process. We contrast foresight detection performance of different outlier detection methods in Sections 2.6.1 and 2.6.2. This analysis focuses on detection performance across the booking horizon, and evaluates the detection approaches' ability to detect outliers early in the booking horizon. We also quantify the gain in outlier detection performance resulting from the inclusion of the extrapolation step proposed in Section 2.4. Subsequently, Section 2.6.3 investigates the effect of different types of outliers on the performance of the outlier detection method. Additionally, Section 2.6.4 consid-

ers an empirical data set to demonstrate the practical implications of the approach. Finally, Section 2.6.5 presents a final set of experiments intended to measure the potential increase in revenue generated by analysts correctly taking actions based on alerts from the proposed method of outlier detection. Note that all experiments analysed in this section implement the EMSRb-MR heuristic, which is a better fit with the given demand model. We have investigated the implications of applying the EMSRb heuristic and assessed the revenue generated as well as the effect on identifying outliers in an ancillary study. The results under EMSRb (found in Appendix A.3.1) and EMSRb-MR were found to yield similar conclusions, regardless of the outlier detection method used. Additional results, including those relating to the hindsight detection of outliers, are available in Appendix A.3.

2.6.1 Benchmarking foresight detection of demand-volume outliers

To evaluate *foresight detection* performance, Figure 2.6.1 displays the average BCR per booking interval. Very early in the booking horizon, all four methods suffer from poor performance but for different reasons – some suffer from low true positive rates, others from high false positive rates (see Appendices C.4 and C.5 for details). At around 21 booking intervals before departure, the average BCR of functional methods quickly accelerate towards 1, whereas the univariate and multivariate approaches at best only show mild improvements in classification performance.

Including ARIMA extrapolation markedly accelerates classification performance,



(a) Comparison of best performing outlier detection methods

(b) Improvement from incorporating extrapolation

Figure 2.6.1: Comparison of foresight outlier detection averaged over different magnitudes of demand outliers with 5% outlier frequency

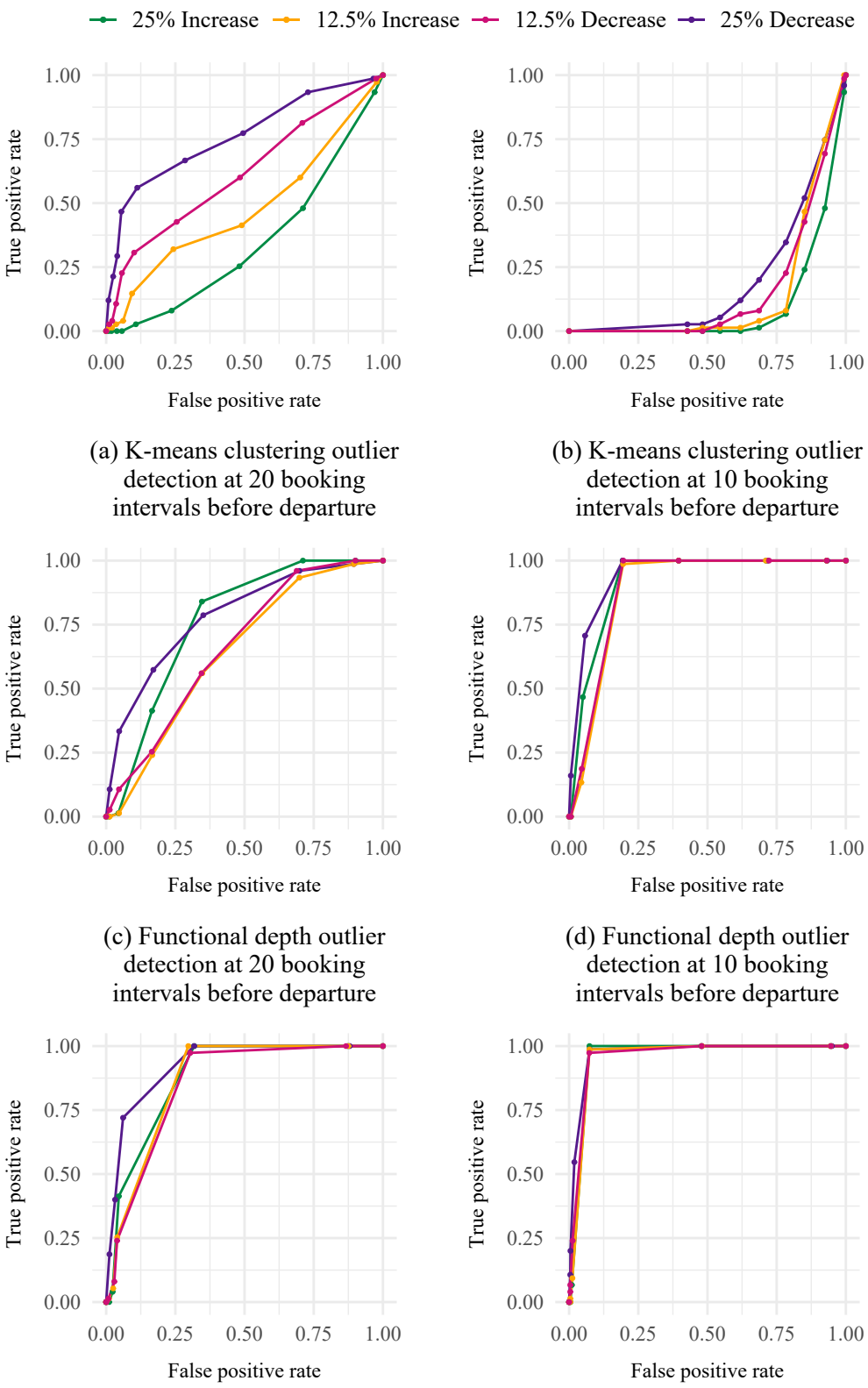
especially between 20 and 10 booking intervals before departure. Note that the aim of extrapolation is not necessarily to increase the overall BCR, but to achieve peak performance earlier in the booking horizon to gain time for analyst intervention. The extrapolation achieves this by increasing the variance of the booking patterns, leading to an increase in the number classified as outliers. See Appendix A.3.6 for further details. Additional analysis of Receiver Operating Characteristic (ROC) curves (see Section 2.6.2), further supports the inclusion of ARIMA extrapolation. In Figure 2.6.1b, we also compare functional depth with IGARCH and SES extrapolation, and similar improvements are observed as with ARIMA extrapolation. ARIMA provides overall larger gains in performance compared to SES and IGARCH. This is likely due to the flexibility of ARIMA in capturing the changing curvature of the booking pattern, and its ability to encapsulate the autocorrelations induced by censoring from the booking limits. In the last third of the booking horizon, the extrapolation makes

up a much smaller part of the pattern, i.e. most of the pattern is now made up of observed rather than extrapolated data. Hence, the input data to the outlier detection algorithm with different extrapolations is similar, and so they produce similar results. Further analysis on the relationship between extrapolation accuracy and the resulting improvement in outlier detection is available in Appendix A.3.8.

As noted in Section 2.4, extrapolation could also be combined with multivariate outlier detection methods such as K-means clustering. Given the superior performance of functional depth we focus our main results on this combination, but additional results regarding combining with multivariate techniques are presented in Appendix A.3.3.

2.6.2 Receiver Operating Characteristic (ROC) curves

To show that our conclusions in Section 2.6.1 are robust to different parameterisations of the outlier detection settings, we perform an ROC curve analysis by varying the thresholds for K-means clustering, functional depth, and functional depth with extrapolation. Figure 2.6.2 shows the results for two time intervals in the booking horizon: one early at 20 intervals before departure, and one later at 10 intervals before departure.



(a) K-means clustering outlier detection at 20 booking intervals before departure

(b) K-means clustering outlier detection at 10 booking intervals before departure

(c) Functional depth outlier detection at 20 booking intervals before departure

(d) Functional depth outlier detection at 10 booking intervals before departure

(e) Functional depth with ARIMA extrapolation outlier detection at 20 booking intervals before departure

(f) Functional depth with ARIMA extrapolation outlier detection at 10 booking intervals before departure

Figure 2.6.2: Receiver operating characteristic (ROC) curves

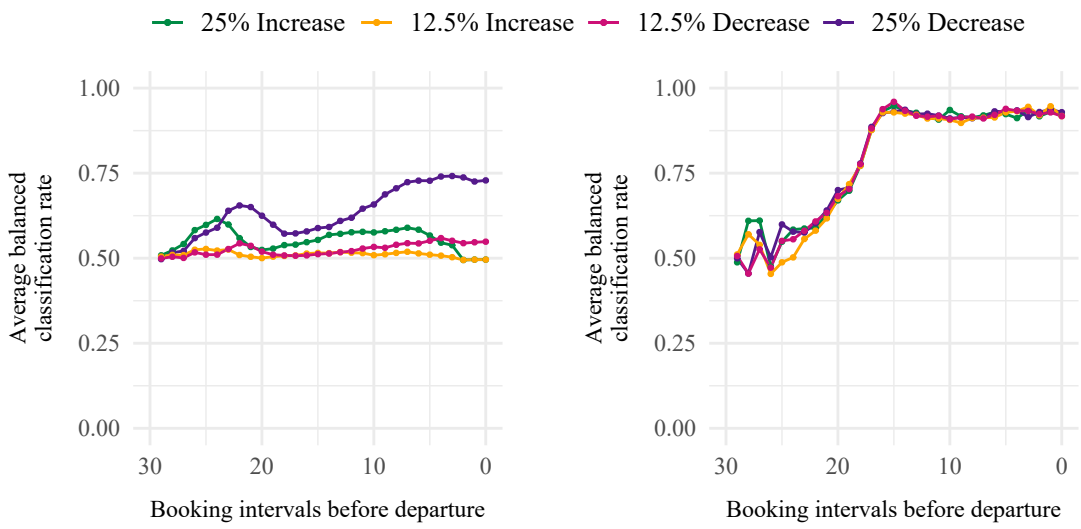
There are three main conclusions that can be drawn from the results of the ROC analysis. (i) The area under the ROC curve is consistently higher for functional approaches than for K-means. Similarly, the area under the curve is even higher when we include extrapolation. (ii) For K-means, the area under the ROC curve diminishes as the number of booking intervals increases, suggesting issues with sparsity caused by high dimensionality.

Thus, even a better choice of threshold criteria would not result in improved performance for K-means. (iii) The improvement between functional depth and functional depth with extrapolation is smaller towards the end of the booking horizon. This is due to the fact that, at this point, the extrapolation makes up a smaller part of the input data and so the two approaches are more similar.

2.6.3 Outlier detection for diverse types of outliers

We next investigate how the average BCR varies depending on the type and magnitude of outliers. All experiments in this section feature an outlier frequency of 5%. When we tested the sensitivity of approaches to different frequencies of outliers (ranging from 1% to 10%, results omitted here), we found little impact on outlier detection performance across methods, such that the conclusions drawn from this section are generally robust. Results on the effect of outlier frequency are available in Appendix A.3.2.

First, we vary the magnitude of *demand-volume outliers* to $\pm 12.5\%$ and $\pm 25\%$. Figure 2.6.3a displays the average BCR over time for parametric (Poisson) tolerance intervals. We observe that higher magnitudes of outliers are easier to classify, but also



(a) Parametric (Poisson) tolerance intervals outlier detection

(b) Functional halfspace depth with ARIMA extrapolation outlier detection

Figure 2.6.3: Balanced Classification Rate under different magnitudes of outliers with 5% outlier frequency

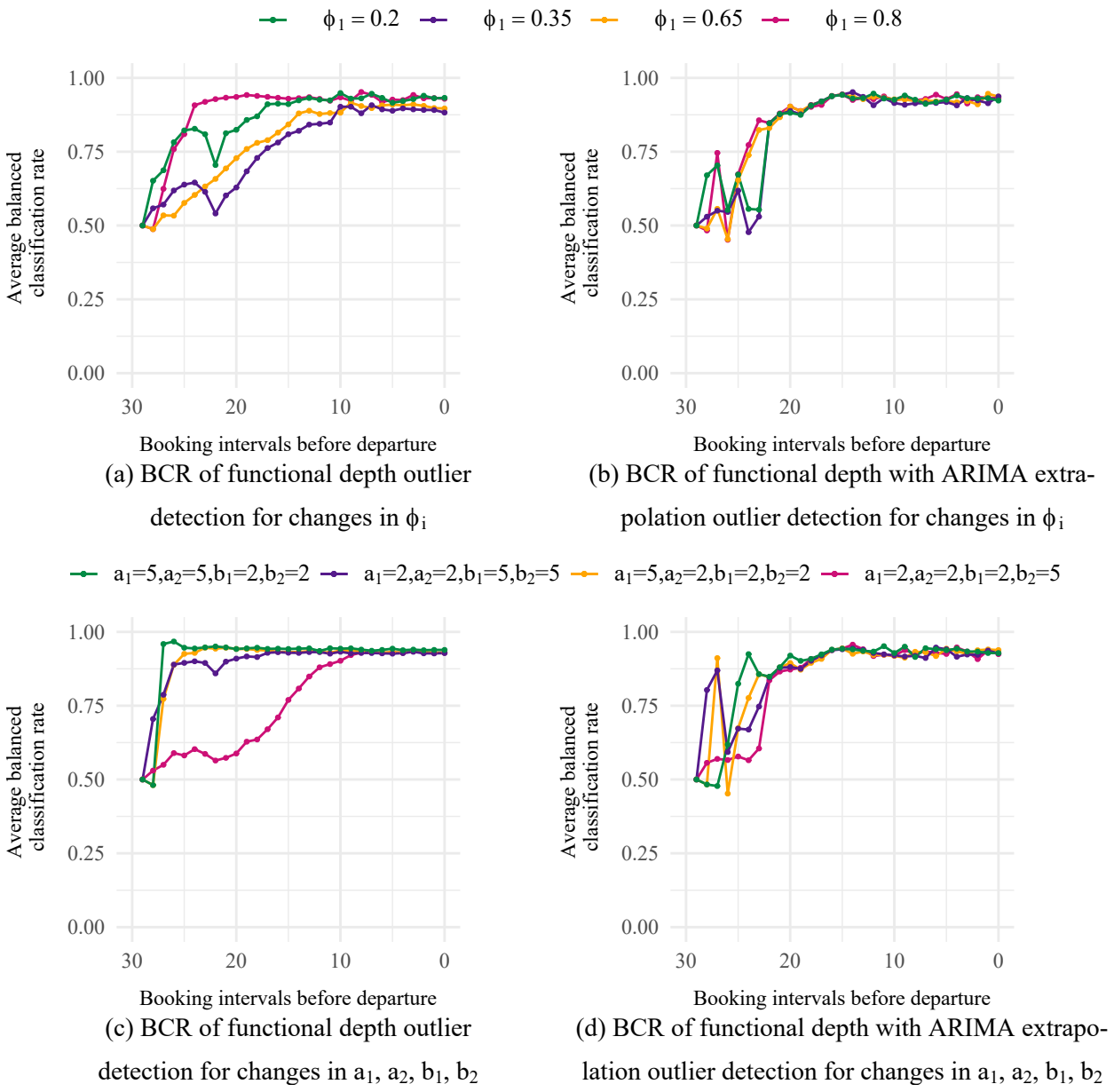
that demand decreases are easier to classify than increases. The latter observation is intrinsic to RM systems: an unexpected decrease in demand causes a decrease in bookings, but an increase in demand does not necessarily result in an increase in bookings if the booking limit for a fare class has been reached, i.e. if the fare class is no longer offered. This censoring leads to the phenomenon of observing a constrained version of demand.

Similar observations arise when testing all other univariate and multivariate outlier detection approaches. In contrast, Figure 2.6.3b displays the average BCR over time with functional halfspace depth and ARIMA extrapolation. Here the average BCR is very similar for all four magnitudes of outliers considered. This classification approach therefore appears to be very robust to the magnitude and direction of out-

liers considered. The robustness to the direction of the outlier demand shift is a result of the choice of depth measure. Hubert et al. (2012) define the multivariate functional halfspace depth for the purposes of identifying curves which are only outlying for a fraction of the time they are observed over. This means that if a booking pattern is affected by censoring, as long as it has still been an outlier before censoring came into effect, it can still be detected later in the horizon. In terms of robustness to magnitude, we hypothesise that much smaller outlier magnitudes would need to be considered before the average BCR decreases. We further consider demand shifts of $\pm 1\%$, $\pm 5\%$, $\pm 10\%$. The results are as expected - for $\pm 10\%$, the performance is only slightly poorer; for $\pm 5\%$, we see a drop in performance with the algorithm at best having a BCR of around 0.75; and a level of $\pm 1\%$ performance is particularly poor with a BCR of close to 0.5. This is behaviour we would expect, given that outliers caused by such a small deviation in demand are unlikely to be considered *outliers* in any real sense. These results are available in Appendix A.3.7.

Figures 2.6.4a and 2.6.4b illustrate effects from *willingness-to-pay outliers*, where the ratio of high-value to low-value arrivals changes. The default value in our simulations is $\phi_1 = \phi_2 = 0.5$ such that there is a 1:1 ratio, but we allow this ratio to change to create outliers. Here, $\phi_1 < 0.5$ creates a higher percentage of total arrivals from low paying, early arriving customers of type 2. Under functional depth outlier detection, it is easier to detect this type of outlier when the change in ϕ_1 is larger. There is a dip in performance around interval 22, as this large number of low-paying arrivals causes censoring when booking limits render cheaper classes unavailable.

Setting $\phi_1 > 0.5$ creates a larger percentage of type 1 customers, who arrive late



(a) BCR of functional depth outlier detection for changes in ϕ_i

(b) BCR of functional depth with ARIMA extrapolation outlier detection for changes in ϕ_i

(c) BCR of functional depth outlier detection for changes in a_1, a_2, b_1, b_2

(d) BCR of functional depth with ARIMA extrapolation outlier detection for changes in a_1, a_2, b_1, b_2

Figure 2.6.4: Performance of functional depth (with and without ARIMA extrapolation) for different types of outliers

and are willing to pay more. Again, this is easier to detect under functional depth when the change in ϕ_1 is larger. Incorporating the ARIMA extrapolation generally improves performance in the last two-thirds of the horizon. However, early in the

booking horizon this provides mixed results.

Figures 2.6.4c and 2.6.4d demonstrate the performance of functional depth (with and without extrapolation) for detecting *arrival-time outliers*. These outliers are caused by changes in the parameters a_1, a_2, b_1, b_2 (resulting in a subset of customer types arriving later or earlier than in the regular case), as outlined in Table 2.6.1.

	a_1	b_1	a_2	b_2	**Effect of parameter choices**
Regular Demand	5	2	2	5	low value customers arrive before high value customers
Setting 1	5	2	5	2	some low value customers arrive a lot later
Setting 2	2	5	2	5	some high value customers arrive a lot earlier
Setting 3	5	2	2	2	some low value customers arrive a little later
Setting 4	2	2	2	5	some high value customers arrive a little earlier

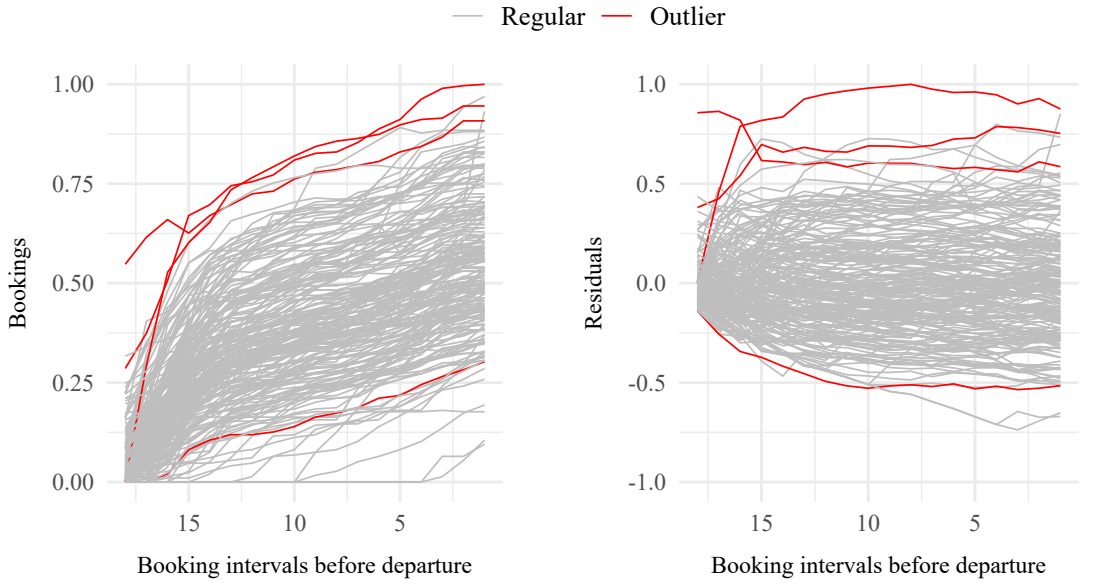
Table 2.6.1: Parameter choices used to generate arrival time outliers

Outliers in Settings 1, 2 and 3 are easy to detect even early in the booking horizon using functional depth without extrapolation. This is fairly intuitive - Settings 1 and 3 create almost no bookings early in the horizon, which is very different from regular behaviour. In contrast, Setting 2 creates far more bookings early in the horizon than the regular setting. ARIMA extrapolation is not needed nor beneficial in Settings 1-3, due to the ease of spotting outliers immediately. In contrast, outliers from Setting 4 are more difficult to detect. This is likely due to the fact that for most of the first half of the horizon, outlier booking patterns and regular booking patterns are similar. In the later half of the horizon, booking limits render the cheaper fare classes unavailable, so that arriving customers purchase higher fare classes only slightly earlier in time.

In Setting 4 extrapolation is found to significantly help the classification performance in this more challenging setting.

2.6.4 Detecting outliers in railway booking patterns

We demonstrate the proposed outlier detection method by identifying outliers in a data set of 1387 booking patterns obtained from the main German railway company, Deutsche Bahn. This preliminary empirical study can be thought of as a guide to practitioners for how to apply the algorithm. A detailed analysis of the algorithm's performance in practice would require a manually annotated data set or, potentially, a field study. While of significant interest, such an analysis is beyond the scope of this chapter.



(a) Booking patterns (outliers shown in red) (b) Residuals (outliers shown in red)

Figure 2.6.5: Pre-processing of data

We consider booking patterns that were observed for a single departure time every

day of the week, for one railway leg that directly connects an origin and a destination. The 1387 booking patterns are observed over 18 booking intervals, where the first booking interval is observed 91 days before departure. Figure 2.6.5a illustrates 148 of these booking patterns, which relate to trains departing on Mondays. For the purposes of Figure 2.6.5a, we have rescaled the number of bookings to be between 0 and 1. The booking data is generated from an RM system that implements an EMSR variant, which sets and updates booking limits based on forecasted demand and observed bookings.

In order to obtain a homogeneous data set to allow for outlier detection, we must account for two factors: (i) departure days of the week and (ii) shortened booking horizons. We compare booking patterns for different days of the week by applying pairwise functional ANOVA tests (Cuevas et al., 2004). In general, booking patterns for different days of the week are not directly comparable (see Appendix A.3.10 for details). In addition, shortened booking horizons are a special characteristic of this data set that are caused by the railway service provider's process for implementing schedule changes. As a consequence, some booking horizons are foreshortened and the majority of bookings typically arrive much closer to departure (see Appendix A.3.10).

To prepare the data for outlier detection, we transform the booking patterns to make them more comparable to each other. To account for both shortened booking horizons and departure days of the week, we apply a functional regression model (Ramsay et al., 2009). This functional regression model accounts for the way in which average booking patterns change from day to day, and fits a mean function (see Appendix A.3.10 for details) to the booking patterns for each day of the week. The

model is of the form:

$$\begin{aligned} bookings_i(t) = \beta_0(t) + \beta_1(t)I_{Monday_i} + \beta_2(t)I_{Tuesday_i} + \beta_3(t)I_{Wednesday_i} + \\ \beta_4(t)I_{Thursday_i} + \beta_5(t)I_{Friday_i} + \beta_6(t)I_{Saturday_i} + \beta_7(t)I_{Shorter\ Horizon_i} + e_i(t), \end{aligned} \tag{2.6.1}$$

where the $\beta_j(t)$ are functions of time. Here, $I_{Monday_i} = 1$ if booking pattern i relates to a departure on a Monday, 0 otherwise, and so on. Since every departure belongs to a single day of the week, $\beta_0(t)$ represents the average bookings for Sunday departures, with a non-shortened booking horizon. This means that $\beta_1(t)$ accounts for the change in average bookings between Sunday and Monday departures. The purpose of allowing the $\beta_j(t)$ to be functions of time is not to remove the trend from the booking patterns but rather to allow the relationship between different days of the week to change over the course of the booking horizon. In this model, $I_{Shorter\ Horizon_i} = 1$ if the booking horizon has been shortened due to scheduling changes (affecting departures from mid-December to mid-March), 0 otherwise.

We run the functional depth outlier detection routine on the residuals, as shown in Figure 2.6.5b, with detected outliers shown in red. We also show these corresponding outliers in red in Figure 2.6.5a. Of the 1387 booking patterns in the data set, we classify 66 ($\approx 5\%$) as outliers. Note that the frequency of outliers is not an assumption provided to the outlier detection routine, and coincides with the frequency of outliers used in the simulation setup (5%), thus justifying this choice in our earlier simulations.

For validation, we provided the labelled data set back to Deutsche Bahn. The company's domain experts have confirmed that the relative proportion of outliers is appropriate to support analyst work on improving demand forecast and booking controls. Furthermore, their hindsight analysis has confirmed that most automatically

identified outliers would have benefitted from such corrections.

In addition, we compared the dates of the booking patterns classified as outliers with a list of known holidays and events. Of the 66 booking patterns classified as outliers, 30 could be attributed to known events e.g. public holidays. This leaves 36 outlying booking patterns which would otherwise have gone undetected. However, we do not aim to solely identify already known events, as there would be little point to only confirming known information. Therefore, the additionally identified outliers are not necessarily false positives – they are in fact the very booking patterns we are attempting to identify.

2.6.5 Revenue improvement under outlier detection of demand-volume outliers

To evaluate the effect of demand deviating from the forecasts used by EMSRb and EMSRb-MR, we now introduce a best-case scenario where the RM system anticipates outliers and generates accurate demand forecasts (as opposed to implementing booking controls based on the initial erroneous forecasts). The percentage change in revenue, when switching from erroneous to correct forecasts, under four demand changes is shown in Table 2.6.2. Results show the impact of detecting and correcting outliers in demand depends on the demand factor, the choice of booking control heuristic, and the magnitude of the demand deviation.

Under EMSRb, the effect on revenue is asymmetric across positive and negative outliers. When the outlier is caused by a decrease in demand, correcting the forecast

and updating controls leads to significant increases in revenue, particularly at higher demand factors. Conversely, when the outlier is caused by an increase in demand, correcting the forecast and updating controls has a negative impact on revenue. Although counter-intuitive at first glance, this agrees with previous findings. EMSRb is known to be too conservative (Weatherford and Belobaba, 2002) and reserve too many units of capacity for high fare classes, thereby rejecting an excessive number of requests from customers with a lower willingness to pay. In consequence, there is left-over capacity at the end of the booking horizon. Hence, under-forecasting can be beneficial under EMSRb.

Optimisation Heuristic	**Forecasted Demand Factor**	**% Change in Demand from Forecast**			
		-25%	**-12.5%**	**+12.5%**	**+25%**
EMSRb	0.90	+0.1%	+0.1%	-0.9%	-3.6%
	1.20	+10.2%	+6.4%	-2.3%	-2.3%
	1.50	+12.2%	+4.4%	-4.5%	-6.8%
	Avg.	**+7.5%**	**+3.6%**	**-2.5%**	**-4.2%**
EMSRb-MR	0.90	+2.3%	+1.3%	+0.4%	+2.9%
	1.20	+2.0%	+4.1%	+4.4%	+9.9%
	1.50	+16.2%	+7.7%	+5.0%	+9.5%
	Avg.	**+6.9%**	**+4.4%**	**+3.3%**	**+7.4%**

Table 2.6.2: % Change in revenue resulting from correcting inaccurate demand forecasts

Under EMSRb-MR booking controls, the results are more symmetric across positive and negative outliers, in that correctly adjusting forecasts increases revenue regardless of whether the initial forecast was too high or too low. Under both types of heuristic, the magnitude of the change in revenue (either positive or negative) is generally larger when the change in demand from the forecast is larger.

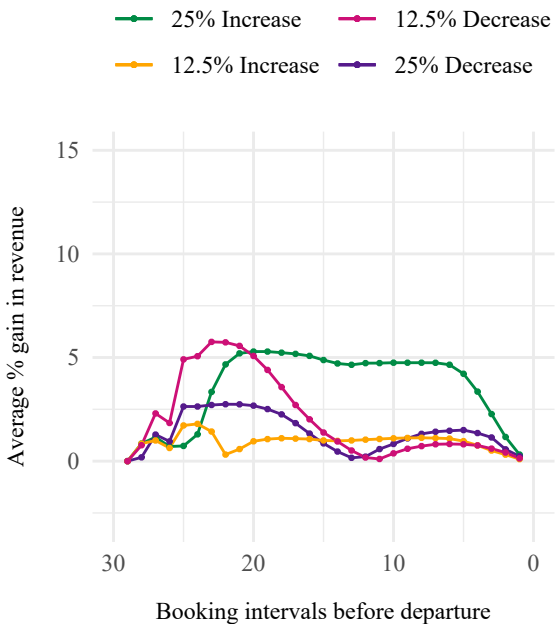


Figure 2.6.6: Gain in revenue under different magnitudes of outliers using functional depth with ARIMA extrapolation

Figure 2.6.6 shows the average percentage gain in revenue, at each point in the booking horizon, from analysts correcting forecasts for those booking patterns identified as outliers. The percentage gain is in comparison to the analyst making no changes and using the incorrect forecast for the entirety of the booking horizon.

The outlier detection method of choice in Figure 2.6.6 is functional depth with ARIMA extrapolation. We consider an idealised scenario, in that when a booking

pattern is flagged as an outlier, if it is a true positive (genuine outlier) then analysts adjust the forecast according to the correct distribution. Similarly, if the flagged outlier is a false positive, analysts do not make any changes to the forecast. Although idealised, the results here highlight the potential gains in revenue from analyst intervention, as well as the utility of using functional outlier detection in detecting true positives and avoiding false negatives (missed outliers).

Results show the use of our method creates a peak early in the booking horizon, when the potential revenue gain is highest. This peak is caused by a combination of being far enough into the booking horizon such that some bookings have occurred and the outlier detection method is able to identify outliers, but being early enough in the horizon such that any actions taken still have time to make an impact.

2.7 Conclusion and Outlook

In conclusion, the work presented in this chapter gives rise to several insights.

We benchmarked a set of outlier detection techniques and find that the functional outlier detection approach offers the best performance and the most scope for further extensions. Our results show that combining functional outlier detection with our proposed extrapolation step significantly improves performance overall, and accelerates the correct identification of outliers earlier in the booking horizon. We do note however that all methods perform poorly very early in the booking horizon where very little data has been gathered, and clearly at this stage analyst expertise or prior information is needed rather than relying on booking data alone.

By analysing an empirical railway booking data set, we demonstrated that such data is similar in shape as the data generated by the simulation model. Furthermore, the frequency of outliers detected via applying functional outlier detection to the empirical data was similar to what was observed with simulation data. In contrast to the simulation setting, the empirical data does not provide information on the labelling of actual outliers, it was therefore not possible to compute detection rates for this data. However, we validated our findings by presenting them to domain experts.

Outliers in demand diminish revenue when they go undetected. The exact effect depends on the combination of outlier and optimisation method, as shown in Section 2.6.5. Nevertheless, we argue that using a heuristic with an intrinsic bias that is then compensated by undetected outliers (as observed for EMSRb and undetected positive demand outliers) cannot be desirable for an automated system.

We have demonstrated that identifying outlier booking curves and adjusting the demand forecast accurately early in the booking horizon supports revenue optimisation. Currently, revenue management analysts decide on which booking patterns are outliers based on their previous experience of observing demand and their knowledge about special events. Automated outlier detection routines provide another procedure of alerting analysts to unusual patterns. If the detection algorithm identifies a booking pattern as an outlier, the RM system alerts the responsible analyst. When the system and the analyst agree that a booking pattern is critical and that it requires intervention, an analyst must decide which action(s) to take. Specifically, they need to decide whether to increase or decrease the forecast or inventory controls, and by

how much. Further work could investigate methods to adjust the initial forecast to account for outliers.

Within the context of RM, thoroughly examining the effects across further outlier situations, e.g. outliers affecting only part of the booking pattern, and optimisation solutions, e.g. dynamic programming, is an interesting area for further research. Furthermore, future research might consider more differentiated forecasting situations, featuring trends and seasonalities. Beyond RM, other paradigms of offer optimisation, such as mark-down pricing or the pricing of Veblen products, might offer different challenges with regards to outlier detection. Given that the resulting sales observations should also take the format of time series, we consider it interesting to find out whether the same methods would broadly apply in such different settings.

Chapter 3

Detecting outlying demand in multi-leg bookings for transportation networks

Network effects complicate demand forecasting in general and outlier detection in particular. For example, in transportation networks, sudden increases in demand for a specific destination in a network not only affect the legs arriving at that destination, but also feeder legs. Network effects are particularly relevant when transport service providers, such as railway or coach companies, offer many multi-leg itineraries. In this chapter, we present a novel method for generating automated outlier alerts to support analysts in adjusting demand forecasts accordingly. To create such alerts, we propose a two-step method for detecting outlying demand from transportation network bookings. The first step clusters network legs to appropriately partition and pool booking patterns. The second step identifies outliers within each cluster and creates a ranked

alert list of affected legs. We show that this method outperforms analyses that independently consider each leg without regard for network implications, especially in highly-connected networks where most passengers book multi-leg itineraries. A simulation study demonstrates the robustness of the approach and quantifies the potential revenue benefits from adjusting network demand forecasts for offer optimisation. We illustrate the applicability on empirical data obtained from Deutsche Bahn.

3.1 Introduction and State of the Art

Transport service providers such as railways (Yuan et al., 2018) or long-distance coach services (Augustin et al., 2014) offer a large number of interconnected legs that let passengers travel along a multitude of itineraries. Such services require providers to solve a variety of related planning problems, ranging from service network design to demand forecasting and offer optimisation across the network. Klein et al. (2020) reviews how single-leg practices to offer optimisation through revenue management (RM) generalise to the network setting. Weatherford (2016a) surveys RM forecasting methods and particularly considers itinerary-level forecasting for airlines. Further contributions, e.g. Weatherford and Belobaba (2002) and Rennie et al. (2021a), demonstrate the negative effects of inaccurate demand forecasts on revenue performance, but neglect network effects.

Little existing research, however, examines how to account for demand outliers in revenue management. For historical hotel booking data, Weatherford and Kimes (2003) discuss a simple method of removing observations that are more than $\pm 3\sigma$

away from the mean. Rennie et al. (2021a) apply functional analysis to detect outliers on the single-leg level. None of these contributions, however, consider outliers occurring in multiple legs of a network, leaving this challenge as an open problem. Furthermore, existing work has focused on binary outlier detection without regard for quantifying how critical an outlier is. This chapter simultaneously addresses both of these challenges with a novel methodological approach.

Outside the RM domain, Barrow and Kourentzes (2018) propose a functional approach for outlier detection in call arrival forecasting, also without regard for network effects. General outlier detection in networks often focuses on identifying outlying parts of the network. Fawzy et al. (2013) use this approach in wireless sensor networks to find faulty nodes. Ranshous et al. (2015) provide an overview of the extension to identifying outlying nodes in the case where the network changes over time. Most of these works on dynamic networks concentrate on analysing a single time series connected to each node, rather than analysing a set of time series, as required when booking patterns are reported for multiple departures. Hyndman et al. (2016) note that the problem of identifying unusual time series (within a collection of similar time series) is not as extensively studied as other outlier detection problems. In this chapter, we shall implement the approach suggested by Hyndman et al. (2016) to identify outlying time series based on principal component analysis (PCA), as a benchmark to our newly proposed approach.

For the remainder of this chapter, the term *departure* indicates a journey that leaves the origin station at a unique time and date. We term a unit sold as a *booking*, and the accumulation of bookings across the booking horizon as a *booking pattern*.

Booking patterns may be reported per resource (e.g. per leg), or per product (e.g. per itinerary).

Network bookings challenge outlier detection in two ways: On the one hand, demand outliers on the itinerary level affect multiple legs included in the itinerary. On the other hand, such outliers may not be recognisable given noise from other itineraries when considering the leg bookings in isolation. Identifying outliers in network revenue management data and quantifying their impact is an open problem. This chapter focuses on outliers in terms of short-term systematic changes in demand in multi-leg bookings. We argue that our proposed method, which jointly considers highly correlated legs within a network, significantly improves the performance of outlier detection.

We suggest to aggregate and analyse booking patterns from multiple *legs* as opposed to *itineraries* based on two considerations. First, when there are many possible itineraries in a large network, each individual itinerary only receives a small number of bookings on average, challenging any data analysis. Secondly, when offering a large number of potential itineraries, providers rarely store all booking patterns on an itinerary level. When applying capacity-based RM, booking patterns are stored on the leg level to ensure the availability of capacity on each leg of a requested itinerary.

Practical network RM often relies on manual forecast adjustments (Currie and Rowley, 2010; Schütze et al., 2020). However, previous research (Lawrence et al., 2006; De Baets and Harvey, 2020) has shown that the resulting judgemental forecasts can be biased and even superfluous. To avoid such collateral damages, we contribute a ranked alert list of outlying departures and affected legs, to help identify the need for further

analysis and adjustments. Perera et al. (2019) note that such forecasting support tools can improve user judgement by reducing complexity for the analyst. Analysts' time is limited and they are unlikely to have the time to investigate every departure which is flagged as an outlier. For example, Deutsche Bahn experts estimate that they can reasonably adjust less than 1% of forecasts. This motivates us to aggregate outlier analysis across multiple legs and to focus analysts' attention by constructing a ranked alert list. We consider an outlier as more critical if it indicates a larger demand shift and if it is identified across multiple legs.In contrast to the single-leg case discussed in Rennie et al. (2021a), when considering network effects there are multiple choices of forecast adjustment that an analyst can make. The best choice of forecast adjustment is not obvious, and this chapter quantifies the impact on revenue of different potential adjustments in a simulation study.

In summary, this chapter contributes (i) a method for identifying network legs that will benefit from joint outlier detection; (ii) a method to aggregate outlier detection across any number of legs to create a ranked alert list; (iii) a wide-ranging simulation study that evaluates the method's performance on various demand scenarios; (iv) simulation results that quantify the potential revenue improvement from alternative approaches to adjusting the network demand forecast to outlier demand; (v) a demonstration of applicability on empirical railway booking data from Deutsche Bahn.

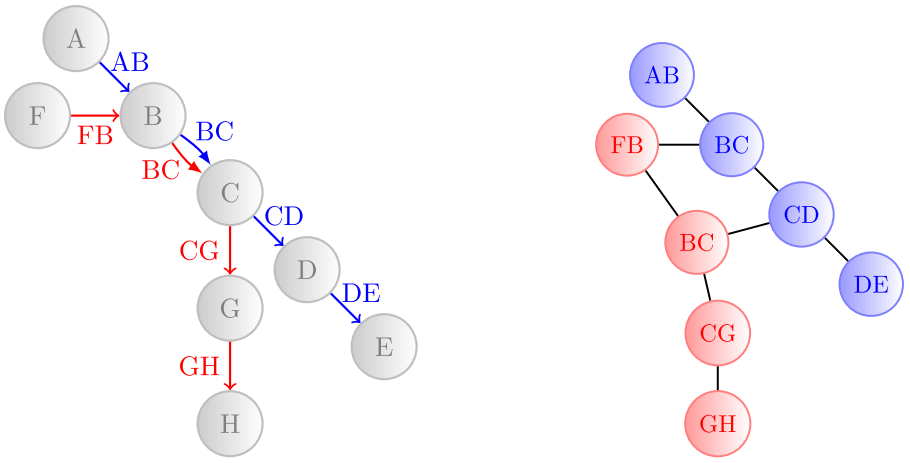
3.2 Method

In transportation networks, certain legs share common outliers, as a common set of passengers traverses them. For example, a sudden increase in demand from passengers travelling from one end of a train line to the other for an event would cause a sudden increase in demand for each of the legs in between these stations. This raises the question of which legs to consider jointly for outlier detection. Neither considering each leg independently, nor jointly considering the network as a whole, will create the best results when a network spans multiple regions that differ strongly in expected demand. Therefore, in Section 3.2.1 we propose a method to *cluster* legs such that (i) legs in the same cluster share common outliers and can be considered jointly for outlier detection, and (ii) legs in different clusters experience independent demand outliers and can be considered separately. Subsequently, in Section 3.2.2 we suggest a method for aggregating booking information within a cluster of similar legs, and then ranking identified outliers by *severity* rather than simply providing a binary classification.

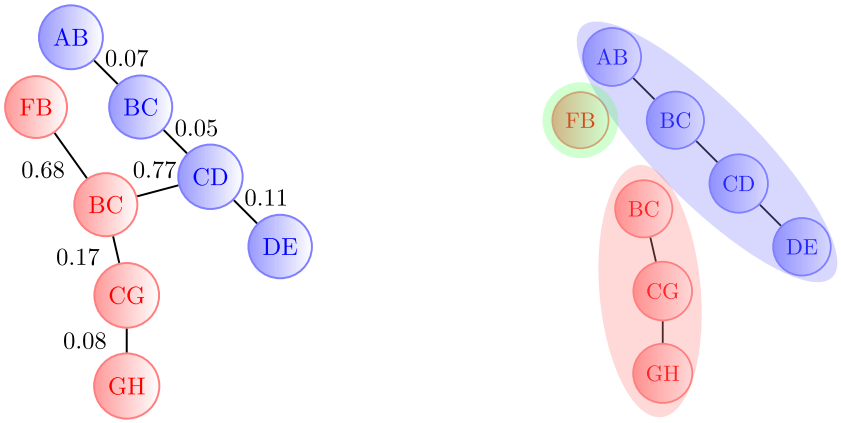
3.2.1 Correlation-based minimum spanning tree clustering

To cluster legs based on correlations in observed bookings, we first represent the network as a graph where nodes represent the stations and edges represent the legs of a journey. We shall illustrate our correlation-based clustering approach on the simple network shown in Figure 3.2.1a. In this example, two train lines (red and blue) intersect at two stations (B and C). The red train arrives at stations B and C

before the blue train, which creates two possible transfer connections for passengers: (i) switch from red to blue at B, (ii) switch from red to blue at C. Transfers from the blue to red train are not feasible.



(a) Original graph where nodes represent stations (b) Inverted graph where nodes represent legs



(c) Minimum spanning tree with edge weights (d) Clusters obtained in inverted graph

Figure 3.2.1: Correlation-based minimum spanning tree clustering

Common graph clustering algorithms seek to cluster the nodes of the graph (Schaeffer, 2007). In contrast, we wish to cluster similar legs, which correspond to edges in the original graph 3.2.1a. Hence, we invert the graph to make existing clustering

algorithms applicable. In this inversion (Figure 3.2.1b), the directed edges become nodes e.g. the edge from A to B becomes node AB. The inverted graph features an undirected edge between two nodes (two legs of the original graph) when:

- both legs are in the same train line and share a common station, e.g., legs CD and DE are connected through station D.
- the legs are in different train lines but share a common transfer station where a connection is possible, e.g., leg FB (red line) and BC (blue line) are connected through station B. However, AB (blue line) and BC (red line) would not be connected by an edge as no connection can be made between them (as we have assumed the red train arrives at B and C before the blue train).

In theory, this transformation could also include an edge between legs that share a common entry or exit node e.g. FB (red line) and AB (blue line), or CG (red line) and CD (blue line). However, in clustering both empirical and simulation data, we found that correlations between these types of legs were not sufficiently high to impact the outcome. Further, such pairs of legs would never occur in the same itinerary, such that no itinerary forecast adjustment would apply to both legs.

The algorithm aims to assign legs that experience *similar bookings* to the same cluster and legs that experience *dissimilar bookings* to separate clusters. A corresponding metric only needs to consider similarity between adjacent legs, which share a connecting station, since edges do not exist in the inverted graph otherwise. We propose to quantify this similarity via the correlation between booking patterns on the legs. To that end, we compute the functional dynamical correlation (Dubin and

Müller, 2005) – see Appendix B.1.1 for details. Unlike more common statistical correlation measures, such as Pearson correlation, functional dynamical correlation does not assume a specific type of relationship between variables (e.g. linearity). It also accounts for the time dependency between observations in the booking horizon, including the differing length of intervals between observations. For example, in the empirical data described in Section 3.4, the time between booking intervals decreases as the departure date approaches. Further, alternative measures for calculating correlations from functional data (such as functional canonical correlation) often make restrictive assumptions, which real data does not fulfil (He et al., 2003). In Appendix B.3.1, we benchmark the clustering algorithm under different correlation measures.

To represent the relationship between legs in the network (the nodes in the inverted graph), we attach weights to the edges in the inverted graph. These weights are interpreted as distances: the higher the edge weight, the further apart i.e. more dissimilar, the connected nodes are. Therefore, an applicable weight function should be non-negative. Further, the weight function needs to ensure that any negatively correlated legs are marked as more dissimilar. Even though negative correlation may imply that outlier demand jointly affects both legs, we expect it to affect negatively correlated legs in different ways. Therefore, the two legs would require different forecast adjustments, and so should be in different clusters. To satisfy these requirements we shall define the edge weights as:

$$w_{(ij,jk)} = 1 - \rho(ij, jk), \tag{3.2.1}$$

where $\rho(ij, jk)$ is the correlation between bookings on legs ij and jk.

We use a minimum spanning tree-based algorithm to allow for clusters of irregular shapes. For example, in Figure 3.2.1b, a cluster may include AB and DE because they are in the same line, rather than clustering AB and FB. Minimum spanning tree approaches work well for clusters with irregular boundaries (Zahn, 1971). Alternative approaches (such as k-means), often assume a specific shape of clusters (spherical, for k-means).

A *spanning tree* of a graph is a subgraph that includes all vertices in the original graph and a minimum number of edges, such that the spanning tree is connected. Then, the minimum spanning tree (MST) is the spanning tree with the minimum summed edge weights – see Figure 3.2.1c. Since the inverted graph is weighted, we use Prim's algorithm (Prim, 1957) to calculate the MST – see Appendix B.1.2 for a detailed introduction. Any one-to-one transformation of the weight function, $w_{(ij,jk)}$ will produce an identical minimum spanning tree.

There are two approaches to obtaining clusters from a minimum spanning tree: (i) pre-defining the number of clusters as k, and removing the $k-1$ edges with highest weight; or (ii) setting a threshold for the edge weights and remove all edges with weights above some threshold, creating an emergent number of clusters. We implement the threshold-based approach, as this ensures that each cluster has the same minimum level of correlation. In contrast, setting the number of clusters in advance could result in very heterogeneous levels of correlation across clusters. Further, setting k too low may result in legs with dissimilar features being grouped together. We apply a threshold correlation of 0.5 – the level at which legs are more correlated than they are not. This corresponds to a transformed edge weight of 0.5. In the example

given in Figure 3.2.1c, this means removing all legs with a weight above 0.5, resulting in the three clusters shown in Figure 3.2.1d.

Note that the outlier detection procedure applies to individual clusters, but does not require a particular clustering approach. Hence, alternative approaches, as reviewed in Schaeffer (2007), could be utilised.

3.2.2 Detecting outliers in clusters of legs

We now detail the process of identifying demand outliers within each of the clusters (as defined in Section 3.2.1) and then quantifying the severity of such outliers. This procedure returns a *ranked alert list* of departures.

To identify which departures should be included in the alert list, we consider the *functional depth* of their booking patterns. The approach presented here can also be implemented with other measures of exceedance, including univariate "threshold" approaches which look at aggregated bookings and ignore the shape of the booking curve. Here, we use functional depth as previous work has found this to be the most effective as an outlier detection mechanism (Rennie et al., 2021a).

Consider N departures, observed over L legs. Let $\boldsymbol{y}_{nl} = (y_{nl}(t_1), \ldots, y_{nl}(t_T))$ be the booking pattern for the n^{th} departure on leg l, observed over T booking intervals $t_1, \ldots, t_T$. Let $\mathcal{Y}_l$ be the set of N booking patterns for leg l. For each leg and departure, we calculate the functional depth (Hubert et al., 2012), with respect to the booking patterns for that leg – see Appendix B.1.3.

For each leg l, we calculate a threshold for the functional depth using the approach of Febrero et al. (2008). This method (i) resamples the booking patterns with

probability proportional to their functional depths (such that any outlying patterns are less likely to be resampled), (ii) smooths the resampled patterns, and (iii) sets the threshold C_l as the median of the 1^{st} percentiles of the functional depths of the resampled patterns. In this chapter, the default setting is to use the 1^{st} percentile of the depths as the threshold, as ths has been found to work well in practice (Febrero et al., 2008; Rennie et al., 2021a). Alternative choices of threshold are explored in Appendix B.3.2. Booking patterns with a functional depth below the threshold C_l are classed as outliers.

Furthering the approach of Rennie et al. (2021a), we now propose a method for creating ranked alert lists. First, we define z_{nl} to be the normalised difference between the functional depth and the threshold:

$$z_{nl} = \frac{C_l - d_{nl}}{C_l}. \tag{3.2.2}$$

This transforms the depth measure d_{nl} into a measure of *threshold exceedance*. Values of z_{nl} greater than zero relate to booking patterns classified as outliers. Normalising by the threshold, C_l, ensures the values of z_{nl} are comparable between different legs.

Next we define the sums of threshold exceedances across legs:

$$z_n = \sum_{l=1}^{L} z_{nl} \mathbb{1}_{\{z_{nl}>0\}}. \tag{3.2.3}$$

We sum only those values of z_{nl} that are greater than zero, to avoid outliers being masked when they occur only in a subset of legs. This sum implicitly accounts for both the size of an outlier – larger outliers further exceeding the threshold, resulting in larger values of z_{nl} – and for the number of legs in which a departure is classified as an outlier (by summing a larger number of non-zero values). To provide an example,

Figure 3.2.2 shows those values of z_n that exceed zero for a four leg section of the Deutsche Bahn network (to be discussed further in Section 3.4.2). These values of z_n correspond to departures where the booking pattern for *at least* one leg is identified as an outlier, whereas all other departures have no detected outliers in any leg such that $z_n = 0$.

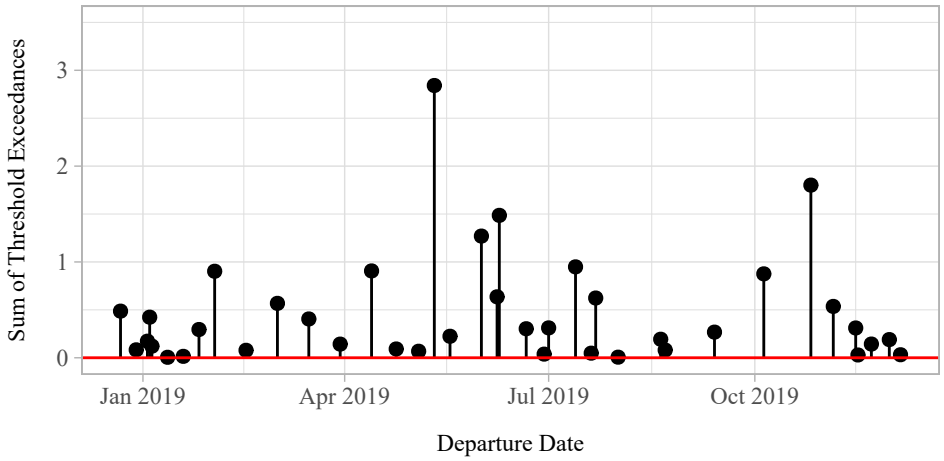


Figure 3.2.2: z_n as defined in equation (3.2.3) for a four leg section of the Deutsche Bahn network

To create a ranked list of outlier departures, i.e. those with a non-zero sum of threshold exceedances, we assign a severity, θ_n. A higher value of θ_n indicates the departure is more likely to be affected by extreme outlier demand, and hence should be targeted first by RM analysts.

To model the threshold exceedances, we turn to extreme value theory (EVT) – a branch of statistics that deals with modelling rare events i.e. those that occur in the tails of the distribution. There are two common approaches to EVT: (i) block maxima, which examines the maximum value in evenly-spaced blocks of time e.g.

annual maxima, and (ii) peaks over threshold, which examines all observations that exceed some threshold (Leadbetter, 1991). The generalised Pareto distribution (GPD) is commonly used to model the tails of distributions in the peaks over threshold approach (Pickands, 1975). Motivated by this, we fit a generalised Pareto distribution (GPD) to the sum of threshold exceedances given in equation (3.2.3). The GPD has three parameters with probability density function:

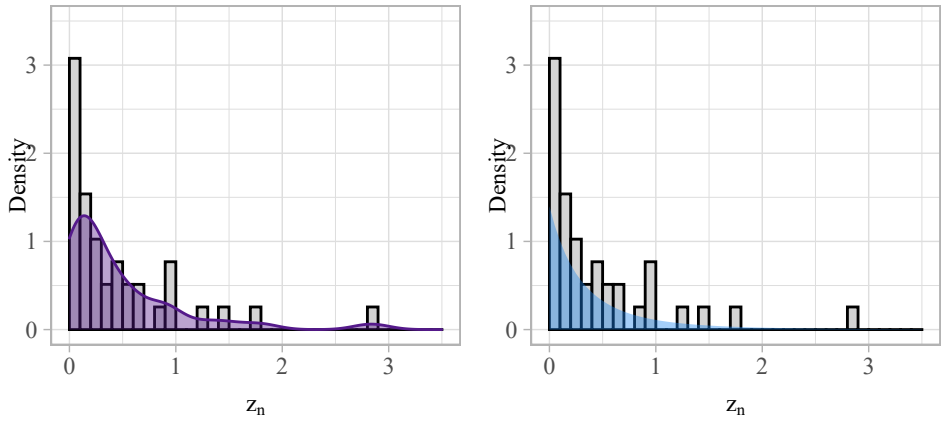
$$f(x|\mu,\sigma,\xi) = \frac{1}{\sigma}\left(1 + \frac{\xi(x-\mu)}{\sigma}^{\left(-\frac{1}{\xi}-1\right)}\right), \tag{3.2.4}$$

for

$$x \in \begin{cases} [\mu, \infty) & \xi \geq 0 \\ [\mu, \mu - \frac{\sigma}{\xi}] & \xi < 0. \end{cases} \tag{3.2.5}$$

Here, μ specifies the location, σ the scale, and ξ the shape of the distribution. We fit the parameters using maximum likelihood estimation (Grimshaw, 1993), using the R package `POT` (Ribatet and Dutang, 2019). A kernel density estimate of the empirical distribution of $z_n > 0$ from Figure 3.2.2 is shown in Figure 3.2.3a. The resulting fitted GPD is shown in Figure 3.2.3b. The GPD fit appears to be reasonable compared to the empirical distribution; further analysis in Appendix B.4.4 supports this.

Two common issues arise in fitting GPDs: (i) the choice of threshold and (ii) the independence of the data points. When the threshold is too low, the assumption of a GPD no longer holds; when it too high, there are too few data points to fit. We select a threshold of 0, i.e., we fit the GPD to values of $z_n > 0$. Rather than change the threshold at GPD level, we control the number of observations the GPD is fitted to by varying the percentile used for the individual leg thresholds, C_l. We choose



(a) Empirical distribution of z_n (b) GPD fitted to z_n

Figure 3.2.3: Distribution of z_n values from Figure 3.2.2

C_l as suggested by Febrero et al. (2008), and found that this choice worked well and provided sufficient outlying points to fit a GPD both in simulated and empirical data. To account for the second issue, applications of extreme value theory frequently first *decluster* the peaks over the threshold to ensure independence between observations (Fawcett and Walshaw, 2007). To that end, the analysis may only consider the maximum of two peaks that occur within some small time window. For mobility departures, it is theoretically possible that observed outliers may be dependent; e.g., increased demand caused by Easter not only affects Easter Sunday but also the surrounding days. However, it also very possible that the outliers are generated by independent events. As we aim to identify outlying departures rather than the underlying events themselves, this argument causes us not to decluster here.

We define θ_n as the non-exceedance probability given by the CDF of the GPD:

$$\theta_n = F_{(\mu,\sigma,\xi)}(z_n) = \begin{cases} 1 - \left(1 + \frac{\xi(z_n-\mu)}{\sigma}\right)^{-\frac{1}{\xi}} & \xi \neq 0 \\ 1 - \exp\left(-\frac{(z_n-\mu)}{\sigma}\right) & \xi = 0 \end{cases} \tag{3.2.6}$$

Formally, θ_n is the probability that, given an outlier occurs, the sum of threshold exceedances is at least as large at z_n. Thus, it is *not* the probability that a departure is an outlier. However, we use this non-exceedance probability as a measure of outlier *severity* on a scale of 0 to 1.

Departures with functional depths that do not fall below the threshold on any legs are given a severity of zero i.e. they are classified as regular departures. It is conceivable to estimate the uncertainty of θ_n (Smith, 1985) to determine further levels of criticality e.g. if there are departures with the same outlier severity, the one with smaller uncertainty would be targeted first. However, given the continuous nature of the data, it is unlikely that two departures will have an identical severity. Hence, we leave uncertainty estimation to future research.

From the severity defined in equation (3.2.6), we construct a ranked **alert list** containing all departures with a non-zero outlier severity. Although the functional depth could be directly used to construct the ranked alert list, the severity provides a measure of the difference between each rank and is more easily interpreted by analysts. The top 8 ranked outliers relating to Figure 3.2.2, are shown in Table 3.2.1.

In practice, RM analysts' time and resources only allow them to examine and adjust controls or forecasts for a limited number of suspicious booking patterns. Further, those departures which (i) exceed the functional depth threshold in only one leg or (ii) exceed the threshold only to a small degree have lower but strictly non-zero severity. These outliers are most likely to be false positives and potentially waste analysts' time. Hence, we suggest limiting the length of the list used in practice.

There are two approaches to shortening the length of the alert list: (i) only in-

Ranking	**Departure**	**Severity**	**Legs with $z_{nl} > 0$**
1	11/05/2019	0.985	AB, BC, CD, DE
2	26/10/2019	0.960	AB, BC, CD, DE
3	09/06/2019	0.942	AB, BC, CD, DE
4	01/06/2019	0.922	AB, BC, CD, DE
5	13/07/2019	0.874	AB, BC, CD, DE
6	13/04/2019	0.865	CD, DE
7	02/02/2019	0.864	CD, DE
8	05/10/2019	0.857	AB, BC, CD, DE
⋮	⋮	⋮	⋮

Table 3.2.1: Ranked alert list for cluster $= \{AB, BC, CD, DE\}$

cluding departures in the alert list if their severity is above some threshold, or (ii) setting a maximum length. Since we wish to control the number of alerts an analyst will receive, we shall analyse outlier detection performance with respect to the maximum length of the alert list. Recall that we classify departures as outliers if and only if their outlier severity exceeds zero. Therefore, if the required length of the alert list exceeds the number of identified outliers, we do not include further departures. Appendix B.3.2 presents further results on the performance of the outlier detection when varying the outlier severity threshold.

3.3 Computational Study

We implement a simulation study to evaluate the performance of outlier detection across a cluster of legs and stations. By varying demand for itineraries, we create outliers that are observable on both the leg and network level. As outliers are deliberately generated, we can evaluate detection quality on either level. By simulating network demand and offer optimisation, we further evaluate revenue implications of adjusting the forecast to account for outlier demand.

In this section, we first outline the simulation model, the choices of parameter values, and the setup of the computational experiments in Sections 3.1–3.5. Subsequently, we document and analyse the experimental results in Sections 3.6–3.7.

The simulation models a network consisting of 5 stations and 4 legs, mirroring the structure of the Deutsche Bahn network studied later in Section 3.4 in Figure 3.4.4. There are 10 possible itineraries represented by set $\mathcal{O} = \{AB, AC, AD, AE, BC, BD, BE, CD, CE, I$ On each itinerary, we assume the firm offers seven fare classes. We consider differentiated demand from two customer types represented by the set $\mathcal{I} = \{1, 2\}$.

3.3.1 Network revenue management system

The simulated RM system controls the offered set of fare classes per itinerary. To that end, it implements a dynamic program to compute bid prices per leg and sums them up per itinerary – compare Strauss et al. (2018) and Appendix B.2.1 for technical details. The bid price describes the marginal difference between the value of selling a seat in the current time period and that of reserving it to sell in a future time period.

The RM system only offers fare classes where the revenue from a booking exceeds the bid price. Bid prices depend on time until departure, unsold capacity, and expected demand. Booking patterns result from combining customer requests with the set of offered fare classes to generate bookings. Booking patterns are not reported for each individual itinerary, but only on the leg level.

Parameterising the dynamic program that computes bid prices requires predicting the expected demand arrival rates per leg l, fare class j, and time slice t of the booking horizon. Given that we know the underlying demand model for each itinerary, we can estimate the arrival rates for each leg l and fare class j by:

$$\hat{\Lambda}_{j,l}(t) = \sum_{o \in \mathcal{O}_l} \sum_{i \in \mathcal{I}} p_{i,j,o} \, \lambda_{i,o}(t), \tag{3.3.1}$$

where $\lambda_{i,o}(t)$ is the arrival rate of customers of type i requesting itinerary o, and $\mathcal{O}_l$ is the set of itineraries which include leg l. This creates an artificially accurate demand forecast. Deriving the demand forecast from the actual demand parameter values ensures that the estimation of revenue loss caused by undetected outliers is not affected by flawed forecasts (see Section 3.3.7). In practice, demand parameter values are not known but are estimated based on previously observed demand and time series forecasting.

3.3.2 Demand settings

The simulation generates booking requests per customer type i according to a non-homogeneous Poisson process, where the arrival rate per itinerary o, $\lambda_{i,o}(t)$, at time

t, is given by:

$$\lambda_{i,o}(t)|(D_o = d_o) = d_o \times \phi_{io} \frac{t^{a_{io}-1}(1-t)^{b_{io}-1}}{B(a_{io}, b_{io})}. \tag{3.3.2}$$

Here, ϕ_{io} is the fraction of customers of type i and $D_o \sim \text{Gamma}(\alpha_o, \beta_o)$ with probability density function:

$$f(d_o|\alpha_o, \beta_o) = \frac{\beta_o^{\alpha_o}}{\Gamma(\alpha_o) d^{\alpha_o - 1} e^{\beta_o d}}. \tag{3.3.3}$$

We generate demand over a horizon of 3,600 time slices to ensure $\lambda_{io}(t) < 1$. This level of detail is required to accurately parameterise the dynamic program for bid price control. The resulting bookings are aggregated into 18 booking intervals.

Next, we define p_{ijo} as the probability that a customer of type i pays up to fare class j on itinerary o. We assume that customers book the cheapest available fare class. Combining this demand model with the four-leg-network creates 210 demand parameters. We set the parameters to mirror common RM assumptions (Weatherford and Bodily, 1992): (i) valuable customers from type 1 book later than customers from type 2, (ii) customers book earlier for longer journeys, and (iii) customers are willing to pay a higher fare class if they are travelling further. The majority of passengers book tickets boarding at A and leaving at E; this ensures the correlation between the legs exceeds 0.5 and guarantees that the legs are correctly modelled in the same cluster. As detailed in Appendix B.4.6, we validated that the functional dynamical correlation between the four legs for simulated data is comparable to the Deutsche Bahn data. Appendix B.4.6 also compares the simulated and empirical booking patterns to validate parameter choices.

We generate all regular demand as described above. The full list of parameter

values can be found in Appendix B.2.3, Table B.2.1. The simulation excludes trend and seasonality to evaluate outlier detection approaches in a best-case-scenario. In other words, if an algorithm fails on observations from stationary demand, it will likely not perform better given more demand variability.

3.3.3 Outlier generation and evaluation

We focus on demand volume outliers, which we generate by changing the parameters of the Gamma distribution which governs the level of total demand (see equations (3.3.2) and (3.3.3)). Previous work found the proportion of outliers had little effect on outlier detection performance in the single-leg case (Rennie et al., 2021a). Therefore, we generate booking patterns for 500 departures, with 1% of departures affected by outlier demand. That is, we generate 495 departures from the regular demand distribution, and 5 outliers from a set of twelve outlier distributions where the mean has shifted by $\pm 10\%$, $\pm 20\%$, $\pm 30\%$, $\pm 40\%$, $\pm 50\%$, and $\pm 60\%$. For every shift in mean, we reduce the variance of the outlier demand distribution by 80%. This still results in an overall increase in variance of total demand in the presence of outliers, but also ensures that we sample sufficiently outlying demand values.

We differentiate outlier scenarios in terms of the affected network components. Firstly, we evaluate a scenario where outlier demand affects all network itineraries. We consider the case where each outlier is randomly drawn from one of the twelve outlier distributions, resulting in outliers from a mixture of different distributions. This lets us test whether the ranking of the alert list mirrors the outliers' underlying degree of demand deviation. We then considers each of the twelve outlier distributions in

isolation to assess the sensitivity of detection. Secondly, we evaluate a scenario where outliers only affect a single itinerary. This evaluates the benefits of clustering multiple legs. Appendix B.3.2 considers the practically relevant case of outliers affecting a subset of itineraries. The full extent of simulation experiments is shown in Appendix B.2.3.

Each combination of outcomes can be classified into one of four categories: (i) assigning a non-zero outlier severity to a genuine outlier creates a true positive (TP); (ii) assigning a zero outlier severity to a regular observation creates a true negative (TN); (iii) assigning a non-zero outlier severity to a regular observation creates a false positive (FP); (iv) assigning a zero outlier severity to a genuine outlier creates a false negative (FN). This classification enables us to compute the true positive rate (TPR) for the top R ranked departures in the alert list:

$$TPR_R = \frac{TP_R}{TP + FN}, \tag{3.3.4}$$

where TP_R is the number of true positives in the top R departures. The true positive rate lies between 0 and 1, where 1 means all genuine outliers were identified. We evaluate performance across 1,000 stochastic simulations.

In an ideal setting, the alert list should feature, from top to bottom, large outliers and subsequently smaller outliers. Therefore, we also use the distribution of outliers within the ranked alert list to evaluate how well the method ranks the most critical outliers.

3.3.4 Benchmarked outlier detection approaches

We benchmark the newly proposed approach to two alternatives from the literature: Principal Component Analysis combined with High Density Regions (PCA + HDR) as inspired by Hyndman et al. (2016), and the leg-based functional depth analysis as proposed in Rennie et al. (2021a).

PCA + HDR: This benchmark approach (i) computes features (e.g. mean, variance, curvature) of the booking patterns for the total demand in a cluster; (ii) uses PCA (Yang and Shahabi, 2004) to identify the first two principle components from the features; (iii) uses HDR, a density based approach (Hyndman, 1996), to find the ν points with lowest density in the first two principal components. These points are classified as outliers. Extended details of the method, including the list of features, can be found in Appendix B.2.2. This method provides an ordering of the outliers but not a severity measure, as illustrated by Figure 3.3.2.

Rennie et al. (2021a): Uses functional depth analysis to classify legs as outliers based on their booking curves. The method we propose in this chapter extends that suggested in Rennie et al. (2021a) through two features: (i) the use of severity measures to rank outliers; and (ii) the inclusion of network effects. To isolate the effects of each of these features, we perform two separate benchmark tests:

We evaluate the effect of ranking outliers by measuring the increase in precision when ranking outliers. For example, we consider the precision in the top 5 ranked departures, versus 5 randomly chosen departures with non-zero outlier probabilities.

The change in precision when considering the top R departures, $\Delta(Precision)_R$, is given by:

$$\Delta(Precision)_R = \frac{TP_R}{TP_R + FP_R} - \frac{TP_{R(random)}}{TP_{R(random)} + FP_{R(random)}}, \tag{3.3.5}$$

where $TP_{R(random)}$ is the number of true positives in a random selection of R departures with non-zero severity, and $FP_{R(random)}$ is defined analogously for false positives. Figure 3.3.1b visualises the relevant results.

We quantify the value of accounting for network effects by computing ranked alert lists for each leg in isolation. We then compare the true positive rates to the aggregated, network-driven approach presented in this chapter. Figure 3.3.4 illustrates that analysis.

3.3.5 Forecast adjustments for outlier demand

The aim of identifying outlier demand in RM systems is to support analyst interventions. This raises the difficult question of predicting the consequences from analyst adjustments throughout the network. As a step in this direction, we analyse a best-case-scenario, assuming that the adjustment is made with foresight, before the start of the booking horizon. We compare the revenue under three different types of adjustment:

- **Adjustment 1 (conservative)**: Adjust only forecasts of affected single-leg itineraries. E.g., for an outlier creating additional demand for itinerary AC, increase the forecasts of itineraries AB and BC.

- **Adjustment 2 (aggressive)**: Adjust forecasts of itineraries that include at least one of the affected legs. E.g., for additional demand for itinerary AC, adjust all itineraries including either leg AB or leg BC – i.e., itineraries AB, AC, AD, AE, BC, BD, and BE.

- **Adjustment 3 (balanced)**: Adjust forecasts of affected single-leg itineraries and the *cluster-spanning* itinerary – in this case, AE. E.g., for additional demand for itinerary AC, adjust itineraries AB, BC, and AE. The motivation for adjusting AE (ahead of other itineraries) is that in general this will be the most popular itinerary in the cluster.

These three adjustments are not the only choices available to analysts. However, they represent options that stretch across the spectrum of how fully network effects are to be taken into account in the adjustment. Adjustments 1 (conservative, leg-based adjustments only) and 2 (aggressive, all potential network effects) are the two extremes of accounting for network effects in the forecast adjustments. Adjustment 3 (balanced) is a compromise between the two previous adjustments, which is more conservative than Adjustment 2 but still identifies the itinerary most likely to be the source of outlier demand. Further options would be to include more than just the cluster-spanning itinerary in an alternative to Adjustment 3, but this leaves another choice of which itineraries to prioritise. As a lower bound, we compute the revenue when **no adjustment** is made. As an upper bound, we implement an **oracle adjustment**, i.e., only adjusting the forecasts of affected itineraries. We compare the revenue as the level of outlier demand ranges from -60% to +60% of the average leg level demand.

3.3.6 Experimental results on detecting outliers in multiple legs

As a first experiment, we consider the scenario where outlier demand equally affects all itineraries and legs within the cluster. For this scenario, Figure 3.3.1a illustrates how the true positive rate (TPR) increases when increasing the length of alert list. In this figure, the red line indicates the number of genuine outliers. The true positive rates are promising, with a TPR of around 0.2 for a list length of 1. Since there are five genuine outliers, this indicates that a genuine outlier is almost always ranked top. Results under different functional depth thresholds are found in Appendix B.3.2.

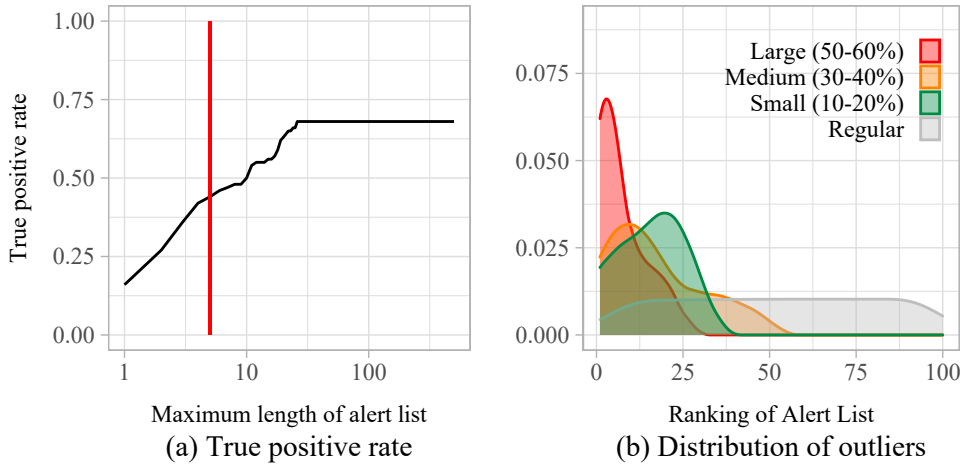


Figure 3.3.1: Performance for demand-volume outliers in all itineraries

Figure 3.3.1b shows the distribution of each magnitude of outlier in the alert lists. Under the proposed method, the modes of the distributions generally fall where they should, as larger outliers are ranked higher. The smaller variance in the ranking of the larger magnitude outliers indicates that they are easier to detect. The higher variance of the medium sized outliers can be explained as the ranking of a medium sized outlier

is dependent on which other types of outliers occur: if there is a large and a medium outlier, the medium outlier is ranked lower; if there is a small and a medium outlier, the medium outlier is ranked higher. Appendix B.3.2 further analyses the distribution of identified outliers across different legs.

PCA+HDR benchmark results. The PCA+HDR approach requires a given number of outliers to detect (rather than a threshold), ν, as input . Therefore, Figure 3.3.2 compares the performance of the benchmark method under different numbers of outliers to our method (denoted as FD+Agg).

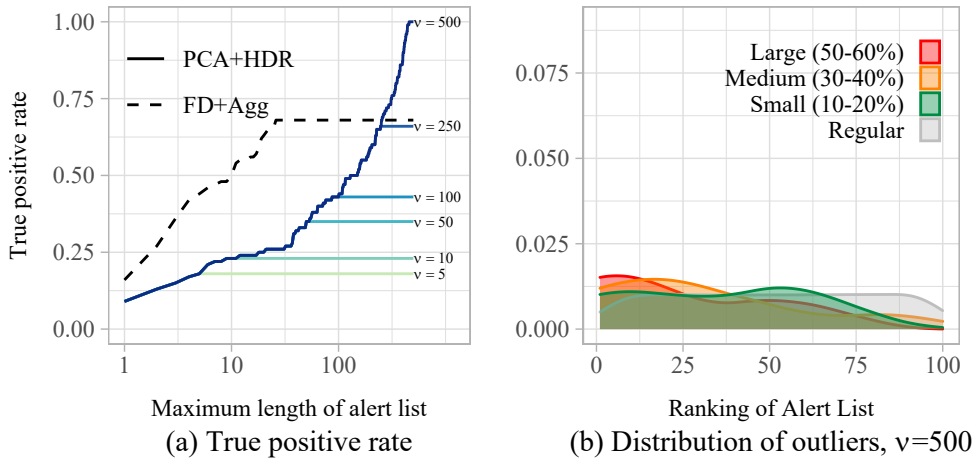
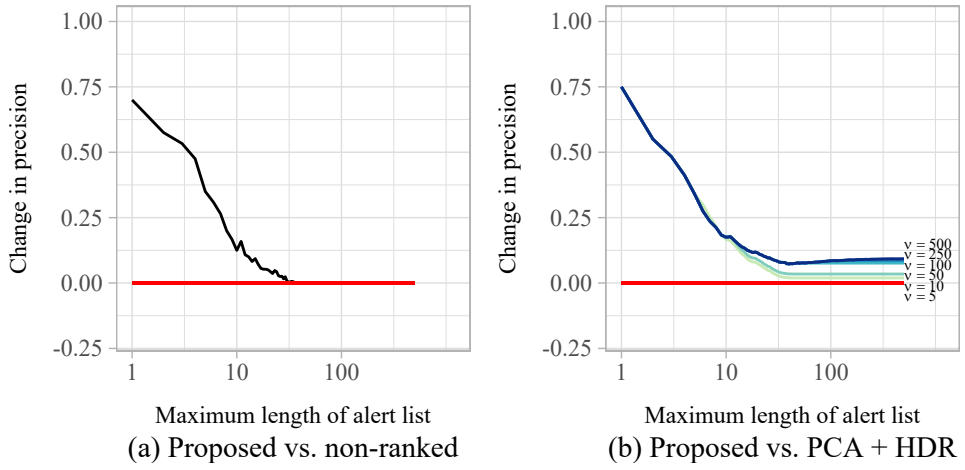


Figure 3.3.2: Performance comparison with PCA+HDR benchmark for demand-volume outliers in all itineraries

Figure 3.3.2a shows that the true positive rate achieved by our proposed approach, FD+Agg, consistently exceeds that achieved by PCA+HDR. In order to achieve the same level of true positive rate, PCA+HDR would need to classify around 250 departures (i.e. 50%) as outliers. In comparison, FD+Agg achieves this rate after about 30 classified outliers. The distribution of PCA+HDR shown in Figure 3.3.2b also returns

the modes in the correct order. However, there is much more overlap between the distributions, showing its inability to correctly rank the outliers.

Non-ranked Rennie et al. (2021a) benchmark results. Figure 3.3.3a highlights how the precision improves when ranking outliers as opposed to listing them in random order. Ranking particularly improves precision when the alert list covers only a small number of departures. As domain experts indicate that analysts cannot target more than 1% of departures, ranking focuses resources and thereby provides large benefits in practice. Nevertheless, Figure 3.3.3a (when contrasted with Figure 3.3.1a) also highlights the trade-off between reducing the number of false alerts and identifying all outliers. A shorter length of alert list increases precision, but reduces the true positive rate.



(a) Proposed vs. non-ranked (b) Proposed vs. PCA + HDR

Figure 3.3.3: Change in precision from ranking detected outliers

The increase in precision from applying our method compared to the PCA+HDR benchmark is similar to the increase in precision from the inclusion of the ranking (see Figure 3.3.3b). This suggests that the PCA+HDR benchmark performs reasonably

well in terms of outlier detection, but poorly in terms of ranking the outliers.

Leg-based Rennie et al. (2021a) benchmark results. Figure 3.3.4 shows the true positive rate when a ranked alert lest is computed for each leg in isolation versus in the proposed aggregated manner. Here, we consider outlier demand generated by a 50% increase in the affected legs as an illustrative example. We analyse detection performance by breaking down results in terms of which itinerary the outlier demand is generated in. We show only the results relating to itineraries AB, AC, AD, and AE. Figure B.3.9 in Appendix B.3.2 details results for the further itineraries yielding similar conclusions. For results when outlier demand is generated across combinations of itineraries, see Appendix B.3.2.

In all cases, the true positive rate for clusters is higher than in any of the individual legs. This is because when considering the leg's bookings in isolation for outlier demand that affects multiple legs, the noise from other itineraries prevents detecting the outlier in every leg. However, clustering increases the number of detected genuine outliers.

Clustering is most beneficial when the outlier demand affects the most legs i.e. itinerary AE, as shown in Figure 3.3.4a. The lower true positive rates in legs AB and DE are due to different combinations of itineraries also utilising these legs. The aggregation is less beneficial when outlier demand affects an itinerary consisting of only one or two legs, since we aggregate the analysis across legs that are actually not affected by outlier demand. However, there is a modest gain in true positive rate even in this case. This is due to the knock-on effects of decreased capacity on the

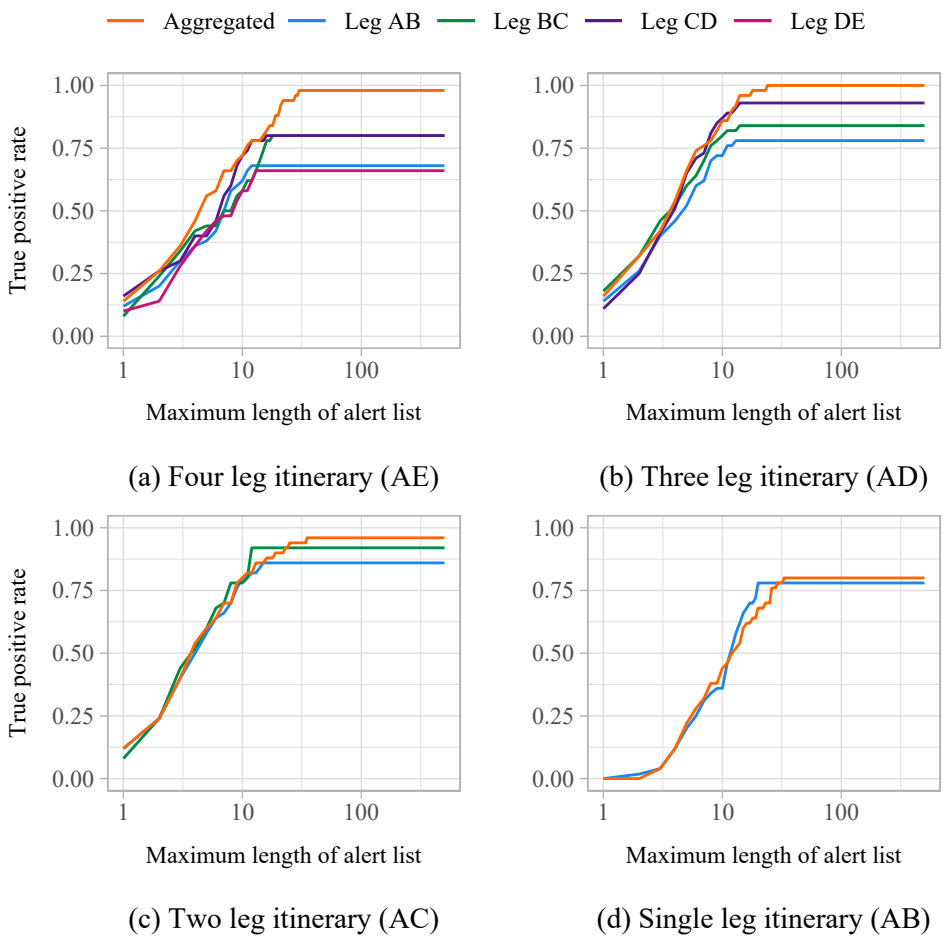
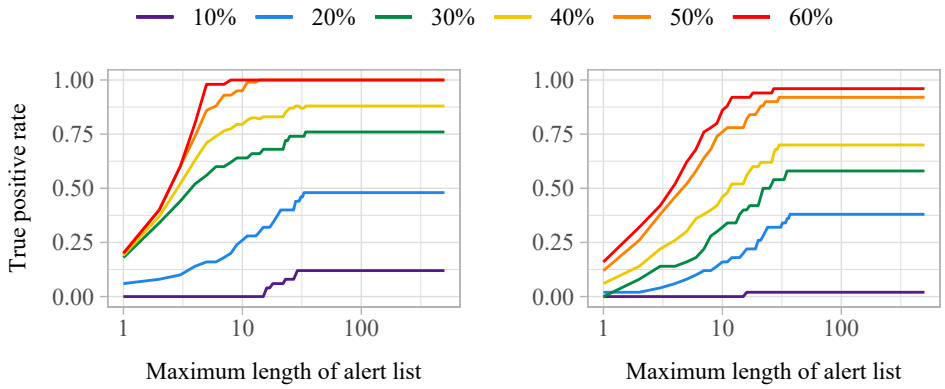


Figure 3.3.4: True positive rate for single itinerary outliers

affected legs, impacting the bid prices for any itineraries which include these legs. For some lengths of alert list, the leg-level true positive rates are higher than the aggregated approach, due to false positives from unaffected legs being included in the list. However, even for itinerary AB (Figure 3.3.4d), where false positives from unaffected legs are most likely, the difference is small and cancelled out by the overall increase in true positive rate.

Different magnitudes of outliers. To better understand outlier detection performance of our method, we break down the results by magnitude of outliers in Figure

3.3.5. When outliers are generated by minor changes in demand levels, they are dif-

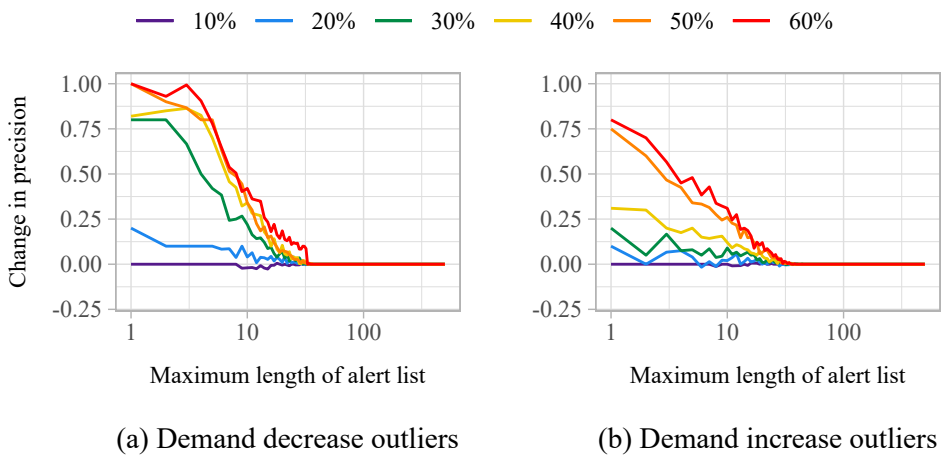


(a) Demand decrease outliers (b) Demand increase outliers

Figure 3.3.5: True positive rate for homogeneous demand-volume outliers by magnitude

ficult to detect, resulting in low true positive rates. Given the significant overlap between the distribution of outlier demand with a 10% change in magnitude and that of regular demand, this is to be expected. Therefore, 10% demand changes effectively provide a lower bound on how big an outlier needs to be in order to be detected.

As the magnitude of the outliers increases, they become easier to detect and true positive rates are higher, with peak rates reached with shorter alert lists. Thus, genuine outliers are more likely to be ranked higher when they are caused by larger demand changes. For demand decreases of at least 50%, the true positive rate is very close to the optimal detection rate. Negative demand outliers are slightly easier to detect than positive demand outliers, meaning shorter alert lists are required. This is due to the demand censoring imposed by the booking controls and capacity restrictions.



(a) Demand decrease outliers (b) Demand increase outliers

Figure 3.3.6: Increase in precision for homogeneous demand-volume outliers by magnitude

Figure 3.3.6 shows the precision gap over randomly ordered lists. Once more, larger magnitude outliers result in larger precision improvements from ranking, while detecting minor outliers gains little over random selection. Similarly, we observe that detecting negative demand outliers gains slightly more precision in comparison to detecting positive outliers of the same magnitude. Additional results regarding false discovery rates are available in Appendix B.3.2.

3.3.7 Revenue benefits from forecast adjustments for outlier demand

Figure 3.3.7 shows the revenue generated by outlier demand for each of the three possible choices of adjustment described in Section 3.3.5. We show the results for four of ten itineraries contained within these four legs. The results for the other six itineraries are similar to those presented here for the same corresponding leg length.

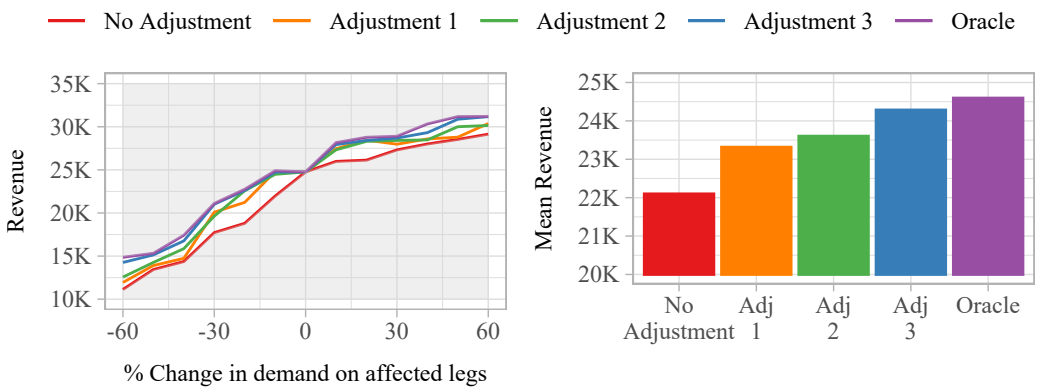
Appendix B.3.3 details these results as well as further results on adjustments after outlier detection.

When outlier demand affects all four legs in the cluster (Figure 3.3.7a), any type of adjustment is always better than no adjustment. Besides the oracle, the best choice is adjustment 3, i.e., the balanced approach where the forecasts of the cluster-spanning itinerary and the individual leg are adjusted. Adjustment 3 is able to obtain, on average, 87% of the additional revenue gained under the oracle adjustment. Similar results are obtained when the outlier demand affects three legs (Figure 3.3.7b).

When outlier demand affects only a single-leg itinerary (Figure 3.3.7d), adjustment 1 (the conservative adjustment) and the oracle adjustment coincide. The aggressive approach of making an adjustment to all itineraries which include the affected leg yields less revenue than no adjustment. For example, although leg AB is correctly adjusted, the erroneous adjustment to itineraries AC, AD, and AE results in incorrect forecasts for legs BC, CD, and DE. The asymmetry between adjustment to positive and negative outlier demand is due to the level of demand being bounded below by 0.

Similar results emerge when the outlier affects only two of the affected legs (Figure 3.3.7c), though the negative consequences of over-adjusting all potentially affected itineraries are less severe, as this causes fewer superfluous adjustments.

The negative impact of adjusting unaffected itineraries highlights the importance of correctly clustering legs ahead of outlier detection. The closer the outlier demand itinerary is to the cluster spanning itinerary, the less risky it is to adjust all affected itineraries within a cluster, and the more benefit can be gained from doing so. From



(a) Four leg itinerary (AE): revenue by demand change (left), mean revenue (right)

(b) Three leg itinerary (AD)

(c) Two leg itinerary (AC)

(d) Single leg itinerary (AB)

Figure 3.3.7: Revenue generated under different itinerary-level forecast adjustments, where the subtitle indicates the location of the outlier

a managerial perspective, the *best* adjustment (other than the oracle) depends on the transport provider's objective. To maximise revenue when the most common outlier (e.g. itinerary AE) occurs, the conservative approach of adjustment 3 is preferable. Conversely, if the objective is to minimise risk to revenue even in the more unlikely scenarios (e.g. an outlier in itinerary AB), adjustment 1 should be preferred. Overall, however, there are clear benefits from forecast adjustment.

3.4 Empirical study of Deutsche Bahn booking data

To demonstrate practical applicability of the proposed clustering and outlier detection, we apply it to a set of empirical data obtained from Deutsche Bahn. The Deutsche Bahn long-distance network consists of over 1,000 train stations, letting the provider offer more than 110,000 origin-destination combinations. The numbers grow further when accounting for alternative transfer itineraries and for multiple departures per day. Figure 3.4.1 shows the empirical distribution of the number of legs included in itineraries that passengers booked with Deutsche Bahn in November 2019. Only 7% of passengers booked single-leg itineraries, whereas almost half of all booked itineraries span five or more legs.

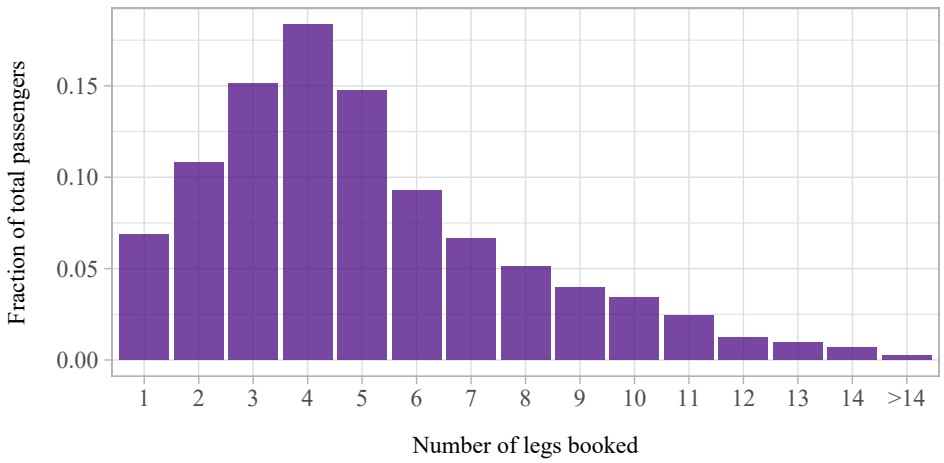


Figure 3.4.1: Distribution of number of legs per booked itinerary

3.4.1 Clustering legs in the Deutsche Bahn network

We consider a section of the Deutsche Bahn railway network that consists of two intersecting train lines over a total of 27 stations – see Figure 3.4.2. The red train

arrives at the connecting stations before the blue train. Hence, the network offers three transfer connections: changing from red to blue at either Fulda, Kassel-Wilhelmshöhe, or Göttingen. This creates 240 potential travel itineraries.

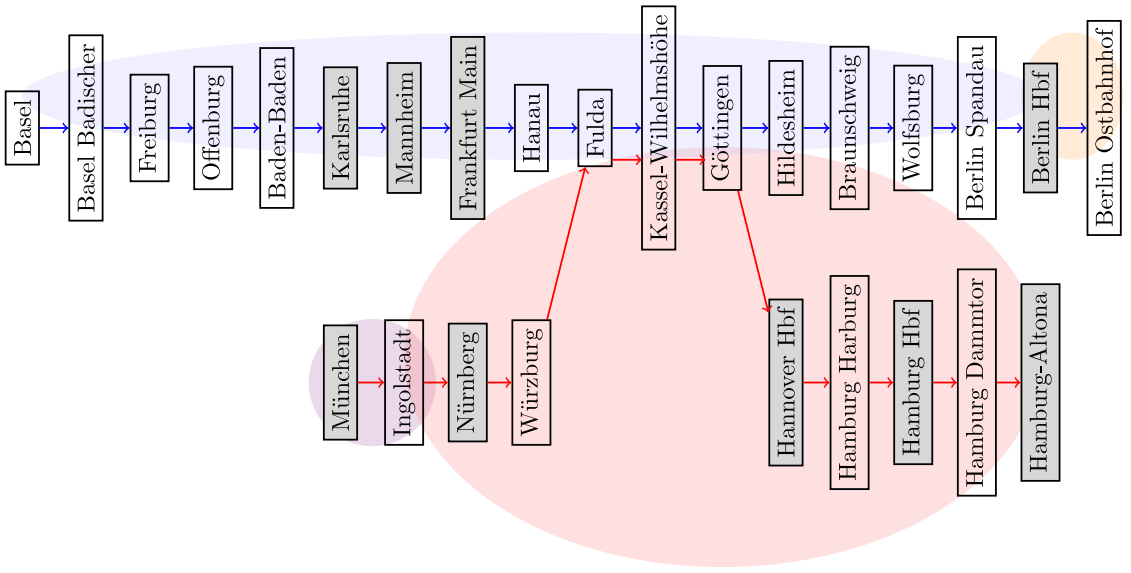
For each leg in this network section, Deutsche Bahn provided 359 booking patterns for departures between December 2018 and December 2019. Each booking pattern ranges over 19 booking intervals; the first observation occurs 91 days before departure.

We firstly apply the correlation-based clustering approach of Section 3.2.1, using a threshold of 0.5, such that only legs with a minimum correlation of 0.5 can be in the same cluster. In Figure 3.4.2a, coloured bubbles indicate the four resulting clusters: Each train line splits into one large and one small cluster.

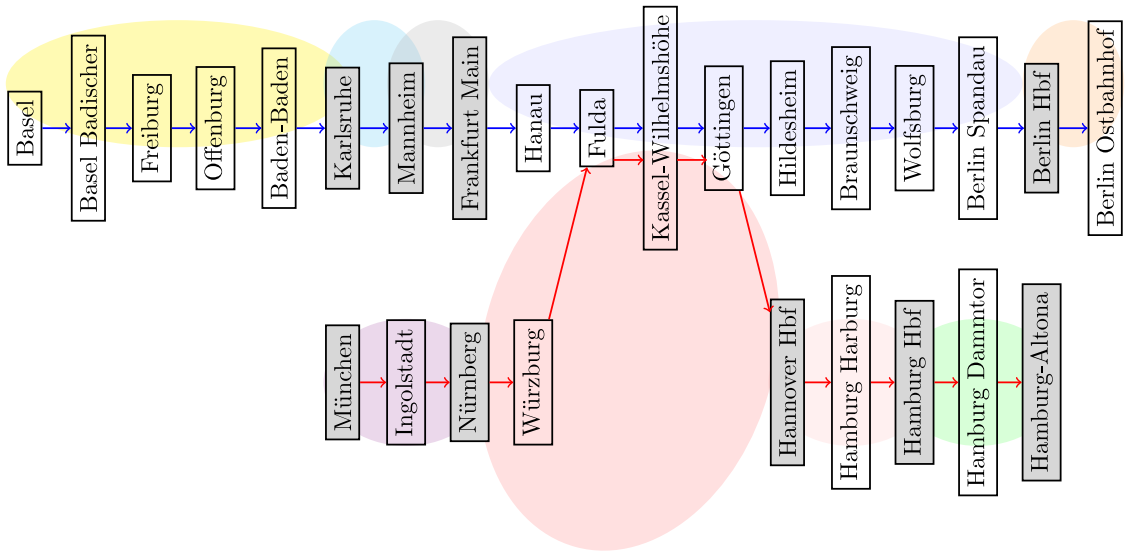
To evaluate clustering on real data, where the true underlying demand for each itinerary is unknown, we use the network topology to check whether resulting clusters are plausible. To that end, we propose the following set of rules:

- Different train lines must be in different clusters. Even when passengers can transfer between lines, we expect relatively few passengers to make the same connection. Further, for forecasting and analyst interventions, it makes sense to consider train lines separately.

- Train lines are further split into different clusters on either side of a major station. As many passengers leave the train at a major station and many *different* passengers board, we shall assume a relatively small proportion of passengers book itineraries that pass a major station. Similarly, given that itinerary demand share is driven by which journeys are most common, and passengers often



(a) Correlation-based clustering, $\rho \geq 0.5$



(b) Rule-based cluster

Figure 3.4.2: Comparison of correlation-based and rule-based clustering of Deutsche Bahn network

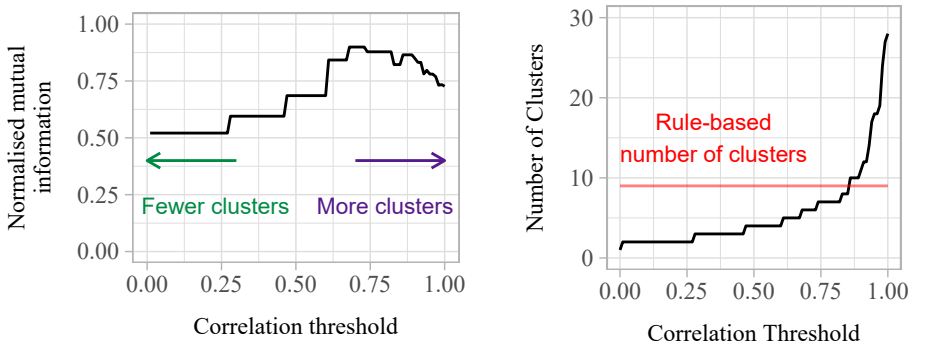
either board or alight at a major station, it is intuitive to have a cluster that contains the legs between major stations.

Deutsche Bahn assigns an ordinal indicator of importance to each station, ranging from from 1 to 7. We define a *major station* to be in *Category 1*. The entire Deutsche Bahn network includes 21 major stations, where the considered network section includes 9. Figure 3.4.2b highlights major stations in grey and shows the clusters resulting from the rules listed above.

Whereas the correlation-based clustering returns four clusters, the rule-based clustering returns nine. Nevertheless, the resulting clusters share similar features. Firstly, the two distinct train lines end up in different clusters in either approach. For legs in distinct train lines, correlation tends to be higher between legs that share a transfer station, but not to a convincing extent – correlation is at most 0.22. A correlation threshold of 0.27 creates two clusters (one for each train line). Secondly, the break points for the correlation-based approach are a subset of the break points, i.e., major stations, in the rule-based approach. We conclude that the correlation-based approach achieves similar results as the rule-based approach without expert input - relying only on booking data.

We can formally compare clustering results using the **Normalised Mutual Information (NMI)** (Amelio and Pizzuti, 2015). The NMI is 1 if two clusterings are identical, and 0 if they are completely different (see Appendix B.1.4 for details). Figure 3.4.3a shows the NMI between the correlation- and rule-based approaches while varying the threshold in the correlation-based approach from 0 to 1. This shows that both approaches achieve similar results, with an NMI reaching 0.899. The approaches are generally more similar at higher correlation thresholds (around 0.7), since the rule-based approach generally creates more clusters. Figure 3.4.3b compares the number

of clusters of the two approaches – as the correlation threshold changes, the number of clusters ranges from 1 (everything in a single cluster) to 28 (each leg in its own cluster), demonstrating the flexibility of the correlation-based approach.



(a) Normalised mutual information for rule-based and correlation -based clustering

(b) Number of clusters for rule-based and correlation-based clustering

Figure 3.4.3: Comparison of rule-based and correlation-based clustering

Here, we applied rule-based clustering only to evaluate the plausibility of the results from correlation-based clustering. We do not advocate for it as a method in itself. A rule-based approach, where the clusters are based on domain experts' categorisations, would not be able to respond to the evolving importance of stations across different train lines and departure times. Notably, the correlation-based method is not simply a data-driven method for uncovering major stations, but rather for identifying legs where multi-leg itineraries cause similar booking patterns, and thus could change and adapt over time. We further evaluate clustering performance in a simulation study, where the itinerary-level demand is known, in Appendix B.3.1. The results in the remainder of the chapter rely on correlation-based clustering.

3.4.2 Detecting outliers in multiple legs

Demonstrating the proposed outlier detection approach on empirical data cannot precisely judge detection accuracy, given there is no labelled data on genuine outliers. However, this analysis illustrates the full process of outlier detection on empirical data including, e.g., seasonality and underlines practical implications.

For this analysis, we consider a cluster of four legs from the Deutsche Bahn network with stations anonymised and denoted by A, B, C, D, and E. This cluster results from applying the correlation-based clustering to a new section of the Deutsche Bahn network to Figure 3.4.2.

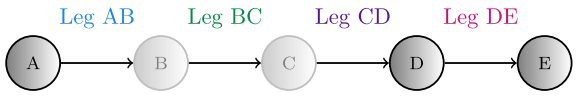
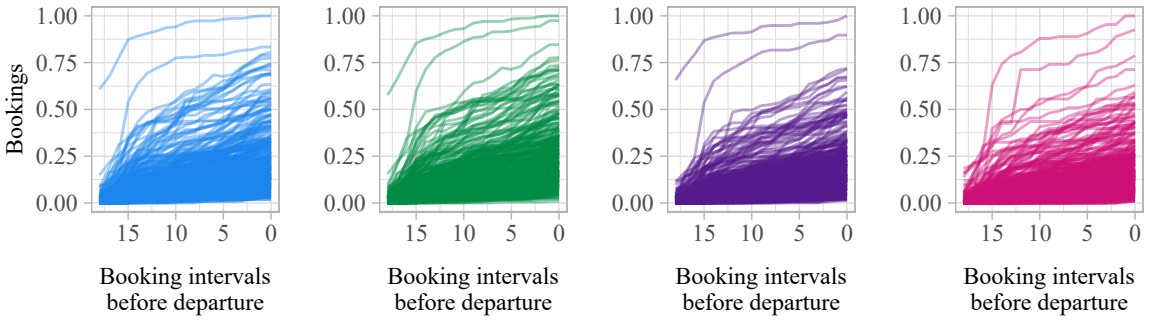


Figure 3.4.4: Four leg cluster within the Deutsche Bahn network

Figure 3.4.5 shows the booking patterns for each of the four legs; bookings are scaled to be between 0 and 1. From initial visual inspection, the structure of the booking patterns appears similar, with some obvious outliers appearing across multiple legs.

To pre-process the data for outlier detection, we transform the booking patterns by applying a functional regression model (Ramsay and Silverman, 1997). We then apply the outlier detection to the residual booking patterns. In this pre-processing, we correct for three factors: (i) departure day of the week; (ii) departure month of the year; and (iii) the length of the booking horizon.[1]

[1]Deutsche Bahn offer a regular booking horizon of 6 months, with the first observation of bookings



(a) Leg AB (b) Leg BC (c) Leg CD (d) Leg DE

Figure 3.4.5: Booking patterns for each leg

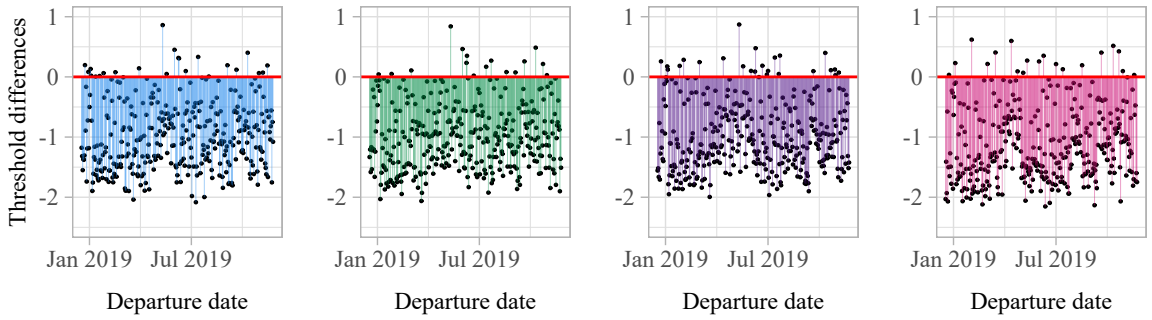
The functional regression fits a mean function to the booking patterns for each different factor in the model. Table B.4.1 in Appendix B.4.1 compares models including different factors. Let $y_{nl}(t)$ be the n^{th} booking pattern for leg l. Then:

$$\begin{aligned} y_{nl}(t) = \beta_{0l}(t) + \beta_{1l}(t)\mathbb{1}_{Mon_{nl}} + \beta_{2l}(t)\mathbb{1}_{Tue_{nl}} + \beta_{3l}(t)\mathbb{1}_{Wed_{nl}} + \\ \underbrace{\beta_{4l}(t)\mathbb{1}_{Thu_{nl}} + \beta_{5l}(t)\mathbb{1}_{Fri_{nl}} + \beta_{6l}(t)\mathbb{1}_{Sat_{nl}}}_{\text{Departure Day of the Week}} + \\ \beta_{7l}(t)\mathbb{1}_{Jan_{nl}} + \beta_{8l}(t)\mathbb{1}_{Feb_{nl}} + \beta_{9l}(t)\mathbb{1}_{Mar_{nl}} + \\ \beta_{10l}(t)\mathbb{1}_{Apr_{nl}} + \beta_{11l}(t)\mathbb{1}_{May_{nl}} + \beta_{12l}(t)\mathbb{1}_{Jun_{nl}} + \beta_{13l}(t)\mathbb{1}_{Jul_{nl}} + \\ \underbrace{\beta_{14l}(t)\mathbb{1}_{Aug_{nl}} + \beta_{15l}(t)\mathbb{1}_{Sep_{nl}} + \beta_{16l}(t)\mathbb{1}_{Oct_{nl}} + \beta_{17l}(t)\mathbb{1}_{Nov_{nl}}}_{\text{Departure Month of the Year}} + \\ \underbrace{\beta_{18l}(t)\mathbb{1}_{Shorter\ Horizon_{nl}}}_{\text{Length of Booking Horizon}} + e_{nl}(t). \end{aligned} \tag{3.4.1}$$

where e.g., $\mathbb{1}_{Mon_{nl}} = 1$ if departure n relates to a Monday, 0 otherwise. In this model, $\beta_{0l}(t)$ represents the average bookings for Sunday departures in December, with a regular length of booking horizon, and $\beta_{pl}(t)$ for $p > 0$ represent deviations from

occurring around 3 months before departure. Due to schedule changes, shorter booking horizons of 3 months apply for departures from mid-December to mid-March.

this mean pattern. The $\beta_{pl}(t)$ are functions of time, which allows for relationships between factors to evolve over the booking horizon. Given that functional depths are calculated independently for each leg, we apply the regression model independently for each leg. The resulting residuals are shown in Appendix B.4.2, Figure B.4.1.



(a) Leg AB (b) Leg BC (c) Leg CD (d) Leg DE

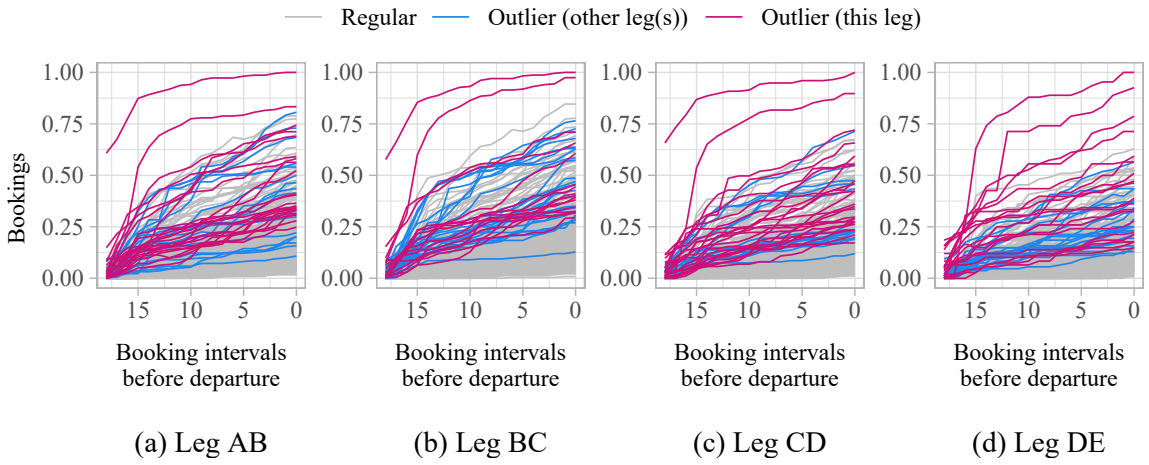
Figure 3.4.6: Threshold exceedances per leg, z_{nl}

Functional regression preserves the correlation between different legs, as verified in Appendix B.4.6, Table B.4.2b. The clustering approach can consider either the correlations between the booking patterns or the residual booking patterns. Given that the functional depths (the basis for the outlier detection) are calculated on the residuals, we suggest using correlation between residual patterns to define the clusters. For this data set, the same clusters resulted in either case.

We calculate the functional depth of each booking pattern and compute the threshold as described in Section 3.2.2. We then transform the depths as per equation (3.2.2) to obtain z_{nl}, as shown in Figure 3.4.6. The sums of threshold exceedances, z_n, were shown earlier in Figure 3.2.2, with the empirical distribution and fitted generalised Pareto distribution shown in Figures 3.2.3a and 3.2.3b, respectively.

Figure 3.4.7 highlights the outliers detected in each leg in pink, while depicting

outliers detected in other legs but *not* in that leg in blue. Regular patterns are grey.



(a) Leg AB (b) Leg BC (c) Leg CD (d) Leg DE

Figure 3.4.7: Outliers detected in booking patterns

Of the 40 outliers (11% of departures) detected across all legs, 23 outliers (almost 60%) could be attributed to known events or holidays. When considering only the top 10 outliers, the percentage rose to 70%. A further departure detected as an outlier had been previously flagged by Deutsche Bahn. The firm implemented a booking stop to control sales on that departure for multiple connected legs. Appendix B.4.5 provides further details on the distribution of identified outliers across legs.

3.5 Conclusion and outlook

In this chapter, we proposed a two-step method for (i) clustering legs in a mobility network that are likely to benefit from joint outlier detection, and (ii) detecting outlying demand within such a set of legs. Furthermore, we presented an approach to rank identified outliers according to their severity, creating an alert list to aid analysts in prioritising demand forecast adjustments.

The simulation study demonstrated the robustness of the method in a range of outlier demand scenarios. It highlighted that aggregating the analysis across clustered legs that share common outliers improves both detection rate and precision. Further, the ranked alert list correctly identified the most critical outliers. Last but not least, we measured the potential revenue benefits of identifying and adjusting for demand outliers in a network setting by applying a choice of forecast adjustments and gauging the resulting revenue. We show that taking into account the similarity of the legs can improve revenue in most scenarios. In the less likely scenario where only one or two legs of a cluster are affected by outlier demand, risk-averse firms may prefer leg-level adjustments.

By applying the proposed approach to empirical booking data, we demonstrated the type of data observed in practice and showed how to account for additional practical considerations, such as trend and seasonality. Based on insights from analysing empirical data, we constructed a simulation to evaluate how successfully our method detects outliers under laboratory conditions.

Further research is needed to consider the practical aspects of outlier detection in live revenue management systems from the perspective of decision support. Such research should particularly focus on effective ways to visualise outliers in networks and to communicate the ranked alert list to RM analysts. An interesting avenue of further research would be to incorporate a feedback element whereby analysts mark outlier alerts as useful or not useful. A supervised learning approach e.g. one-class-classifiers, could then be combined with our proposed outlier detection routine to filter out false alerts.

Another research opportunity would be to consider how the aggregated outlier detection may be adapted for other areas of network revenue management, e.g., in hotels, where correlation is induced by bookings for multiple consecutive nights. In particular, investigating the use of alternative clustering approaches is of interest - especially where the clusters are likely to be of different structures compared to the rail industry e.g. in the airline industry where hub and spoke networks are more common than lines. Whilst this chapter relied on clustering to improve outlier detection, we believe that the clustering approach is a useful contribution in and of itself. For example, clustering presents additional research avenues such as its application to improving network-level forecasting; supporting the planning for future new stations; evaluating how the transport network structure is changing over time; or defining different travel zones.

Chapter 4

Analysing and visualising bike-sharing demand with outliers

Bike-sharing is a popular component of sustainable urban mobility. It requires anticipatory planning, e.g. of terminal locations and inventory, to balance expected demand and capacity. However, external factors such as extreme weather or glitches in public transport, can cause demand to deviate from baseline levels. Identifying such outliers keeps historic data reliable and improves forecasts. In this chapter we show how outliers can be identified by clustering terminals and applying a functional depth analysis. We apply our analysis techniques to the Capital bike-sharing data set as the running example throughout the chapter, but our methodology is general by design. Furthermore, we offer an array of meaningful visualisations to communicate findings and highlight patterns in demand. Last but not least, we formulate managerial recommendations on how to use both the demand forecast and the identified outliers in the bike-sharing planning process.

4.1 Introduction and Background

As a component of sustainable urban mobility, bike-sharing is on the rise in cities around the world (Shaheen et al., 2010). However, careful planning is required to make it more attractive than fuel-based mobility alternatives and thereby maximise its positive environmental impact. In particular, bike-sharing systems require terminals to cover relevant locations and balanced inventory levels to ensure adequate service provision. When, for example, a terminal is mostly used to pick-up bikes, re-balancing ensures a steady supply and avoids service denials.

Optimisation routines can be used to determine, for example, the best distribution of terminals across the service area (Ciancio et al., 2017), the best distribution of bikes across terminals (Zhu, 2021), and the best path for truck drivers to take when re-distributing bikes each day (Schuijbroek et al., 2017). Optimisation procedures that determine stock levels per terminal rely on predicted demand. Since inefficient re-balancing operations are a major cost driver for operators (Schuijbroek et al., 2017), identifying demand outliers to improve efficiency in bike-sharing systems is highly important. Unaccounted-for outliers can affect bike-sharing systems in two ways: (i) outliers in historic data contaminate the forecasts used in future inventory management, and (ii) on the day demand levels may indicate that the schedule is non-optimal for the current day and drivers should be re-routed.

We define *outlier demand* as a short-term systematic change in demand, resulting in usage levels which deviate from *regular* usage. In this chapter, we focus on terminals, as these are the target of inventory rebalancing efforts. In contrast to other

classical mobility problems, such as those related to buses or trains, bike-sharing capacity is on the vertices of the transport network, rather than on the edges.

As an example of existing work in this area, Neumann-Saavedra et al. (2021) discuss the problem of variability in bike-sharing demand and propose a rule-based method to adjust the redistribution plan when demand differs from the forecast. In a simulation study, they show that service levels can be improved when adjustments are made to the optimal redistribution plan. Wider literature on outlier detection in transport planning is scarce – e.g., Rennie et al. (2021b) consider identifying and correcting for outliers in revenue management systems in railways. Talvitie and Kirshner (1978) find that outliers can have a substantial effect on the predictions of usage of different urban transport modes, but only apply a simplistic trimming method to identify outliers. In the road traffic domain, Guo et al. (2015) suggest a procedure for identifying outliers in real time based on the conditional variance of predictions, and determine that incorporating information on such outliers into future predictions increases the systems performance.

Furthermore, as indicated, e.g., in Basole et al. (2020), to account for demand outliers and adjust planning, experts require meaningful visualisations. Therefore, we propose a set of visualisations to help identify and analyse spatial and temporal patterns in the detected outliers. For example, a subset of terminals may be predisposed to outliers and as such, this area would be a good target for a new terminal, should one be added.

To combine automated outlier detection, manual analysis, demand forecasts, and planning, we suggest the following process for analysing bike-sharing demand data (see

Figure 4.1.1): First, a baseline demand forecast supports anticipative planning, e.g. of inventory levels. Second, this baseline can be used to normalise observed usage data. Using the resulting observations, analysts can cluster terminals with similar usage patterns to support both planning adjustments and outlier detection. When detecting outliers in a cluster's usage patterns, these are visualised to enable manual outlier evaluation. Insights from this analysis can be used to both clean the data that underlies the baseline forecast and to extend the baseline forecasting model.

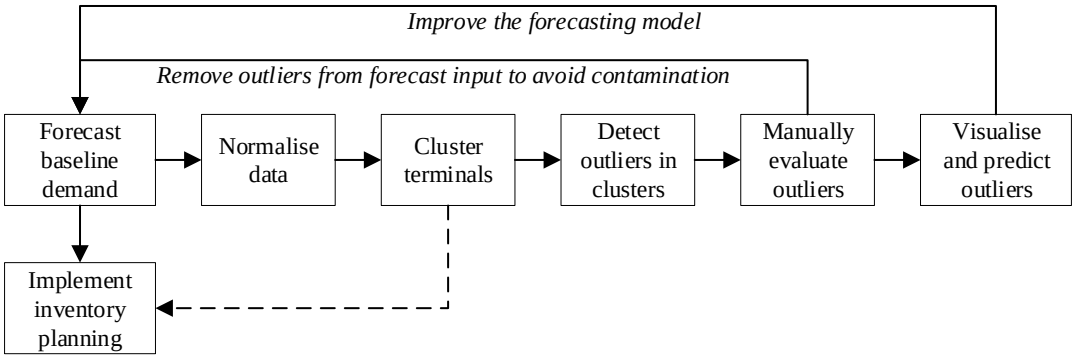


Figure 4.1.1: Flowchart of process for analysing bike-sharing demand data

In this chapter, we analyse the Capital Bikeshare data set, which is publicly available at Capital Bikeshare (2021). This data set is commonly used to test forecasting approaches for bike-sharing (Ma et al., 2015; Hamilton and Wichman, 2018), yet these methods typically do not account for outliers. In Section 4.2, we introduce the data set and perform an exploratory analysis. Sections 4.3 and 4.4 then model the temporal and spatial patterns in demand for bike-sharing. In Section 4.5, we provide a methodology for identifying outlying demand for bike-sharing services. The

results of applying the outlier detection method to the Capital Bikeshare data are then discussed in Section 4.6.

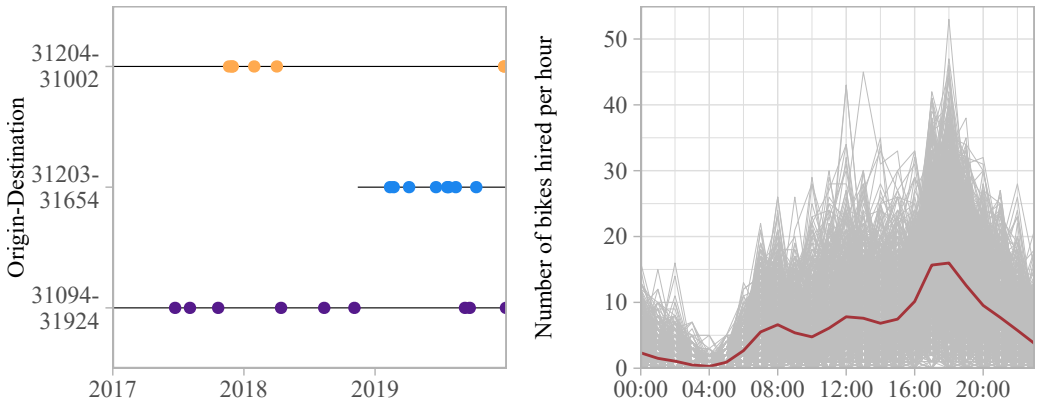
In summary, this chapter contributes (i) an in-depth analysis of temporal patterns in usage of Capital Bikeshare services; (ii) a method for spatial clustering of bike-sharing terminals based on geographic proximity and similarity of usage patterns; (iii) an investigation of temporal trends in detected outliers and the factors that may cause them; and (iv) an analysis of spatial patterns of the outliers detected. Our methodology is data-driven and general by design, and not tailored to specifics related to Washington D.C., and can thus be readily applied to all bike-sharing data sets around the world.

4.2 Capital Bikeshare data

The Capital Bikeshare data set spans a three year period, from January 1 2017 to December 31 2019. It describes every recorded trip by its time of pick-up, time of drop-off, pick-up terminal location, and drop-off terminal location. The data set features only those 578 terminals that recorded at least one pick-up or drop-off within the recorded time frame.

Out of a potential 334,084 unique *origin-destination (O-D)* pairs, the data set records 105,735 O-D pairs that customers completed based on pick-ups and drop-offs. As examples, the times of bike rentals for three different O-D pairs are shown in Figure 4.2.1a, with each dot representing one journey. Note that station 31654 opened in November 2018, and so data is only available from that date onward for O-D pair

31203-31654.



(a) Time-stamp data for three origin-destinations

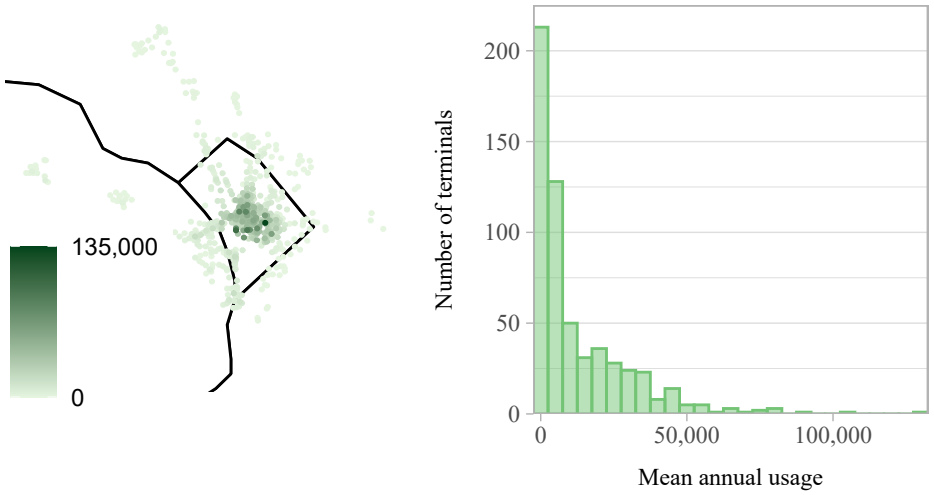
(b) Daily usage patterns for terminal 31203

Figure 4.2.1: Origin-destination (O-D) level data and aggregated daily usage patterns

Data cleansing If the first use of a terminal in the data set was not January 1st 2017, we check historical data from 2016 for any usage to determine if the terminal was open. If there are no earlier bookings, we consider the terminal as newly opened from the time of its first recorded trip. Capital Bikeshare pre-process the data to remove trips that are made by staff for system maintenance and any trips with a journey time of less than 60 seconds (as these may be false starts).

Data aggregation Given the large number of O-D pairs, very few journeys are recorded per unique pair on average. This makes it difficult to detect meaningful patterns, or any deviation from such a pattern, on the O-D level. When numbers are this small, noise dominates over any trend, as also pointed out in related research on forecasting slow-moving retail products (Jha et al., 2015). To alleviate the prob-

lem of small numbers, we aggregate trips as pick-up and drop-off events, considering usage per terminal rather than O-D pairs. To further reduce the problem of sparse observations, and to make observations comparable over time, we aggregate usage by hour of day (Petropoulos and Kourentzes, 2015). Specifically, we define the *daily usage pattern* to be a time series of the number of times per hour that a terminal is used, either pick-up or drop-off – see Figure 4.2.1b. When considering pick-ups and drop-offs separately, we differentiate the *daily pick-up pattern* and the *daily drop-off pattern*.



(a) Mean annual usage by terminal

(b) Histogram of mean annual usage of terminals

Figure 4.2.2: Mean annual usage per terminal

Exploratory analysis: Spatial variation. The total usage varies greatly across terminals, with those closer to the centre of Washington D.C. being more popular on average. Figure 4.2.2a visualises this idea by indicating the mean annual usage per terminal across the region. The most popular terminals observe more than 130,000 uses per year, whereas the least popular terminals observe less than one on average.

Over half of the terminals (51%) recorded fewer than 5,000 pick-up or drop-off events per year. To indicate the distribution, Figure 4.2.2b provides the mean annual usage per terminal in a histogram.

Exploratory analysis: Temporal variation. In addition to daily usage patterns varying across space, there is also significant temporal variation. Figure 4.2.3 shows that there are significantly different mean usage patterns and *inter-daily variance* for different days, months, and, to some extent, years. Here, we define inter-daily variance as the daily variability in the usage at a terminal at a given hour of the day.

4.3 Modelling baseline temporal usage patterns

As discussed in Section 4.2, usage patterns vary across time and space. If we do not first remove temporal patterns observed in baseline demand, any outlier detection procedure will likely simply detect baseline trend characteristics as outliers. For example, there is a much higher level of variability in demand on weekends in summer. If we failed to account for this before performing the outlier detection procedure, many of the detected outliers would occur on Saturdays in the summer months. By first accounting for known temporal patterns, the detected outliers are more likely to be genuine outliers rather than explainable patterns.

Similarly, if we do not account for spatial baseline variability and instead aggregate data across all terminals then we will simply detect unused or extremely busy terminals as outliers. Conversely, if we assume all terminals behave independently, then the increased noise makes it more difficult to detect outlying usage patterns. As

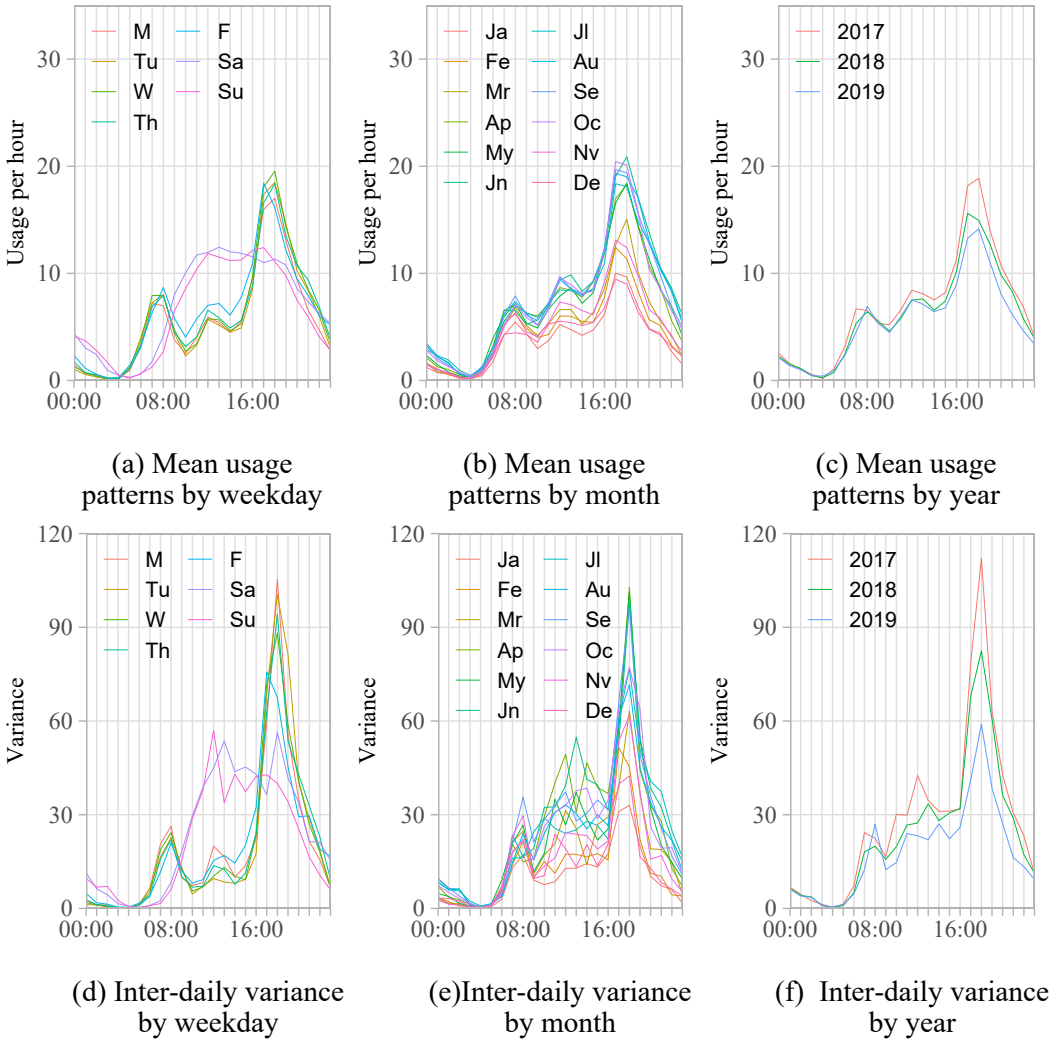


Figure 4.2.3: Mean usage patterns and inter-daily variance for terminal 31203 by hour of day, which is a representative pattern as seen across the network

such, before implementing the outlier detection procedure outlined later in Section 4.5, we carry out a two-step process to (i) remove known temporal patterns; and (ii) spatially cluster terminals which behave similarly. These two steps are key in identifying meaningful outliers, as we shall show.

4.3.1 Background: Bike-sharing demand forecasting

Within the bike-sharing literature, a range of techniques have been considered to predict demand, both spatially and temporally. Zhou et al. (2018) apply a Markov Chain based model to predict daily pick-ups and drop-offs at each station within the Zhongshan City bike-sharing system. The problem of predicting demand in the presence of spatial heterogeneity is further considered by Gao et al. (2021) who estimate a distance decay function and then use multiple linear regression to predict temporal demand in the dockless bike-sharing system in Shanghai. Dockless bike-sharing systems are also discussed by Xu et al. (2018), who use long short-term memory neural networks to predict demand, and capture the spatial and temporal imbalance in usage. Sohrabi and Ermagun (2021) use a combination of pattern recognition on historic data traffic patterns and K-nearest neighbours to make spatiotemporal demand predictions over short time horizons (between 15 minutes and 4 hours), for the Capital Bikeshare data.

The choice of forecasting approach will likely affect the outcome of outlier detection. In the following, we discuss and apply two methods of predicting the baseline temporal patterns in the data: (i) functional regression to account for changes in mean; and (ii) temporal partitioning to account for changes in variance.

Beyond the model presented here, alternative approaches could be used to account for trend and seasonality, and establishing a baseline for bike-sharing demand. In general, any forecasting or modelling approach from which residuals can be obtained could be used instead. After the temporal patterns have been accounted for and

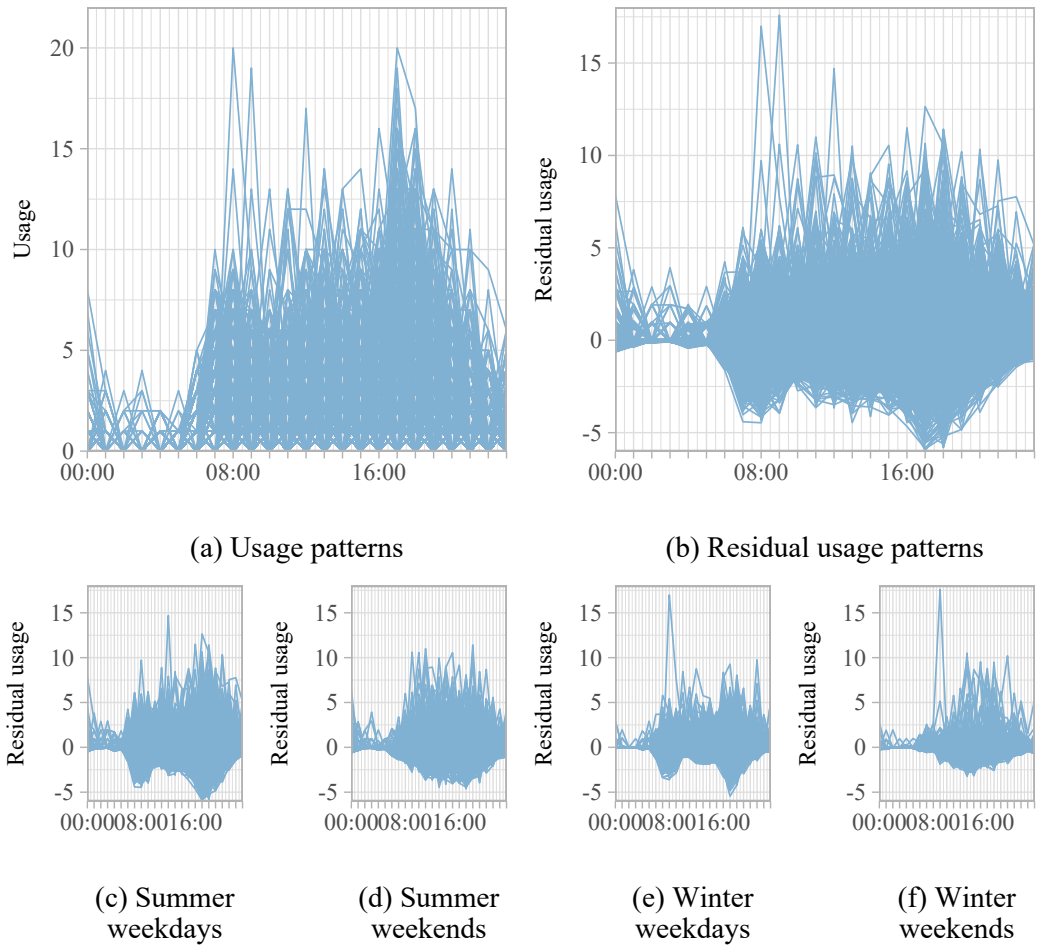


Figure 4.3.1: Residual usage patterns for terminal 31005

the residuals obtained, we are then able to analyse the spatial correlations to group together terminals which deviate from the baseline demand forecast in a similar way, as we shall discuss later in Section 4.4.

4.3.2 Functional regression.

Mean daily usage patterns differ systematically across days of the week, months, and years. We apply a functional regression model (Ramsay et al., 2009) to remove the different mean patterns. We will demonstrate this process on the daily usage patterns

of terminal 31005 as shown in Figure 4.3.1a.

Let $x_{ns}(t)$ be the usage pattern for day n for terminal s. We implement the following functional regression model:

$$\begin{aligned} x_{n,s}(t) = \beta_{0,s}(t) + \underbrace{\beta_{1,s}(t)\mathbb{1}_{Mon_n} + \beta_{2,s}(t)\mathbb{1}_{Tue_n} + \beta_{3,s}(t)\mathbb{1}_{Wed_n} + \beta_{4,s}(t)\mathbb{1}_{Thu_n} + \beta_{5,s}(t)\mathbb{1}_{Fri_n} + \beta_{6,s}(t)\mathbb{1}_{Sat_n} +}_{\text{Day}} \\ \beta_{7,s}(t)\mathbb{1}_{Jan_n} + \beta_{8,s}(t)\mathbb{1}_{Feb_n} + \beta_{9,s}(t)\mathbb{1}_{Mar_n} + \\ \beta_{10,s}(t)\mathbb{1}_{Apr_n} + \beta_{11,s}(t)\mathbb{1}_{May_n} + \beta_{12,s}(t)\mathbb{1}_{Jun_n} + \\ \beta_{13,s}(t)\mathbb{1}_{Jul_n} + \beta_{14,s}(t)\mathbb{1}_{Aug_{nl}} + \beta_{15,s}(t)\mathbb{1}_{Sep_n} + \\ \underbrace{\beta_{16,s}(t)\mathbb{1}_{Oct_n} + \beta_{17,s}(t)\mathbb{1}_{Nov_n} +}_{\text{Month}} \\ \underbrace{\beta_{18,s}(t)\mathbb{1}_{2017_n} + \beta_{19,s}(t)\mathbb{1}_{2018_n}}_{\text{Year}} + e_{n,s}(t). \end{aligned} \tag{4.3.1}$$

where e.g., $\mathbb{1}_{Mon_n} = 1$ if day n relates to a Monday, 0 otherwise. Here, $\beta_{0,s}(t)$ represents the mean usage pattern for a Sunday in December 2019. Appendix C.1.2 contains details of the model selection process where we consider the significance of each of the factors (day, month, year) for a range of terminals. The vast majority of terminals select the full model containing all three factors as the best-fitting model.

As Figure 4.3.1b indicates, the core of the distribution of residuals is symmetric around 0 as desired. The majority of "spikes" in usage are caused by increased demand, resulting in a slight positive skew to the residual patterns. We note that the variance of these residuals is clearly not constant over time and we shall discuss this shortly. Further discussion of the residual distribution is included in Appendix C.1.3. Other features of the usage patterns including positive skew, and inter-daily

correlation are discussed in Appendices C.1.4 and C.1.5.

4.3.3 Temporal partitioning.

Our functional regression approach accounts for different mean usage patterns, but it does not account for the differing inter-daily variances. The simplest option to obtain a data set with homogeneous inter-daily variance is to temporally partition the full data set. While we could partition based on each weekday, month, and year, this would result in around 4 observations per partition - an insufficient number to inform outlier detection. In deciding how to partition the data, there is a trade-off between having reasonably constant inter-daily variance within each group and ensuring there is enough data within each group in order to establish patterns. Therefore, we group together days, months, and years where the inter-daily variances are sufficiently similar.

From Figure 4.2.3, it is clear that weekdays (Mon-Fri) are similar to each other, and weekends (Sat-Sun) are also similar to each other. The differences between the inter-daily variance across different months is less clear. Defining *summer* as April through to October, then months within summer exhibit similar inter-daily variance patterns, as do months within winter (November to March). Further analysis of the variance in Appendix C.1.1 supports this partitioning. All years are grouped together. This results in four partitions: (i) summer weekdays, (ii) winter weekdays, (iii) summer weekends, and (iv) winter weekends, as displayed in Figure 4.3.1c–f. Note that we do not attempt to remove the *intra*-daily variability of these residuals with further parametric modelling, as instead we turn to functional data analysis to detect

outlying curves from these residual daily usage patterns.

The choice of partition is important and should reflect the choices made in the planning process, e.g. with regard to inventory redistribution. If the increased inter-daily variance on weekends, for example, is already known and accounted for in planning, such that there are different schedules for redistribution, then partitioning as we propose would be appropriate. However, if the general planning process (including demand forecast and inventory optimisation) assumes uniformity across all days of the week, it would then be informative to do the same in the outlier detection to flag the weekend effect when it occurs.

4.4 Clustering terminals by spatial usage patterns

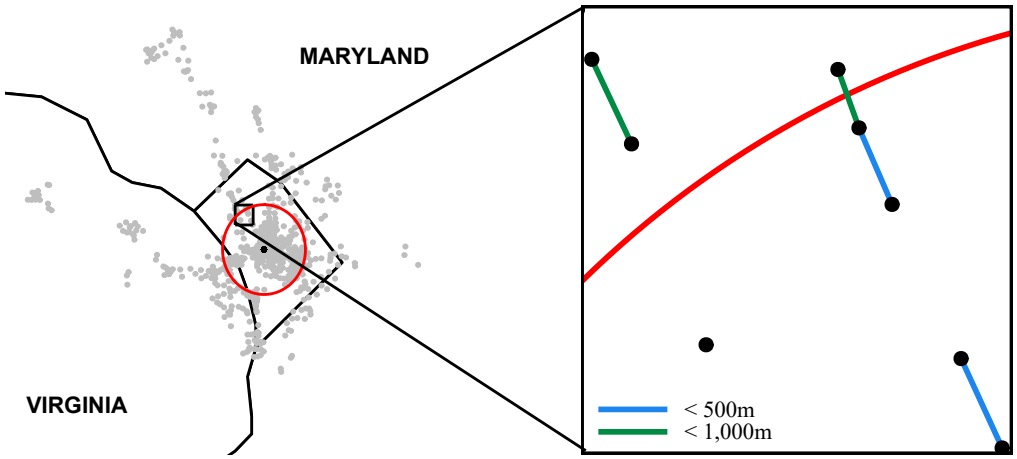
When outlier demand is driven by factors such as regional events or weather, we expect it to affect more than a single, isolated terminal. At the same time, we cannot assume that all terminals experience outlier demand at the same time and in the same way. Therefore, we first cluster the terminals such that those in the same cluster are likely to experience similar effects from demand outliers.

We propose a two-stage process to determine which terminals should be clustered. First, we construct a graph based on the geographic co-ordinates of the terminals to determine which terminals are permitted to be in the same cluster based on geographic distance. Secondly, we follow an idea from Zahn (1971) who suggests the removal of edges from a graph's minimum spanning tree (MST) as a method of finding clusters of nodes.

The first step of constructing the graph is non-trivial. Graph construction in the bike-sharing setting is more open-ended than in situations where mobility networks rely on established legs, as, e.g., in the railway application studied in Rennie et al. (2021b). For bike-sharing, direct journeys between any two terminals are possible, so that in theory, all terminals could be vertices in a fully connected graph. Although we could simply add edges between every node i.e. a complete graph, there are two reasons for not doing so: (i) For the purposes of aiding planning, we do not want two terminals which are geographically far apart to be in the same cluster if no terminals in between are similar to both. (ii) The algorithm used to compute the MST is slowed down by an increased number of edges, and due to the greedy nature of it, we are more likely to end up at a non-optimal solution if we add in extraneous edges.

4.4.1 Graph construction from geographical distance.

We first construct a graph where the nodes represent the terminals and the edges indicate which terminals are permitted to be in the same cluster. This approach implicitly assumes that similarity of usage is driven by the terminals' geographical proximity. That is, if two terminals are close together, potential customers are more likely to treat them as interchangeable, causing similar usage patterns. In the Capital Bikeshare data set, terminals are more densely distributed in the centre of D.C., so that customers can choose from a large variety of terminals. We expect this to render them more sensitive to distance, such that they are less willing to travel to a more distant terminal. Therefore, we use different criteria to add an edge between terminals depending on how close to the centre of D.C. those terminals are.



(a) Radius for defining edges (b) Construction of graph

Figure 4.4.1: Graph construction when $R = 5000m$, $D_{inner} = 500m$, and $D_{outer} = 1,000m$.

To identify the dense city-centre, we establish a circle around the centre of D.C. of radius R, with the median co-ordinates of all terminals as the centre, as shown in Figure 4.4.1a with $R = 5000m$. We add an edge between terminals i and j if: (i) both terminals lie inside the radius, and are less than D_{inner} apart; or (ii) one or both terminals lie outside the radius R, and are less than D_{outer} apart.

Not all terminals that are geographically close exhibit similar usage patterns e.g. due to proximity to railway stations. Therefore, to quantify how similar the usage patterns of two terminals are, we add weights to the edges of the graph. For each edge between terminals i and j, we also compute an edge weight representing the dissimilarity between the usage patterns for those terminals. The edge weights are given by:

$$w(i,j) = 1 - \rho(i,j), \tag{4.4.1}$$

where $\rho(i, j)$ is the average functional dynamical correlation (Dubin and Müller, 2005) between the daily usage patterns for terminals i and j. Here, the average correlation is based on the correlations between daily usage patterns across the entire time period considered (2017 - 2019), as there is no evidence of the clusters changing over time. However, if the correlations (and therefore clusters) are changing over time, a moving window approach could be used to update the average correlation and clusters over time.

4.4.2 Minimum spanning tree clustering.

We apply a minimum spanning tree approach to cluster terminals that are connected in the geographical proximity graph. A graph's *spanning tree* is a subgraph that includes all vertices in the original graph and a minimum number of edges, such that the spanning tree is connected. If the original graph is disconnected, we compute a spanning tree for each component – termed a *spanning forest*. A *minimum spanning tree (MST)* is the spanning tree with the minimum sum of edge weights. Since the graph is weighted, we use Prim's algorithm (Prim, 1957) to calculate the MST.

To obtain the clusters from the MST, we set a threshold, ρ_τ, for the correlation and remove all edges with weights above $1 - \rho_\tau$.

4.4.3 Clustering results: daily usage patterns

Figure 4.4.2 visualises the outcome from four different values of ρ_τ. These values of ρ_τ are chosen to illustrate the clustering for two reasons: (i) $\rho_\tau = -1$ indicates that all

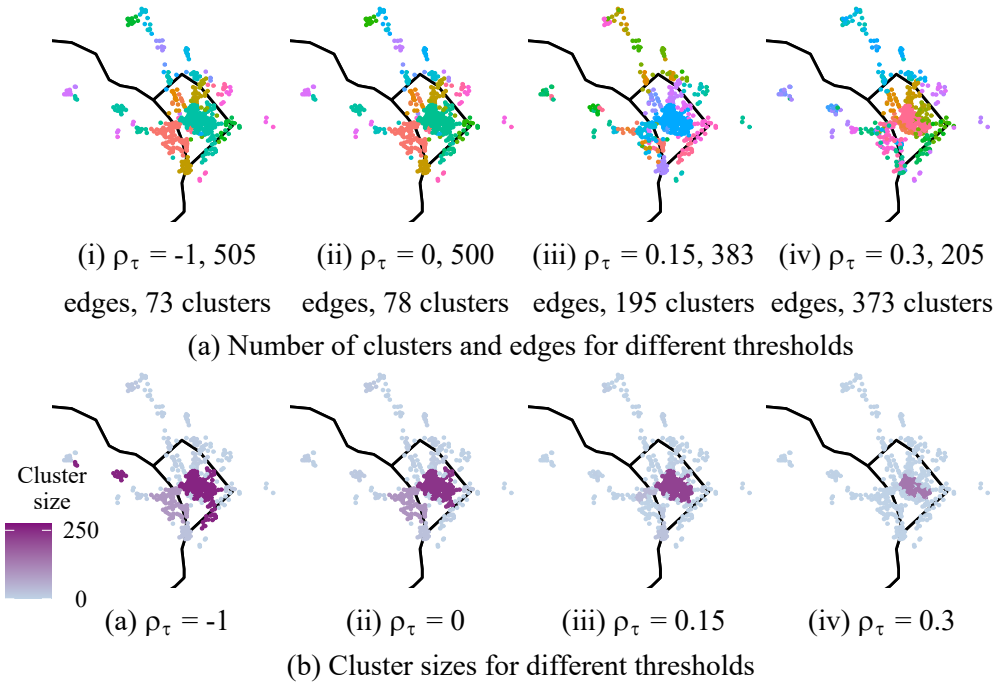


Figure 4.4.2: Clustering of terminals under different values of ρ_τ.

initially connected edges stay in place. In fact, the minimum correlation observed is -0.57, and any threshold between -1 and -0.57 results in all edges of the MST remaining in place. (ii) 90% of the observed correlations lie between 0 and 0.3, therefore the values of $\rho_\tau = 0$, 0.15, and 0.3 demonstrate the clustering when the threshold is close to the minimum, mean, and maximum correlations.

Figure 4.4.2 can be used by analysts to determine the most appropriate threshold, depending on which aspect of planning they are considering. Figure 4.4.2a shows which terminals are clustered together. If an analyst has expert knowledge regarding which terminals are likely to behave similarly, they can cross-check with the clustering and choose the threshold which supports this decision. Figure 4.4.2b visualises

the sizes the clusters that each terminal belongs to, demonstrating the non-uniform distribution of cluster size across the geographic area. This can also be used to determine an appropriate threshold. For example, if an analyst is interested in the general demand patterns of central D.C., they can choose a threshold that highlights all of central D.C. in a single large cluster e.g. ρ_τ=0. In contrast, if the analyst is more interested in obtaining clusters of similar size, Figure 4.4.2b(iv) would guide them towards a higher threshold.

Across all thresholds, the terminals closer to the centre of D.C. form a larger cluster, with those further away from the centre branching into smaller clusters. Clearly, the choice of threshold values ρ_τ impacts the precise clustering results.

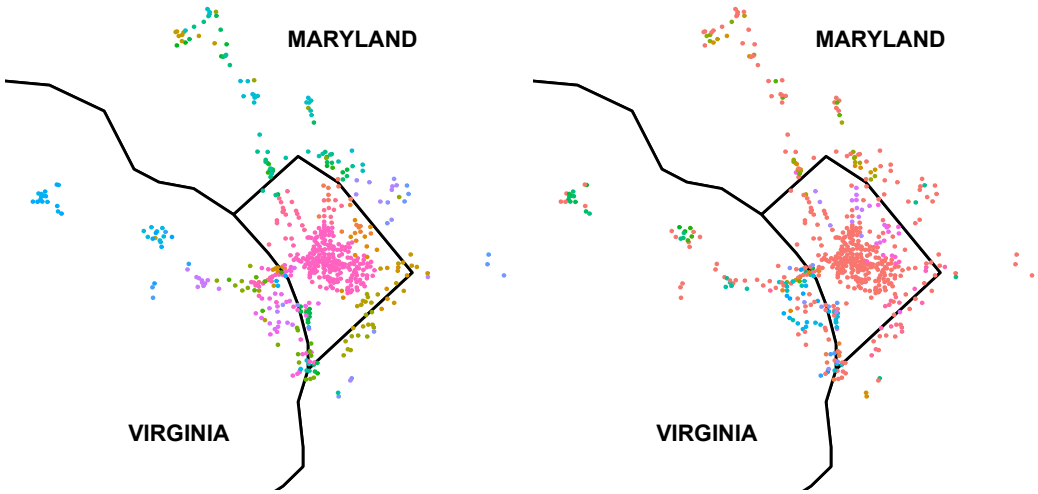
The distance parameters, $\{R, D_{inner}, D_{outer}\}$, also affect clustering. The number of clusters increases as ρ_τ or R is increased, whereas increasing D_{inner} or D_{outer} has the opposite effect. There is an inverse relationship between the number of clusters and the uniformity of cluster size. As the number of clusters increases, individual terminals tend to split off to form their own cluster whilst the majority of terminals remain in the large central cluster, resulting in decreased uniformity of cluster size. For decision-making, clusters of similar sizes are often more informative (compared to a large cluster consisting of most terminals, and the remaining terminals each in their own cluster).

We leave the choice of parameters to analyst input, such that analysts may use their expertise to select appropriate values based on the visualisation and their business case (Vock et al., 2021). For the remainder of this chapter, unless otherwise specified, we set the parameter values as $\rho_\tau = 0.15$, $R = 5000$m, $D_{inner} = 500$m,

and D_{outer} = 1000m. These values are chosen to balance the number of clusters with more similar cluster sizes. Appendix C.2.1 includes further details on the reasoning for these choices.

4.4.4 Clustering results: daily pick-up and drop-off patterns

So far, we have focused on clustering terminals based on the similarity of their daily usage patterns. However, when considering forecasting for inventory rebalancing, differentiating pick-ups and drop-offs is highly important. Depending on the aggregation level of forecasting, it may be desirable to consider separate clusterings for drop-off and pick-up patterns.

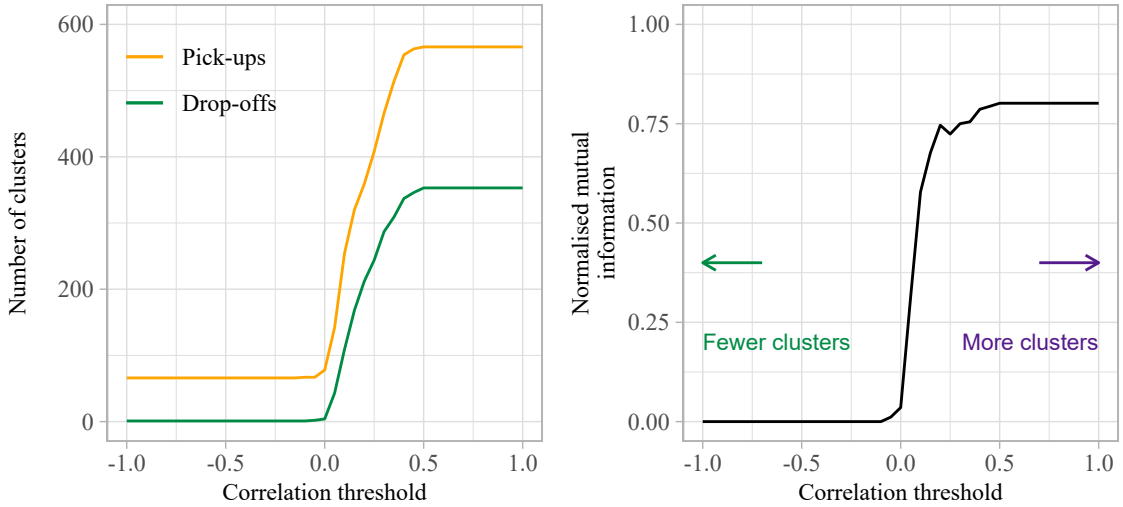


(a) Pick-up pattern terminal clustering resulting in 320 clusters

(b) Drop-off pattern terminal clustering resulting in 168 clusters

Figure 4.4.3: Comparison of clustering terminals based on pick-up and drop-off patterns for ρ_τ=0.15, R = 5000m, D_{inner} = 500m, and D_{outer} = 1000m.

Figure 4.4.4a shows that this increased homogeneity of drop-off patterns is con-



(a) Number of clusters for pick-up and drop-off terminal clustering

(b) NMI for pick-up and drop-off terminal clustering

Figure 4.4.4: Comparison of pick-up and drop-off terminal clustering

sistent across all values of the correlation threshold, ρ_τ. Although the number of clusters resulting from both pick-up and drop-off terminal clustering follows a similar relationship with the correlation threshold – increasing steeply between 0 and 0.4 – the drop-off clustering consistently results in fewer clusters.

To formally compare the output of these two clusterings, we use the *Normalised Mutual Information (NMI)* (Amelio and Pizzuti, 2015). The NMI is 1 if two clusterings are identical, and 0 if they are completely different (see Appendix C.2.2 for details). Figure 4.4.4b shows that the similarity of the pick-up and drop-off clusterings are highly dependent on the correlation threshold. When a low threshold is used, the clusterings are completely different. However, as the correlation threshold increases above 0.25, the clusterings become more similar, achieving an NMI of around 0.75.

This evidence that pick-ups and drop-offs are not spatially homogeneous motivates the need for separate forecasting of the two. The differences across the varying

threshold also indicates that the need for separate forecasting is more critical when considering a larger area i.e. when considering total demand, but is less critical over smaller areas closer to the terminal level. Monitoring the difference in the number of clusters and the similarity of the two clusterings can help analysts to decide on the level of forecasting. Analysts could also examine changes in the NMI over time for a given correlation threshold. For example, if the pick-up and drop-off clusterings are becoming more similar to each other over time, this could indicate increasing levels of homogeneity in pick-up patterns.

4.5 Detecting outliers within a cluster of terminals

To demonstrate the outlier detection procedure, we focus on one of the resulting clusters. The nine terminals in the cluster we consider are highlighted in green in Figure 4.5.1.

Figure 4.5.1 demonstrates how the currently analysed cluster may be highlighted for analysts. On the one hand, the location of the cluster within the D.C. area can provide contextual information for analysts in the search for an explanation of outlier demand. On the other hand, the zoomed in section on the right shows how the terminals relate to one another within the cluster. This could be useful if the outlier demand is not detected in all terminals. For example, all but one of the green cluster terminals lie in a relatively straight line. If the terminal which lies to the North East of the main group of terminals in the cluster behaves differently, analysts can look to nearby clusters for further information.

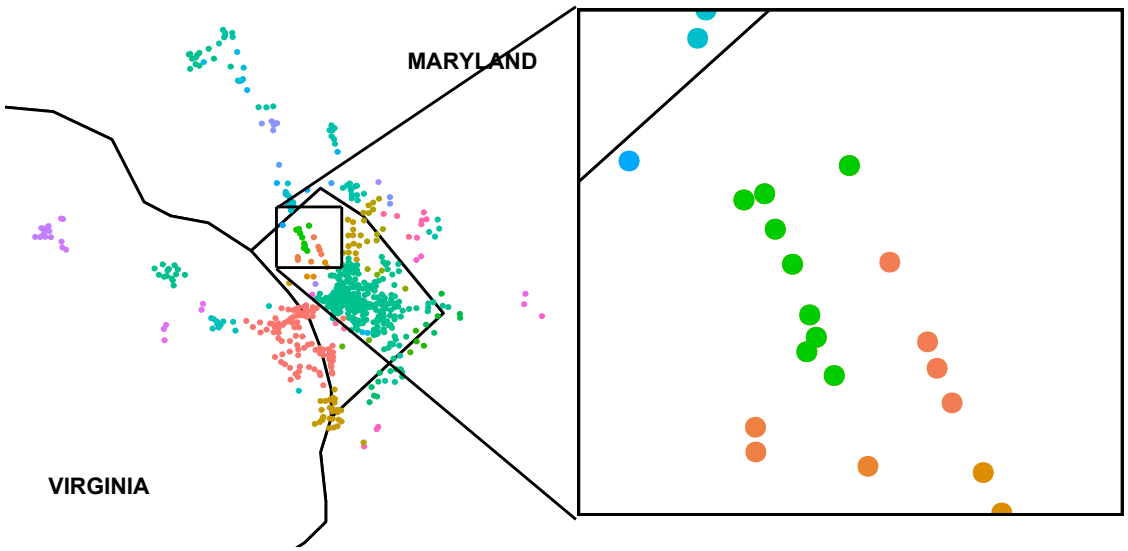
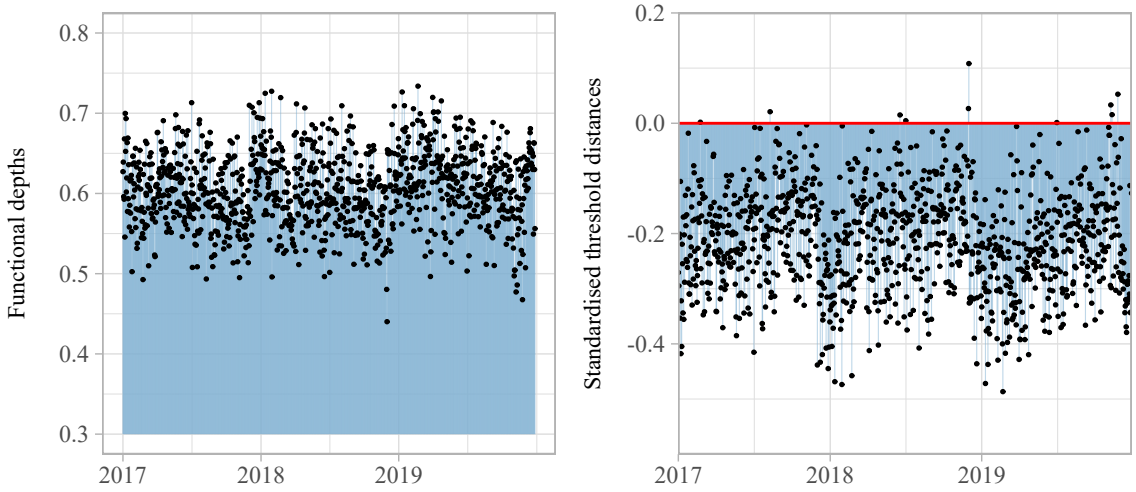


Figure 4.5.1: Cluster chosen for further investigation

To identify outlier demand in usage patterns, we use the notion of *statistical depth*. In statistics, depth provides an ordering of observations, where those near the centre of the distribution have higher depth and those far from the centre have lower depth. In the case where each observation is a time series of usage throughout the day, the *functional depth* can measure how close to the central trajectory, i.e. median usage pattern, each daily usage pattern is. Therefore, to measure the outlyingness of each daily usage pattern, we calculate its functional depth (with respect to other daily usage patterns that lie in the same partition of data). Days whose usage pattern has lower functional depth are more outlying. In particular, if the depth is below some threshold, we classify the day as an outlier.

For each partition of data, p, and for each terminal s, we calculate a threshold, $C_{s,p}$, for the functional depth as per Febrero et al. (2008). To calculate the threshold, we (i) resample the daily usage patterns with probability proportional to their functional

depths (such that any usage patterns affected by outlier demand are less likely to be resampled), (ii) smooth the resampled patterns, and (iii) sets the threshold $C_{s,p}$ as the median of the 1^{st} percentile of the functional depths of the resampled patterns.



(a) Functional depths

(b) Normalised threshold exceedances

Figure 4.5.2: Normalisation of the functional depths, exemplified for terminal 31316 where there are two partitions of data (summer/winter)

Let $d_{n,s,p}$ be the functional depth for day n (which occurs in partition p) for terminal s. We then transform the functional depths to lie between 0 and 1 such that they are comparable between different terminals and aggregated over the different partitions of data. Define $z_{n,s}$ to be the normalised functional depth on day n for terminal s:

$$z_{n,s} = \sum_{p=1}^{P} \left(\mathbb{1}_{n \in p} \left(\frac{C_{s,p} - d_{n,s,p}}{C_{s,p}} \right) \right). \tag{4.5.1}$$

The functional depths for terminal 31303 are shown in Figure 4.5.2a, and their normalised counterparts in Figure 4.5.2b. Figure 4.5.2a provides a way to check for unaccounted for trend and seasonality in the usage patterns. However, much like

univariate regression residuals which can be used to visually identify residuals patterns, the functional depths should appear random with no obvious patterns. If an analyst can identify a pattern in the functional depths, this would suggest that the forecasting model may need to be reconsidered. Weekdays could be highlighted in different colours to help identify temporal patterns on a smaller scale. Figure 4.5.2b can also be used by analysts to check how many non-outlier days are close to, but do not exceed, the threshold. Analysts can use this information to manually vary the threshold to detect further outliers they perceive to be false negatives.

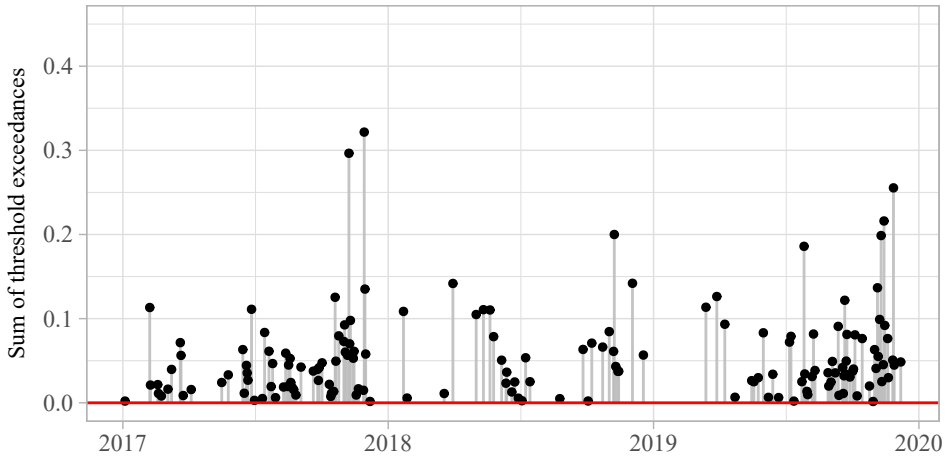


Figure 4.5.3: Sum of threshold exceedances, z_n

We then define the sum of threshold exceedances across all S terminals in the cluster to be:

$$z_n = \sum_{s=1}^{S} z_{n,s} \mathbb{1}_{\{z_{ns}>0\}}. \tag{4.5.2}$$

The values of z_n for this cluster are shown in Figure 4.5.3. The value of z_n is only positive for days which have been classified as an outlier.

4.5.1 Computing outlier severity

Although the values of z_n give an indication of how severe the outlier is (with z_n being larger if the magnitude of the outlier demand is larger, or if it affects a larger number of terminals), we wish to make the severity easier to interpret across different clusters. Therefore, we fit a distribution to the sum of threshold exceedances and use the non-exceedance probability given by the CDF of the distribution as a measure of severity.

In contrast to Rennie et al. (2021b) who fit a generalised Pareto distribution (GPD) to the sum of threshold exceedances, here we fit a four-parameter Beta distribution (Carpenter and Mishra, 2001). For a GPD, assuming the shape parameter is non-negative, the support has no upper bound. In this application the upper bound is finite and known to be equal to the number of terminals within the cluster. Since $z_{n,s}$ lies between 0 and 1, the sum across S terminals must lie between 0 and S. Therefore, a four parameter Beta distribution, bounded on $(0, S)$ is likely to provide a better fit – see Figure 4.5.4.

The severity of an outlier on day n, θ_n, is therefore given by the CDF of the four parameter Beta distribution:

$$\theta_n = F_{(\alpha,\beta,a,c)}(z_n) = \int_0^{z_n} \frac{(q)^{\alpha-1}(S-q)^{\beta-1}}{B(\alpha,\beta)S^{\alpha+\beta+1}} dq, \tag{4.5.3}$$

where $B(\alpha, \beta)$ is the Beta function. This results in an outlier severity, between 0 and 1, for each cluster on each day, as exemplified in Table 4.5.1.

We differentiate *positive* and *negative* outliers. Positive outliers are primarily caused by increased usage i.e. where the sum of the functional residual is greater

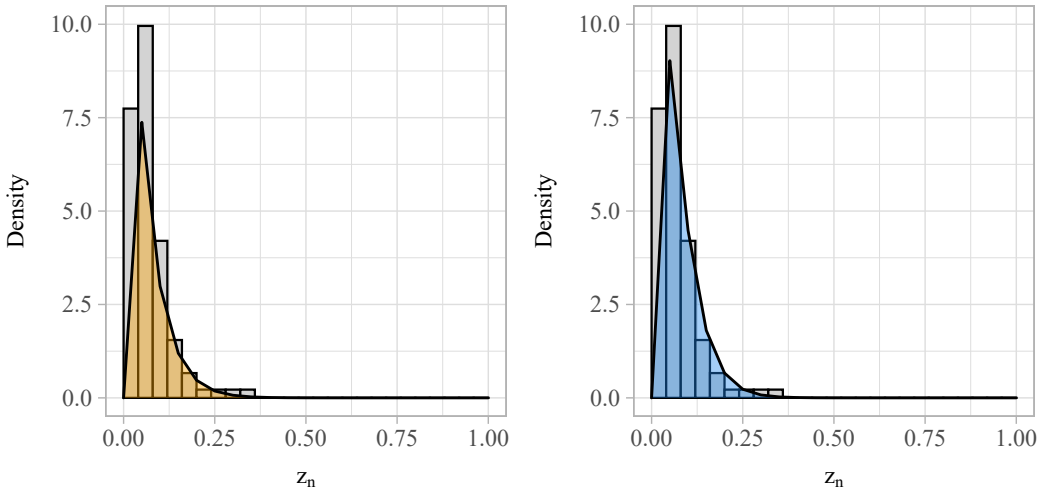


Figure 4.5.4: Comparison of fitted distributions

than zero, whereas negative outliers indicate a shortfall in usage.

Date	Cluster 1	Cluster 2	Cluster 3	...
11/01/2017	0.033 ↓	-	-	...
12/01/2017	0.852 ↑	0.720 ↑	-	...
⋮	⋮	⋮	⋮	⋱

Table 4.5.1: Examples of outlier severities for different days, where arrows indicate positive or negative outliers

4.5.2 Visualising detected outliers for analysts

There are multiple different visualisations that could be used to present the information from Table 4.5.1 to analysts.

To support on-the-day forecast adjustments, the simplest approach is as a ranked

alert list, as exemplified by Table 4.5.2. By presenting the alert list as a table rather than a visualisation, this gives a clear, prioritised list of tasks to complete. Here, the different colours show where the outlier was detected: red showing the terminals where the outlier was detected, and blue showing other terminals in the same cluster likely to be affected. By displaying the severity alongside the ranking, analysts are better able to prioritise their adjustments. For example, an analyst may choose to only adjust the forecasts for the top two outliers in Table 4.5.2, as the third has a much lower severity in comparison.

Rank	Severity	Direction of change	Cluster
1	0.892	↑	31104, 31115, 31129, 31217, 31219, 31222, ...
2	0.828	↑	32008, 32048
3	0.347	↑	31000, 31001, 31002, 31003, 31004, 31005, ...
⋮	⋮	⋮	⋮

Table 4.5.2: Example of ranked alert list for 30/03/2018

To give a wider view of how outliers have occurred and to account for them in the historic data, the severities can be visualised in a spatiotemporal heatmap as exemplified in Figure 4.5.5. This figure shows the severity of detected outliers over time for every cluster, where clusters are arranged from left to right by nearest to furthest from the centre of Washington D.C. The order of the clusters along the x-axis could also be arranged to highlight further spatial patterns e.g. from North to South. This type of visualisation can help to identify large-scale patterns in the

outliers. For example, knowing for which times of year or days of the week outliers are more likely to be detected can help determining the number of required analysts. Similarly, different analysts may be assigned to monitor different clusters, and this helps to identify which clusters may need more input from analysts.

Figure 4.5.5: Outlier severity for each cluster between 2017 and 2019

4.6 Discussion

In this section, we discuss patterns in the outliers detected in the Capital Bikeshare data, and suggest potential explanations for their causes.

Figure 4.6.1 visualises the number of positive and negative outliers over the days of the time span recorded in the data set. By visualising the positive and negative outliers jointly, analysts can immediately see that (i) negative outliers typically affect

far fewer clusters than positive outliers; (ii) the spikes where outlier demand affects a large number of clusters do not occur at the same time for positive and negative outliers; and (iii) the seasonal patterns in the detected outliers are not the same for positive and negative outliers. This can aid analysts in their predictions of outliers: if an outlier is detected in winter – it is more likely to be negative, if detected in summer – more likely to be positive.

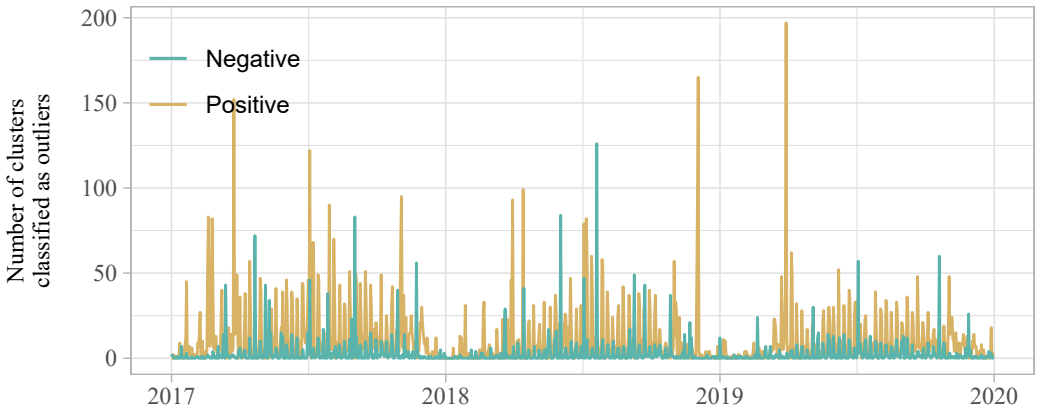


Figure 4.6.1: Positive and negative outliers

Outliers occur independently in different clusters. In fact, only four days observe outliers in more than 125 of the 195 clusters – one negative and three positive. The three positive outliers occur on 25 March 2017, 3 December 2018, and 30 March 2019. Explanations from events arise for two of these dates: 3 December 2018 relates to the funeral of George H. W. Bush, and a NATO protest occurred in Washington D.C. on 30 March 2019. 25 March 2017 and 30 March 2019 were both warm days, and the last Saturday in March - perhaps suggesting that the definition of *summer* should be from the last Saturday in March, rather than April 1.

It is interesting to note that the next most widespread positive outlier relates to Independence Day in 2017. Independence Day was detected as a positive outlier in 123, 80, and 35 clusters in 2017, 2018, and 2019. However it was detected as a negative outlier in 46, 47, and 57 terminals respectively. The date of the widespread negative outlier is 21 July 2018 which relates to one of the worst storms Washington D.C. has seen. Further discussion of how weather is related to outliers can be found in Section 4.6.2.

4.6.1 Spatiotemporal patterns in detected outliers

In this section, we analyse the detected outliers and consider spatial and temporal patterns within the outliers.

Temporal patterns. Even after accounting for the lower means and reduced inter-daily variance of the winter months, we detect fewer outliers in winter (indicated by the two horizontal white bars in Figure 4.5.5). Otherwise, we find no clear systematic temporal patterns to the detected outliers. Appendix C.3.1 provides additional discussion on the temporal aspects of the detected outliers, including the visible temporal patterns in the outliers when we fail to account for temporal patterns in the forecasting step.

However, Figure 4.6.2 shows that although outliers can sometimes occur as one-off events, they are also quite likely to occur in temporal clusters. Therefore, once an outlier has been identified, the information can be used to support adjustments to planning in the subsequent days.

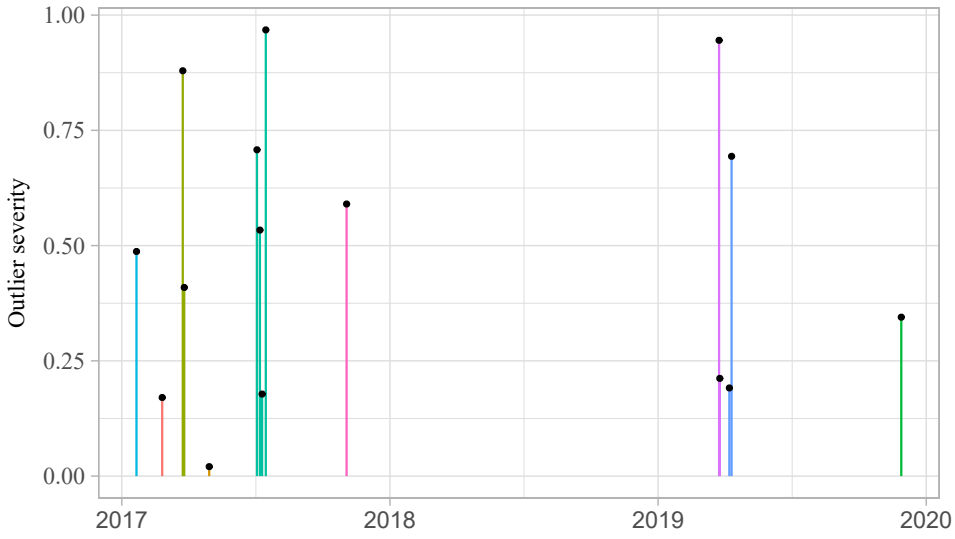


Figure 4.6.2: Exemplified severity for outliers detected in one cluster, showing temporal clustering of outliers

Spatial patterns. Next, we discuss spatial patterns in the detected outliers and consider the relationship between outliers in pick-up and drop-off usage patterns. Figure 4.5.5 shows that the cluster which is formed around the centre of Washington D.C. (indicated by the first column on the left) experiences more frequent and higher probability outliers. Otherwise, there is little geographic pattern to how often outliers occur in terms of distance from the centre.

Two other clusters besides the central D.C. cluster exhibit a higher number of outliers with higher severity than other clusters. Figure 4.5.5 indicates these by darker vertical lines. These clusters are highlighted in Figure 4.6.3. These clusters are both fairly close to the centre of Washington D.C., and are close by the two main bridges across the Potomac River into the centre. The terminals in these clusters are also situated close to The Pentagon, Arlington National Cemetery, and Ronald

Reagan Washington National Airport. Therefore these clusters are likely to experience business commuter demand, tourist demand, and potentially also airport travellers i.e. have multiple sources of outlier demand.

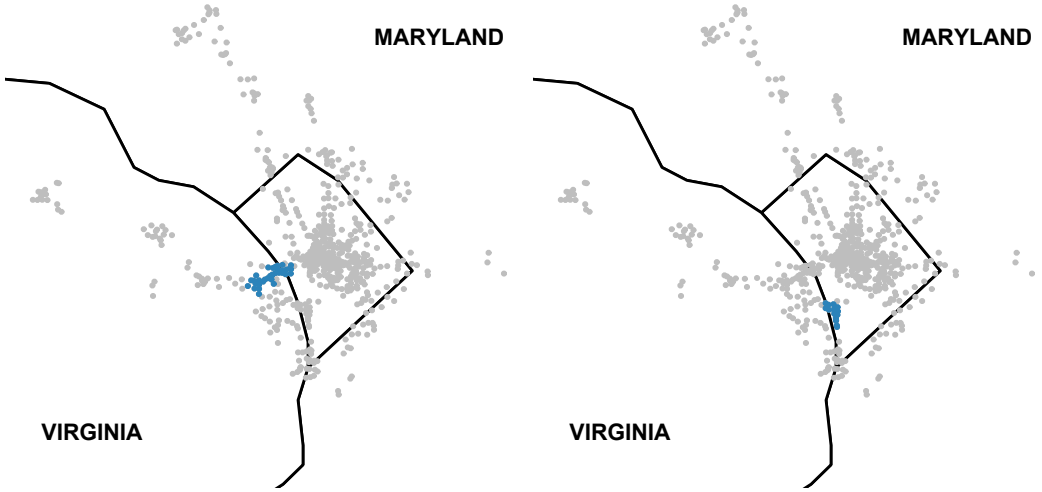


Figure 4.6.3: Two (non-central D.C.) clusters which exhibit higher numbers of outliers.

We also consider the frequency of outliers on the terminal level. Figure 4.6.4 shows the number of days that each individual terminal was classified as an outlier between 2017 and 2019. Terminals where no outliers were detected are not shown. Outliers are more commonly detected in terminals nearer the centre of D.C.

We analyse the differences in the spatial patterns of the outliers detected in pick-up and drop-off usage patterns. For this, we use the clustering based on the overall usage patterns as it allows direct comparison of outliers in different clusters. Subsequently, we apply the outlier detection procedure separately to the pick-up and drop-off usage patterns. This enables us to isolate how the detected outliers and their severities

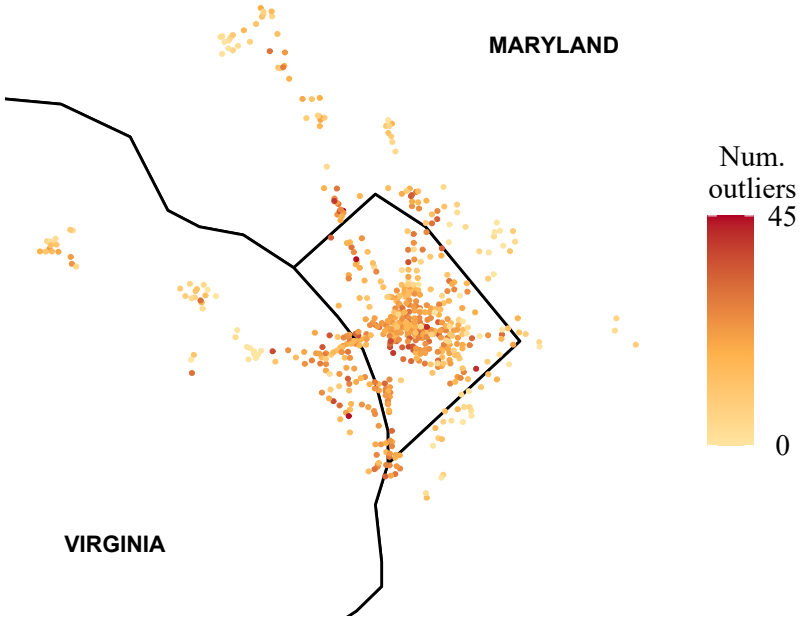


Figure 4.6.4: Number of days each terminal was classified as an outlier between 2017-2019

change when the terminals in each cluster remain constant.

To formally compare the output of the outlier detection procedure for pick-up and drop-off usage patterns, we use cosine similarity (Leydesdorff, 2005). That is, the cosine of the angle between two vectors, where 0 represents complete dissimilarity, and 1 complete similarity. Figure 4.6.5a provides the cosine similarity between clusters i.e. the cosine similarity of the vector of outlier severities for those detected in pick-up terminals over the three year period, and that for drop-off terminals.

Figure 4.6.5a shows that outliers detected in pick-up and drop-off terminals are fairly similar, although this changes depending on the correlation threshold used in the clustering step. As the correlation threshold ranges from 0 to 0.3, the average cosine similarity ranges from 0.69 to 0.44. As the correlation threshold increases, the

Figure 4.6.5: Cosine similarity between outliers detected in pick-up and drop-off usage patterns under different correlation thresholds

number of clusters increases. Therefore, when we look at outliers on a small cluster or terminal level, there is less similarity between pick-ups and drop-offs. However, when the clusters are larger and the outliers aggregated, there is a clearer pattern between pick-ups and drop-offs. Further, this similarity is not uniform across the different clusters - those closer to the centre of D.C. have a higher cosine similarity. That is, the closer a cluster is to the centre of D.C., the more likely it is that if a day is a pick-up terminal cluster outlier, it will also be a corresponding drop-off terminal cluster outlier. We did not find any temporal patterns in the comparison of pick-up and drop-off outliers.

4.6.2 Weather as an explanatory factor for demand outliers

It is widely acknowledged that weather can be used as a predictor for average bike rental demand (Lin et al., 2020). Therefore, we examine whether extreme temperature or rainfall drive extreme changes in demand i.e. outliers. To that end, we analyse weather data obtained from Visual Crossing (2021) and investigate whether weather can be used to explain and eventually predict the outliers in demand. The data is included in Appendix C.3.2.

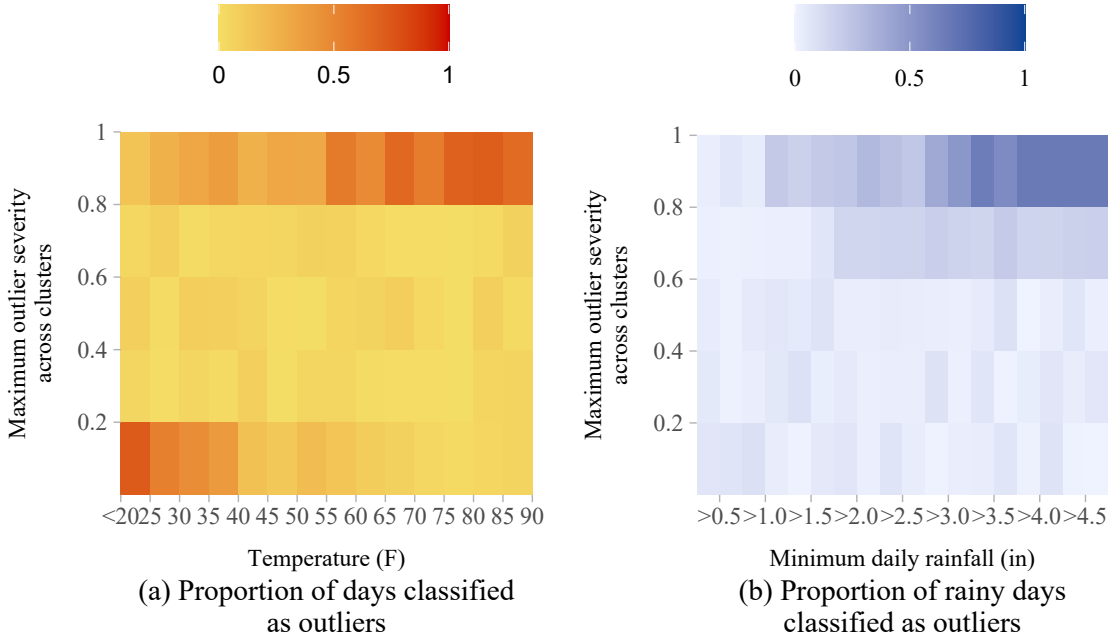


Figure 4.6.6: Severity of outliers at different temperatures and precipitation levels

Figure 4.6.6a shows the proportion of days in each temperature range that have a maximum outlier severity within each severity range. Higher temperatures (between 70 and 90 °F) result in higher severity outliers, indicated by the red region in the top right of Figure 4.6.6a. The red region in the bottom left shows that a high proportion of days with a very low average daily temperatures ($\leq$ 25 °F) are classified as outliers.

However, these outliers typically have a low severity. This can be explained by these outliers being negative demand outliers – and low temperatures typically occur in winter, when demand is already low.

In addition to temperature, we also expect precipitation levels to affect demand for bike-sharing. As we expect increased rainfall to have a negative impact of usage, we consider only the severity of negative demand outliers here. Figure 4.6.6b shows the proportion of days with a minimal level of precipitation which were classified as outliers with some severity. Higher rainfall generally results in higher likelihood of the day being classed as a negative demand outlier. When precipitation levels are especially high, the outliers that are detected also tend to have higher severities.

There are likely many other factors which cause and influence outlier demand in the Capital Bikeshare network. For example, Ma et al. (2015) have previously linked usage of bike-sharing terminals to usage of Metrorail services. We anticipate that cancellations, or short-term changes to Metrorail services may also generate outlier demand for bike-sharing. However, due to lack of available of data on such cancellations, we leave this to future research.

4.7 Conclusion

In this chapter, we identified temporal patterns in the Capital Bikeshare data set and applied a combination of functional regression and temporal partitioning to remove such trends and obtain a homogeneous data set. We also accounted for spatial patterns in temporal usage by clustering together similar terminals. By basing our

clustering algorithm on a combination of geographical knowledge, and similarity of usage patterns, we were able to identify how usage changes as terminals get further away from the city centre. Throughout this chapter, we have presented visualisations to illustrate our findings and provided detailed descriptions of how such visualisations may be used by analysts to aid in their decision making.

Our in-depth study of detected outliers showed that not all terminals are equally prone to outliers - those closest to the centre of D.C. exhibit far more outlying demand. This is also true for known outlier days e.g. national holidays such as July 4, where some clusters of terminals exhibit increased demand and others decreased. For forecasting and planning purposes, this knowledge is highly important since outlier demand changes not only the magnitude of demand, but also the spatial distribution of where customers go. In terms of rebalancing bikes at terminals, this could have a large impact on the efficiency of the schedule. Further, we also showed that outliers are more likely to occur in the summer months (even after accounting for increased usage and usage variability), suggesting rebalancing needs to be more reactive in the summer months.

Our analysis of weather patterns showed that outliers are more likely to occur when the weather conditions are more extreme. Both temperature and precipitation were found to have an impact on demand - with excessively high precipitation or very low temperatures causing negative demand outliers, and high temperatures causing positive demand outliers.

Further research is needed to evaluate the effects that identifying and correcting for outliers may have on revenue and planning in the bike-sharing domain. The method

outlined in this chapter could be used to generate an outlier *alert*, to notify Capital Bikeshare when the rebalancing policy for a given day is non-optimal. Further analysis of how these alerts could be deployed in an automated system, and how they may affect the complexity of the routing problem, is needed.

Chapter 5

Conclusion

In this chapter the thesis is concluded by summarising the contributions it has made to the area of outlier demand detection and quantification in transport systems. Suggestions of further research are also discussed.

5.1 Contributions

The main focus of this thesis was to develop new methods to detect demand outliers in transport networks. As discussed in previous chapters, not accounting for such outliers can have significant impacts on both forecasting and optimisation in demand planning for transport systems. The first contribution of this thesis was a novel outlier detection method, which combines functional data analysis and time series forecasting, to detect outlier demand in bookings for transport systems. This method incorporated an extrapolation step to allow its application in an online setting, where the analysed demand patterns have only been observed over part of the booking horizon. This

chapter also contributed the design of a simulation-based framework to evaluate the effects of outliers in revenue management systems and the performance of the outlier detection method in a controlled environment.

Chapter 3 then extended the approach described in Chapter 2 to a two-step method for detecting outlier demand in a network setting, and contributed an evaluation of its application to railway booking data from Deutsche Bahn. This chapter also contributed an extensive computational study of outlier detection performance and the revenue impacts of outlier demand under different forecast adjustments.

The final contribution of this thesis was an application of the previously described method to bike-sharing systems, where the generalisability of the approach was demonstrated, showing it is applicable either in the setting where capacity is on the edges or vertices of the graph. A particular focus was the communication of outlier alerts to analysts and how outliers may be visualised, alongside the temporal and spatial patterns of demand.

Overall, a key finding of this thesis is that methods from functional data analysis prove to be very powerful at revealing outliers, however the raw data must first be aggregated and modelled appropriately, by e.g. appropriately clustering legs or accounting for seasonality.

5.2 Further Work

In this section possible extensions to the methodology and ideas for further studies are described. Firstly, a method to account for the temporal dependence of outliers.

Secondly, suggestions of further computational studies to further examine the performance in different settings and under a wider variety of actions. Finally, a discussion of issues that may be faced in implementing such a method, and studies that may be undertaken to further evaluate the method's performance.

5.2.1 Modelling temporal dependence of outliers

In this thesis we made an assumption that outliers occur independently, and only allocated an outlier severity to departures which are detected as outliers. As noted in Chapter 4, outliers sometimes occur in temporal clusters. There may also be knock-on effects of outliers onto subsequent (or previous departures) that were not detected as outliers due to the noise in the data. For example, an event may cause an increase in demand for 09:00 and 10:00 departures, but only the 09:00 departure is detected as an outlier. By accounting for these knock-on effects, the detection rate may be improved.

In the approach described in Chapter 3, we set the outlier severity of the n^{th} departure to be the non-exceedance probability:

$$\theta_n = F_{(\mu,\sigma,\xi)}(z_n). \tag{5.2.1}$$

A more appropriate approach which considers neighbouring departures, may be of the form:

$$\theta_n = \sum_{m=M_l}^{M_u} g(m) F_{(\mu,\sigma,\xi)}(z_{n+m}). \tag{5.2.2}$$

where $g(m)$ is some decaying function centred at zero, M_l and M_u are the limits on the number of departures either side of the n^{th} departure that may be affected,

and $g(0) = 1$. There are multiple elements of this approach which still need to be explored: (i) how the function $g(m)$ should behave, including how quickly it decays; (ii) The choice of M_l and M_u which will depend on the specific application and the time between departures e.g. hourly vs. daily departures; and (iii) The choice of normalisation. Since this approach may result in values of θ_n greater than 1, some further normalisation would be needed such that the outlier severity remains easily interpreted by analysts.

The temporal dependence of outliers could also be accounted for with spatiotemporal clustering. In this thesis, we have primarily focused on clustering legs (or terminals) which are spatially dependent. However, this could be extended to cluster together neighbouring departures on the same legs. For example, the aforementioned 09:00 and 10:00 departures could be clustered together, and outliers detected jointly.

5.2.2 Further analysis of forecast adjustments

In Chapter 3, the impact on revenue of different types of forecast adjustment was analysed. The revenue gained under the discussed adjustments could be described as a best-case-scenario, since we assumed the correct magnitude of the adjustment is known. In practice, this correct magnitude of adjustment is not known. Therefore, there is opportunity to adapt the method to tell the analyst not only where, when, and how severe an outlier is, but also to give a recommended change to the value of the forecast.

There is some scope to utilise the extrapolation step discussed in Chapter 2 as a method of obtaining such a recommendation. However, the aim of the extrapolation

was to improve the outlier detection performance, rather than to obtain an accurate prediction of future demand. Therefore, alternative forecasting approaches may be more appropriate for extrapolating with higher forecast accuracy, without compromising on outlier detection performance. Further computational studies could be carried out to consider a wider range of forecasting methods for extrapolation. Further, the extrapolation currently predicts the number of bookings in the remainder of the horizon. However, most revenue management systems require a demand forecast rather than a bookings forecast. Therefore, any extrapolation approach with the aim of recommending a forecast change, would likely need to include an unconstraining step to estimate the underlying demand from the observed bookings.

There is also the question of whether forecast adjustments to *easy to identify* outliers can be fully-automated. For example, outliers with a high severity and a high confidence in the prediction from the extrapolation could have their forecasts adjusted without analyst intervention. This would allow analysts to focus their expert judgement on departures where it is less clear what the adjustment should be.

5.2.3 Implementation and further empirical studies

The main focus of the empirical studies in this thesis has been the identification of outliers in hindsight. There are some further considerations which should be taken into account when the methods described in this thesis are implemented in a real system, and used to guide analysts. Some of these considerations, such as how to communicate the outlier alerts to analysts, have been discussed in Chapter 4.

Further avenues of future research include how often the clustering procedure de-

scribed in Chapter 3 should be run, how often the outlier detection procedure should be run, and whether these choices can be made independently of each other. Additionally, the computational aspect of the procedure needs to be considered before implementation. The current method of calculating the functional depth threshold described in Chapter 2 is computationally intensive and likely to be a bottleneck in terms of the time it takes to detect outliers. Further research may consider alternative approaches to threshold calculation which are less computationally intensive e.g. fitting a distribution to the functional depths. Additional computational studies would be needed to examine the performance on outlier detection before it is used in an empirical setting.

After the outlier detection method has been implemented as a decision support tool for analysts, further empirical studies could be carried out to determine its success, both in terms of (i) how well it supports decision making, and (ii) its impact on revenue. With regard to the former problem, this may incorporate feedback from analysts. For example, analysts may rate the *usefulness* of the outlier alert, or mark the alert as a false alert. This information could then be incorporated into a more advanced outlier detection procedure which includes a supervised learning step. With regard to the second suggested study on analysing the revenue impact, this would extend the findings in Chapters 2 and 3. The revenue implications of correcting for outlier demand have been quantified in this thesis. However, these results are a best-case scenario based on an analyst taking the correct action after an outlier has been detected. Further work is needed to analyse the impact of the necessarily imperfect actions that analysts in a real system will take.

Appendix A

Appendix: Identifying and responding to outlier demand in revenue management

A.1 Technical Description of Methodologies

A.1.1 Outlier Detection Approaches

Let N be the number of booking patterns. We observe the cumulative number of bookings for each booking pattern at T time points over a booking horizon of length t_T: $t_1, \ldots, t_\tau, \ldots, t_T$. Note that $t_1, \ldots, t_\tau, \ldots, t_T$ do not necessarily need to be equally spaced. Then $\boldsymbol{y}_n(t_\tau)$ is a time series of bookings for pattern n, up to time t_τ: $\boldsymbol{y}_n(t_\tau) = (y_n(t_1), y_n(t_2), \ldots, y_n(t_\tau))$.

Nonparametric Percentiles

Let $\boldsymbol{y}(t_\tau) = (y_1(t_\tau), \ldots, y_N(t_\tau))$ be the cumulative number of bookings for patterns $1, \ldots, N$ at time t_τ. Find the lower and upper (2.5% and 97.5%) percentiles of the ordered sample, L and U. For any booking pattern n, if the number of bookings at time t_τ, $y_n(t_\tau)$ is less than L or greater than U, it is defined as an outlier at time t_τ. Note that an alternative (parametric) approach would be to fit a distribution to the data and use the lower and upper percentiles of the fitted distribution.

Tolerance Intervals

For $Y(t_\tau)_1, \ldots, Y(t_\tau)_n$, a random sample from a population with distribution $F(Y(t_\tau))$, if:

$$\mathbb{P}\left(F(U(t_\tau)) - F(L(t_\tau)) > \beta\right) = 1 - \alpha, \tag{A.1.1}$$

then the interval $(L(t_\tau), U(t_\tau))$ is called a $(\beta, 1-\alpha)$ two-sided tolerance interval (Hahn and Chandra, 1981). At each booking interval, a tolerance interval for the number of bookings until that point in time, can be defined. If the number of bookings lies outside of this tolerance interval, the booking pattern can be deemed an outlier.

- **Nonparametric Tolerance Intervals**: Let $Y(t_\tau)_{(1)}, \ldots, Y(t_\tau)_{(n)}$ be the ordered observations of the sample $Y(t_\tau)_1, \ldots, Y(t_\tau)_n$. Wilks (1941) details that a $(\beta, 1-\alpha)$ tolerance interval can be calculated as follows:

  1. Let $B \sim Binomial(n, \beta)$, then let k be the smallest integer such that:

$$\mathbb{P}\left(B \leq k-1\right) \geq 1 - \alpha \tag{A.1.2}$$

2. Letting $k = s - r$, where $1 \leq r < s \leq n$, then $\left(Y(t_\tau)_{(r)}, Y(t_\tau)_{(s)}\right)$ is a tolerance interval, for any such r and s. It is common to choose:

$$r = \left\lfloor \frac{n-k+1}{2} \right\rfloor, \tag{A.1.3}$$

then $s = k + r$ i.e. $s = n - r + 1$.

- **Parametric Tolerance Intervals**: Given the discrete, count nature of the data, an obvious first choice for the number of bookings at time t_τ, is a Poisson distribution. Supposing $y(t_\tau)$ is the observed value of a random variable $Y(t_\tau)$ which has a Poisson distribution, $Po(n\lambda)$, a $(\beta, 1-\alpha)$ tolerance interval based on $y(t_\tau)$ is constructed in two steps, as described by Hahn and Chandra (1981):

1. Calculate a two-sided $(1-\alpha)$-level confidence interval, $(l(t_\tau), u(t_\tau))$ for λ, such as:

$$(l(t_\tau), u(t_\tau)) = \left(\frac{\chi^2_{(\alpha/2;2y(t_\tau))}}{2n}, \frac{\chi^2_{(1-\alpha/2;2y(t_\tau)+2)}}{2n} \right) \tag{A.1.4}$$

2. Find the minimum number $U(t_\tau)$, and the maximum number $L(t_\tau)$ such that:

$$\mathbb{P}\left(Y(t_\tau) < U(t_\tau) | \lambda = u(t_\tau)\right) \geq \frac{1+\beta}{2} \tag{A.1.5}$$

$$\text{and } \mathbb{P}\left(Y(t_\tau) > L(t_\tau) | \lambda = l(t_\tau)\right) \geq \frac{1+\beta}{2}. \tag{A.1.6}$$

Given the desire for a general method, the presence of differing mean-variance relationships between fare classes and over time, suggests that assuming a Poisson distribution may not be appropriate, given the fixed (equal) mean-variance relationship of this distribution. Alternative distributions which could be tested

include the Negative Binomial, which has two parameters for mean and variance (although only allows the variance to be larger than the mean), or the Generalised Poisson distribution, which has an additional parameter allowing the variance to change.

Robust Z-Score

Let $y_n(t_\tau)$ be the cumulative number of bookings for flight n at time t_τ. The robust Z-score can be calculated as (Iglewicz and Hoaglin, 1993):

$$\tilde{Z}_n = \frac{0.6745\,(y_n(t_\tau) - \tilde{y}(t_\tau))}{MAD(t_\tau)}, \tag{A.1.7}$$

where $\tilde{y}(t_\tau)$ is the median number of bookings at time t_τ across all booking patterns, and the Median Absolute Deviation at time t_τ, $(MAD(t_\tau))$, is given by:

$$MAD(t_\tau) = median\,\{|y_n(t_\tau) - \tilde{y}(t_\tau)|\} \tag{A.1.8}$$

A booking pattern, n, can be classified as an outlier at time t_τ, if the number of bookings at time t_τ, $y_n(t_\tau)$, has a modified Z-score with magnitude above 3.5, as described by Iglewicz and Hoaglin (1993).

Distance

Given that a time series of length τ can be thought of as a point in a τ-dimensional space, the distance between two time series can be calculated and used as a measure of the difference between them. In particular, for a time series $\boldsymbol{y}_n(t_\tau) = (y_n(t_1), y_n(t_2), \ldots, y_n(t_\tau))$, we define:

$$D_n(t_\tau) = \frac{1}{N-1} \sum_{m=1}^{N} \mathcal{D}(\boldsymbol{y}_n(t_\tau), \boldsymbol{y}_m(t_\tau)) \tag{A.1.9}$$

where $\mathcal{D}(\boldsymbol{y}_n(t_\tau), \boldsymbol{y}_m(t_\tau))$ is the distance between two booking patterns, n and m, up to time t_τ, and N is the total number of booking patterns being considered. Here the distance-based outlier score is given as the mean distance of a point to its k-nearest neighbours, and we set $k = N - 1$, all other points. Hence, for some given threshold, all booking patterns whose mean distance is larger than the threshold can be marked as an outlier. Booking pattern n can be defined as an outlier, at time t_τ, if:

$$D_n(t_\tau) \geq \mu_d + 3\sigma_d \tag{A.1.10}$$

where μ_d is the mean of the mean distances, and σ_d the standard deviation. We consider both Euclidean and Manhattan distance metrics:

- **Euclidean**:

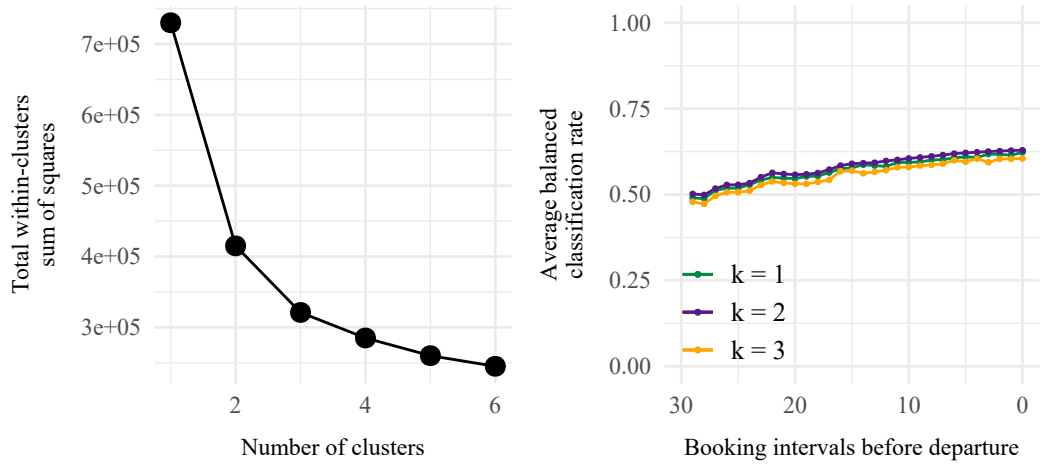
$$\mathcal{D}_E(\boldsymbol{y}_n(t_\tau), \boldsymbol{y}_m(t_\tau)) = \left(\sum_{u=1}^{\tau} \left(y_n(t_u) - y_m(t_u) \right)^2 \right)^{\frac{1}{2}} \tag{A.1.11}$$

- **Manhattan**:

$$\mathcal{D}_M(\boldsymbol{y}_n(t_\tau), \boldsymbol{y}_m(t_\tau)) = \sum_{u=1}^{\tau} \left| y_n(t_u) - y_m(t_u) \right| \tag{A.1.12}$$

K-Means Clustering

It should be noted that clustering algorithms, such as K-means clustering, are optimised to determine clusters instead of outliers meaning that the success of the outlier detection relies on an algorithm's ability to accurately determine the structure of the clusters. The distance threshold at which a point is classified as an outlier also needs to be specified. Deb and Dey (2017) describe a global threshold distance, at which point those observations which are further away from their cluster centre are classed



(a) Within cluster sum of squares for choosing K

(b) BCR under different values of K

Figure A.1.1: Choosing K

as outliers, as being half the sum of the maximum and minimum distances. The procedure for identifying booking patterns observed up to time t_τ as outliers is as follows:

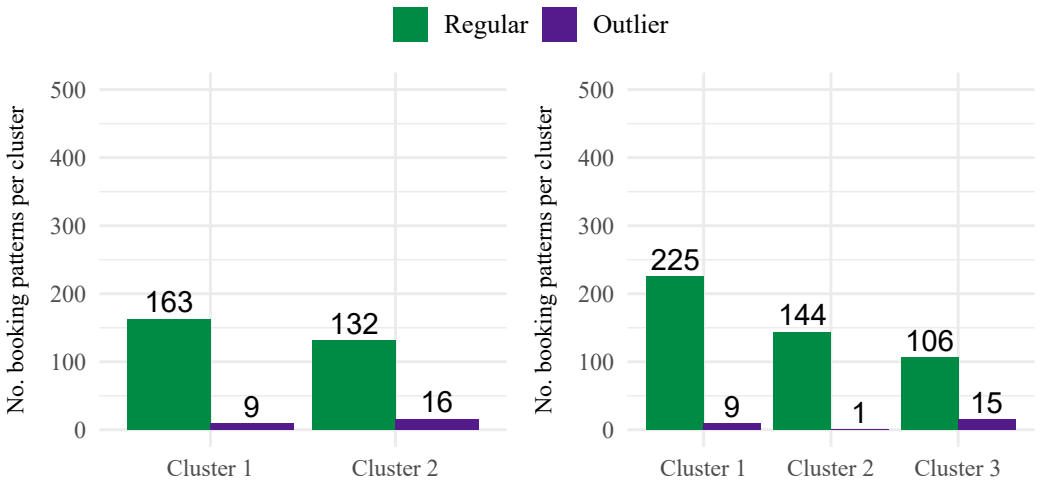
1. Choose K, the number of clusters.

2. Randomly assign K booking patterns to be the initial cluster centres.

3. Calculate the τ-dimensional distance (Euclidean or Manhattan) from each booking pattern in the data set to each cluster centre, and assign each booking pattern to the cluster centre from which it is the smallest distance.

4. Recalculate the centre of each cluster based on the booking patterns assigned to it.

5. Repeat steps (3) and (4) until the assignment of booking patterns to clusters no longer changes.



(a) K = 2 (b) K = 3

Figure A.1.2: Distribution of Genuine Outliers Across Clusters

K-means clustering relies on specifying the number of clusters in advance. The optimal number of clusters should seek to minimise the within cluster sum of squares without overfitting. Choosing k is a difficult problem as it requires fitting k-means with multiple values of k and choosing the best one. Figure A.1.1a demonstrates the within cluster sum of squares for multiple values of k, where the optimal number of clusters is chosen as the *elbow* of the plot, $k = 2$. To investigate the impact that the choice of K has on the outlier detection performance, we compare the balanced classification for $K = 1$, 2, and 3. These results are shown in Figure A.1.1b. There is little difference between the different values of K, suggesting that the choice of K is not what drives the poor performance of K-means clustering.

It may be surprising that the optimal number of clusters is chosen as 2 rather than 1, given that the regular demand is generated from a single distribution. It raises the question of whether the algorithm is clustering the booking patterns into regular and

outlying patterns. This would mean that it would fail to detect the outlying booking patterns as they have their own cluster and so distance to their cluster centre is small. However, upon investigation of the distribution of outlying booking patterns across clusters (shown in Figure A.1.2), it was found not to be the case. It is possible that the booking limits introduce some element of bi-modality.

Multivariate Functional Halfspace Depth

The general procedure for detecting outliers at time τ using functional depth, as described by Febrero et al. (2008) and Hubert et al. (2015), is as follows:

1. Define $\mathcal{D}_n(\boldsymbol{y}_n(t_\tau))$ to be the functional depth of the $\boldsymbol{y}_n(t_\tau) = (y_n(t_1), y_n(t_2), \ldots, y_n(t_\tau))$, booking pattern n at time t_τ.

2. Define a threshold, C, for the functional depth.

3. Those booking patterns with functional depths, $\mathcal{D}_n(\boldsymbol{y}_n(t_\tau))$, below the threshold are classified as outliers, delete them from the sample.

4. Recalculate functional depths on the new sample, and remove further outliers. Repeat until no more outliers are found.

As described by Febrero et al. (2008), the threshold, C, is ideally chosen such that:

$$\mathbb{P}(\mathcal{D}_n(\boldsymbol{y}_n(t_\tau)) \leq C) = 0.01, \ n = 1, \ldots, N, \tag{A.1.13}$$

when there are no genuine outliers present in the sample. However, this would require knowing the distribution of functional depths when there are no outliers. Febrero et al. (2008) discuss two bootstrapping-based procedures for estimating C. The general

idea of the bootstrapping method used in this chapter, as described by Febrero et al. (2008), is to (i) resample the booking patterns, with probability proportional to their functional depths (such that any outlying patterns are less likely to be resampled), (ii) smooth the bootstrap samples, then (iii) set C as the median value of the 1% percentiles of the empirical distributions of the depths of the bootstrapped samples. More specifically:

1. Calculate the functional depths for each booking pattern, $\mathcal{D}_n(\boldsymbol{y}_1(t_\tau)), \ldots, \mathcal{D}_n(\boldsymbol{y}_n(t_\tau))$.

2. Resample the original booking patterns to obtain B bootstrap samples, where each booking pattern is sampled with probability proportional to its functional depth. Denote the n^{th} booking curve in the b^{th} bootstrap sample as $\boldsymbol{x}_n^b$.

3. Smooth the bootstrap samples to obtain $\boldsymbol{s}_n^b = \boldsymbol{x}_n^b + \boldsymbol{z}_n^b$, where $\boldsymbol{z}_n^b = (z_n(t_1), z_n(t_2), \ldots, z_n(t_\tau))$ is normally distributed with mean 0 and covariance matrix $\gamma\Sigma$. γ is a smoothing parameter, and Σ is the covariance matrix of the original sample.

4. Calculate the functional depths for the resampled booking patterns in each of the smoothed bootstrap samples. Let C^b be the empirical 1^{st} percentile of the distribution of these depths for the b^{th} sample.

5. Choose the threshold C as the median of the values of C^b, for $b = 1, \ldots, B$.

For full details, see Febrero et al. (2008).

In this chapter, we restrict our attention to halfspace depth. In the case of one-dimensional random variables, the halfspace depth of a point y_n with respect to a sample $y1, \ldots, y_N$ drawn from distribution F is:

$$HD(y_n) = min\left\{F_N(y_n), 1 - F_N(y_n)\right\} \tag{A.1.14}$$

where F_N is the empirical cumulative distribution of the sample $y_1, \ldots, y_N$ (Febrero et al., 2008). This definition has been extended to the functional data setting, see Hubert et al. (2012) and Claeskens et al. (2014). Let $\boldsymbol{y}_n(t_\tau) = (y_n(t_1), y_n(t_2), \ldots, y_n(t_\tau))$ be booking pattern n up to time t_τ, where $n = 1, \ldots, N$, and each $y_n(t_i)$ is a K-variate vector. In the functional setting, the multivariate functional halfspace depth of a pattern $\boldsymbol{y}_n(t_\tau) = (y_n(t_1), y_n(t_2), \ldots, y_n(t_\tau))$ is given by:

$$MFHD_{N,\tau}(\boldsymbol{y}_n(t_\tau); \alpha) = \sum_{j=1}^{\tau} w_{\alpha,N}(t_j) HD_{N,j}(\boldsymbol{y}_n(t_j)) \tag{A.1.15}$$

where, using $t_{\tau+1} = t_\tau + 0.5(t_\tau - t_{\tau-1})$, the weights, $w_{\alpha,N}(t_j)$, are, according to Hubert et al. (2012):

$$w_{\alpha,N}(t_j) = \frac{(t_{j+1} - t_j)\text{vol}\left[\left\{\boldsymbol{x} \in \mathbb{R}^k : HD_{N,j}(\boldsymbol{x}) \geq \alpha\right\}\right]}{\sum_{j=1}^{\tau}(t_{j+1} - t_j)\text{vol}\left[\{\boldsymbol{x} \in \mathbb{R}^k : HD_{N,j}(\boldsymbol{x}) \geq \alpha\}\right]} \tag{A.1.16}$$

and the sample halfspace depth of a K-variate vector x at time t_j is given by (Hubert et al., 2012):

$$HD_{N,j}(x) = \frac{1}{N} \min_{\boldsymbol{u}, ||\boldsymbol{u}||=1} \#\left\{y_n(t_j), n = 1, \ldots, N : \boldsymbol{u}^T y_n(t_j) \geq \boldsymbol{u}^T \boldsymbol{x}\right\} \tag{A.1.17}$$

In this chapter, we are considering a univariate, $K = 1$, functional halfspace depth since we choose to monitor booking patterns only. However, the definition of a multivariate functional halfspace depth opens up the possibility of jointly monitoring

booking patterns and revenue patterns, for example. As described by Hubert et al. (2012), computing the multivariate functional halfspace depth can be done with fast algorithms, and in this chapter we use the R-package `mrfDepth` to do so.

A.1.2 Univariate forecasting techniques for extrapolation

Although an important element of a revenue management system is forecasting, there are multiple reasons why we create new forecasts to extrapolate rather than using the existing ones generated by the RM system. Three particular reasons are (i) depending on the optimisation routine used to set booking limits, forecasts of how demand builds up over time may not have been calculated. Some methods only require forecasts of final demand, and so the type of forecasts we wish to use for extrapolation may not exist. (ii) In the event that forecasts of how demand builds up over time do exist, historical forecasts may not be stored. In terms of identifying critical booking patterns in historical data, this also means the forecasts used for extrapolation are not available. (iii) Forecasts for how demand accumulates over time are typically based on data from similar historical booking patterns. The use of data from other booking patterns to extrapolate has the potential to mask outliers by normalising behaviour. Hence, at each time point we wish to create a forecast based solely on the data for an individual booking pattern, with the goal not being to accurately predict demand, but rather to amplify the differences between booking patterns.

Simple Exponential Smoothing (SES)

SES works on the principle of averaging whilst down-weighting older observations. Further details can be found in Chatfield (1975). Given a time series $y_n(t_1), y_n(t_2), \ldots, y_n(t_\tau)$, a forecast for time $t_{\tau+1}$, $\hat{y}_n(t_{\tau+1})$ is given by:

$$\hat{y}_n(t_{\tau+1}) = \alpha y_n(t_\tau) + (1-\alpha)\hat{y}_n(t_\tau), \tag{A.1.18}$$

for some smoothing constant, α. Note that this results in a constant forecast for the bookings from time $t_{\tau+1}, \ldots, t_T$. Due to the inability of SES to cope with trend, we apply SES to the time series of demand per booking interval, rather than the time series of cumulative demand.

Autoregressive Integrated Moving Average (ARIMA)

ARIMA models incorporate a trend component, and assume that future observations are an additive, weighted combination of previous observations and previous errors. Let $\boldsymbol{x}_n(t_\tau)$ be the d^{th} differenced time series relating to $\boldsymbol{y}_n(t_\tau)$. See Box and Jenkins (1970) for an overview of differencing procedures, and Chatfield (1975) for a description of ARIMA processes. The one-step ahead forecast $\hat{x}_n(t_{\tau+1})$ is given by:

$$\hat{x}_n(t_{\tau+1}) = \mu + \phi_1 x_n(t_\tau) + \ldots + \phi_p x_n(t_{\tau-p+1}) - \theta_1 \epsilon(t_\tau) - \ldots - \theta_q \epsilon(t_{\tau-q+1}) \tag{A.1.19}$$

for some constant mean μ, parameters $\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q$ and white noise process $\left(\epsilon_{t_j}\right)$. We use AIC and Dickey-Fuller tests, in combination with visual inspection, to select the orders p, q, and d. See Box and Jenkins (1970), and the R package `forecast`.

Integrated Generalised Autoregressive Conditional Heteroskedasticity (IGARCH)

IGARCH models incorporate a trend component and assume that the variance structure follows an autoregressive moving average model. Again, let $\boldsymbol{x}_n(t_\tau)$ be the d^{th} differenced time series relating to $\boldsymbol{y}_n(t_\tau)$. See Tsay (2002) for further details on IGARCH processes. IGARCH(1,d,1) models assume the following structure:

$$x_n(t_{\tau+1}) = \mu + \epsilon_n(t_{\tau+1}) \quad \text{(A.1.20)}$$

$$\epsilon_n(t_{\tau+1}) = z_n(t_{\tau+1})\sigma_n(t_{\tau+1}) \quad \text{(A.1.21)}$$

$$\sigma_n^2(t_{\tau+1}) = w + \alpha\epsilon_n^2(t_{\tau+1}) + \beta\sigma_n^2(t_\tau) \quad \text{(A.1.22)}$$

We assume that the order of the IGARCH model is $(1, d, 1)$ to reduce computational time.

A.2 Details of Simulation-based Framework

A.2.1 Forecasts

In terms of choosing the number of replications of the simulation, N_S, to use in the calculations of the forecasts, we consider the standard errors of the estimates. The standard error of the mean is given by:

$$se(\hat{\mu_j}) = \frac{\hat{\sigma_j}}{\sqrt{N_S}}, \quad \text{(A.2.1)}$$

	Fare Class	Fare (€)	f_D = 0.9		f_D = 1.2		f_D = 1.5	
j		r_j	$\hat{\mu}_j$	$\hat{\sigma}_j{}^2$	$\hat{\mu}_j$	$\hat{\sigma}_j{}^2$	$\hat{\mu}_j$	$\hat{\sigma}_j{}^2$
1	A	400	31.9	23.0	46.2	25.3	52.7	32.2
2	O	300	17.5	14.2	24.2	18.8	28.3	30.5
3	J	280	20.0	14.2	28.6	25.5	33.6	31.8
4	P	240	16.8	16.1	22.9	26.6	26.1	23.8
5	R	200	13.4	11.5	18.5	16.5	21.6	18.8
6	S	185	12.3	14.3	16.9	11.2	21.0	21.1
7	M	175	52.6	19.2	69.8	28.2	81.8	33.8

Table A.2.1: Forecasts of mean and variance of demand for each fare class

such that it is typically in the range of 0.3 - 0.6 when $N_S = 100$. The standard error of the variance is given by:

$$se(\hat{\sigma_j^2}) = \hat{\sigma_j^2}\sqrt{\frac{2}{N_S - 1}}, \tag{A.2.2}$$

and is typically in the range of 2 - 5 when $N_S = 100$. Therefore the number of simulations provides reasonable estimates of the demand mean and variance forecasts for each fare class.

A.2.2 Optimisation Heuristics to Compute Booking Limits

Expected Marginal Seat Revenue-b (EMSRb)

It is assumed that demand for each fare class, d_i, is independent and normally distributed:

$$d_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right), \tag{A.2.3}$$

where μ_i and σ_i^2 are forecasted as described above. The protection level for fare class j is given by (Belobaba, 1992):

$$PL_j = F_j^{-1}\left(1 - \frac{r_{j+1}}{\tilde{r}_j}\right) \text{ for } j = 1, \ldots, |\mathcal{J}| - 1 \tag{A.2.4}$$

where F_j is the (Gaussian) distribution of demand for fare class j, and r_j is the fare in fare class j. $\tilde{r}_j$ is the weighted-average revenue from classes $1, \ldots, j$:

$$\tilde{r}_j = \frac{\sum_{k=1}^{j} r_k \mu_k}{\sum_{k=1}^{j} \mu_k}. \tag{A.2.5}$$

Note that the protection level for all fare classes, $PL_{|\mathcal{J}|}$, is simply equal to the capacity, C. As stated by Talluri and Van Ryzin (2004), Equation (A.2.4) becomes:

$$PL_j = \mu + \Phi^{-1}\left(1 - \frac{r_{j+1}}{\tilde{r}_j}\right)\sigma \text{ for } j = 1, \ldots, |\mathcal{J}| - 1, \tag{A.2.6}$$

where $\mu = \sum_{k=1}^{j} \mu_k$ is the mean, and $\sigma^2 = \sum_{k=1}^{j} \sigma_k^2$ is the variance, of the aggregated demand. Hence, the booking limit for class j is given by the capacity minus the protection level for classes $j-1$ and higher:

$$BL_j = C - PL_{j-1}. \tag{A.2.7}$$

Expected Marginal Seat Revenue-b with Marginal Revenue Transformation (EMSRb-MR)

The following marginal revenue transformation, described by Fiig et al. (2010), assumes that customers only buy the lowest available fare, even if they would be willing to pay more. In this setting, letting k be the lowest available fare product, the demand for all other fare products becomes zero:

$$\mu_j = 0 \ \forall j \neq k. \tag{A.2.8}$$

Therefore the adjusted demand for fare class j becomes:

$$\mu_j^{'} = \mu_j - \mu_{j-1}. \tag{A.2.9}$$

The adjusted fares are given by:

$$r_j^{'} = \frac{r_j \mu_j - r_{j-1}\mu_{j-1}}{\mu_j - \mu_{j-1}}. \tag{A.2.10}$$

An alternative method of calculating adjusted fares without explicitly forecasting demand for each fare class is to assume that:

$$\mu_j = \mu_n psup_j, \tag{A.2.11}$$

the demand for a particular fare class is the baseline demand for the lowest fare class, μ_n, multiplied by a sell-up probability, $psup_j$. In practice, these sell-up probabilities can be forecasted instead of the fare class demand assuming an independent model. In our case, due to comparing EMSRb with EMSRb-MR, we have the fare class forecasts already. The two methods are equivalent.

The booking controls under EMSRb and EMSRb-MR are shown in Table A.2.2, where the demand factor, f_D, is defined as the ratio of demand, D, to capacity, C.

Fare Class	f_D = 0.9		f_D = 1.2		f_D = 1.5	
	EMSRb	**EMSRb-MR**	**EMSRb**	**EMSRb-MR**	**EMSRb**	**EMSRb-MR**
A	200	200	200	200	200	200
O	171	165	157	151	151	144
J	155	155	134	134	125	125
P	134	125	105	95	90	79
R	117	109	81	72	62	52
S	104	109	62	72	39	52
M	91	96	45	51	18	24

Table A.2.2: Booking limits under EMSRb and EMSRb-MR

A.3 Additional Results

A.3.1 Comparison of Booking Limit Heuristics

Table A.3.1 shows the resulting revenue under EMSRb and EMSRb-MR booking limits with different demand factors, as compared to accepting bookings on a first-come-first-served basis (FCFS). Both heuristics offer an improvement over FCFS. Given the presence of buy-down in the demand model, EMSRb-MR outperforms EMSRb, particularly in situations that feature a high demand-to-capacity ratio. Given the significant impact of heuristic choice on revenue, we investigate whether the superior performance of EMSRb-MR also results in a change in outlier detection performance. Figure A.3.1 shows the balanced classification rate of functional depth with ARIMA extrapolation outlier detection, under EMSRb and EMSRb-MR heuristics. There is

Optimisation	**Magnitude of Outliers**	**Frequency of Outliers**	Nonparametric	Percentiles Nonparametric	Poisson Tolerance	Intervals Robust	Euclidean	Distance Manhattan	k-Means Clustering	(Euclidean) k-Means Clustering	Functional Depth	Functional Depth	Functional Depth + ARIMA	Functional Depth + IGARCH
EMSRb	-25%	1%	0.69	0.63	0.73	0.50	0.69	0.68	0.68	0.68	0.94	0.94	0.93	0.93
		5%			0.74				0.67		0.93		0.94	
		10%			0.70				0.66		0.93		0.93	
	-12.5%	1%	0.56	0.55	0.55	0.50	0.56	0.55	0.58	0.57	0.93	0.92	0.95	0.93
		5%			0.56				0.57		0.92		0.93	
		10%			0.54				0.56		0.93		0.93	
	+12.5%	1%	0.56	0.55	0.53	0.51	0.53	0.53	0.59	0.56	0.92	0.92	0.93	0.93
		5%			0.53				0.58		0.93		0.93	
		10%			0.51				0.59		0.92		0.92	
	+25%	1%	0.65	0.60	0.59	0.54	0.61	0.61	0.65	0.68	0.94	0.93	0.92	0.94
		5%			0.62				0.69		0.92		0.94	
		10%			0.60				0.68		0.92		0.93	
EMSRb-MR	-25%	1%	0.68	0.62	0.73	0.50	0.68	0.68	0.65	0.67	0.92	0.93	0.92	0.92
		5%			0.77				0.66		0.92		0.93	
		10%			0.71				0.67		0.91		0.92	
	-12.5%	1%	0.56	0.54	0.55	0.50	0.55	0.55	0.57	0.56	0.92	0.93	0.92	0.92
		5%			0.57				0.57		0.93		0.93	
		10%			0.59				0.60		0.93		0.92	
	+12.5%	1%	0.56	0.55	0.51	0.51	0.53	0.53	0.56	0.55	0.93	0.93	0.93	0.92
		5%			0.54				0.57		0.92		0.93	
		10%			0.51				0.59		0.93		0.92	
	+25%	1%	0.66	0.61	0.63	0.54	0.62	0.62	0.68	0.69	0.94	0.93	0.92	0.92
		5%			0.65				0.69		0.92		0.94	
		10%			0.61				0.70		0.93		0.92	

Table A.2.3: Balanced classification rate (offline) results for extended simulation study

Demand Factor	FCFS Revenue (€)	EMSRb as Factor of FCFS	EMSRb-MR as Factor of FCFS
0.90	28948.50	1.03	1.06
1.20	34835.50	1.04	1.08
1.50	35000.00	1.05	1.09

Table A.3.1: Revenue generated under EMSRb vs EMSRb-MR booking controls

no significant impact on outlier detection performance.

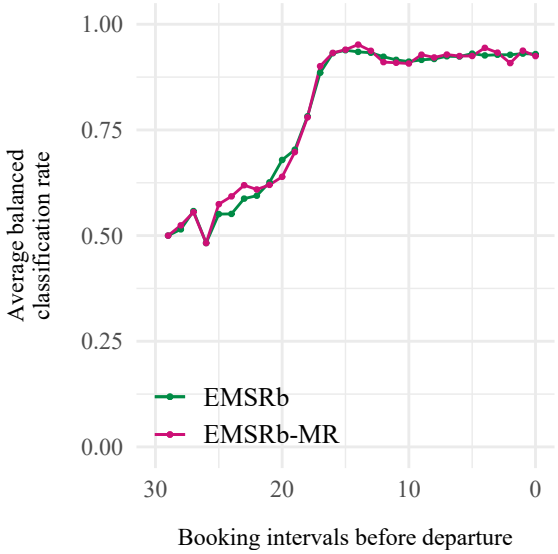


Figure A.3.1: EMSRb vs. EMSRb-MR under functional depth with ARIMA extrapolation

A.3.2 Sensitivity to Frequency of Outliers

We test the sensitivity of the functional depth (with and without extrapolation) to the different frequencies of outliers i.e. the proportion of booking patterns considered which are genuine outliers. There is no significant change in performance as the frequency of outliers changes, shown in Figure A.3.2. This consistent performance of the functional depth-based methods is down to the fact that it does not classify a specific proportion of the data as outlying. Given this, our simulation study considers the case where 5% of the $N = 500$ booking patterns are genuine outliers.

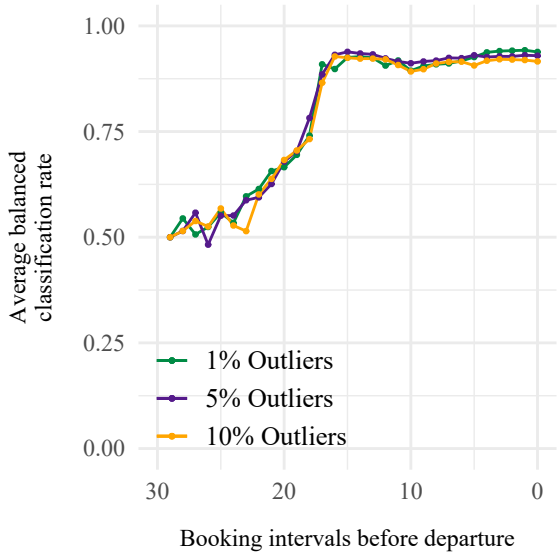


Figure A.3.2: Balanced Classification Rate under different frequencies of outliers for functional depth with ARIMA extrapolation

A.3.3 K-means clustering with ARIMA extrapolation

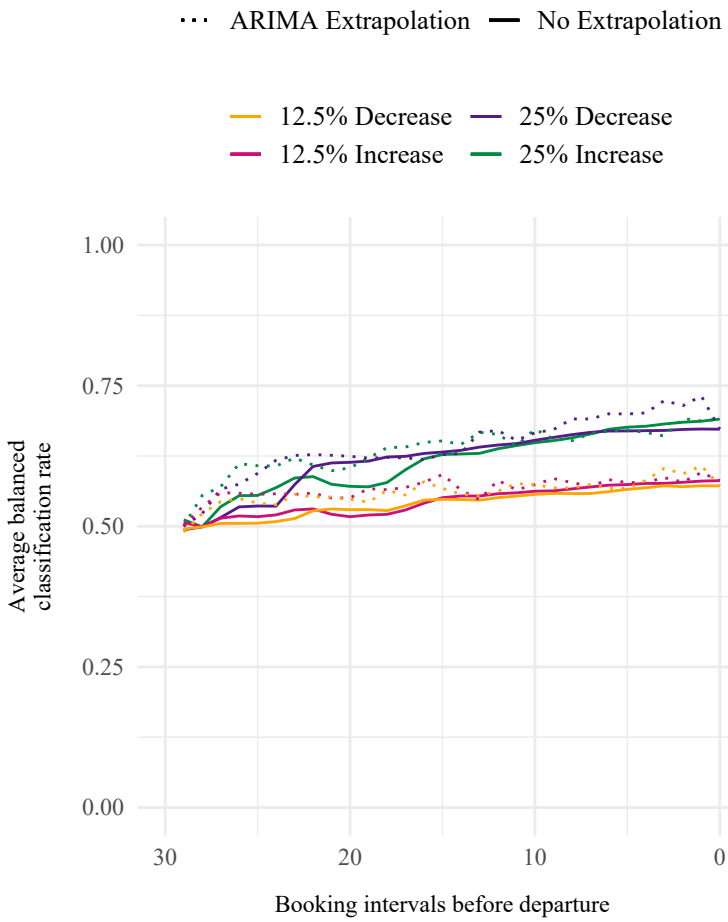


Figure A.3.3: Balanced Classification Rate for K-means clustering with ARIMA extrapolation for 5% outlier frequency over different magnitudes of demand outliers

As noted in Section 2.4, extrapolation could also be used with the multivariate outlier detection approaches. Although in this chapter we have chosen to focus on combining the extrapolation with the most promising outlier detection method (functional depth), we also present results here (see Figure A.3.3) on combining extrapolation with K-means clustering. As when combining extrapolation with functional depth (see Appendix C.6), the extrapolation increases the number of booking pat-

terns classified as outliers. Extrapolation does provide an improvement in outlier detection performance, though the increase in performance is smaller in comparison when combined with functional depth. This is unsurprising given the poor performance of K-means clustering even when the curves are fully observed. The overall performance is still not as good as combining extrapolation with functional depth (or even functional depth without extrapolation).

A.3.4 Motivating the Use of Functional Analysis

Importance of Time-Ordered Observations

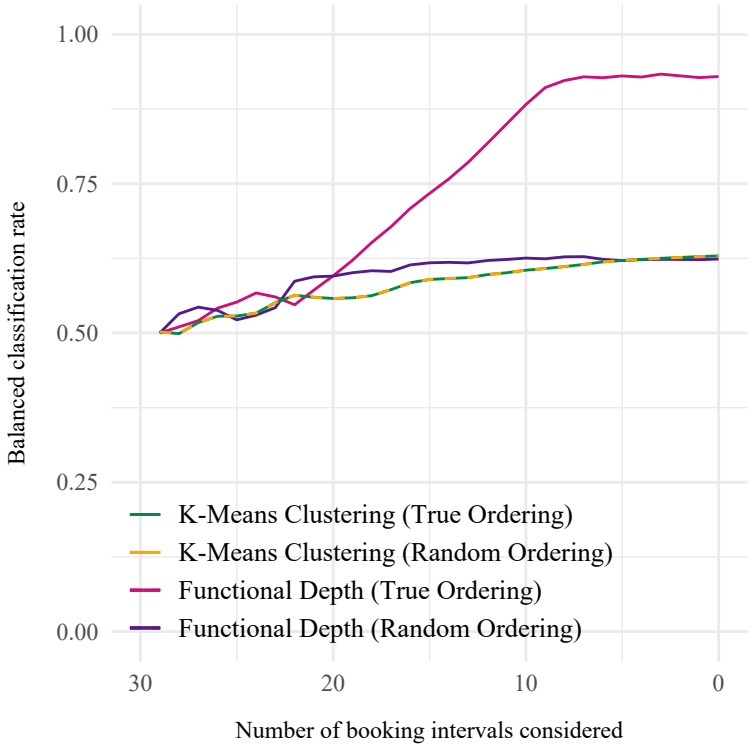


Figure A.3.4: Comparison of correct time-ordering vs. random time-ordering

To stress the importance of the time-ordering of the observations, we benchmark

K-means and functional depth on two cases where (i) the observations per pattern are correctly ordered, and (ii) the observations are shuffled randomly. As shown in Figure A.3.4, the performance of K-means clustering is independent of this change, as the observations stay the same and only their order, which K-means does not consider, changes. However, Figure A.3.4 shows that the performance of functional depth improves when the data is ordered, as this method can exploit this characteristic of the booking patterns. In other words, if the booking patterns were not time-ordered, multivariate approaches could indeed outperform functional depth. However, this temporal dependency does exist and motivates our use of functional analysis.

Dealing with High Dimensionality

The ROC analysis in Section 2.6.2 shows that the performance of K-means clustering becomes poorer as the dimensionality increases. The use of principal component analysis (PCA) was considered as a method to reduce the dimensionality for both distance measures and K-means clustering approaches. However, preliminary results were poor. When selecting the principal components to maximise the proportion of variance explained, it was deemed best to chose the time points early in the booking horizon. This is due to the censoring caused by the booking controls, i.e. when no booking limits have yet been reached, demand is more varied. This means that if we choose the principal components which best explain the variance, we no longer take into account the newest information. As such, detection accuracy was inferior to using the full vector of bookings, despite the dimensionality issue.

A.3.5 True Positive Rates

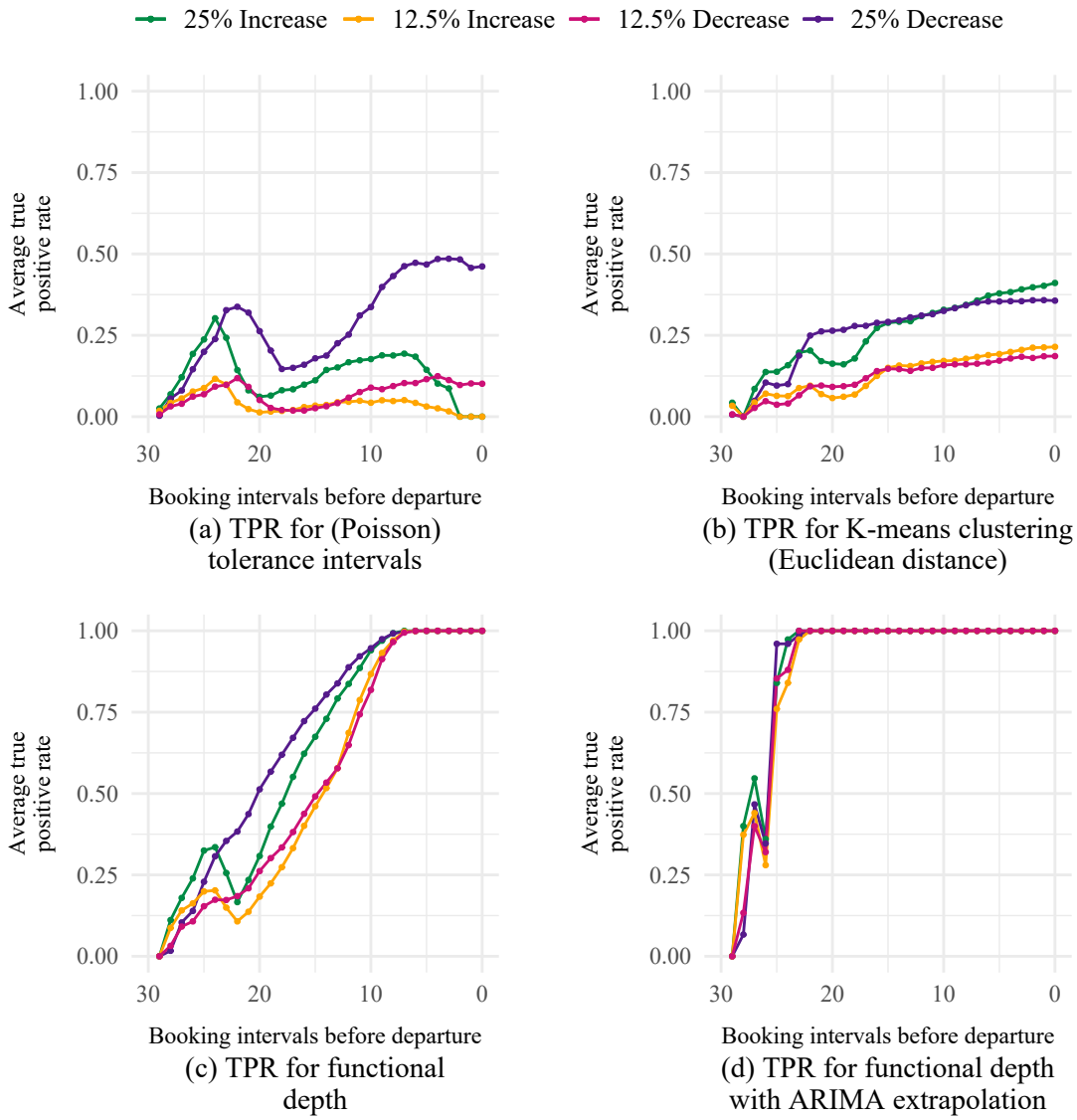


Figure A.3.5: True positive rates for various outlier detection methods

A.3.6 False Positive Rates

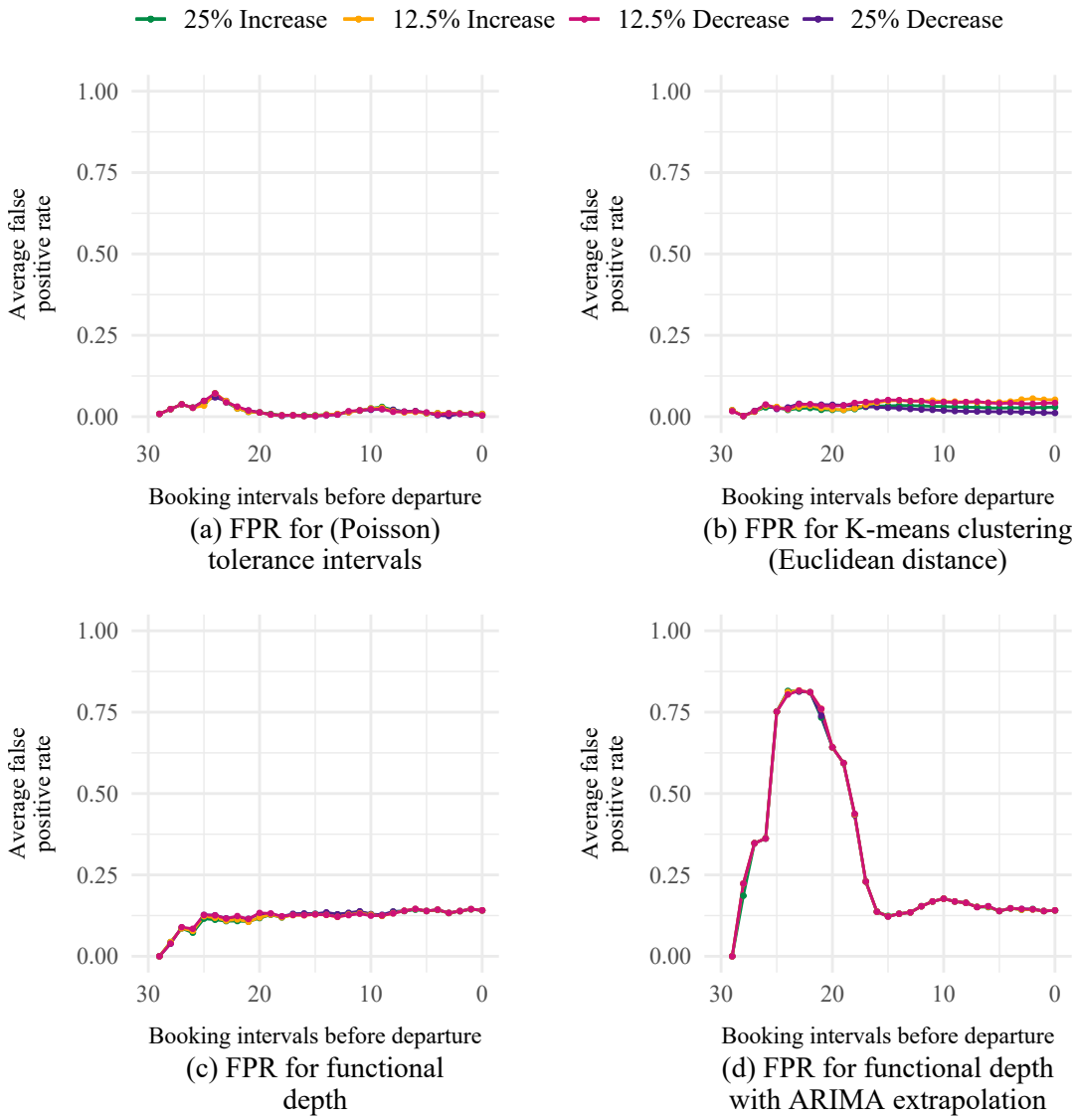
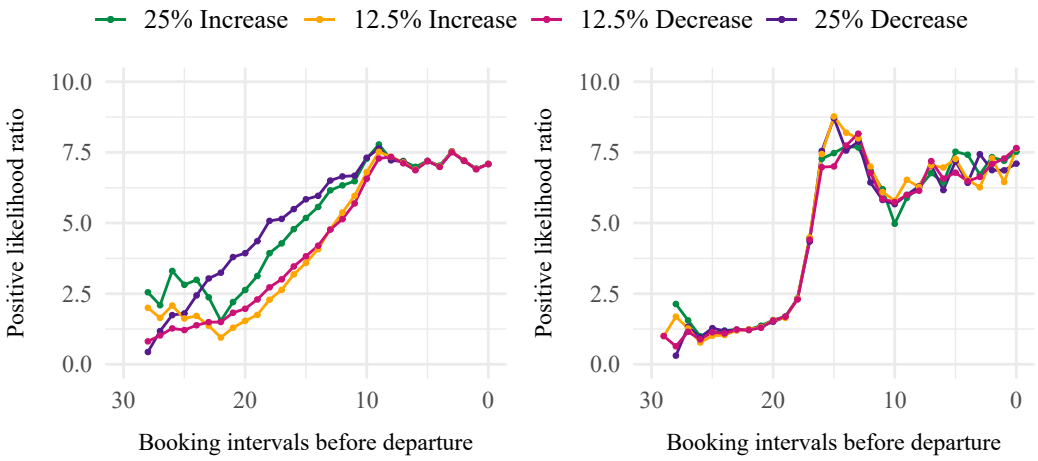


Figure A.3.6: False positive rates for various outlier detection methods

We suggest that early in the booking horizon, all methods perform poorly but for different reasons – some suffer from low true positive rates, others from high false positive rates. The balanced classification rate (BCR) does not allow us to easily compare these two situations. In order to test this hypothesis, and investigate the spike in false positives early in the booking horizon when incorporating extrapolation, we additionally consider the Positive Likelihood Ratio (LR+) (Habibzadeh and Habibzadeh, 2019). That is, the ratio between the true positive rate, and the false positive rate:

$$LR+ = \frac{TP/(TP+FN)}{FP/(FP+TN)} \tag{A.3.1}$$

A higher LR+ (specifically those greater than 1), represents the fact that a booking pattern classified as an outlier is more likely to be a genuine outlier.



(a) LR+ for functional depth

(b) LR+ for functional depth with ARIMA extrapolation

Figure A.3.7: Positive likelihood ratio for functional depth with and without ARIMA extrapolation

The results, shown in Figure A.3.7, show that functional depth both with and

without extrapolation performs poorly early in the horizon – with a LR+ just slightly above 1. Due to the high false positive rate, functional depth without extrapolation may even be marginally better early in the horizon. However, outlier detection with the inclusion of extrapolation reaches its peak LR+ around 16 intervals before departure, compared to 9 intervals for functional depth alone. This peak in functional depth with ARIMA extrapolation corresponds to both the sharp increase in true positives (Figure A.3.5), and sharp drop-off in false positives (Figure A.3.6). A similar impact was observed when K-means clustering was combined with extrapolation, that is, the number of booking patterns classified as outliers increased. These results, in addition with the ROC curves shown in Section 2.6.2, show that, on balance, it is still beneficial to include extrapolation into the outlier detection, especially in the middle portion of the booking horizon. However, classification results from all methods should be treated with caution very early in the booking horizon for the reasons outlined.

Given the superior performance across a range of thresholds (evidence by the ROC curves in Section 2.6.2), of functional depth with extrapolation, we consider whether using the same parameters to calculate the threshold across the booking horizon (following those implemented by Febrero et al. (2008)) is the best approach. We compare the percentage of booking patterns classified as outliers by functional depth with and without extrapolation (Figure A.3.8a). In addition, Figure A.3.8b the variance, across the booking horizon, of the ARIMA extrapolation at time t_T. We see that there is a relationship between the variance of the ARIMA extrapolation the number of patterns classified as outliers, and therefore the false positives. It may perhaps be possible to vary the threshold parameters according to the functional

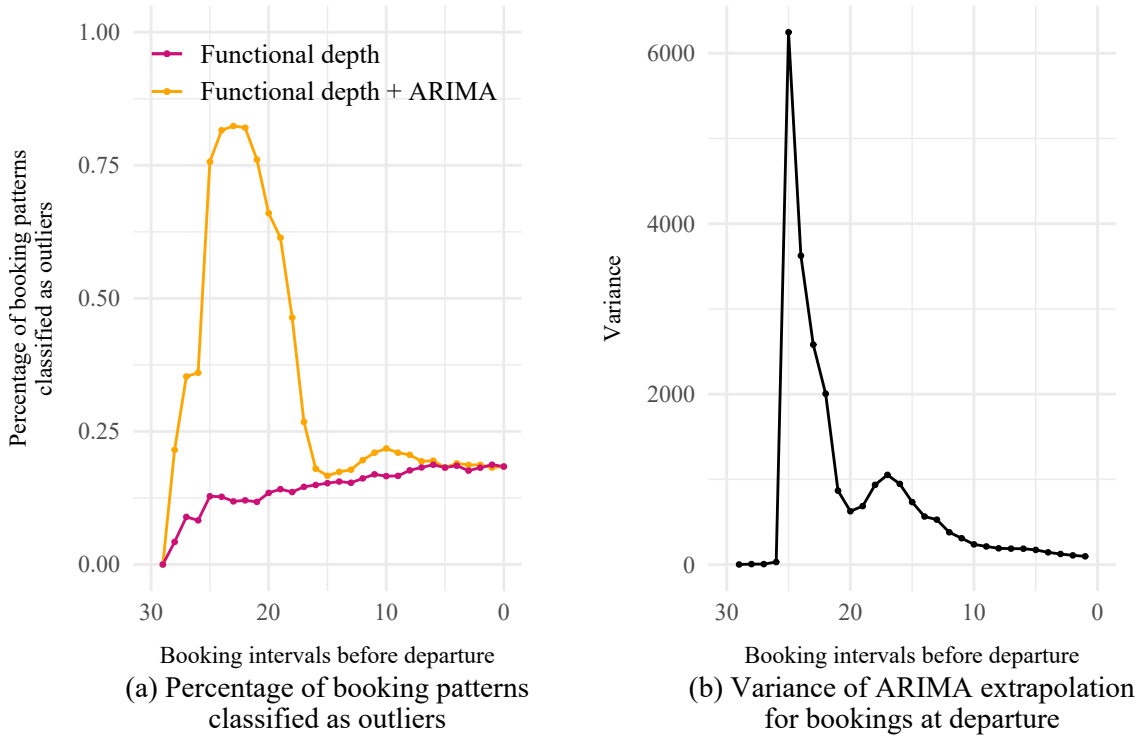
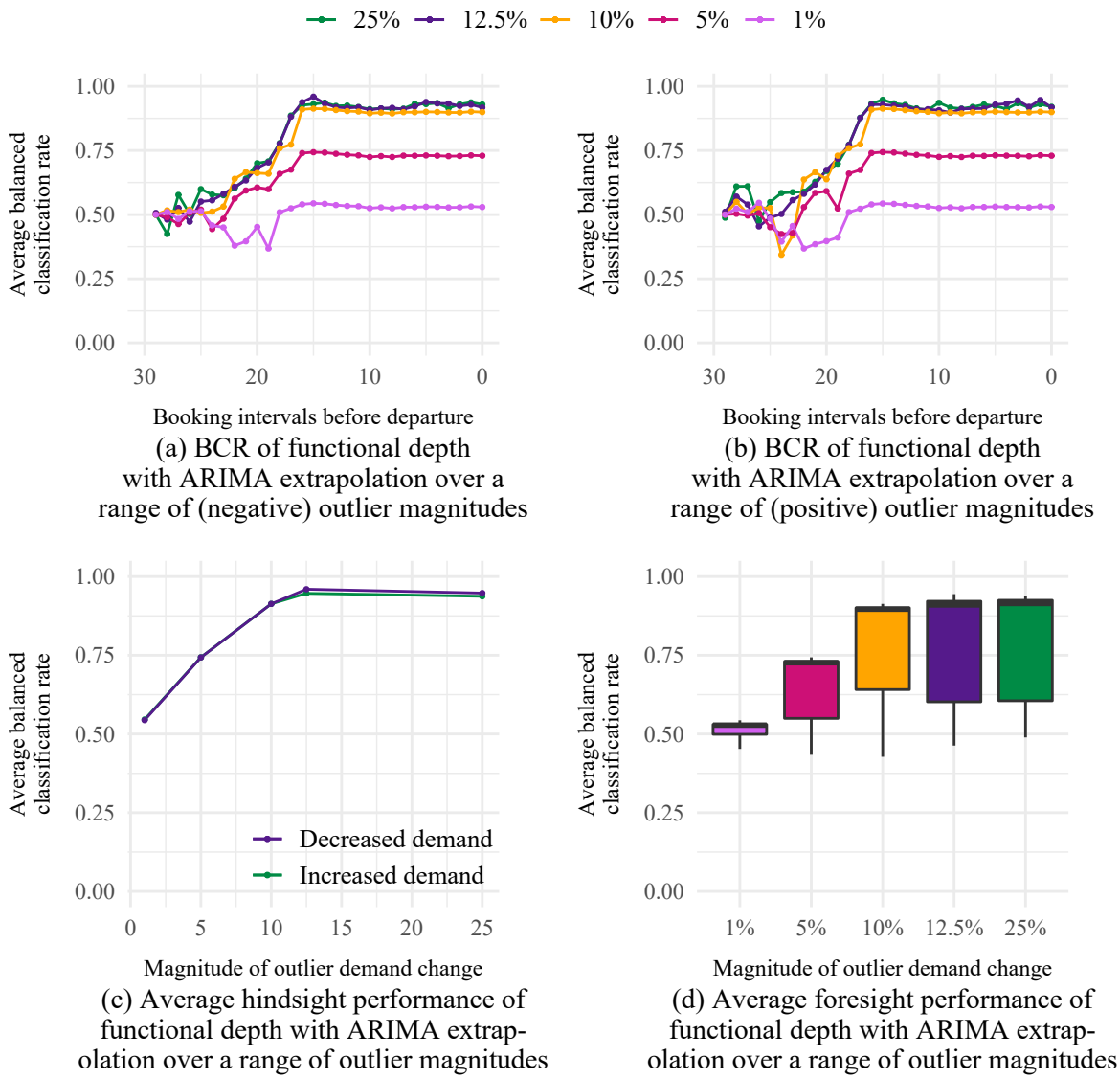


Figure A.3.8: Variance of ARIMA extrapolation for bookings at departure

variance, as it changes across the booking horizon with extrapolation. We see this as an opportunity for further work.

In practice, companies have a limited number of analysts to respond to outlier detection-based alerts. Hence, the threshold would likely be left variable. That is, if an analyst is receiving too many alerts (caused by the high false positive rate), they can reduce the threshold. In this case, given the results from the consideration of the ROC curves (Section 2.6.2) where the threshold changes, extrapolation would still be preferred.

A.3.7 Effect of Magnitudes of Outliers



(a) BCR of functional depth with ARIMA extrapolation over a range of (negative) outlier magnitudes

(b) BCR of functional depth with ARIMA extrapolation over a range of (positive) outlier magnitudes

(c) Average hindsight performance of functional depth with ARIMA extrapolation over a range of outlier magnitudes

(d) Average foresight performance of functional depth with ARIMA extrapolation over a range of outlier magnitudes

Figure A.3.9: Effects of magnitude of demand outliers on functional depth with ARIMA extrapolation outlier detection

A.3.8 Relationship between extrapolation accuracy and outlier detection improvement

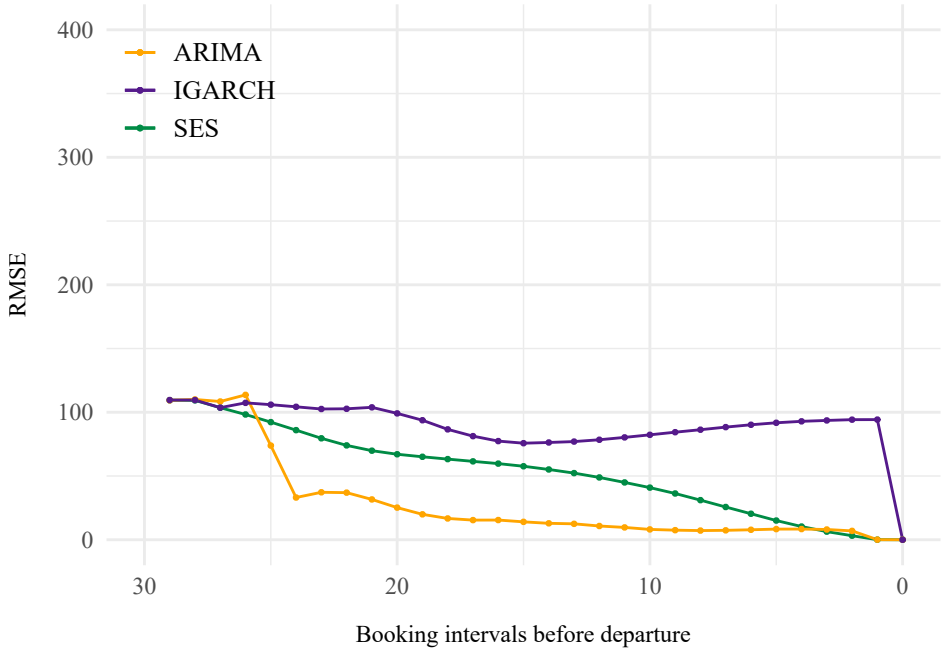


Figure A.3.10: RMSE of different extrapolation methods

To investigate the relationship between the accuracy of the extrapolation and the improvement in outlier detection from an extrapolation method, we computed the average root mean square error (RMSE) of each of the extrapolation methods across the booking horizon – see Figure A.3.10. The RMSE of each individual method means little on its own. As we have increased data available to input to our forecast and are forecasting fewer steps ahead, it is of little surprise that the RMSE decreases over time. However, from the comparison of the RMSE of the different extrapolation methods, we gain some insight into the performance of the outlier detection when using that method. Generally, ARIMA forecasts have the lower RMSE of the methods, and

also provide the largest gain in performance overall when used as the extrapolation method. The exception to this is the IGARCH model where the RMSE has a slight increase in the later part of the booking horizon. This is most likely due to the fact that we have fixed the order of the IGARCH model to be (1,d,1) for computational reasons and are therefore imposing a variance structure in the forecast that does not exist in the data. It is interesting to note that despite the poorer performance of the IGARCH forecast, it still provides a reasonable improvement in outlier detection performance as an extrapolation method.

A.3.9 Comparison of Methods for Hindsight Detection of Demand-volume Outliers

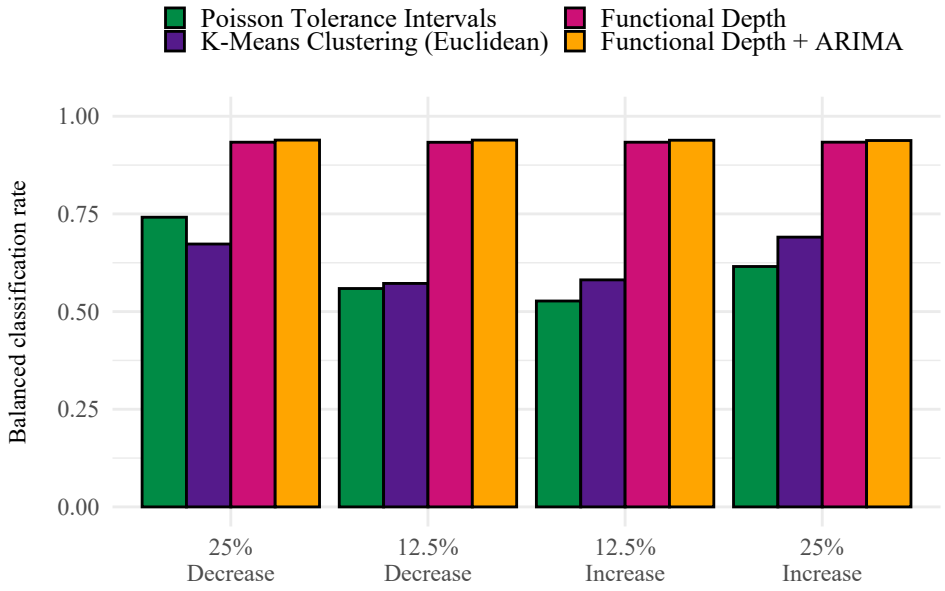


Figure A.3.11: Comparison of hindsight outlier detection under different magnitudes of demand outliers with 5% outlier frequency

For *hindsight detection* performance, we rely on the BCR averaged across all booking intervals. As shown in Figure A.3.11, hindsight detection performance typically increases as the complexity of the outlier detection method increases across all categories of outliers tested. These results are consistent with those for foresight detection. Figure A.3.11 shows that including the extrapolation step induces only a small improvement in hindsight detection performance. However, outliers are detected early in the horizon, meaning any actions taken as a result of their identification will have a significant positive impact in terms of revenue overall, both within and beyond the booking horizon.

Within the revenue management process, identifying outliers and adjusting controls as early as possible provides the most benefit. Nevertheless, even detecting outliers in hindsight promises some advantages over not identifying them at all.

A.3.10 Additional Analysis of Railway Booking Patterns

Here, we compare the simulated booking patterns with those from the railway company. Note that both the railway and simulated booking patterns in Figure A.3.12 have been rescaled to be between 0 and 1. Therefore, although it may appear that the variance of the railway booking patterns is much higher than that of the simulated patterns, it is not necessarily significant (given the rescaling only transforms the mean, not the variance of the booking patterns). The main takeaway from Figure A.3.12 is the similar shape of the booking patterns – starting with a steep increase, followed by a slight flattening out, then another increase.

In order to compare the simulated booking patterns with the railway booking

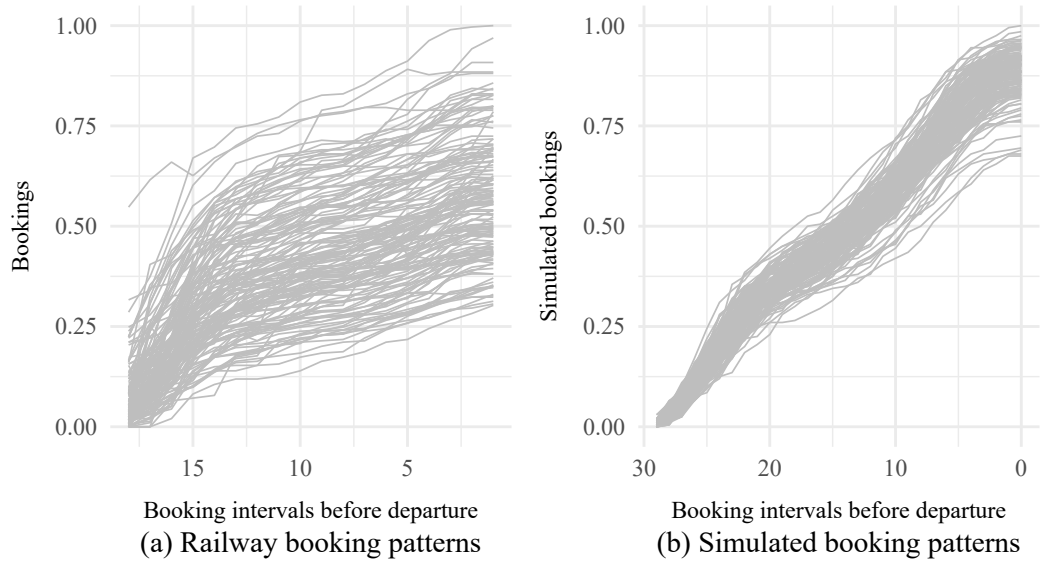


Figure A.3.12: Railway vs simulated booking patterns

patterns, we analyse the relationship between the mean and standard deviation of bookings across the horizon. Figure A.3.13a shows the standard deviation divided by the mean number of bookings in the railway booking data, and Figure A.3.13b analogously for the simulated booking patterns. The two figures show a similar shape – higher at the start of the horizon, then quickly flattening out. The values of the standard deviation / mean are also of a similar magnitude.

As discussed in Section 2.6.4, we compare booking patterns for different days of the week by applying pairwise functional ANOVA tests (Cuevas et al., 2004). We test the null hypothesis that, for two different days m and n, their mean functions are equal:

$$H_0 : \mu_m(t) = \mu_n(t), \text{ vs. } H_A : \mu_m(t) \neq \mu_n(t), \tag{A.3.2}$$

The p-values are shown in Table A.3.2. The only non-significant p-values are for comparison between Monday-Wednesday and Friday-Saturday. However, the p-values

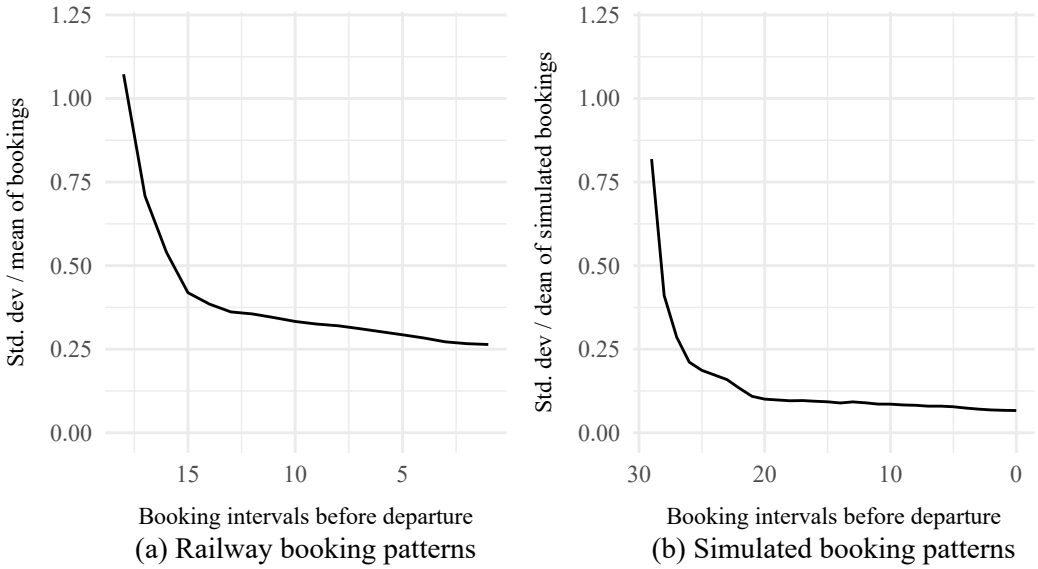


Figure A.3.13: Standard deviation / Mean of railway vs simulated booking patterns

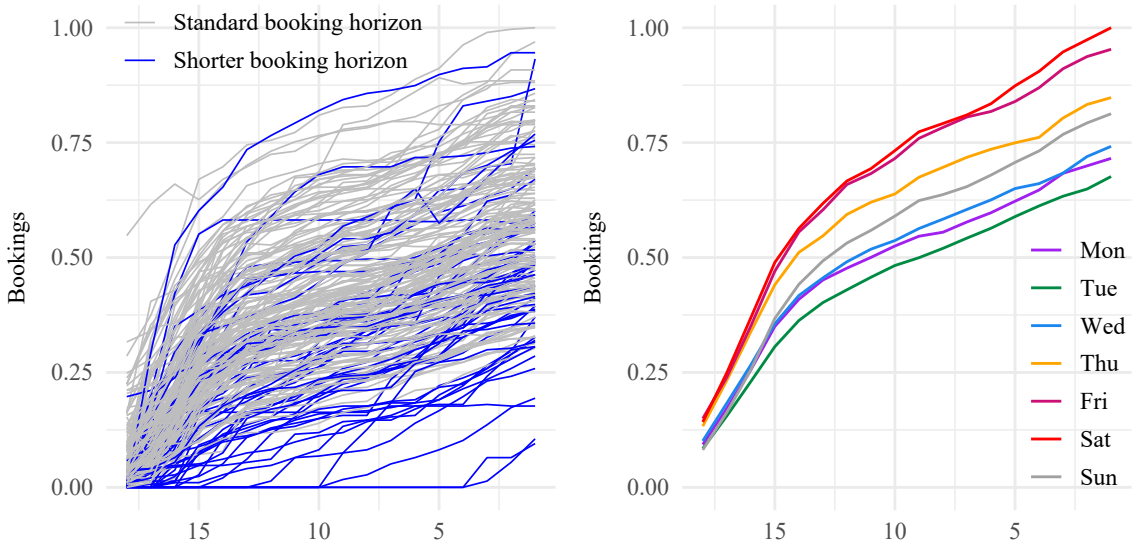
are not overly convincing, especially when considering multiple testing issues, so we choose to model each departure day separately. A similar comparison can be made between booking patterns which are affected by the shortened booking horizons (see Figure A.3.14a), and those of standard length. In that test, all of the p-values were 0.

We account for both the shortened booking horizons and the effect of different departure days through fitting a functional regression model, as per Equation (12). Figure A.3.14b shows the regression curves for each day of the week (without shortened booking horizon effects). The functional regression model works by fitting a linear regression at each time point. That is a different value of at each booking interval. In order to make the $\beta_j(t)$ smooth functions, we penalise the integrated square error such that we seek to minimise (Ramsay et al., 2009):

$$\sum_{i=1}^{n} \int (y_i(t) - \hat{y}_i(t))^2 dt + \sum_{j=0}^{7} \lambda_j \int [L_j \beta_j]^2 dt, \tag{A.3.3}$$

	Mon	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
Mon							
Tue	0.001						
Wed	**0.093**	0.000					
Thu	0.000	0.000	0.000				
Fri	0.000	0.000	0.000	0.000			
Sat	0.000	0.000	0.000	0.000	**0.122**		
Sun	0.000	0.000	0.000	0.001	0.000	0.000	

Table A.3.2: p-values for functional ANOVA test



(a) Shorter booking horizons in railway booking patterns

(b) Functional regression curves for departure days of the week

Figure A.3.14: Functional regression to homogenise booking patterns

where

$$\hat{y}_i(t) = \beta_0(t) + \beta_1(t)I_{Monday_i} + \beta_2(t)I_{Tuesday_i} + \beta_3(t)I_{Wednesday_i} + \beta_4(t)I_{Thursday_i} + \beta_5(t)I_{Friday_i} + \beta_6(t)I_{Saturday_i} + \beta_7(t)I_{Shorter\ Horizon_i}. \tag{A.3.4}$$

and $lambda_j$ a non-negative real number controlling the amount of smoothing, and L_j is either a non-negative integer or a linear differential operator object. Due to the relatively short nature of the booking patterns (18 observations), for this data set we use a smoothing parameter of $\lambda_j = 0 \forall j$.

Appendix B

Appendix: Detecting outlying demand in multi-leg bookings for transportation networks

B.1 Additional details of method

Appendix B.1 provides additional details on the proposed method described in Section 3.2, including the specifics of the correlation-based minimum spanning tree clustering, and the calculation of the functional depths.

B.1.1 Functional dynamical correlation

Let $y_{n,ij}(t)$ be the total observed bookings for the n^{th} departure on leg ij up to booking interval t, and similarly for $y_{n,jk}(t)$. The functional dynamical correlation between the

booking patterns $y_{n,ij}(t)$ and $y_{n,jk}(t)$ is:

$$\rho_n(ij, jk) = \mathbb{E}\langle y^*_{n,ij}(t), y^*_{n,jk}(t)\rangle. \tag{B.1.1}$$

where

$$\langle y^*_{n,ij}(t), y^*_{n,jk}(t)\rangle = \int y^*_{n,ij}(t) y^*_{n,jk}(t) w(t) dt, \tag{B.1.2}$$

and $w(t)$ is a weight function that accounts for the time gap between observations. Here, $y^*_{n,ij}(t)$ is a standardised version of $y_{n,ij}(t)$:

$$y^*_{n,ij}(t) = \frac{y_{n,ij}(t) - M_{ij} - \mu_{ij}(t)}{\left[\int \{y_{n,ij}(t) - M_{ij} - \mu_{ij}(t)\}^2 w(t) dt\right]^{1/2}}, \tag{B.1.3}$$

where $\mu_{ij}(t)$ is a mean function, and:

$$M_{ij} = \langle y_{n,ij}(t), 1\rangle. \tag{B.1.4}$$

The functional dynamical correlation is then the average across all N departures:

$$\rho(ij, jk) = \frac{1}{N} \sum_{n=1}^{N} \rho_n(ij, jk). \tag{B.1.5}$$

B.1.2 Prim's algorithm

Prim's algorithm is a greedy algorithm with the following basic steps. Assuming the original graph G has $V(G)$ vertices.

- Initialise the MST, T, with the edge with minimum weight and the two vertices it connects. Let $V(T)$ be the number of edges in T.
- While $V(T) < V(G)$:

- go through the remaining edges in G in order from smallest to largest weights, until one is found that is connected to T, but does not form a circuit (i.e. the edge does not form a loop such that T is no longer a tree).
- Add this edge (and the vertices it connects) to T.

More computationally efficient algorithms exist but given the reasonable size of the graphs considered, and more specifically their sparsity (very few stations are adjacent), computational time is reasonable using Prim's algorithm.

B.1.3 Functional depth

The functional halfspace depth is given by:

$$d_{nl}(\boldsymbol{y}_{nl} \in \mathcal{Y}_l; \alpha) = \sum_{j=1}^{T} w_\alpha(t_j) HD_j(\boldsymbol{y}_{nl}(t_j)), \tag{B.1.6}$$

where, using $t_{\tau+1} = t_\tau + 0.5(t_\tau - t_{\tau-1})$, the weights $w_\alpha(t_j)$ are, according to Hubert et al. (2012):

$$w_\alpha(t_j) = \frac{(t_{j+1} - t_j)\text{vol}\left[\left\{\boldsymbol{x} \in \mathbb{R}^k : HD_j(\boldsymbol{x}) \geq \alpha\right\}\right]}{\sum_{j=1}^{T}(t_{j+1} - t_j)\text{vol}\left[\{\boldsymbol{x} \in \mathbb{R}^k : HD_j(\boldsymbol{x}) \geq \alpha\}\right]}, \tag{B.1.7}$$

where $\alpha \in (0, 0.5]$, with a default value of $\alpha = 1/T$. The sample halfspace depth of a K-variate vector x at time t_j is given by (Hubert et al., 2012):

$$HD_j(y_{nl}(t_j)) = \frac{1}{N} \min_{\boldsymbol{u}, ||\boldsymbol{u}||=1} \#\left\{y_{nl}(t_j), n = 1, \ldots, N : \boldsymbol{u}^T y_{nl}(t_j) \geq \boldsymbol{u}^T \boldsymbol{x}\right\} \tag{B.1.8}$$

B.1.4 Normalised Mutual Information

For a graph containing M legs, the mutual information between two clusterings $\mathcal{A}$ and $\mathcal{B}$ of the M nodes in the inverted graph is defined as:

$$I(\mathcal{A},\mathcal{B}) = \sum_{a=1}^{|\mathcal{A}|}\sum_{b=1}^{|\mathcal{B}|}\frac{|\mathcal{A}\cap\mathcal{B}|}{M}\log\left(|\mathcal{A}\cap\mathcal{B}|\frac{M}{M_a M_b}\right), \tag{B.1.9}$$

where M_a is the number of nodes in the a^{th} cluster of clustering $\mathcal{A}$, and similarly for M_b. The **normalised mutual information (NMI)** between two clusterings is defined as (Amelio and Pizzuti, 2015):

$$NMI(\mathcal{A},\mathcal{B}) = \frac{2I(\mathcal{A},\mathcal{B})}{H(\mathcal{A})+H(\mathcal{B})}, \tag{B.1.10}$$

where $H(\mathcal{A})$ is the entropy (a measure of uncertainty) defined as:

$$H(\mathcal{A}) = -\sum_{a=1}^{|\mathcal{A}|}\frac{M_a}{M}\log\left(\frac{M_a}{M}\right). \tag{B.1.11}$$

$NMI(\mathcal{A},\mathcal{B}) = 1$ if $\mathcal{A}$ and $\mathcal{B}$ are identical, and 0 if they are completely different.

B.2 Details of computational study

Appendix B.2 contains additional details of the simulation set up described in Section 3.3, including the computation of the bid prices, and a validation of the chosen parameter values.

B.2.1 Dynamic programming for bid price control

From Talluri and Van Ryzin (2004), let x be the remaining capacity, and define $V_t(x)$ denote the value function at time t. Define $R(t)$:

$$R(t) = \begin{cases} r_j & \text{if request for fare class } j \text{ arrives in interval } t \\ 0 & \text{otherwise} \end{cases} \tag{B.2.1}$$

where r_j denotes the revenue from accepting a request for fare class j. The probability that $R(t) = r_j$ is equal to the arrival rate for fare class j at time t. Note the arrival rates are such that at most one request arrives in each time period. Define:

$$u = \begin{cases} 1 & \text{if request for fare class } j \text{ arrives } \textbf{and} \text{ is accepted} \\ 0 & \text{otherwise} \end{cases} \tag{B.2.2}$$

We wish to maximise the combined revenue in the current time period, and the revenue to come in future time periods:

$$\max_{u \in \{0,1\}} \left(R(t)u + V_{t+1}(x-u) \right) \tag{B.2.3}$$

The Bellman equation for $V_t(x)$ is:

$$V_t(x) = \mathbb{E}\Big[\max_{u\in\{0,1\}}\{R(t)u + V_{t+1}(x-u)\}\Big] \tag{B.2.4}$$

$$= V_{t+1}(x) + \mathbb{E}\Big[\max_{u\in\{0,1\}}\{(R(t) + \Delta V_{t+1}(x))u\}\Big] \tag{B.2.5}$$

$$V_t(x) = \sum_{j=1}^{|\mathcal{J}|}\lambda_j(t) max\{(r_j - \Delta V_{t+1}(x)), 0\} \tag{B.2.6}$$

where $\lambda_j(t)$ is the arrival rate of demand for fare class j in interval t, and $\Delta V_{t+1}(x) = V_{t+1}(x) - V_{t+1}(x-1)$ is the marginal cost of capacity in the next time period. The problem is solved with backwards recursion, with the following boundary conditions apply:

$$V_{T+1}(x) = 0, \qquad x = 0, 1, \ldots, C \tag{B.2.7}$$

$$V_t(0) = 0, \qquad t = 1, \ldots, T \tag{B.2.8}$$

These ensure (i) no revenue can be generated beyond the booking horizon i.e after departure; and (ii) that no further revenue can be generated if there is no capacity remaining. The bid price at time t with remaining capacity x is given by $\Delta V_t(x)$.

B.2.2 Details of benchmark method

We use the method proposed by Hyndman et al. (2016) as a benchmark comparison for our proposed method in Section 3.3. The method works as follows:

- Define the total demand booking patterns as the sum of the demand for each leg within the cluster.
- Compute f features of the n total demand booking patterns. Features include: mean, variance, first order autocorrelation, trend, linearity, seasonality, peak, trough, entropy, lumpiness, spikiness, change in variance, Kullback-Leibler score, among others. See Hyndman et al. (2016) for a full list.
- Apply principal component analysis (PCA) as per Yang and Shahabi (2004) to determine the first two principle components i.e. those that explain the most variance.
- Use a density-based multi-dimensional approach (Hyndman, 1996) to finds points in the first two principal components with lowest density.
- The nu points with the lowest densities relate to the departures which are classified as outliers.

B.2.3 Parameter values for simulation study

Table B.2.1: Regular demand generation parameter values

Parameter	Value	Effect of parameter
$\boldsymbol{\alpha} = \{\alpha_{AB}, \alpha_{AC}, \alpha_{AD}, \alpha_{AE}, \alpha_{BC}, \alpha_{BD}, \alpha_{BE}, \alpha_{CD}, \alpha_{CE}, \alpha_{DE}\}$	$\boldsymbol{\alpha} = \{32, 14, 14, 180, 4, 4, 14, 4, 14, 32\}$	Parameters of the Gamma distribution which controls the level of total demand across all fare classes and customer types such that the mean demand for itinerary o is: $\mathbb{E}(D_o) = \frac{\alpha_o}{\beta_o}$.
$\boldsymbol{\beta} = \{\beta_{AB}, \beta_{AC}, \beta_{AD}, \beta_{AE}, \beta_{BC}, \beta_{BD}, \beta_{BE}, \beta_{CD}, \beta_{CE}, \beta_{DE}\}$	$\boldsymbol{\beta} = \{1, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$	
$\boldsymbol{a_1} = \{a_{1,AB}, a_{1,AC}, a_{1,AD}, a_{1,AE}, a_{1,BC}, a_{1,BD}, a_{1,BE}, a_{1,CD}, a_{1,CE}, a_{1,DE}\}$	$\boldsymbol{a_1} = \{5, 5, 5, 5, 5, 5, 5, 5, 5, 5\}$	Parameters of Beta distribution which controls the arrival times of type 1 customers
$\boldsymbol{b_1} = \{b_{1,AB}, b_{1,AC}, b_{1,AD}, b_{1,AE}, b_{1,BC}, b_{1,BD}, b_{1,BE}, b_{1,CD}, b_{1,CE}, b_{1,DE}\}$	$\boldsymbol{b_1} = \{2, 2, 2, 2, 2, 2, 2, 2, 2, 2\}$	

Parameter	Value	Effect of parameter
$\boldsymbol{a_2} = \{a_{2,AB}, a_{2,AC}, a_{2,AD}, a_{2,AE}, a_{2,BC}, a_{2,BD}, a_{2,BE}, a_{2,CD}, a_{2,CE}, a_{2,DE}\}$	$\boldsymbol{a_2} = \{2, 2, 2, 2, 2, 2, 2, 2, 2, 2\}$	Parameters of Beta distribution which controls the arrival times of type 2 customers
$\boldsymbol{b_2} = \{b_{2,AB}, b_{2,AC}, b_{2,AD}, b_{2,AE}, b_{2,BC}, b_{2,BD}, b_{2,BE}, b_{2,CD}, b_{2,CE}, b_{2,DE}\}$	$\boldsymbol{b_2} = \{2, 3, 5, 7, 2, 3, 5, 2, 3, 2\}$	
$\boldsymbol{p_{1jo}} = \{p_{1Ao}, p_{1Oo}, p_{1Jo}, p_{1Po}, p_{1Ro}, p_{1So}, p_{1Mo}\}$	$\boldsymbol{p_{1jo}} = \{0.30, 0.25, 0.20, 0.15, 0.10, 0, 0\}$	Probability of purchase for each customer type. It is assumed these are constant across itineraries. The no-purchase probability for customer type i is equal to $1 - \sum_{j \in \mathcal{J}} p_{ijo}$.
$\boldsymbol{p_{2jo}} = \{p_{2Ao}, p_{2Oo}, p_{2Jo}, p_{2Po}, p_{2Ro}, p_{2So}, p_{2Mo}\}$	$\boldsymbol{p_{2jo}} = \{0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.25\}$	
$\boldsymbol{\phi_o} = \{\phi_{1,o}, \phi_{2,o}\}$	$\boldsymbol{\phi_o} = \{0.5, 0.5\} \forall o$	Proportion of total demand from each customer type for each itinerary. It is assumed these are constant across itineraries.

Outliers considered in computational study

Table B.2.2 shows the different experiments that were carried out as part of the computational study. We consider *cluster* outliers in which every itinerary within the cluster is equally affected; *itinerary* outliers where only a single itinerary within the cluster is affected; and *station* outliers which affect all itineraries that end at a particular station.

Experiment	Outlier Type	Itineraries Affected	Magnitudes
1	**Cluster**	All	+10%, +20%, +30%, +40%, +50%, +60%, -10%, -20%, -30%, -40%, -50%, -60%
2	**Itinerary**	AB	+50%
3		AC	+50%
4		AD	+50%
5		AE	+50%
6		BC	+50%
7		BD	+50%
8		BE	+50%
9		CD	+50%
10		CE	+50%
11		DE	+50%
12	**Station**	AB	+50%
13		AC, BC	+50%
14		AD, BD, CD	+50%
15		AE, BE, CE, DE	+50%

Table B.2.2: Different types of outliers considered in computational study

B.3 Computational results

Appendix B.3 includes the extended results from the computational study described in Section 3.3. Results from additional simulation experiments to test the proposed clustering approach are also presented here.

B.3.1 Evaluation of network clustering

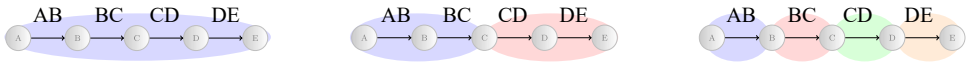
For the correlation-based clustering to perform well it needs to (i) accurately estimate similarity between adjacent legs, and (ii) use information about pairwise similarity between adjacent legs to detect similarity between (potentially) more than two legs to form clusters. We use the proportion of total demand belonging to each itinerary to determine a clustering benchmark. For example, in Figure B.3.1a, when *all* passengers travel the itinerary from A to E, the resulting bookings in each of the four legs would be identical. In this case, the correlation between legs would be 1 – giving a single cluster of four legs.

To evaluate the clustering when the underlying demand is known, we define the **common traffic ratio** between two adjacent legs as the proportion of total demand that relates to itineraries over both legs. That is, for two legs ij and jk, we define the common traffic ratio, $r(ij, jk)$, to be:

$$r(ij, jk) = \frac{D_{ik}}{D_{ij} + D_{jk} + D_{ik}}, \tag{B.3.1}$$

where D_{ij} is the demand for itinerary ij, and D_{ik} is the total demand for all itineraries which include both legs ij and jk. If all passengers book itineraries that traverse both legs, then $r(ij, jk) = 1$. Conversely, if no passengers book journeys that traverse both

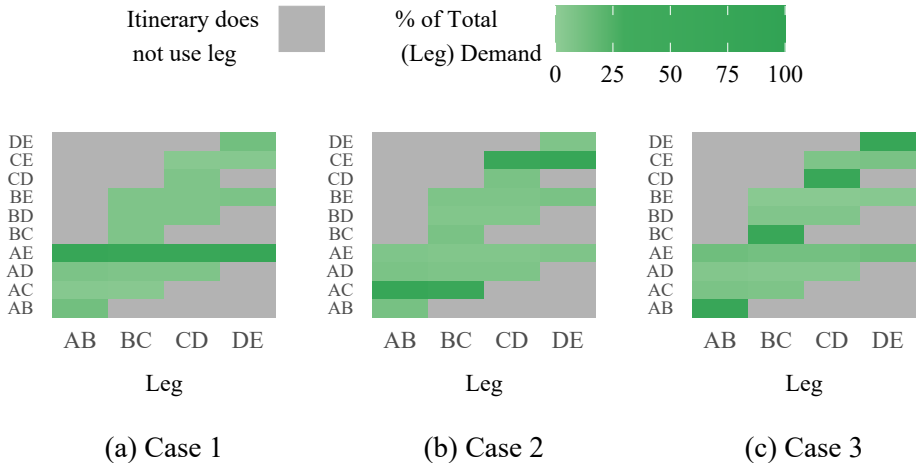
legs, then $r(ij, jk) = 0$.



(a) Case 1 (b) Case 2 (c) Case 3

Figure B.3.1: Benchmark clustering

We vary the level of demand for each itinerary to generate different benchmark clusterings. The output of the correlation-based clustering is then compared with benchmark clustering using the NMI. We consider three cases: the four legs belong in a single cluster (Figure B.3.1a); they belong in two clusters (Figure B.3.1b); and they belong in four clusters (Figure B.3.1c).



(a) Case 1 (b) Case 2 (c) Case 3

Figure B.3.2: Itinerary demand per leg

- **Case 1**: When itinerary AE accounts for at least 50% of the network demand, we expect legs AB, BC, CD, and DE to belong to the same cluster, as they experience mostly the same demand. Remaining demand is calibrated across

itineraries such that total demand for each leg is reasonably uniformly distributed. We compare the correlation-based clustering with the benchmark clustering of all four legs in a single cluster, when the average percentage of demand on each leg from itinerary AE is 50%, 60%, 70%, 80%, 90%, or 100%. Figure B.3.2a shows the fraction of total demand on each leg, from each itinerary, in the case where 60% of demand is for itinerary AE.

- **Case 2**: We calibrate the majority of demand on leg AB and BC to be for itinerary AC, and the majority of demand on legs CD and DE to be demand for itinerary CE. For simplicity, the distribution of demand is symmetric across the four legs. We compare the performance when the average percentage of demand on each leg belonging to the clustering benchmark itinerary is 50%, 60%, 70%, 80%, 90%, or 100%. Figure B.3.2b shows the case where 60% of demand on each leg is for the respective cluster itineraries (AC or CE).

- **Case 3**: We calibrate the majority of demand on leg AB for itinerary AB, the majority of demand on leg BC for itinerary BC, and so on. We compare the performance when the average percentage of demand on each leg belonging to the leg itinerary is 50%, 60%, 70%, 80%, 90%, or 100%. Figure B.3.2c shows the case where 60% of demand on each leg is for the itinerary consisting of only that leg.

The results are shown in Table B.3.1.

In almost all cases, the normalised mutual information between the correlation-based clustering and the benchmark equals 1, indicating congruence. We now extend

	Fraction of Leg Demand Resulting from Cluster Itinerary Demand					
	50%	**60%**	**70%**	**80%**	**90%**	**100%**
Case 1	0.99	1.00	1.00	1.00	1.00	1.00
Case 2	0.98	0.99	1.00	1.00	1.00	1.00
Case 3	0.94	0.97	0.99	1.00	1.00	1.00

Table B.3.1: Normalised mutual information

the simulation study by comparing the output of the correlation-based clustering under different correlation measures. In additional to the functional dynamical correlation measure described in Section 3.2.1, we compare *Pearson correlation* (Pearson, 1895) and *Kendall rank correlation* (Kendall, 1938). Let $y_{n,ij}(t)$ be the observed bookings for the n^{th} departure on leg ij, and $y_{n,pq}(t)$ analogous for leg pq.

- **Pearson correlation**: calculate the Pearson correlation between corresponding booking patterns, then average across all booking patterns. That is, for the n^{th} of N booking patterns observed over T booking intervals, we calculate the Pearson correlation coefficient as:

$$\rho_n(ij,pq) = \frac{\sum_{t=1}^{T}(y_{n,ij}(t) - \overline{y_{n,ij}})(y_{n,pq}(t) - \overline{y_{n,pq}})}{\sqrt{\sum_{t=1}^{T}(y_{n,ij}(t) - \overline{y_{n,ij}})^2}\sqrt{\sum_{t=1}^{T}(y_{n,pq}(t) - \overline{y_{n,pq}})^2}} \tag{B.3.2}$$

  where $\overline{y_{n,ij}}$ is the mean number of bookings for the n^{th} booking pattern. Then:

$$\rho(ij,pq) = \frac{1}{n}\sum_{n=1}^{N}\rho_n(ij,pq). \tag{B.3.3}$$

- **Kendall rank correlation**: observations $(y_{n,ij}(s), y_{n,pq}(s))$ and $(y_{n,ij}(t), y_{n,pq}(t))$ where $s < t$, are *concordant* if their ordering agrees,

and *discordant* otherwise. The Kendall rank correlation is defined between the n^{th} booking patterns in legs ij and pq as:

$$\rho_n(ij, pq) = \frac{t_c - t_d}{\sqrt{(t_0 - t_1)(t_0 - t_2)}} \tag{B.3.4}$$

where t_c is the number of concordant pairs, t_d is the number of discordant pairs, and t_0, t_1, and t_2 are defined as follows:

$$t_0 = \frac{T(T-1)}{2}, \tag{B.3.5}$$

$$t_1 = \sum_s u_s(u_s - 1)/2, \tag{B.3.6}$$

$$t_2 = \sum_t v_t(v_t - 1)/2, \tag{B.3.7}$$

where u_s is the number of tied values in the s^{th} group of ties for in booking patterns for leg ij, and v_t is analogous for leg pq. Then:

$$\rho(ij, pq) = \frac{1}{n}\sum_{n=1}^{N} \rho_n(ij, pq). \tag{B.3.8}$$

We compare the cases where the correlation measure is (i) applied directly to the booking patterns, and (ii) applied to the differenced booking patterns where the within-booking pattern relationships e.g. trend have been removed. The normalised mutual information between the clustering produced by the correlation-based clustering under each of the different correlation measures, and the benchmark clustering is shown in Table B.3.2.

For case 1, all three correlation measure seem to be performing equally well, with the normalised mutual information almost always indicating congruence. For cases 2 and 3, the Pearson and Kendall correlation results in extremely poor performance

Case	Correlation Measure	Fraction of Leg Demand Resulting from Cluster Itinerary Demand					
		50%	**60%**	**70%**	**80%**	**90%**	**100%**
Case 1	Booking patterns						
	Functional dynamical correlation	0.99	1.00	1.00	1.00	1.00	1.00
	Pearson correlation	1.00	1.00	1.00	1.00	1.00	1.00
	Kendall rank correlation	1.00	1.00	1.00	1.00	1.00	1.00
	Differenced booking patterns						
	Functional dynamical correlation	0.99	1.00	1.00	1.00	1.00	1.00
	Pearson correlation	0.98	1.00	1.00	1.00	1.00	1.00
	Kendall rank correlation	1.00	1.00	1.00	1.00	1.00	1.00
Case 2	Booking patterns						
	Functional dynamical correlation	0.98	0.99	1.00	1.00	1.00	1.00
	Pearson correlation	0.00	0.00	0.00	0.00	0.00	0.00
	Kendall rank correlation	0.00	0.00	0.00	0.00	0.00	0.00
	Differenced booking patterns						
	Functional dynamical correlation	0.98	0.99	1.00	1.00	1.00	1.00
	Pearson correlation	0.00	0.00	0.00	0.00	0.00	0.00
	Kendall rank correlation	0.00	0.00	0.00	0.00	0.00	0.00
Case 3	Booking patterns						
	Functional dynamical correlation	0.94	0.97	0.99	1.00	1.00	1.00
	Pearson correlation	0.00	0.00	0.00	0.00	0.00	0.00
	Kendall rank correlation	0.00	0.00	0.00	0.00	0.00	0.00
	Differenced booking patterns						
	Functional dynamical correlation	0.93	0.96	0.99	1.00	1.00	1.00
	Pearson correlation	0.00	0.00	0.00	0.00	0.00	0.00
	Kendall rank correlation	0.00	0.00	0.00	0.00	0.00	0.00

Table B.3.2: Normalised mutual information under different correlation measures

in terms of NMI, with the benchmark clustering never being achieved. Functional dynamical correlation, however, continues to perform well with an NMI close to 1.

In order to determine why the Pearson and Kendall rank correlations initially appear to perform well in the single cluster case, but fail in the two cluster case, we also compare the value of the correlation coefficient with the known demand share in a simple two leg example. Consider the simple two leg network shown in Figure B.3.3.

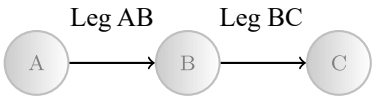


Figure B.3.3: Network with two legs

The common traffic ratio of legs AB and BC is:

$$r(AB, BC) = \frac{D_{AC}}{D_{AB} + D_{BC} + D_{AC}}, \tag{B.3.9}$$

If $r(AB, BC) = 1$, then the number of bookings on leg AB and leg BC are identical, and the correlation between them is 1. Conversely, if $r(AB, BC) = 0$, then the bookings on leg AB and leg BC are independent with correlation 0. Table B.3.3 shows the estimates of the correlation, compared to the true ratio, $r(AB, BC)$.

Functional dynamical correlation, applied directly to the data, performs best in all cases. In case 1, where the benchmark clustering is a single cluster, poor clustering performance can only result from under-estimating the demand share. Both Pearson and Kendall rank correlation over-estimate the correlation between booking patterns, even when the within-booking pattern effects have been removed. This explains the

$r(AB, BC)$	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
Correlation between booking patterns											
Functional dynamical correlation	0.12	0.22	0.35	0.40	0.46	0.55	0.66	0.82	0.86	0.90	1.00
Pearson correlation	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Kendall rank correlation	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	1.00	1.00	1.00
Correlation between differenced booking patterns											
Functional dynamical correlation	0.14	0.18	0.29	0.42	0.50	0.53	0.66	0.83	0.88	0.91	1.00
Pearson correlation	0.70	0.71	0.77	0.82	0.85	0.89	0.92	0.95	0.96	0.98	1.00
Kendall rank correlation	0.88	0.90	0.91	0.91	0.92	0.94	0.94	0.95	0.96	0.97	1.00

Table B.3.3: Comparison of correlation measures

good performance of Pearson and Kendall rank correlation in case 1, despite extremely poor performance in cases 2 and 3.

B.3.2 Detecting outliers in multiple legs

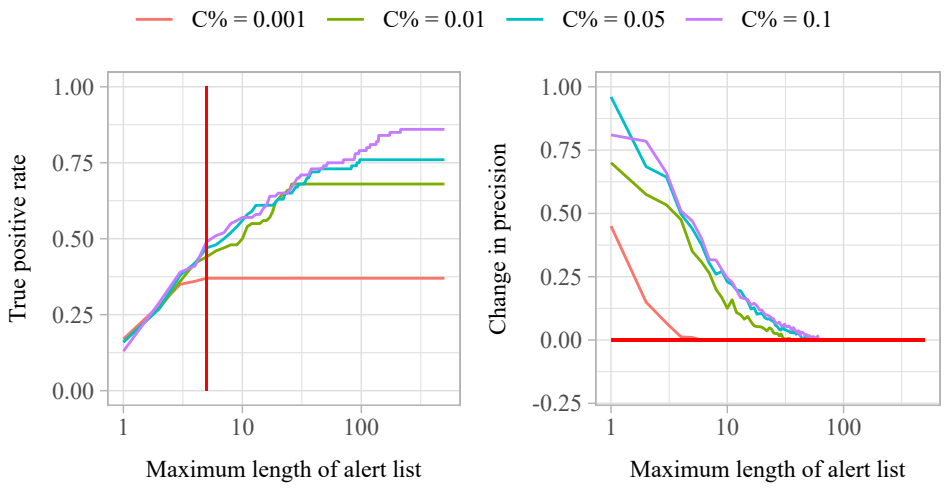
Outlier detection under different functional depth thresholds

We recognise that the percentage of departures that analysts are able to adjust strongly depends on the ratio of analysts to departures, and that this is likely to be domain dependent. Therefore, here we consider outlier detection performance as the functional depth threshold varies.

In terms of true positive rates, the choice of threshold of 0.01, 0.05, or 0.1 produces similar results, at least near the top of the alert list. Our method ranks the departures classified as outliers such that genuine outliers are more likely to be at the top of the ranked list, and false positives at the bottom of the list. Therefore, using a higher

threshold tends to add more departures to the bottom of the list, and increase the risk of more false positives. As shown in Section 4 of the manuscript, the outliers that a threshold of 0.01 fails to detect tend to be small changes in magnitude. It is these small magnitude outliers that are added to the bottom of the list as the threshold increases. Notably, a threshold of 0.001 results in reduced performance even at the top of the list, suggesting this would be too low a threshold. Similar results are seen in the change in precision (compared to the non ranked method with the same threshold).

A higher threshold does result in higher overall true positive rates as more departures are classified as outliers. However, the maximum true positive rate for a threshold of 0.05 results in around 1 in 5 departures being classified as outliers. This is quite a high percentage for them all to be considered *outliers*.

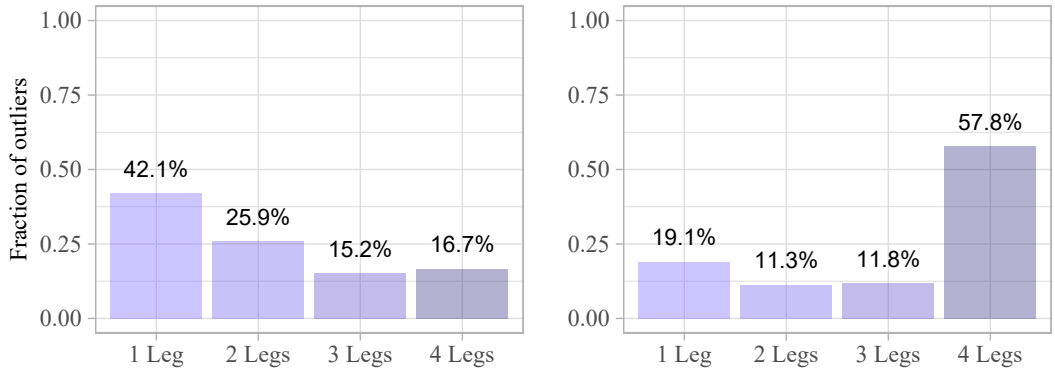


(a) True positive rate (b) Change in precision

Figure B.3.4: Outlier detection performance under different functional depth thresholds

Distribution of outliers across multiple legs

In the scenario where all itineraries are equally affected, a high proportion of outliers should be detected in more than one leg. Figure B.3.5a illustrates the proportion of outliers detected in 1, 2, 3 or 4 legs: More than half were detected in multiple legs. Figure B.3.5b shows the proportion of true positives (genuine outliers which were detected), by the number of legs in which they were detected. In contrast with Figure B.3.5a, a much higher percentage of genuine outliers are detected in all four legs.



(a) Fraction of all outliers detected in 1, 2, 3, or 4 legs

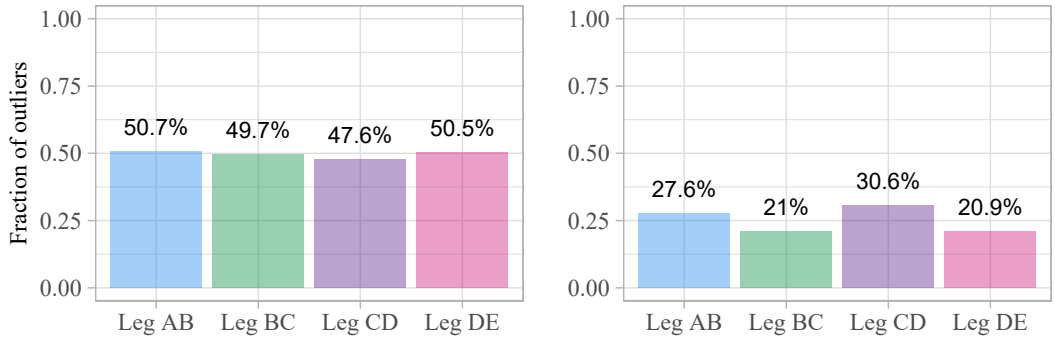
(b) Fraction of true positives detected in 1, 2, 3, or 4 legs

Figure B.3.5: Fraction of outliers detected in 1, 2, 3, or 4 legs

Given the clustering is correct, we expect an approximately equal number of single leg outliers in each leg, as shown in Figure B.3.6b. If one leg, say DE, had not belonged in this cluster, we would expect a higher proportion of single leg outliers to have been detected in leg DE. This could be utilised as a method for checking the clustering, after the outlier detection.

These results motivate aggregating threshold exceedances across legs in two ways: (i) since less than 100% of genuine outliers were detected in all legs, if outlier detection

was carried out only on the leg level, outliers could be missed on some legs. (ii) Given that a much higher proportion of outliers detected in four legs were genuine outliers, by ranking booking patterns detected in all legs as more likely to be outliers, we focus analysts' attention to those more likely to be genuine outliers.



(a) Fraction of outliers detected in each leg, for outliers detected in at least one leg

(b) Fraction of outliers detected in each leg, for outliers detected in only one leg

Figure B.3.6: Fraction of outliers detected in each leg

False Discovery Rate

The *false discovery rate (FDR)* is defined as the proportion of booking patterns classified as outliers which were false positives:

$$FDR = \frac{FP}{TP + FP} \tag{B.3.10}$$

See Section 3.3.6 for definitions of true and false positives. Figure B.3.7 shows the FDR for the case where outlier demand affects all itineraries, and the magnitude is randomly chosen from each of the distributions described in Section 3.3.3.

Figure B.3.8 shows the FDR for each of the magnitudes of outliers considered in the simulation study. Given that smaller magnitude outliers are more similar to the

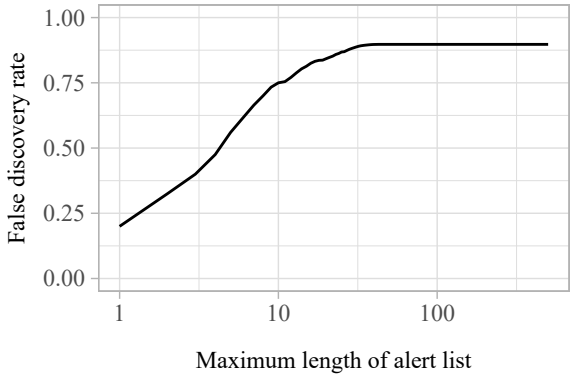
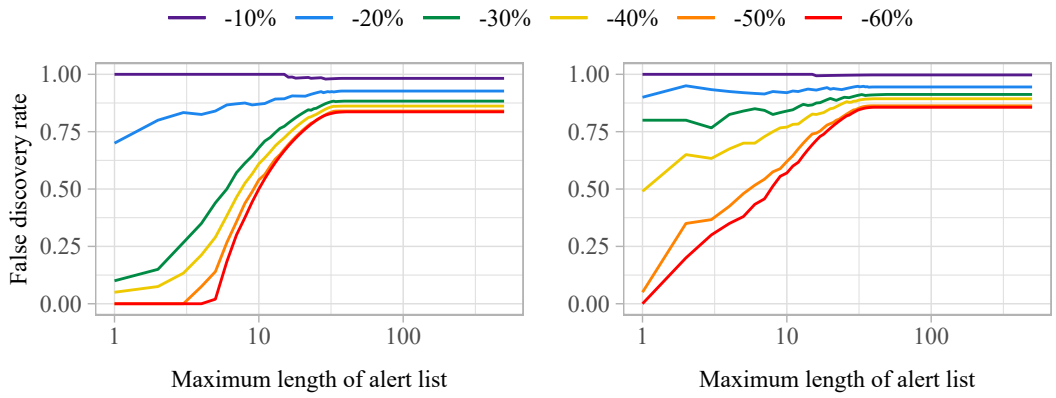


Figure B.3.7: False discovery rate for nonhomogeneous demand-volume outliers

regular demand, these result in higher false discovery rates.



(a) Demand decrease outliers

(b) Demand increase outliers

Figure B.3.8: False discovery rate for homogeneous demand-volume outliers by magnitude

Outliers affecting a single itinerary

Figure B.3.9 shows the true positive rate for the remaining itineraries in Figure 3.3.4 of Section 3.3.6.

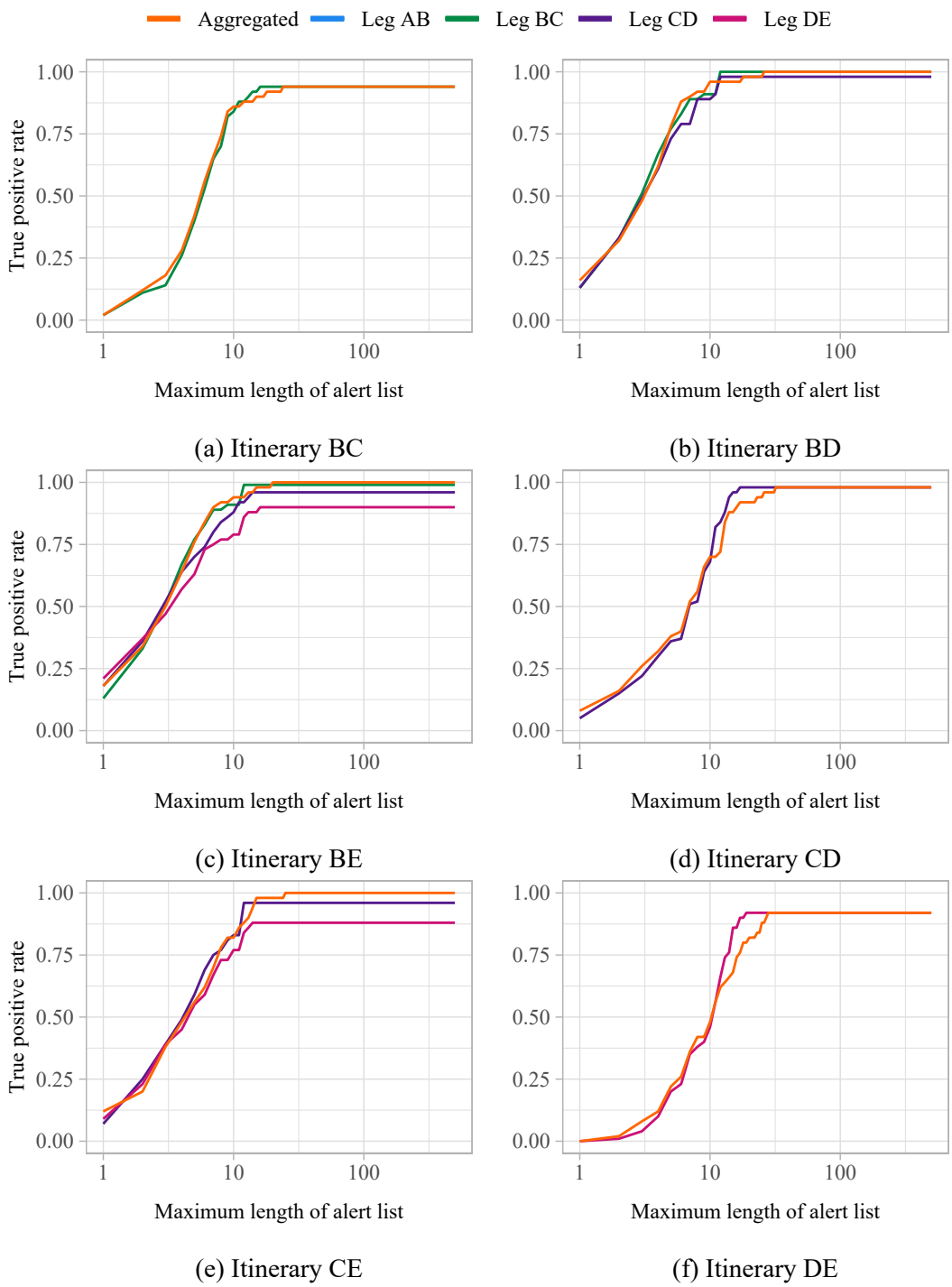


Figure B.3.9: True positive rate for single itinerary outliers (cont.)

Outliers affecting a subset of itineraries

We consider a case where demand outliers affect only a subset of itineraries. Practical examples for this phenomenon could include trade fairs or conventions as well as

regional crises. In such situations, demand towards (or from) a specific destination is most affected. Here, clustering offers additional benefits in guiding analysts towards those itineraries where they should adjust the forecast or controls.

We differentiate four scenarios based on the four-leg-network described in Section 3.3 , where events affect demand for itineraries travelling to stations B, C, D, and E respectively. We expect analogous results when customers aim to travel home from events that happened at stations A, B, C, or D respectively, given the symmetry of the demand parameters chosen for the computational study.

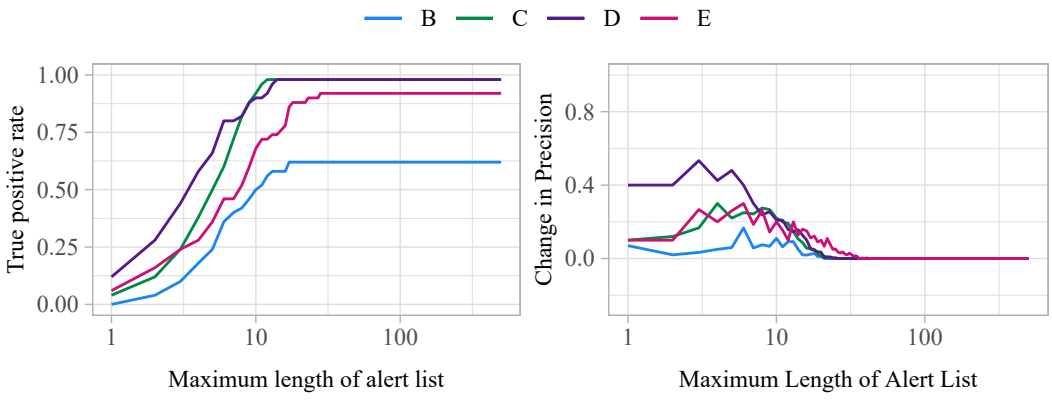
For each of the four possible events considered, we investigate the case where this generates 50% increase in average leg demand. For simplicity, we assume these passengers are equally split between the itineraries which alight at the relevant station. Table B.3.4 shows the resulting demand increases for each leg.

Event at Station	**Itineraries Affected**	**Additional 120 Passengers in Itineraries** Resulting Demand Increase per Leg			
		Leg AB	**Leg BC**	**Leg CD**	**Leg DE**
B	A-B	+120 (+50%)	-	-	-
C	A-C, B-C	+60 (+25%)	+120 (+50%)	-	-
D	A-D, B-D, C-D	+40 (+16.6%)	+80 (+33.3%)	+120 (+50%)	-
E	A-E, B-E, C-E, D-E	+30 (12.5%)	+60 (+25%)	+90 (+37.5%)	+120 (+50%)

Table B.3.4: Changes in leg demand resulting from an additional 120 passengers in itinerary demand

Figure B.3.10a shows the true positive rate for each of the cases. Although the event at E generates outliers in more legs, it is not the case that it has the highest

true positive rate. This shows that though the approach aggregates across legs, it does not ignore outliers only in a subset of those legs, provided they are sufficiently large. These effects may also be caused by interactions between the booking limits on different legs. For example, in the case of an event at C, large increases in demand in legs AB and BC may cause booking limits to be reached earlier for these legs, which also limits bookings in itineraries such as AD and AE. Hence, an increase in demand for some legs may caused a decrease in bookings for different legs. By jointly considering multiple legs for outlier detection, we are able to detect the knock-on effects of outliers even when the change in demand only affects a subset of legs. The change in precision can be interpreted similarly, in Figure B.3.10b.



(a) True positive rate

(b) Change in precision

Figure B.3.10: Performance for demand-volume outliers in a subset of itineraries caused by an absolute increase in demand

Had we considered outlier detection on a leg-by-leg basis, the outliers were more likely to be missed in some of the legs. By combining information across legs, we are better able to determine which itineraries are affecting the volume of demand.

Using outlier severity threshold to limit alert list length

The results in this chapter focus on limiting the length of the ranked alert list simply by the number of alerts it contains as this is most relevant to analysts. However, an alternative approach limits the length of the list by the outlier severity assigned to each departure. For example, classifying a train as an outlier only if its outlier severityy is above 80%.

Detection results when outliers affect all itineraries

Figure B.3.11 shows the true positive rate as the outlier severity decreases from

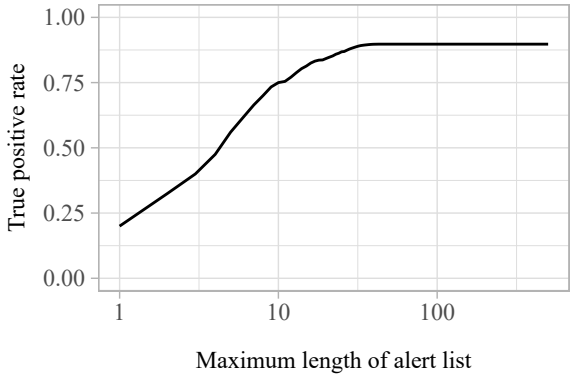
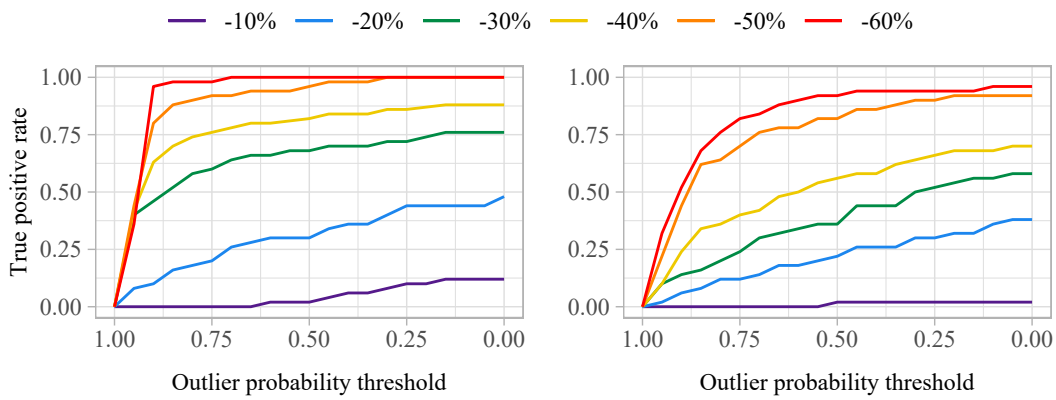


Figure B.3.11: True positive rate for nonhomogeneous demand-volume outliers as minimum outlier severity varies

100% to 0%. Results are similar to those shown in Figure 3.3.1a. Figure B.3.12 shows the true positive rate as the outlier severity decreases from 100% to 0%, for each magnitude of outlier considered. Results are similar to those shown in Figure 3.3.5.

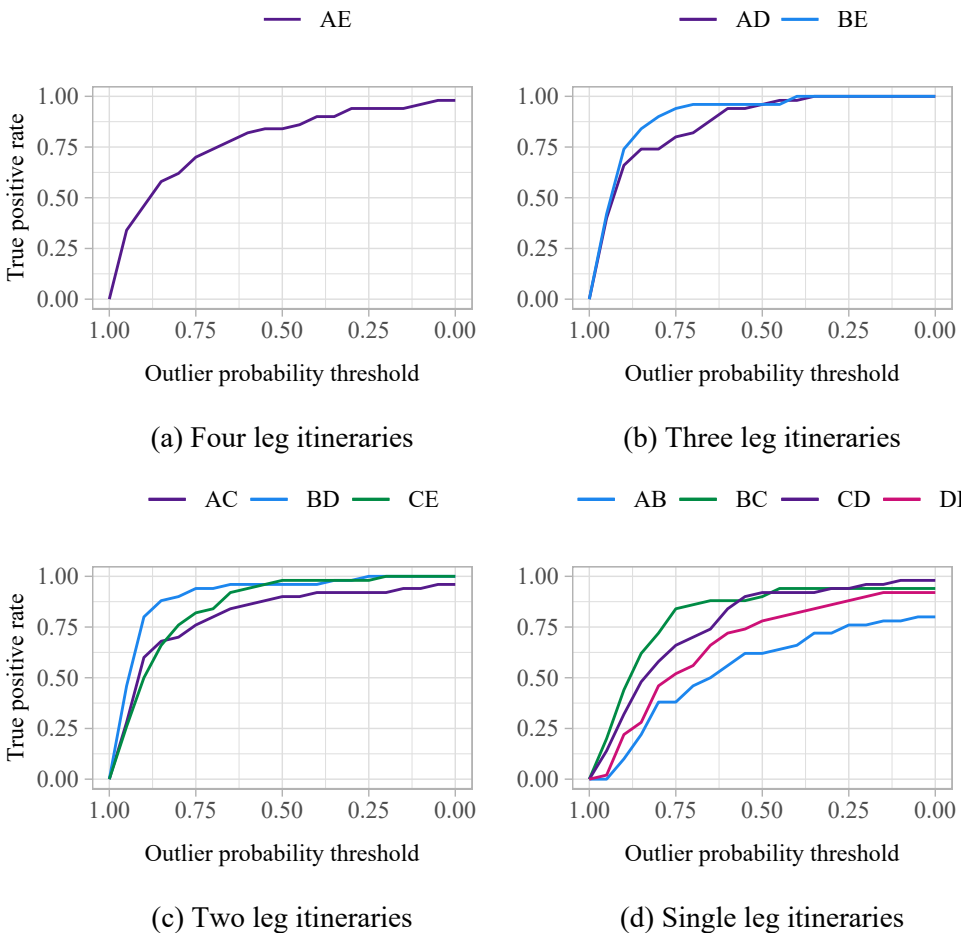


(a) Demand decrease outliers

(b) Demand increase outliers

Figure B.3.12: True positive rate for homogeneous demand-volume outliers by magnitude

Detection results when outliers affect a single itinerary



(a) Four leg itineraries

(b) Three leg itineraries

(c) Two leg itineraries

(d) Single leg itineraries

Figure B.3.13: True positive rate for single itinerary demand-volume outliers as minimum outlier severity varies

B.3.3 Revenue benefits from forecast adjustments for outlier demand

Figure B.3.14 shows the true positive rate for the remaining itineraries in Figure 3.3.7 of Section 3.3.7.

The analysis in Section 3.3.5 constitutes a best-case scenario in which we assume that, if outlier demand affects a particular leg, the outlier is detected in that leg. However, as we show in Section 3.3.6, even when demand outliers affect multiple legs, the outlier is not always detected in every leg due to noise. Therefore, we additionally compare different adjustments based on the output of the outlier detection, for an outlier in itinerary AE.

- **Adjustment A**: Adjust only the forecasts of the affected single-leg itineraries for those legs in which the outlier is detected.
- **Adjustment B**: Adjust the forecasts of the affected single-leg itineraries for those legs in which the outlier is detected, **and** the cluster spanning itinerary (AE).

We compare these both to making no adjustment, and to the oracle adjustment. This is still a best-case scenario to some extent, given that we assume the correct magnitude of adjustment is made.

Figure B.3.15 shows the revenue under adjustments A and B (as described in Section 3.3.5) depending on the output of the outlier detection procedure. Combining adjustments on the leg-level with those on the cluster level provides superior results

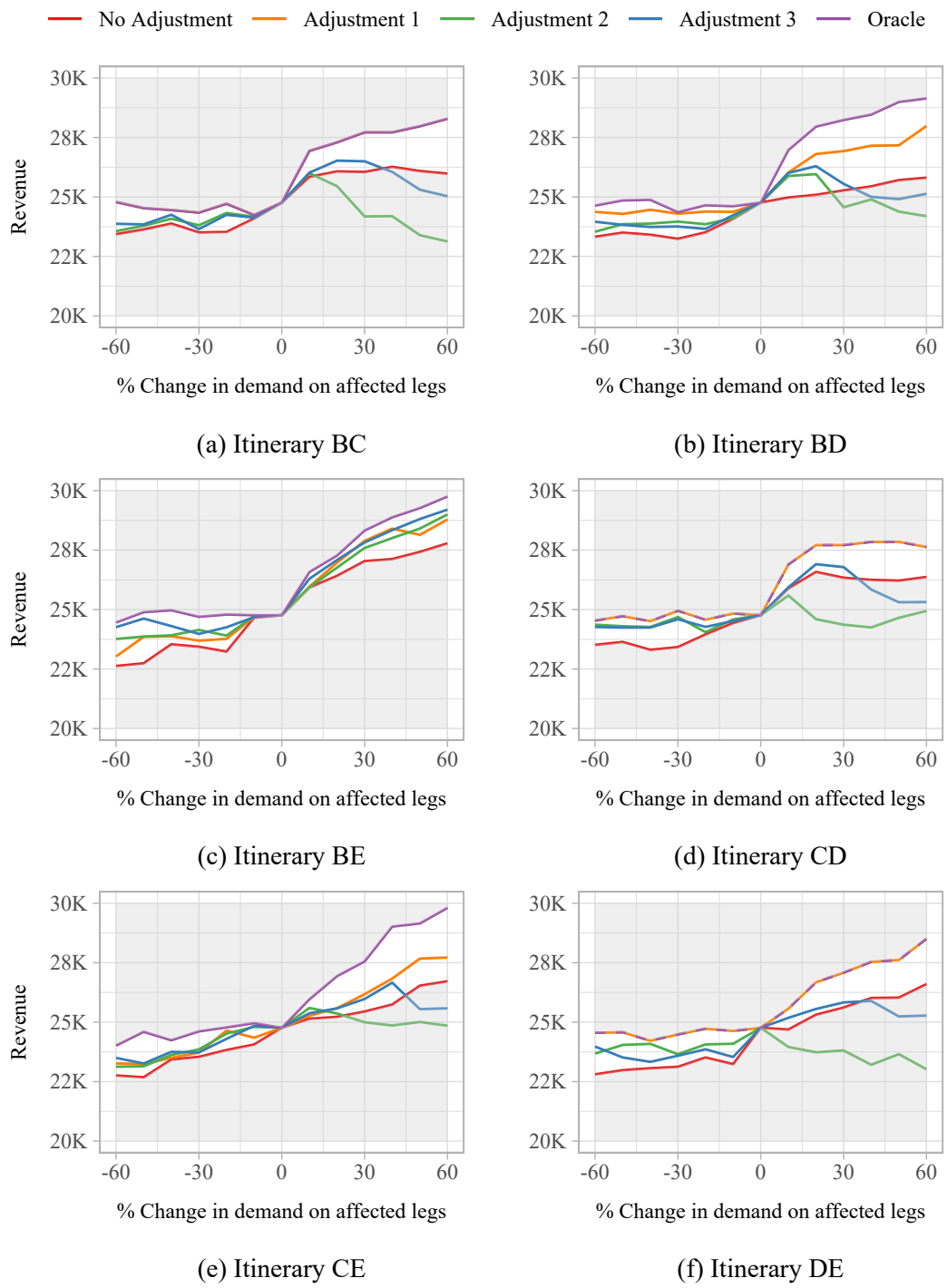


Figure B.3.14: Revenue generated under different itinerary-level forecast adjustments (cont.)

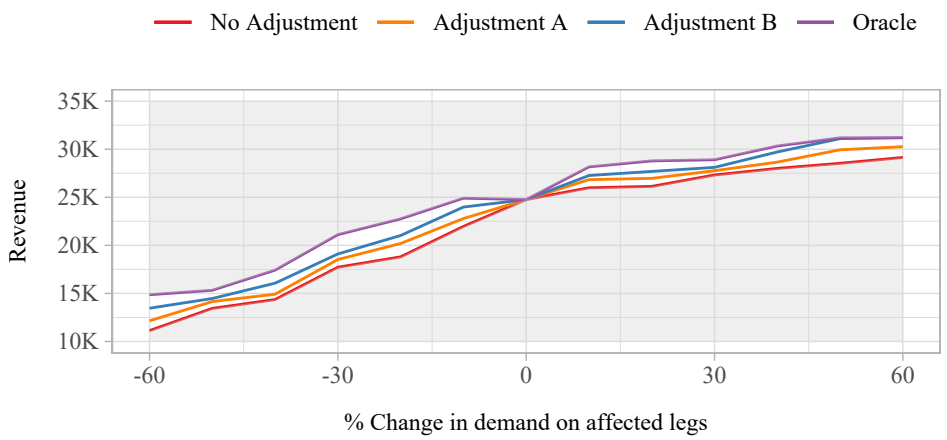


Figure B.3.15: Revenue generated under different forecast adjustments resulting from the outlier detection for outlier demand in itinerary AE

in contrast to leg level adjustments alone. Though making adjustments to only the single-leg itineraries may be risk averse in the rare cases where an outlier affects only a small subset of the legs within a cluster, it may be detrimental to revenue when outliers affect multiple legs.

B.4 Empirical study of Deutsche Bahn booking data

Appendix B.4 contains additional analysis of the empirical booking data from Deutsche Bahn, as described in Section 3.4.

B.4.1 Model selection for functional regression

Due to the functional nature of the data, in order to determine which of the factors result in a better fitting model, we use the **Cross-Validated Sum Of Integrated Squared Errors** (CV-SSE).

$$CV\text{-}SSE = \sum_{n=1}^{N} \int (y_{nl}(t) - \hat{y_{nl}}(t))dt, \tag{B.4.1}$$

where $\hat{y_{nl}}(t)$ is the prediction for the n^{th} booking pattern on the leg l, under the model fitted to all but the n^{th} booking pattern. The model which produces the lowest CV-SSE is chosen as the best fitting. Note that unlike other model selection criterion (e.g. AIC), CV-SSE does not take into account the number of parameters. Given that we are not interested in out of sample prediction, only in obtaining the best fitting model for our data, over-fitting is not of great concern. The values of the CV-SSE for each of the 12 models considered are shown in Table B.4.1.

Across all legs, we find that day, month, and shortened booking horizons are all factors that must be taken into account. The inclusion of the days of the week as factors significantly reduces the CV-SSE. In comparison, the inclusion of the booking horizon variable has a smaller, though still positive, effect. We compare two different approaches to accounting for the shortened booking horizon: (i) an indicator function (I) equal to 1 if the booking horizon is shorter, and (ii) a continuous variable (C) between 0 and 1 which gives the length of the shortened horizon as a proportion of the regular length horizon. Based on the CV-SSE scores, shortened booking horizons are best represented by the indicator function i.e. it is important to know that it is shorter but not by how much. The smaller effect of the horizon length variable

Model	Intercept	Day	Month	Short Horizon	Short Horizon (C)	CV-SSE			
						Leg AB	Leg BC	Leg CD	Leg DE
Model 1	✓			**(I)**		79974160	75034839	79529280	73824611
Model 2	✓			✓		58617546	52622148	52424683	50009080
Model 3	✓				✓	58620898	52863263	52506946	50014984
Model 4	✓	✓				27227350	35376732	32789181	30037659
Model 5	✓	✓		✓		26551341	33724380	32282900	29989390
Model 6	✓	✓			✓	26704943	34154782	32439972	30019196
Model 7	✓		✓			58620649	57895619	52638923	50015645
Model 8	✓		✓	✓		58608640	57865403	52615801	49996331
Model 9	✓		✓		✓	58878374	57885484	52654330	50033157
Model 10	✓	✓	✓			24574978	25700166	21691111	21880038
Model 11	✓	✓	✓	✓		**24519539**	**25691637**	**21689686**	**21878259**
Model 12	✓	✓	✓		✓	24546715	25697938	21724073	21896889

Table B.4.1: Model comparison for functional regression

may be related to the inclusion of the month variable, which is unsurprising given the overlap in the definition of these variables. The values of the CV-SSE is similar for the models 2 and 7, where we only consider one of month or horizon length as a factor.

B.4.2 Residual booking patterns

Figure B.4.1 shows the residual booking patterns resulting from the functional regression applied in equation (3.4.1) of Section 3.4.2. Compare with Figure 3.4.5 of Section 3.4.2 – the obvious outliers are preserved.

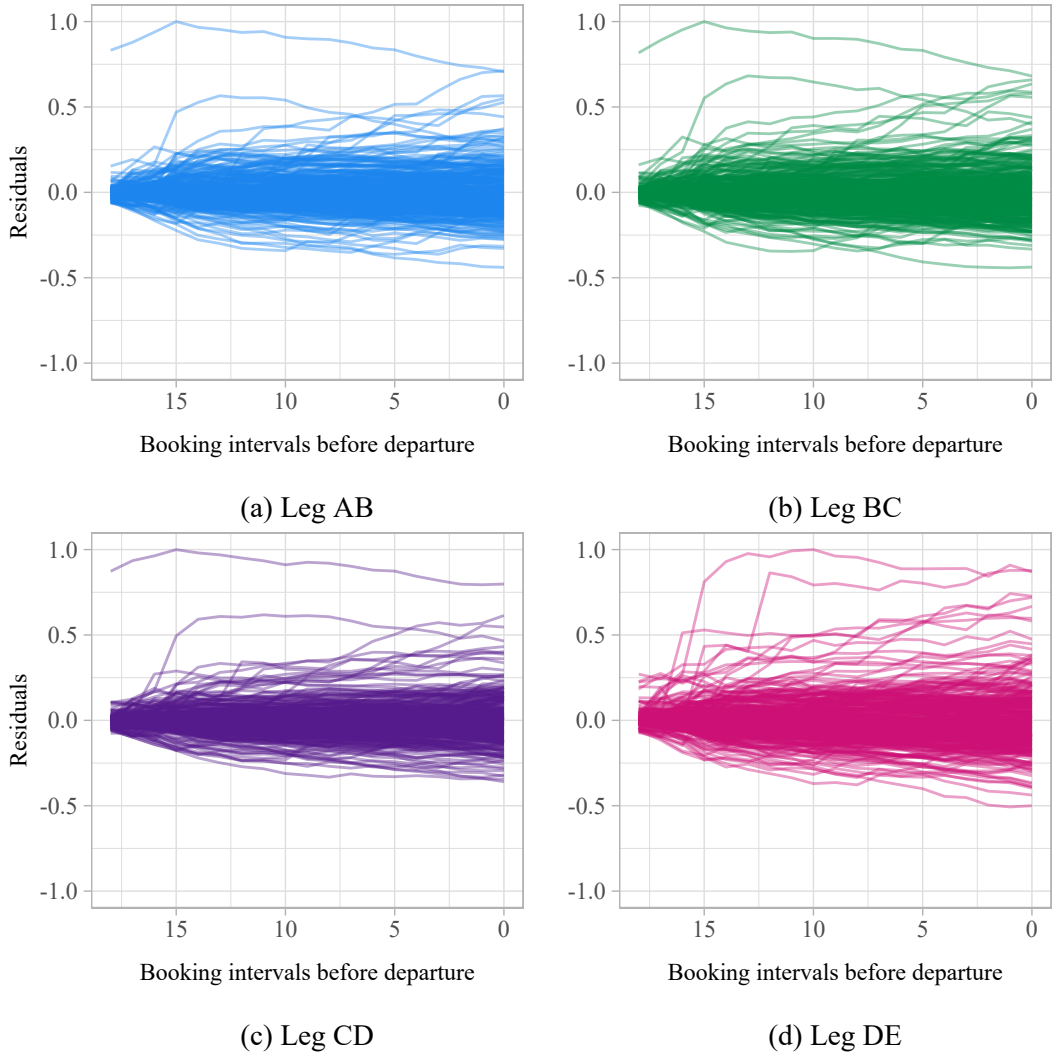


Figure B.4.1: Residual booking patterns

B.4.3 Functional depths

Figure B.4.2 shows the functional depths for the empirical residual booking patterns, before the functional depths are transformed into the z_{nl}, as shown in Figure 3.4.6 of Section 3.4.2.

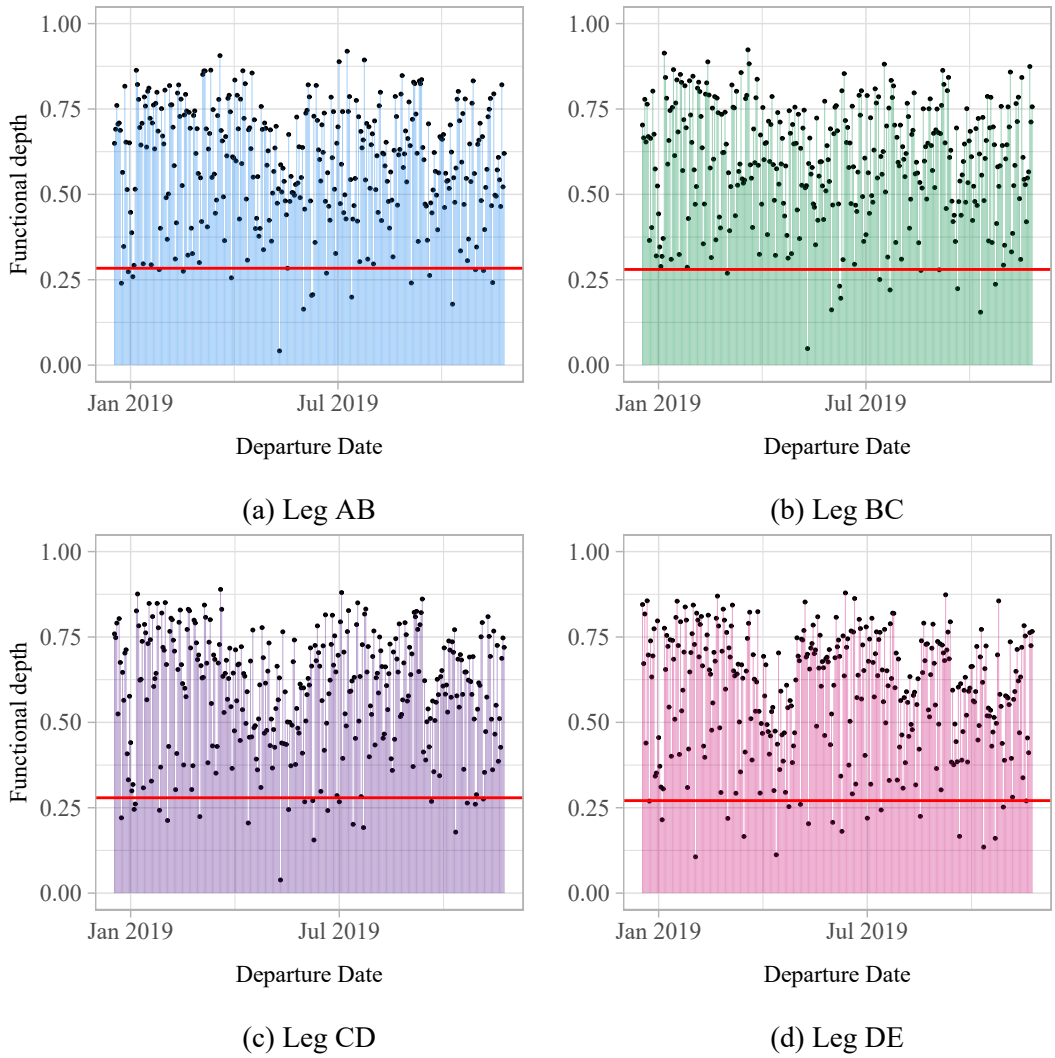
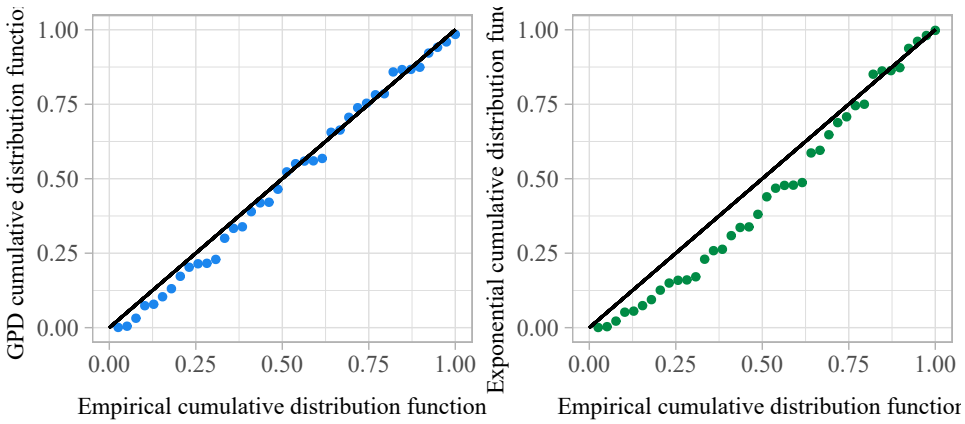


Figure B.4.2: Functional depths

B.4.4 Probability plots for GPD and Exponential distributions

Given that, if both $\mu = 0$ and $\xi = 0$, the GPD reduces to an exponential distribution, it is appropriate to compare the fit of the GPD with an exponential distribution to check if the inclusion of additional parameters is beneficial. Figure B.4.3 shows the P-P plots, i.e. the fitted theoretical CDF against the empirical CDF for the GPD (Figure

B.4.3a) and the Exponential distribution (Figure B.4.3b). The GPD provides a closer fit to the empirical data and the additional parameters better account for the shape of the distribution. The GPD does not provide a perfect fit, with the probabilities in the



(a) Generalised Pareto distribution (b) Exponential distribution

Figure B.4.3: P-P plots

bottom left of Figure B.4.3a on consistently being underestimated. However, given that we assume points with very low probability are more likely to be false positives, under-estimating may actually be beneficial. Further, only the highly-ranked outliers i.e. those with high probability, are likely to be considered by an analyst due to time-constraints. The GPD provides a very good fit for those data points. If there is a sufficiently large number of threshold exceedances, an empirical distribution could alternatively be used to compute the probabilities.

B.4.5 Distribution of outliers across multiple legs

The proportion of outliers found in each number of legs is shown in Figure B.4.4, with over half of the outliers detected in multiple legs. Compared with Figure B.3.5, this shows a similar proportion of outliers as found in the simulation study.

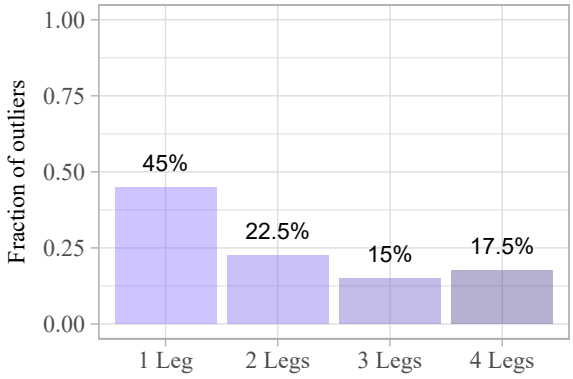
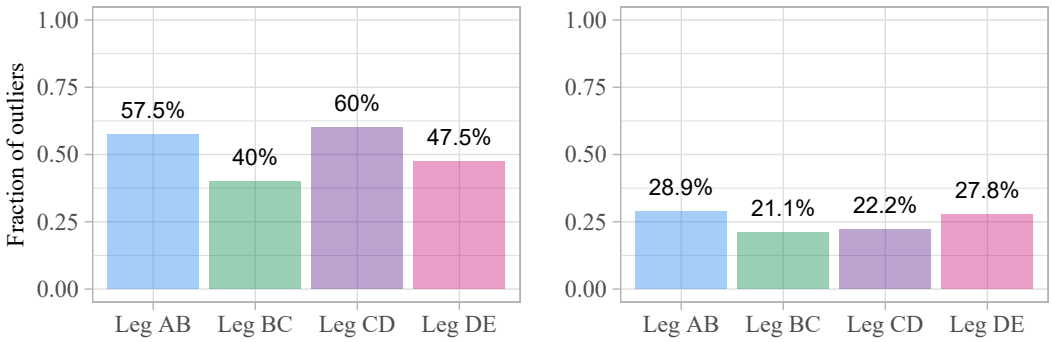


Figure B.4.4: Fraction of all outliers detected in 1, 2, 3, or 4 legs

Figure B.4.5a shows the proportion of total outlying booking patterns in terms of which legs they were detected as outliers in. Figure B.4.5b shows the proportion in each leg of outlying booking patterns detected in one leg only. The proportions are fairly evenly split between the different legs. This reassures us that the correct clustering was chosen - if leg DE did in fact belong to a separate second cluster, we would expect a higher proportion of single leg outliers to have been found in leg DE – compare with Figure B.3.6.

B.4.6 Simulation verification

In order to validate the parameter choices used to simulate booking patterns, we compare the resulting simulated booking patterns with the empirical booking patterns.

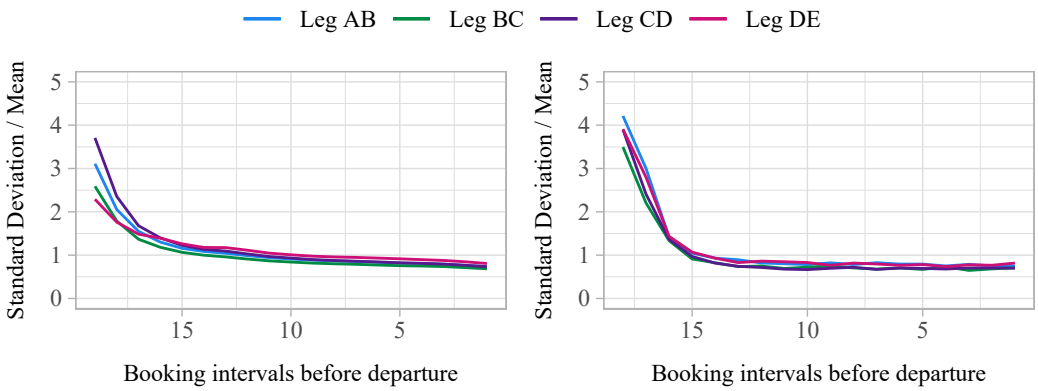


(a) Fraction of outliers detected in each leg, for outliers detected in at least one leg

(b) Fraction of outliers detected in each leg, for outliers detected in only one leg

Figure B.4.5: Fraction of outliers detected in each leg

We consider the standard deviation and mean of the bookings across the booking horizon of each in Figure B.4.6. Both the empirical and simulated booking patterns show a similar shape and magnitude of relationship between the mean and standard deviation across the booking horizon.



(a) Empirical booking patterns

(b) Simulated booking patterns

Figure B.4.6: Comparison of standard deviation divided by mean of booking patterns

We also compare the correlations between the different legs for both the empirical and simulated data. Table B.4.2 shows the functional dynamical correlation between

the empirical booking patterns, and empirical residual booking patterns, for each leg. Table B.4.3 shows the corresponding correlations between the simulated booking patterns. The values are similar and the rate of decay between legs as they get further apart follows a similar pattern.

	Leg AB	**Leg BC**	**Leg CD**	**Leg DE**
Leg AB	-	0.95	0.83	0.70
Leg BC	-	-	0.83	0.66
Leg CD	-	-	-	0.78
Leg DE	-	-	-	-

(a) Booking patterns

	Leg AB	**Leg BC**	**Leg CD**	**Leg DE**
Leg AB	-	0.92	0.75	0.58
Leg BC	-	-	0.88	0.74
Leg CD	-	-	-	0.84
Leg DE	-	-	-	-

(b) Residual booking patterns

Table B.4.2: Functional dynamical correlation of empirical booking patterns

	Leg AB	**Leg BC**	**Leg CD**	**Leg DE**
Leg AB	-	0.81	0.72	0.60
Leg BC	-	-	0.86	0.68
Leg CD	-	-	-	0.78
Leg DE	-	-	-	-

Table B.4.3: Functional dynamical correlation of simulated booking patterns

Appendix C

Appendix: Analysing and visualising bike-sharing demand with outliers

C.1 Forecasting baseline demand

C.1.1 Temporal partitioning

In Section 4.3, for the purposes of temporal partitioning of data, we define summer to be the months April through October. Winter is therefore November through March. The partitioning is chosen to give constant variance within a partition whilst also ensuring there is a sufficient number of observations within each partition to make outlier detection feasible. Figure C.1.1 shows the rolling daily variance for terminal 31203, with the summer months highlighted.

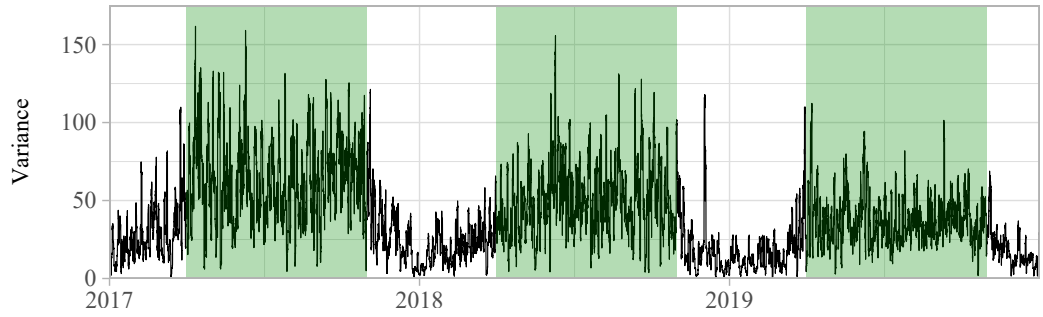


Figure C.1.1: Variance of usage patterns with summer months highlighted in green

The variance of winter is not constant, being slightly higher in the months that border the summer season. Further partitioning could be carried out e.g. partition by month. However, this results in much less data within each partition, which then makes outlier detection more difficult. When applying binary segmentation changepoint detection (Scott and Knott, 1974) to identify the partitions with different levels of variance, the algorithm returns 8 changepoints: 24 March 2017, 4 Nov 2017, 6 Dec 2017, 31 Mar 2018, 24 May 2018, 4 Nov 2018, 20 Mar 2019, and 4 Nov 2019. These are highlighted in Figure C.1.2. These are relatively close to our pre-defined summer and winter partitions (indicated by red vertical lines in Figure C.1.2.

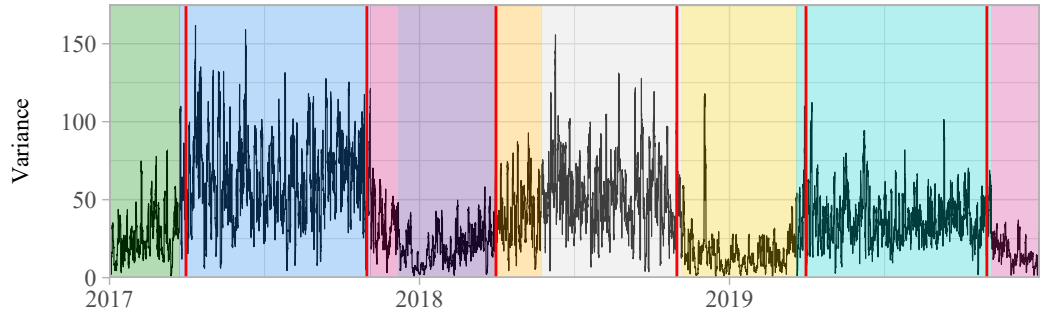


Figure C.1.2: Changepoints in variance of rental patterns

If there are already pre-defined seasons in use for planning purposes, these may be more appropriate.

C.1.2 Functional regression model comparison

In this section, we perform model comparison for the functional regression model used to account for different daily trends as detailed in Section 4.3, equation 4.3.1.

We use the **Cross-Validated Mean Integrated Squared Error** (CV-MSE) to determine the best-fitting model. The CV-MSE is given by:

$$CV\text{-}MSE = \frac{1}{N}\sum_{n=1}^{N}\int (x_{n,s}(t) - \hat{x}_{n,s}(t))dt, \tag{C.1.1}$$

where $\hat{x}_{n,s}(t)$ is the prediction for the n^{th} daily rental pattern at the terminal s, under the model fitted to all but the n^{th} rental pattern. The model which produces the lowest CV-MSE is chosen as the best fitting. Unlike other model selection criterion such AIC, CV-MSE does not take into account the number of parameters. The CV-MSE for each of the 8 models considered is shown in Table C.1.1, for the terminals in the cluster discussed in Section 4.5.

In most cases, the model which achieves the minimum mean squared error is model 8, which includes all three factors (day, week, and year). However, model 5 (day and month) also produces very similar results.

C.1.3 Distribution of residuals

Figure C.1.3 shows the distribution of the residuals for each hour of the day for terminal 31005 – see also Figure 4.3.1. The core of the distribution is symmetric

Model	Factors			Terminal Number								
	Day	Month	Year	31303	31308	31309	31315	31316	31317	31319	32014	32040
1				135.90	72.98	16.04	25.80	15.29	27.96	34.27	52.42	22.75
2	✓			120.38	60.17	16.04	25.64	14.29	27.71	33.43	50.80	22.78
3		✓		98.01	59.65	12.67	18.82	11.93	19.57	25.78	37.97	17.09
4			✓	133.96	71.89	15.49	25.70	14.86	28.03	33.97	48.01	22.63
5	✓	✓		82.40	46.50	12.61	18.65	10.83	**19.20**	24.80	36.07	17.01
6		✓	✓	96.59	58.52	12.07	18.79	11.52	19.61	25.44	33.34	16.95
7	✓		✓	119.04	59.02	15.47	25.63	13.87	27.77	33.13	46.28	22.66
8	✓	✓	✓	**80.85**	**45.29**	**12.00**	**18.62**	**10.39**	19.23	**24.46**	**31.32**	**16.87**

Table C.1.1: Cross validated mean square error for functional regression model comparison applied to unpartitioned data

around zero, but the tails of the distribution are positively skewed.

C.1.4 Accounting for skewness

Figure C.1.5a shows the distribution of the normalised total daily usage for terminal 31235, which exhibits positive skew. Not all terminals exhibit such positively skewed distributions – see Figure C.1.4.

The distributions of total daily usage have a skewness lying between -0.4 and 15.8, with the median skewness across all terminals being 0.71. Larger positive skew is more common in terminals where mean usage is very low, and since demand is bounded below by zero, only increases in demand are observed. This results in more *positive* outliers than *negative* (see Section sec:discussion). Given that most terminals exhibit

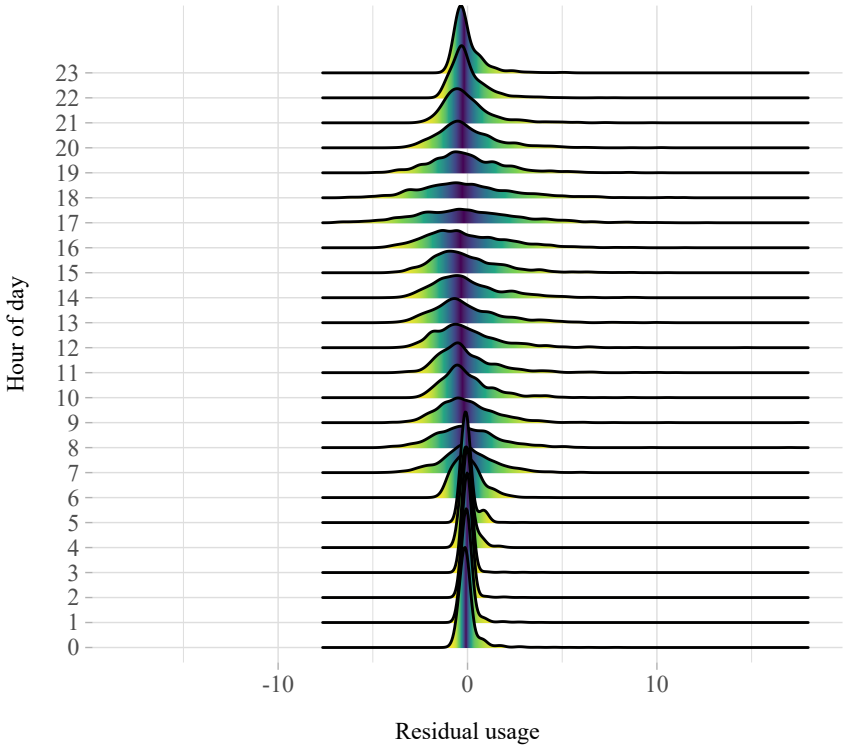


Figure C.1.3: Distribution of residual usage for each hour of the day for terminal 31005

slight positive skew, it may be desirable to transform the data before perfroming outlier detection. To account for the skew, the rental patterns can first be transformed e.g. with a logarithmic transform. However, this is not applicable to all terminals (as some are already negatively skewed) and can result in a negatively skewed distribution – Figure C.1.5b.

When applying the outlier detection procedure to the untransformed data, the fraction of positive outliers is consistently higher than the fraction of negative outliers. On average, 78% of outliers are positive. That is, outliers are more likely to be caused by increased demand than decreased. This is easily explained by the fact that demand is bounded below by zero, and in many cases the mean usage pattern

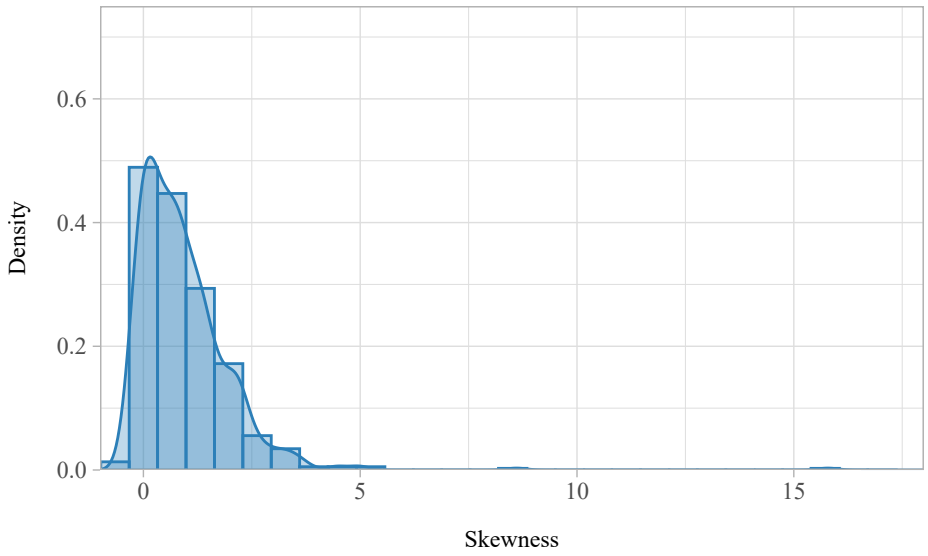


Figure C.1.4: Distribution of skewness of distributions of total daily usage across all terminals

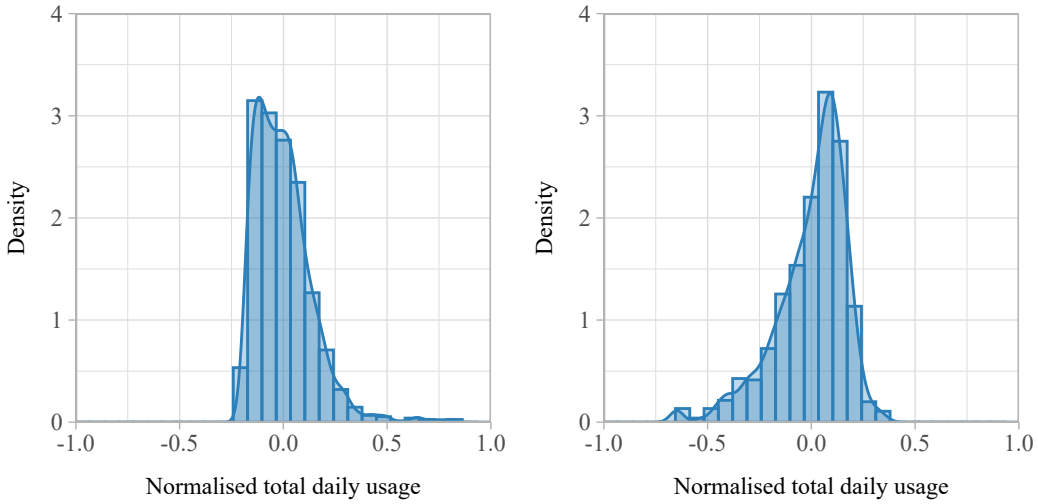
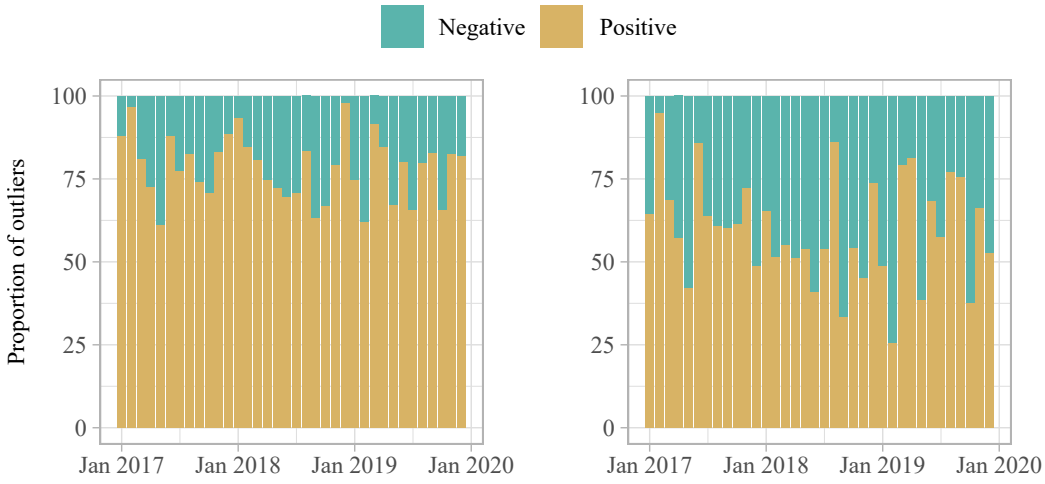


Figure C.1.5: Distribution of total daily usage for terminal 31235

is close to zero, such that negative demand is unobservable. Applying a logarithmic transformation before carrying out the outlier detection results in around 60% of outliers being positive.



(a) Untransformed data (b) Logarithmic transform

Figure C.1.6: Fraction of outliers that are positive and negative, before and after applying a logarithmic transformation

C.1.5 Inter-daily autocorrelation

Figure C.1.7 shows the inter-daily autocorrelations between the residual patterns for different days, at each hour. The early hours of the morning - especially at 04:00 and 06:00 - exhibit some autocorrelation of lag 7 i.e. weekly. A functional ARIMA model could be fitted to remove the autocorrelation. However, as it only affects a so few hours of the day, we do not investigate this further here.

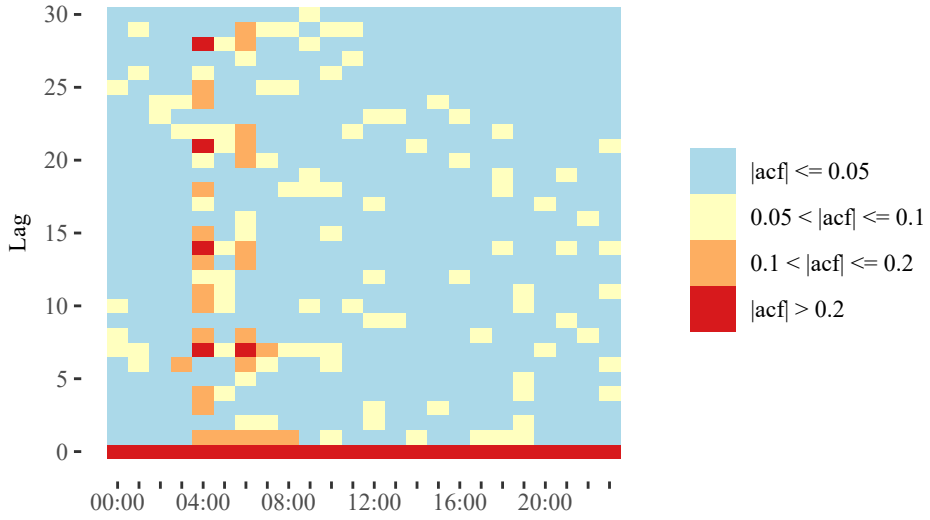


Figure C.1.7: Inter-daily autocorrelations of residuals for terminal 31005

C.2 Using spatial patterns to cluster terminals

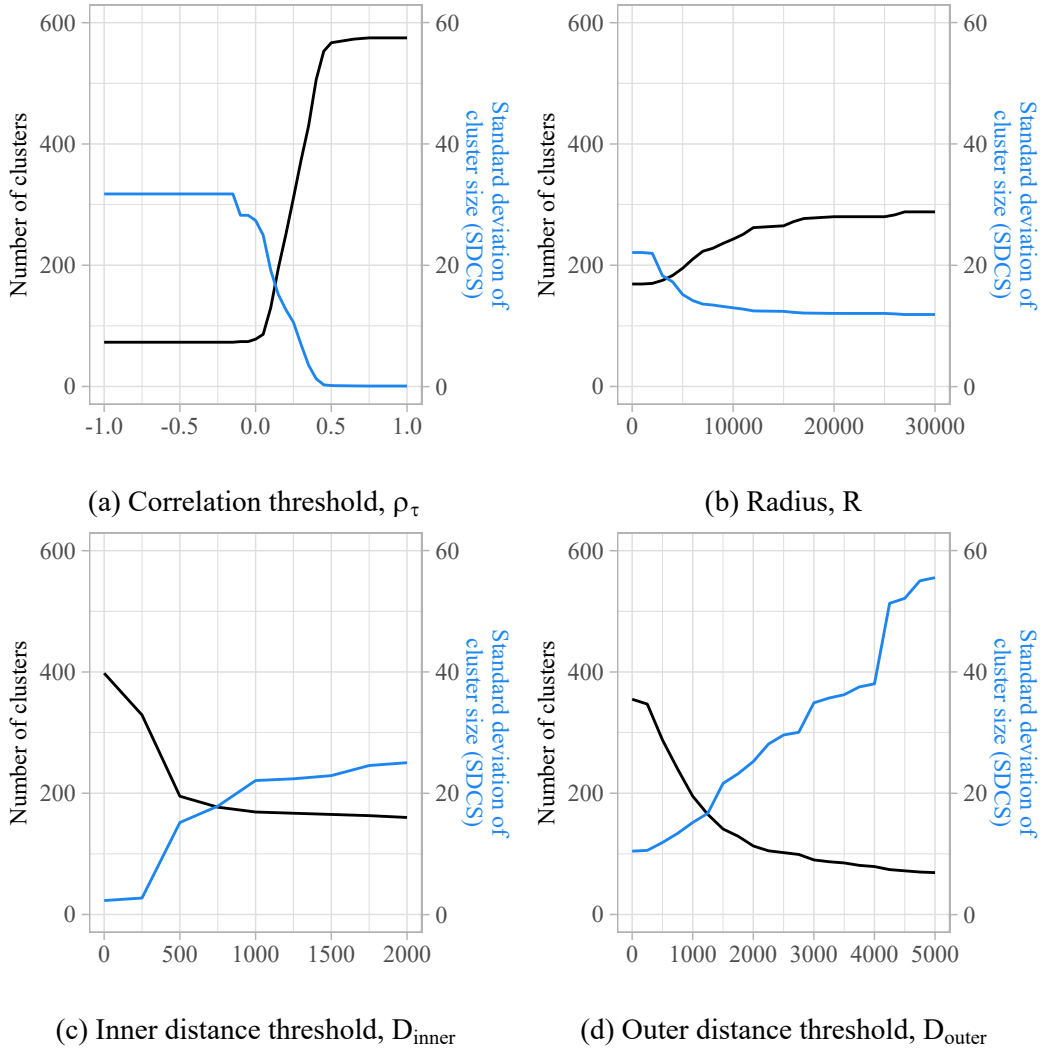
C.2.1 Effect of parameter choices on clustering

Our clustering method is tuned using four parameters: the correlation threshold ρ, as well as distance metrics introduced in Section 4.4. We now evaluate the sensitivity of changing these parameters on (i) the number of clusters obtained, and (ii) the standard deviation in cluster sizes (SDCS) (Lin et al., 2019). The SDCS is given by:

$$SDCS = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}\left(S_k - \frac{S}{K}\right)^2}, \tag{C.2.1}$$

where S is the number of terminals, K is the number of clusters, S_k is the number of terminals in cluster k. The SDCS quantifies a measure of the balance of the different cluster sizes. We do not seek to minimise nor maximise the SDCS – since choosing extreme parameter values trivially creates clusters of size 1 or one giant cluster.

Figure C.2.1 shows the change in number of clusters and SDCS as we vary pa-



(a) Correlation threshold, ρ_τ

(b) Radius, R

(c) Inner distance threshold, D_{inner}

(d) Outer distance threshold, D_{outer}

Figure C.2.1: Cluster sensitivity to parameter changes when other parameters remain fixed at ρ_τ=0.15, R = 5000m, D_{inner} = 500m, and D_{outer} = 1000m.

rameter values. There is an inverse relationship between the number of clusters and the SDCS across all four variables. While an increase in either correlation threshold or radius results in a decrease of SDCS, increasing either of the distance thresholds increases the SDCS. In order to achieve a balance between number of clusters and SDCS, we choose parameter values close to the intersection of the two lines. This results in a correlation threshold of between 0 and 0.4; a radius between 5,000m

and 10,000m; an inner distance threshold between 500m and 1,000m, and an outer distance threshold of approximately 1,000m.

C.2.2 Normalised Mutual Information

For a graph containing M terminals, the mutual information between two clusterings $\mathcal{A}$ and $\mathcal{B}$ of the M nodes in the graph is defined as:

$$I(\mathcal{A},\mathcal{B}) = \sum_{a=1}^{|\mathcal{A}|}\sum_{b=1}^{|\mathcal{B}|}\frac{|\mathcal{A}\cap\mathcal{B}|}{M}\log\left(|\mathcal{A}\cap\mathcal{B}|\frac{M}{M_a M_b}\right), \tag{C.2.2}$$

where M_a is the number of nodes in the a^{th} cluster of clustering $\mathcal{A}$, and similarly for M_b. The **normalised mutual information (NMI)** between two clusterings is defined as (Amelio and Pizzuti, 2015):

$$NMI(\mathcal{A},\mathcal{B}) = \frac{2I(\mathcal{A},\mathcal{B})}{H(\mathcal{A})+H(\mathcal{B})}, \tag{C.2.3}$$

where $H(\mathcal{A})$ is the entropy (a measure of uncertainty) defined as:

$$H(\mathcal{A}) = -\sum_{a=1}^{|\mathcal{A}|}\frac{M_a}{M}\log\left(\frac{M_a}{M}\right). \tag{C.2.4}$$

$NMI(\mathcal{A},\mathcal{B}) = 1$ if $\mathcal{A}$ and $\mathcal{B}$ are identical, and 0 if they are completely different.

C.3 Additional Discussion

C.3.1 Effects of data temporal patterns on outlier detection

In section 4.3, we outlined two steps (functional regression and temporal partitioning) that could be undertaken to account for different patterns in the data. Here, we consider how the inclusion of these steps affects the outcome of the outlier detection. For a homogeneous data set, we would expect approximately equal numbers of outliers detected on each day of the week, and month of the year. Figure C.3.1 shows the difference between the mean fraction of outliers per day (or month) and fraction of outliers which are observed on each day of the week (or month).

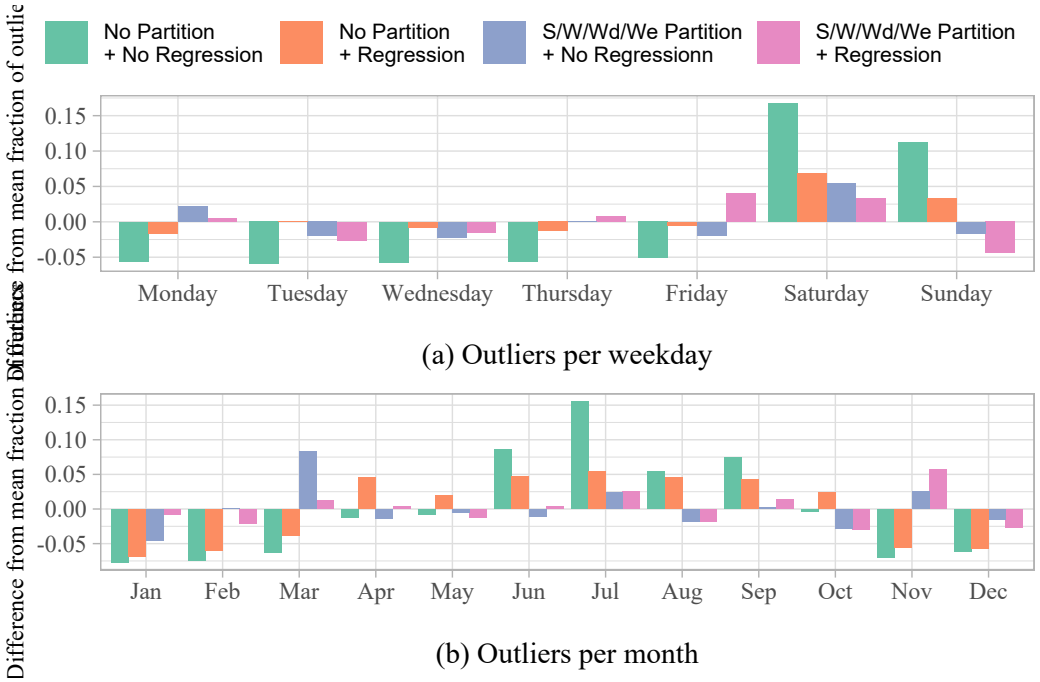


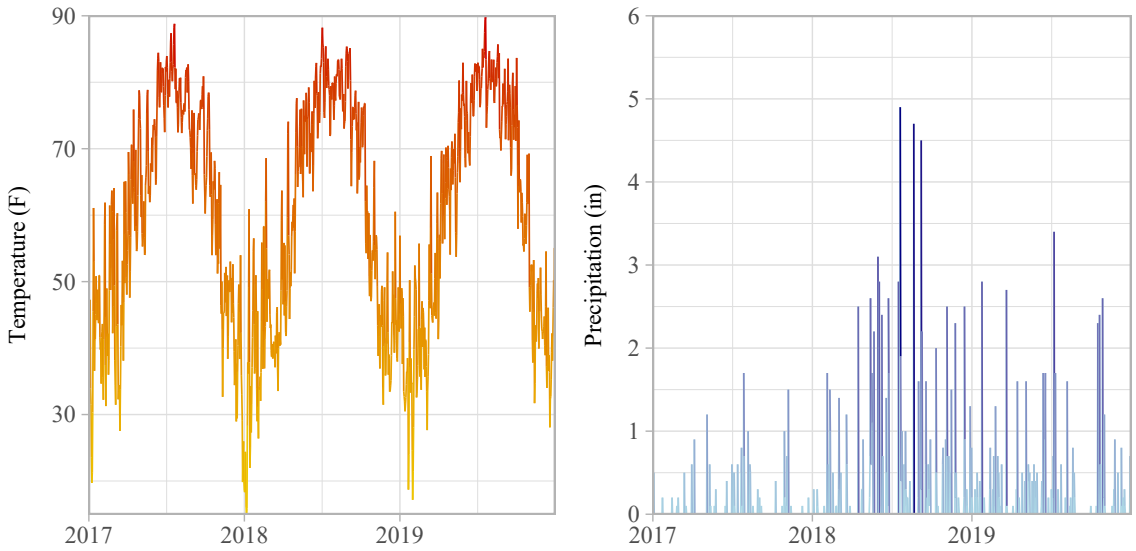
Figure C.3.1: Fraction of outliers occurring on each day of the week and month of the year, with and without applying functional regression model

The results are shown for the case where the (i) there is no accounting for tem-

poral patterns; (ii) the regression model is applied with no partitioning; (iii) only partitioning is applied with no regression, and (iv) both regression and partitioning is applied. When we do not account for temporal patterns in the data, we detect far more outliers on weekends and in the summer months. Including the regression step (without partitioning) improves this imbalance somewhat. When the data has been partitioned, regression makes little difference to the proportion of outliers detected on each day or month. Although we partition the data to account for different variance, this implicitly takes care of differences in mean between the same groups. As there is little difference in mean trend between days or months in the same groups. partitioning with or without regression gives similar results.

C.3.2 Weather as an explanatory factor for demand outliers

Figure C.3.2 shows the weather data used for analysis in Section 4.6.2.



(a) Average daily temperature (F)

(b) Total daily precipitation (in)

Figure C.3.2: Weather data obtained from Visual Crossing for 2017 - 2019

Bibliography

C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the Surprising Behavior of Distance Metrics in High Dimensional Space. *In: Van den Bussche J., Vianu V. (eds) Database Theory. ICDT 2001. ICDT 2001. Lecture Notes in Computer Science, vol 1973. Springer, Berlin, Heidelberg*, pages 420–434, 2001.

A. Amelio and C. Pizzuti. Is normalized mutual information a fair measure for comparing community detection methods? *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2015*, pages 1584–1585, 2015.

K. Augustin, R. Gerike, M. J. Martinez Sanchez, and C. Ayala. Analysis of intercity bus markets on long distances in an established and a young market: The example of the U.S. and Germany. *Research in Transportation Economics*, 48:245–254, 2014.

N. Banerjee, A. Morton, and K. Akartunalı. Passenger demand forecasting in scheduled transportation. *European Journal of Operational Research*, 286:797–810, 2020.

D. Barrow and N. Kourentzes. The impact of special days in call arrivals forecast-

ing: A neural network approach to modelling special days. *European Journal of Operational Research*, 264(3):967–977, 2018.

P. Bartke, N. Kliewer, and C. Cleophas. Benchmarking filter-based demand estimates for airline revenue management. *EURO Journal on Transportation and Logistics*, 7(1):57–88, 2018.

Rahul Basole, Elliot Bendoly, Aravind Chandrasekaran, and Kevin Wayne Linderman. Visualization in Operations Management Research. *SSRN Electronic Journal*, pages 1–19, 2020.

P. P. Belobaba. OR Practice: Application of a Probabilistic Decision Model to Airline Seat Inventory Control. *Operations Research*, 37(2), 1989.

P. P. Belobaba. Optimal vs. heuristic methods for nested seat allocation. In *Proceedings of AGIFORS Reservations and Yield Management Study Group (1992)*, pages 28–53, 1992.

G.E.P. Box and G.M. Jenkins. *Time Aeries Analysis: Forecasting and Control.* San Francisco: Holden-Days., 1970.

J.E. Boylan, P. Goodwin, M. Mohammadipour, and A.A. Syntetos. Reproducibility in forecasting research. *International Journal of Forecasting*, 31(1):79–90, 2015.

S. L. Brumelle and J. I. McGill. Airline seat allocation with multiple nested fare classes. *Operations Research*, 41(1):127–137, 1993.

Ralph Buehler and John Pucher. Demand for Public Transport in Germany and

the USA: An Analysis of Rider Characteristics. *Transport Reviews*, 32(5):541–567, 2012.

Capital Bikeshare. System data, 2021. `https://www.capitalbikeshare.com/system-data` [Accessed: 2021-04-01].

Mark Carpenter and Satya N. Mishra. Fitting the generalized beta distribution to data. *American Journal of Mathematical and Management Sciences*, 21(1-2):165–182, 2001.

V. Chandola, A. Banerjee, and V. Kumar. Survey of Anomaly Detection. *ACM Computing Survey*, 41(3):1–72, 2009.

C. Chatfield. *The Analysis of Time Series: An Introduction.* Chapman and Hall, 1975.

Claudio Ciancio, Giuseppina Ambrogio, and Demetrio Laganá. A stochastic maximal covering formulation for a bike sharing system. In Antonio Sforza and Claudio Sterle, editors, *Optimization and Decision Science: Methodologies and Applications*, pages 257–265. Springer International Publishing, 2017.

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

C. Cleophas, M. Frank, and N. Kliewer. Simulation-based key performance indicators for evaluating the quality of airline demand forecasting. *Journal of Revenue and Pricing Management*, 4(8):330–342, 2009.

C. Cleophas, D. Kadatz, and S. Vock. A Literature Survey of Recent Theoretical Advances. *Journal of Revenue and Pricing Management*, 16(5):483–498, 2017.

A. Cuevas, M. Febrero, and R. Fraiman. An anova test for functional data. *Computational Statistics & Data Analysis*, 47:111–122, 2004.

C. S. M. Currie and I. T. Rowley. Consumer behaviour and sales forecast accuracy: what's going on and how should revenue managers respond? *Journal of Revenue and Pricing Management*, 9:374–376, 2010.

S. De Baets and N. Harvey. Using judgment to select and adjust forecasts from statistical models. *European Journal of Operational Research*, 284(3):882–895, 2020.

A. B. Deb and L. Dey. Outlier Detection and Removal Algorithm in K-Means and Hierarchical Clustering. *World Journal of Computer Application and Technology*, 5(2):24–29, 2017.

W.J. Dixon and K. K. Yuen. Trimming and winsorization: A review. *Statistische Hefte*, 15(2-3):157–170, 1974.

G. R. Doreswamy, A. S. Kothari, and S. Tirumalachetty. Simulating the flavors of revenue management for airlines. *Journal of Revenue and Pricing Management*, 6 (14):421–432, 2015.

J. A. Dubin and H. G. Müller. Dynamical correlation for multivariate longitudinal data. *Journal of the American Statistical Association*, 100:872–881, 2005.

L. Fawcett and D. Walshaw. Improved estimation for temporally clustered extremes. *Environmetrics*, 18(2):173–188, 2007.

A. Fawzy, H. M.O. Mokhtar, and O. Hegazy. Outliers detection and classification in wireless sensor networks. *Egyptian Informatics Journal*, 14(2):157–164, 2013.

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

T. Fiig, K. Isler, C. Hopperstad, and P. Belobaba. Optimization of mixed fare structures: Theory and applications. *Journal of Revenue & Pricing Management*, 9(12): 152–170, 2010.

M. Frank, M. Friedemann, and A. Schröder. Principles for simulations in revenue management. *Journal of Revenue and Pricing Management*, 1(7):215–236, 2008.

Kun Gao, Ying Yang, Aoyong Li, and Xiaobo Qu. Spatial heterogeneity in distance decay of using bike sharing: An empirical large-scale analysis in Shanghai. *Transportation Research Part D: Transport and Environment*, 94(May 2021), 2021.

J. Gönsch. A survey on risk-averse and robust revenue management. *European Journal of Operational Research*, 263(2):337–348, 2017.

S. D. Grimshaw. Computing maximum likelihood estimates for the generalized pareto distribution. *Technometrics*, 35(2):185–191, 1993.

Jianhua Guo, Wei Huang, and Billy M. Williams. Real time traffic flow outlier detection using short-term traffic conditional variance prediction. *Transportation Research Part C: Emerging Technologies*, 50:160–172, 2015.

F. Habibzadeh and P. Habibzadeh. The likelihood ratio and its graphical representation. *Biochemia medica*, 29(2), 2019.

G. J. Hahn and R. Chandra. Tolerance Intervals for Poisson and Binomial Variables. *Journal of Quality Technology*, 13(2):100–110, 1981.

Timothy L. Hamilton and Casey J. Wichman. Bicycle infrastructure and traffic congestion: Evidence from DC's Capital Bikeshare. *Journal of Environmental Economics and Management*, 87:72–93, 2018.

D. Hawkins. *Identification of Outliers*. Chapman and Hall, 1980.

G. He, H. G. Müller, and J. L. Wang. Functional canonical analysis for square integrable stochastic processes. *Journal of Multivariate Analysis*, 85(1):54–77, 2003.

M. Hubert, G. Claeskens, B. De Ketelaere, and K. Vakili. A new depth-based approach for detecting outlying curves. In A. Colubi, K. Fokianos, G Gonzalez-Rodriguez, and E.J. Kontoghiorghes, editors, *Proceedings of COMPSTAT 2012*, pages 329–340, 2012.

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

Rob J. Hyndman. Computing and Graphing Highest Density Regions. *The American Statistician*, 50(2):120–126, 1996.

Rob J. Hyndman, Earo Wang, and Nikolay Laptev. Large-Scale Unusual Time Series Detection. *Proceedings - 15th IEEE International Conference on Data Mining Workshop, ICDMW 2015*, pages 1616–1619, 2016. doi: 10.1109/ICDMW.2015.104.

B. Iglewicz and D. Hoaglin. The ASQC Basic References in Quality Control: Statistical Techniques. *In: Mykytka, E.F., (eds), How to Detect and Handle Outliers*, 16, 1993.

Abhay Jha, Shubhankar Ray, Brian Seaman, and Inderjit S. Dhillon. Clustering to forecast sparse time-series data. *Proceedings - International Conference on Data Engineering*, 2015-May:1388–1399, 2015.

M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30:81–93, 1938.

A. Kimms and M. Müller-Bungart. Simulation of stochastic demand data streams for network revenue management problems. *OR Spectrum*, 1(29):5–20, 2007.

R. Klein, S. Koch, C. Steinhardt, and A. K. Strauss. A review of revenue management: Recent generalizations and advances in industry applications. *European Journal of Operational Research*, 284(2):397–412, 2020.

M. Lawrence, M. O'Connor, and B. Edmundson. A field study of sales forecasting accuracy and processes. *European Journal of Operational Research*, 122(1):151–160, 2000.

M. Lawrence, P. Goodwin, M. O'Connor, and D. Onkal. Judgmental forecasting: A review of progress over the last 25 years. *International Journal of forecasting*, 22 (3):493–518, 2006.

M.R. Leadbetter. On a basis for 'Peaks over Threshold' modeling. *Statistics and Probability Letters*, 12(4):357–362, 1991.

Loet Leydesdorff. Similarity Measures, Author Cocitation Analysis, and Information Theory. *Journal of the American Society for Information Science*, 56(7):769–772, 2005.

T. X. Liang and C. X. Cao. Outliers detect methods for time series data. *Journal of Discrete Mathematical Sciences and Cryptography*, 21(4):927–936, 2018.

Pengfei Lin, Jiancheng Weng, Quan Liang, Dimitrios Alivanistos, and Siyong Ma. Impact of Weather Conditions and Built Environment on Public Bikesharing Trips in Beijing. *Networks and Spatial Economics*, 20(1):1–17, 2020.

Weibo Lin, Zhu He, and Mingyu Xiao. Balanced clustering: A uniform model and fast algorithm. *IJCAI International Joint Conference on Artificial Intelligence*, 2019-August:2987–2993, 2019.

Ting Ma, Chao Liu, and Sevgi Erdoğan. Bicycle sharing and public transit: Does capital bikeshare affect metrorail ridership in Washington, D.C.? *Transportation Research Record*, 2534(2534):1–9, 2015.

J. MacQueen. Some methods for classification and analysis of multivariate observa-

tions. In *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob.*, volume 1, pages 281–297. University of California Press, 1967.

B. J. McNeil and J. A. Hanley. Statistical Approaches to the Analysis of Receiver Operating Characteristic (ROC) Curves. *Medical Decision Making*, 4(2):137–150, 1984.

D. R. Morales and J. Wang. Forecasting cancellation rates for services booking revenue management using data mining. *European Journal of Operational Research*, 202 (2010):554–562, 2010.

S. Mukhopadhyay, S. Samaddar, and G. Colville. Improving revenue management decision making for airlines by evaluating analyst-adjusted passenger demand forecasts. *Decision Sciences*, 38(2):309–327, 2007.

Bruno Albert Neumann-Saavedra, Dirk Christian Mattfeld, and Mike Hewitt. Assessing the operational impact of tactical planning models for bike-sharing redistribution. *Transportation Research Part A: Policy and Practice*, 150(June):216–235, 2021.

M. O'Connor, W. Remus, and K. Griggs. Judgemental forecasting in times of change. *International Journal of Forecasting*, 9(2):163–172, 1993.

K. Pearson. VII. Note on regression and inheritance in the case of two parents. *Proc. R. Soc. Lond.*, 58, 1895.

L. N. Pereira. An introduction to helpful forecasting methods for hotel revenue management. *International Journal of Hospitality Management*, 58:13–23, 2016.

H. N. Perera, J. Hurley, B. Fahimnia, and M. Reisi. The human factor in supply chain forecasting: A systematic review. *European Journal of Operational Research*, 274 (2):574–600, 2019.

F. Petropoulos, S. Makridakis, V. Assimakopoulos, and K. Nikolopoulos. 'horses for courses' in demand forecasting. *European Journal of Operational Research*, 237(1): 152–163, 2014.

Fotios Petropoulos and Nikolaos Kourentzes. Forecast combinations for intermittent demand. *Journal of the Operational Research Society*, 66(6):914–924, 2015.

J. Pickands. Statistical Inference using Extreme Order Statistics. *The Annals of Statistics*, 3(1):119–131, 1975.

M. A.F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

R. Pincus, V. Barnett, and T. Lewis. Outliers in Statistical Data. 3rd Edition. *Biometrical Journal*, 37(2):256, 1995.

R.C. Prim. Shortest connection networks and some generalizations. *Bell Systems Technology Journal*, 36:1389–1401, 1957.

J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer, New York, 1997.

J. O. Ramsay, G. Hooker, and S. Graves. *Functional Data Analysis in R and Matlab*. Springer, New York, 2009.

S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova. Anomaly detection in dynamic networks: A survey. *WIREs: Computational Statistics*, 7(3):223–247, 2015.

Yasmine Rashed, Hilde Meersman, Eddy Van De Voorde, and Thierry Vanelslander. Short-term forecast of container throughout: An ARIMA-intervention model for the port of Antwerp oa. *Maritime Economics and Logistics*, 19(4):749–764, 2017.

N. Rennie, C. Cleophas, A. M. Sykulski, and F. Dost. Identifying and responding to outlier demand in revenue management. *European Journal of Operational Research*, 293:1015–1030, 2021a.

N. Rennie, C. Cleophas, A. M. Sykulski, and F. Dost. Detecting outlying demand in multi-leg bookings for transportation networks. *arXiv (pre-print)*, 2021b.

M. Ribatet and C Dutang. *POT: Generalized Pareto Distribution and Peaks Over Threshold*, 2019. URL `https://CRAN.R-project.org/package=POT`. R package version 1.1-7.

S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

J. Schuijbroek, R. C. Hampshire, and W. J. van Hoeve. Inventory rebalancing and vehicle routing in bike sharing systems. *European Journal of Operational Research*, 257(3):992–1004, 2017.

C. Schütze, C. Cleophas, and M. Tarafdar. Revenue management systems as symbiotic analytics systems: insights from a field study. *Business Research*, 13(3):1007–1031, 2020.

Author A J Scott and M Knott. A Cluster Analysis Method for Grouping Means in the Analysis of Variance. *Biometrics*, 30(3):507–512, 1974.

Susan Shaheen, Stacey Guzman, and Hua Zhang. Bikesharing in Europe, the Americas, and Asia. *Transportation Research Record*, 2143:159–167, 2010.

R. L. Smith. Maximum Likelihood Estimation in a Class of Nonregular Cases. *Biometrika*, 72(1):67–90, 1985.

Soheil Sohrabi and Alireza Ermagun. Dynamic bike sharing traffic prediction using spatiotemporal pattern detection. *Transportation Research Part D: Transport and Environment*, 90(December 2020):102647, 2021.

A. K. Strauss, R. Klein, and C. Steinhardt. A review of choice-based revenue management: Theory and methods. *European Journal of Operational Research*, 271(2): 375–387, 2018.

P. D. Talagala, R. J. Hyndman, K. Smith-Miles, S. Kandanaarachchi, and M. A. Muñoz. Anomaly Detection in Streaming Nonstationary Temporal Data. *Journal of Computational and Graphical Statistics*, 2019.

K. Talluri and G. van Ryzin. Revenue Management Under a General Discrete Choice Model of Consumer Behavior. *Management Science*, 50(1):15–33, 2004.

K. T. Talluri and G. J. Van Ryzin. *The Theory and Practice of Revenue Management*. Kluwer Academic Publishers, 2004.

Antti Talvitie and Daniel Kirshner. Specification, transferability and the effect of

data outliers in modeling the choice of mode in urban travel. *Transportation*, 7(3): 311–331, 1978.

C. Temath, S. Pölt, and L. Suhl. On the robustness of the network-based revenue opportunity model. *Journal of Revenue and Pricing Management*, 4(9):341–355, 2010.

A. Tharwat. Classification assessment methods. *Applied Computing and Informatics*, pages 1–13, 2018. in press.

R.C. Tsay. *Analysis of Financial Time Series*. John Wiley and Sons, 2002.

Visual Crossing. Weather data api, 2021. `https://www.visualcrossing.com/weather-api` [Accessed: 2021-04-01].

S. Vock, L.A. Garrow, and C. Cleophas. Clustering as an approach for creating data-driven perspectives on air travel itineraries. *Journal of Revenue and Pricing Management*, 2021.

L. R. Weatherford. The history of forecasting models in revenue management. *Journal of Revenue and Pricing Management*, 15(3):212–221, 2016a.

L. R. Weatherford and P. P. Belobaba. Revenue impacts of fare input and demand forecast accuracy in airline yield management. *The Journal of the Operational Research Society*, 53(8):811–821, 2002.

L. R. Weatherford and S. E. Bodily. A taxonomy and research overview of perishable-

asset revenue management: Yield management, overbooking, and pricing. *Operations Research*, 40:831–844, 1992.

L. R. Weatherford and S. E. Kimes. A comparison of forecasting methods for hotel revenue management. *International Journal of Forecasting*, 19(3):401–415, 2003.

L. R. Weatherford, S. E. Bodily, and P. E. Pfeifer. Modeling the Customer Arrival Process and Comparing Decision Rules in Perishable Asset Revenue Management Situations. *Transportation Science*, 27(3):239–251, 1993.

Larry Weatherford. The history of forecasting models in revenue management. *Journal of Revenue and Pricing Management*, 15(3-4):212–221, 2016b.

S. S. Wilks. Determination of Sample Sizes for Setting Tolerance Limits. *The Annals of Mathematical Statistics*, 12(1):91–96, 1941.

Chengcheng Xu, Junyi Ji, and Pan Liu. The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets. *Transportation Research Part C: Emerging Technologies*, 95(September 2017):47–60, 2018.

Kiyoung Yang and Cyrus Shahabi. A PCA-based similarity measure for multivariate time series. *MMDB 2004: Proceedings of the Second ACM International Workshop on Multimedia Databases*, pages 65–74, 2004. doi: 10.1145/1032604.1032616.

W. Yuan, L. Nie, X. Wu, and H. Fu. A dynamic bid price approach for the seat inventory control problem in railway networks with consideration of passenger transfer. *PloS one*, 13(8):e0201718, 2018.

C. T. Zahn. Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computers*, C-20(1):68–86, 1971.

R. H. Zeni. The value of analyst interaction with revenue management systems. *Journal of Revenue and Pricing Management*, 2(1):37–46, 2003.

Yajun Zhou, Lilei Wang, Rong Zhong, and Yulong Tan. A Markov Chain Based Demand Prediction Model for Stations in Bike Sharing Systems. *Mathematical Problems in Engineering*, 2018.

Siying Zhu. Stochastic bi-objective optimisation formulation for bike-sharing system fleet deployment. *Proceedings of the Institution of Civil Engineers - Transport*, 2021.