

Time Series Forecasting and Categorization: an Empirical Study on Intermittent and Lumpy Demand

Rodrigo Cardoso Miranda Santos

Master's Dissertation

Supervisor: Prof. Maria Beatriz Brito Oliveira



Mestrado em Engenharia e Gestão Industrial

2022-07-11

Abstract

Precise forecasts serve as a strong foundation for decision-making and guide the course of action inside the organization, leading to the maximization of productivity and efficiency of business operations. Within the wide research field of forecasting, intermittent and lumpy time series have often been overlooked in the past, either due to the daunting effort required in their analysis or, sometimes, general misconceptions. Although practitioners typically associate these series with items of lesser value for the business context, ignoring them might limit the maximum achievable return as these products might have high holding costs. Moreover, due to these series' dual uncertainty in positive value occurrence and magnitude, conventional forecasting approaches often fail to produce satisfactory outcomes. Given the limited time budget detained by decision-makers, these instances are either disregarded or treated with a suboptimal strategy.

This thesis stems from a project at a boutique business analytics firm which consists in the development of a forecasting tool that has reached a development stage of consideration for the accuracy of forecasts generated for irregular time series (lumpy and intermittent). Regarding this issue, a cross-cutting concern in the work mirrored in this document was to create diversity in the methods devoted to predicting these series. These methods include parametric approaches, a machine learning approach that splits the global forecasting problem into two distinct and almost independent challenges, and temporal aggregation techniques. In addition, there is a desire to optimize the use of the decision-makers' limited time in reviewing these forecasts and developing a proper strategy for handling them. The issue is solved by clustering, producing groups of time series requiring parallel forecasting validation efforts and presenting similar importance for the operation.

An experiment was designed to evaluate the outcomes of the proposed approaches. The exercise was based on five datasets from diverse Portuguese enterprises operating in various industries. The findings suggest that implementing these strategies significantly improves the predictive performance of the tool, increasing the overall accuracy, in generated predictions, on average, by 34.04% regarding the mean absolute scaled error and reducing bias by 26.14%. The clustering experiment also accomplished its intended purpose since each cluster was able to receive operational planning-related recommendations.

This work contributes in two ways to the present state of research. First, empirical applications of numerous approaches are offered, with adjustments made to some of them to open up new deployment possibilities. Second, a process adaptable to various business situations is provided to bridge the gap between automated and human forecasting.

Resumo

Prognósticos rigorosos constituem uma base sólida para a tomada de decisões e estabelecimento do plano de ação que leva à maximização da produtividade e eficiência dos processos internos de uma organização. Embora a disciplina de geração de previsões apresente uma vasta gama de estudos teóricos e práticos, as séries cronológicas intermitentes e *lumpy* foram comumente negligenciadas no passado, quer devido ao esforço árduo exigido na sua análise, quer, por vezes, devido a conceções gerais erradas. Embora profissionais da indústria associem tipicamente estas séries a artigos de menor valor para o contexto empresarial, ignorá-los pode limitar o retorno máximo alcançável, uma vez que estes produtos podem ter elevados custos de manutenção/posse associados. Além disso, devido à dupla incerteza na frequência e magnitude de valores positivos, abordagens convencionais de previsão muitas vezes não conseguem produzir resultados satisfatórios. Dado o orçamento limitado de tempo detido pelos decisores, estes tópicos são ignorados ou tratados com uma estratégia sub-ótima.

Esta tese deriva de um projeto numa empresa de *business analytics* que consiste no desenvolvimento de uma ferramenta de *forecasting* que atingiu uma fase de desenvolvimento de preocupação com a precisão das previsões geradas para séries temporais irregulares (intermitentes e *lumpy*). Relativamente a esta questão, uma preocupação transversal no trabalho espelhada neste documento foi a de criar diversidade nos métodos dedicados a prever estas séries. Estes incluem abordagens paramétricas, uma abordagem de *machine learning* que envolve dividir o problema de previsão global em dois desafios distintos e quase independentes, e técnicas de agregação temporal. Além disso, existe um desejo de otimizar o uso do tempo limitado dos decisores na revisão destas previsões e no desenvolvimento de um plano adequado para lidar com os itens associados. Esta questão foi abordada como um problema de *clustering*, produzindo grupos de séries temporais que requerem esforços semelhantes de validação de previsões e que apresentam importância similar para a operação.

Foi concebida uma experiência para avaliar os resultados das abordagens propostas. O exercício foi baseado em cinco conjuntos de dados de procura de diversas empresas portuguesas que operam em vários setores. Os resultados recolhidos sugerem que a implementação destas estratégias melhora significativamente a performance preditiva da ferramenta, verificando-se um aumento da precisão global nas previsões geradas em 34,04% em relação ao erro médio absoluto escalonado e uma redução do enviesamento de 26,14%. O exercício de *clustering* permitiu gerar sugestões relacionadas com gestão de operações para cada classe, cumprindo assim o seu propósito.

Este trabalho tem uma contribuição bipartida para o atual estado de investigação. Em primeiro lugar, são oferecidas aplicações empíricas de numerosas abordagens, com ajustes aplicados a algumas por forma a criar novas oportunidades de implementação. Em segundo lugar, é fornecido um processo adaptável a várias situações empresariais para atenuar a lacuna entre a previsão automatizada e humana.

Acknowledgments

This dissertation concludes my Master's degree in Industrial Engineering and Management at FEUP. It marks the most significant milestone in my academic journey so far - a path that I could never have guessed the many endeavors and fortuitous occurrences it would present, but I am grateful that it did.

First, I would like to thank the people at LTPlabs, who provided me with a great challenge and wide space to employ my creativity to the fullest and improve my technical expertise. I am grateful for all the orientation and constructive feedback given by Eng. Andreia Cardoso, André Morim, João Alves, and Dr. Luis Guimarães, whose vast knowledge certainly contributed to the expansion of mine. A special acknowledgment goes to Eng. Francisco Amorim, who was assigned a supervisor role over my dissertation. Besides the otherworldly patience demonstrated in reviewing my work, discussing the best course of action for each stage, and being available to resolve my doubts, just to name a few efforts undergone towards my success, you showed an incredible character - comprehensive, approachable, joyful, and considerate - and always encouraged me to push further and defy myself, never limiting my capabilities. I've learned a lot from working closely with you, and I can only wish you the best and that we may meet again in the future.

I would also like to praise Dr. Beatriz Oliveira, my supervisor at FEUP, for the promptness demonstrated in providing constructive feedback every time it was requested, and for the constant concern with the status of the work development.

Because life is not only about work, I have to thank the people who contributed to the person I am today. A special appraisal goes to my family - Armanda, Beatriz, Catarina, João Pedro, José, Júlia, Pedro, Sandra - the names of those who are closest to me and from whom I learned the most and from a tender age. The path I walked had the potential to be treacherous, but every obstacle was overcome thanks to all your efforts and dedication.

My last but certainly not least acknowledgments are due to my close circle of friends. From those who have been with me for the longest time, to those who shared these last five years with me: thank you for all the moments shared along this journey, good and bad alike. Who I am today is a contribution from every single one of you.

"It is not our part to master all the tides of the world, but to do what is in us for the succour of those years wherein we are set, uprooting the evil in the fields that we know, so that those who live after may have clean earth to till."

J.R.R. Tolkien

Contents

1	Introduction	1
1.1	Motivation	1
1.2	The “AIR” project at LTPLabs	3
1.3	Dissertation outline	4
2	Literature Review	5
2.1	Intermittent and lumpy time series forecasting	5
2.1.1	Challenges in the field	5
2.1.2	Statistical methods	6
2.1.3	Machine learning methods	8
2.1.4	Temporal aggregation	9
2.1.5	Evaluation metrics	13
2.2	Time series classification	13
2.2.1	Time series categorization frameworks	13
2.2.2	Machine learning and time series classification	15
3	Problem Description	17
3.1	LTPlabs’ forecasting module	18
3.2	As-is situation and improvement opportunities	19
4	Methodology	21
4.1	Proposed approach structure	21
4.2	Forecasting methods	23
4.2.1	SBA and TSB algorithms	23
4.2.2	A rolling window approach to ADIDA and MAPA	24
4.2.3	A dual strategy motivated by the Croston method	27
4.3	Assessing time series forecastability	29
5	Results	33
5.1	Case studies overview	33
5.2	Forecasting methods results	34
5.2.1	Classification and regression	35
5.2.2	Temporal aggregation results	37
5.2.3	Overall forecast performance assessment	39
5.2.4	Hierarchical temporal reconciliation	41
5.3	Clustering and categorization findings	42

6	Discussion and conclusions	47
6.1	Practical implications	47
6.2	Future research	49
A	Forecast performance metrics	57
B	Variables included and hyperparameters tuned in the GBM models	59
C	Input variables for clustering	61
D	Dataset characterization according to established frameworks	63
E	Implemented methodologies results	67
F	Clustering procedure results	77

Acronyms and Symbols

ADI	Average inter-Demand Interval
ADIDA	Aggregate Disaggregate Intermittent Demand Approach
AUC	Area Under Curve
AUC ROC	Area Under the Receiver Operating Characteristic Curve
B2B	Business-to-Business
B2C	Business-to-Consumer
CRISP-DM	CRoss Industry Standard Process for Data Mining
DL	Deep Learning
GBM	Gradient Boosting Machine
MAPA	Multiple Aggregation Prediction Algorithm
MAPE	Mean Absolute Percentage Error
ME	Mean Error
ML	Machine Learning
MSE	Mean Squared Error
OEM	Original Equipment Manufacturers
SAE	Sum of Absolute Errors
SBA	Syntetos-Boylan Approximation
SEIL	Smooth, Erratic, Intermittent and Lumpy Classification framework
SES	Simple Exponential Smoothing
SKU	Stock Keeping Unit
SSE	Sum of Squared Errors
TSB	Teunter, Syntetos and Babai method
VRC	Variance Ratio Criterion
wMAPE	Weighted Mean Absolute Percentage Error
wMASE	Weighted Mean Absolute Scaled Error

List of Figures

1.1	Decision-making matrix regarding product criticality.	2
2.1	The MAPA forecasting algorithm. Source: Kourentzes et al. (2014)	11
2.2	An example of a 3-level temporal hierarchy. Source: Spiliotis et al. (2019)	12
2.3	The SEIL framework for demand characterization. Adapted from: Syntetos and Boylan (2005)	14
2.4	Example series for which the coefficient of variation fails to determine which is more challenging to predict.	16
3.1	Erratic time series decomposition in its smooth and lumpy components.	17
4.1	Blueprint of the methodology proposed	22
4.2	Forecasting module time series cross-validation scheme.	23
4.3	Temporal rolling window sales aggregation scheme.	25
4.4	Example of forecasts generated at different aggregation levels.	26
4.5	Procedure for final forecast generation using the dual approach.	29
4.6	Elbow chart example with suitable selections for the optimal number of clusters. This is a frequent heuristic in mathematical optimization to pick a point where the added expenditure isn't worth declining benefits. Increasing the number of groupings will enhance the fit (explain more variance) since there are more parameters (more clusters), but at some point (past the elbow), over-fitting is more prominent.	31
5.1	Relative variable importance of features for the classification model.	37
5.2	wMASE values for ADIDA, MAPA and the best currently implemented (original) method.	38
5.3	Overall outcomes from the proposed implementation evaluated on the test folds. .	40
5.4	Elbow chart obtained for the K-Means algorithm.	42
5.5	Distribution of the clusters generated throughout the S&OP matrix.	46

List of Tables

5.1	Datasets characteristics.	33
5.2	Average results obtained for the classification model.	35
5.3	validation demand size prediction error metrics.	36
5.4	Best performing algorithms in the validation folds.	39
5.5	Overall results of the multiple planning horizon techniques employed.	41
5.6	Best performing methods regarding MSE.	43
5.7	Number of time series per cluster for each best-performing methods.	44
5.8	Forecast accuracy measures, total and relative demand volume and time series quantity per cluster.	45
B.1	Hyperparameters tuned and values tested for the GBM models.	59
B.2	Input variables included in the GBM models.	60
C.1	List of forecast algorithms applied.	61
C.2	Features used for clustering algorithms.	62
D.1	SEIL data for each case study.	63
D.2	ABC data for each case study.	64
D.3	XYZ data for each case study.	64
E.1	Average results obtained in the validation folds of case study I.	67
E.2	Average results obtained in the test folds of case study I.	68
E.3	Average results obtained in the validation folds of case study II.	69
E.4	Average results obtained in the test folds of case study II.	70
E.5	Average results obtained in the validation folds of case study III.	71
E.6	Average results obtained in the test folds of case study III.	72
E.7	Average results obtained in the validation folds of case study IV.	73
E.8	Average results obtained in the test folds of case study IV.	74
E.9	Average results obtained in the validation folds of case study V.	75
E.10	Average results obtained in the test folds of case study V.	76
F.1	Characterization of the time series belonging to each cluster according to SEIL, ABC and XYZ analyses.	77
F.2	Best performing methods and results per cluster and point in the forecast horizon.	78
F.3	Average results obtained from selecting the best performing methods for each gap and class. The existing solution is indicated as “no_class”.	78

Chapter 1

Introduction

1.1 Motivation

The fierce and ever-growing competitiveness in many business sectors has set the decision-making task as a cornerstone of the success of any company. In the current Information Revolution era paradigm, knowledge in the form of data is increasingly recognized as a key organizational asset, continuously leveraged by decision-makers to make better-informed decisions (Citroen, 2011). These decisions create more realistic expectations regarding their outcome, allow for new perspectives to be considered, and ultimately provide more control over the problem. Time series forecasting arises as a task designed to inform crucial supply chain management decisions whilst serving as a guide for long-term strategy planning (Hyndman and Athanasopoulos, 2021).

The value of generating good forecasts has been highly recognized throughout history, and many records of successful applications along several business sectors are available to date (Nenni et al., 2013; Fildes and Kumar, 2002; Suganthi and Samuel, 2012). The most notorious benefits of such forecasts include risk reduction, efficient allocation of resources and personnel, opportunities for expansion, inventory accounting, operating costs reduction, and overall spend cutting, just to name a few.

Despite being commonly researched and employed in many industries, forecasting still presents considerable limitations regarding demand with intermittent or lumpy behavior. This behavior pertains to demand with occasional occurrence, exhibiting no orders at all at certain times, and potentially substantial fluctuation in order volume when it does occur, implying that a pattern is difficult to extract from historical records. These items usually have lower contributions to revenues than their faster-moving counterparts. However, significant cost implications - mainly related to stock holding costs - often arise due to high quantities of these slow-movers in a business product portfolio (Boylan and Syntetos, 2021). Numerous industries frequently handle items with these characteristics, especially those laboring in a business-to-business (B2B) context and forecasting at a low cross-sectional¹ levels, typically also manifesting in products nearing the end

¹Product categories that are nested within broader categories, establishing distinct hierarchies.

of their life-cycle (Willemain et al., 2004). As stated by Johnston et al. (2003), the weight of these items may account for up to 60% of an organization's inventory investment.

Naturally, products that display the previously described behavior tend to present higher forecast residuals than regular and stable demand flow products. This problem has created two issues that are at the core of the attention in the current paradigm of intermittent demand research: the need to develop new forecasting methods for this demand pattern and assessing a product's forecastability - a quantitative statistic used to evaluate the relevance of revenue/volume and the ease of predicting (demand characteristics such as variability, life cycle stage, intermittency, pattern, among others) in the portfolio of a company. The latter is especially significant for the decision-makers in a company, as this type of product often requires an extra step of validation and forecast value correction based on other exogenous information available or overall business knowledge. As business experts possess limited availability to undertake this task, it is very relevant for them to know what products will probably require a review in the forecast generated beforehand. This step makes the forecasting task more efficient and sets the expectations on the maximum forecast accuracy achievable (Bower, 2017). Figure 1.1 depicts the most common actions employed and efforts allocated in managing products regarding their predictability and relevance to the business context.

Revenue/Volume	HIGH	Statistical Forecast • Statistical forecasting • Higher Planner Intervention • Make-to-Stock/Purchase-to-Stock • Higher Safety Stock Levels S&OP Effort: Medium	Market Intelligence • High S&OP/CPFR Input • Stat Forecast + Human Intervention • Fixed Stocking Programs • Inventory Spend Budgets S&OP Effort: High
	LOW	Pure Statistical Forecast • Statistical forecasting • Minimal Planner Intervention • Make-to-Stock/Purchase-to-Stock • Lower Safety Stock Levels S&OP Effort: Minimal	Alternative Strategies • SKU Rationalization • Make-to-Order • Avoid high LT • Avoid Safety Stock S&OP Effort: Medium
		HIGH	LOW
		Forecastability	

Figure 1.1: Decision-making matrix regarding product criticality.

Having established the advantages that accurate demand forecasts bring to the operations management of a company and the value obtained from knowing in advance what products will require

added attention from the decision-makers, the scope of this dissertation can be presented as follows:

1. Development of a forecasting framework fit for the prediction of items presenting an intermittent or lumpy demand behavior;
2. Categorization of time series according to the need for demand forecast review and human validation.

The foundation of this academic document relies on the work developed for an internal project at an analytical consultancy firm. This project is part of a bigger platform developed by the company, and it acts as a basis for the forecasting services that clients (other businesses) often require. Additionally, it assists the consultants by providing a set of standard tasks, typical in these kinds of services, programmed logically and sequentially, requiring only adaptation to the current client’s data and needs. Five distinct real-world datasets from Portuguese companies spread across several sectors are applied in a comprehensive evaluation to test the effectiveness of this work.

1.2 The “AIR” project at LTPLabs

LTPLabs is a boutique analytical-driven management consultancy company providing highly tailored solutions to its clients. “AIR”, an internal company project currently under development, aims to become a leading platform for fast-track analytics, enabling the company to present a very short time-to-market of its customized solutions.

Demand forecast emerges as a component of the “AIR” platform analytical suite. For this purpose, a forecast module has been developed, aiming to combine a plethora of algorithms, data analysis tools, and evaluation metrics in a single framework. The solution reduces the lead time of forecasting projects by offering a standardized and mechanized basis for repetitive tasks and a scalable service that guarantees all projects follow state-of-the-art methodologies.

Despite presenting state-of-the-art content regarding forecasting methods, the module currently includes only a few methods devised for intermittent or lumpy demand forecasting. New efficient techniques are necessary to reduce forecasting projects’ lead time and improve overall forecast accuracy when dealing with this data.

Regarding information to the client about product forecastability, the module presents several analytical tools that consider both time series data and product relevance in sales. These complement themselves and form a pool of time series identifiers that can help steer the attention of the decision-makers. However, additional demand predictability diagnostic information could prove itself valuable, including error values obtained from the forecasting exercise, as the information provided contains added realism.

The work developed for this thesis attempts to mitigate the previously mentioned shortages in the forecasting module.

1.3 Dissertation outline

This first chapter introduced the problem and context tackled in this dissertation. The remaining structure of this dissertation will now be described. Chapter 2 presents theoretical explanations on the topic, accompanied by empirical applications and current research efforts. Chapter 3 describes the problem to be addressed by this thesis. The following chapter, 4, presents a detailed explanation of the methodology adopted to tackle the issue, with chapter 5 exposing and commenting on the results extrapolated from this application. Lastly, chapter 6 presents conclusions derived from this work and discusses the enhancements and future steps that could follow.

Chapter 2

Literature Review

The present chapter is divided in two main topics. The first (section 2.1) exposes the current research efforts employed in the intermittent and lumpy behavior forecasting area, along with theoretical background on the subjects. Subsection 2.1.1 showcases the main challenges in generating accurate forecasts for irregular series. A review of the existing literature on parametric methods (subsection 2.1.2) and non-parametric methods (subsections 2.1.3 and 2.1.4) is then provided. Section 2.2 summarizes the main time series categorization frameworks and their applications.

2.1 Intermittent and lumpy time series forecasting

Intermittent and lumpy time series prediction has been a topic of discussion for many years. The forecasting task associated with these patterns can be seen as challenging, mildly speaking. Intermittence is proportional to the frequency of demand-free intervals, while demand size variance can dictate the lumpiness of a series (Syntetos and Boylan, 2005). In the following subsections, a deeper disclosure of this area will be carried out to provide theoretical and state-of-the-art context to this thesis.

2.1.1 Challenges in the field

As stated previously in the introductory chapter, forecasting is a ubiquitous task. Although much attention has been given to modeling time series of fast-moving items (Thomakos and Nikolopoulos, 2018), research in the field of irregular demand series has only recently begun to receive considerable engagement from academics and practitioners alike.

The difficulty in forecasting intermittent or lumpy time series stems from two sources of uncertainty: the inter-demand time intervals, which correspond to the chance of demand happening, and the quantity when it does occur. While the former dictates the degree of intermittence of a series, the latter relates to the irregularity of the demand magnitude behavior or, in other words, lumpiness (Li et al., 2022; Boylan and Syntetos, 2021).

Several approaches have attempted to tackle this issue, all sharing the assumption that demand follows a stationary mean process. These processes imply that demand variation only occurs due to

random deviations from the mean. Such is not the case for most of the demand patterns observed in practice (Silver, 2008). However, these models have previously shown good performance in various empirical situations (Syntetos and Boylan, 2005). In contrast, it is possible to assume that demand can present random fluctuations in its mean value over time, following a non-stationary mean process (Boylan and Syntetos, 2021). This behavior is typical in industrial environments that exhibit a trend, seasonal patterns, and business and product life cycles (Tunc et al., 2011). Capturing these elements constitutes an additional challenge in generating accurate forecasts for these time series.

2.1.2 Statistical methods

Parametric methods assume that the values in a time series follow an hypothesized probability distribution, whose mean and variance are determined using a forecasting method. While this strategy can theoretically be applied with any sort of data and forecasting method, traditional strategies often produce erroneous forecasts for intermittent demand. This is because demand arrivals are coupled with a size dispersion of demand. Tailored statistical methods for this type of pattern and others often used will be discussed in the paragraphs bellow.

A method widely used in intermittent time series forecasting is Simple Exponential Smoothing (SES), often referred to in the literature as Exponentially Weighted Moving Average (EWMA). Although initially developed to handle fast non-trended non-seasonal demands (Davies and Brown, 1960), its simplicity and acceptable performance have led to its widespread implementation in slow-moving and intermittent demand items. Equation 2.1 describes the way forecasts are generated using this method:

$$\hat{y}_{t+1|t} = \sum_{j=0}^{t-1} \alpha(1-\alpha)^j y_{t-j} + (1-\alpha)^t \ell_0 \quad (2.1)$$

Where y_{t-j} is the actual value of the series observed at a time $t-j$, $\hat{y}_{t+1|t}$ is the forecast value for time $t+1$ knowing the actual value at time t , ℓ_0 is the value used for initialization, and α represents the smoothing constant. Each forecast created by SES consists of a weighted average of past values. The choice of an appropriate value for the smoothing constant will result in more accurate forecasts, being that, the lower this value, the more relevant are most recent observations. Croston (1972) argued that a value of α between 0.1 and 0.3 is optimal, while Syntetos and Boylan (2001) state that it should be no larger than 0.15. A careful selection of ℓ_0 is also important, as this algorithm is sensitive to poor initialization (Boylan and Syntetos, 2021).

Croston (1972) examined the application of SES to intermittent time series forecasting and found the method to be biased. More specifically, he stated that if we isolate the forecasts generated after a demand occurring period, the technique exhibits positive statistical bias, consistently generating forecast values higher than the real ones.

The method developed by Croston (1972) aims to correct such bias and consists of creating separate forecasts for the demand value (when such takes place) and for the average interval of periods between demand occurrences. Individual SES estimates are made and updated after demand

occurs, using the same smoothing constant value for demand size and time interval. The mean demand value can then be calculated using the set of equations in 2.2:

$$\begin{aligned}\hat{\mu}_{S,t+1} &= \alpha S_t + (1 - \alpha) \hat{\mu}_{S,t} \\ \hat{\mu}_{I,t+1} &= \alpha I_t + (1 - \alpha) \hat{\mu}_{I,t} \\ \hat{d}_{t+1} &= \hat{\mu}_{t+1} = \frac{\hat{\mu}_{S,t+1}}{\hat{\mu}_{I,t+1}}\end{aligned}\tag{2.2}$$

Where:

S_t = Demand size in period t

I_t = Number of time periods between the last two demand occurrences

$\hat{\mu}_{S,t+1}$ = Average demand size forecast for period $t+I$

$\hat{\mu}_{I,t+1}$ = Average demand interval forecast until next occurring demand period

$\hat{\mu}_{S,t}$ = Average demand size forecast for period t

$\hat{\mu}_{I,t}$ = Last average demand interval size forecast

$\hat{d}_{t+1}$ = Forecast of mean demand for future time periods

Although this method does correct the bias inserted by SES, Syntetos and Boylan (2001) demonstrated that it suffers from a different type of bias due to the division undergone in the final step that generates the average demand for future periods, called “inversion bias”. This bias arises from the point that the expected value of the inverse of a random variable is not the same as the inverse of the expected value of a random variable. This fact leads to an overestimation of the probability of demand occurrence:

$$\mathbb{E}\left(\frac{\hat{\mu}_{S,t+1}}{\hat{\mu}_{I,t+1}}\right) = \mathbb{E}(\hat{\mu}_{S,t+1})\mathbb{E}\left(\frac{1}{\hat{\mu}_{I,t+1}}\right)\tag{2.3}$$

$$\mathbb{E}\left(\frac{1}{\hat{\mu}_{I,t+1}}\right) \neq \frac{1}{\mathbb{E}(\hat{\mu}_{I,t+1})}$$

Syntetos and Boylan (2005) went on to quantify the inversion bias introduced by Croston’s method and developed a correction factor for the average demand in future periods (see equation 2.4). This new estimator is known in the literature as the Syntetos-Boylan Approximation (SBA) method and is currently regarded as the benchmark method for intermittent time series forecasting, with a plethora of empirical applications across several industries.

$$\hat{d}_{t+1} = \hat{\mu}_{t+1} = \left(1 - \frac{\alpha}{2}\right) \frac{\hat{\mu}_{S,t+1}}{\hat{\mu}_{I,t+1}}\tag{2.4}$$

Teunter et al. (2011), in his paper, suggested a different forecasting method designed for intermittent and lumpy time series with the previously discussed methods acting as a basis. This method, named TSB (after Teunter, Syntetos, and Babai, the authors), updates demand probability

every period, unlike the Croston and SBA methods that only update the expected demand interval after a period of positive demand occurrence. By automatically adjusting forecasts downward after long periods with zero demand, TSB can handle item obsolescence with good performance. Moreover, following the conclusions derived by Schultz (1987), the method presents different smoothing constant values for demand size and demand occurrence probability estimation, which provide an added flexibility in the forecasting process.

2.1.2.1 Empirical applications of statistical methods

Empirical applications on the exposed methods reveal that Croston consistently outperforms SES when forecasting intermittent demand (Boylan and Syntetos, 2021).

When we compare Croston's method with SBA, the results are somewhat mixed. SBA regularly delivers more accurate projections than Croston for spare parts demand data. Croston, on the other hand, outperforms or ties with SBA in inventory metrics, depending on the data type. Croston beats SBA for extremely intermittent and declining demand patterns, but SBA slightly exceeds Croston for demand patterns with moderate to low intermittence and variability (Çerağ Pinçe et al., 2021).

Teunter et al. (2011) demonstrate that TSB is more accurate than SES, Croston, and SBA in terms of mean error (ME) and mean squared error (MSE) for both stationary and non-stationary demand patterns in a simulated analysis. Babai et al. (2014) show that TSB does not lead to significantly more accurate forecasts than the other techniques in a follow-up empirical work using industrial datasets. The authors also state that simulation studies are often designed for specific reasons and cannot represent many elements influencing real-world demand patterns. These include maintenance regimens, discrepancies between Original Equipment Manufacturers (OEMs) and end-users of spare parts, or unexpected bulk ordering of components by end-users.

Using the same datasets, they also demonstrate that the Modified SBA surpasses all other methods in terms of bias (ME) and MSE. The Modified SBA delivers more benefits due to increased precision, prominently for inventory with declining demand. Their research also shows that the Croston and SBA approaches should be implemented with caution in ERP systems, as these methodologies do not cope well with obsolescence or declining demand trends.

2.1.3 Machine learning methods

The exponential growth in computational power experienced in recent years has opened up a myriad of possibilities regarding the application of computationally intensive methods to intermittent and lumpy time series forecasting. This phenomenon has led practitioners to shift their interest to Machine Learning (ML) and Deep Learning (DL) methods as alternatives to statistical ones (Makridakis et al., 2018). The main advantage associated with these (and other types) of non-parametric methods is that no or very limited underlying assumptions about the data generating process are made, with the algorithms learning and evolving by trial and error by looking at training data (Barker, 2020).

These models can be created using either supervised or unsupervised learning. The former refers to models that can generalize knowledge derived from historical data with known target values, while the latter attempts to create groups with entries that have similar characteristics amongst themselves, with no known category or label (Berry et al., 2020). On the topic of supervised learning, one can also distinguish models into single and ensemble models. A single model is described as an instance of an algorithm that is trained to generate predictions, while an ensemble is described as a combination of single models that produces a more powerful, less biased estimator (Dietterich, 2000).

Boosting

Boosting is a form of ensemble models that combines a collection of weak regressors into a strong one. Originally devised by Schapire (1990), this method is based on a learning algorithm in the distribution-free (PAC) learning model. The authors developed an algorithm that sequentially fits “weak” regressors to different weightings of the observations in a dataset. The observations with worst prediction performance receive greater weight on the next iteration, and the weight of correctly classified observations is reduced. The final predictions are then obtained via application of an aggregation function of outputs given by each “weak” regressor, with greater weight given to more accurate predictors.

Gradient Boosting Machines (GBMs) are ensembles of several decision trees. The decision tree algorithm partitions the input variables recursively, using these variables for classification or regression by applying one or several decision functions over the best feature and splitting point, in the attempt to improve a given cost function (Swain and Hauska, 1977).

GBMs have been used successfully in intermittent and lumpy demand forecasting context. In particular, participants of the M5 forecasting competition, a competition that featured several intermittent and lumpy demand time series (Makridakis et al., 2021), who finished in the top five places all used some form of GBM, along with many other contestants. The outcomes of the competition highlighted that these algorithms can provide excellent results in this context, however it also proves that performance is affected by correct tuning of hyperparameters. Rožanec and Mladenčić (2021) conducted two GBM implementations inspired on the method proposed by Croston, one with the purpose of forecasting demand occurrence and the other of demand size estimation, which revealed excellent results when compared to classical statistical methods and the ADIDA framework referenced in subsection 2.1.4.

2.1.4 Temporal aggregation

In the previous sections, forecasting has been looked at from a single time frequency perspective, in which a time series exhibits several observations with null values. Through aggregation, such series may be manipulated to present a smoother demand pattern, which enables the use of traditional, more robust forecasting methods and models with wide application in predicting time series

that present low variability (Petropoulos et al., 2014). Aggregation strategies are non-parametric and include temporal or cross-sectional time series aggregation.

Temporal aggregation refers to altering the time frequency of a given series to a more coarse granularity (e.g., from monthly to quarterly) and using the generated sequence to obtain adequate forecasts. Aggregation can be done in the form of overlapping or non-overlapping time blocks: in the former case, each time window consists of unique historical observations, while in the latter, each time window is formed in a rolling fashion, adding the newest and dropping the oldest observation (Boylan and Babai, 2016). Forecasts at the original frequency can then be obtained by disaggregating the generated values. Several methodologies are available for this last task.

Willemain et al. (1994) were the first to recognize the potential that temporal aggregation presents in intermittent time series, and later, Nikolopoulos et al. (2011) introduced a temporal aggregation procedure devised specifically for intermittent and lumpy time series. The “Aggregate Disaggregate Intermittent Demand Approach” (ADIDA) consists of a generalized forecasting framework that can be split into the following three phases:

1. Aggregation of an irregular time series into non-overlapping time blocks, such that the average size of inter-demand time intervals is reduced;
2. Forecast generation at the aggregate level, using any technique deemed as appropriate for the time series characteristics;
3. Disaggregation of generated values to the original time frequency, using an adequate method.

In the same study, Nikolopoulos et al. (2011) empirically demonstrated the strength of the approach proposed by testing its effectiveness against established intermittent demand forecasting methods such as Croston’s and SBA, as well as methods used for regular time series forecasting. The results obtained on a Royal Air Force dataset comprising monthly demand data of 5,000 SKUs over seven years revealed that ADIDA might substantially improve a single method’s accuracy and variance, acting as a self-improving forecast mechanism.

The results obtained by Nikolopoulos et al. (2011) were verified in research conducted by Babai et al. (2012), sustaining the method’s robustness by evaluating it regarding inventory metrics besides forecast accuracy measures. In a paper released by Spithourakis et al. (2011), ADIDA was tested on the dataset from the M3 forecasting competition (Makridakis and Hibon, 2000), leading to a reduction of forecast error over simple methods.

Spithourakis et al. (2011) also pointed out the importance of carefully selecting an appropriate aggregation level, as this variable can significantly influence the results obtained. Based on this fact, a new forecast framework, also based on temporal aggregation, was proposed by Kourentzes et al. (2014) that is in most aspects similar to ADIDA, with the exception being the disaggregation step. While ADIDA considers only a single level for generating the finer granularity forecasts, MAPA (Multiple Aggregation Prediction Algorithm) combines forecasts generated for the original series from all levels, by calculating the mean or median of these. An illustration of the process is displayed in figure 2.1.

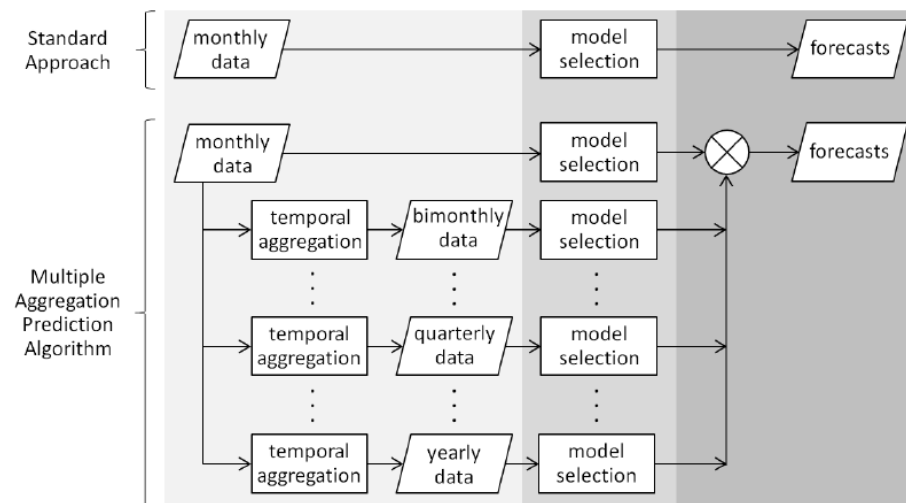


Figure 2.1: The MAPA forecasting algorithm. Source: Kourentzes et al. (2014)

By calculating final forecasts in this manner, the algorithm allows the information from all levels to be considered, which can be relevant especially due to trend and seasonality components. Although originally employed using exponential smoothing as the forecasting method, due to its capacity to model seasonality, trend, and level, the authors claim that in practice any technique is suitable for each aggregation level. An empirical evaluation was conducted for the SKUs of a cider brand, revealing that MAPA was the best performer both in terms of average bias and forecast accuracy (Kourentzes and Petropoulos, 2016). However, further empirical evidence on the subject is still scarce.

2.1.4.1 Temporal hierarchical reconciliation

Particular business applications may require the user to obtain forecasts at two or more temporal frequencies. In these instances, there may be conflicting results between the predictions generated for different aggregation levels, as forecasts obtained at finer granularities may not add up to more coarse ones. The concept of temporal hierarchies and their reconciliation was introduced to tackle this problem (Athanasopoulos et al., 2017).

Temporal hierarchies refer to the architecture underlying the temporally aggregated sales previously described, depicted across different aggregation levels. These, like hierarchical reconciliation in general, rely on the well-known benefits of forecast combination to be efficient. Forecasts are generated at various aggregation levels to create the base, incoherent forecasts. These initial forecasts are then combined with optimal weights based on ordinary or weighted least squares estimators. The result is a new set of predictions consistent at all levels.

Representation of the data in this organized structure creates a property that all the nodes directly below a more coarse node must add up to the values of that node. For the example illustrated in figure 2.2, data from quarters 1 and 2 must equal the first semester values, and both semesters added must totalize to the yearly data. Assuming that vector b contains bottom-level data, values for each component inside a hierarchy are provided by $y = Sb$, where S is a summing

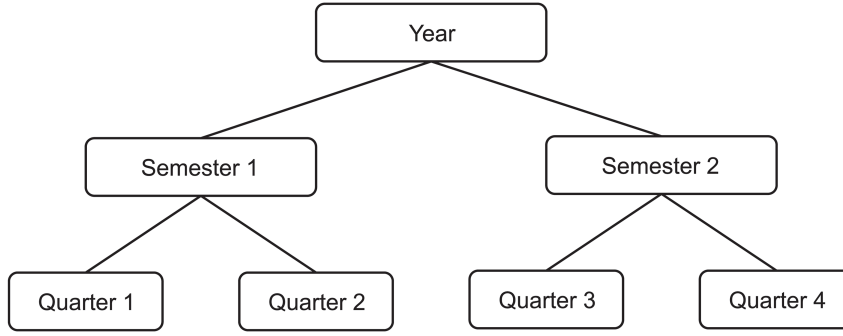


Figure 2.2: An example of a 3-level temporal hierarchy. Source: Spiliotis et al. (2019)

matrix of shape $m \times n$, where m is the total number of nodes in the hierarchy and n is the number of bottom-level components. The summation matrix for the hierarchy present in figure 2.2 is given by:

$$S = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Coherent forecasts at all levels in the hierarchy can then be calculated by equation 2.6. A full deduction of this expression can be found in Hyndman and Athanasopoulos (2021).

$$\tilde{y} = S(S'W^{-1}S)^{-1}S'W^{-1}\hat{y} \quad (2.6)$$

Where $\tilde{y}$ are the hierarchically reconciled predictions generated, S is the summation matrix described above, $\hat{y}$ are the incoherent base forecasts calculated for each temporal aggregation level, and W is the variance-covariance matrix of the corresponding base forecast errors. Calculating for W can be excessively demanding, particularly for forecast horizons superior to one time period (Wickramasuriya et al., 2019). In their book, Hyndman and Athanasopoulos (2021) provide four different simplifying approximations to W that have demonstrated good performance in both simulations and practice. Emphasis goes to the estimator $W = \text{diag}(S1)$, where 1 is a column unit vector of size m . This estimator is simple to use since it is independent of the data and forecasting methods, depending only on the data structure. Furthermore, its application outperformed other

variance-based estimators in the temporal scenario (Athanasopoulos et al., 2017).

Generating forecasts using the aforementioned procedure leads to greater accuracy in forecasts produced for several data frequencies. Nonetheless, even if the gains in accuracy are negligible, the property of coherence is highly attractive from a managerial point of view, since it allows for decision-making alignment across different levels of an organization.

2.1.5 Evaluation metrics

Having presented the main methods and models used for forecasting this type of series, it is relevant to discuss how to assess their performance. Many different accuracy metrics have been used in the literature, with no particular metric having an advantage over the rest. Some, however, present characteristics that are desirable for the evaluation of intermittent or lumpy time series, both in the statistical and business sense.

These series are characterized by having long periods of no demand. Consequently, a critical point of discussion in metric selection is division by null values. Percentage errors, such as MAPE, have an intrinsic disadvantage, as these present division by the actual value. These also place heavier penalties on positive errors than negative ones, creating asymmetry. Absolute or scaled metrics can overcome this disability. However, these don't often have direct interpretability and are scale-dependent (Hyndman and Koehler, 2006).

To mitigate such deficiencies, the solution commonly adopted is to evaluate the results according to several metrics of different types. This procedure ensures the robustness of the forecasts generated. Appendix A contains the formulas for the error metrics applied in this work.

2.2 Time series classification

With the discussion of appropriate intermittent and lumpy time series forecasting methods and the current state of empirical applications concluded, the topic of debate now becomes the categorization of time series. Real-world datasets often present large quantities of time series, for which individual analysis can become a cumbersome task. An automated (or guided) identification of the most appropriate forecasting methods for the data, better expectations management, and identification of the items that will require added human effort for validation becomes crucial in this scenario.

2.2.1 Time series categorization frameworks

The primary objective of any time series categorization is to classify items systematically based on the similarity of their many attributes. There are several company perspectives through which the study of time series attributes is relevant, including production and operations management. Within this angle, decision-makers intervening in inventory management and forecasting frequently leverage various categorization techniques. The way these time series are classified depends on the classification purpose (van Kampen et al., 2012).

The idea of dividing time series into groups of similar items regarding their demand pattern initially appeared in Williams (1984). The author presents a simple classification procedure, dividing demand patterns into smooth, sporadic, or slow-moving. Said division occurs by splitting demand variance during a lead time into its constituent causal parts, namely: the number of orders arriving in successive units of time, sizes of orders, and lead times. Eaves and Kingsman (2004) follow the same ideas and methodology categorizing demand patterns for forecasting and stock control policies.

Syntetos and Boylan (2005) analyzed the methods proposed by the aforementioned authors and claim that the *ad hoc* procedure of determining the cutoff values that split the categories renders both frameworks' applicability to several potential contexts questionable. Despite this statement, they recognize that the criteria used for time series categorization are meaningful. Inspired by the methodology followed by Johnston and Boylan (1996), Syntetos and Boylan (2005) further claim that it is more logical to categorize demand based on the actual performance of several forecasting methods across time series rather than arbitrarily label demand patterns and test which technique performs best for each category.

In the same study, the authors propose a categorization scheme that groups time series based on a pairwise comparison of the forecasting performance of the SES, SBA, and Croston methods. Through theoretical comparison of the MSE of these methods, and under the assumption that the same smoothing constant is employed in all estimates produced, the authors can reach standard cutoff values for the squared coefficient of variation of demand size (CV^2) and average inter-demand interval (ADI). The framework (commonly denoted as SEIL) then categorizes the time series as Smooth, Erratic, Intermittent, or Lumpy, according to figure 2.3.

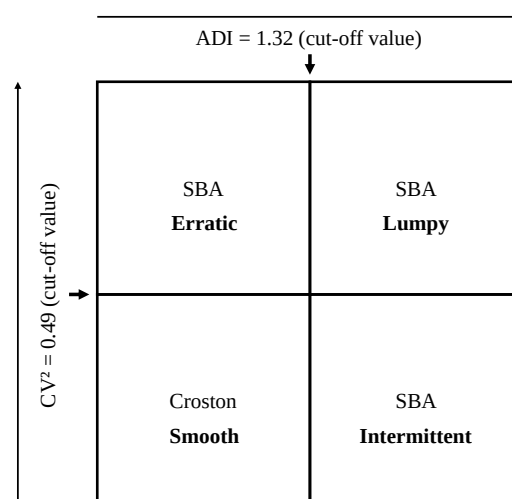


Figure 2.3: The SEIL framework for demand characterization. Adapted from: Syntetos and Boylan (2005)

All the previously mentioned frameworks are forecasting-based. However, another framework that can be used to characterize SKUs through their time series is the ABC framework, widely used in inventory management context. Sometimes referred to as the Pareto classification¹, this classification scheme is used to determine product categories of diminishing importance to the company, either evaluated through demand quantity or revenue contribution. Products are split into three classes: A, B, or C, with a typical contribution to these variables of about 80%, 15%, and 5%, respectively.

This principle has been validated multiple times in inventory management context (Teunter et al., 2009). Nonetheless, some criticism arises for this classification framework in a forecasting context. Neither the demand value nor the demand volume criteria adequately describe an item's demand pattern, as intermittency and sales weight are not directly related. Therefore, this framework is not optimal for selecting appropriate forecasting methods for each product.

To mitigate such difficulties, a multi-criteria variation of ABC can be implemented, such as ABC-XYZ. Generally, XYZ relates to the regularity in demand quantities ordered, with X being constituted by items with mainly stable demand, with rare fluctuations, Y presenting unstable but still predictable demand variability, and Z composed of completely irregular demand. Usually, the measure selected for this type of classification is the coefficient of variation, which is the ratio between the standard deviation of an item's demand over a certain period and the average demand (Scholz-Reiter et al., 2012). However, this measure can wrongly classify a time series as troublesome to forecast due to the presence of trend or seasonality (see figure 2.4). Nonetheless, this framework still proves its utility when the objective is to prioritize the effort to be made in the forecasting task.

2.2.2 Machine learning and time series classification

In the last two decades, time series categorization for forecasting using ML methods has seen a few applications being employed by the scientific community. These usually consist of multi-class algorithms or unsupervised clustering for similar demand patterns. As far as the latter goes, the most popular application is the creation of meta-learning algorithms. These allow the algorithm to learn about the structure of the time series in an automated manner, linking their main descriptive features to the best forecasting methods associated. This association creates natural clusters of time series with the same best-performing technique (Ma and Fildes, 2021). The usage of meta-learning algorithms has demonstrated enhanced forecasting accuracy compared to base forecasters, according to the literature (Ulrich et al., 2022).

Literature on the subject indicates that the majority of clustering time series-related studies fall into three categories: “whole time series clustering”, “subsequence clustering”, and “time point clustering”. Whole time series clustering is the similarity-based grouping of a collection

¹Named after Vilfredo Pareto (1848-1923), an Italian engineer who, around the beginning of the XX century, observed that 80% of the land in Italy was owned by 20% of the population. In fact, many different phenomena present this rule that 80% of the facts derive from 20% of the causes, and this has become acknowledged as the Pareto Principle (Sanders, 1987).

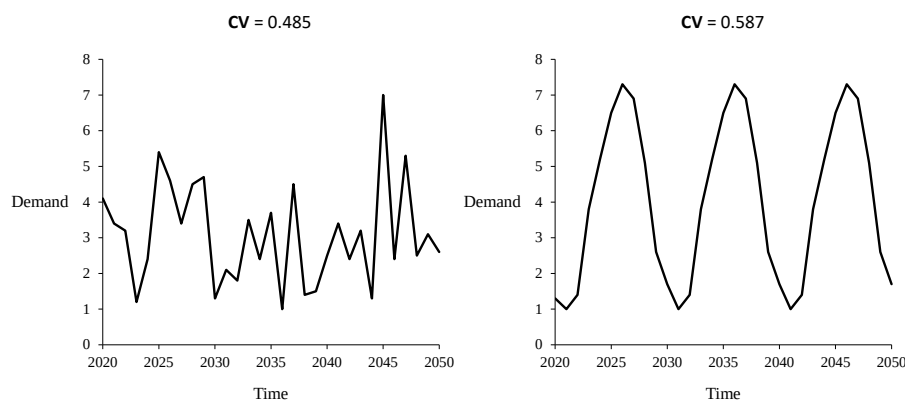


Figure 2.4: Example series for which the coefficient of variation fails to determine which is more challenging to predict.

of individual time series. Subsequence clustering refers to clustering on a set of subsequences taken from a time series using a sliding window. Clustering of time points based on their temporal proximity and the similarity of their corresponding values is known as time point clustering. As far as forecasting is concerned, whole time series clustering is the most relevant form of this strategy.

Time series can then be clustered using different techniques, such as model-based, feature-based, and shape-based approaches. In the first, clusters consist of time series that possess equal best parameters for a given model. For feature-based clustering, raw time series are translated into features that describe their pattern. Both of these techniques usually employ traditional clustering algorithms and distance metrics. As far as the shape-based approach is concerned, clusters are created by comparison of the raw time series data through non-linear stretching and contracting of the time axes. These usually employ conventional clustering algorithms - however, the distance metric evaluated is modified to handle this data.

Although several research on time series clustering has been conducted, the unique characteristics of time series data prevent the majority of conventional clustering approaches from functioning effectively with this type of data (Aghabozorgi et al., 2015).

Chapter 3

Problem Description

In the introductory chapter of this dissertation, we underlined the importance of robust forecasts for time series presenting irregular patterns with great intermittence in a business context. The subject, however, has had multiple applications in several areas besides Operations Management, such as renewable energy forecasting (Notton and Voyant, 2018), government's defense sector inventory management (Boylan and Syntetos, 2021), and even for forecasting the COVID-19 pandemic spread in the USA (Sbrana, 2022).

Generalizing the problem, Nikolopoulos (2021) states that the interest in such a form of series increases by considering that it is possible to disassemble erratic time series by splitting the entries above a certain user-specified threshold. This procedure generates two individual time series patterns: one presenting stable demand and another classified as intermittent or lumpy. Figure 3.1 provides an illustrative example of this procedure.

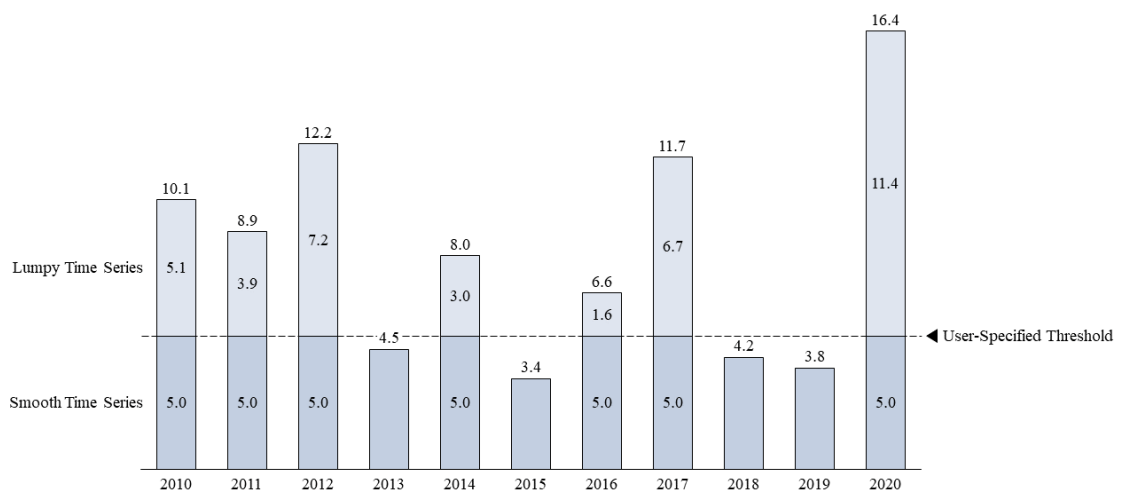


Figure 3.1: Erratic time series decomposition in its smooth and lumpy components.

A preemptive selection of time series according to their predictability can contribute to efficiently allocating the efforts of decision-makers and academics in time series analysis, optimizing their limited time budget. Additionally, it is common for a portion of an operation's knowledge to be absent from any tangible data source and rather present in the information owned by operators and gained through labor experience. Leveraging this knowledge in conjunction with the careful selection of time series may produce a more accurate view of the future, allowing for more realistic operations planning, creating space for more inventive and context-specific methodologies, and gathering attention to the most value-adding tasks.

When forecasting is performed regularly, validation for irregular time series and classification of time series become of the utmost significance. This case is that of LTPlabs, which keeps multiple projects that are either dedicated to forecasting or present this task as an intermediate stage on the path to a larger objective. These projects are often lengthy and require the consultants' daily attention. Furthermore, they often act as the initial collaboration between a new client and the company, setting the tone for the development of a potential partnership and becoming the gateway for other value-adding analytical approaches.

With the previous points in mind, LTPlabs decided to create an analytical tool dedicated to making the forecasting task of its projects more efficient. In the following section, an overview of this tool is presented. Furthermore, in section 3.2, an exposure of its current features regarding intermittent and lumpy forecasting techniques and time series criticality assessment is provided, alongside improvement opportunities.

3.1 LTPlabs' forecasting module

Before the introduction of the analytical tool previously mentioned, forecasting projects at LTPlabs followed a standard methodology for data mining and exploratory analysis, with initial steps of business understanding and data preparation that were unique to the project and final steps of modelling and evaluation that were highly repetitive. For each project, several forecasting methods were tested that required a complete implementation effort by the analyst that had to get acquainted with the inner workings of the algorithms. Similar models were employed across different use cases - thus, their standardization constituted an opportunity for operational efficiency improvement. With the accumulated experience of multiple forecasting projects spread across several sectors, the company evolved to a stage of analytical maturity where this knowledge was centralized in a unique framework.

The tool delivers a solid base for the forecasting task by gathering several recurrently employed forecast methods, models, and evaluation metrics in one coherent place. By doing so, the platform provides accelerated development until a satisfying baseline is achieved, leaving more time for the analyst to apply more complex and creative approaches to reach the best results possible. The centralized nature of the module also improves the support experience and ensures that methodological improvements are rolled out to the entire organization simultaneously.

Since its initial form, referenced in Fradique (2020), the module was deployed in several forecasting projects undergone by the company. Lead times were greatly diminished for these projects when compared to the previous methodology, and the module has been subject to several modifications in both the features presented and overall structure according to the expertise provided by each business case. As time passed and the module became increasingly stable and sturdy, concern grew for the cases of time series previously mentioned in chapter 2.

In the company's efforts to satisfy the customer's demands in these projects, offering a selection of the most crucial time series that require a manual evaluation and necessary adjustment has also been an area of attention.

3.2 As-is situation and improvement opportunities

To understand the possibilities of improvement in the areas mentioned in the previous section, there is a need to perform a diagnostic on the module's current features that contribute to the solution of the problem.

Regarding methods tailored for series with high intermittence and variation, the module offers SES and Croston's methods, the two most applied in the literature. As mentioned before, these series typically manifest low volume in demand, and concern for them only arises when the baseline of projections - that covers the most regular and impactful items - is already adequate. Several options exist for method implementations in the current scenario, considering the information offered in chapter 2. It is worth pinpointing that, when dealing with time series forecasting, no methodology or model acts as a "one size fits all" kind of solution. A possible path to tackle this problem is to employ various methods for the task, even creating combinations of those. This approach is utilised often throughout the literature on the subject to maximise the potential of prediction accuracy.

With three frameworks for time series analysis, the module already provides precautionary information regarding the forecasts produced. Although these have served as solid starting points to bring the attention of the decision-makers to time series that requires the most input from human expertise and exogenous factors, there is value to be added in this field in the generated forecasts diagnostic to the customer. This categorization becomes even more prominent when considering the limited time budget that decision-makers typically possess. Human attention should be directed according to the most critical time series regarding their ability to be projected accurately and their criticality for business. A methodology to do so is also presented in the following chapter.

Chapter 4

Methodology

This chapter describes the suggested methodology for developing new approaches to intermittent and lumpy time series forecasting, shifting the paradigm from a universal procedure to a differentiated treatment for regular and irregular series. The methodology undertaken for categorizing time series according to their criticality is also exposed. Section 4.1 aims at describing the proposed approach. Explanation and context are given on the structural aspects of the module and its utilization. Section 4.2 delves deeper into the implemented algorithms: the procedure followed and the mode of utilization for each. At last, section 4.3 describes the implementation made of a time series categorization framework using an unsupervised machine learning approach.

4.1 Proposed approach structure

The ambition of this thesis is twofold: first, to introduce diversity in forecasting methods for time series presenting intermittent or lumpy behavior; second, to devise a time series categorization technique that generates groups of similar predictability and relevance for the context. The coalition between the two tasks renders a complete framework for irregular time series forecasting.

In light of the oversights in the current approach, we are proposing a novel forecasting workflow described in Figure 4.1. Initially, data undergoes the initial stages of understanding and preparation as described in the CRISP-DM framework (Wirth and Hipp, 2000). Forecast methods destined for regular and irregular time series patterns are then applied and predictions generated. These predictions, along with other features related to each time series, serve as input for the following categorization task. Ultimately, a set of time series ordered by degree of criticality is provided to the decision-maker.

As previously stated in chapter 3, the work developed settles on a framework devised by the company. Therefore, to ensure that all the proposed steps, terminology, and decisions made in this methodology are understood, general knowledge of the module’s architecture is required.

Forecasts generated in the module are evaluated with a train-test split in rolling window fashion. For each time series, the user specifies how many windows should be generated, along with

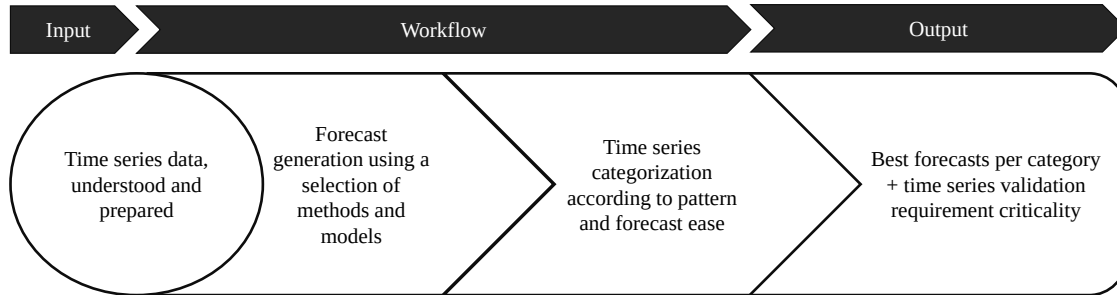


Figure 4.1: Blueprint of the methodology proposed

the desired forecast horizon and gap¹. Periodicity of sales (e.g. daily, weekly, monthly), to be denoted as data frequency from now on, may also be specified by the user, otherwise the framework will attempt to infer it by employing a best-effort heuristic.

Backtesting - applying a strategy or predictive model to historical data to determine its accuracy - is conducted using a temporal cross-validation technique. Folds are created in a rolling window fashion, with each fold expanding the historical observations by a given time increment over the previous one. Folds are segmented further into validation and testing folds. The validation folds are used to benchmark and select the best models, whereas the testing folds offer unbiased estimates of predictive performance. Each fold is comprised of historical observations upon which the models are re-fitted before making forecasts for the predetermined forecasting horizon. The length of the past values provided inside each fold varies based on the longevity of each time series. An example of this structure is demonstrated in figure 4.2.

In this example, the user indicated a total of four folds to be used for prediction, two for validation, and the remaining for testing. A prediction horizon of three time periods was also established, with a gap of two time periods. Generated forecasts are then evaluated and compared. Finally, the module can automatically offer the best projections based on the provided evaluations, or the user can combine several indicators and select the ideal forecasts.

The module, and consequently this thesis work, were entirely developed using the Python and R programming languages, using the “rpy2” package - an interface to run R code embedded in a Python process.

¹Number that indicates the point distance between the last historical observation used for forecasting and the first forecast generated.

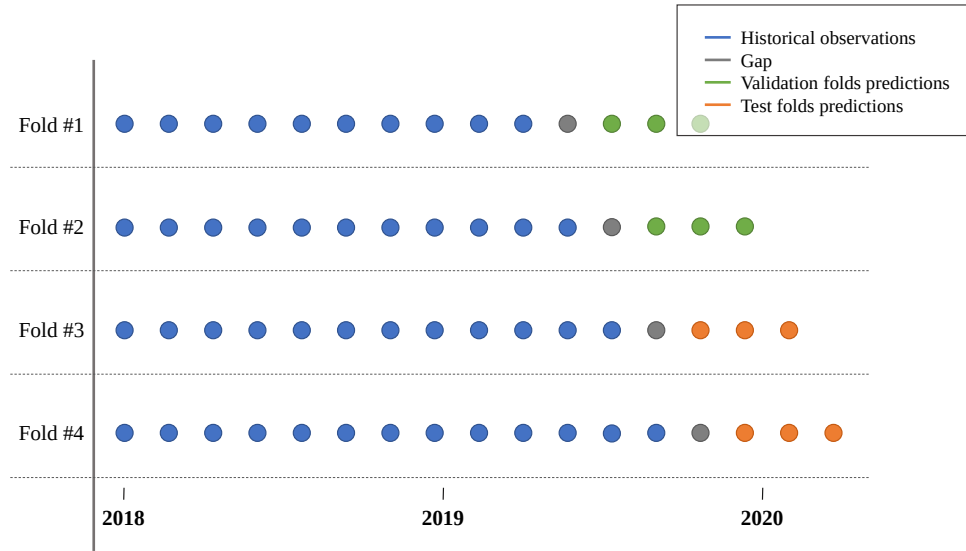


Figure 4.2: Forecasting module time series cross-validation scheme.

4.2 Forecasting methods

The methods developed sought algorithmic efficiency regarding computational time, complexity, and interpretability since these are often the concerns that arise from decision-makers. In this sense, there was a careful selection of all the techniques found in the literature, discarding some, such as bootstrapping methodologies, for their heavy processing effort and deep learning models for their lack of intelligible backing.

4.2.1 SBA and TSB algorithms

The first methods implemented pose as improvements to existing techniques in the module, according to a set of theoretical assumptions. The SBA and TSB algorithms, described in subsection 2.1.2, build on the predictions generated by the Croston method, which consist of two individual SES algorithms applied to demand occurrence and demand size.

When dealing with these algorithms, the key challenges are initialization and optimization. As demonstrated by Boylan and Syntetos (2021), SBA, and consequently TSB, are not robust to poor initial forecasts. Additionally, high smoothing constant values lead to early exhaustion of the weights attributed to each historical observation. Therefore, the length of the time series and the importance of the most recent values observed are decisive factors for the performance of this method.

For added flexibility, and following the work done by Schultz (1987), individual smoothing constants are calculated for each time series. In this case, the smoothing constant value used for bias correction corresponds to the demand interval smoothing constant. Backtesting was used to fine-tune the smoothing constants and the starting projection for each estimate. This strategy was

used in each fold by minimizing MSE using a Limited-memory BFGS² algorithm, implemented using the “optimize” module of the “SciPy” package³. Bounds were set for both variables under consideration for optimization. The value of the smoothing constant is constrained between 0.05 and 0.8. The initial forecast value is required to be positive for the following predictions to be positive.

4.2.2 A rolling window approach to ADIDA and MAPA

Sometimes, the original time series temporal frequency does not allow forecasting algorithms to pick up on factors such as trend or seasonality. The methods consisting of time series aggregation, projection, and disaggregation were conceived to fill these gaps. Furthermore, for a sufficiently coarse temporal granularity, intermittency can be removed. The absence (or shortage) of null historical values enables the application of traditional, more robust methods to the aggregated series, which may translate into more accurate disaggregated projections.

The implementation developed for the project offers a twist on the traditional procedure described in the literature for ADIDA and MAPA algorithms (referenced in chapter 2). Contrary to the initial discrete non-overlapping temporal aggregation of sales described by Nikolopoulos et al. (2011) and Kourentzes et al. (2014), we propose a novel way of performing temporal aggregation in an overlapping rolling window fashion. In this manner, the aggregated sales table maintains the original data periodicity. However, each aggregated historical record consists of the sum of the previous n original historical records, where n corresponds to the aggregated time bucket granularity. This twist diminishes the amount of data lost in non-overlapping aggregation. A visual representation of this aggregation is illustrated in figure 4.3.

In light of this modification, the first decision to be made for each algorithm is in defining the desired data frequencies of temporally aggregated sales to be generated. As discussed in chapter 2, determination of the most appropriate aggregation frequencies is not trivial, as high aggregation levels suffer from excessive smoothing and an overblown amount of aggregated sales tables leads to computational inefficiency when generating forecasts. Having the previous points in mind and knowing that this value is highly dependent on the dataset at hand, the amount and target data frequency values were left as a parameter for the user to specify.

Having created projections for each level of aggregation, the issue at hand becomes disaggregation to the original frequency. Predictions at higher aggregation levels were disaggregated to the source frequency using an equal-weighted average. Thus, the resulting original time point forecast is given by:

$$\hat{y}_{t,1} = \frac{\hat{y}_{t,n}}{n} \quad (4.1)$$

²An iterative optimization approach for tackling nonlinear optimization problems. Its limited-memory variant approximates its answer using a limited amount of computer memory (Liu and Nocedal, 1989).

³Detailed information about the tool is available at: <https://docs.scipy.org/doc/scipy/reference/optimize.html>.

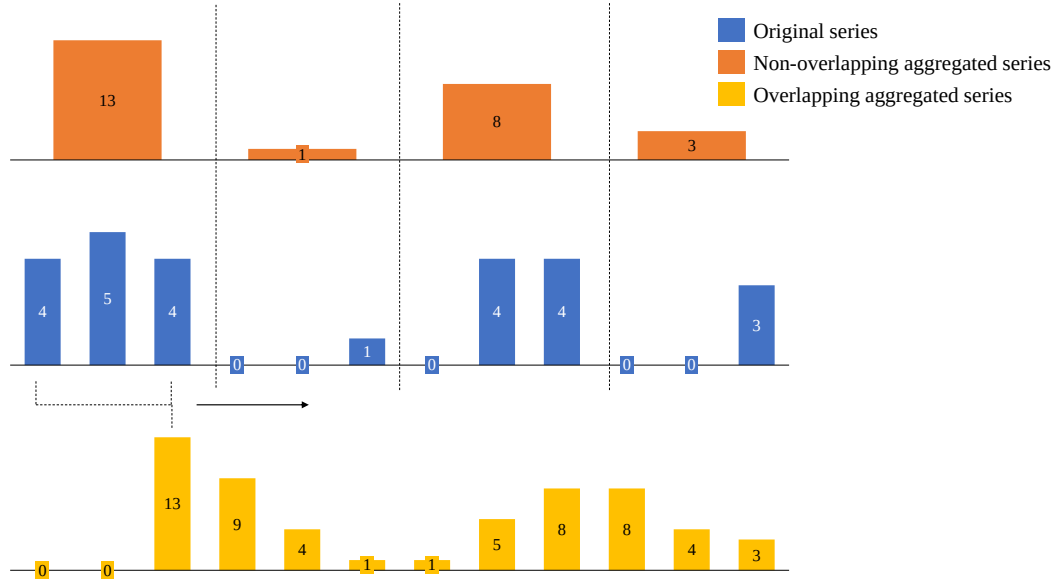


Figure 4.3: Temporal rolling window sales aggregation scheme.

Where n is the level of aggregation from which disaggregation ensues, $\hat{y}_{t,1}$ is the original time point forecast, and $\hat{y}_{t,n}$ is the forecast generated at aggregation level n . This notation will be used throughout the remainder of this thesis. The literature demonstrates that this procedure is a straightforward and effective method of getting disaggregated predictions, even topping the experiment done by Nikolopoulos et al. (2011).

The steps described thus far represent the shared procedures for both ADIDA and MAPA implementations. The decoupling point of these methodologies occurs in the selection of disaggregated forecasts. The ADIDA algorithm evaluates all aggregation levels and forecast methods tested and selects the one that possesses the best predictions at the original frequency, according to a metric specified by the user. The resulting procedure can be symbolized as ADIDA(A, F, D), where A represents the optimal level of aggregation determined, F the forecasting method used for aggregate forecasts generation, and D the disaggregation methodology employed in generating the final forecasts. A similar terminology can be found in Spithourakis et al. (2012).

The MAPA algorithm, on the other hand, attempts to introduce the influence of all aggregation levels used by performing a statistical average or median to all the disaggregate forecasts generated by each method to generate the final predictions.

As stated in section 2.1.4.1, the user may wish to generate forecasts at multiple temporal frequencies. For this purpose, a hierarchical forecast reconciliation method, similar to the one exposed in the same chapter, was introduced in the module. However, to create consistency between the forecasting projects developed by the company and the tool's structure, a rolling window overlapping approach was applied alternatively to discrete non-overlapping time buckets, similar to the adaptations made to ADIDA and MAPA. The flow of work of the modified algorithm can be described as follows:

1. Let D be a time series recorded at a given frequency. Given these historical records, suppose that predictions are required for the following one to four periods on aggregate. Initially, sales are aggregated temporally and forecasts for each level are generated using any forecasting method, similar to the original algorithm. Figure 4.4 illustrates this example. The original algorithm is based on the assumption that these forecasts are unbiased, a condition that is rarely fulfilled in practice. Assuming that validation folds are representative of future behavior, in the sense that there will be, on average, a consistent distance between actual values and predictions, forecasts can be corrected in terms of bias by the mean error (ME) calculated in the validation folds. A similar approach can be found in Spiliotis et al. (2019).

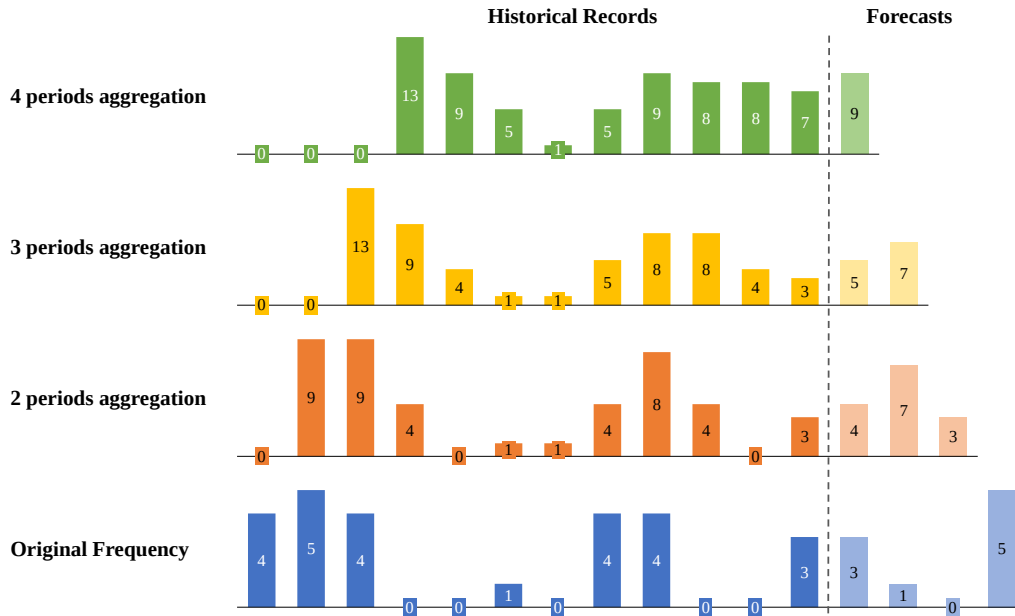


Figure 4.4: Example of forecasts generated at different aggregation levels.

2. The matrix that imposes the forecasts summation constraint for different levels is generated. Forecasts must be consistent with the format of aggregation of historical data. Consequently, the value of each point in the forecast horizon at any given aggregation level n above the bottom level is given by equation 4.2.

$$\hat{y}_{t,n} = \sum_{i=t}^{t+(n-1)} \hat{y}_{i,1} \quad (4.2)$$

The summation matrix for the example started above would then be:

$$S = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

3. The summation matrix S and forecasts are then supplied to the reconciliation procedure to obtain reconciled forecasts. The added requirement of non-negativity is not satisfied by the original algorithm for hierarchical reconciliation. Borrowing from Wickramasuriya et al. (2020) the problem is formulated as a least square minimization problem with non-negativity constraints. The algorithm used to solve the constrained quadratic is the Fortran subroutine BVLS (bounded variable least-squares) that solves linear least-squares problems with lower and upper bounds on the variables. This algorithm uses an active set strategy, like the algorithm that showed the best results in the previously mentioned academic work. This algorithm was implemented via the “optimize” module of the “SciPy” package. At the end of this step, the non-negative reconciled forecasts are obtained.

4.2.3 A dual strategy motivated by the Croston method

Given the dual-source of uncertainty in positive value occurrence and magnitude that intermittent and lumpy time series present, it makes sense to follow the logic provided by Croston (1972) in generating separate forecasts for each problem. The approach proposed in this section treats these topics as two independent issues. The first issue of demand occurrence is treated as a machine learning classification problem, generating a probability of prediction for each point in the forecast horizon based on previous demand. Following this stage, demand size is predicted using the best of two hypotheses. The first consists of creating a ML regression model similar in all aspects to the classification one except for the response variable. The other consists of generating predictions by applying forecasting methods to the positive data in the time series. By doing so, the emphasis may be placed only on the volume of demand when it occurs, removing any noise generated by zero values. In instances of time series with fewer positive values than the number of requested folds, the null values were replaced with the last observed positive value. The following topics expose the several phases of growing complexity that were surpassed in implementing this technique.

Time series feature extraction

The first step to take in the direction of generating forecasts following this approach is to transform the time series into a set of features that serve as input for each ML algorithm. Various features created were founded both on the knowledge the work done by Rožanec and Mladenić (2021) provides in reviewing the literature on this subject and the results obtained by them.

The features extracted aimed at describing the main aspects of historical demand. Variables include, but are not limited to, the number of periods since the last positive value, the average inter-demand interval, and the ratio of null values to time series length. These features are expanded by adding information related to the prediction date in the forecast horizon, item classification regarding ABC, XYZ, and SEIL analyses, and seasonal naïve predictions were also added as input variables. A full list of predictors considered for the GBM models can be found in table B.2.

Model training

The different nature of the machine learning models requires distinct target variables for each. For the classification task, the response consisted of a binary variable that equals one when the actual values are positive, and zero for null time series entries. For the regression problem, the response variable is the demand size.

Training for each model was conducted in an expanding window fashion. Each fold uses the preceding folds data to train the algorithm. As such, predictions for the first fold are not generated. The algorithm used for both tasks was implemented using the H2O machine learning interface for the R programming language.

Hyperparameter tuning and forecast evaluation

A gridsearch of algorithm parameters was conducted to maximize the area under the receiver operating characteristic curve (AUC ROC, or simply AUC) for the classification problem and minimize the MSE for the regression problem. The AUC calculates the area under a curve that demonstrates the performance of a classification model at all classification thresholds, proving an overall quality measure of the classification model. This metric is recommended when evaluating classification systems, as it possesses high discriminative power and statistical consistency (Ling et al., 2003). To convert the probabilities of demand occurrence into binary values a threshold maximizing the F1 score is selected. This metric is obtained by calculating the harmonic mean between the precision and the recall of the model.

This procedure was only employed in the first fold for which predictions are generated. The following folds are trained under the best combination of hyperparameters found by the gridsearch. A table summarising the different parameters susceptible to tuning for each model trained and the respective values tested can be found in table B.1. Final forecasts were obtained via multiplication of the output generated by each algorithm. Figure 4.5 demonstrates the procedure.

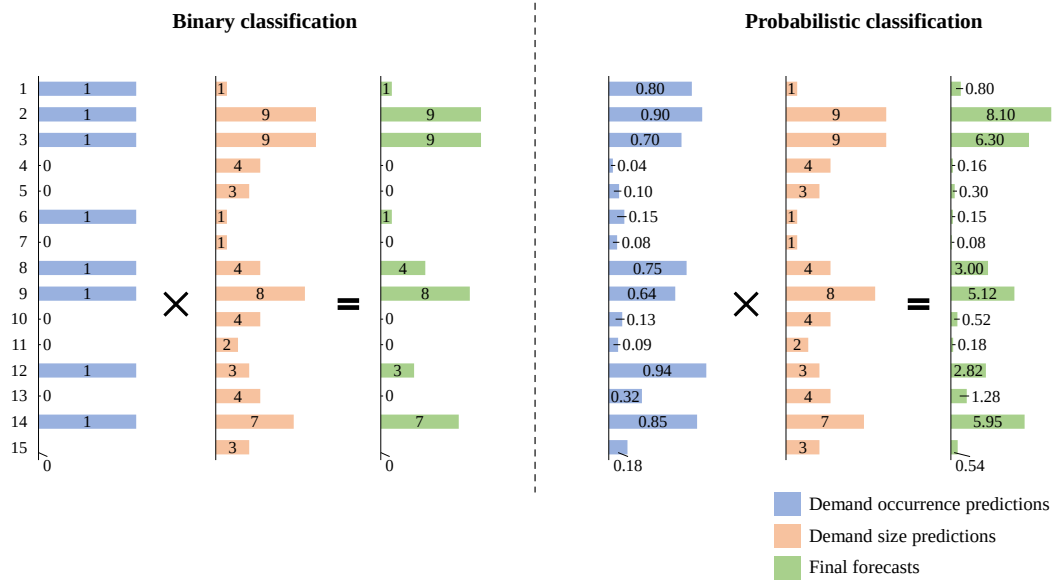


Figure 4.5: Procedure for final forecast generation using the dual approach.

4.3 Assessing time series forecastability

Up to this point, we presented and employed methods to improve the prediction accuracy of automatically generated forecasts for irregular data with several null observations. We now shift the focus from time series forecasting to time series grouping according to forecast ease and context criticality. The final goal for this task is to obtain descriptive statistics for each group created regarding the behavior of forecasts generated while assessing the relative importance of each series in the overall picture. By doing so, it is possible to present a priority arranged list of time series that require human expertise to produce more accurate forecasts or devise an alternative S&OP strategy to handle these series efficiently.

We employed feature-based whole-time series clustering to create the previously mentioned groupings that demonstrate similarity in predictive performance. This approach for clustering time series was selected owing to its relative efficiency in terms of processing time, simplicity in cluster creation - allowing for typical clustering techniques and distance metrics employment on the retrieved features - and purpose of the clustering process. Shape-based approaches were discarded due to the difficulty in finding suitable distance/similarity measures and the high computational cost associated with their calculation since a significant number of these measures require dynamic programming techniques for computation (Kotsakos et al., 2014). In the case of model-based approaches, the literature shows that these procedures have scalability issues, and their performance decreases as the clusters get closer to each other (Aghabozorgi et al., 2015). The following paragraphs describe the methodology followed that led to the final groupings.

Time series feature engineering and pre-processing

The first step in feature-based time series clustering consists of a dimensionality reduction procedure that generates features describing the original time series. Input variables created included several aspects of the series under analysis. Being that the primary objective of this clustering procedure was to add information relative to the predictions generated to the forecasts criticality report, the majority of features created are statistical measures that describe the error metrics obtained by employing multiple forecast methods for each series. Features include the mean, median, and standard deviation, among other numerical measures for each error metric. Additionally, a handful of descriptive statistics of the raw time series data were included, along with the series classification according to ABC, XYZ, and SEIL analyses. Appendix C contains a comprehensive list of the extracted features and the forecast methods employed.

For forecast evaluation purposes, two absolute error metrics were used - the absolute errors sum (SAE) and the squared error sum (SSE). Relative error measurements such as MAPE were not employed to prevent conflicts generated by dividing data by zero. It is also important to highlight that minimalism was a constant preoccupation in this feature generation stage since it is desirable to produce an actionable clustering algorithm with some interpretability.

The generated features were then subject to scaling, namely min-max normalization⁴. This step was carried out to ensure that each feature generated has the same impact on distance calculation.

Clustering algorithm

Clusters were produced using K-means and Hierarchical Agglomerative Clustering. The first technique requires *a priori* knowledge of the number of groups to be formed (Lloyd, 1982). This variable is unknown at the time of cluster elaboration, as the purpose is to determine the appropriate time series groupings for the data under examination. An heuristic was then employed to determine this variable, known as the Elbow method (Thorndike, 1953). This technique involves charting the explained variation as a function of the number of clusters and selecting the curve's elbow as the optimal number of clusters. An illustrative example of these charts can be found in figure 4.6.

Hierarchical Agglomerative Clustering allows the user to choose between specifying the desired number of groups to be constructed or a certain distance metric threshold above which clusters do not merge (Müllner, 2011). The algorithm was subject to hyperparameter tuning to better adapt the technique to the data. A randomized gridsearch was applied to determine the optimal number of clusters to form and the best linkage criteria to employ. For the former, values from 4 to 15 were tested. For the latter, the following criteria were applied:

1. **Ward** - reduces the total of squared differences among all clusters.
2. **Maximum or complete linkage** - reduces the maximum distance between cluster observations.

⁴Consists in rescaling the each feature range to the [0, 1] boundaries.

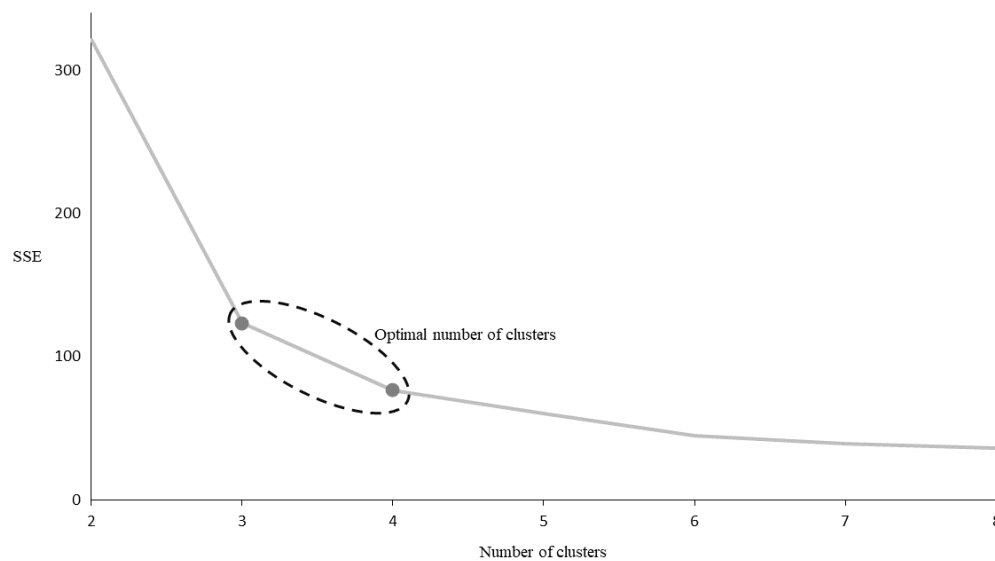


Figure 4.6: Elbow chart example with suitable selections for the optimal number of clusters. This is a frequent heuristic in mathematical optimization to pick a point where the added expenditure isn't worth declining benefits. Increasing the number of groupings will enhance the fit (explain more variance) since there are more parameters (more clusters), but at some point (past the elbow), over-fitting is more prominent.

3. **Average linkage** - reduces the average distance between all pairs of cluster observations.
4. **Single linkage** - reduces the distance between the clusters with the closest observations.

The resulting clusters were then assessed using internal evaluation measures that quantified the clusters' quality without the necessity for class labels. Three metrics were used for the effect:

- The **Davies-Bouldin index** pretends to evaluate similarity, defined as the ratio between within-cluster and inter-cluster distances. Thus, clusters with more separation and less dispersion will rank better. The minimum score is zero, with lower values indicating better clustering.
- The **Silhouette score** is used to evaluate the quality of clusters created in terms of how well samples are clustered with other samples that are similar to each other. The best value for this criteria is 1 and the worst value is -1. Values near 0 indicate overlapping clusters. Negative values generally indicate that a sample has been assigned to the wrong cluster, as a different cluster is more similar.
- The **Variance Ratio Criterion** is the ratio of the total dispersion between clusters and within clusters, where dispersion is defined as the sum of squared distances. The higher the value of this metric, the better the results.

To create the final groups, the clustering technique that achieved the most balanced results was selected.

Chapter 5

Results

The present chapter summarises the results obtained by applying the methodology described in Chapter 4. The section 5.1 initiates the discussion with an overview of the datasets used in the experiment and their business background. The findings of the intermittent and lumpy time series approaches are presented in section 5.2. The last segment, 5.3, analyses the outcomes of the clustering method and the resulting categories.

5.1 Case studies overview

To ensure the robustness of the forecasting algorithms applied and presented in the approach, we used five distinct datasets differing in the amount of time series, observations, frequency, forecast horizon, validation/test split, and business context. This data is summarised in table 5.1. Furthermore, the table in appendix D summarises all the information regarding the ABC, XYZ, and SEIL frameworks per dataset.

Case study I was the foundation for all approaches implemented. This dataset comprises monthly demand values from the beginning of the year 2016 to the end of 2020. The information pertains to a global Business-to-Business (B2B) organization specializing in wine cultivation, production, and distribution. More than ninety percent of this dataset consists of intermittent and

Table 5.1: Datasets characteristics.

	# Irregular Series	Average series length	Frequency	FH ¹	# Folds	# Test Folds	Context
Case Study							
I	6186	38.99	Monthly	3	5	2	B2B
II	81	173.28	Weekly	4	6	2	B2B
III	168	55.43	Monthly	3	6	2	B2B
IV	4106	41.68	Weekly	4	8	3	B2C
V	2211	92.82	Bi-Weekly	4	8	3	B2B

¹Forecast Horizon

lumpy time series used to validate the forecasting methods. Regular series (smooth and erratic) were not considered for the analysis, as in all other datasets utilized.

Case study II is a dataset from a player in the global beverages market, mainly focusing on beer and bottled water, containing the weekly sales volume of products exported. The time series it includes are mostly lumpy and present the highest average values of coefficient of variation and average inter-demand interval of all data utilized, making the series the most challenging to predict accurately. Data comprises the period between the beginning of the year 2018 and March of 2022.

Case study III contains data from a Portuguese company acting in the pulp and paper industry. The data holds the monthly sales volume from the beginning of 2015 to the end of 2021. Unlike the previous case study, less than 50 percent of this dataset are lumpy time series, and these present the lowest average coefficient of variation of all irregular series under analysis.

Case study IV refers to domestic sales of a Portuguese retailer of consumer electronics. The analyzed data amounts to a little more than one year of weekly sales (from June 2020 to August 2021) for a limited subset of the company's items (primarily home appliances and refrigerators). Almost 98 percent of the dataset consists of irregular time series, which may be explained by the slower movement of these products relative to other electronic retail items when seen from a weekly and shop granularity. This stands as the sole analysis of Business-to-Consumer (B2C) information.

The last case study - case study V - shows the demand volume for a firm specializing in the design, development, and production of fruit, vegetable, and cereal-based preparations for the food sector. This demand is collected every two weeks from the beginning of 2016 until the end of 2021. Almost ninety percent of the dataset consists of lumpy and intermittent time series. However, these exhibit less variability on average than the time series comprising the remaining datasets, except for case study III.

The forecast horizon, number of folds, and number of test folds were chosen based on experience in similar use cases, keeping in mind the length and time-frequency of the time series under analysis.

5.2 Forecasting methods results

The present section elaborates on the findings attained from the application of the techniques implemented, followed by an assessment of overall gains resulting from this implementation. The final subsection is dedicated to the special case of forecasting for multiple temporal frequencies with coherent values for all temporal levels.

The forecasting methods applied to obtain a comparison between the actual state and the change brought by the new algorithms correspond to the methods dedicated to irregular series already implemented in the module (SES and Croston's algorithm) with the addition of seasonal Naïve methods with a lag of six and twelve periods, a regular Naïve predictor with lag one, an average of the last three periods, and an ARIMA model fitted for each time series.

5.2.1 Classification and regression

Table 5.2 summarizes the results of the classification model's thorough gridsearch employed on the first fold of each model's input data, followed by cross-validation on a rolling basis. The threshold that was considered for converting demand occurrence probabilities to binary values was the one that maximized the F1 score for both train and test data.

Table 5.2: Average results obtained for the classification model.

Case Study	AUC		F1	
	Train	Test	Train	Test
I	0.9871	0.9846	0.9282	0.9200
II	0.9942	0.9681	0.9865	0.9270
III	0.9757	0.9079	0.9500	0.8306
IV	0.9501	0.9553	0.6905	0.7692
V	0.9528	0.9619	0.8183	0.8442

Regarding the AUC statistic, all created classification models provided encouraging results, having AUC values greater than 90% and similar predictive quality. Case Study III had the highest disparity between train and test outcomes, even though it corresponds to the dataset with less variability in the intermittent and lumpy time series that compose it. It is also the dataset with fewer training data points, which might affect a data-hungry method such as GBM.

The F1 score, on the other hand, demonstrated a substantial variance in the achieved values. This result is due to the extreme disparity between the number of demand periods and non-demand times. Only 6% of the periods in case study IV - which exhibits the poorest outcomes in this regard - are demand occurring times.

The insights gained from these measurements are that despite the model's solid predictive ability, the thresholds selected for the algorithm may result in sub-optimal outcomes. Nonetheless, both procedures will be tested when generating the final forecasts for this technique.

Table 5.3 contains error statistics for the best performing classification job and demand size prediction technique combo in the validation folds. The most effective demand size estimating strategies were producing SES projections from positive data and developing a GBM model using the dataset's characteristics. The GBM model regularly provided outcomes with a wBias value much greater than that of SES predicted projections. The dataset from case study III was the only one where this conclusion was not validated, which may be explained by the limited amount of positive observations in this dataset, precluding the creation of accurate forecasts. All applied approaches yielded similar findings for this dataset.

Demand size multiplication estimates with binary classification proved to be more efficient than the product of demand size estimates with probabilistic classification. Considering wMAPE

and wMASE, the binary values for demand occurrence have consistently exceeded the probabilities. However, it usually came at the expense of incorporating considerable bias into the model. Case study III was the only dataset where this was not verified, as the binary technique was the top performer overall. For the remaining datasets, the best-performing algorithms were selected based on the optimal balance across all measures.

The results achieved in demand size estimation were not as consistent as the categorization model's results. This fact leads to the conclusion that the majority of the variability that contributes to the lumpy nature of the examined time series derives from the significant variation in demand volume rather than the timing of demand occurrences. The findings are comparable with those of Rožanec and Mladenčić (2021), which also found SES as one of the most effective methods for anticipating the volume of demand.

Table 5.3: validation demand size prediction error metrics.

Case Study	Gap	DSEM ²	Class. Type	wMAPE	wMASE	wBias
I	2	GBM	binary	0.6316	0.2638	-0.3988
	3	SES	binary	0.7714	0.4947	0.0857
	4	SES	probabilistic	0.8328	0.5905	0.0625
II	1	SES	binary	0.4874	0.4778	-0.1665
	2	SES	binary	0.7165	0.4640	0.1081
	3	GBM	binary	0.6396	0.2572	0.0095
	4	GBM	probabilistic	1.3506	0.5361	0.1847
III	1	GBM	binary	0.5063	0.7340	-0.2612
	2	GBM	binary	0.8204	1.3094	-0.1952
	3	GBM	binary	0.9828	1.7433	-0.3225
IV	2	SES	binary	0.5266	0.3997	0.1379
	3	SES	binary	0.8219	0.5821	0.0107
	4	SES	probabilistic	1.2244	0.6545	0.1317
	5	SES	binary	1.1448	0.5794	0.2110
V	2	SES	binary	0.4903	0.4486	-0.1144
	3	SES	binary	0.7200	0.6758	-0.0797
	4	SES	binary	0.7364	0.6817	-0.1661
	5	SES	binary	1.2509	0.8937	0.2167

²Demand size estimation method.

Variable importance assessment

Machine learning models such as GBM are often referred to as “Black Box” models - algorithms whose mathematical behavior is too complex for direct interpretation of the influence of specific input variables on the output. Decision-makers in S&OP are frequently reluctant to trust the forecasts generated by these models since they are not straightforward to understand. As such, in an attempt to add interpretability to the results obtained by the classification model, each variable’s weight in the final solution was assessed.

The aforementioned analysis is displayed in figure 5.1, in which only variables with a relative score of 2 percent or above are shown. The research revealed that the amount of time since the previous incidence of demand predominated over all other factors. The point in the forecast horizon for which forecasts are produced was also a regular top performer. The supremacy of these variables may be attributable to the fact that items exhibiting lumpy behavior could be obsolete, highly seasonal, or have relatively few instances of demand in the series. The only exception was case study I, which presents highly seasonal lumpy patterns, leading to the high relative importance of the seasonal naïve binary prediction of demand occurrence.

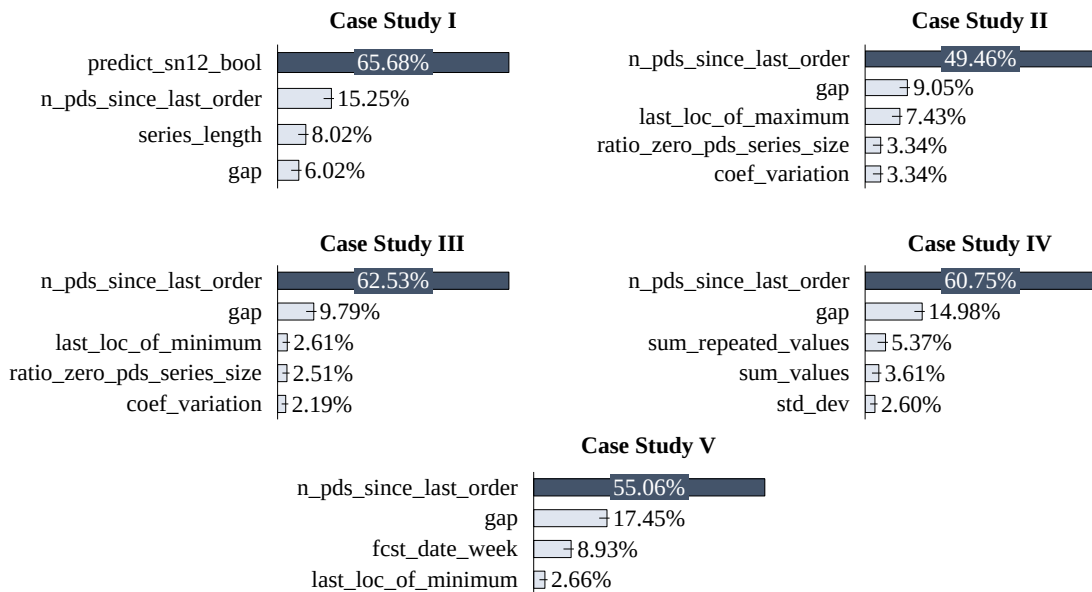


Figure 5.1: Relative variable importance of features for the classification model.

5.2.2 Temporal aggregation results

The algorithms for temporal aggregation (ADIDA and MAPA) were examined utilizing aggregation levels ranging from two to ten time buckets in a rolling fashion. The upper bound of this range is equivalent to the highest median value of time series average inter-demand interval of all

datasets. This range constitutes a good balance between intermittency mitigation and computational effort. Figure 5.2 contains the prediction error values obtained in the test folds corresponding to the best performing ADIDA and MAPA configurations for each dataset tested, along with the same values obtained from the combination of the best results from existing techniques across the different points in the forecast horizon. These configurations and combinations were chosen based on the performance verified in the validation folds.

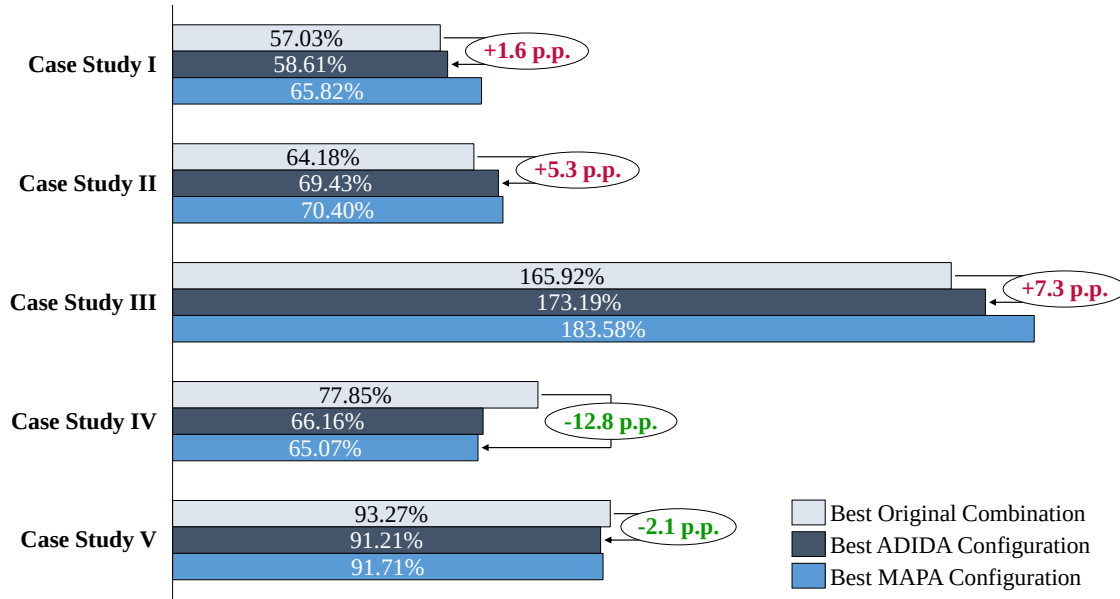


Figure 5.2: wMASE values for ADIDA, MAPA and the best currently implemented (original) method.

The original approaches performed better in case studies I, II, and III than the optimal configuration of ADIDA or MAPA. The ideal aggregation level for the first dataset was determined to be two periods, which is the minimum value. This result might be due to the fact that the applied disaggregation approach (an equal-weighted average) may not be the most appropriate for the datasets. Study of more complex disaggregation strategies goes beyond the scope of this document. It may also originate from the fact that seasonality or trend characteristics are better recorded at the original frequency than at a more coarse granularity. This result is not unforeseen given the monthly periodicity of the series. For the second dataset ADIDA demonstrated the best results for the more distanced periods in the forecast horizon. Despite this, the original method performed better in the first periods, resulting in a rise in the forecast error brought about by the aggregation technique. As for case study III, the best existing methods outperformed the temporal aggregation strategies considerably. In the remaining datasets, ADIDA or MAPA were the best performers for all points in the forecast horizon, which indicates that some aspects of seasonality, trend, or level were better captured at more coarse granularities.

By shifting the scope of the investigation from the overall best configurations to the outcomes delivered by this technique, method by method, it was possible to demonstrate that temporal ag-

gregation significantly improved forecasting results for specific series. Mean or median statistical aggregation of disaggregated forecasts produced extremely similar accuracy outcomes for the MAPA algorithm's estimations of forecast accuracy. Differences between the two were minimal for every method used to generate predictions at different time frequencies - less than 2% on average regarding wMAPE, wMASE, and wBias values. Consequently, this algorithm's primary source of distinction is its underlying prediction approach. These are similar findings to those of Kourntzes et al. (2014). For detailed results of all temporal aggregation procedures, please refer to appendix E.

5.2.3 Overall forecast performance assessment

Similar to the previous procedure, algorithms were selected for each point in the forecast horizon based on their performance in the validation folds. These results are displayed in table 5.4, along with the values corresponding to the forecast error metrics.

Table 5.4: Best performing algorithms in the validation folds.

Case Study	Gap	Method	wMAPE	wMASE	wBias
I	2	Seasonal Naïve(12)	0.6510	0.2954	0.1255
	3	Seasonal Naïve(12)	0.7530	0.4875	0.0249
	4	Seasonal Naïve(12)	0.8130	0.5765	0.0966
II	1	SES Binary	0.4874	0.4778	-0.1665
	2	SES Binary	0.7165	0.4640	0.1081
	3	GBM Binary	0.6396	0.2572	0.0095
	4	TSB	1.0564	0.4440	-0.1436
III	1	GBM Binary	0.5063	0.7340	-0.2612
	2	GBM Binary	0.8204	1.3094	-0.1952
	3	Naïve Lag 1	0.8646	1.6022	-0.2922
IV	2	SES Binary	0.5266	0.3997	0.1379
	3	SES Binary	0.8219	0.5821	0.0107
	4	MAPA(Seas. Naïve(6), Median)	1.0863	0.5795	-0.2082
	5	ADIDA(5, Naïve Lag 1, Mean)	1.0672	0.5494	-0.3836
V	2	SES Binary	0.4903	0.4486	-0.1144
	3	SES Binary	0.7200	0.6758	-0.0797
	4	SES Binary	0.7364	0.6817	-0.1661
	5	ADIDA(9, Seas. Naïve(6), Mean)	1.3174	0.8795	0.1892

Case study I is the only dataset in which usage of newly introduced strategies did not yield any improvement. Again, this may be owing to the high yearly seasonality of the sales time series

that comprise it, which makes the seasonal naïve technique the best performer. For the remaining datasets, the approaches employed improved 14 of 18 prediction moments. The most effective strategy among those implemented was the one that handles categorization and demand size estimation as separate issues, achieving the highest performance for ten out of eighteen periods. Appendix E includes further information regarding the comparisons between approaches for the various datasets.

Figure 5.3 illustrates the improvement in forecast accuracy brought about by the implementation of the proposed methodologies. As before, unbiased evaluation metrics were obtained by evaluating the findings solely on the test folds.

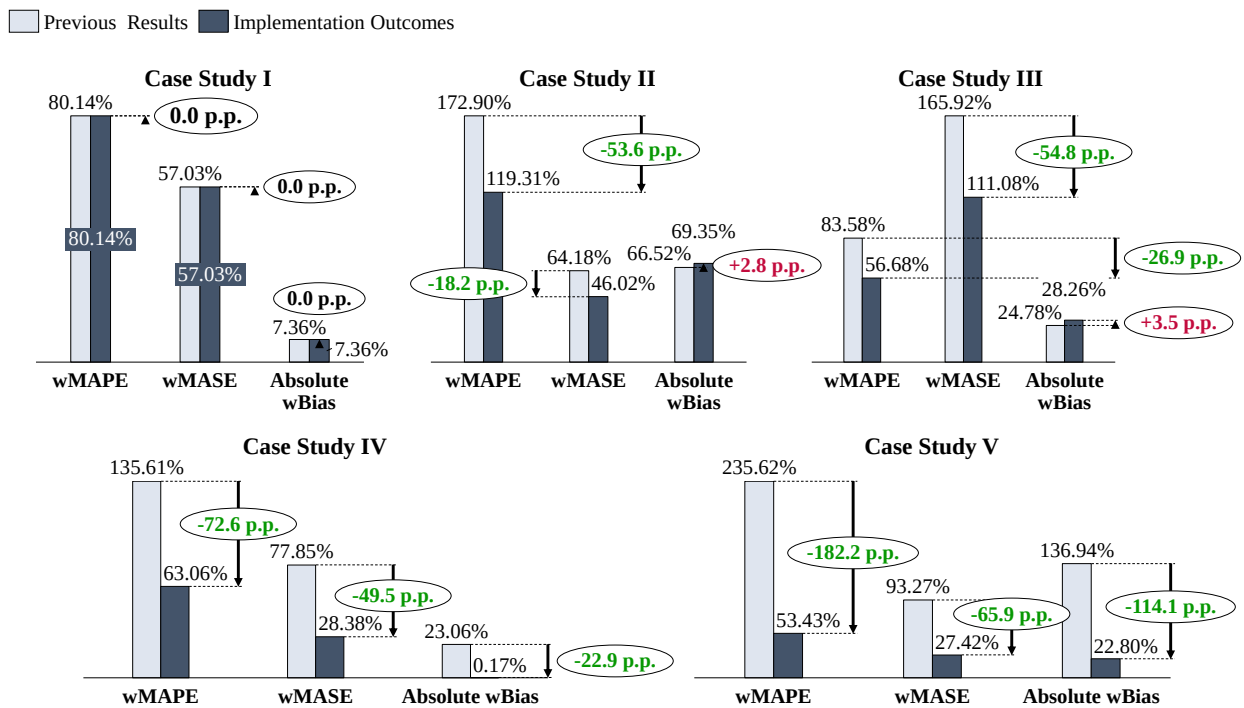


Figure 5.3: Overall outcomes from the proposed implementation evaluated on the test folds.

The results demonstrate significant decreases in forecast error for most datasets. For case studies II and III, the cost of this implementation was an increase in the total wBias. However, depending on the importance given to each metric, balance can still be positive for both. Case study V was the dataset where the most dramatic performance enhancements were verified. The strategy that combines the GBM classification model with the SES projections of demand size reduced wMAPE by 182.2%, wMASE by 65.9%, and wBias by 114.1% on average when compared with the second-best procedure for the test folds. This fact might be the outcome between the interplay of two factors:

1. The test folds of this dataset correspond to periods of substantially declining sales volume, whereas the validation folds demonstrate a rather steady total demand;

2. The classification GBM trains with data from previous evaluation folds in an expanding window manner until the final fold. As a result, the model is always up to date with the data from the preceding folds, enabling it to adjust to the shift in sales level.

5.2.4 Hierarchical temporal reconciliation

Forecast coherence supports cohesive decision-making across multiple planning horizons, preventing distinct decision-making units from planning on different future views. To evaluate the results brought by the implementation of a forecast reconciliation tool across different temporal frequencies, subsets of the aforementioned case study datasets were generated for the larger case studies. A total of 500 time series were randomly selected and the best performing method at the finer granularity level was used to generate reconciled forecasts. Projections were then evaluated at three different time buckets size: for case studies I and III, forecasts were evaluated at one, two, and three months time buckets; for the remaining datasets evaluation was conducted at one, two, and four temporal aggregation levels. The overall results were then assessed by averaging the errors obtained for each time period considered.

To create a benchmark of comparison for this method, a simple bottom-up strategy was implemented. In this strategy, forecasts generated at the bottom level of the hierarchy are summed up to form the forecasts generated for the levels above. Table 5.5 summarises these results, providing a global measure of improvement calculated by averaging the differences in the metrics between both techniques.

Table 5.5: Overall results of the multiple planning horizon techniques employed.

Case Study	Method	wMAPE	wMASE	wBias
I	Cross-temporal reconciliation	0.7220	0.8244	0.3010
	Bottom-up	0.7884	0.9064	0.3111
II	Cross-temporal reconciliation	0.8445	1.1221	0.0231
	Bottom-up	0.8598	1.1639	-0.1967
III	Cross-temporal reconciliation	0.6978	1.7671	-0.3257
	Bottom-up	0.6432	1.6160	-0.2685
IV	Cross-temporal reconciliation	1.3561	0.7710	0.3556
	Bottom-up	1.5571	0.8892	0.5361
V	Cross-temporal reconciliation	0.7480	1.6023	-0.2147
	Bottom-up	0.7089	1.5159	-0.2068
Total Average Improvement		0.0378	0.0009	0.0599

Introducing cross-temporal reconciliation proved to be a fruitful endeavor, as all of the analyzed measures demonstrated an overall improvement. For three out of five datasets, the algorithm was the best performer across the board compared to the bottom-up strategy. However, given that

for the remaining datasets, the algorithm showed worst performance than the same strategy in all metrics evaluated, it emerges as a “hit-or-miss” technique. In addition, among the datasets in which the algorithm increased global prediction accuracy, wBias was the measure that improved the most, indicating that the corrective technique for this metric, described in subsection 4.2.2, served its intended purpose.

5.3 Clustering and categorization findings

The intermittent and lumpy time series forecasting methods implemented required diverse applications to assess their robustness. On the other hand, time series clustering is done here with the intention of grouping the series constituting a dataset, enabling efficient allocation of time, effort, and attention of decision-makers, among other benefits (recall figure 1.1). Consequently, it is unproductive to compare clusters across unrelated data, as each case will bestow specific recommendations. Thus, the procedure followed was evaluated solely using the data from Case study I.

Time series filtering and K-Means algorithm tuning

A portion of the input data for clustering included missing values resultant of the time series feature engineering step outlined in section 4.3. Time series with very low longevity (less historical records than the number of chosen folds, present in table 5.1) were excluded from the analysis as no robust predictions nor error metrics could be generated.

As stated in chapter 4, the K-Means algorithm requires a heuristic to be applied to determine the optimal number of clusters for the data being fitted. The results of this approach, showcased in figure 5.4, indicated that nine could be an adequate number of clusters.

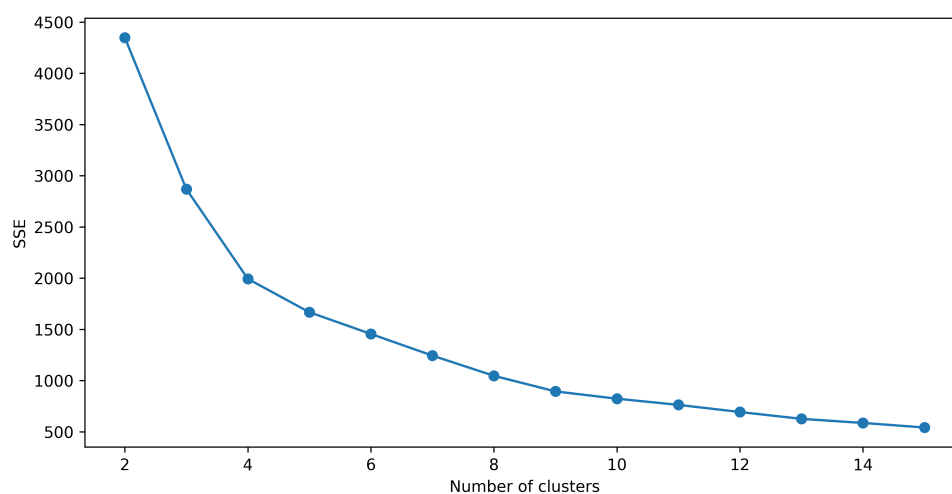


Figure 5.4: Elbow chart obtained for the K-Means algorithm.

Clustering algorithm selection

The clusters formed by the different algorithmic setups devised were subject to evaluation and comparison to determine the best groupings achieved. Before doing so, however, it is vital to identify the properties of the optimal outcome of the clustering process for the given circumstance. The main aspects that constitute good clustering in this instance are:

- Series presenting similar forecast error metrics results and historical observations statistical distribution should be grouped together;
- Each group should be representative of a class of time series and be able to be placed within a quadrant of the decision-making matrix;
- The clusters formed should present no overlapping, be condensed, and be considerably distanced from the nearest clusters.

In order to make this last quality assessment, table 5.6 condenses the comparisons over the four metrics used for groupings evaluation described in section 4.3 for the ten best-performing algorithms regarding the sum of the MSE across all points in the forecast horizon, from which we will select the best algorithm.

Table 5.6: Best performing methods regarding MSE.

Method	Davies-Bouldin Index	Silhouette Score	VRC	MSE ($\times 10^6$)
AC_4_ward	1.0060	0.5203	5309.81	1.0737
AC_5_ward	0.9468	0.5438	5333.48	1.4435
AC_6_ward	0.9034	0.4862	5075.80	1.4437
AC_7_ward	0.8803	0.4474	5064.64	1.4464
K-Means_9	0.8596	0.4850	5786.49	1.7469
AC_8_ward	0.9045	0.4613	5204.99	2.3158
AC_9_ward	0.8686	0.4656	5383.74	2.3172
AC_4_single	0.3710	0.4499	8.41	2.5719
AC_5_single	0.5568	0.2525	17.80	2.5722
AC_6_single	0.5742	0.3961	909.19	2.6242

The agglomerative clustering algorithm with the “ward” linkage criterion forming four clusters (referenced in table 5.6 as “AC_4_ward”) was the best performer according to the aforementioned metric. However, the same algorithm also presents the lowest performance for the Davies-Bouldin index out of the ten under comparison. These results imply that the computed groups have a high ratio of within-cluster to between-cluster distances among the most similar clusters, indicating that

the groups exhibit considerable internal dispersion, overlap, or both. The Silhouette score yields more positive clustering outcomes, rating the approach second out of the ten best with a satisfying value for this parameter. The technique also presents good results for the Variance Ratio Criterion (VRC), which indicates that clusters present low variance relative to between-cluster variance, in accordance to the results provided by the Silhouette score.

Adding table 5.7 to the analysis provides a clearer vision of the series distribution across clusters. Upon examining the groups generated using the “single” linkage criterion, we can determine that the splits formed are not representative of any class since many clusters include only a single time series and the remaining concentrate on a single group. This undesired outcome is mostly observed in the top performers according to the Davies-Bouldin index score, indicating that this metric should not be regarded as the defining indicator for success in solving the problem under analysis. The Silhouette score and Variance Ratio Criterion ranked these series as worst performers, with the latter providing a large gap between them and the remaining (more representative) clusters, and thus seeming a better fit to guide the selection of the algorithm to be employed.

Table 5.7: Number of time series per cluster for each best-performing methods.

Cluster	0	1	2	3	4	5	6	7	8
Method									
AC_4_ward	2245	2185	1184	904	-	-	-	-	-
AC_5_ward	2245	2185	1184	497	407	-	-	-	-
AC_6_ward	2185	1597	1184	648	497	407	-	-	-
AC_7_ward	1597	1262	1184	923	648	497	407	-	-
K-Means_9	1232	1106	1081	1013	694	504	435	270	183
AC_8_ward	1597	1262	1184	923	648	407	301	196	-
AC_9_ward	1597	1262	923	648	644	540	407	301	196
AC_4_single	6515	1	1	1	-	-	-	-	-
AC_5_single	6493	22	1	1	1	-	-	-	-
AC_6_single	4248	2245	22	1	1	1	-	-	-

Analyzing the overall outcomes, the method for agglomerative clustering with five clusters and the “ward” linkage condition appears to be the most well-rounded. This method yields the best results regarding the Silhouette score. Furthermore, it ranks second for the mean squared error of predictions provided by the best forecast method per cluster and third concerning the Variance Ratio Criterion. In addition, it improves the Davies-Bouldin index over the similar approach that produced four groups. Consequently, the strategy will be applied to the ensuing clustering research.

Classification scheme assessment

Table 5.8 summarizes the average forecast error metrics for the selected algorithm and the respective sum of demand volume for all time series within each group. Cluster 1 analysis reveals series classified as regular under the XYZ framework presenting little variability for which all forecast methods employed demonstrated good performance, with classical methods outperforming the methods dedicated to intermittent and lumpy time series in all aspects. Cluster 4, on the other hand, is composed of items presenting lumpy demand and larger variability. Traditional forecasting methods still outperformed those tailored to irregular series. However, the results were much less unbalanced. The techniques all showed a tendency to predict below the actual value. Clusters 0 and 2 represent time series with high variability and an almost unpredictable pattern, confirmed by the poor results obtained by the forecasting exercise. Predictions generated were widely out of target for the series constituting cluster 0. For both groups, the best performing method was a simple naïve of lag one, which, coupled with the high variability, demonstrates that the series resemble random walks. Cluster 3 is composed of lumpy time series with medium variability in occurring demand. The methods specially developed to handle this type of series present the minimum wBias among all applied, which indicates that more sophisticated intermittent and lumpy forecasting could yield satisfying results for this group. A table containing the distribution of time series per ABC, XYZ, and SEIL class for each cluster is present in appendix F, along with a list of the best performing methods and their outcomes per group for every point in the forecast horizon.

Table 5.8: Forecast accuracy measures, total and relative demand volume and time series quantity per cluster.

Category	wMAPE	wMASE	wBias	Volume	Volume %	# Time Series	Time Series %
0	12.4703	1.2298	10.7276	1182	0.17	2245	34.44
1	0.5666	0.1439	0.1222	539206	78.18	497	7.63
2	2.2537	0.7578	0.6339	11301	1.64	2185	33.52
3	1.3016	0.8778	0.0387	21057	3.05	1184	18.17
4	1.0543	0.7330	-0.0877	116956	16.96	407	6.24

Figure 5.5 illustrates the allocation of the generated groups to the quadrants of the S&OP introduced in chapter 1. Having conducted a deep analysis on the clusters generated, we can now assemble a list regarding cluster criticality according to forecastability and relevance for total demand. As such, the descending order of priorities is as follows:

1. **Cluster 1.** This category comprises the most important series to be studied. Given the high quality of the automatically generated predictions, these should serve as a solid foundation for S&OP activities. Nonetheless, they should be actively monitored, with the majority

of the forecasting and validation work allocated to the series in this category, to allow for a prompt intervention by the decision-maker should the forecasts fall below the intended accuracy level.

2. **Cluster 4.** This cluster has high values for prediction error metrics, with a wMAPE score above one and a wMASE value comparable to the lowest-performing groups. Additionally, its elements account for roughly seventeen percent of overall demand, an amount too significant to forecast incorrectly. In a more prescriptive tone, human validation of the automatically generated projections would be advised for this cluster, with specific care given to the time series with the highest volatility.
3. **Cluster 3.** External factors may be able to explain the series variation due to the lumpy but not too irregular character of the series comprising this cluster, which encourages the use of more advanced prediction algorithms for improved forecasting outcomes. As the group has little impact on demand, the decision-maker should not dedicate an extensive effort to these time series unless they correspond to goods with high storage costs or a crucial service level.
4. **Clusters 2 and 0.** Due to the low weight in demand volume and extremely irregular structure of these clusters, the work required to analyze their components should be kept to a minimum. It can be cumbersome to generate reliable projections for the data comprising cluster 0 since the series are essentially random walks. Cluster 2 should only be considered if there is significant value to be added to the business to justify the high effort required or if there is sufficient attention budget after analyzing the preceding clusters.

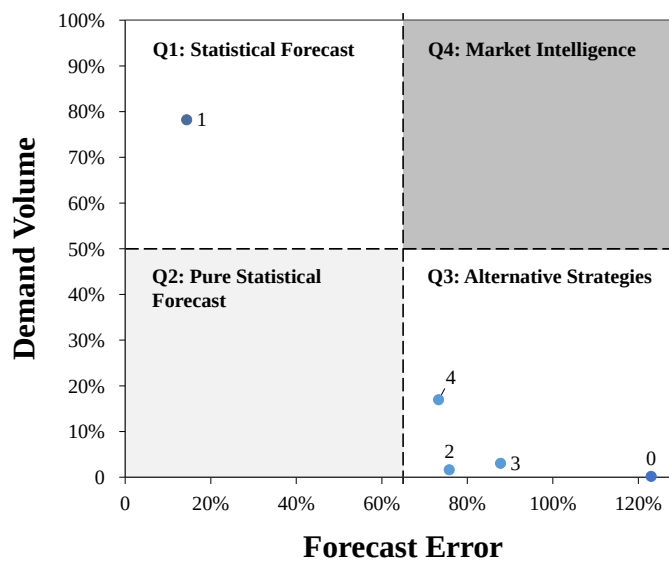


Figure 5.5: Distribution of the clusters generated throughout the S&OP matrix.

Chapter 6

Discussion and conclusions

This concluding chapter outlines the key findings and offers recommendations for further research in the area. Section 6.1 elaborates on the acquired results, compares them to scientific literature, and proposes a set of short-term enhancements to the employed algorithms that could add value to the provided forecasts. This dissertation concludes with section 6.2, which provides high-level suggestions to improve the information communicated by the computed projections.

6.1 Practical implications

This dissertation contributes to the study of forecasting methods suitable for intermittent and lumpy time series by presenting a rigorous empirical application of these approaches on five separate datasets from real-world enterprises operating in diverse sectors. In addition, modifications were made to some of these approaches, allowing for their application in previously unexplored settings. This work also attempts to bridge the gap between automatically generated and judgmental forecasts by introducing a framework that directly assesses time series forecastability and enables specific recommendations regarding when and how to act upon automated projections, given the limited time budget decision-makers typically possess. The following paragraphs exhibit a segmented analysis of the implications of the work developed.

Forecasting methods

As shown in chapter 5, the use of cutting-edge forecasting approaches for irregular series in the current project led to considerable improvements over the previously existing solution. Outcomes derived from the experiment conducted demonstrate that the performance of the approaches varies significantly amongst different datasets. Thus, there is no simple response to the issue of when or why a certain strategy for forecasting irregular demand should be adopted.

Nevertheless, the strategy of dividing the forecasting problem into two distinct issues - positive value occurrence and quantity projections - and employing contemporary machine learning algorithms for the effect yielded the best results in the majority of examined predictions. Reviewing the outcomes obtained by the approach, it can be concluded that its primary advantage over the

other evaluated methods is its higher accuracy in predicting the periods of occurrence of positive values. In some instances, however, the model injects a substantial negative bias to predictions, which leads to recurrent demand underestimation. Since only a global threshold that reduced the overall F1 score for the classification job was chosen, this decline in accuracy is comprehensible. Optimizing this value for each time series might enhance the effectiveness of this technique. A bias correction method may also have the ability to improve the outcomes. Furthermore, accuracy may be additionally improved by incorporating relevant external factors that might explain the observed values in both classification and regression models. As these practices go beyond the scope of this thesis, we leave this note for future work.

Results also demonstrate that simple strategies such as temporal aggregation or more straightforward modifications to the Croston method may be just as successful as complex mathematical techniques for intermittent or lumpy forecasting. TSB, one of these adaptations, for instance, was the best performing method in the test folds of the dataset of case study II. Proper identification of the method's potential in the validation folds could have improved the forecasting exercise considerably, as table E.4 demonstrates. This method performed better than Croston and SBA regarding wMAPE and wMASE for all datasets. Regarding wBias, however, the performance of the latter techniques - which was essentially equivalent across all datasets - was superior for case studies II and III. All methods were employed using different smoothing parameter values for each projection. The enhanced performance of Croston and SBA with regard to wBias for two datasets may be attributed to the infringement of the assumption of stationary demand, as the assessed moments display high positive value fluctuation. This finding suggests that TSB should be the preferred approach for datasets under low demand variability regimes with high values of average inter-demand interval, since it updates the probability of demand every period in detriment of updating it only after occurring.

An adaptation to the ADIDA and MAPA algorithms was conducted, shifting the aggregation procedure from discrete time buckets to overlapping rolling windows. This implementation yielded similar results to those found by the authors, constituting a self-improvement mechanism for all metrics assessed when temporal aggregation proved beneficial for the time series. This reality demonstrates that the method is entirely adaptable to this new context, expanding its range of applications to include short span time series that could not be employed before due to the original technique's data reduction behavior while aggregating data. Although the results were satisfying, some interesting expansions could be beneficial in improving the accuracy of these methods. Individually, ADIDA could benefit from the introduction of more sophisticated disaggregation methods that account for aspects such as trend, seasonality, or cycle of the underlying time series. MAPA may additionally gain from a careful selection of disaggregate predictions used to produce the final forecasts.

Temporal aggregation was also applied in a paradigm of temporal hierarchies and their reconciliation, shifting the focus from improving the forecast accuracy at a single time-frequency to multiple ones. The proposed method demonstrated promising results with overall improvements in all metrics evaluated, outperforming the simple bottom-up summation of forecasts. For some

datasets, the algorithm improved the bottom-level ex-ante forecasts. However, the major setback of this tool is the heavy computational power required to apply it, as the intermediate processes of generating a summation matrix and conducting an error minimization algorithm is very memory intensive.

Time series categorization

An attempt at connecting automatic base projections to human judgment was conducted. Time series were categorized by applying a clustering technique using several features regarding series forecastability and demand shape as input. The clusters formed were able to aggregate time series by differing levels of criticality as intended originally. This outcome enables specific targeting of the attention of the decision-makers and efficient use of their limited time. In addition, predictive accuracy was enhanced by selecting the optimal forecasting technique for each period within the forecast horizon and cluster, as table F.3 indicates. Comparing this deeper forecast segmentation strategy and other already established frameworks reveals the method presents well-rounded results. Nonetheless, more empirical applications are required to prove its efficiency.

6.2 Future research

Despite the accuracy improvement of the company's forecasts for intermittent and lumpy time series, the work developed still presents its limits and potential. Several modifications that might add value to the suggested implementations were mentioned in the previous section, but we now shift the focus to more conceptual recommendations.

Regarding the proposed dual approach to forecasting of irregular series, only the GBM algorithm was tested. The choice for this algorithm settled on the results originated from the M5 competition, in which this decision tree-based ML approach was used in practice by the top 50 solutions (Makridakis et al., 2021). Although it provided good results, further breakthroughs in the study of machine learning, deep learning, and other AI-based techniques, along with an increase in processing power, should lead to the development of more accurate predictors. Moreover, more complex forecasting ensemble techniques could be explored as an alternative to generating predictions for the demand size estimation, as they have proven their value before, or even combine the forecasts generated by this method with other approaches.

Using temporal aggregation and reconciliation, operational units within a corporation may be aligned on the same basis, ensuring everyone is working under the same forecast knowledge. The alignment of the company's decision-making process can be strengthened by simultaneously exploring temporal and cross-sectional hierarchies.

In addition, only predictive accuracy was evaluated in this study. Methods should also be empirically verified and assessed using performance measures specific to the decision process they are feeding. Forecasting and inventory management emerges as an example of this statement. The

two paradigms have demonstrated discrepancies in prior research across several areas, despite being frequently intertwined - bridging the gap between the two would benefit operations, smoothing the transformation of projections into actions.

As a concluding remark, forecasting has positioned itself as a crucial decision-making instrument that sets the pace for the demand planning process and other sales and operations management tasks. Given the limited forecasting ability of most businesses, organizations that intend to stay competitive should prioritize utilizing this tool in the most effective manner possible by combining analytical and human judgment. The attempted method for analyzing time series criticality in terms of predictability and contextual value in an automated form was beneficial as it allowed for prescriptive attention segmentation. However, the strategy still requires intermediate judgments, namely in the number of groups formed and the quality of sectioning, which may compromise its use as a general purpose framework. Further research on a more automated and uniform framework for time series criticality evaluation might enhance the procedure.

Bibliography

- Aghabozorgi, S., Shirkhorshidi, A. S., and Wah, T. Y. (2015). Time-series clustering – a decade review. *Information Systems*, 53:16–38. <https://doi.org/10.1016/j.is.2015.04.007>.
- Athanasopoulos, G., Hyndman, R. J., Kourentzes, N., and Petropoulos, F. (2017). Forecasting with temporal hierarchies. *European Journal of Operational Research*, 262(1):60–74. <https://doi.org/10.1016/j.ejor.2017.02.046>.
- Babai, M. Z., Ali, M. M., and Nikolopoulos, K. (2012). Impact of temporal aggregation on stock control performance of intermittent demand estimators: Empirical analysis. *Omega*, 40:713–721. <https://doi.org/10.1016/j.omega.2011.09.004>.
- Babai, M. Z., Syntetos, A., and Teunter, R. (2014). Intermittent demand forecasting: An empirical study on accuracy and the risk of obsolescence. *International Journal of Production Economics*, 157:212–219. <https://doi.org/10.1016/j.ijpe.2014.08.019>.
- Barker, J. (2020). Machine learning in m4: What makes a good unstructured model? *International Journal of Forecasting*, 36:150–155. <https://doi.org/10.1016/j.ijforecast.2019.06.001>.
- Berry, M. W., Mohamed, A., and Yap, B. W. (2020). *Supervised and Unsupervised Learning for Data Science*. Springer International Publishing. <https://doi.org/10.1007/978-3-030-22475-2>.
- Bower, P. (2017). Is forecastability a real thing?. *Journal of Business Forecasting*, 36(3):26 – 32. <https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,shib&db=bsu&AN=128147095&lang=pt-pt&site=ehost-live&scope=site>.
- Boylan, J. E. and Babai, M. Z. (2016). On the performance of overlapping and non-overlapping temporal demand aggregation approaches. *International Journal of Production Economics*, 181:136–144. <https://doi.org/10.1016/j.ijpe.2016.04.003>.
- Boylan, J. J. E. and Syntetos, A. A. (2021). *Intermittent demand forecasting : context, methods and applications*. John Wiley & Sons, first edition.
- Citroen, C. L. (2011). The role of information in strategic decision-making. *International Journal of Information Management*, 31(6):493–501. <https://doi.org/10.1016/j.ijinfomgt.2011.02.005>.
- Croston, J. D. (1972). Forecasting and stock control for intermittent demands. *Journal of the Operational Research Society*, 23:289–303. <https://doi.org/10.1057/jors.1972.50>.
- Davies, O. L. and Brown, R. G. (1960). Statistical forecasting for inventory control. *Journal of the Royal Statistical Society. Series A (General)*, 123:348. <https://doi.org/10.2307/2342487>.
- Dietterich, T. G. (2000). Ensemble methods in machine learning. In: Multiple Classifier Systems. MCS 2000. Lecture Notes in Computer Science, vol 1857. Springer, Berlin, Heidelberg. 1-15. https://doi.org/10.1007/3-540-45014-9_1.

- Eaves, A. H. C. and Kingsman, B. G. (2004). Forecasting for the ordering and stock-holding of spare parts. *Journal of the Operational Research Society*, 55:431–437. <https://doi.org/10.1057/palgrave.jors.2601697>.
- Fildes, R. and Kumar, V. (2002). Telecommunications demand forecasting—a review. *International Journal of Forecasting*, 18(4):489–522. [https://doi.org/10.1016/S0169-2070\(02\)00064-X](https://doi.org/10.1016/S0169-2070(02)00064-X).
- Fradique, V. (2020). Developing a forecasting analysis framework to fast track analytical developments. Master’s Thesis, Faculdade de Engenharia da Universidade do Porto.
- Hyndman, R. and Athanasopoulos, G. (2021). *Forecasting: Principles and Practice*. OTexts: Melbourne, Australia, 3rd edition.
- Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688. <https://doi.org/10.1016/j.ijforecast.2006.03.001>.
- Johnston, F. R. and Boylan, J. E. (1996). Forecasting for items with intermittent demand. *Journal of the Operational Research Society*, 47:113–121. <https://doi.org/10.1057/jors.1996.10>.
- Johnston, F. R., Boylan, J. E., and Shale, E. A. (2003). An examination of the size of orders from customers, their characterisation and the implications for inventory control of slow moving items. *Journal of the Operational Research Society*, 54(8):833–837. <https://doi.org/10.1057/palgrave.jors.2601586>.
- Kotsakos, D., Trajcevskic, G., Gunopulos, D., and Aggarwal, C. C. (2014). Time-series data clustering. In Charu C. Aggarwal, C. K. R., editor, *Data Clustering: Algorithms and Applications*, chapter 15, pages 358–376. Chapman and Hall/CRC. <https://doi.org/10.1201/9781315373515>.
- Kourentzes, N. and Petropoulos, F. (2016). Forecasting with multivariate temporal aggregation: The case of promotional modelling. *International Journal of Production Economics*, 181:145–153. <https://doi.org/10.1016/j.ijpe.2015.09.011>.
- Kourentzes, N., Petropoulos, F., and Trapero, J. R. (2014). Improving forecasting by estimating time series structural components across multiple frequencies. *International Journal of Forecasting*, 30(2):291–302. <https://doi.org/10.1016/j.ijforecast.2013.09.006>.
- Li, L., Kang, Y., Petropoulos, F., and Li, F. (2022). Feature-based intermittent demand forecast combinations: bias, accuracy and inventory implications. 10.48550/ARXIV.2204.08283.
- Ling, C. X., Huang, J., and Zhang, H. (2003). Auc: A statistically consistent and more discriminating measure than accuracy. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence, IJCAI’03*, page 519–524, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. <https://doi.org/10.5555/1630659>.
- Liu, D. C. and Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528. <https://doi.org/10.1007/BF01589116>.
- Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137. <https://doi.org/10.1109/TIT.1982.1056489>.
- Ma, S. and Fildes, R. (2021). Retail sales forecasting with meta-learning. *European Journal of Operational Research*, 288:111–128. <https://doi.org/10.1016/j.ejor.2020.05.038>.

- Makridakis, S. and Hibon, M. (2000). The m3-competition: results, conclusions and implications. *International Journal of Forecasting*, 16:451–476. [https://doi.org/10.1016/S0169-2070\(00\)00057-1](https://doi.org/10.1016/S0169-2070(00)00057-1).
- Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. *PLOS ONE*, 13:e0194889. <https://doi.org/10.1371/journal.pone.0194889>.
- Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2021). The m5 competition: Background, organization, and implementation. *International Journal of Forecasting*. <https://doi.org/10.1016/j.ijforecast.2021.07.007>.
- Müllner, D. (2011). Modern hierarchical, agglomerative clustering algorithms. <https://arxiv.org/abs/1109.2378>.
- Nenni, M. E., Giustiniano, L., and Pirolo, L. (2013). Demand forecasting in the fashion industry: A review. *International Journal of Engineering Business Management*, 5:37. <https://doi.org/10.5772/56840>.
- Nikolopoulos, K. (2021). We need to talk about intermittent demand forecasting. *European Journal of Operational Research*, 291(2):549–559. <https://doi.org/10.1016/j.ejor.2019.12.046>.
- Nikolopoulos, K., Syntetos, A. A., Boylan, J. E., Petropoulos, F., and Assimakopoulos, V. (2011). An aggregate–disaggregate intermittent demand approach (adida) to forecasting: an empirical proposition and analysis. *Journal of the Operational Research Society*, 62:544–554. <https://doi.org/10.1057/jors.2010.32>.
- Notton, G. and Voyant, C. (2018). Chapter 3 - forecasting of intermittent solar energy resource. In Yahyaoui, I., editor, *Advances in Renewable Energies and Power Technologies*, pages 77–114. Elsevier. <https://doi.org/10.1016/B978-0-12-812959-3.00003-4>.
- Petropoulos, F., Makridakis, S., Assimakopoulos, V., and Nikolopoulos, K. (2014). ‘horses for courses’ in demand forecasting. *European Journal of Operational Research*, 237:152–163. <https://doi.org/10.1016/j.ejor.2014.02.036>.
- Rožanec, J. M. and Mladenović, D. (2021). Reframing demand forecasting: a two-fold approach for lumpy and intermittent demand. <https://doi.org/10.48550/ARXIV.2103.13812>.
- Sanders, R. (1987). The pareto principle: Its use and abuse. *Journal of Services Marketing*, 1:37–40. <https://doi.org/10.1108/eb024706>.
- Sbrana, G. (2022). Modelling intermittent time series and forecasting covid-19 spread in the usa. *Journal of the Operational Research Society*, 0(0):1–11. <https://doi.org/10.1080/01605682.2022.2055499>.
- Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, 5:197–227. <https://doi.org/10.1007/BF00116037>.
- Scholz-Reiter, B., Heger, J., Meinecke, C., and Bergmann, J. (2012). Integration of demand forecasts in abc-xyz analysis: practical investigation at an industrial company. *International Journal of Productivity and Performance Management*, 61:445–451. <https://doi.org/10.1108/17410401211212689>.

- Schultz, C. R. (1987). Forecasting and inventory control for sporadic demand under periodic review. *The Journal of the Operational Research Society*, 38(5):453–458. <https://doi.org/10.2307/2582735>.
- Silver, E. A. (2008). Inventory management: An overview, canadian publications, practical applications and suggestions for future research. *INFOR: Information Systems and Operational Research*, 46:15–27. <https://doi.org/10.3138/infor.46.1.15>.
- Spiliotis, E., Petropoulos, F., and Assimakopoulos, V. (2019). Improving the forecasting performance of temporal hierarchies. *PLOS ONE*, 14(10):1–21. <https://doi.org/10.1371/journal.pone.0223422>.
- Spithourakis, G. P., Petropoulos, F., Babai, M. Z., Nikolopoulos, K., and Assimakopoulos, V. (2011). Improving the performance of popular supply chain forecasting techniques. *Supply Chain Forum: An International Journal*, 12:16–25. <https://doi.org/10.1080/16258312.2011.11517277>.
- Spithourakis, G. P., Petropoulos, F., Nikolopoulos, K., and Assimakopoulos, V. (2012). A systemic view of the ADIDA framework. *IMA Journal of Management Mathematics*, 25(2):125–137. <https://doi.org/10.1093/imaman/dps031>.
- Suganthi, L. and Samuel, A. A. (2012). Energy models for demand forecasting—a review. *Renewable and Sustainable Energy Reviews*, 16(2):1223–1240. <https://doi.org/10.1016/j.rser.2011.08.014>.
- Swain, P. H. and Hauska, H. (1977). The decision tree classifier: Design and potential. *IEEE Transactions on Geoscience Electronics*, 15:142–147. <https://doi.org/10.1109/TGE.1977.6498972>.
- Syntetos, A. and Boylan, J. (2001). On the bias of intermittent demand estimates. *International Journal of Production Economics*, 71:457–466. [https://doi.org/10.1016/S0925-5273\(00\)00143-2](https://doi.org/10.1016/S0925-5273(00)00143-2).
- Syntetos, A. A. and Boylan, J. E. (2005). The accuracy of intermittent demand estimates. *International Journal of Forecasting*, 21:303–314. <https://doi.org/10.1016/j.ijforecast.2004.10.001>.
- Teunter, R. H., Babai, M. Z., and Syntetos, A. A. (2009). Abc classification: Service levels and inventory costs. *Production and Operations Management*, 19:343–352. <https://doi.org/10.1111/j.1937-5956.2009.01098.x>.
- Teunter, R. H., Syntetos, A. A., and Zied Babai, M. (2011). Intermittent demand: Linking forecasting to inventory obsolescence. *European Journal of Operational Research*, 214(3):606–615. <https://doi.org/10.1016/j.ejor.2011.05.018>.
- Thomakos, D. and Nikolopoulos, K. (2018). *Forecasting With The Theta Method: Theory and Applications*. Wiley.
- Thorndike, R. L. (1953). Who belongs in the family? *Psychometrika*, 18:267–276. <https://doi.org/10.1007/BF02289263>.
- Tunc, H., Kilic, O. A., Tarim, S. A., and Eksioglu, B. (2011). The cost of using stationary inventory policies when demand is non-stationary. *Omega*, 39(4):410–415. <https://doi.org/10.1016/j.omega.2010.09.005>.

- Ulrich, M., Jahnke, H., Langrock, R., Pesch, R., and Senge, R. (2022). Classification-based model selection in retail demand forecasting. *International Journal of Forecasting*, 38:209–223. <https://doi.org/10.1016/j.ijforecast.2021.05.010>.
- van Kampen, T. J., Akkerman, R., and van Donk, D. P. (2012). Sku classification: a literature review and conceptual framework. *International Journal of Operations & Production Management*, 32:850–876. <https://doi.org/10.1108/01443571211250112>.
- Wickramasuriya, S. L., Athanasopoulos, G., and Hyndman, R. J. (2019). Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, 114(526):804–819. <https://doi.org/10.1080/01621459.2018.1448825>.
- Wickramasuriya, S. L., Turlach, B. A., and Hyndman, R. J. (2020). Optimal non-negative forecast reconciliation. *Statistics and Computing*, 30:1167–1182. <https://doi.org/10.1007/s11222-020-09930-0>.
- Willemain, T. R., Smart, C. N., and Schwarz, H. F. (2004). A new approach to forecasting intermittent demand for service parts inventories. *International Journal of Forecasting*, 20:375–387. 10.1016/S0169-2070(03)00013-X.
- Willemain, T. R., Smart, C. N., Shockor, J. H., and DeSautels, P. A. (1994). Forecasting intermittent demand in manufacturing: a comparative evaluation of croston’s method. *International Journal of Forecasting*, 10:529–538. [https://doi.org/10.1016/0169-2070\(94\)90021-3](https://doi.org/10.1016/0169-2070(94)90021-3).
- Williams, T. M. (1984). Stock control with sporadic and slow-moving demand. *Journal of the Operational Research Society*, 35:939–948. <https://doi.org/10.1057/jors.1984.185>.
- Wirth, R. and Hipp, J. (2000). Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, volume 1, pages 29–40. Manchester. https://www.researchgate.net/publication/239585378_CRISP-DM_Towards_a_standard_process_model_for_data_mining.
- Çerağ Pinçe, Turrini, L., and Meissner, J. (2021). Intermittent demand forecasting for spare parts: A critical review. *Omega*, 105:102513. <https://doi.org/10.1016/j.omega.2021.102513>.

Appendix A

Forecast performance metrics

The formulas depicted below were applied to each period in the forecast horizon. Every average calculated was weighted according to relative importance in demand to avoid indeterminacy originating from division by null values.

$$ME = \frac{\sum_{i=1}^n \hat{y}_i - y_i}{n} \quad (A.1)$$

$$MSE = \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n} \quad (A.2)$$

$$wMAPE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{\sum_{i=1}^n y_i} \quad (A.3)$$

$$wMASE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{\sum_{i=1}^n |y_i - y_{i-1}|} \quad (A.4)$$

$$wBias = \frac{\sum_{i=1}^n \hat{y}_i - y_i}{\sum_{i=1}^n y_i} \quad (A.5)$$

$$SAE = \sum_{i=1}^n |\hat{y}_i - y_i| \quad (A.6)$$

$$SSE = \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (A.7)$$

Where:

n = Total amount of time series in the dataset;

y_i = Actual value observed in the date of the forecast;

$\hat{y}_i$ = Forecast value.

Appendix B

Variables included and hyperparameters tuned in the GBM models

Table B.1: Hyperparameters tuned and values tested for the GBM models.

	Parameter description	Values tested
max_depth	Maximum tree depth	0:12 by = 1
sample_rate	Ratio of rows sampled from training set	0.2:0.8 by = 0.01
col_sample_rate	Ratio of columns sampled from training set	0.2:0.8 by = 0.01
col_sample_rate_per_tree	Column sample rate per tree; This value is multiplicative with col_sample_rate	0.2:0.8 by = 0.01
col_sample_rate_change_per_level	Specifies to change the column sampling rate as a function of the depth in the tree	0.9:1 by = 0.01
min_rows	Minimum number of observations for a leaf	x^2 for x in 1:log(n_rows)
nbins	Maximum number of bins for the histogram to build	4:13 by = 1
ntrees	Number of trees to build	200,300,400
learn_rate	Rate of error correction between successive trees	0.05:0.2 by = 0.01

Table B.2: Input variables included in the GBM models.

Variable description	
actual_sum_values	Sum of the time series historical records
actual_median	Median of the time series historical records
actual_mean	Average of the time series historical records
actual_standard_deviation	Standard deviation of the time series historical records
actual_variance	Variance of the time series historical records
actual_root_mean_square	Square root of sum of squared values of the time series historical records
actual_maximum	Maximum value of the time series historical records
actual_absolute_maximum	Maximum absolute value of the time series historical records
actual_minimum	Minimum of the time series historical records
actual_variation_coefficient	Coefficient of variation of the time series historical records
actual_sum_of_reoccurring_data_points	Amount of repeated values of the time series historical records
actual_sum_of_reoccurring_values	Sum of repeated values of the time series historical records
actual_ratio_value_number_to_time_series_length	Returns a factor which is 1 if all values in the time series occur only once, and below one if this is not the case
actual_longest_strike_above_mean	Returns the length of the longest consecutive subsequence in x that is bigger than the mean of x
actual_longest_strike_below_mean	Returns the length of the longest consecutive subsequence in x that is smaller than the mean of x
actual_count_above_t_0	Amount of positive historical records
actual_last_location_of_maximum	Returns the relative last location of the maximum value of historical records
actual_last_location_of_minimum	Returns the relative last location of the minimum value of historical records
gap_feature	Point in the forecast horizon for which predictions are to be generated
predicted_from_day¹	Day of the last period of historical records being used for prediction
forecast_date_day¹	Day of the period for which forecasts are to be generated
actual_length	Length of the time series
n_periods_since_last_order	Number of periods passed since a positive value was registered
predict_seasonal_naive_1²	Returns a seasonal naïve prediction of lag 1
predict_seasonal_naive_1_bool	Returns 1 if the seasonal naïve prediction of lag 1 is positive
abc_class_B	Returns 1 if the time series under evaluation corresponds to B class item according to the ABC framework
abc_class_C	Returns 1 if the time series under evaluation corresponds to C class item according to the ABC framework
xyz_class_Y	Returns 1 if the time series under evaluation corresponds to Y class item according to the XYZ framework
xyz_class_Z	Returns 1 if the time series under evaluation corresponds to Z class item according to the XYZ framework

¹ Analogous variables were created for the week, month, year, day of week, day of month, day of year, week of month, week of year, and month of year.² Analogous variables were created for 2, 3, 4, 5, 6, 9, and 12 periods.

Appendix C

Input variables for clustering

Table C.1: List of forecast algorithms applied.

Method	Description
Naïve Last 1	Each forecast is equal to the last observed value
Seasonal Naïve(12)	Each forecast is equal to the observed value 12 periods before
ARIMA	Forecasts from the best ARIMA method fitted to each series
Naïve Mean Last 3	Each forecast is equal to the average of last three observed values
ETS	Weighted average across all observations in the input time series dataset. The weights decrease exponentially over time, as opposed to remaining constant in typical moving average approaches.
SES	Simple exponential smoothing forecasts with optimal smoothing constant and initial value for each series
Linear Ensemble	Linear ensemble of all forecasts calculated with the remaining methods that attributes weights according to forecast accuracy
GBM Ensemble	GBM model that receives as input the forecasts calculated by each remaining method and the actual value observed as the response variable
Croston	Croston's method with optimized hyperparameters for each series

Table C.2: Features used for clustering algorithms.

	Variable description
count	Number of historical records in the time series
mean	Average of the historical records in the time series
std	Standard deviation of the historical records in the time series
min	Minimum value of the historical records in the time series
25%	Value for the 25% percentile of the historical records in the time series
50%	Value for the 50% percentile of the historical records in the time series
75%	Value for the 75% percentile of the historical records in the time series
max	Maximum value of the historical records in the time series
median	Median value of the historical records in the time series
count_above_0	Number of positive values in the historical records in the time series
n_periods_since_last_order	Number of periods passed since a positive value was registered
cv	Coefficient of variation of the time series
adi	Average inter-demand interval of the time series
abc_class_B	Returns 1 if the time series is classified as B according to the ABC framework
abc_class_C	Returns 1 if the time series is classified as C according to the ABC framework
xyz_class_Y	Returns 1 if the time series is classified as Y according to the XYZ framework
xyz_class_Z	Returns 1 if the time series is classified as Z according to the XYZ framework
seil_Lumpy	Returns 1 if the time series is classified as Lumpy according to the SEIL framework
seil_Smooth	Returns 1 if the time series is classified as Smooth according to the SEIL framework
seil_Intermittent	Returns 1 if the time series is classified as Intermittent according to the SEIL framework
min_sq_err_sum	Minimum value registered for the sum of squared errors across all forecasting methods employed
min_abs_err_sum	Minimum value registered for the sum of absolute errors across all forecasting methods employed
max_sq_err_sum	Maximum value registered for the sum of squared errors across all forecasting methods employed
max_abs_err_sum	Maximum value registered for the sum of absolute errors across all forecasting methods employed
median_sq_err_sum	Median of the sum of squared errors calculated across all forecasting methods employed
median_abs_err_sum	Median of the sum of absolute errors calculated across all forecasting methods employed
std_sq_err_sum	Standard deviation of the sum of squared errors calculated across all forecasting methods employed
std_abs_err_sum	Standard deviation of the sum of absolute errors calculated across all forecasting methods employed

Appendix D

Dataset characterization according to established frameworks

Table D.1: SEIL data for each case study.

		Total Sum	# Series	CV^2	ADI
Case Study	SEIL				
I	Erratic	5698307	296	1.2613	1.1222
	Intermittent	5763	1	0.4535	1.3750
	Lumpy	8091248	6185	16.6498	15.9489
	Smooth	10209382	318	0.1878	1.0126
II	Erratic	25408903	8	0.8579	1.2123
	Lumpy	157810925	81	33.6648	32.4934
	Smooth	13987596	2	0.3877	1.1433
III	Erratic	529417	66	0.9581	1.1073
	Lumpy	157169	168	11.3248	9.6083
	Smooth	5330639	118	0.2254	1.0125
IV	Erratic	18728	100	0.8962	1.2321
	Intermittent	3	1	0.3333	1.3333
	Lumpy	97782	6414	15.4678	15.9094
	Smooth	1740	45	0.0954	1.0148
V	Erratic	51238681	111	0.8184	1.1947
	Intermittent	152256	3	0.4801	1.3917
	Lumpy	137058437	2208	23.2596	22.4583
	Smooth	212255591	188	0.2850	1.0737

Table D.2: ABC data for each case study.

Case Study	ABC class	Total Sum	# Series	Relative Total Weight	Relative Amount Weight
I	A	16801828	99	0.6999	0.0146
	B	6001927	610	0.2500	0.0897
	C	1200945	6091	0.0500	0.8957
II	A	136900692	11	0.6942	0.1209
	B	48527423	17	0.2461	0.1868
	C	11779309	63	0.0597	0.6923
III	A	4157896	32	0.6910	0.0909
	B	1549878	63	0.2576	0.1790
	C	309451	257	0.0514	0.7301
IV	A	82767	893	0.6999	0.1361
	B	29572	2281	0.2501	0.3477
	C	5914	3386	0.0500	0.5162
V	A	279955478	163	0.6987	0.0649
	B	100707840	503	0.2513	0.2004
	C	20041647	1844	0.0500	0.7347

Table D.3: XYZ data for each case study.

Case Study	xyz_class	Total Sum	# Series	Relative Total Weight	Relative Amount Weight	CV
I	X	20972970	1360	0.8737	0.2000	1.0182
	Y	2450079	2039	0.1021	0.2999	2.4575
	Z	581651	3401	0.0242	0.5001	4.9444
II	X	61703525	18	0.3129	0.1978	0.9463
	Y	123697238	27	0.6272	0.2967	1.5176
	Z	11806661	46	0.0599	0.5055	6.4955
III	X	4796330	70	0.7971	0.1989	0.3402
	Y	909136	105	0.1511	0.2983	0.7434
	Z	311759	177	0.0518	0.5028	2.8333
IV	X	79284	1312	0.6705	0.2000	1.2496
	Y	26117	1967	0.2209	0.2998	2.2511
	Z	12852	3281	0.1087	0.5002	4.9278
V	X	319668756	502	0.7978	0.2000	0.7719
	Y	64763350	752	0.1616	0.2996	1.7941
	Z	16272859	1256	0.0406	0.5004	5.6316

Appendix E

Implemented methodologies results

Table E.1: Average results obtained in the validation folds of case study I.

Method	wMAPE	wMASE	wBias	MSE
Seasonal Naïve (12)	0.7015	0.4008	0.0582	37749.50
Best Class + Size Estimate Combo	0.7431	0.4194	-0.0714	53177.82
ADIDA(2, Seasonal Naïve (12), Mean)	0.8674	0.4961	-0.0005	43383.86
ARIMA	0.9544	0.5440	-0.0102	59224.00
ADIDA(9, Seasonal Naïve (6), Mean)	0.9825	0.5698	-0.1664	58778.17
MAPA(Seasonal Naïve (12), Mean)	1.0265	0.5872	0.0126	65208.78
ADIDA(9, Average Last 3, Mean)	0.9955	0.5901	-0.1808	54688.27
MAPA(Seasonal Naïve (12), Median)	1.0345	0.5918	-0.0191	72295.24
MAPA(Seasonal Naïve (6), Median)	1.0657	0.6099	-0.2882	81075.87
ADIDA(9, SES, Mean)	1.0355	0.6155	-0.1584	64534.62
MAPA(Seasonal Naïve (6), Mean)	1.0791	0.6177	-0.2574	81335.70
ADIDA(9, Naïve Lag 1, Mean)	1.0394	0.6180	-0.1596	64554.71
MAPA(Average Last 3, Median)	1.0794	0.6182	-0.2129	82697.43
MAPA(Average Last 3, Mean)	1.0892	0.6236	-0.1552	81870.52
MAPA(Naïve Lag 1, Median)	1.1007	0.6298	-0.1500	100007.24
MAPA(SES, Median)	1.1113	0.6361	-0.1322	99965.64
ADIDA(3, SES, Mean)	1.0994	0.6394	0.0652	103398.49
ADIDA(3, Naïve Lag 1, Mean)	1.1057	0.6394	0.0676	103766.98
MAPA(Naïve Lag 1, Mean)	1.1183	0.6396	-0.0692	96705.77
ADIDA(2, Average Last 3, Mean)	1.1021	0.6422	-0.0098	106978.79
MAPA(SES, Mean)	1.1255	0.6439	-0.0574	96519.29
Average Last 3	1.2269	0.6991	0.1018	137839.84
Seasonal Naïve (6)	1.2906	0.7366	-0.2776	190284.96
SES	1.2986	0.7387	0.1857	143641.50
Naïve Lag 1	1.3934	0.7923	0.2615	203174.07
TSB	1.6308	0.9114	-0.2492	156715.44
SBA	3.3626	1.8654	1.7065	309541.99
Croston	3.3626	1.8654	1.7066	309542.02

Table E.2: Average results obtained in the test folds of case study I.

Method	wMAPE	wMASE	wBias	MSE
Best Class + Size Estimate Combo	0.5023	0.3575	-0.1474	18742.15
Seasonal Naïve (12)	0.8014	0.5703	0.0736	57338.13
ADIDA(2, Seasonal Naïve (12), Mean)	0.8210	0.5861	-0.1374	43646.71
ARIMA	0.8300	0.5906	-0.1706	50863.05
ADIDA(9, Seasonal Naïve (6), Mean)	0.8897	0.6464	-0.3708	62239.97
MAPA(Seasonal Naïve (12), Mean)	0.9220	0.6582	-0.1566	76365.81
MAPA(Seasonal Naïve (12), Median)	0.9332	0.6662	-0.1792	90096.66
MAPA(Seasonal Naïve (6), Median)	0.9702	0.6926	-0.4608	77277.42
MAPA(Seasonal Naïve (6), Mean)	0.9931	0.7089	-0.3991	88297.28
Naïve Lag 1	1.0005	0.7119	-0.4814	77604.28
SES	1.0392	0.7395	-0.1298	111286.01
MAPA(Naïve Lag 1, Median)	1.0506	0.7500	-0.2132	144391.91
MAPA(Naïve Lag 1, Mean)	1.0577	0.7550	-0.1874	138917.72
MAPA(SES, Median)	1.0657	0.7607	-0.1851	144150.78
MAPA(SES, Mean)	1.0661	0.7610	-0.1598	138291.41
ADIDA(3, Naïve Lag 1, Mean)	1.0627	0.7620	-0.0899	171209.47
Average Last 3	1.0833	0.7709	-0.0681	169952.01
MAPA(Average Last 3, Mean)	1.0833	0.7733	-0.1814	145118.44
ADIDA(3, SES, Mean)	1.0785	0.7734	-0.0578	171016.67
MAPA(Average Last 3, Median)	1.0929	0.7801	-0.2067	158158.23
ADIDA(2, Average Last 3, Mean)	1.1420	0.8152	-0.0271	198129.36
Seasonal Naïve (6)	1.1652	0.8292	-0.2809	138208.38
TSB	1.4462	1.0292	-0.4222	116981.76
SBA	2.7407	1.9503	1.1109	273108.75
Croston	2.7408	1.9504	1.1110	273109.57

Table E.3: Average results obtained in the validation folds of case study II.

Method	wMAPE	wMASE	wBias	MSE
Best Class + Size Estimate Combo	0.6996	0.4029	-0.0085	5.282167e+08
ADIDA(8, Seasonal Naïve (6), Mean)	1.5787	0.5898	0.3774	5.780252e+08
ADIDA(9, SES, Mean)	1.6186	0.6072	0.4770	6.700313e+08
ADIDA(9, Naïve Lag 1, Mean)	1.6294	0.6105	0.4885	6.845603e+08
TSB	0.9944	0.6162	-0.3864	1.144002e+09
ADIDA(9, Average Last 3, Mean)	1.4263	0.6526	0.2135	9.247890e+08
Seasonal Naïve (12)	1.0792	0.6645	-0.0623	1.232371e+09
ADIDA(3, Seasonal Naïve (12), Mean)	1.2251	0.6823	0.1753	8.078178e+08
ARIMA	1.1701	0.7104	0.0846	1.048666e+09
MAPA(Seasonal Naïve (6), Median)	1.1772	0.7189	-0.1660	1.319854e+09
SBA	1.2040	0.7250	0.2215	1.076261e+09
Croston	1.2040	0.7250	0.2215	1.076261e+09
MAPA(Average Last 3, Median)	1.2267	0.7357	0.0059	1.252675e+09
MAPA(Average Last 3, Mean)	1.2316	0.7393	0.0395	1.260016e+09
MAPA(SES, Median)	1.2433	0.7435	0.0768	1.269813e+09
MAPA(Seasonal Naïve (6), Mean)	1.2229	0.7440	-0.1209	1.382016e+09
MAPA(Naïve Lag 1, Median)	1.2493	0.7450	0.0847	1.326259e+09
SES	1.2265	0.7463	0.0430	1.105305e+09
MAPA(SES, Mean)	1.2527	0.7490	0.1021	1.294597e+09
MAPA(Naïve Lag 1, Mean)	1.2672	0.7537	0.1379	1.350991e+09
MAPA(Seasonal Naïve (12), Mean)	1.2466	0.7598	0.1297	1.033113e+09
ADIDA(9, Seasonal Naïve (6), Mean)	1.1106	0.7598	-0.1545	1.239787e+09
ADIDA(2, Seasonal Naïve (12), Mean)	0.9601	0.7599	-0.1244	1.079252e+09
ADIDA(2, SES, Mean)	1.2034	0.7619	0.1557	1.342723e+09
MAPA(Seasonal Naïve (12), Median)	1.2604	0.7674	0.1231	1.066283e+09
ADIDA(8, Naïve Lag 1, Mean)	1.1746	0.7709	-0.0255	1.371221e+09
ADIDA(2, Average Last 3, Mean)	1.0314	0.7831	-0.0370	1.457031e+09
ADIDA(4, Average Last 3, Mean)	1.0742	0.7849	-0.0573	1.414927e+09
Average Last 3	1.3413	0.7927	0.2490	1.686492e+09
ADIDA(5, SES, Mean)	1.1148	0.8175	-0.0299	1.425668e+09
ADIDA(5, Naïve Lag 1, Mean)	1.1232	0.8221	-0.0094	1.458377e+09
Seasonal Naïve (6)	1.5221	0.9047	0.0375	2.960218e+09
Naïve Lag 1	1.6120	0.9339	0.4361	2.738426e+09

Table E.4: Average results obtained in the test folds of case study II.

Method	wMAPE	wMASE	wBias	MSE
TSB	1.1899	0.4524	-0.0417	3.961693e+08
Best Class + Size Estimate Combo	1.1931	0.4602	0.6935	8.792373e+08
Naïve Lag 1	1.2866	0.4813	-0.0622	5.485637e+08
ADIDA(2, Seasonal Naïve (12), Mean)	1.9163	0.5622	0.3612	6.377895e+08
ADIDA(3, Seasonal Naïve (12), Mean)	1.2862	0.5700	0.0518	5.740561e+08
ADIDA(9, Seasonal Naïve (6), Mean)	1.4795	0.5742	0.3422	6.423206e+08
Seasonal Naïve (12)	1.5527	0.5841	0.0940	8.485043e+08
MAPA(Seasonal Naïve (6), Median)	1.5566	0.5951	0.4257	8.696642e+08
MAPA(Seasonal Naïve (12), Mean)	1.6049	0.6099	0.4454	6.359716e+08
SES	1.6491	0.6231	0.6613	6.701129e+08
MAPA(Seasonal Naïve (12), Median)	1.6680	0.6340	0.5228	6.927279e+08
MAPA(Seasonal Naïve (6), Mean)	1.6797	0.6420	0.5437	1.097819e+09
MAPA(Naïve Lag 1, Mean)	1.7344	0.6632	0.7847	1.071583e+09
ARIMA	1.7814	0.6686	0.7625	9.701137e+08
MAPA(Naïve Lag 1, Median)	1.7615	0.6735	0.8232	1.147315e+09
ADIDA(5, Naïve Lag 1, Mean)	1.5790	0.6753	0.6611	1.062693e+09
ADIDA(8, Naïve Lag 1, Mean)	2.2862	0.6792	1.3465	1.208894e+09
SBA	1.8003	0.6804	0.9074	7.778057e+08
Croston	1.8003	0.6804	0.9074	7.778057e+08
MAPA(SES, Median)	1.7889	0.6835	0.8475	1.204361e+09
ADIDA(5, SES, Mean)	1.6056	0.6843	0.6892	1.144075e+09
MAPA(SES, Mean)	1.8025	0.6876	0.8570	1.186106e+09
MAPA(Average Last 3, Median)	1.8433	0.7034	0.8891	1.234698e+09
MAPA(Average Last 3, Mean)	1.8590	0.7095	0.9366	1.249119e+09
Average Last 3	1.8821	0.7283	0.9177	1.263017e+09
ADIDA(4, Average Last 3, Mean)	1.7410	0.7399	0.8135	1.262824e+09
ADIDA(2, SES, Mean)	2.5996	0.7627	1.3653	1.531630e+09
ADIDA(2, Average Last 3, Mean)	2.7213	0.7984	1.8304	1.610853e+09
Seasonal Naïve (6)	2.3364	0.8854	0.9536	3.194159e+09

Table E.5: Average results obtained in the validation folds of case study III.

Method	wMAPE	wMASE	wBias	MSE
Best Class + Size Estimate Combo	0.7815	1.2263	-0.2620	8011.58
ADIDA(2, Average Last 3, Mean)	0.8592	1.2264	-0.3939	10139.63
Average Last 3	0.8978	1.3505	-0.4325	10941.51
ADIDA(2, Naïve Lag 1, Mean)	0.8752	1.3751	-0.3909	10475.68
MAPA(Naïve Lag 1, Mean)	0.9097	1.3818	-0.4352	10738.75
MAPA(Naïve Lag 1, Median)	0.9139	1.3879	-0.4676	11063.01
MAPA(SES, Mean)	0.9236	1.4023	-0.4312	10780.21
MAPA(SES, Median)	0.9278	1.4090	-0.4607	11100.42
ADIDA(2, SES, Mean)	0.9057	1.4170	-0.3742	10544.21
ADIDA(3, Seasonal Naïve (12), Mean)	1.0050	1.4218	-0.1827	7953.38
MAPA(Average Last 3, Mean)	0.9463	1.4366	-0.4538	11291.18
Naïve Lag 1	0.9541	1.4395	-0.2808	12228.23
MAPA(Average Last 3, Median)	0.9505	1.4425	-0.4593	11286.82
ADIDA(7, Naïve Lag 1, Mean)	0.9246	1.4531	-0.4273	11542.37
ADIDA(7, SES, Mean)	0.9309	1.4636	-0.4270	11574.81
SES	0.9942	1.4981	-0.2653	10140.28
ADIDA(7, Average Last 3, Mean)	0.9597	1.5079	-0.4226	11628.89
MAPA(Seasonal Naïve (6), Median)	1.0295	1.5630	-0.3800	10850.07
MAPA(Seasonal Naïve (6), Mean)	1.0299	1.5635	-0.3728	10706.31
MAPA(Seasonal Naïve (12), Median)	1.0364	1.5774	-0.3625	9817.30
TSB	1.0505	1.5795	-0.7007	12737.19
MAPA(Seasonal Naïve (12), Mean)	1.0390	1.5820	-0.3192	9610.85
ARIMA	1.0619	1.6023	-0.2744	11639.94
ADIDA(9, Seasonal Naïve (6), Mean)	1.0222	1.6217	-0.3146	10510.94
Seasonal Naïve (6)	1.0887	1.6276	-0.4736	13290.56
Seasonal Naïve (12)	1.1832	1.7788	-0.2200	15649.75
ADIDA(9, Average Last 3, Mean)	0.9779	1.8369	-0.4783	15082.19
Croston	1.2862	1.9277	-0.3047	11901.14
SBA	1.2862	1.9277	-0.3047	11901.14
ADIDA(5, Seasonal Naïve (12), Mean)	1.0235	1.9278	-0.3760	12563.74

Table E.6: Average results obtained in the test folds of case study III.

Method	wMAPE	wMASE	wBias	MSE
Best Class + Size Estimate Combo	0.5668	1.1108	-0.2826	5407.80
ADIDA(2, Average Last 3, Mean)	0.8534	1.5701	-0.2200	9711.43
Average Last 3	0.8358	1.6278	-0.2478	10468.20
ADIDA(2, SES, Mean)	0.9298	1.6441	-0.1649	14113.79
ADIDA(2, Naïve Lag 1, Mean)	0.9389	1.6588	-0.1645	14267.37
MAPA(Naïve Lag 1, Mean)	0.8471	1.6634	-0.3902	11989.18
MAPA(SES, Mean)	0.8489	1.6667	-0.3909	11998.15
MAPA(Naïve Lag 1, Median)	0.8502	1.6699	-0.4334	12529.34
SES	0.8604	1.6740	-0.3796	12379.04
MAPA(SES, Median)	0.8527	1.6745	-0.4313	12517.09
MAPA(Average Last 3, Mean)	0.8677	1.7100	-0.4633	12842.57
MAPA(Average Last 3, Median)	0.8741	1.7242	-0.5002	13675.27
Naïve Lag 1	0.8984	1.7691	-0.2809	19347.05
ARIMA	0.9916	1.9327	-0.3700	15919.89
Seasonal Naïve (6)	0.9870	1.9381	-0.6742	19032.20
MAPA(Seasonal Naïve (6), Median)	0.9885	1.9505	-0.6127	17223.10
TSB	0.9994	1.9617	-0.7718	17695.55
MAPA(Seasonal Naïve (6), Mean)	0.9993	1.9723	-0.6099	17635.28
MAPA(Seasonal Naïve (12), Median)	1.0541	2.0745	-0.4798	15364.61
MAPA(Seasonal Naïve (12), Mean)	1.0736	2.1127	-0.4584	15704.40
ADIDA(7, Naïve Lag 1, Mean)	0.8692	2.1128	-0.5763	16536.37
ADIDA(7, SES, Mean)	0.8706	2.1162	-0.5795	16567.63
ADIDA(9, Seasonal Naïve (6), Mean)	1.0275	2.1579	-0.5582	18199.20
ADIDA(7, Average Last 3, Mean)	0.9032	2.1954	-0.6171	18941.84
ADIDA(3, Seasonal Naïve (12), Mean)	1.1236	2.2260	-0.4640	17769.66
SBA	1.1522	2.2566	-0.4690	16411.68
Croston	1.1522	2.2566	-0.4690	16411.68
Seasonal Naïve (12)	1.2256	2.4031	-0.3795	21824.14

Table E.7: Average results obtained in the validation folds of case study IV.

Method	wMAPE	wMASE	wBias	MSE
Best Class + Size Estimate Combo	0.8970	0.5247	0.1080	9.32
ADIDA(5, SES, Mean)	1.0699	0.5660	-0.2987	1.30
ADIDA(5, Average Last 3, Mean)	1.0766	0.5679	-0.1844	1.34
ADIDA(4, Seasonal Naïve (6), Mean)	1.0873	0.5728	-0.2192	1.29
ADIDA(7, Seasonal Naïve (12), Mean)	1.2079	0.6409	-0.0724	2.08
MAPA(Naïve Lag 1, Median)	1.1797	0.6439	-0.3924	46.90
MAPA(Naïve Lag 1, Mean)	1.2058	0.6572	-0.3625	47.59
ADIDA(5, Naïve Lag 1, Mean)	1.1356	0.6600	-0.3638	47.02
MAPA(Average Last 3, Median)	1.2105	0.6613	-0.2680	45.13
MAPA(Average Last 3, Mean)	1.2167	0.6644	-0.2628	45.19
MAPA(SES, Median)	1.2203	0.6665	-0.3380	46.09
MAPA(Seasonal Naïve (6), Median)	1.2315	0.6725	-0.1417	44.81
Average Last 3	1.2412	0.6727	-0.4169	49.13
MAPA(SES, Mean)	1.2420	0.6779	-0.3126	46.17
ARIMA	1.2681	0.6898	-0.1507	44.89
MAPA(Seasonal Naïve (6), Mean)	1.2638	0.6899	-0.1106	44.97
ADIDA(4, Naïve Lag 1, Mean)	1.2249	0.6913	-0.4130	70.80
MAPA(Seasonal Naïve (12), Median)	1.2767	0.7008	-0.1348	45.12
ADIDA(2, Average Last 3, Mean)	1.1519	0.7057	-0.4774	64.78
ADIDA(2, SES, Mean)	1.2890	0.7157	-0.4031	55.80
Seasonal Naïve (6)	1.3170	0.7179	-0.2655	46.06
Naïve Lag 1	1.3408	0.7214	-0.4122	82.60
ADIDA(2, Seasonal Naïve (6), Mean)	1.3071	0.7256	-0.1998	56.73
TSB	1.3070	0.7274	-0.2550	54.54
MAPA(Seasonal Naïve (12), Mean)	1.3278	0.7282	-0.0673	45.17
SES	1.3848	0.7529	-0.0890	43.25
ADIDA(2, Seasonal Naïve (12), Mean)	1.4087	0.7867	-0.0178	56.32
Seasonal Naïve (12)	1.4887	0.8111	0.0978	46.66
ADIDA(3, Average Last 3, Mean)	NaN	0.8372	NaN	126.68
SBA	3.9593	2.1820	2.8508	43.27
Croston	3.9594	2.1821	2.8509	43.27

Table E.8: Average results obtained in the test folds of case study IV.

Method	wMAPE	wMASE	wBias	MSE
Best Class + Size Estimate Combo	0.6306	0.2838	-0.0017	0.38
Seasonal Naïve (6)	1.1018	0.4910	-0.7983	0.69
ADIDA(2, Seasonal Naïve (6), Mean)	1.1274	0.5032	-0.5917	0.64
MAPA(Seasonal Naïve (6), Median)	1.1586	0.5146	-0.2653	0.64
MAPA(Seasonal Naïve (6), Mean)	1.1822	0.5253	-0.2508	0.61
ADIDA(5, Naïve Lag 1, Mean)	1.5013	0.5448	0.0587	11.10
TSB	1.3069	0.5776	-0.1106	0.53
ARIMA	1.3561	0.5965	0.2306	1.80
MAPA(Average Last 3, Median)	1.3717	0.6047	-0.1127	8.37
MAPA(Seasonal Naïve (12), Median)	1.4028	0.6187	0.2972	1.07
MAPA(Average Last 3, Mean)	1.4167	0.6246	-0.0312	10.46
MAPA(Naïve Lag 1, Median)	1.4349	0.6362	0.0764	12.63
MAPA(SES, Median)	1.4429	0.6393	0.0821	12.07
ADIDA(2, Average Last 3, Mean)	1.7903	0.6444	0.3932	29.39
MAPA(Seasonal Naïve (12), Mean)	1.4642	0.6455	0.3849	1.21
Naïve Lag 1	1.4624	0.6477	0.3755	3.74
SES	1.4793	0.6504	0.3491	1.77
MAPA(Naïve Lag 1, Mean)	1.5099	0.6705	0.2119	16.43
MAPA(SES, Mean)	1.5285	0.6773	0.2229	16.76
ADIDA(2, Seasonal Naïve (12), Mean)	1.5416	0.6795	0.5172	2.72
ADIDA(4, Naïve Lag 1, Mean)	1.4095	0.6994	0.0496	17.33
ADIDA(3, Average Last 3, Mean)	1.4447	0.7053	-0.0298	19.73
Seasonal Naïve (12)	1.6461	0.7207	0.5385	8.97
Average Last 3	1.7482	0.7659	0.5375	29.55
ADIDA(2, SES, Mean)	1.8170	0.8136	0.7175	49.31
SBA	4.5834	2.0072	3.5824	2.49
Croston	4.5835	2.0072	3.5825	2.49

Table E.9: Average results obtained in the validation folds of case study V.

Method	wMAPE	wMASE	wBias	MSE
Best Class + Size Estimate Combo	0.7424	0.6162	-0.0621	5771009.28
ADIDA(7, Naïve Lag 1, Mean)	0.7478	0.6549	-0.1088	3970188.71
ADIDA(8, SES, Mean)	0.7539	0.6556	-0.1037	4108495.68
ADIDA(2, Average Last 3, Mean)	0.7905	0.7217	-0.1182	4931259.29
ADIDA(4, Seasonal Naïve (6), Mean)	0.8232	0.7299	-0.1498	4796303.25
ADIDA(6, SES, Mean)	0.8161	0.7510	-0.1490	5816640.02
MAPA(Naïve Lag 1, Median)	0.9097	0.7510	-0.0622	5551605.12
MAPA(Naïve Lag 1, Mean)	0.9164	0.7564	-0.0445	5573183.58
MAPA(SES, Median)	0.9177	0.7577	-0.0601	5615633.48
ADIDA(2, Naïve Lag 1, Mean)	0.8198	0.7631	-0.1014	5534891.61
MAPA(SES, Mean)	0.9247	0.7633	-0.0448	5652290.61
MAPA(Average Last 3, Median)	0.9277	0.7663	-0.0710	5747604.88
MAPA(Average Last 3, Mean)	0.9293	0.7676	-0.0625	5722072.18
Average Last 3	0.9309	0.7700	-0.0110	5822249.37
ADIDA(9, Naïve Lag 1, Mean)	1.0889	0.7723	0.0754	6420855.38
ADIDA(9, SES, Mean)	1.0914	0.7752	0.0774	6416730.76
ADIDA(7, Average Last 3, Mean)	1.0538	0.7827	0.0357	6763704.88
MAPA(Seasonal Naïve (6), Median)	0.9566	0.7909	-0.0925	6060530.62
ADIDA(6, Seasonal Naïve (6), Mean)	0.8625	0.7930	-0.1723	6342878.95
MAPA(Seasonal Naïve (6), Mean)	0.9632	0.7962	-0.0777	6082679.31
ADIDA(7, Seasonal Naïve (6), Mean)	1.0702	0.7998	0.0262	6925550.92
ADIDA(9, Seasonal Naïve (6), Mean)	1.1456	0.8172	0.0887	6797849.41
ADIDA(9, Average Last 3, Mean)	1.1457	0.8173	0.0910	6807624.90
ADIDA(8, Seasonal Naïve (12), Mean)	1.0251	0.8251	-0.0209	6683436.98
Naïve Lag 1	1.0102	0.8361	0.0245	7182155.69
SES	1.0115	0.8367	0.0756	5820516.30
ARIMA	1.0246	0.8475	0.0087	6812159.07
MAPA(Seasonal Naïve (12), Median)	1.0381	0.8602	-0.0444	6791359.68
MAPA(Seasonal Naïve (12), Mean)	1.0461	0.8670	-0.0300	6711289.55
Seasonal Naïve (6)	1.0518	0.8734	-0.0594	8958468.14
Seasonal Naïve (12)	1.1415	0.9504	-0.0239	8580315.17
TSB	1.6240	1.4059	-0.1511	11692407.47
SBA	3.3758	2.9101	1.8574	23597349.54
Croston	3.3759	2.9101	1.8574	23597397.59

Table E.10: Average results obtained in the test folds of case study V.

Method	wMAPE	wMASE	wBias	MSE
Best Class + Size Estimate Combo	0.5343	0.2742	-0.2280	3025249.36
ADIDA(6, Seasonal Naïve (6), Mean)	9.1321	0.8016	7.9971	6557866.04
ADIDA(6, SES, Mean)	9.7588	0.8566	8.8363	6831843.23
MAPA(Seasonal Naïve (6), Median)	2.1902	0.8841	1.0023	7979878.35
MAPA(Naïve Lag 1, Median)	2.2457	0.8891	1.2021	7245347.09
ADIDA(2, Naïve Lag 1, Mean)	10.1092	0.8903	9.2889	7947781.39
MAPA(SES, Median)	2.2502	0.8923	1.2084	7234618.37
MAPA(Average Last 3, Median)	2.2540	0.8952	1.1756	7555944.86
MAPA(Naïve Lag 1, Mean)	2.2682	0.8965	1.2418	7237113.65
MAPA(Seasonal Naïve (6), Mean)	2.2227	0.8967	1.0416	8138829.90
MAPA(Average Last 3, Mean)	2.2638	0.8983	1.1975	7460383.24
MAPA(SES, Mean)	2.2679	0.8996	1.2443	7220921.90
ADIDA(2, Average Last 3, Mean)	2.2939	0.9042	1.3052	7381015.28
ADIDA(8, Seasonal Naïve (12), Mean)	2.2590	0.9136	1.1262	8366382.13
ADIDA(7, Naïve Lag 1, Mean)	1.6389	0.9166	0.5629	8708362.76
MAPA(Seasonal Naïve (12), Median)	2.2526	0.9170	1.0834	8121680.87
Average Last 3	2.3562	0.9180	1.3694	7419629.98
MAPA(Seasonal Naïve (12), Mean)	2.2781	0.9248	1.1090	7965575.42
ADIDA(8, SES, Mean)	1.6206	0.9273	0.5161	9032862.34
ARIMA	2.3833	0.9323	1.1735	7921579.66
Seasonal Naïve (6)	2.3617	0.9410	1.1370	9876740.94
Seasonal Naïve (12)	2.3413	0.9471	0.9742	9136951.00
Naïve Lag 1	2.4512	0.9534	1.3724	8735958.85
SES	2.4517	0.9572	1.3737	7289915.70
ADIDA(4, Seasonal Naïve (6), Mean)	1.6449	0.9813	0.4406	10639060.54
TSB	2.6971	1.1452	0.8917	10135620.10
SBA	6.9182	2.7681	5.3782	22810824.05
Croston	6.9184	2.7682	5.3783	22810872.00

Appendix F

Clustering procedure results

Table F.1: Characterization of the time series belonging to each cluster according to SEIL, ABC and XYZ analyses.

SEIL	ABC	cluster	0	1	2	3	4
		XYZ					
Erratic	A	X	-	36	-	-	-
	B	X	-	115	-	-	-
	C	X	-	141	-	-	-
		Y	-	-	2	-	-
Intermittent	B	X	-	1	-	-	-
Lumpy	A	X	-	28	-	-	-
		Y	-	3	-	-	-
	B	X	-	-	-	-	306
		Y	-	-	-	-	79
		Z	-	-	-	-	22
	C	X	-	-	-	1184	-
		Y	-	-	2183	-	-
		Z	2245	-	-	-	-
Smooth	A	X	-	32	-	-	-
	B	X	-	86	-	-	-
	C	X	-	55	-	-	-

Table F.2: Best performing methods and results per cluster and point in the forecast horizon.

			wMAPE	wMASE	wBias
Cluster	Gap	Method			
0	2	Seasonal Naïve (12)	1.2227	0.1322	-0.5830
	3	Seasonal Naïve (12)	1.3876	0.1259	-0.2334
1	2	GBM Ensemble	0.2423	0.0635	0.0084
	3	GBM Ensemble	0.2577	0.0633	0.0479
2	2	Naïve Lag 1	1.4714	0.4580	-0.4563
	3	Naïve Lag 1	1.3663	0.4951	-0.4761
3	2	Seasonal Naïve (12)	0.9689	0.6526	0.1604
	3	GBM Ensemble	1.0157	0.6871	-0.0553
4	2	GBM Ensemble	0.5777	0.3797	-0.1139
	3	GBM Ensemble	0.7227	0.5348	-0.0799

Table F.3: Average results obtained from selecting the best performing methods for each gap and class. The existing solution is indicated as “no_class”.

	wMAPE	wMASE	wBias
no_class	0.3694	0.3582	0.0092
ABC + XYZ	0.3521	0.2307	-0.0154
Clusters	0.3548	0.1512	-0.0137
ABC + SEIL	0.3607	0.3041	-0.0091
XYZ	0.3630	0.1293	-0.0148
ABC	0.3647	0.1989	-0.0151
SEIL	0.3694	0.1450	0.0094