technische universität
dortmund

fakultät
statistik

Forecasting in the Semiconductor Industry

Dissertation zur Erlangung des Doktorgrades Dr. rer. nat. der Fakultät Statistik der Technischen Universität Dortmund

Vorgelegt von

Louis Steinmeister

Dortmund, November 2024

Amtierender Dekan:
Prof. Dr. Philipp Doebler

Gutachter:
Prof. Dr. Markus Pauly (Technische Universität Dortmund)
Dr. Uwe Ligges (Technische Universität Dortmund)

Tag der Prüfung:
19.12.2024

Abstract

Since the invention of the transistor at Bell Labs in 1947, semiconductors have slowly taken over the world. Today, they are ubiquitous in daily life. They power most devices with electronic components from washing machines to cars (conventional and electronic), from gaming consoles to data centers, from solar panels to industrial power plants. They don't only help us fend off tooth decay with electronic toothbrushes but also keep authoritarian aggressors at bay with modern weaponry. They also enable applications such as AI assistants and AI chatbots like ChatGPT, which have made knowledge more accessible and helped automate mundane tasks. Consequently, the semiconductor industry has become a significant driver and, as Chow and Choy (2006) note, a leading indicator of the economy. This dissertation investigates different statistical questions in the domain of semiconductor market forecasting and trend detection. The first investigation brings us into the realm of technological trend detection and technology life cycles. Common models to achieve these ends are technology diffusion models, such as the simple logistic growth model. Since this model represents a closed system view of a highly interconnected and complex phenomenon, we investigate the robustness of derived life cycle estimates with regard to an external shock, namely the COVID-19 pandemic, and its consequences on consumer demand and supply chains. While the logistic growth model takes a long-term perspective, we focus on the detection of shorter-term market dynamics with an investigation into forecasting the semiconductor market more broadly. A premier provider of semiconductor market data and industry forecasts is the World Semiconductor Trade Statistics (WSTS). As our second focus of attention, we investigate the precision of industry expert forecasts and explore if data-driven models can enhance such forecasts. Given that the WSTS product classification hierarchy has a degenerate structure, we proceed with an exploration of degenerate hierarchical forecast reconciliation to enable coherent forecasts. Lastly, we investigate the performance of a popular reconciliation algorithm, Trace Minimization (MinT), which was originally introduced by Wickramasuriya et al. (2019), in the context of very short time series through an elaborate simulation study and compare a proposed iterative alternative which requires far fewer parameters.

Acknowledgments

I first met Prof. Markus Pauly in 2017, when I was a Masters student in one of his lectures at Ulm University. Markus not only impressed me with his broad knowledge of statistics but also with his efforts to communicate the material effectively and his approachability outside of classes. Markus has been the main reason for my commitment to the doctoral program at TU Dortmund University, and I would like to thank him for his years of academic guidance and supervision.

I would also like to thank my co-authors for the collaboration. In particular, I would like to acknowledge Dr. Burim Ramosaj, who played a special role as my day-to-day supervisor during the early stages of my Ph.D. I thank my colleagues from the Chair of Mathematical Statistics and Applications in Industry for their support, particularly Lena Schmid and Lara Kuhlmann. Special thanks also go to Dr. Uwe Ligges as my second examiner.

My doctoral work was supported by a scholarship from the Graduate School of Logistics in collaboration with Infineon Technologies AG. I am grateful for the support and collaboration over the years. Particularly, I would like to recognize Dr. Natalie Gentner for her steady encouragement, and Henrik Zeller for his cooperation and support.

Furthermore, I would like to acknowledge Steven Schub and Marcus Holzheimer's invaluable support as my mentors. I am also grateful to Anjaya Shrestha, Austin Vandegriffe, and Marcos Carbonell for their friendship and advice.

Last but most importantly, I thank my parents, Markus and Walburga, my brother, Marvin, and my wonderful wife, Yuan, for their love and unwavering support. They stood by me through every high and low of this roller coaster journey. Without them, it would have been immeasurably more challenging and much less enjoyable.

"A theory is an identification of the facts of reality and/or of guidelines for human action. A good theory is a true theory, one that recognizes all the relevant facts, including the facts of human nature, and integrates them into a noncontradictory whole. Such a theory has to work in practice."

– Leonard Peikoff

List of Publications

This cumulative thesis is based on the following four manuscripts:

Article 1: Steinmeister, L., Ramosaj, B., Schröter, L., & Pauly, M. (2023). Testing The Limits: A Robustness Analysis Of Logistic Growth Models For Life Cycle Estimation During The COVID-19 Pandemic. In D. Herberger & M. Hübner (Eds.), Proceedings of the Conference on Production Systems and Logistics: CPSL. publish-Ing. doi.org/10.15488/15265

CRediT Author Statement:

Louis Steinmeister:	Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Project administration
Burim Ramosaj:	Writing - Review & Editing, Supervision
Leo Schröter:	Validation, Data Curation, Writing - Review & Editing
Markus Pauly:	Conceptualization, Funding acquisition, Methodology, Writing - Review & Editing, Supervision

Article 2: Steinmeister, L., & Pauly, M. (2025). Human vs. Machines: Who wins in semiconductor market forecasting? Expert Systems with Applications, 263(October 2024), 125719. doi.org/10.1016/j.eswa.2024.125719

CRediT Author Statement:

Louis Steinmeister:	Conceptualization, Data curation, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Project administration

Markus Pauly: Conceptualization, Funding acquisition, Methodology, Writing - Review & Editing, Supervision

The reuse of this article in the thesis is granted under the terms of the Creative Commons 4.0 BY License.

Article 3: Steinmeister, L., & Pauly, M. (2024). Degenerate Hierarchical Time Series Reconciliation With The Minimum Trace Algorithm in R. In D. Herberger & M. Hübner (Eds.), Proceedings of the Conference on Production Systems and Logistics: CPSL. publish-Ing. doi.org/10.15488/17729

CRediT Author Statement:

Louis Steinmeister: Conceptualization, Data curation, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Project administration

Markus Pauly: Conceptualization, Funding acquisition, Methodology, Writing - Review & Editing, Supervision

The reuse of this article in the thesis is granted under the terms of the Creative Commons 3.0 DE License.

Article 4: Steinmeister, L., Pauly, M., 2024, Iterative Trace Minimization for the Reconciliation of Very Short Hierarchical Time Series. doi.org/10.48550/arXiv.2409.18550

CRediT Author Statement:

Louis Steinmeister: Conceptualization, Data curation, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Project administration

Markus Pauly: Conceptualization, Funding acquisition, Methodology, Writing - Review & Editing, Supervision

Further publications:

(1) Kuhlmann, L., Fesca, F., Steinmeister, L., & Pauly, M. (2024). Hierarchical Sales Forecasting In Multichannel Distribution Considering Marketing Campaigns. In D. Herberger, & M. Hübner (Ed.), Proceedings of the Conference on Production Systems and Logistics: CPSL. publish-Ing. doi.org/10.15488/17741

Contents

Abstract iii

Acknowledgments v

List of Publications vii

I Introduction 1

1 Motivation 3

2 Statistical Methods 11

2.1 The Logistic Growth Model 11

2.1.1 Non-linear Least Squares Estimation 13

2.1.2 χ^2-Quantity Minimization 14

2.1.3 OLS Variance Minimization 16

2.2 A Very Brief Introduction to Time Series Analysis 18

2.3 Time Series Forecasting 21

2.3.1 Statistical Forecasting Methods 21

ARIMA 21

Simple Exponential Smoothing 23

Error, Trend, and Seasonality 24

2.3.2 Machine Learning Forecasting Methods 24

Random Forest 26

Extremely Randomized Trees 27

Gaussian Process Regression 28

K-Nearest Neighbors 29

Support Vector Regression 30

2.3.3 Forecast Evaluation Methods 31

2.4 Hierarchical Time Series Reconciliation 31

3 Summary of the Articles **37**
3.1 Article 1: Testing the Limits 37
3.2 Article 2: Human vs. Machines 39
3.3 Article 3: Degenerate Hierarchical Time Series Reconciliation 42
3.4 Article 4: Iterative Trace Minimization 44

4 Discussion and Outlook **47**

Bibliography **51**

II Publications **61**

Part I

Introduction

1 Motivation

Semiconductors; we usually think of them as critical components in computers and smartphones. Often, the first thing that comes to mind is central processing units (CPUs), which provide the "brain power" for computers and other interactive, multi-functional electronics. "Artificial Intelligence" (AI) gained a lot of attention with the release of ChatGPT by OpenAI on November 30, 2022, (Roumeliotis and Tselikas, 2023). This has raised public attention to the importance of AI-related chips such as graphics processing units (GPUs). This is also illustrated by the meteoric rise of NVIDIA's stock value from about USD 15.50 on 01 December 2022, to a high of about USD 140 on 20 June, 2024[1]. That marks a rise of over 900% in less than two years.

However, CPUs and GPUs only make up a fraction of the diverse semiconductor landscape that powers modern life: sensors, power switches, controllers, and more[2]. A myriad of these tiny chips with various purposes power devices we encounter in daily life: from smart LEDs and electronic toothbrushes to washing machines, cars (whether conventional or electronic), wind farms, and solar panels.

But few are aware that the semiconductor industry was born out of military applications, as Miller (2022) records. With modern precision weapons, semiconductors have become more important to national security than ever before. Recent events, such as Russia's 2022 invasion of Ukraine and subsequent Western sanctions, have highlighted this reliance (Nocetti, 2023; Axe, 2024).

The economic and military importance of semiconductors has been recognized with massive subsidies by policymakers in the U.S., the EU, and China: USD 280 billion in the U.S. (Taylor, 2023), EUR 43 billion in the EU (European Comission, 2023), and CNY 20.5 billion in China in 2023 alone (Pan, 2024). Even more impressively, Sam Altman, the CEO of OpenAI is seeking USD 5-7 trillion to "boost the world's chip-building capacity and expand its ability to power AI, among other things" according to Rajan

[1] tradingview.com/symbols/NASDAQ-NVDA

[2] In fact, the World Semiconductor Trade Statistics (WSTS) lists a total of over 100 product categories: semiconductors.org/wp-content/uploads/2021/02/Product_Classification_2021.pdf

(2024). At the time of writing, this amounts to the total market capitalization of the two worldwide largest corporations: Apple with USD 3.47 trillion and NVIDIA with USD 3.3 trillion in market capitalization[3]. These highlights provide ample motivation for the study of the semiconductor industry.

As previous authors have noted, the semiconductor industry is characterized by shortening life cycles, rapid technological advancements, and long lead times for fabrication and expanding fabrication capacity (Macher, 2006; Wu and Chien, 2008; Lv et al., 2018). Moreover, strategic decisions such as investments in R&D and fabrication capacity are extremely capital intensive (Chien et al., 2010; Aubry and Renou-Maissant, 2014), highlighting the need for reliable forecasts to inform such high-stakes decisions.

However, forecasting the semiconductor market comes with challenges. The semiconductor industry is heavily intertwined in global supply chains: on the one hand, there is a complex supply chain for fabrication, such as for lithography machines. On the other hand, semiconductors are crucial components of almost any electronics product. This means that the semiconductor industry lies upstream in the electronics supply chain. Consequently, it is exposed to the bullwhip effect, which describes the phenomenon where small fluctuations in consumer demand can result in large demand fluctuations upstream in the supply chain (Geary et al., 2006; Lee et al., 1997). Additionally, shortening product life cycles and rapid technological advancements mean that there is little time and data to identify new trends.

Two academic fields related to semiconductor market forecasting have recently received attention: 1) demand and sales forecasting and 2) semiconductor cycle prediction. These can be thought of as two opposing ends on a granularity spectrum.

Demand and sales forecasting usually aims to provide manufacturing intelligence and focuses on specific products or product groups for one or a few specific semiconductor companies. Examples of related studies are Wang and Chen (2019); Kapur et al. (2019); Chen and Chien (2018); Xu and Sharma (2017); Chien and Lin (2012); Chien et al. (2010). A common model in this domain is technology diffusion, see Kapur et al. (2019); Chen and Chien (2018); Chien et al. (2010).

Semiconductor cycle prediction, on the other hand, focuses on the cyclical behavior of the semiconductor market at large. Major research goals include the identification of leading indicators and providing explanations of the semiconductor cycle. Studies in this domain include Aubry and Renou-Maissant (2014, 2013); Chow and Choy (2006); Liu

[3]Based on: tradingview.com/markets/world-stocks/worlds-largest-companies. Accessed 10 October 2024.

and Chyi (2006); Liu (2005). These often use the Vector Autoregression or Vector Error Correction models[4].

Long-term demand forecasts are often of strategic interest, particularly in a rapidly advancing environment such as the semiconductor industry. Early identification of technologies under threat of disruption and promising new technologies can be vital for a business as the late adoption and subsequent demise of the mobile phone champions Nokia (Yueh, 2014) and Blackberry (Thompson, 2013) illustrates.

A common model for deriving life cycles and gauging these risks and opportunities is the logistic growth model, often referred to as S-curve (Modis, 1994; Foster, 1986), which is a type of diffusion model, see, for example, Sarkar (1998); Geroski (2000). It was originally introduced by Verhulst (1838) to describe population dynamics. However, as Modis (1994); Bejan and Lorente (2012); Kucharavy and De Guio (2011a,b) argue, growth patterns outside the discipline of population dynamics such as technological forecasting resemble very similar characteristics.

While there are several extensions of the logistic growth model such as the Bass diffusion model (Bass, 1969), the Gompertz curve (Tjørve and Tjørve, 2017), or the Richards' curve (Richards, 1959), the logistic growth model is simple and may have an upper hand when it comes to identifiability (Simpson et al., 2022). However, the logistic growth model presents a closed-system view which does not capture the interconnectivity of global supply chains, especially in the semiconductor market. This motivates our first investigation into the robustness of technology life cycle estimates derived from the logistic growth model under the presence of an external shock: the COVID-19 pandemic in the first Article.

Article 1: Steinmeister, L., Ramosaj, B., Schröter, L., & Pauly, M. (2023). Testing The Limits: A Robustness Analysis Of Logistic Growth Models For Life Cycle Estimation During The COVID-19 Pandemic. In D. Herberger & M. Hübner (Eds.), Proceedings of the Conference on Production Systems and Logistics: CPSL. publish-Ing. doi.org/10.15488/15265

While this contribution analyses an aspect of long-term demand forecasting for specific technologies, broader semiconductor market forecasts often inform important strategic and operational decision-making. In fact, Aubry and Renou-Maissant (2014) highlights

[4]We don't consider either model for two main reasons: 1) the time series lengths of the data considered here differ, meaning that a significant portion of the data would not be used, and 2) the complete time series history for training is very short (24 obs. at its minimum) while the number of time series is large (110). Thus, the data does not support these more complex models.

the importance of timely semiconductor forecasts in the early detection of future trends and in adequately responding on an organizational level. Furthermore, Chow and Choy (2006) stress the importance of semiconductor market forecasts as a macroeconomic indicator and Aubry and Renou-Maissant (2014) note the discrepancy between the importance of semiconductor market forecasting and the number of academic studies in the field.

A premier provider of such detailed semiconductor market data and forecasts is the World Semiconductor Trade Statistics (WSTS)[5]. Their data is utilized as the basis for many of the contributions in the domain of semiconductor cycle prediction (see, for example, Aubry and Renou-Maissant (2014, 2013); Liu and Chyi (2006); Liu (2005)). Furthermore, WSTS is an important provider of market data and forecasts to the semiconductor industry. In fact, their market numbers are based on the input of member companies, and their forecasts are based on industry expert estimates.

Despite the importance of WSTS' market forecasts and their common use in the industry and by industry analysts, no evaluation of the accuracy of these forecasts has been conducted to our knowledge. Furthermore, while progress has been made in explaining and predicting the semiconductor cycle, this field typically focuses on the total semiconductor market. On the other end of the granularity spectrum, researchers have made valuable contributions to semiconductor demand and sales forecasting, typically focusing on specific products or product groups of one or several specific semiconductor companies. To our knowledge, no attempt has been made to provide detailed data-driven multi-granularity forecasts for the semiconductor market.

This motivates our comparative analysis of semiconductor market forecasts in the second Article.

Article 2: Steinmeister, L., & Pauly, M. (2025). Human vs. Machines: Who wins in semiconductor market forecasting? Expert Systems with Applications, 263(October 2024), 125719. doi.org/10.1016/j.eswa.2024.125719

The main objective of this study was the evaluation of WSTS expert forecasts through a comparison with various data-driven forecasts across all 110 product categories. We wanted to find out whether data-driven models might enhance expert forecasts even in a complex market and with short time series despite extensive proprietary information available to the experts, which was unavailable to the data-driven models. We show that WSTS provides a strong baseline but find that the expert forecasts can, indeed, be enhanced by incorporating the latest available market data into the models. This

[5]Webpage: wsts.org

finding holds even for short time series, where experts are typically believed to be more accurate (Hyndman and Athanasopoulos, 2018). In the process, we also find out that multi-granularity forecasting of the semiconductor market with data-driven models is feasible and we provide valuable guidance for future practitioners by sharing accuracy measures for each model and each product WSTS category.

An aspect that is neglected in this work is the coherence of the forecasts across hierarchical levels. Coherence describes the desired property of hierarchical forecasts: subcategory forecasts should sum to the respective higher-level forecast. If business line A expects s_A in sales and sells products AA and AB, then their respective sales forecasts s_{AA} and s_{AB} should sum up to s_A, i.e. $s_A = s_{AA} + s_{AB}$. This is not automatically the case but is desired for coherent planning across organizational levels. Wickramasuriya et al. (2019) define forecast reconciliation as "the process of adjusting forecasts to make them coherent." The easiest way to obtain coherent forecasts is to only forecast on the lowest level and to obtain respective higher-level forecasts by appropriate summation. This reconciliation method is known as the bottom-up approach. However, this approach ignores information from higher hierarchical levels, which are often observed to have a preferable signal-to-noise ratio (Grunfeld and Griliches, 1960). Therefore, the bottom-up approach is not optimal (Hyndman and Athanasopoulos, 2018; Athanasopoulos et al., 2023). Wickramasuriya et al. (2019) introduce a novel approach that aims to find an optimal coherent forecast by minimizing the sum of forecast variances and, thus, utilizing all hierarchical levels. Accordingly, they call the method trace minimization (MinT).

MinT is implemented in the R packages `hts` (Hyndman et al., 2021) and `fable` (O'Hara-Wild et al., 2023). Similar Python implementations exist with the `pyhts` and `HierarchicalForecast` libraries (Olivares et al., 2022). However, the interface of the R implementations implicitly assumes that the hierarchical structures are non-degenerate. As illustrated in Figure 1.1, a degenerate hierarchical structure does not have equal depth across all possible branches. In our example, node B in the hierarchy in Figure 1.1 b) does not have any subnodes whereas node A has two subnodes, AA and AB.

However, in practice, such degenerate hierarchical time series structures do exist – the WSTS product categorization being one. Yet the `hts` and `fable` packages are not able to handle them directly and without introducing bias. Therefore, we provide an adapted implementation of the `hts` package[6], the `htsDegenerateR` package, in the third Article.

[6]We mistakenly called R packages "libraries" our articles. Here, we correctly refer to them as "packages." This naming convention is different for other programming languages such as Python.

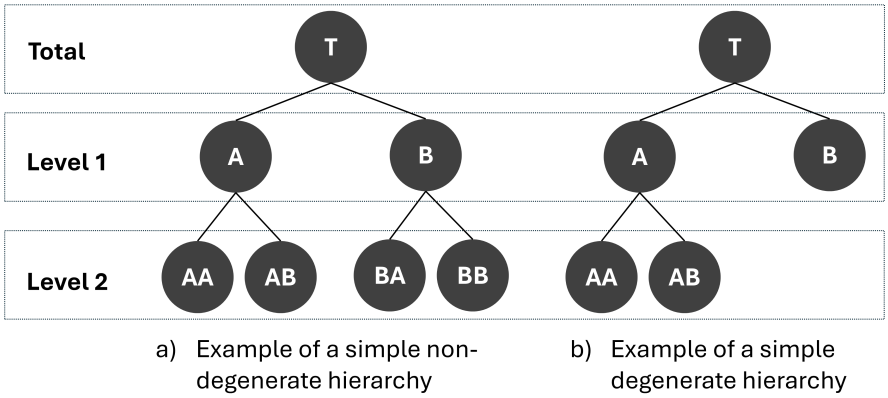


Figure 1.1: Illustration of an example non-degenerate and an example degenerate hierarchical structure.

Article 3: Steinmeister, L., & Pauly, M. (2024). Degenerate Hierarchical Time Series Reconciliation With The Minimum Trace Algorithm in R. In D. Herberger & M. Hübner (Eds.), Proceedings of the Conference on Production Systems and Logistics: CPSL. publish-Ing. doi.org/10.15488/17729

The adapted algorithm is demonstrated to potential users with a small and reproducible demo using simulated time series. Additionally, the paper provides a small case study at the hand of WSTS' power transistor hierarchy consisting of two hierarchical levels and a total of eight time series.

However, with 110 product categories, the complete WSTS product categorization category we considered is significantly larger. Furthermore, some of the time series are very short – in the extreme case only encompassing 24 observations. To derive coherent forecasts with MinT reconciliation necessitates the estimation of an in-sample residual variance matrix. In the case of the WSTS product categorization hierarchy, this implies the estimation of 6105 parameters – with only 24 observations! This motivates our inquiry into the performance of MinT with very short time series and the evaluation of a proposed iterative alternative, which requires far fewer parameters, in the fourth Article.

Article 4: Steinmeister, L., Pauly, M., 2024, Iterative Trace Minimization for the Reconciliation of Very Short Hierarchical Time Series. doi.org/10.48550/arXiv.2409.18550

This paper conducts an extensive simulation study, similar to Wickramasuriya et al. (2019), to assess the performance of MinT with very short time series across various

scenarios and benchmark the performance of the proposed iterative method.

The rest of this thesis is structured as follows: the next chapter gives an overview of the employed methods, including the logistic growth model in Section 2.1, a few fundamentals of time series analysis in Section 2.2, an overview of used time series forecasting methods – including their evaluation – in Section 2.3, and important methods and results of hierarchical time series reconciliation in Section 2.4. Chapter 3 summarizes the four articles and Chapter 4 provides discussions of the findings.

2 Statistical Methods

2.1 The Logistic Growth Model

Over the centuries, academics have grappled with models describing population growth. A reasonable assumption is growth proportional to the population size, i.e.

$$\frac{\partial Y(t)}{\partial t} = rY(t), \tag{2.1}$$

where $Y(t)$ is the population at time t and r is the growth rate. For example, a scenario where every individual is replaced by two offspring would be a discrete version of this assumption. One probably does not need a background in ordinary differential equations to realize that this results in exponential growth. In fact, the solution to Equation (2.1) is given by

$$Y(t) = Y_0 e^{rt}, \tag{2.2}$$

with an initial population Y_0. However, we do not observe sustained exponential population growth. Usually, counteracting forces such as scarcity of food result in slowing growth. This effect is captured by the logistic growth model introduced by Verhulst (1838), which is described by the logistic differential equation:

$$\frac{\partial Y(t)}{\partial t} = rY(t)\left(1 - \frac{Y(t)}{M}\right), \tag{2.3}$$

where M represents some carrying capacity. Both the exponential growth model and the logistic growth model exhibit (near) exponential growth when the population is "small". However, in contrast to Equation (2.1), where growth increasingly accelerates, a growth dynamic as modeled by Equation (2.3) approaches zero as the population approaches the carrying capacity. In summary, logistic growth is characterized by two factors: 1) self-proportionality like exponential growth (the $rY(t)$-part of Equation (2.3))

and 2) cost of complexity (the $(1 - Y(t)/M)$-part). With some notational rewriting for statistical convenience and the introduction of a vertical shift parameter C, the solution of Equation (2.3) is given by

$$Y(X) = \frac{M}{1 + e^{aX+b}} + C, \tag{2.4}$$

where a captures the growth rate and b captures the initial population size. X represents the independent variable, which is often time. Equation (2.4) results in the typical S-shaped curve seen in Figure 2.1. As illustrated, the inflection point of this curve is at $-b/a$, where the maximum growth of $aM/4$ is achieved.

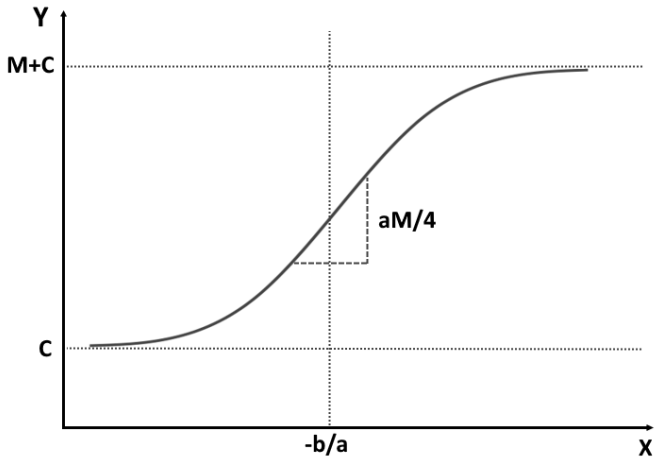


Figure 2.1: Illustration of a logistic curve and its parameters, similar to Figure 3 of Steinmeister et al. (2023).

Aside from its popularity for describing growth dynamics, such as in Zwietering et al. (1990); Berry et al. (1984), several authors have noted that these growth patterns apply in disciplines outside the ecological study of population dynamics as well (Modis, 1994; Bejan and Lorente, 2012). Some domains in which the logistic growth model, also commonly referred to as S-curve or sigmoidal curve, has been applied include country GDP growth (Kwasnicki, 2013; Modis, 2013), estimating the spread of COVID-19 (Wu et al., 2020; Shen, 2020; Aviv-Sharon and Aharoni, 2020), technology performance analysis (Nieto et al., 1998), and modeling the adoption and technological progress in renewable energies (Schilling and Esmundo, 2009). The logistic growth model has also spawned a large number of extensions, commonly referred to as technology diffusion models (Sarkar, 1998; Geroski, 2000), the most famous one probably being the Bass

model introduced by Bass (1969). Kapur et al. (2019); Chen and Chien (2018); Chien et al. (2010) provide examples of the application of technology diffusion models in the semiconductor industry. There are several estimation methods for the logistic growth model, which are covered in the following.

2.1.1 Non-linear Least Squares Estimation

Least squares estimation arises in the context of solving over-determined equations, particularly in the domain of curve and data-fitting (Strutz, 2011). As the name suggests, least squares estimation aims to estimate a parameter vector $\hat{\beta}$ characterizing a family of considered functions $\{f_\beta\}_{\beta\in\mathbb{R}^p}$ for a given set of N data points $\{y_i, x_i\}_{i=1,...,N}$ such that the sum of squared errors is minimized:

$$\hat{\beta} = \underset{\beta\in\mathbb{R}^p}{\arg\min} \sum_{i=1}^{N} \left(y_i - f_\beta(x_1)\right)^2, \tag{2.5}$$

where x_i are observation of the independent variable and y_i are observations of the dependent (target) variable. In the case where $\{f_\beta\}_{\beta\in\mathbb{R}^p}$ is a family of linear functions, this problem reduces to an ordinary least squares problem and has a closed form solution:

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y, \tag{2.6}$$

where $\mathbf{X} = (x_1, ..., x_N)^T$ is the matrix containing all observations of the independent variable in its rows and $Y = (y_1, ..., y_N)^T$ is a vector of the observations of the dependent variable. For more details, see (Christensen, 2002).

However, as Oliver (1964) notes, the estimation of the logistic growth model does not fall into this category: the unknown parameter M cannot be separated from the other parameters a and b. However, the quantity in Equation (2.5) can still be minimized by any mathematical optimization algorithm, such as the Newton method or quasi-Newton methods such as the BFGS algorithm (Wright and Nocedal, 1999). It should be noted that minimizing the sum of squared residuals has some desirable statistical properties: in the case of distributions belonging to the regular exponential family (like the normal distribution), the (standardized) least squares estimate is the maximum likelihood estimate under mild conditions (Charnes et al., 1976).

2.1.2 χ^2-Quantity Minimization

According to Modis (2007), the best way to fit an S-curve is by minimizing a chi-square quantity:

$$U(\beta) = \sum_{i=1}^{N} \left(\frac{y_i - E[Y_i]}{\sigma(Y_i)} \right)^2, \tag{2.7}$$

where $E[Y_i]$ is the expected "population" and $\sigma^2(Y_i)$ is the variance of the "population" at time x_i. Debecker and Modis (1994) assume that Y_i are discrete quantities which are limited by an upper bound, the "carrying capacity", M. They argue that Y_i obeys a binomial distribution $Binom(M, \zeta_{t_i})$, where

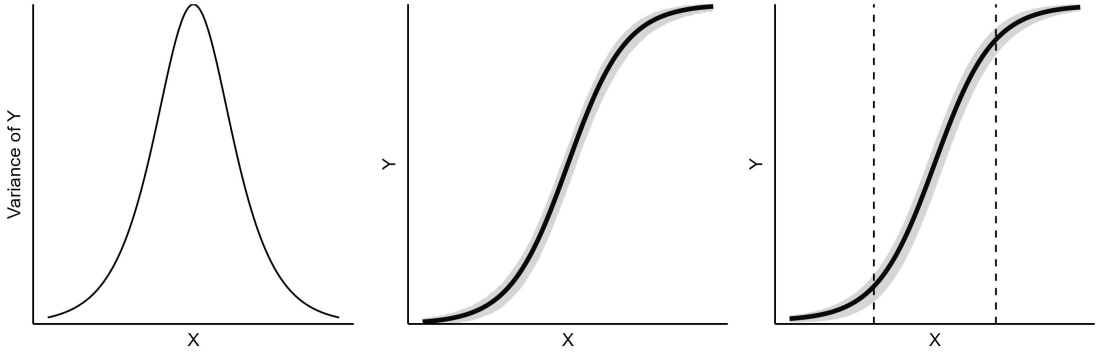
$$\zeta_{x_i} = \frac{E[Y_i]}{M} = \frac{1}{1 + e^{ax_i + b}}$$

is the expected niche occupation fraction at time x_i. Thus, the "population" variance (illustrated in 2.2a) is given by

$$\sigma^2(Y_i) = \frac{M}{(1 + e^{ax_i+b})(1 + e^{-(ax_i+b)})}.$$

The S-curve with resulting pointwise binomial confidence bands is illustrated in Figure 2.2b.

If M is large and $0 \ll Y_i \ll M$, then by employing the central limit theorem, Y_i can be approximated with a normal distribution. Assuming that all $0 \ll y_i \ll M$, for all $i = 1, ..., N$, $U(\beta)$ then approximately follows a χ^2-distribution with $N - 3$ degrees of freedom. To ensure that this approximation is "accurate",Debecker and Modis (1994) suggest ensuring that all y_i are within 10% and 90% of the range of the S-curve. The S-curve with normal-approximation pointwise confidence bounds is illustrated in Figure 2.2c. In contrast to the binomial confidence bounds in Figure 2.2b, the normal confidence bounds stay symmetric, highlighting a visible deviation from the binomial distribution outside the 10-90% range.



(a) S-Curve Variance (b) Binomial S-Curve (c) Normal S-Curve

Figure 2.2: Figure (a) illustrates the variance, as in Section 2.1.2, over the progression of the S-curve. Figure (b) illustrates a logistic curve with 99% pointwise confidence band under the binomial distribution of Debecker and Modis (1994) while (c) illustrates the same logistic curve with 95% pointwise confidence band under the normal distribution approximation with the 10% and 90% cut-offs (the vertical dashed lines) suggested in Debecker and Modis (1994). $M = 300$ was used to generate these figures.

Substituting the logistic curve family into Equation (2.7), we obtain the minimization problem

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\arg\min} U(\beta) = \underset{\beta \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{N} \left(\frac{y_i - E[Y_i]}{\sigma(Y_i)} \right)^2 \tag{2.8}$$

$$= \underset{\beta \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{N} \left(\frac{y_i - f_\beta(x_i)}{\sigma(f_\beta(x_i))} \right)^2 . \tag{2.9}$$

Note that Equation (2.9) is very similar to Equation (2.5). The difference lies in weighing the individual errors with the inverse of their standard deviation. Thus, under the distributional assumptions of Debecker and Modis (1994), this method is equivalent to standardized least squares estimation as described in Charnes et al. (1976).

A clear advantage of this method is the incorporation of an expectation regarding the variance along the S-curve into the minimization problem of Equation (2.9). Novel technologies, like small initial populations, are not expected to exhibit strong variation; neither are technologies that near their limitations at the end of their life cycle. Note, however, that, while incorporating an expectation regarding residual variance, Equation (2.9) does not make use of the skew of residuals expected particularly at the tail ends of the S-curve, as seen in Figure 2.2b.

2.1.3 OLS Variance Minimization

Restating the S-curve model allows the incorporation of expected residual skew at the tail ends of the S-curve (upward skew at the beginning and downward skew at the end) into the parameter estimation process. Let

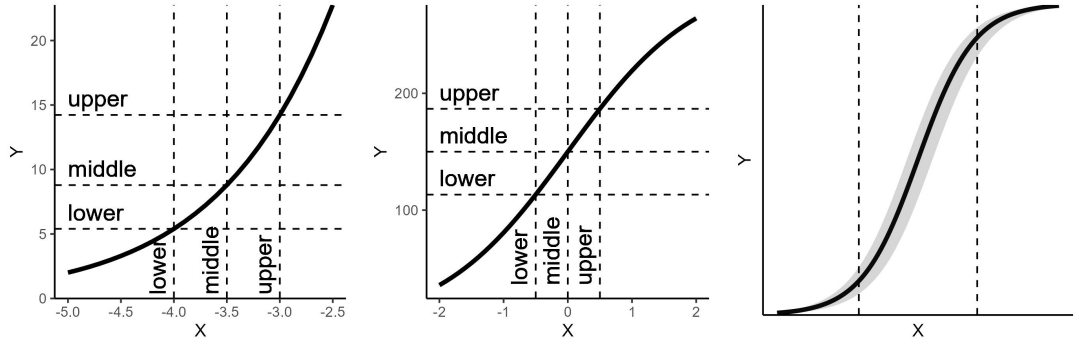
$$Y(X) = \frac{M}{1 + e^{aX+b+\epsilon}} + C, \tag{2.10}$$

with an error term ϵ. For the observations $\{y_i, x_i\}_{i=1,\dots,N}$, we obtain

$$y_i = \frac{M}{1 + \exp(ax_i + b + \epsilon_i)}, \tag{2.11}$$

where $\epsilon_i \overset{i.i.d.}{\sim} N(0, \sigma)$ are independent and identically distributed error term.

This formulation inherently integrates a skew of S-curve residuals on the tail ends of the curve, as illustrated in Figure 2.3a and Figure 2.3b. The corresponding pointwise confidence band of the S-curve is illustrated in Figure 2.3c.



(a) Asymmetric behavior of S-Curve residuals on the left side as implied by Equation (2.10).

(b) Symmetric behavior of S-Curve residuals in the middle as implied by Equation (2.10).

(c) S-Curve with respective pointwise 95% confidence band based on Equation (2.10)

Figure 2.3: Figures (a) and (b) illustrate the behavior of residuals following the model specified in Equation (2.10). Note how residuals on the left tail skew upwards whereas residuals in the center of the curve are near symmetric. Figure (c) illustrates the OLS S-curve with the corresponding pointwise confidence band.

Equation (2.11) can be rewritten to

$$\tilde{y_i} := \log\left(\frac{M - y_i}{y_i}\right) = ax_i + b + \epsilon_i. \tag{2.12}$$

In other words, if M is known we essentially have an ordinary linear model after a suitable transformation. The problem, of course, is that M is not known but rather a significant parameter of interest. The parameters can be determined iteratively, by computing standard OLS estimates for the parameters a and b given some value for M. M can then be updated by minimizing the OLS residual variance, $\mathrm{var}(\epsilon)$, as outlined in Algorithm 1.

Algorithm 1: Iterative OLS fit

Data: Given observations $(X, Y) = \{(x_i, y_i)_{i=1,...,N}\}$, initial value $M_0 > \max_{i=1,...,N} y_i$, error tolerance $\kappa > 0$

Return: Model parameters a, b, M

Procedure OLSResVar*(X,Y,M)*
  let $\tilde{Y} = \log\left(\frac{M-Y}{Y}\right)$
  let $\mathcal{X} = [1, X]$ the linear model design matrix.
  let $\hat{R} = \tilde{Y} - \mathcal{X}\left(\mathcal{X}^T\mathcal{X}\right)^{-1}\mathcal{X}^T\tilde{Y}$
  return $\left(\frac{1}{N-2}\hat{R}^T\hat{R}\right)$
end
let $M = \text{minimize}(\text{OLSResVar}, \text{initial Parameter} = M_0)$
let $\tilde{Y} = \log\left(\frac{M-Y}{Y}\right)$
let $(b, a) = \left(\mathcal{X}^T\mathcal{X}\right)^{-1}\mathcal{X}^T\tilde{Y}$

Supplying exact derivatives of the loss function E (ResVar) instead of using automatic numerical differentiation may substantially enhance computation time and result in better convergence properties due to reduced numerical error. With simple calculus, the following derivatives can be obtained:

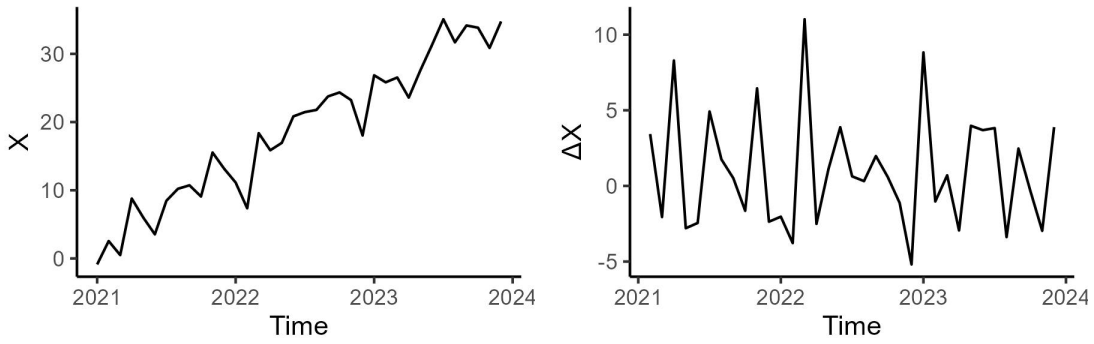
$$\frac{\partial E}{\partial M} = \frac{2}{N-2}\left(\log\left(\frac{M[\mathbf{1}] - Y}{Y}\right) - b\mathbf{1} - aX\right)^T \frac{1}{M\mathbf{1} - Y}, \tag{2.13}$$

where $\mathbf{1} = [1, ..., 1]^T$.

2.2 A Very Brief Introduction to Time Series Analysis

The goal of this section is to establish a minimum of time series analysis basics up to the ARIMA process. This is instrumental in discussing time series forecasting, particularly using the ARIMA model, and introduces some useful terminology. Most of this section is based on Brockwell and Davis (2002), which provides a good introduction to time series analysis. It is also a good reference for further details on the content covered in this section. A deeper and more mathematical treatment is also available in Brockwell and Davis (1991).

Time series are the mathematical framework to deal with data that is observed over time. For mathematical modeling, it is assumed that this data, $\{x_t\}_{t\in T}$, are the realization of a stochastic process $\{X_t\}_{t\in T}$, where X_t are random variables on some probability space $(\Omega, \mathcal{F}, P)$. In the following, the index set, T, which refers to the time over which the time series is modeled, is assumed to be a set of discrete time steps, i.e. $T \subset \mathbb{N}$. A simple time series ranging from Jan 2021 to Dec 2023 is illustrated in Figure 2.4.



(a) Time Series with Linear Trend

(b) Differenced Time Series

Figure 2.4: Illustration of a simple time series $X_t = t + z_t$, where $z_t \overset{i.i.d.}{\sim} N(0, 3)$ (here $t = 0$ corresponds to Jan 2023) and the corresponding time series of differences, which eliminates the trend.

There are several important properties of time series, which we define and discuss in the following.

Definition 1. *Given a time series $X = \{X_t\}_{t\in T}$ with finite second moments, then, the **mean function** of X is defined as $\mu_X(t) = E[X_t]$ and the **covariance function** of X is defined as $\gamma_X(r, s) = \mathrm{cov}(X_r, X_s)$.*

Definition 2. *A time series, X, is called* ***stationary*** *if*

(i) μ_X *is constant w.r.t.* t

(ii) $\gamma_X(t+h,t) = \gamma_X(h)$ *is independent of* t *for any* h.

Stationarity is an important and often desired feature of time series. It implies that the time series exhibits a certain degree of stability which is often desired for time series forecasting. When dealing with non-stationary time series, it is common procedure to consider its differences instead.

Definition 3. *The* ***lag-***d ***differencing operator*** *is defined as*

$$\nabla_d X_t = (1 - B^d)X_t = X_t - X_{t-d},$$

where B is the backward shift operator $BX_t = X_{t-1}$ and $B^j X_t = X_{t-j}$.

Consider the time series illustrated in Figure 2.4a. It clearly exhibits a linear trend and thus it is not stationary. Indeed the mean function clearly depends on t: $\mu_X(t) = t$. However, (simple) differencing removes the linear trend and transforms this time series into a stationary one in Figure 2.4b. Differencing with higher lags is often useful in the presence of seasonality. In this case the lag is usually chosen equal to the period of the seasonality.

Definition 4. *A time series, $W = \{W_t\}$, is called* ***white noise process*** *if it is a sequence of uncorrelated random variables with equal variance σ^2 and zero mean and we write*

$$W \sim \mathrm{WN}(0, \sigma^2).$$

White noise processes are useful to model error terms in common time series models. Note that random variables from a white noise process are not necessarily $i.i.d.$ (this is a special case when all W_t follow a normal distribution). A time series with $i.i.d.$ variables with zero mean and variance σ^2 is commonly referred to as an $IID(0, \sigma^2)$ process. The assumption of independence and identical distribution are much stronger than the assumptions of a white noise process. The assumption of error terms following a white noise process suffices for our purposes.

Definition 5. *A time series $X = \{X_t\}$ is called an* ***ARMA****(p, q)* ***process (autoregressive moving average process) if*** *X is stationary and there are polynomials $\phi(z) = 1 - \phi_1 z -$*

... $- \phi_p z^p$ and $\theta(z) = 1 + \theta_1 z + ... + \theta_q z^q$ with no common factors such that

$$\phi(B)X_t = \theta(B)Z_t,$$

where $Z = \{Z_t\} \sim \mathrm{WN}(0, \sigma^2)$ is a suitable white noise process.

Thus, an ARMA process consists of a component that depends solely on its prior realizations, the autoregressive part, and a moving average of random "innovations", Z. It is commonly known that an ARMA process X has a unique stationary solution if and only if $\phi(z) \neq 0$ for any $|z| = 1$.

Definition 6. *Let X be an ARMA(p, q) process. It is called **causal** or a **causal function of** Z if there are constants $\{\psi_i\}_{i \in \mathcal{N}}$ with $\sum_{i=0}^{\infty} |\psi_i| < \infty$ such that*

$$X_t = \sum_{i=0}^{\infty} \psi Z_{t-i}.$$

In other words, causality describes ARMA processes which can be expressed in a way that they only depend on past "innovations" – a condition that is often desirable for practical purposes. With this, we get to the ARIMA process:

Definition 7. *$X = \{X_t\}$ is called an **ARIMA**(p, d, q) **process** if $Y_t = (1 - B)^d X_t$ is a causal ARMA(p, q) process for $d \in \mathbb{N}_0$.*

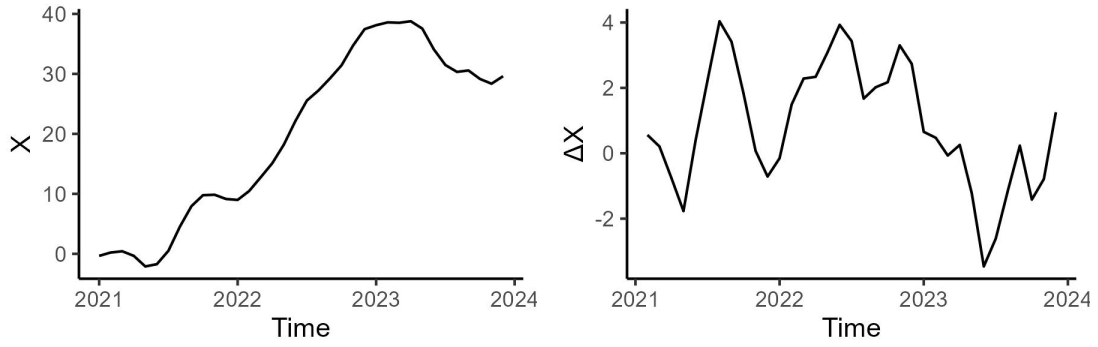
Note that an ARIMA process with $d = 0$ is simply a causal ARMA process. Further, the linear time series in Figure 2.4 is an ARIMA$(0, 1, 0)$ process. A realization of an ARIMA$(1, 1, 1)$ process is illustrated in Figure 2.5.

However, real time series often exhibit seasonality. A seasonal extension of the ARIMA model is given by the seasonal ARIMA (SARIMA) model.

Definition 8. *X is called a **seasonal ARIMA**$(p, d, q) \times (P, D, Q)_s$ **process with period** s for $d, D \in \mathbb{N}_0$ if the differenced time series given by $Y_t = (1 - B)^d (1 - B^s)^D X_t$ is a causal ARMA process that fulfills*

$$\phi(B)\Phi(B^s)Y_t = \theta(B)\Theta(B^s)Z_z,$$

for a white noise process Z and polynomials Φ of degree P and Θ of degree Q.



(a) Realization of an ARIMA$(1, 1, 1)$ Process (b) Differenced Time Series

Figure 2.5: Figure (a) illustrates a time series generated by an ARIMA$(1, 1, 1)$ process and Figure (b) shows the corresponding differenced time series.

Brockwell and Davis (2002) note that D is usually either one or zero and P and Q are rarely larger than three. This is very helpful for forecasting applications as this observation helps to constrain the hyper-parameter space significantly.

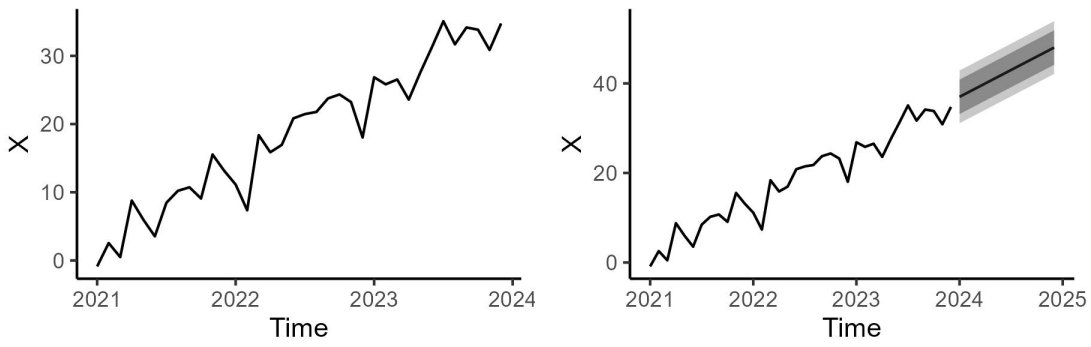
2.3 Time Series Forecasting

Time series forecasting describes the process of systematically making predictions, often utilizing a model and historical data. This section explores several statistical and machine learning methods to accomplish this task and explores measures to assess the quality of such forecasts. An example is presented in Figure 2.6.

2.3.1 Statistical Forecasting Methods

ARIMA

The ARIMA and SARIMA models described in Definition 6 and Definition 8 can not only be used to describe a class of time series or simulate time series data but also for deriving forecasts. Given a time series, X, of interest and the hyper-parameters p, d, and q and possibly P, D, and Q, this requires the estimation of the polynomial parameters. After appropriate differencing, this can be accomplished via several methods, such as the Innovations Algorithm presented in Brockwell and Davis (2002) or likelihood estimation detailed in Shumway and Stoffer (2017).



(a) Time Series with Linear Trend (b) Forecast with Prediction Bands

Figure 2.6: Illustration of a simple time series $X_t = t + z_t$, where $z_t \overset{i.i.d.}{\sim} N(0, 3)$ (here $t = 0$ corresponds to Jan 2023) in Figure (a). Forecasts are easily derived, especially when the linear trend is known, as is seen in Figure (b). Pointwise prediction bands are computed based on the distribution of the error terms.

However, the hyper-parameters are typically not known. Traditionally, statisticians would consult autocorrelation plots and partial autocorrelation plots to determine appropriate values. However, this process is often perceived as subjective and there has been an effort to automate this search, such as described by Hyndman and Khandakar (2008) and implemented in the `auto.arima` function of the `forecast` package in R.

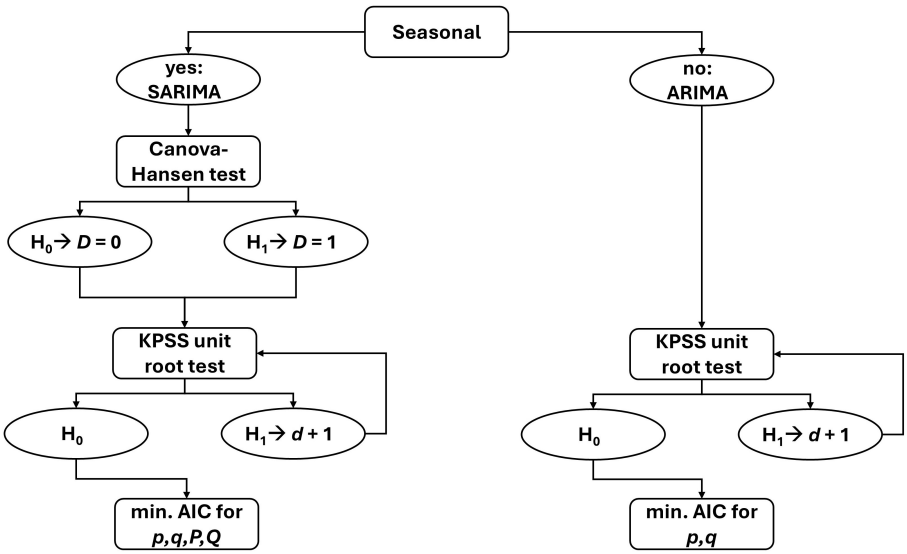


Figure 2.7: Illustration of the procedure to determine the hyper-parameters of the `auto.arima` function as described by Hyndman and Khandakar (2008).

The `auto.arima` function works as illustrated in Figure 2.7: If the time series is seasonal, Hyndman and Khandakar (2008) propose to identify $D \in \{0, 1\}$ by employing the Canova-Hansen test (Canova and Hansen, 1995). A rejection of the null hypothesis (H_0) that seasonal patterns are stable over time leads to the choice $D = 1$. If the test fails to reject H_0, D is chosen as 0.

In the next step (or if the time series is not seasonal), the differencing parameter d is chosen via an iterative approach. d is initially assumed to be zero and the KPSS unit-root test (H_0 : no unit-roots vs. H_1 there is a unit-root) is conducted (Kwiatkowski et al., 1992). If H_0 is rejected, then d is increased and the time series is differenced. The procedure is repeated until the test fails to reject the null hypothesis.

The remaining parameters p, q (and possibly P and Q) are determined by minimizing the Akaike information criterion (AIC), see Hastie et al. (2009) for example:

$$AIC = -\log(L) + 2(p + q + P + Q + k),$$

where k is linked to a constant, c, introduced to the characteristic equation if $d + D < 2$. If this constant is not zero, the model effectively contains an extra parameter and $k = 1$. The complete process is illustrated in Figure 2.7.

Simple Exponential Smoothing

Since their introduction in the 1950s (Gardner, 1985), exponential smoothing models have been popular for their simplicity, computational efficiency, and their ability to achieve high predictive performance despite this (Hyndman, 2001; Satchell and Timmermann, 1995).

Simple exponential smoothing (SES) only requires the estimation of two parameters: an initial value $\hat{X}_\mathbf{0}$ and the smoothing constant α. With these, forecasts can be computed successively as

$$\hat{X}_{t+1} = (1 - \alpha)\, \hat{X}_t + \alpha X_t, \tag{2.14}$$

where $\hat{X}_{t+1}$ denotes the one-step forecast at time $t + 1$. SES has performed well on the M3 and M5 forecasting competitions (Makridakis et al., 2018, 2022) and is implemented in the `forecast` package.

Error, Trend, and Seasonality

Error, Trend, and Seasonality (ETS) can be viewed similarly to `auto.arima` in the sense that the model automatically searches for an ideal exponential smoothing model, possibly encompassing trend and seasonal components. It searches models with 1) no trend, 2) an additive trend, 3) a dampened additive trend, 4) a multiplicative trend, 5) a dampened multiplicative trend and a) no seasonality, b) an additive seasonality, and c) a multiplicative seasonality. Thus, there are a total of 15 model types (30 with additive and multiplicative errors). The simplest one is SES (no trend or seasonality). The additive trend and seasonality model is also known as the additive Holt-Winters' model (Holt, 2004; Winters, 1960).

These methods are primarily concerned with deriving point forecasts. However, there are stochastic models – the innovation state space models – with which these algorithms can be derived and which also provide frameworks for computing prediction intervals. For more details, see Hyndman et al. (2008).

Hyndman and Khandakar (2008) propose a two step solution to finding the optimal exponential smoothing model:

1) Use maximum likelihood estimation to determine the parameters of the appropriate models.

2) Use the AIC to select the best model.

Hyndman et al. (2002) demonstrated that this procedure yielded good forecasts on the M-competition (Makridakis et al., 1982) and M3-competition data (Makridakis and Hibon, 2000), particularly for short term forecasts. Similarly, ETS has performed strongly in a comparison study on the M3-competition data conducted by Makridakis et al. (2018).

2.3.2 Machine Learning Forecasting Methods

Similar to the previous section, we explore several models which can be used to obtain forecasts. However, we first give some general background to clarify the training and hyper-parameter optimization methodology.

An additional approach to time series forecasting is to view it as a supervised learning problem: we are given a training set $\mathcal{T} = (x_i, y_i)_{i=1,\dots,N}$ where x_i are the covariates, also known as observed variables or inputs, and y_i are the labels or outputs. Assume a model such as $Y = f(X) + \epsilon$. Using the training data $\mathcal{T}$ and given a risk or loss function

$\mathcal{L}(X, Y)$ a supervised learning algorithm estimates a function $\hat{f}$ to approximate f by minimizing the loss. Often the loss is given by the mean squared error (MSE) defined by

$$\mathcal{L}_{MSE}(X, Y) = E[(Y - \hat{f}(X))^2].$$

Often, $\hat{f}$ is chosen from a parameterized family of functions $(f_\theta)_{\theta in \Theta}$ as was the case with the logistic growth model in Section 2.1. In this case, the choice of $\hat{f}$ reduces to a corresponding choice of $\hat{\theta}$, which can often be obtained through the minimization of the loss function on the training data over the parameter set Θ. But sometimes this parameter set or the particular family (f_θ) depends on a set of hyper-parameters. We have seen how hyper-parameters can be chosen in the case of (S)ARIMA (Section 2.3.1) and ETS (Section 2.3.1).

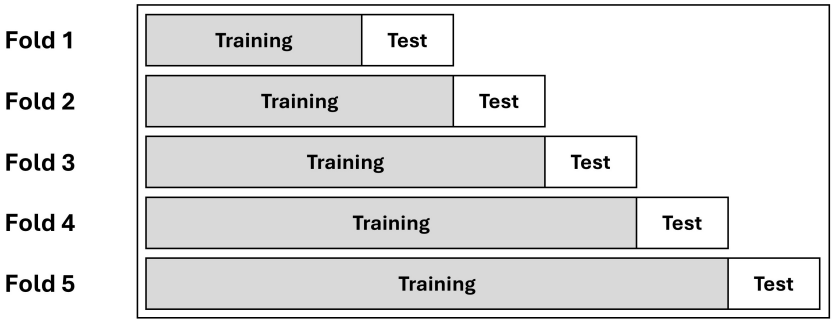


Figure 2.8: Illustration of 5-fold time series cross-validation.

A general way to select hyper-parameters is by minimizing an estimate for out-of-sample error (or loss). Such out-of-sample loss can be estimated through (time series) cross-validation, as illustrated in Figure 2.8. After fitting several models with different sets of hyper-parameters, these are successively evaluated on a hold-out set. The loss estimated over all the hold-out sets gives an estimate of how the method might perform in the future. Then, one simply chooses the set of hyper-parameters that minimizes the out-of-sample loss – see, for example, Kuhn (2008).

However, obtaining the sets of hyper-parameters to test can be a challenge – particularly when these constitute continuous quantities. Often, it is not possible or computationally feasible to try all possible hyper-parameter combinations. A simple but standard procedure to obtain hyper-parameter combinations is grid search, as illustrated in Figure 2.9. As the name suggests, a grid is constructed from values A, B, and C for the first hyper-parameter and a, b, and c for the second (note that grid search does not require an equal number of values for each hyper-parameter). These are often chosen equidistant but don't have to. A problem of this method is that the complexity increases quickly

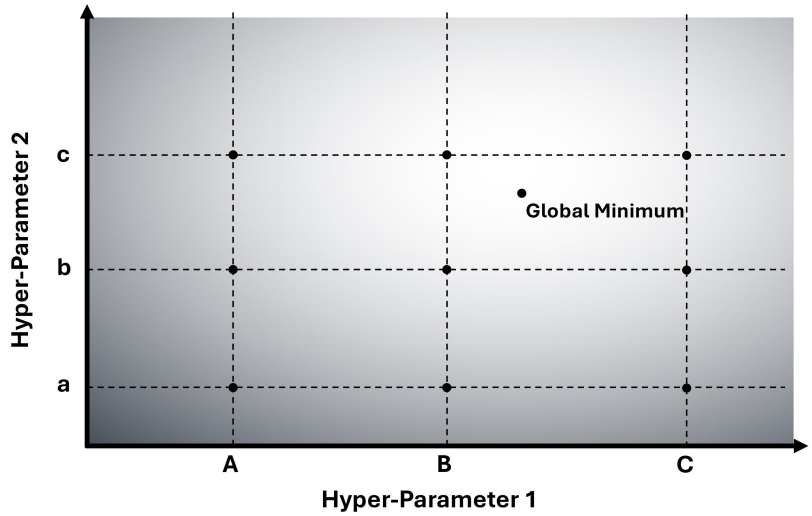


Figure 2.9: Illustration of grid search hyper-parameter optimization.

as the number of hyper-parameters increases and the grid points may not be optimal. Nevertheless, we use this method due to its simplicity and because the training of the considered models was relatively fast. Also, note that the computation can also easily be parallelized. Nevertheless, it is worth mentioning that methods such as Bayesian optimization (Snoek et al., 2012) may be preferable in some situations – particularly when model training is very resource-consuming.

Random Forest

Random Forest (RF) as introduced by Breiman (2001) is an ensemble machine learning algorithm that combines multiple Categorization and Regression Trees (CART) into a single model. The CART algorithm creates a binary tree where each node represents a split-decision based on a single feature chosen to minimize some impurity measure for categorical outcomes or a measure such as the MSE for continuous outcomes (Breiman et al., 1984). An example decision tree trained by the CART algorithm is illustrated in Figure 2.10 (created with the `rpart` package in R).

Random Forest combines multiple such trees through a process called Bootstrap Aggregating (bagging). Bagging in Random Forest works by generating several datasets from the original dataset by drawing with replacement (the distribution of the "bootstrapped" datasets thus follows the empirical distribution of the original data). Additionally, the set of features considered for a split is chosen randomly at each node.

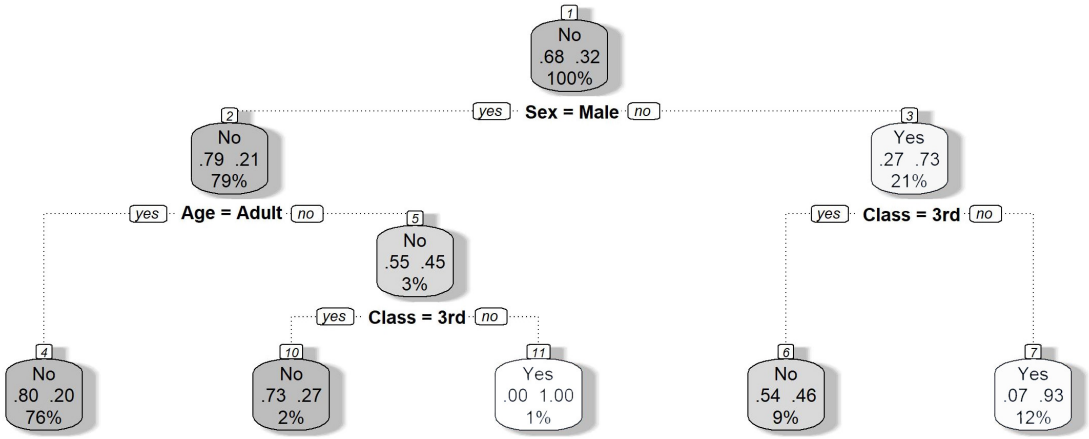


Figure 2.10: Illustration of a decision tree trained with the `rpart` R package on the Titanic dataset.

This introduces several hyper-parameters:

`ntree` The number of trees to grow in the forest (default of the `randomForest` implementation in R is 500).

`mtry` The number of sampled features to try at each split (default of the `randomForest` implementation in R is $p/3$ for regression and $\sqrt{p}$ for classification, where p is the total number of features)

`nodesize` The minimum number of observations contained at terminal nodes. Smaller numbers may lead to overfitting.

`maxnodes` The maximum number of terminal nodes. Larger values may lead to overfitting.

`sampsize` The size of the "bootstrapped" datasets (typically equal to the size of the original dataset).

We primarily focus on `mtry` and `nodesize` in hyper-parameter tuning. We also consider both the `variance` and split-rules (more on this in the next section). For more details on the hyper-parameters of Random Forest, see Probst et al. (2019).

Extremely Randomized Trees

Extremely Randomized Trees, also known as ExtraTrees (ET) and introduced by Geurts et al. (2006), can be viewed as a variation on Random Forest. In contrast to Random

Forest, where splits are optimally chosen based on a criterion such as MSE, ExtraTrees selects random split points within each feature subset. Furthermore, bagging is optional with the ExtraTrees algorithm, though we always use bagging. ExtraTrees is implemented in R with the `ranger` package by setting the split-rule to `extratrees`.

Gaussian Process Regression

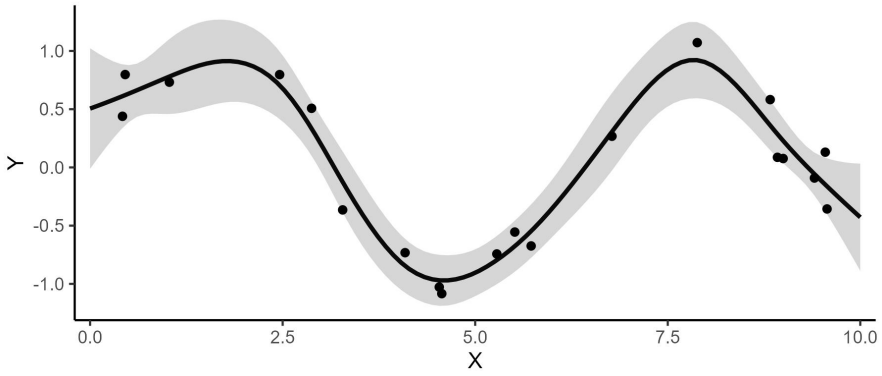


Figure 2.11: Illustration of Gaussian process regression. Data points were generated from a sine curve with normal errors added. The curve represents the fitted curve with a Gaussian kernel. The grey area is a pointwise confidence band.

A Gaussian Process assumes that observations follow a joint Normal distribution defined by

- a mean function $m(X)$, which represents the expected output for an input x, and
- a kernel or covariance function $K(X, X)$, which describes the relationship between outputs for different inputs

and we write $Y = f(X) \sim \mathcal{GP}(m(X), K(X, X))$. Gaussian Process regression (GPR) is a Bayesian approach, with the prior being the Gaussian Process distribution. Training data is used to calculate the posterior distribution over the function space, which can then be used to make predictions for a new input x^*:

$$m(x^*) = K(x^*, X)K(X, X)^{-1}Y$$
$$\sigma^2(x^*) = K(x^*, x^*) - K(x^*, X)K(X, X)^{-1}K(X, x^*).$$

Then, $Y^* \sim N(mu(x^*), \sigma^2(x^*))$ and the point prediction for x^* is $EY^* = m(x^*)$. Figure 2.11 illustrates a Gaussian process regression fit and Wang (2023) provides a

tutorial with further details. Implementations of GPR are provided by the `kernlab` R package (Karatzoglou et al., 2004).

K-Nearest Neighbors

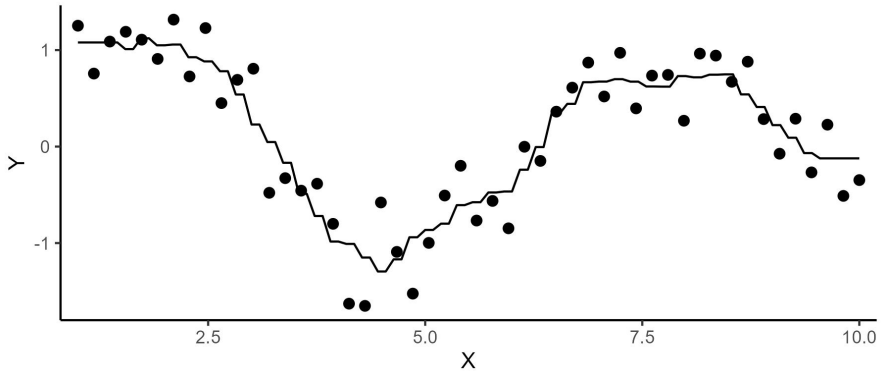


Figure 2.12: Illustration of K-NN regression. Data points were generated from a sine curve with normal errors added. The curve represents the K-NN regression, with $k = 5$.

K-Nearest Neighbors (K-NN) Regression is a simple non-parametric learning model. Predictions are based on the average outputs of the k nearest data points: Desired is a predictor $\hat{y}$ for a new observation x^* and we are given training data $(x_j, y_j)_{j=1,\dots,N}$, then

$$\hat{y} = \frac{1}{k} \sum_{j \in S_k(x^*)} y_j,$$

where x^* are the covariates for which a prediction is desired and $S_k(x^*)$ is the index set of the k closest observations around x^*.

Additionally, a kernel function $K(\cdot, \cdot)$ can be introduced for weighing. Predictions are then calculated as

$$\hat{y} = \sum_{j \in S_k(x^*)} \frac{K(x_j, x^*)}{\sum_{l \in S_k(x^*)} K(x_l, x^*)} y_j.$$

K-NN regression is implemented in the `kknn` R package (Schliep and Hechenbichler, 2016), which has an option to automatically choose the optimal kernel. We tune the hyper-parameters `k` and `distance`, which defines the distance measure to use.

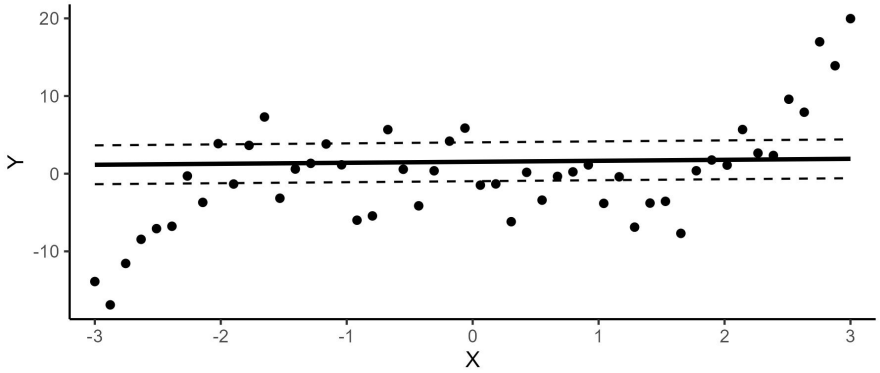


Figure 2.13: Illustration of Support Vector Regression (SVR) with a linear kernel. The drawn curve represents the fitted SVR model and the dashed lines illustrate the ϵ-tube margins.

Support Vector Regression

Support Vector Regression (SVR) was introduced by Drucker et al. (1996) as a regression analog to the classical Support Vector Machines used for classification (Cortes and Vapnik, 1995). Instead of basing the fit on all data points, observations within some ϵ-tube are ignored (similar to how the vectors that are not support vectors are ignored in the classification case). This is often argued to make the regression more robust because SVR does not fit the finer patterns within this ϵ-tube but "focuses on the bigger picture," see Hastie et al. (2009); Awad and Khanna (2015).

Additionally, kernels can be used to capture non-linear relationships. A common kernel is the radial or Gaussian kernel $K(x_1, x_2) = \exp(-\gamma|x_1 - x_2|^2)$. SVR with a linear kernel contains one hyper-parameter C, the cost parameter, which controls the penalization of residuals falling outside the ϵ-tube. Higher values make the model more sensitive to data points outside the ϵ-tube resulting in a tighter fit which may come with risks of overfitting. The ϵ parameter is often left at a default value of $\epsilon = 0.1$, such as set in the `e1071` (Meyer et al., 2023) and the `kernlab` R packages (Karatzoglou et al., 2004). Additional hyper-parameters may be introduced through a different choice of kernel, such as the radial kernel with γ, which is also sometimes parameterized with σ, where $\gamma = \frac{1}{2\sigma}$.

2.3.3 Forecast Evaluation Methods

Assessing the accuracy of produced forecasts plays an important role in model selection and has implications for subsequent planning and goal-setting.

Hyndman and Koehler (2006) provide a detailed discussion of forecast accuracy measures including the mean squared error (MSE), the mean absolute percentage error (MAPE), the mean absolute error (MAE), and the mean absolute standardized error (MASE) defined as

$$MSE = \frac{1}{n} \sum_{i=1,...,n} (\hat{y}_i - y_i)^2, \tag{2.15}$$

$$MAPE = \frac{1}{n} \sum_{i=1,...,n} \left| \frac{\hat{y}_i - y_i}{y_i} \right|, \tag{2.16}$$

$$MAE = \frac{1}{n} \sum_{i=1,...,n} |\hat{y}_i - y_i|, \tag{2.17}$$

$$MASE = \frac{1}{n} \sum_{i=1,...,n} \frac{|\hat{y}_i - y_i|}{\frac{1}{n-1} \sum_{j=2,...,n} |y_j - y_{j-1}|}, \tag{2.18}$$

where $\hat{y}_i$ are predictions with corresponding observations y_i for $i = 1, ..., n$. These can be computed after a model is fit on the training data, but this usually leads to an overestimation of accuracy or an underestimation of error and can be solved through out-of-sample evaluation (Hastie et al., 2009). Cerqueira et al. (2020) explore several such performance evaluation methods for time series forecasting within an elaborate simulation study and Hewamalage et al. (2023) aim to provide guidance. They point out that standard evaluation methods such as k-fold cross-validation may not be adequate as they do not preserve temporal dependencies; something that can be achieved by methods such as time series cross-validation (tsCV) explored in Section 2.3.2 in the context of hyper-parameter tuning. In some cases, this yields a nested tsCV setup, where an outer tsCV is used for evaluation, and within each outer training step, an inner tsCV is performed to select the optimal hyper-parameters.

2.4 Hierarchical Time Series Reconciliation

So far, we have mostly talked about single time series. However, time series sometimes occur with a hierarchical order, where subordinate time series sum to the respective

higher time series, such as the 2-level hierarchy illustrated in Figure 2.14: in this example, the time series labeled AA and AB sum up to time series A, etc.

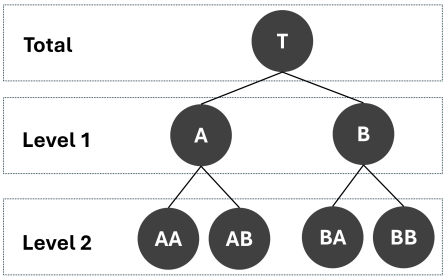


Figure 2.14: Illustration of a small time series hierarchy.

For example, sales time series may be split up according to sales channel and product type as in Kuhlmann et al. (2024); or time series of semiconductor market volumes might be categorized according to product categories as is the case with WSTS forecasts and data.

When standard forecasting methods are applied to the individual time series, this hierarchical structure is usually not reflected in the resulting forecasts, which we call base forecasts. However, there are two main reasons why coherent forecasts, i.e. forecasts that reflect the hierarchical structure, are desirable: 1) the hierarchy constraint reflects knowledge we have about the nature of the time series, and incorporating this information into forecasts may increase forecast accuracy (Wickramasuriya et al., 2019), and 2) coherent forecasts allow for coherent interpretation, planning, and goal setting (Hyndman and Athanasopoulos, 2018).

The hierarchical structure can also be expressed in terms of a summation or structure matrix $\mathbf{S}$. Equation (2.19) provides the structure matrix for the example hierarchy in Figure 2.14.

$$\mathbf{S} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.19}$$

There are several reconciliation methods to ensure forecast coherence. The most straightforward solution is to only forecast on the lowest level and to obtain higher-level forecasts through aggregation respective to the hierarchical structure. This reconciliation method is also called the **bottom-up (BU) approach**. In matrix form, it can be written as

$$\tilde{Y} = \mathbf{S}\mathbf{G}_{BU}\hat{Y}, \tag{2.20}$$

where $\tilde{Y}$ is the vector of reconciled forecasts, $\hat{Y}$ is the vector of base forecasts, and $\mathbf{G}_{BU}$ is a matrix that maps the base forecasts onto the space of reconciled bottom-level forecasts. In fact, Hyndman et al. (2011) showed that any existing reconciliation method can be expressed by an appropriate $\mathbf{G}$-matrix. For the BU approach in our example hierarchy, this matrix is given by $\mathbf{G}_{BU} = [\mathbf{0}_{3\times3}, \mathbf{I}_{4\times4}]$, where $\mathbf{0}_{3\times3}$ is the three by three matrix only containing zeros (all higher-level forecasts are ignored), and $\mathbf{I}_{4\times4}$ is the four-dimensional identity matrix (all bottom-level forecasts are taken as they are).

Hyndman et al. (2011) identify a criterion for obtaining unbiased reconciled forecasts, given that the base forecasts are unbiased – a property that is often desirable. Their main theoretical result can be summarized as follows.

Theorem 1. *Let $\hat{Y}_{T+h}$ be a vector of unbiased base forecasts with horizon h for a hierarchical time series $\{Y_t\}_{t=1,\dots,T}$ with structure matrix $\mathbf{S}$. The corresponding reconciled forecasts $\tilde{Y}_{T+h}$ for a reconciliation method expressed by the corresponding $\mathbf{G}$-matrix are unbiased if and only if $\mathbf{SGS} = \mathbf{S}$.*

Proof. Let E_T be the expectation with regard to the probability measure conditioned on the observations up until time T. The proof directly follows from

$$\begin{aligned} E_T\left[\tilde{Y}_{T+h}\right] &= E_T\left[\mathbf{S}\mathbf{G}\hat{Y}_{T+h}\right] = \mathbf{S}\mathbf{G}E_T\left[\hat{Y}_{T+h}\right] \\ &\overset{\text{unbiased}}{=} \mathbf{S}\mathbf{G}E_T\left[Y_{T+h}\right] = \mathbf{S}\mathbf{G}\mathbf{S}E_T\left[Y^{(\text{bottom})}_{T+h}\right], \end{aligned}$$

where $Y^{(\text{bottom})}_{T+h}$ are the (unknown) values of the bottom-level time series at time $T+h$. □

From Theorem 1, it directly follows that forecasts reconciled with the BU approach are unbiased if the bottom-level forecasts are unbiased. However, the BU approach ignores information from higher levels of aggregation, which may be problematic for highly disaggregated hierarchical structures as Wickramasuriya et al. (2019) note.

Another simple approach requiring even fewer forecasts is the **top-down (TD) approach**. In this case, only the top-level forecast is considered and then broken down to the

lower levels. See Athanasopoulos et al. (2009) for some top-down strategies. All of these variants can be expressed with a $\mathbf{G}$-matrix of the form $\mathbf{G} = [\mathbf{p}, \mathbf{0}_{m_b \times m}]$, where $\mathbf{p} = [p_1, ..., p_{m_b}]^T$ is a vector of proportions of the bottom-level time series summing to one and m_b and m are the number of bottom-level time series and total time series respectively. This means that TD-reconciled forecasts can never be unbiased even when all base forecasts are unbiased, as is shown in Theorem 2.

Theorem 2. *Let* $\mathbf{S}$ *be a structure matrix and* $\mathbf{G} = [\mathbf{p}, \mathbf{0}_{m_b \times m}]$ *be a corresponding* $\mathbf{G}$*-matrix characterizing a TD approach, where* $\mathbf{p} = [p_1, ..., p_{m_b}]^T$ *is a vector of proportions summing to one. Then the resulting reconciled forecasts are biased.*

Proof. We show that $\mathbf{SGS} \neq \mathbf{S}$. We have that $\mathbf{SGS} = \mathbf{S}[\mathbf{p}, ..., \mathbf{p}]$. Since $\mathbf{p}$ is a vector summing to one, the first row of $\mathbf{SGS}$ only contains ones, as does $\mathbf{S}$. However, from the second row of $\mathbf{S}$ on, there should be at least one zero and at least one one. Let $\mathcal{A}_j$ with $j > 1$ be the set of column indices for which the j-th row contains ones. Let $s_j := \sum_{k \in \mathcal{A}_j} p_k$. Then the j-th row of $\mathbf{SGS}$ is given by $[s_j, ..., s_j]$. Thus, it has to be different from the j-th row of S, which means that $\mathbf{SGS} \neq \mathbf{S}$. □

This motivates the search by Hyndman et al. (2011) for "optimal combination forecasts," which uses the information from all time series to derive optimal reconciled forecasts. The main idea revolves around decomposing the base forecasts $\hat{Y}_{T+h}$ into an expected value and an error term:

$$\hat{Y}_{T+h} = E_T \left[Y_{T+h} \right] + \epsilon_h, \tag{2.21}$$

where ϵ_h is a vector of errors for the various time series. Due to the hierarchical structure, this expectation can be expressed in terms of the bottom time series

$$\hat{Y}_{T+h} = E_T \left[\mathbf{S} Y_{T+h}^{(\text{bottom})} \right] + \epsilon_h = \mathbf{S} E_T \left[Y_{T+h}^{(\text{bottom})} \right] + \epsilon_h = \mathbf{S}\beta_h + \epsilon_h, \tag{2.22}$$

where $\beta_h := E_T \left[Y_{T+h}^{(\text{bottom})} \right]$. Now, Equation (2.22) resembles a weighted least squares problem, where the unknown vector β_h takes the role of the linear model parameters. Consequently, we can estimate β_h as

$$\hat{\beta}_h = \left(\mathbf{S}^T \mathbf{\Sigma}_h^{-1} \mathbf{S} \right)^{-1} \mathbf{S}^T \mathbf{\Sigma}_h^{-1} \hat{Y}_{T+h}, \tag{2.23}$$

where $\mathbf{\Sigma}_h^{-1}$ is the (generalized) inverse of the covariance matrix $\mathbf{\Sigma}_h$ of the forecast errors ϵ_h. However, $\mathbf{\Sigma}_h$ is neither known nor identifiable (Wickramasuriya et al., 2019).

Depending on the assumptions regarding $\mathbf{\Sigma}_h$, different reconciliation methods can be derived. For example, Hyndman et al. (2011) show the following result.

Theorem 3. *Let $\mathbf{\Sigma}_h = S\mathbf{\Omega}_h S^T$, where $\mathbf{\Omega} = \text{cov}(\epsilon_h^{(bottom)})$ – an assumption yielded by the approximation that higher level residuals are approximately the sum of their subsumed residuals, i.e. $\epsilon_h \approx \mathbf{S}\epsilon_h^{(bottom)}$. Then, the reconciled forecasts are given by*

$$\hat{\beta}_h = \left(\mathbf{S}^T\mathbf{S}\right)^{-1}\mathbf{S}^T\hat{Y}_{T+h}. \tag{2.24}$$

Proof. Note that $\mathbf{\Sigma}_h$ is singular. Hence, the Moore-Penrose inverse of $\mathbf{\Sigma}_h$ is used. As Hyndman et al. (2011) note in their proof of their Theorem 1, this is given by

$$\mathbf{\Sigma}_h^+ = \mathbf{S}\left(\mathbf{S}^T\mathbf{S}\right)^{-1}\left(\mathbf{\Omega}_h^T\mathbf{S}^T\mathbf{S}\mathbf{\Omega}_h\right)^{-1}\mathbf{\Omega}_h^T\mathbf{S}^T. \tag{2.25}$$

Note, that $\mathbf{\Omega}_h$ is symmetric. Further, $\mathbf{S}^T\mathbf{S}$ and $\mathbf{\Omega}_h$ are full rank and consequently invertible. Hence, Equation (2.25) can be rewritten as

$$\mathbf{\Sigma}_h^+ = \mathbf{S}\left(\mathbf{S}^T\mathbf{S}\right)^{-1}\mathbf{\Omega}_h^{-1}\left(\mathbf{S}^T\mathbf{S}\right)^{-1}\mathbf{S}^{-1}. \tag{2.26}$$

Substituting Equation (2.26) into Equation (2.23) yields

$$\begin{aligned}
\hat{\beta}_h &= \left(\mathbf{S}^T\mathbf{S}\left(\mathbf{S}^T\mathbf{S}\right)^{-1}\mathbf{\Omega}_h^{-1}\left(\mathbf{S}^T\mathbf{S}\right)^{-1}\mathbf{S}^{-1}\mathbf{S}\right)^{-1}\mathbf{S}^T\mathbf{S}\left(\mathbf{S}^T\mathbf{S}\right)^{-1}\mathbf{\Omega}_h^{-1}\left(\mathbf{S}^T\mathbf{S}\right)^{-1}\mathbf{S}^{-1}\hat{Y}_{T+h} \\
&= \left(\mathbf{\Omega}_h^{-1}\right)^{-1}\mathbf{S}^T\mathbf{S}\left(\mathbf{S}^T\mathbf{S}\right)^{-1}\mathbf{\Omega}_h^{-1}\left(\mathbf{S}^T\mathbf{S}\right)^{-1}\mathbf{S}^{-1}\hat{Y}_{T+h} \\
&= \left(\mathbf{S}^T\mathbf{S}\right)^{-1}\mathbf{S}^{-1}\hat{Y}_{T+h},
\end{aligned}$$

which was to show. □

The reconciled forecasts stemming from Equation (2.24) in Theorem 3 are known as the **ordinary least squares (OLS) approach**. Wickramasuriya et al. (2019) show that the reconciliation forecasts described in Equation (2.24) can also be interpreted as the solution to minimizing the trace of the reconciled forecast error covariance matrix $\mathbf{V} = \text{cov}\left(Y_{T+h} - \tilde{Y}_{T+h}\right) = \mathbf{SGW}_h\mathbf{G}^T\mathbf{S}^T$, where $\mathbf{W}_h = \text{cov}\left(Y_{T+h} - \hat{Y}_{T+h}\right)$[1]. In other words, forecasts computed by

$$\tilde{Y}_{T+h} = \mathbf{S}\left(\mathbf{S}^T\mathbf{W}_h^{-1}\mathbf{S}\right)^{-1}\mathbf{W}_h^{-1}\hat{Y}_{T+h} \tag{2.27}$$

[1] In Equation (2.24), $\mathbf{\Sigma}_h$ simply changes to $\mathbf{W}_h$.

minimize the sum of errors over all time series. Therefore Wickramasuriya et al. (2019); Hyndman and Athanasopoulos (2018) refer to the estimator in Equation (2.27) as the **minimum trace (MinT) estimator**. Note that assuming $\mathbf{W}_h = k_h\mathbf{I}$, where k_h is some factor that conveniently cancels out in Equation (2.27), yields the same OLS reconciliation as in Theorem 3.

Another reconciliation method, called **structural scaling** and denoted WLS_s, is obtained by assuming $\mathbf{W}_h = k_h\text{diag}\left(\mathbf{S}(1, ..., 1)^T\right)$. Structural scaling assumes that the forecast errors are uncorrelated and proportional to the number of subsumed bottom-level time series. While this represents a strong assumption, the approach yields good results when the data is insufficient to use estimators with looser assumptions or when these cannot be computed – such as when considering temporal hierarchies (Athanasopoulos et al., 2017).

The **variance scaling** (WLS_v) reconciliation method is obtained by setting $\mathbf{W}_h = k_h\text{diag}\left(\hat{\mathbf{W}}\right)$, where $\hat{\mathbf{W}}$ is the in-sample covariance matrix of the base forecasts. This method allows for forecast variances to vary beyond the summation structure in the structural scaling approach and thus allows for stronger weighing of more accurate time series forecasts. This may be particularly useful when higher-level forecasts exhibit a higher signal-to-noise ratio, as is often observed in practice (Wickramasuriya et al., 2019).

Lastly, the **minimum trace (MinT) appraoch** is obtained by assuming $\mathbf{W}_h = k_h\hat{\mathbf{W}}$. However, when the hierarchy is large in comparison to the length of the time series, this estimate can be unstable. Therefore, Wickramasuriya et al. (2019) propose to use the shrinkage estimator proposed by Schäfer and Strimmer (2005) for $\hat{\mathbf{W}}$. Wickramasuriya et al. (2019) highlight the potential of the MinT approach to generate more accurate reconciled forecasts.

3 Summary of the Articles

3.1 Article 1: Testing The Limits: A Robustness Analysis Of Logistic Growth Models For Life Cycle Estimation During The COVID-19 Pandemic

The main objective of the paper was an analysis of the stability of life cycle estimates based on the logistic growth model in the presence of a severe external shock: the COVID-19 pandemic.

This is of particular interest given the dynamic nature of the semiconductor market, which is characterized by long lead times not only in fabrication but also in expanding fabrication capacity and corresponding capital-intensive investments (Wu and Chien, 2008; Lv et al., 2018). Additionally, product life cycles have been shortening, and semiconductor technology is rapidly advancing (Macher, 2006). This complicated the task of recognizing trends and forecasting future developments. However, missing out on an important development can mean months or years to catch up, highlighting the need for accurate forecasting to provide support to management (Chien et al., 2010; Aubry and Renou-Maissant, 2014; Cakanyildirim and Roundy, 2002). This includes the estimation of life cycles, which can provide insight into growth potentials and risk of disruption (Modis, 1994).

The logistic growth model, which is based on the logistic differential equation $\frac{dN(t)}{dt} = rN(t)\left(1 - \frac{N(t)}{K}\right)$ and stems from population growth modeling in ecology (Tsoularis and Wallace, 2002), is a simple and common framework to accomplish this task. Here, $N(t)$ denotes the population size at time t, r the growth rate, and K the carrying capacity. This model is also known as S-curve or sigmoidal curve due to the characteristic S-shape of its solution and was first introduced by (Verhulst, 1838).

However, the application did not remain confined to population growth modeling as similar growth patterns can be observed in "natural growth" scenarios, such as technology cycles and innovation diffusion, more broadly. These scenarios exhibit similar

characteristics, namely: a) self-proportional growth and b) a progressive slowdown, i.e. costs of complexity. Rewriting the solution of the logistic growth model and introducing a horizontal offset (C), we obtain

$$Y(X) = \frac{M}{1 + \exp(aX + b)} + C, \tag{3.1}$$

where M is the carrying capacity, technological limitation, or maximum cumulative market size, $Y(X)$ is the response variable (in place of the population size $N(t)$, and a corresponds to the growth rate, with the maximum growth rate of $aM/4$ being reached at the inflection point at $x = -b/a$. Once the parameters are determined, life cycle estimates are easily obtained as $\frac{\max(Y)-C}{M-C}$.

This model assumes that the market or technology operates as a closed system – an assumption that is flawed in any realistic use case as the COVID-19 pandemic or the Russian invasion of Ukraine have shown. And even without these external disruptions, the industry is heavily interconnected upstream and downstream in the supply chain.

As a case study, this paper examined an extreme economic disruption, namely the COVID-19 pandemic, to analyze the sensitivity of logistic growth model-derived life cycle estimates to violations of the closed-system assumption.

To this end, revenues for 18 level 1 technology groups and 10 level 2 technology groups of a higher aggregation level were obtained on a quarterly basis ranging from October 2006 to January 2023. Logistic model fits were evaluated with residual plots and validated by an industry expert at Infineon to verify that the model appropriately mirrored the market dynamics for the respective technology grouping. Mismatches can, for example, occur when demand for a technology is driven by a few (or single) large customers. In such cases, customer-specific demand and contract structures may outweigh market dynamics and the model may not be applicable. This eliminated seven level 2 technology groups and four level 1 technology groups, leaving eleven and six respectively.

For these remaining technology groups, life cycle estimates were obtained 1) before the onset of the COVID-19 pandemic and 2) based on the complete dataset (Oct. 2006 - Jan. 2023). The differences in life cycle estimates were then analyzed.

Overall, we found that life cycle estimates were robust and differences were relatively slim. For level 1 technology groups, life cycle estimates increased slightly, which would be expected given the passing of roughly three years. For level 2 technologies, life cycle estimates decreased slightly. This could be attributed to the higher aggregation level containing more infant technologies which were excluded due to a lack of sufficient data

on level 1. It is possible that these infant technologies weighed more heavily as their markets developed over the passing 3 years, pushing life cycle estimates slightly down.

Some authors suggest incorporating external variables into the logistic growth model, for example, by modeling the upper bound dynamically (Seidl and Tisdell, 1999; Calatayud et al., 2020). This could further improve robustness and help to model some of the complexities of real-world applications. However, this was not examined in this paper and is an opportunity for further inquiry.

Additionally, we only considered a single event and the effects on only a single industry at the example of relatively few technology groups. To draw definite conclusions, similar studies should be performed for other industries and other external effects of varying impact.

Given the data we have considered, logistic growth models remain an important tool in assessing technology and product life cycles.

3.2 Article 2: Human Vs. Machines: Who Wins In Semiconductor Market Forecasting?

Both semiconductor cycle prediction and semiconductor demand and sales forecasting have received attention from academics in recent years. Semiconductor cycle prediction studies the cyclical behavior of the broader semiconductor market and identifies indicators and explanatory variables. Examples of recent studies include Aubry and Renou-Maissant (2014, 2013); Chow and Choy (2006); Liu and Chyi (2006); Liu (2005). On the other end of the granularity spectrum, semiconductor demand, and sales forecasting focuses on providing manufacturing intelligence for specific product groups and to a limited number of semiconductor companies, see, e.g., Wang and Chen (2019); Kapur et al. (2019); Chen and Chien (2018); Xu and Sharma (2017); Chien et al. (2010); Aytac and Wu (2013); Chow and Choy (2006).

However, no one, to our knowledge, has tried to forecast the whole semiconductor market with its different levels of granularity simultaneously using data-driven methods. Forecasts of this kind play an important role in strategic decision-making and internal benchmarking, such as the calculation of market shares. Analysts and industry practitioners often rely on providers, such as the World Semiconductor Trade Statistics (WSTS)[1], for such forecasts (Nagao, 2019; Corder et al., 2024; Simons, 2024). WSTS consolidates

[1] wsts.org

forecasts of industry experts to produce its market forecasts. But despite their regular use, they have not been systematically benchmarked to our knowledge.

The evaluation of the expert forecasts provided by WSTS through a comparative study was the main objective of this paper. To this end, three research hypotheses were formed:

(H1) Expert forecasts exhibit higher accuracy compared to autoregressive data-driven models.

We assumed this hypothesis because the industry experts have a wealth of proprietary information available, such as the order book or the sentiment of important customers, which the autoregressive models did not have access to. Additionally, WSTS meetings take place in the middle of the quarter, which means that the experts already have access to the results of the first half of the quarter when they are expected to share their prognosis. Luckily, WSTS publishes the results for the first month of the quarter at roughly the same time and this information is always available when the forecasts are published. This motivated the second hypothesis:

(H2) The incorporation of additional autoregressive information in data-driven forecasts enhances their competitiveness against expert forecasts.

The last hypothesis we investigated pertained to time series lengths. Due to the dynamic nature of the semiconductor market, some product categories have been introduced much more recently than others. This means that some time series are very short. As a general rule, this should be a situation in which experts excel (Hyndman and Athanasopoulos, 2018). This is what the third hypothesis aims to investigate:

(H3) Expert forecasts outperform data-driven forecasts, particularly in the context of short time series.

The setup for testing these hypotheses included several statistical and machine learning models, which served as benchmarks: 1) Seasonal Autoregressive Integrated Moving Average (SARIMA), 2) Simple Exponential Smoothing (SES), 3) Error, Trend, and Seasonality (ETS), 4) Random Forest (RF), 5) Extremely Randomized Trees (ET), 6) Gaussian Process Regression (GPR), 7) K-Nearest Neighbors (KNN), 8) Support Vector Regression (SVR), and 9) a simple ensemble of these methods. These models were chosen either for their popularity in machine learning circles or for their performance on extensive studies using the M3-Competition dataset (Hyndman, 2020; Makridakis et al., 2018), which is comparable in type[2] and length to the dataset studied here. More

[2]The M3-Competition dataset includes many quarterly and monthly time series stemming from various domains, including microeconomics, industry, macroeconomics, and finance.

complex methods such as boosting or deep learning models were not used because of short time series lengths.

The dataset consisted of monthly semiconductor market data across a total of 110 product categories. New expert forecasts were available every second quarter and on a quarterly basis. The longest time series ranged from Jan 1991 to Aug 2023 while the shortest time series started in Jan 2016.

Each model was trained on monthly data, and hyper-parameters were optimized via grid search with cross-validation except in the cases of SARIMA and ETS (implemented in the `auto.arima` and `ets` functions of the `forecast` package), which internally search for an optimal model (Hyndman and Khandakar, 2008). Out-of-sample forecasts were then computed with horizons of up to 3 months starting from Jan 2018. For comparison with WSTS' expert forecasts, the appropriate (multi-step) monthly forecasts were aggregated into quarterly forecasts. Both the data-driven forecasts and the expert forecasts were then evaluated using 1) the mean squared error (MSE), 2) the mean absolute error (MAE), and the mean absolute percentage error (MAPE).

To investigate hypothesis (H1), 3-month ahead forecasts were evaluated as described. Despite GPR, ETS, and SARIMA yielding the strongest results among the data-driven methods, they still exceeded the average error measures of the expert forecasts by 9% to 61% depending on the model and error measure. Thus, the experts delivered more accurate than the purely data-driven methods on average. However, we also found that WSTS' forecast updates, which are posted between the half-yearly meetings at which the expert forecasts are consolidated, can be significantly enhanced with data-driven methods in terms of MSE and MAE.

For hypothesis (H2), the quarterly forecasts were calculated by adding the result for the first month as supplied by WSTS and adding the corresponding 2-month ahead forecasts. Again, the highest-performing models were SARIMA, GPR, and ETS. These showed clear accuracy gains ranging from 25% to 39%, indicating that expert forecasts can be enhanced by incorporating the most recent data into data-driven forecasts.

Lastly, the product groups were divided according to the available historical data to investigate hypothesis (H3). This yielded three categorizations: 1) long, 2) medium, and 3) short time series. These contained 50, 31, and 29 time series with minimum lengths of 392, 296, and 92 months respectively. However, even for short time series, the data-driven methods produced superior forecasts. The best-performing models were ETS, GPR, and SES with average improvements ranging from 23% to 43%. A notable difference is which models performed best. While SES was among the three most accurate models

for short time series, SARIMA, GPR, and ETS achieved the highest accuracy for longer time series. Thus, even forecasts for short time series can be improved with data-driven methods.

In conclusion, this study has contributed by providing an extensive evaluation of the accuracy of expert semiconductor market forecasts provided by WSTS. It showed that these forecasts can often be enhanced with data-driven methods. In doing so, it has also shown that comprehensive multi-granularity forecasts of the semiconductor market with data-driven methods are feasible and it provides practitioners guidance for modeling by providing detailed out-of-sample error measures for all compared models.

3.3 Article 3: Degenerate Hierarchical Time Series Reconciliation With The Minimum Trace Algorithm in R

Hierarchical time series, where observations are made at varying levels of granularity, occur in many real-world scenarios: sales are often internally reported by product and sales channel (Kuhlmann et al., 2024), technology classifications are often hierarchically arranged Steinmeister et al. (2023), and various data is often geographically arranged and aggregated by successively larger regions such as in the Australian tourist dataset (Athanasopoulos et al., 2009) – to name a few.

This raises a significant challenge in forecasting such time series: how can coherence between forecasts on different hierarchical levels be ensured? This is often important for consistent operational or logistical planning. The most naïve method is to forecast on the lowest hierarchical level and then aggregate these forecasts accordingly. However, this bottom-up approach is often not optimal as the information of all the higher-level time series is ignored. To make matters worse, these higher-level time series often exhibit better signal-to-noise ratios than less aggregated time series (Grunfeld and Griliches, 1960).

(Wickramasuriya et al., 2019) suggest an optimal reconciliation approach, trace minimization (MinT), that is designed to consider the forecasts on all hierarchical levels and produce coherent forecasts with minimal error: it minimizes the trace of the covariance matrix of the reconciled forecasts. (Wickramasuriya et al., 2019) show that this approach is superior to other reconciliation methods in a wide array of scenarios and (Hyndman et al., 2021) provide a readily available implementation of the algorithm with the `hts` package for R.

Unfortunately, the provided implementation has an interface that limits the application of the algorithm to specific scenarios. Hierarchical time series forecasts with degenerate hierarchical structures, for example, cannot be directly reconciled due to this restriction.

The interface of the `hts` package assumes non-degenerate hierarchical structures, where the number of hierarchical levels is constant across branches whereas degenerate hierarchical structures may have differing hierarchical depths as illustrated in Figure 3.1. This can be the case in business applications as the power transistor hierarchy from the World Semiconductor Trade Statistics (WSTS)[3] illustrates.

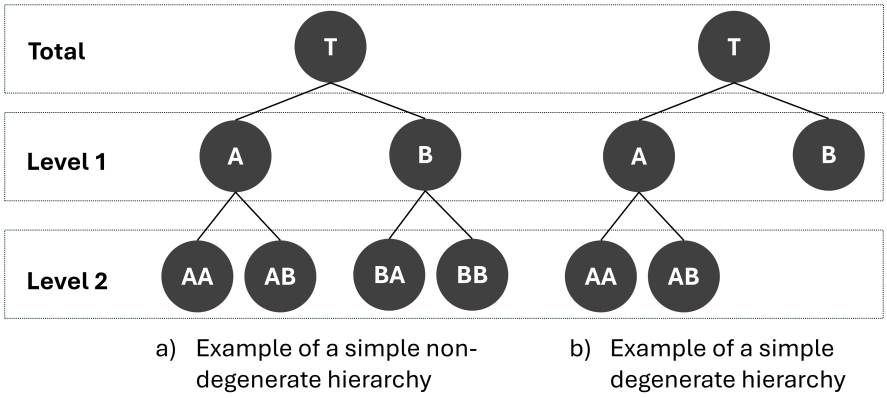


Figure 3.1: Illustration of an example non-degenerate and an example degenerate hierarchical structure similar to a Figure in Steinmeister and Pauly (2024a).

A workaround suggested by Hyndman in an online forum[4] works by duplicating the lowest-level time series of degenerate branches until the hierarchy becomes non-degenerate. As Hyndman notes, this has the obvious drawback of introducing a bias: the time series that are duplicated in the hierarchy are weighed more highly accordingly: MinT minimizes the sum of the reconciled forecast errors, and duplicating a time series implies that it is weighed proportionally to the number of duplications. Additionally, this means that the covariance matrix of reconciled forecast errors becomes singular, which presents a problem as the computation of MinT reconciliation requires an inverse of this matrix.

While these adverse effects can be offset by employing the shrinkage covariance estimator proposed by Schäfer and Strimmer (2005), these problems are not necessitated by MinT

[3] wsts.org

[4] stackoverflow.com/questions/64012331/can-fable-reconcile-hierarchical-time-series-where-the-hierarchy-has-different

itself. Therefore, this paper presents an adapted version of MinT as was originally implemented in the `hts` package and offers an interface capable of handling degenerate hierarchical structures[5].

Furthermore, the use of the package is demonstrated with a small simulated dataset (all the code is provided in the paper), and the potential for performance improvements is showcased at the hand of a case study based on the WSTS power transistor dataset.

3.4 Article 4: Iterative Trace Minimization for the Reconciliation of Very Short Hierarchical Time Series

Wickramasuriya et al. (2019) showed that MinT generally achieves a high reconciliation performance[6] across a wide array of scenarios. To obtain MinT-reconciled forecasts, the trace of the reconciled forecast error matrix is minimized. Wickramasuriya et al. (2019) show that this requires the inversion of the base forecast error matrix.

However, the inversion of this matrix is problematic when the hierarchy is large and the length of the time series is comparably short, resulting in a singular covariance estimate. In this case, a biased covariance estimate, the shrinkage estimator proposed by Schäfer and Strimmer (2005), can be utilized.

Table 3.1: Comparison of the number of parameters required to estimate the covariance matrix for simple hierarchies each with constant widths of 2 and 3 and depths ranging from 2 to 5. This is an excerpt from (Steinmeister and Pauly, 2024b)

Width	2			3		
Depth	MinT	Iterative	Fraction	MinT	Iterative	Fraction
2	28	9	0.32	91	24	0.26
3	120	21	0.18	820	78	0.10
4	496	45	0.09	7381	240	0.03
5	2016	93	0.05	66430	726	0.01

Wickramasuriya et al. (2019) test the performance of MinT with this shrinkage estimator with a simulation and a real-world dataset with a large hierarchical structure: the

[5]This adapted package can be installed via the command `install.packages("htsDegenerateR")`.

[6]Performance here is usually measured in terms of out-of-sample RMSE reduction.

Australian tourism dataset. They show that MinT is capable of outperforming other reconciliation methods in these scenarios.

However, Athanasopoulos et al. (2023) mention that MinT is often unstable in this circumstance. Therefore, this paper proposes an iterative algorithm, which greatly reduces the number of estimated parameters as illustrated in Table 3.1, and conducted a thorough simulation study similar to (Wickramasuriya et al., 2019) to examine whether it can further increase reconciliation performance in the setting of large hierarchical structures with very short time series.

Data is simulated to test the performance in the presence of various effects:

1. **Correlation**: where time series closer within the hierarchical structure are generally more strongly correlated.
2. **Smoothing**: where noise is added which cancels out during the aggregation process and thus leads to time series with a higher signal-to-noise ratio on higher hierarchical levels.
3. **Seasonality**: time series were generated to include random seasonality and trends.
4. **Differing lengths**: time series were simulated according to the first setting but with differing time series lengths. Time series were longer higher up in the hierarchy to mirror the effect of increasing granularity of categorizations over time.
5. **Degeneracy**: time series were generated according to the second (smoothing) setting but degeneracy is introduced by deleting some of the lower-level time series.
6. **Large hierarchical structure**: a large hierarchical structure with 510 total time series is simulated.

With the exception of the last scenario, time series with $T = 15, 30,$ and 60 observations are generated and the process is repeated for a total of 5000 Monte Carlo iterations. In the "large hierarchy" case, $T = 30, 60,$ and 90 observations were generated for 500 Monte Carlo iterations (in comparison, Wickramasuriya et al. (2019) used 1000 and 200 iterations respectively).

Base forecasts are produced with the Error, Trend, and Seasonality (ETS) models, the Autoregressive Integrated Moving Average (ARIMA) models as implemented via `auto.arima`, and the Gaussian Process Regression (GPR) model. The first two are implemented in the `forecast` package for R (Hyndman and Khandakar, 2008), while the `kernlab` implementation is used for GPR.

The results indicate that employing the iterative reconciliation algorithm can result in a reduction of expected forecasting error in a wide array of scenarios for the ETS and ARIMA models while MinT generally performed better for forecasts generated by the GPR model. However, it should be noted that the GPR forecasts were generally considerably worse than those from ETS and ARIMA.

An exception was the scenario with seasonality. Apparently, ARIMA and ETS had difficulty identifying both the trend and seasonal components given the short time series. The best reconciliation method for the GPR forecasts was the weighted least squares approaches (variance scaling and structural scaling), which are explained in more detail in the paper.

Another strength of the iterative algorithm, which is worth highlighting, is the presence of differing time series lengths. In this situation, the iterative reconciliation approach is capable of handling information more efficiently as it only requires local covariance estimation.

The largest improvements of the iterative algorithm compared to MinT are observed in the large hierarchy setting, where MinT slightly worsened forecast accuracy for ETS and ARIMA base forecasts while the iterative algorithm showed the largest average reduction in RMSE.

Overall, the iterative algorithms showed small to moderate performance gains in many situations in conjunction with the stronger performing ETS and ARIMA base forecasts.

Lastly, a case study at the hand of the World Semiconductor Trade Statistics (WSTS) data set with 110 product categorizations is presented in addition to the simulation study to showcase the application of the iterative reconciliation method to a real data set.

4 Discussion and Outlook

This cumulative doctoral thesis addressed challenges of forecasting within the semiconductor industry. Given the rapid technological advances, complex supply chains, the bullwhip effect, and the strategic importance of semiconductors, accurate forecasts are vital but challenging. In particular, this thesis dealt with the application, analysis, and further development of methods for forecasting the semiconductor market.

In the first paper (Article 1), we investigated the robustness of the life cycle estimates derived from the logistic growth model for several semiconductor technology groups. We found that the life cycle estimates for the technology groups that we considered were robust to the shock imposed by the COVID-19 pandemic. This analysis involved the comparison of long-term forecasts as the market capacity parameter is needed to calculate life cycles. The time gap in the data between the pre-COVID and post-COVID fits was roughly three years. Many studies utilizing diffusion models for demand and sales forecasting consider significantly shorter forecast horizons. For example, Chien et al. (2010) evaluate their model's ability to produce satisfying forecasts with only two quarters worth of data. Nevertheless, several technology groups had to be excluded from our analysis because the data exhibited clear patterns that were inconsistent with the logistic growth model. Extended diffusion models such as the model described by Chien et al. (2010) may mitigate this issue to some extent. Incorporating a multi-generation perspective (Norton and Bass, 1987) or adopting a dynamic market capacity (Calatayud et al., 2020) could lead to additional insights. Another challenge of the logistic growth model is the estimation of the market capacity when the technology is very young (remember that Debecker and Modis (1994) only use observations falling into the 10% to 90% range of the curve). In the very early stages, growth is near exponential and the signal for estimating the market capacity essentially lies in the deviation of the observations from a perfect exponential fit. Considering that observations are noisy, this yields an extremely poor signal-to-noise ratio. Possibly, a different estimate derived from the binomial distribution might allow for more accurate fits when only the first to 10% or 20% of the curve are observed.

With (Article 2), we turned our attention to concrete forecasts of the semiconductor market. While we found that previous researchers have investigated company- and product-specific sales and demand forecasts on one hand, and the industry-wide semiconductor cycle on the other hand, little research has previously been devoted to providing data-driven multi-granularity forecasts for the wider semiconductor market. Such forecasts are usually provided by market research firms and based on industry experts, such as the forecasts provided by WSTS. To our knowledge, we were the first to systematically evaluate the performance of such expert forecasts through a comparative analysis with data-driven models. We found that the evaluated expert forecasts can be enhanced with data-driven models. In the process, we showed that multi-granularity semiconductor market forecasts are feasible – even with challenges such as a dynamic market environment and, subsequently, short time series. Extending our work by integrating semiconductor cycle indicators and proprietary data could further enhance forecast accuracy. On the other hand, sales and demand forecasts might be enhanced by incorporating market forecasts for related product categories. We hope that our contribution has made a small step towards unifying semiconductor cycle prediction with semiconductor sales and demand forecasting, which might support cohesive long-term strategic planning alongside more immediate operational decision-making. Due to the short length of the time series, we were limited to simpler models and shorter forecast horizons of one quarter. As time progresses and more data becomes available, a study of longer-horizon semiconductor market forecasts might be of interest to industry practitioners.

Furthermore, we did not account for the hierarchical structure in (Article 2). While our data-driven forecasts achieved high accuracy, they were not necessarily coherent – unlike the WSTS expert forecasts. Forecast coherence can be achieved through hierarchical time series reconciliation. In fact, Trace Minimization (MinT) has shown the potential to increase forecast accuracy in addition to ensuring coherence (Wickramasuriya et al., 2019). However, several challenges arise when applying the MinT algorithm to the WSTS semiconductor market data: 1) the WSTS product categorization hierarchy is degenerate but existing MinT implementations require non-degenerate hierarchies, and 2) the time series vary in length, with some very short series, complicating internal covariance matrix estimation—particularly due to the high dimensionality of the WSTS hierarchy. We tackled the first issue in (Article 3), where we provided an adapted version of the MinT algorithm from the `hts` package (Hyndman et al., 2021) in R.

The second issue was explored in (Article 4). We investigated the performance of MinT for very short time series across several scenarios in an extensive simulation study and compared an iterative approach we proposed to drastically reduce the number of estimated

parameters. We found that MinT is remarkably robust but that the iterative algorithm has a slight edge in terms of accuracy in many of the considered scenarios, particularly when a large hierarchical structure is considered. However, this comes at a computational cost: the iterative approach often requires several iterations to converge, which requires more time and computational resources. We have not proven the convergence of the iterative approach, and we have encountered instances where it failed to converge. A potential solution might involve updating the residuals (which are used for weighing forecasts in MinT) alongside the forecasts in each iterative step. When a single forecast becomes less accurate (while maintaining a small residual variance because the initial residuals are not updated) this may amplify errors in later iterations. Adjusting the residuals in each iteration might stabilize the algorithm. However, this approach remains unexplored in our work. In addition to the simulation study, we presented the WSTS forecasts as a case study. Even though the iterative approach showed strong performance in simulations similar to the WSTS data, we were not able to replicate its strong performance in the case study. MinT, despite the challenges, resulted in the largest average accuracy gains (with the exception of the Gaussian Process Regression forecasts) and proved the most robust.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this chapter, the authors used the ChatGPT 3.5 model from OpenAI to suggest minor language edits, aiming to enhance readability. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Bibliography

Athanasopoulos, G., Ahmed, R. A., and Hyndman, R. J. (2009). Hierarchical forecasts for Australian domestic tourism. *International Journal of Forecasting*, 25(1):146–166.

Athanasopoulos, G., Hyndman, R. J., Kourentzes, N., and Panagiotelis, A. (2023). Forecast reconciliation: A review. *International Journal of Forecasting*.

Athanasopoulos, G., Hyndman, R. J., Kourentzes, N., and Petropoulos, F. (2017). Forecasting with temporal hierarchies. *European Journal of Operational Research*, 262(1):60–74.

Aubry, M. and Renou-Maissant, P. (2013). Investigating the semiconductor industry cycles. *Applied Economics*, 45(21):3058–3067.

Aubry, M. and Renou-Maissant, P. (2014). Semiconductor industry cycles: Explanatory factors and forecasting. *Economic Modelling*, 39:221–231.

Aviv-Sharon, E. and Aharoni, A. (2020). Generalized logistic growth modeling of the COVID-19 pandemic in Asia. *Infectious Disease Modelling*, 5:502–509.

Awad, M. and Khanna, R. (2015). *Efficient Learning Machines*. Apress, Berkeley, CA.

Axe, D. (2024). Bad news for Putin: Su-57 'Felon' 5G stealth fighter production is all but crippled. telegraph.co.uk/news/2024/10/09/russia-war-air-force-5g-su-57-fighter-jet-stealth. Accessed: 09 October 2024.

Aytac, B. and Wu, S. D. (2013). Characterization of demand for short life-cycle technology products. *Annals of Operations Research*, 203(1):255–277.

Bass, F. M. (1969). A new product growth for model consumer durables. *Management Science*, 15(5):215–227.

Bejan, A. and Lorente, S. (2012). The s-curves are everywhere. *Mechanical Engineering*, 134(5):44–47.

Berry, S. J., Coffey, D. S., Walsh, P. C., and Ewing, L. L. (1984). The Development of Human Benign Prostatic Hyperplasia with Age. *Journal of Urology*, 132(3):474–479.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification And Regression Trees*. Routledge.

Brockwell, P. J. and Davis, R. A. (1991). *Time Series: Theory and Methods*. Springer Series in Statistics. Springer New York, New York, NY.

Brockwell, P. J. and Davis, R. A. (2002). *Introduction to time series and forecasting*. Springer.

Cakanyildirim, M. and Roundy, R. O. (2002). SeDFAM: semiconductor demand forecast accuracy model. *IIE Transactions*, 34(5):449–465.

Calatayud, J., Cortés, J.-C., Dorini, F. A., and Jornet, M. (2020). On a stochastic logistic population model with time-varying carrying capacity. *Computational and Applied Mathematics*, 39(4):288.

Canova, F. and Hansen, B. E. (1995). Are Seasonal Patterns Constant Over Time? A Test for Seasonal Stability. *Journal of Business & Economic Statistics*, 13(3):237–252.

Cerqueira, V., Torgo, L., and Mozetič, I. (2020). Evaluating time series forecasting models: an empirical study on performance estimation methods. *Machine Learning*, 109(11):1997–2028.

Charnes, A., Frome, E. L., and Yu, P. L. (1976). The Equivalence of Generalized Least Squares and Maximum Likelihood Estimates in the Exponential Family. *Journal of the American Statistical Association*, 71(353):169–171.

Chen, Y.-J. and Chien, C.-F. (2018). An empirical study of demand forecasting of non-volatile memory for smart production of semiconductor manufacturing. *International Journal of Production Research*, 56(13):4629–4643.

Chien, C.-F., Chen, Y.-J., and Peng, J.-T. (2010). Manufacturing intelligence for semiconductor demand forecast based on technology diffusion and product life cycle. *International Journal of Production Economics*, 128(2):496–509.

Chien, C.-F. and Lin, K.-Y. (2012). Manufacturing intelligence for Hsinchu Science Park semiconductor sales prediction. *Journal of the Chinese Institute of Industrial Engineers*, 29(2):98–110.

Chow, H. K. and Choy, K. M. (2006). Forecasting the global electronics cycle with leading indicators: A Bayesian VAR approach. *International Journal of Forecasting*, 22(2):301–315.

Christensen, R. (2002). *Plane answers to complex questions*, volume 35. Springer.

Corder, G., Lafond, E., and Lorenzen, C. (2024). Worldwide semiconductor market expected to recover strongly and hit $611 billion in 2024. Technical report, European Semiconductor Industry Association. eusemiconductors.eu/sites/default/files/ESIA_WSTS_Spring_Forecast 2024.pdf. Accessed: 01 October 2024.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Debecker, A. and Modis, T. (1994). Determination of the uncertainties in S-curve logistic fits. *Technological Forecasting and Social Change*, 46(2):153–173.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A., and Vapnik, V. (1996). Support Vector Regression Machines. In Mozer, M., Jordan, M., and Petsche, T., editors, *Advances in Neural Information Processing Systems*, volume 9, pages 155–161. MIT Press.

European Comission (2023). European chips act: Online factpage. commission.europa.eu/strategy-and-policy/priorities-2019-2024/europe-fit-digital-age/european-chips-act_en. Accessed: 19 February 2024.

Foster, R. N. (1986). Working The S-Curve: Assessing Technological Threats. *Research Management*, 29(4):17–20.

Gardner, E. S. (1985). Exponential smoothing: The state of the art. *Journal of Forecasting*, 4(1):1–28.

Geary, S., Disney, S., and Towill, D. (2006). On bullwhip in supply chains—historical review, present practice and expected future impact. *International Journal of Production Economics*, 101(1):2–18.

Geroski, P. (2000). Models of technology diffusion. *Research Policy*, 29(4-5):603–625.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1):3–42.

Grunfeld, Y. and Griliches, Z. (1960). Is aggregation necessarily bad? *The Review of Economics and Statistics*, 42(1):1.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*, volume 20 of *Springer Series in Statistics*. Springer New York, New York, NY.

Hewamalage, H., Ackermann, K., and Bergmeir, C. (2023). Forecast evaluation for data scientists: common pitfalls and best practices. *Data Mining and Knowledge Discovery*, 37(2):788–832.

Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20(1):5–10.

Hyndman, R., Koehler, A., Ord, K., and Snyder, R. (2008). *Forecasting with Exponential Smoothing*. Springer Series in Statistics. Springer Berlin Heidelberg, Berlin, Heidelberg.

Hyndman, R., Lee, A., Wang, E., and Wickramasuriya, S. (2021). *hts: Hierarchical and Grouped Time Series*.

Hyndman, R. J. (2001). It's time to move from "what" to "why". *International Journal of Forecasting*, 17(4):567–570.

Hyndman, R. J. (2020). A brief history of forecasting competitions. *International Journal of Forecasting*, 36(1):7–14.

Hyndman, R. J., Ahmed, R. A., Athanasopoulos, G., and Shang, H. L. (2011). Optimal combination forecasts for hierarchical time series. *Computational Statistics and Data Analysis*, 55(9):2579–2589.

Hyndman, R. J. and Athanasopoulos, G. (2018). *Forecasting: Principles and practice*. OTexts, Melbourne, 2nd edition edition.

Hyndman, R. J. and Khandakar, Y. (2008). Automatic Time Series Forecasting: The forecast Package for R. *Journal of Statistical Software*, 27(3):1–22.

Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688.

Hyndman, R. J., Koehler, A. B., Snyder, R. D., and Grose, S. (2002). A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3):439–454.

Kapur, P., Panwar, S., and Singh, O. (2019). Modeling two-dimensional technology diffusion process under dynamic adoption rate. *Journal of Modelling in Management*, 14(3):717–737.

Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. (2004). kernlab - An S4 Package for Kernel Methods in R. *Journal of Statistical Software*, 11(9).

Kucharavy, D. and De Guio, R. (2011a). Application of S-shaped curves. *Procedia Engineering*, 9:559–572.

Kucharavy, D. and De Guio, R. (2011b). Logistic substitution model and technological forecasting. *Procedia Engineering*, 9:402–416.

Kuhlmann, L., Fesca, F., Steinmeister, L., and Pauly, M. (2024). Hierarchical Sales Forecasting In Multichannel Distribution Considering Marketing Campaigns. In Herberger, D.; Hübner, M., editor, *Proceedings of the Conference on Production Systems and Logistics: CPSL*, pages 527–538, Hannover.

Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. *Journal of Statistical Software*, 28(5).

Kwasnicki, W. (2013). Logistic growth of the global economy and competitiveness of nations. *Technological Forecasting and Social Change*, 80(1):50–76.

Kwiatkowski, D., Phillips, P. C., Schmidt, P., and Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root. *Journal of Econometrics*, 54(1-3):159–178.

Lee, H. L., Padmanabhan, V., and Whang, S. (1997). The bullwhip effect in supply chains. *MIT Sloan Management Review*, 38(3):93–102.

Liu, W.-H. (2005). Determinants of the semiconductor industry cycles. *Journal of Policy Modeling*, 27(7):853–866.

Liu, W.-H. and Chyi, Y.-L. (2006). A markov regime-switching model for the semiconductor industry cycles. *Economic Modelling*, 23(4):569–578.

Lv, S., Kim, H., Zheng, B., and Jin, H. (2018). A Review of Data Mining with Big Data towards Its Applications in the Electronics Industry. *Applied Sciences*, 8(4):582.

Macher, J. T. (2006). Technological Development and the Boundaries of the Firm: A Knowledge-Based Examination in Semiconductor Manufacturing. *Management Science*, 52(6):826–843.

Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E., and Winkler, R. (1982). The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of Forecasting*, 1(2):111–153.

Makridakis, S. and Hibon, M. (2000). The M3-Competition: results, conclusions and implications. *International Journal of Forecasting*, 16(4):451–476.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLOS ONE*, 13(3):e0194889.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 38(4):1346–1364.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., and Leisch, F. (2023). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. R package version 1.7-14.

Miller, C. (2022). *Chip War: The Fight for the World's Most Critical Technology*. Scribner.

Modis, T. (1994). Life cycles: forecasting the rise and fall of almost anything. *Futurist*, 28(5):20–25.

Modis, T. (2007). Strengths and weaknesses of S-curves. *Technological Forecasting and Social Change*, 74(6):866–872.

Modis, T. (2013). Long-term GDP forecasts and the prospects for growth. *Technological Forecasting and Social Change*, 80(8):1557–1562.

Nagao, M. (2019). The semiconductor trend from the front-end view. In *2019 Pan Pacific Microelectronics Symposium (Pan Pacific)*, pages 1–6. IEEE.

Nieto, M., Lopéz, F., and Cruz, F. (1998). Performance analysis of technology using the S curve model: The case of digital signal processing (DSP) technologies. *Technovation*, 18(6-7):439–457.

Nocetti, J. (2023). Sanctions calling: The dire prospects for russia's chips industry. *Russian Analytical Digest*, (298):2–5.

Norton, J. A. and Bass, F. M. (1987). A Diffusion Theory Model of Adoption and Substitution for Successive Generations of High-Technology Products. *Management Science*, 33(9):1069–1086.

O'Hara-Wild, M., Hyndman, R., and Wang, E. (2023). *fable: Forecasting Models for Tidy Time Series*.

Olivares, K. G., Garza, F., Luo, D., Challú, C., Mergenthaler, M., Taieb, S. B., Wickramasuriya, S. L., and Dubrawski, A. (2022). HierarchicalForecast: A Reference Framework for Hierarchical Forecasting in Python.

Oliver, F. R. (1964). Methods of Estimating the Logistic Growth Function. *Applied Statistics*, 13(2):57.

Pan, C. (2024). China pumps up state subsidies as chip war with us intensifies. scmp.com/tech/tech-war/article/3274599/tech-war-china-pumps-state-subsidies-chip-industry-counter-us-sanctions. Accessed: 10 October 2024.

Probst, P., Wright, M. N., and Boulesteix, A. L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3):e1301.

Rajan, G. (2024). OpenAI's Altman in talks to raise funds for chips, AI initiative - WSJ. reuters.com/technology/openais-altman-talks-raise-funds-chips-ai-initiative-wsj-2024-02-09. Accessed: 19 February 2024.

Richards, F. J. (1959). A Flexible Growth Function for Empirical Use. *Journal of Experimental Botany*, 10(2):290–301.

Roumeliotis, K. I. and Tselikas, N. D. (2023). Chatgpt and open-ai models: A preliminary review. *Future Internet*, 15(6):192.

Sarkar, J. (1998). Technological diffusion: Alternative theories and historical evidence. *Journal of Economic Surveys*, 12(2):131–176.

Satchell, S. and Timmermann, A. (1995). On the optimality of adaptive expectations: Muth revisited. *International Journal of Forecasting*, 11(3):407–416.

Schäfer, J. and Strimmer, K. (2005). A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(1).

Schilling, M. A. and Esmundo, M. (2009). Technology S-curves in renewable energy alternatives: Analysis and implications for industry and government. *Energy Policy*, 37(5):1767–1781.

Schliep, K. and Hechenbichler, K. (2016). *kknn: Weighted k-Nearest Neighbors*. R package version 1.3.1.

Seidl, I. and Tisdell, C. A. (1999). Carrying capacity reconsidered: from malthus' population theory to cultural carrying capacity. *Ecological Economics*, 31(3):395–408.

Shen, C. Y. (2020). Logistic growth modelling of COVID-19 proliferation in China and its international implications. *International Journal of Infectious Diseases*, 96:582–589.

Shumway, R. H. and Stoffer, D. S. (2017). *Time Series Analysis and Its Applications*. Springer Texts in Statistics. Springer International Publishing, New York.

Simons, C. (2024). 2024 semiconductor industry outlook: Trends and predictions for a cyclical industry. Technical report, Deloitte. deloitte.com/us/en/pages/technology-media-and-telecommunications/articles/semiconductor-industry-outlook.html. Accessed: 01 October 2024.

Simpson, M. J., Browning, A. P., Warne, D. J., Maclaren, O. J., and Baker, R. E. (2022). Parameter identifiability and model selection for sigmoid population growth models. *Journal of Theoretical Biology*, 535:110998.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'12, pages 2951–2959, Red Hook, NY, USA. Curran Associates Inc.

Steinmeister, L. and Pauly, M. (2024a). Degenerate Hierarchical Time Series Reconciliation With The Minimum Trace Algorithm in R. In Herberger, D. and Hübner, M., editors, *Proceedings of the Conference on Production Systems and Logistics: CPSL*, pages 380–390. publish-Ing.

Steinmeister, L. and Pauly, M. (2024b). Iterative Trace Minimization for the Reconciliation of Very Short Hierarchical Time Series. *arXiv*.

Steinmeister, L., Ramosaj, B., Schröter, L., and Pauly, M. (2023). Testing The Limits : A Robustness Analysis Of Logistic Growth Models For Life Cycle Estimation During The COVID-19 Pandemic. In *Proceedings of the Conference on Production Systems and Logistics: CPSL*, volume 2, pages 33–44.

Strutz, T. (2011). *Data Fitting and Uncertainty*. Springer Vieweg, Wiesbaden.

Taylor, M. (2023). The us chips and science act of 2022. *MRS Bulletin*, 48(9):874–879.

Thompson, D. (2013). BlackBerry's Ridiculously Fast Demise in 3 Charts. theatlantic.com/business/archive/2013/09/blackberrys-ridiculously-fast-demise-in-3-charts/279913/. Accessed: 06 November 2024.

Tjørve, K. M. C. and Tjørve, E. (2017). The use of Gompertz models in growth analyses, and new Gompertz-model approach: An addition to the Unified-Richards family. *PLOS ONE*, 12(6):e0178691.

Tsoularis, A. and Wallace, J. (2002). Analysis of logistic growth models. *Mathematical Biosciences*, 179(1):21–55.

Verhulst, P.-F. (1838). Notice sur la loi que la population suit dans son accroissement. *Correspondence Mathematique et Physique*, 10:113–126.

Wang, C.-H. and Chen, J.-Y. (2019). Demand forecasting and financial estimation considering the interactive dynamics of semiconductor supply-chain companies. *Computers & Industrial Engineering*, 138(June):106104.

Wang, J. (2023). An Intuitive Tutorial to Gaussian Processes Regression. *Computing in Science & Engineering*, pages 1–8.

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.

Winters, P. R. (1960). Forecasting Sales by Exponentially Weighted Moving Averages. *Management Science*, 6(3):324–342.

Wright, S. and Nocedal, J. (1999). Numerical Optimization. *Springer Science*, 35(67-68).

Wu, J. Z. and Chien, C. F. (2008). Modeling strategic semiconductor assembly outsourcing decisions based on empirical settings. *OR Spectrum*, 30(3):401–430.

Wu, K., Darcet, D., Wang, Q., and Sornette, D. (2020). Generalized logistic growth modeling of the COVID-19 outbreak: comparing the dynamics in the 29 provinces in China and in the rest of the world. *Nonlinear Dynamics*, 101(3):1561–1581.

Xu, Q. and Sharma, V. (2017). Ensemble sales forecasting study in semiconductor industry. In Perner, P., editor, *Advances in Data Mining. Applications and Theoretical Aspects*, pages 31–44. Springer.

Yueh, L. (2014). Nokia, Apple and creative destruction. bbc.com/news/business-27238877. Accessed: 06 November 2024.

Zwietering, M. H., Jongenburger, I., Rombouts, F. M., and van 't Riet, K. (1990). Modeling of the Bacterial Growth Curve. *Applied and Environmental Microbiology*, 56(6):1875–1881.

Part II

Publications

Article 1

Steinmeister, L., Ramosaj, B., Schröter, L., & Pauly, M. (2023). Testing The Limits: A Robustness Analysis Of Logistic Growth Models For Life Cycle Estimation During The COVID-19 Pandemic. In D. Herberger & M. Hübner (Eds.), Proceedings of the Conference on Production Systems and Logistics: CPSL. publish-Ing.
doi.org/10.15488/15265

5th Conference on Production Systems and Logistics

Testing The Limits: A Robustness Analysis Of Logistic Growth Models For Life Cycle Estimation During The COVID-19 Pandemic

Louis Steinmeister[1,2,4], Burim Ramosaj[1], Leo Schröter[4], Markus Pauly[1,3]

[1]*Department of Statistics, TU Dortmund University, Dortmund, Germany*
[2]*Graduate School of Logistics, Dortmund, Germany*
[3]*Research Center Trustworthy Data Science and Security, UA Ruhr, Dortmund, Germany*
[4]*Infineon Technologies AG, Neubiberg, Germany*

Abstract

The semiconductor industry operates in a dynamic environment characterized by rapid technological advancements, extensive research and development investments, long planning horizons, and cyclical market behavior. Consequently, staying vigilant to technological disruptions and shifting trends is crucial. This is especially challenging when external shocks seriously affect supply chain processes and demand patterns. Particularly, recent events, such as the COVID-19 pandemic, the ongoing Russian invasion of Ukraine, and high consumer price inflation impacting the semiconductor cycle emphasize the need to account for these influences.

In this context, we analyze growth patterns and life cycles of various technologies within the semiconductor industry by estimating logistic growth models. The logistic growth model was originally formulated to describe population dynamics. However, many processes outside the discipline of ecology share the fundamental characteristics of natural growth: self-proportionality and a self-regulating mechanism. Out of the different applications, two are of particular interest in the context of strategic business decisions: (1) modeling innovation diffusion and technological change to predict the mid- to long-term growth of a market, and (2) modeling of product life cycles. To obtain market growth and life cycle predictions, we apply the logistic growth model to forecast cumulative revenues by technology over time.

This model treats the analyzed technology as a closed system. However, in practice, external shocks are the norm. To analyze the robustness to such external shocks, we compare technology life cycle estimates derived from logistic growth models before and after the effects of COVID-19 became evident for a wide array of semiconductor technologies. We find that the impact of COVID-19 on these life cycle estimates is mixed, but the median change is low. Our findings have implications for the application of logistic growth models in strategic decision-making, helping stakeholders navigate the complexities of technological innovation, diffusion, and market growth.

Keywords

Innovation Management; Business Strategy; Technology; Forecasting; S-Curve; Sigmoidal Growth; Robustness; External Shocks; Supply-Chain Disruption; Semiconductor; Trustworthiness

DOI: https://doi.org/10.15488/15265
ISSN: 2701-6277

1. The semiconductor industry

The semiconductor industry is characterized by long lead times for expanding fabrication capacity, shortening life cycles, and rapid technological advances [1–3]. Consequently, strategic decisions are often long range and high impact, especially when involving R&D and fabrication capacities. Missing out on an important development can cost months or years to catch up market share. Capital-intensive investments in fabrication and high R&D costs raise the stakes further [4,5]. This highlights the need for forecasting methods that provide support to managers [4–6]. Life cycle modelling can give an indication of technologies prone to stagnation and disruption [7], providing managers with important information.

Apart from its importance to the wider economy [8], the semiconductor industry is a great test case for forecasting methods, which include technology diffusion and life cycle models [7,9,10], due to the challenges involved. Rapid technological advancements and shortening product life cycles imply that demand is volatile and difficult to predict [4,11,12]. Furthermore, the semiconductor industry does not only have complicated supply chains but also lies upstream in the supply chain for many consumer products. Consequently, it is exposed to the bullwhip effect. This effect refers to the phenomenon where small fluctuations in consumer demand can result in amplified variations in ordering patterns along the supply chain [13,14].

The above-mentioned challenges suggest that a closed-system view might be simplistic. In fact, the semiconductor industry has experienced several external shocks with severe consequences over the last few years: From geographic risks, such as earth quakes damaging sensitive fabrication equipment [15], to the impact of the COVID-19 pandemic, including subsequent government responses, and trade frictions on the global semiconductor supply chain [16] or the ongoing Russian invasion of Ukraine with its effects on inflation and consumer sentiment [17]. Therefore, the consideration of external disruptions in forecasts and technology life cycle analysis is particularly relevant in this industry, which motivates this study.

The main objective of this paper is the assessment of the trustworthiness of technology life cycle estimates derived from the logistic growth model under extreme events. Hence, a case study involving a significant external shock - the onset of the COVID-19 pandemic – is presented and its effects on these life cycle estimates are examined.

Structure: the next section provides an overview of the history of the logistic growth model and its application to innovation management and technology life cycle analysis. The methodology of this paper is presented in Section 3 with an emphasis on the estimation of the logistic curve and the derivation of technology life cycles. Section 4 completes the paper by applying the methodology to a technology portfolio of a leading semiconductor company and discussing the findings of this case study.

2. The logistic growth model

The logistic growth model, also referred to as S-shaped or sigmoidal curves, is characterized by the logistic differential equation $\frac{\mathrm{dN}}{\mathrm{dt}} = \mathrm{rN}\left(1 - \frac{\mathrm{N}}{\mathrm{K}}\right)$. Here, N represents the population size, K the carrying capacity, and r the growth rate. It was first derived by Verhulst in the early to mid-nineteenth century to describe population dynamics [18,19] and plays an important role in the Lotka-Volterra equations [20,21]. Since then, they have become popular to quantify natural growth more broadly. Growth patterns outside the discipline of ecology resemble a similar dynamic and their application across diverse disciplines has been studied by several authors [7,22–24]. For example, logistic growth models have been employed in studying the adoption of renewable energies [25], the development of prostatic hyperplasia [26], production forecasting in extremely low permeability oil and gas reservoirs [27], performance analysis of technologies [28], modelling bacterial growth [29], forecasting long-term country GDP development [30,31], and more recently, modelling the development of COVID-19 cases [32–34]. Furthermore, S-curves are popular in corporate

strategic decision-making revolving around innovation, such as anticipating disruptive attacks on one's business [35] or life cycles and investments in the adoption of new technologies as they diffuse through the marketplace [7,36,37]. This technological progression is exemplified by the evolution of mobile phones in Figure 1. The lower left logistic curve represents classical cell phones, starting from their introduction in the 1970's. Despite their vast technological improvements over the decades, their design was a limiting factor to the value they could offer to consumers: the small screen and buttons meant that they were primarily used for voice calls, messaging, and short emails. The utility of the mobile phone was radically redefined with the introduction of the iPhone, the first commercially viable smart phone. This marked the launch of the second logistic curve, which subsequently disrupted the classical cell phone market (including the decline of its former champions, Nokia and Blackberry).

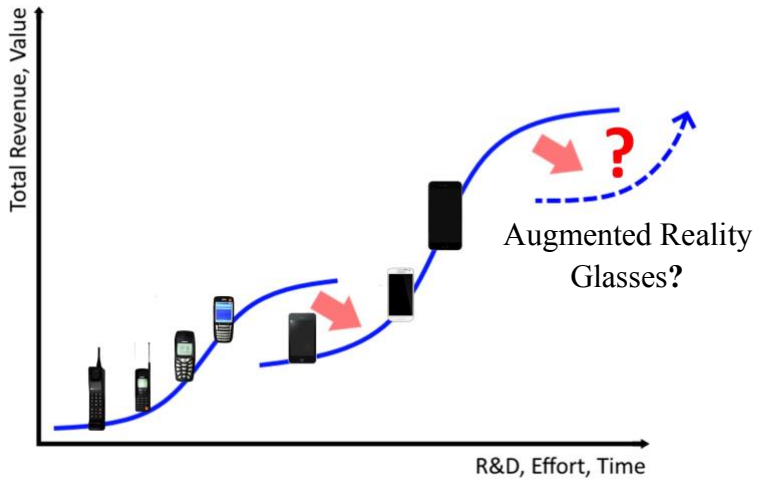


Figure 1: Technological progression exemplified by the evolution of mobile phones.

This motivates the application of logistic curves in the context of technological progress and life cycle analysis. To study the impact of the technology life cycle on strategic business decisions and corporate structure, the S-curve is often partitioned into 5 phases, as illustrated in Figure 2 [23,38,39]:

1. Birth /winter (1.4% - 6.3%): the technology is barely known or explored. Growth is slow and a large degree of effort is needed to progress. Entrepreneurship and a decentralized company structure are common.
2. Growth / spring (6.3%-30%): the technology is slowly getting adopted as "the next big thing". Growth remains nearly exponential. This phase is characterized by learning, product innovation, and continuous improvement.
3. Maturity / summer (30%-70%): the technology is being adopted and growing at its maximum rate. However, there are first signs of costs of complexity as the rate is approximately constant and departs from earlier near-exponential growth. Processes are driven by vertical integration, refinement, and bureaucratization.
4. Decline / autumn (70%-92.7%): the technology nears its limit. The growth rate remains positive but accelerates its decline. Managers should be on the lookout for the next "next big thing". This is the ear of process innovations and face lifts.
5. Death / winter (92.7%-98.6%): the technology has nearly reached its peak. There is little growth left to achieve, which requires increasingly higher investments. This means that technological progress stalls and the technology becomes prone to disruptions. Managers are looking to transition to alternative technologies while the current one phases out (compare the transition from cell to smartphones in Figure 1).

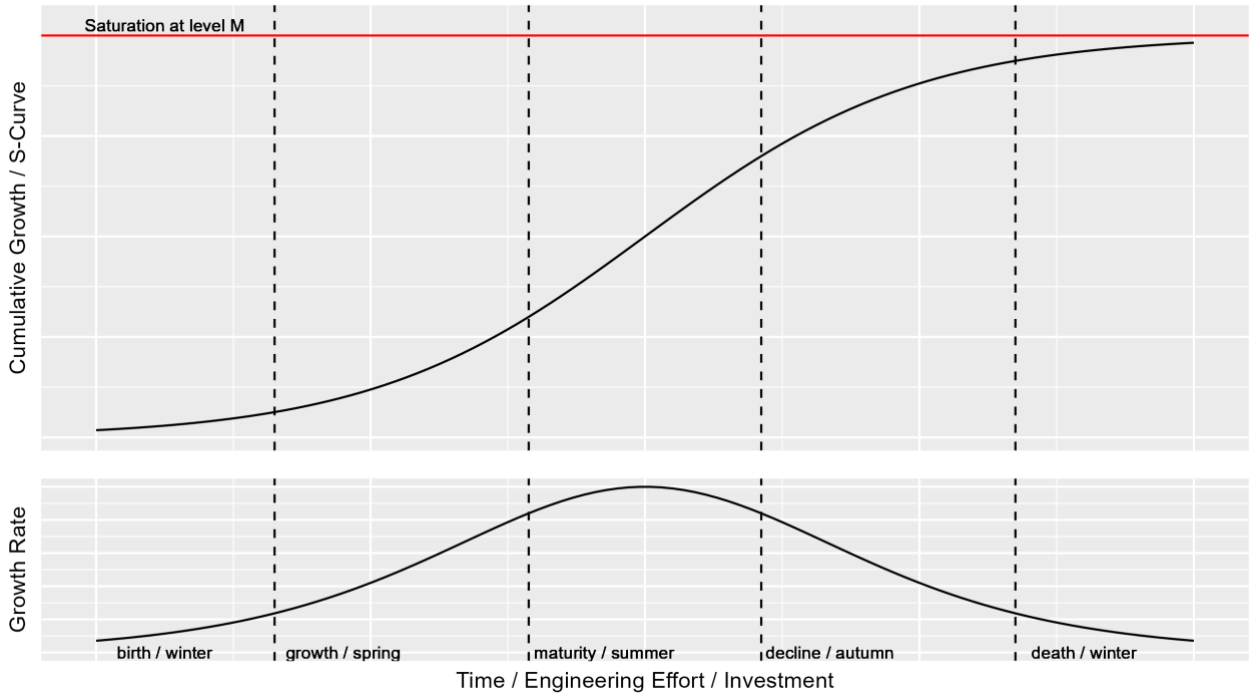


Figure 2: Partitioning the S-curve into life cycle stages.

This highlights the interpretability and usability of the simple logistic model, which played a pivotal role in formalizing the study of life cycles, on a business level. However, various extensions and generalizations of the simple logistic curve have been proposed [10, 19,40–44], such as the Richards' curve [45] or Gompertz curve. Ex post, these extended models may yield better fits [29]. However, the generalized models usually require more parameters. This can lead to identifiability issues [46,47]. Furthermore, the trajectory is seldomly observed completely, which can exacerbate the issue. Therefore, we focus on the simple logistic model in our analysis.

3. Methodology

We introduce the notation and explain the interpretation and meaning of the different parameters of the logistic growth model in 3.1. Subsection 3.2 provides details of the assumed data generating process and the estimation of model parameters.

3.1 Notation

As described in Section 2, the logistic growth model is described by the differential equation $\frac{\mathrm{dN}}{\mathrm{dt}} = \mathrm{rN}\left(1 - \frac{\mathrm{N}}{\mathrm{K}}\right)$. For consistency with common statistical notation, we rewrite the solution of this differential equation as

$$Y = \frac{M}{1+e^{aX+b}} + C, \tag{1}$$

where Y is the response variable (in place of N), M is the carrying capacity or maximum cumulative market size, a corresponds to the growth rate (the maximum growth rate of $aM/4$ is reached at the inflection point), b determines the location of the inflection point (at $x = -b/a$), C is introduced to allow for vertical offsets of the logistic curve, and X denotes the dependent variable. The logistic curve with its parameters and their

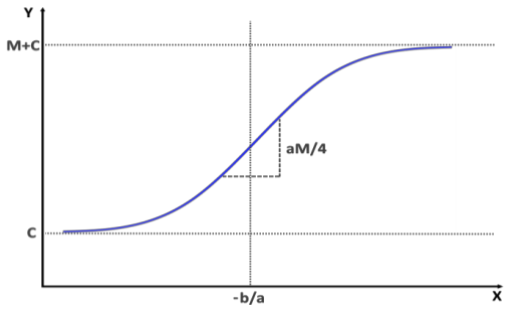


Figure 3: Parametrization and interpretation of parameters of the logistic curve.

respective interpretations is illustrated in Figure 3. This parametrization implies that the life cycle stage can be estimated as $(max(Y) - C)/(M - C)$.

3.2 Estimation

We assume the data generating process

$$Y = \frac{M}{1+e^{aX+b+\varepsilon}} + C, \tag{2}$$

where ε is an identically and independently distributed error variable with existing first and second moments. This model formulation allows us to estimate the parameters M, C, a, b in an iterative two-step process:

i. Estimate M and C. At initialization, start with two reasonable first guesses $\hat{M} > max(Y)$, $\hat{C} < min(Y)$. For later iterations minimize the mean squared error of the step ii. regression with regard to M and C.

ii. Given M and C, the remaining parameters can be obtained via a simple linear regression

$$log\left(\frac{\hat{M}-Y+\hat{C}}{Y-\hat{C}}\right) = aX + b + \varepsilon. \tag{3}$$

The mean squared error is given by $\frac{1}{N}\sum_{i=1}^{N}\left[log\left(\frac{\hat{M}-Y_i+\hat{C}}{Y_i-\hat{C}}\right) - \hat{a}X_i - \hat{b}\right]^2$, where N is the sample size.

The optimization in step i. can be performed with any conventional optimization routine, such as Nelder and Mead's Simplex algorithm [48] or quasi-Newton methods such as BFGS or L-BFGS-B [49]. However, these local optimization methods tend to struggle with local optima [50]. This is particularly problematic when the error surface is rough. We used Generalized Simulated Annealing, which is implemented in the "GenSA" R package [51], as this yielded the most reliable results.

4. Empirical Analysis

Subsection 4.1 covers the introduction to the data and includes all pre-processing steps in. Subsection 4.2 presents the results of the analysis and discusses the implications.

4.1 Data

We obtained revenues for a diverse portfolio of products ranging from October 2006 to January 2023 from a leading semiconductor company. However, it is not reasonable to assume that one Euro today has the same value it had 17 years ago. Therefore, it is important to adjust for the general price level. This was done with the Harmonized Index of Consumer Inflation, which is published by the European Central Bank[1] [52]. These revenues were mapped to technology categorizations on the aggregation levels: Level 1, containing 18 technologies and Level 2, containing 10. Here, Level 1 (component) technology groups are more granular than Level 2 technology groups, see Figure 4 (the technologies and their corresponding technology groups, which are covered in this paper, can be seen in Table 1 in the appendix).

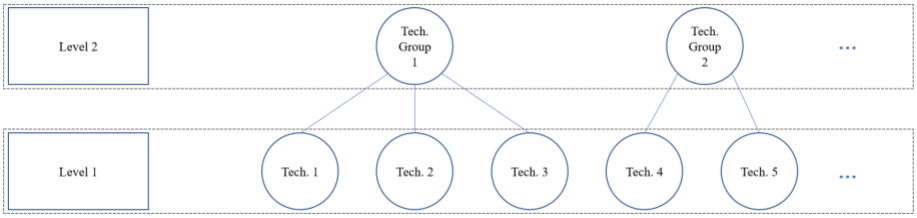


Figure 4: Example of a technology hierarchy.

We consider both, Level 1 and Level 2 technologies, because Level 1 technologies are more homogeneous but Level 2 technologies include more sub-technologies that would need to be excluded due to lack of data history. By aggregating several similar technologies, we further hope to reduce the effects of substitution and similar interactions. Furthermore, the logistic model should be applicable on either level because of its fractal property.

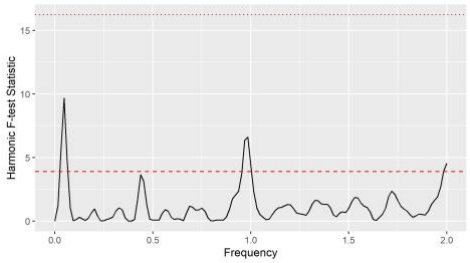


Figure 5: Harmonic F-test statistic over the frequency domain. The dashed red line corresponds to a point wise 95% confidence threshold, the dotted red line to a global 95% confidence threshold.

Further, we analyzed cyclicality by means of spectral analysis, particularly using a periodogram [53]. However, no frequency surpassed the significance threshold of the harmonic F-test [54] provided in the "multitaper" R package. As Figure 5 shows, there is no statistically significant cyclicality at any frequency when adjusting for multiple testing.

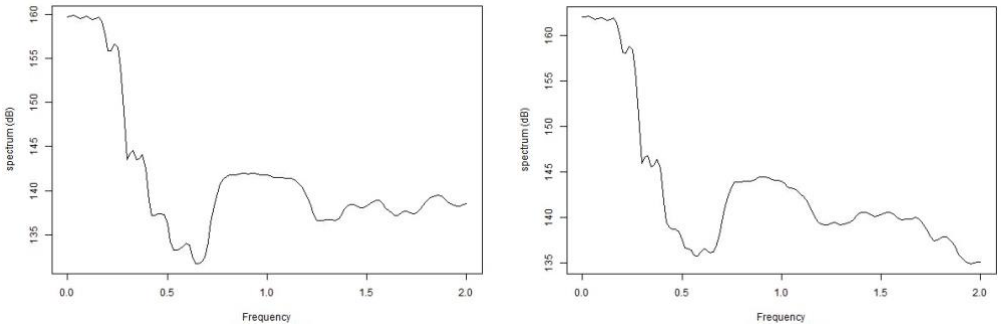


Figure 6: Periodograms before (left) and after (right) seasonal adjustment.

[1] Information on the Harmonized Index of Consumer Inflation and the corresponding time series can be found at www.ecb.europa.eu/stats/macroeconomic_and_sectoral/hicp

Additionally, we performed a time series decomposition into trend, seasonal effects, and a random component [55] using the "decompose" function in R. However, the time series post seasonal adjustment was not noticeably less noisy than the original time series, nor did the periodogram change much (see Figure 6). Therefore, we proceeded with the original data without seasonal adjustments.

4.2 Result

Logistic growth models were fit to cumulative technology revenues using the algorithm described in Section 3.2**.** We visually checked the models for consistency with the logistic curve by observing the quality of the fit and producing residual plots. These residual plots were helpful in checking if the models converged as expected and to identify residual patterns, which indicate a poor model fit.

Figure 7: Example residual plots

Figure 7 indicates a residual plot of a good logistic fit for Tech. 14, whereas the residual plot for Tech. 17 raises questions. Residuals are indicated by red circles. The drawn blue line represents the linear fit of the data in step ii. of section 3.2. If this line differs from the horizontal line at $Y = 0$, it indicates that the algorithm did not converge as expected. The dashed and dotted blue lines represent point-wise confidence intervals and confidence bands at 95%, respectively. Hence, a residual lying outside the 95% point-wise confidence interval would be expected to be observed every 20 data points, whereas a residual lying outside the 95% confidence band would be expected every 20 residual plots. These plots can help identify external shocks or patterns that are not captured by the model. In case of the left plot, the residuals are well dispersed and no clear trend is identifiable. This indicates a good fit. On the right, there is a clear pattern, which indicates a poor model fit.

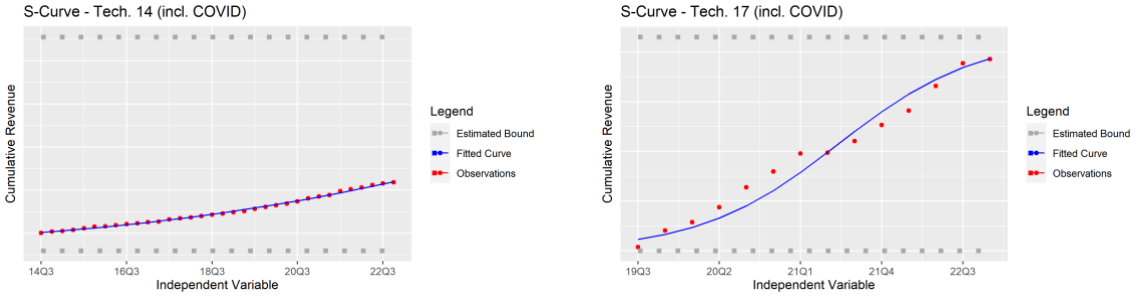


Figure 8: Example logistic fits. Tech. 14 (left) is an example of a good fit, Tech. 17 (right) one of a questionable one.

Additionally, we validated the logistic life cycle estimates by expert opinion. We excluded poor fits and technologies that were clearly driven by external structural effects (the semiconductor industry is largely a B2B business, where sales are often conducted through direct relationships – in segments where revenues are overwhelmingly driven by a few large customers, patterns in the data may be dominated by individual decisions at the level of a single customer and not always be reflective of the technological potential of the product). Overall, we excluded seven of the eighteen Level 1 technologies and four Level 2 technology from further analysis (see Table 1 in the Appendix).

Figure 8 illustrates logistic fits for Tech. 14 (left), which fits well, and Tech. 17 (right), which does not. Tech. 17 is an example of a technology that is driven by individual projects (also observe that the data in the right plot appears to be consisting of two logistic curves, not one). The blue curve indicates the estimated logistic model, the red dots correspond to the observed cumulative revenues, and the grey dotted lines

correspond to the estimated lower bound, C, and the estimated upper bound, $M + C$. Generally, we did not estimate lower bounds unless we had reason to believe that we were missing previous revenues.

These logistic curves were estimated based on the complete history (from October 2006 to January 2023) and on the basis of the historical data preceding the COVID-19 Pandemic. After the curves were obtained and validated for the various Level 1 and Level 2 Technologies, we compared the life cycle estimates of the pre- and post-COVID models.

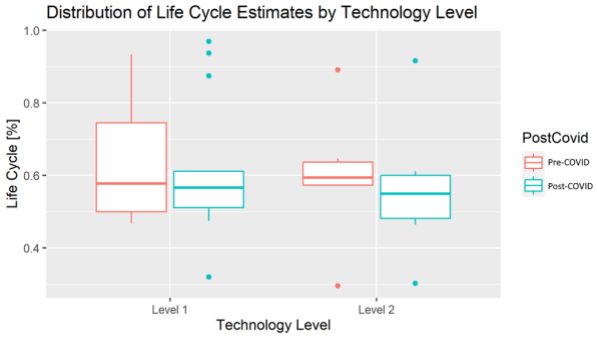


Figure 9: Boxplots of life cycle estimates before and after COVID.

Figure 9 illustrates the distributions of the Pre- (red) and Post- (cyan) COVID life cycle estimates on technology Level 1 (left) and 2 (right), respectively. Given the strong demand for consumer electronics and the subsequent rejuvenation of sales numbers, we expected lower life cycle estimates as an impact of the pandemic. Meanwhile, a moderate increase is expected without the interference of external shocks, given that two years elapsed and the technology has aged. On Level 1, the median life cycle estimate remained relatively stable, though more concentrated in the lower range. Thus, the logistic fits for the Level 1 technologies seem relatively unaffected by the pandemic. This is confirmed by Figure 10, which shows that the median life cycle estimate has increased slightly after COVID, which is consistent with our expectation. This result is slightly different for Level 2 technologies. As Figure 9 indicates, the distribution of life cycle estimates has noticeably shifted downwards. This observation is confirmed by Figure 10, which indicates that the median life cycle estimates have decreased by 2% after the revenues during the pandemic are included. This seems to confirm our hypothesis that COVID has had an impact on the logistic growth models. On the other hand, this decrease could be due to the inclusion of new emerging sub-technologies in the broader technology group. These would have been omitted during the analysis of the Level 1 technologies, due to small volumes and an insufficient amount of historical data.

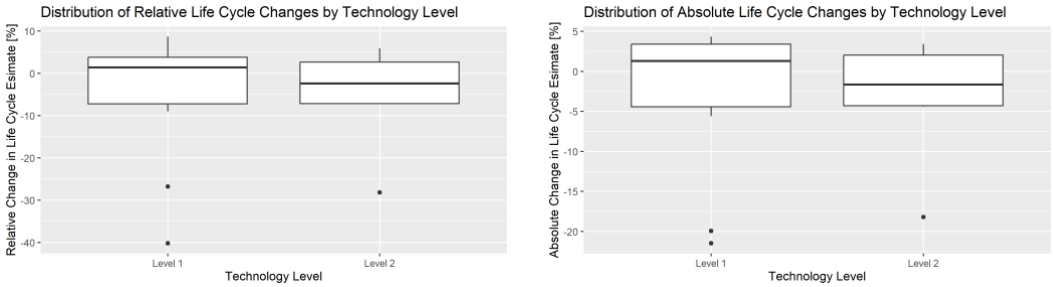


Figure 10: Boxplots of the change in life cycle estimates before and after COVID.

We conclude that logistic growth models are a valuable tool for managers in assessing product life cycles if the model is applicable. This is particularly useful in understanding technological limitations and guarding against the risk of disruption. The robustness to external effects could be further improved by incorporating them into the logistic growth model. For example, researchers have modelled the upper bound dynamically [56,57]. Given the dynamic nature of the semiconductor industry, the complexity of supply chain dependencies, and the exposure to other external factors, this highlights a potential for future research in the field.

Acknowledgements

This paper was supported by a research cooperation between Infineon Technologies AG and TU Dortmund University through the Graduate School of Logistics.

Appendix

Table 1: Technology table of technology groups and the corresponding technologies. Technologies marked with an (X) were excluded from further analysis.

Technology Group (Level 2)	Technology (Level 1)
Tech. Group 1	Tech. 7, Tech. 3 (X), Tech. 1 (X)
Tech. Group 2	Tech. 2, Tech. 4
Tech. Group 3	Tech. 17 (X), Tech. 14, Tech. 5
Tech. Group 4 (X)	Tech. 11, Tech. 9 (X), Tech. 6
Tech. Group 5 (X)	Tech. 12, Tech. 10 (X)
Tech. Group 6	Tech. 8
Tech. Group 7	Tech. 13
Tech. Group 8	Tech. 15
Tech. Group 9 (X)	Tech. 16 (X)
Tech. Group 10 (X)	Tech. 18 (X)

References

[1] Macher, J. T., 2006. Technological development and the boundaries of the firm:a knowledge-based examination in semiconductor manufacturing. Management Science, 52 (6), 826–843.

[2] Wu, J. Z., Chien, C. F., 2008. Modeling strategic semiconductor assembly outsourcing decisions based on empirical settings. OR Spectrum, 30 (3), 401–430.

[3] Lv, S., Kim, H., Zheng, B., Jin, H., 2018. A review of data mining with big data towards its applications in the electronics industry. Applied Sciences, 8 (4), 582.

[4] Chien, C.-F., Chen, Y.-J., Peng, J.-T., 2010. Manufacturing intelligence for semiconductor demand forecast based on technology diffusion and product life cycle. International Journal Of Production Economics, 128 (2), 496–509.

[5] Aubry, M., Renou-Maissant, P., 2014. Semiconductor industry cycles: explanatory factors and forecasting. Economic Modelling, 39, 221–231.

[6] Cakanyildirim, M., Roundy, R. O., 2002. SeDFAM: semiconductor demand forecast accuracy model. IIE Transactions, 34 (5), 449–465.

[7] Modis, T., 1994. Life cycles: forecasting the rise and fall of almost anything. Futurist, 28 (5), 20–25.

[8] Chow, H. K., Choy, K. M., 2006. Forecasting the global electronics cycle with leading indicators: a Bayesian VAR approach. International Journal of Forecasting, 22 (2), 301–315.

[9] Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., et al., 2022. Forecasting: theory and practice. International Journal of Forecasting, 38 (3), 705–871.

[10] Meade, N., 1985. Forecasting using growth curves-an adaptive approach. The Journal of The Operational Research Society, 36 (12), 1103.

[11] Simchi-Levi, D., Kaminsky, P., Simchi-Levi, E., Ji, J., 2021. Designing and managing the supply chain, 4th ed. Irwin McGraw-Hill, New York.

[12] Chou, Y. C., Cheng, C. T., Yang, F. C., Liang, Y. Y., 2007. Evaluating alternative capacity strategies in semiconductor manufacturing under uncertain demand and price scenarios. International Journal of Production Economics, 105 (2), 591–606.

[13] Geary, S., Disney, S. M., Towill, D. R., 2006. On bullwhip in supply chains—historical review, present practice and expected future impact. International Journal of Production Economics, 101 (1), 2–18.

[14] Lee, H. L., Padmanabhan, V., Whang, S., 1997. The bullwhip effect in supply chains. MIT Sloan Management Review, 38 (3), 93–102.

[15] Matsuo, H., 2015. Implications of the Tohoku earthquake for Toyota′s coordination mechanism: supply chain disruption of automotive semiconductors. International Journal of Production Economics, 161, 217–227.

[16] Wu, X., Zhang, C., Du, W., 2021. An analysis on the crisis of "chips shortage" in automobile industry —— based on the double influence of COVID-19 and trade friction. Journal of Physics: Conference Series, 1971 (1), 012100.

[17] Caldara, D., Conlisk, S., Iacoviello, M., Penn, M., 2022. The effect of the war in Ukraine on global activity and inflation. Board of Governors of the Federal Reserve System, Washington.

[18] Verhulst, P.-F., 1838. Notice sur la loi que la population suit dans son accroissement. Correspondence Mathematique Et Physique, 10, 113–126.

[19] Tsoularis, A., Wallace, J., 2002. Analysis of logistic growth models. Mathematical Biosciences, 179 (1), 21–55.

[20] Lotka, A. J., 1910. Contribution to the theory of periodic reactions. The Journal of Physical Chemistry, 14 (3), 271–274.

[21] Berryman, A. A., 1992. The orgins and evolution of predator-prey theory. Ecology, 73 (5), 1530–1535.

[22] Bejan, A., Lorente, S., 2012. The S-curves are everywhere. Mechanical Engineering, 134 (5), 44–47.

[23] Kucharavy, D., De Guio, R., 2011. Application of S-shaped curves. Procedia Engineering, 9, 559–572.

[24] Kucharavy, D., De Guio, R., 2011. Logistic substitution model and technological forecasting. Procedia Engineering, 9, 402–416.

[25] Schilling, M. A., Esmundo, M., 2009. Technology S-curves in renewable energy alternatives: analysis and implications for industry and government. Energy Policy, 37 (5), 1767–1781.

[26] Berry, S. J., Coffey, D. S., Walsh, P. C., Ewing, L. L., 1984. The development of human benign prostatic hyperplasia with age. Journal Of Urology, 132 (3), 474–479.

[27] Clark, A. J., Lake, L. W., Patzek, T. W., 2011. Production forecasting with logistic growth models, in: SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers, Denver.

[28] Nieto, M., Lopéz, F., Cruz, F., 1998. Performance analysis of technology using the S curve model: the case of digital signal processing (DSP) technologies. Technovation, 18 (6–7), 439–457.

[29] Zwietering, M. H., Jongenburger, I., Rombouts, F. M., van 't Riet, K., 1990. Modeling of the bacterial growth curve. Applied and Environmental Microbiology, 56 (6), 1875–1881.

[30] Kwasnicki, W., 2013. Logistic growth of the global economy and competitiveness of nations. Technological Forecasting and Social Change, 80 (1), 50–76.

[31] Modis, T., 2013. Long-term GDP forecasts and the prospects for growth. Technological Forecasting and Social Change, 80 (8), 1557–1562.

[32] Wu, K., Darcet, D., Wang, Q., Sornette, D., 2020. Generalized logistic growth modeling of the COVID-19 outbreak: comparing the dynamics in the 29 provinces in China and in the rest of the world. Nonlinear Dynamics, 101 (3), 1561–1581.

[33] Shen, C. Y., 2020. Logistic growth modelling of COVID-19 proliferation in China and its international implications. International Journal of Infectious Diseases, 96, 582–589.

[34] Aviv-Sharon, E., Aharoni, A., 2020. Generalized logistic growth modeling of the COVID-19 pandemic in Asia. Infectious Disease Modelling, 5, 502–509.

[35] Foster, R. N., 1986. Working the S-Curve: assessing technological threats. Research Management, 29 (4), 17–20.

[36] Christensen, C. M., 1992. Exploring the limits of the technology S-Curve. part I: component technologies. Production and Operations Management, 1 (4), 334–357.

[37] Rogers, E. M., 2014. Diffusion of innovations. Simon and Schuster, New York.

[38] Modis, T., 1999. A second lease on life for technological forecasting. Technological Forecasting and Social Change, 62 (1–2), 29–32.

[39] Modis, T., 1998. Conquering uncertainty: understanding corporate cycles and positioning your company to survive the changing environment. McGraw-Hill, New York.

[40] Banks, R. B., 1993. Growth and diffusion phenomena: mathematical frameworks and applications. Springer, Berlin.

[41] Oliver, F. R., 1969. Another generalisation of the logistic growth function. Econometrica, 37 (1), 144.

[42] Easingwood, C., Mahajan, V., Muller, E., 1981. A nonsymmetric responding logistic model for forecasting technological substitution. Technological Forecasting and Social Change, 20 (3), 199–213.

[43] Skiadas, C., 1985. Two generalized rational models for forecasting innovation diffusion. Technological Forecasting and Social Change, 27 (1), 39–61.

[44] Bewley, R., Fiebig, D. G., 1988. A flexible logistic growth model with applications in telecommunications. International Journal of Forecasting, 4 (2), 177–192.

[45] Richards, F. J., 1959. A flexible growth function for empirical use. Journal of Experimental Botany, 10 (2), 290–301.

[46] Simpson, M. J., Browning, A. P., Warne, D. J., Maclaren, O. J., Baker, R. E., 2022. Parameter identifiability and model selection for sigmoid population growth models. Journal of Theoretical Biology, 535, 110998.

[47] Benzekry, S., Lamont, C., Beheshti, A., Tracz, A., Ebos, J. M. L., et al., 2014. Classical mathematical models for description and prediction of experimental tumor growth. PLoS Computational Biology, 10 (8), e1003800.

[48] Nelder, J. A., Mead, R., 1965. A simplex method for function minimization. The Computer Journal, 7 (4), 308–313.

[49] Byrd, R. H., Lu, P., Nocedal, J., Zhu, C., 1995. A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing, 16 (5), 1190–1208.

[50] Wright, S., Nocedal, J., 1999. Numerical optimization. Springer, New York.

[51] Xiang, Y., Gubian, S., Suomela, B., Hoeng, J., 2013. Generalized simulated annealing for global optimization: the GenSA package. The R Journal, 5 (1), 13.

[52] Diewert, W. E., 2002. Harmonized indexes of consumer prices: their conceptual foundations. SSRN Electronic Journal, 138 (4), 547–637.

[53] Shumway, R. H., Stoffer, D. S., 2017. Time series analysis and its applications. Springer, New York.

[54] Thomson, D. J., 1982. Spectrum estimation and harmonic analysis. Proceedings of the IEEE, 70 (9), 1055–1096.

[55] Kendall, M., Stuart, A., 1983. The advanced theory of statistics, 3rd ed. Griffin, London.

[56] Seidl, I., Tisdell, C. A., 1999. Carrying capacity reconsidered: from Malthus' population theory to cultural carrying capacity. Ecological Economics, 31 (3), 395–408.

[57] Calatayud, J., Cortés, J.-C., Dorini, F. A., Jornet, M., 2020. On a stochastic logistic population model with time-varying carrying capacity. Computational and Applied Mathematics, 39 (4), 288.

Biography

Louis Steinmeister was awarded his MS degree in applied mathematics from Missouri S&T, where he was a member of the Applied Computational Intelligence Lab. His research interests include machine learning, statistics, and their application to business and financial problems. Currently, he is pursuing a Ph.D. in Statistics through the Graduate School of Logistics at TU Dortmund University, Germany.

Burim Ramosaj (*1991) did his PhD on Statistical Inference Models using ML in 2019 and was a Post-Doc at TU Dortmund University. He researched significant feature selection using advanced ML-models with his own funding from the state of NRW through the KI-Starter Programme. Currently, he is working as a Data Scientist at Google in Zurich, Switzerland and continues research in his free time.

Leo Schröter (*1994) has been working as a Financial Controller at Infineon Technologies AG since 2021. Mr. Schroeter studied Industrial Engineering majoring in Finance and Communication Engineering at the University of Duisburg-Essen.

Markus Pauly received his diploma degree in mathematics from the Heinrich-Heine University Düsseldorf (2005), where he also completed his Ph.D (2008) and his habilitation (2013). From 2014 to 2019 he was full professor for Statistics at Ulm University. Since 2019 he is full professor at the Statistics Department at TU Dortmund University, where he investigates a wide range of statistical and machine learning topics.

Article 2

Steinmeister, L., & Pauly, M. (2025). Human vs. Machines: Who wins in semiconductor market forecasting? Expert Systems with Applications, 263(October 2024), 125719. doi.org/10.1016/j.eswa.2024.125719

Expert Systems With Applications 263 (2025) 125719

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Human vs. Machines: Who wins in semiconductor market forecasting?

Louis Steinmeister [a,b,*], Markus Pauly [a,c]

[a] Department of Statistics, TU Dortmund University, Vogelpothsweg 87, 44227 Dortmund, Germany
[b] Graduate School of Logistics, Leonhard-Euler-Straße 5, 44227 Dortmund, Germany
[c] Research Center Trustworthy Data Science and Security, Joseph-von-Fraunhofer-Straße 25, 44227 Dortmund, Germany

ARTICLE INFO

Dataset link: wsts.org

Keywords:
Prediction
Sales forecast
Semiconductor cycle
Demand forecast
Machine learning
Statistical learning

ABSTRACT

"*If you ask ten experts, you will get ten different opinions.*" This common proverb illustrates the common association of expert forecasts with personal bias and lack of consistency. On the other hand, digitization promises consistency and explainability through data-driven forecasts employing machine learning (ML) and statistical models. Despite the importance of the semiconductor industry being widely recognized, little research has gone into forecasting the whole semiconductor market including all major product categories. Instead, analysts have generally relied on expert forecasts such as those provided by the World Semiconductor Trade Statistics (WSTS). In the following, we generate data-driven forecasts and evaluate whether existing industry expert forecasts can be further enhanced through statistical and ML models. This study contributes by systematically evaluating the accuracy of expert forecasts, examining comprehensive multi-granularity forecasts for the entire semiconductor market, and offering performance insights through out-of-sample error measures to guide future forecasting practitioners.

1. Executive summary

Objectives: The objective of this paper is to evaluate and contrast expert predictions of the World Semiconductor Trade Statistics (WSTS), a leading semiconductor market data provider, with data-driven forecasts. In this context, we compare expert forecasts with different data-driven forecast approaches with respect to three research hypotheses detailed in the introduction.

Motivation: WSTS plays a crucial role in the semiconductor industry. According to their website, WSTS is the "*most respected source of market data and forecasts for the semiconductor industry*" and their forecasts "*are the only ones that leverage the collective experience of the industry's major players with the market intelligence of a large portion of the semiconductor industry*" (WSTS.org, 2024). As one of the top providers of comprehensive semiconductor industry data and indicators, WSTS plays a pivotal role in business decision making and industry analyst research. Additionally, the well-being of the semiconductor industry, which lies upstream in the supply chain, has been identified to be a leading indicator for the broader economy (Chow & Choy, 2006). This highlights the importance of accurate and reliable semiconductor industry forecasts even outside of this specific industry.

Methods: Several popular statistical and ML methods for time series forecasting are evaluated against official forecasts provided by WSTS by means of a time series cross validation.

Results: This paper finds that the expert forecasts provided by WSTS compare favorably to ML forecasts on a quarterly horizon but can nevertheless be enhanced by data driven forecasts. However, the performance of WSTS forecasts is put in perspective when the WSTS algorithmic updates, which are published bi-quarterly, are included. Furthermore, it can be argued that additional information should be incorporated into the forecasts, which results in a clear outperformance of the data-driven methods in comparison to the official WSTS forecasts. This discovery remains consistent regardless of the length of available data points. In other words, the outcome remains unaffected whether we analyze product categories with short histories or those with long histories.

Contribution: This study contributes in the following ways: (1) It provides a novel evaluation of the accuracy of expert forecasts within the semiconductor market, (2) it shows that comprehensive forecasting across various levels of granularity for the entire semiconductor industry is feasible, even with simple models, and (3) it provides valuable guidance to forecasting practitioners by supplying out-of-sample error measures for all analyzed models and product categories.

Conclusion: While WSTS forecasts provide a strong starting point, it is possible to improve the forecast accuracy through data-driven approaches.

* Correspondence to: Graduate School of Logistics, TU Dortmund University, Leonhard-Euler-Straße 5, 44227 Dortmund, Germany.
E-mail addresses: louis.steinmeister@tu-dortmund.de (L. Steinmeister), pauly@statistik.tu-dortmund.de (M. Pauly).

https://doi.org/10.1016/j.eswa.2024.125719
Received 2 August 2024; Received in revised form 4 October 2024; Accepted 4 November 2024
Available online 13 November 2024

2. Introduction

2.1. Motivation

In the today's dynamic business landscape, accurate forecasts play an increasingly important role in shaping operative and strategic decisions. The article "Bringing a real-world edge to forecasting" released by McKinsey & Company in 2020 makes the case that "*[a 'good' forecasting process] should be accurate enough to inform a range of critical business decisions – capital reallocation, hiring, strategy, production, and more*" (Agrawal et al., 2020).

For example, Wang et al. (2024) propose a freight rate index forecasting model to help industry players mitigate risks in the shipping market and inform investment opportunities and Wu et al. (2024) developed a model to forecast container throughput to inform port logistics industry decision makers. Additionally, Yu et al. (2012) propose a forecasting model to predict the fashion color trend leading to a higher success rate of new fashion products and an example of how resource allocation can be improved by increased operational efficiency through more accurate short to mid-term demand forecast is provided by Pauly and Kuhlmann (2023). Furthermore, forecasts play an important role in anticipating technological change (Foster, 1986; Modis, 1999) and estimating technology and product life cycles, which inform important portfolio and product development decisions (Modis, 1994; Petropoulos et al., 2022; Steinmeister et al., 2023). This is particularly true for the semiconductor industry with long lead times, a dynamic technological environment, and shortening product life cycles (Lv et al., 2018; Macher, 2006; Wu & Chien, 2008).

Accurate forecasts of the semiconductor market are relevant to the broader economy. The global semiconductor market reached sales of $618 billion in 2022 according to Alsop (2024b). This amounted to about 0.61% of global GDP in 2022 compared to 0.22% in 1990, highlighting the increasing importance of the industry to the global economy (Alsop, 2024a; World Bank, 2024). Semiconductors are ubiquitous in modern life. They enable AI applications, modern defense equipment, data centers, automobiles, wireless communications, all the way to home appliances like your washing machine, gaming console, and electronic toothbrush. As Chow and Choy (2006) observe, the semiconductor market is a leading indicator for the broader economy.

The strategic importance of the semiconductor industry has further been recognized by governments. The US authorized about $280 billion for research and manufacturing of semiconductors in the US with the CHIPS and Science Act passed in August 2022 (Taylor, 2023). Likewise, the EU subsidizes the industry with roughly 43 billion Euros (roughly $46.3 billion) (European Comission, 2023). Even more impressively, Sam Altman, the current CEO of OpenAI, is seeking $5 trillion to $7 trillion of investments to "*boost the world's chip-building capacity and expand its ability to power AI, among other things*" according to Reuters (Rajan, 2024). To add perspective: this amounts to the combined market capitalization of Microsoft and Apple, the two largest American companies by market capitalization, at the time of writing.[1]

This motivates a detailed study of semiconductor market forecasts, which often inform industry and financial analysts. Furthermore, semiconductor market forecasts often inform internal goal setting and benchmarking. Semiconductor market data also factors into the calculation of market share by product groups. Projections of these quantities can have strategic implications.

The World Semiconductor Trade Statistics (WSTS)[2] is a premier provider of such data and forecasts. Several academic studies and industry reports cite WSTS as an authority for semiconductor market forecasts, see Corder et al. (2024), Nagao (2019) and Simons (2024) to name a few. Furthermore, the standing of WSTS as a data provider is highlighted by the common use of their data in academic fields such as the semiconductor cycle prediction, as outlined in Table 2 which provides a selection of related works in the field.

Industry forecasts, such as those from WSTS, serve as the basis for important strategic decisions and are often based on expert judgement. A popular method for consolidating these forecasts is the Delphi method (Armstrong, 2008; Delphi, 1975; Hyndman & Athanasopoulos, 2018). However, several researchers have discovered that these collective predictions, also referred to as "wisdom of the crowds" or "collective intelligence", are susceptible to inaccuracies and low precision, particularly when the individuals surveyed are pundits or uninformed laypersons (Atanasov et al., 2015; Modis, 1999).

However, the industry members polled by WSTS are industry experts and have access to detailed insider information, such as customer orders and sentiment, customer-related project status, the status of customer contracts, and new product development. Expert forecasts are generally thought to perform well when systems are complex, dynamic, and available history is sparse (Hyndman & Athanasopoulos, 2018). The semiconductor industry is heavily intertwined in the global economy, with complex upstream, and downstream supply chains. This means that the semiconductor industry is exposed to the bullwhip effect (Lee et al., 1997). It is also increasingly subjected to geopolitical considerations. While these factors highlight the importance of accurate semiconductor industry forecasts, the dynamic technological environment, geopolitical factors, and the complex supply chains concurrently complicate the data-driven forecast process due to the amount of potentially relevant extrinsic information. Therefore, expert forecasts from industry insiders queried by WSTS, who possess extensive access to quantitative and qualitative insider information, are anticipated to be highly reliable. Moreover, the lack of research into the area of detailed semiconductor market forecasts, illustrated in the related works section (Section 2.2), raises the question whether industry experts may simply be more accurate than data-driven models in this field. This prompts inquiry into the potential competitiveness of data-driven forecasts, and if and how they might enhance existing expert forecasts. To our knowledge the accuracy of semiconductor industry forecasts has not been systematically evaluated despite their regular use. The present paper closes this gap. To this end, the following three research hypotheses are examined in the following:

(H1) Expert forecasts exhibit higher accuracy compared to autoregressive data-driven forecasts.

WSTS publishes quarterly forecasts for each product category in an alternating pattern: expert forecasts are issued in May and November, while algorithmically computed updates are provided in February and August. These algorithmic updates are derived from the preceding quarter's results. Industry experts also only have access to official WSTS figures dating back to the previous quarter. Nevertheless, the numbers for the initial month of the forecasted quarter are disclosed simultaneously with the forecasts. Furthermore, it can be anticipated that industry experts possess internal information regarding the first month's data (for a detailed account of the data, see Section 4.1). This gives rise to the second hypothesis:

(H2) The incorporation of additional autoregressive information in data-driven forecasts enhances their competitiveness against expert forecasts.

Fig. 1 within Section 4.1 illustrates that certain product categories exhibit significantly shorter historical data compared to others. This observation holds significance as it aligns with the understanding that experts possess the capability to generate accurate forecasts when historical data is limited (Hyndman & Athanasopoulos, 2018). Consequently, this observation motivates the formulation of the third hypothesis:

[1] Based on: https://www.tradingview.com/markets/world-stocks/worlds-largest-companies/. Accessed 03 April 2024.

[2] wsts.org.

Table 1
Selection of related works in semiconductor sales and demand forecasting including response variables and utilized forecasting methods.

Authors	Methods	Response variable
Wang and Chen (2019)	ARIMA, VAR[a]	Quarterly sales time series from Taiwanese semicondoctor companies from Q1 2009 to Q4 2018
Kapur et al. (2019)	Technology diffusion	Sales and price data for DRAM, LCD monitors, and room air-conditioners
Chen and Chien (2018)	Technology diffusion	27 quarters of shipments for two technology generations of non-volatile memory products from a semiconductor company
Xu and Sharma (2017)	XGBoost, Linear model, RF[b], ARIMA, Ensemble	Weekly Intel CPU sales in 2012
Chien and Lin (2012)	Rolling grey forecasting method	Annual sales of companies in Hsinchu Science Park from 1983 to 2010
Aytac and Wu (2013)	Extended, Bayesian logistic growth model	Monthly sales data of about 5300 short life-cyle products from three semiconductor companies
Chien et al. (2013)	Technology diffusion	36 quarters of demand data for four technologies of a leading foundry from Hsinchu Science Park

[a] Vector autoregression.
[b] Random Forest.

(H3) Expert forecasts outperform data-driven forecasts particularly in the context of short time series.

Structure: The following two subsections summarize the related work (Section 2.2) and highlight the research gap and our contributions (Section 2.3). To investigate the hypotheses, the ML and statistical methods used are summarized in Section 3. The results are discussed in Section 5, with Section 5.1 addressing (H1), Section 5.2 examining (H2), Section 5.3 (H3), and finally Section 5.4, which offers insights of the results at a product category level. Section 6 completes this work with a brief discussion of the findings.

2.2. Related works

Despite the recognition of the semiconductor industry's importance in politics and business, little research was dedicated to data-driven forecasting the semiconductor market. A Scopus search ("market forecast" AND "semiconductor" AND (statistic* OR "machine learning")) yields only one search result which discusses front-end drivers of changes in the semiconductor market (Nagao, 2019). However, this paper does not apply statistical or machine learning methods to generate forecasts directly. The small number of academic studies in the sector of the semiconductor industry was also noted by Aubry and Renou-Maissant (2014).

Likewise, a Scopus search for the "world semiconductor trade statistics" yields 11 results which largely cite WSTS as an authority for market data or forecasts. However, none of these use WSTS data as a basis for industry wide forecasts, nor do they assess the accuracy of WSTS' forecasts.

More related results can be found in the field of company-specific sales and demand forecasting and operational planning. Several contributions in this domain are summarized in Table 1. However, it should be noted that, while these studies often analyze different product or technology groups, the analyses are specific to one or several companies and usually do not encompass the whole semiconductor market with its different levels of granularity.

Another related field is the prediction of the semiconductor cycle. For a short summary, see Table 2. The semiconductor cycle, similarly to the economic cycle, describes cyclical fluctuations in semiconductor industry. These cycles are characterized by growth and contraction phases. Contributions in this domain usually focus on the overall semiconductor market, particularly WSTS global semiconductor sales, as the target variable. However, total semiconductor sales are usually considered in this scenario while a break down into finer product categories is often of interest for analysts and internal benchmarking.

2.3. Contributions

(1) **Evaluation of the Accuracy of Expert Semiconductor Market Forecasts:** Despite the frequent reliance on expert forecasts for semiconductor market forecasts, the accuracy of these forecasts has not, to our knowledge, been openly or systematically evaluated. This study addresses this gap by providing an evaluation of short-term forecasts through a comparative analysis with forecasts derived from several statistical and machine learning models.

(2) Comprehensive, Multi-Granularity Forecasting of the Semiconductor Market: While recent studies have focused on granular demand and sales forecasts for specific products or companies, they do not comprehensively cover the semiconductor market across all segments and as a whole, see Table 1 in Section 2.2 for an overview. Additionally, research into the semiconductor cycle often concentrates on high-level market trends and the identification of leading indicators, see Table 2 in Section 2.2. This study, however, provides a comprehensive analysis of the semiconductor market across various levels of granularity, utilizing the WSTS product categorization hierarchy. Forecasts are systematically generated for 110 product categories, covering the entire semiconductor market. This approach captures higher-granularity product groups as well as broader market trends, offering a comprehensive perspective that takes a step towards unifying these two areas of research.

(3) Guidance for Forecasting Practitioners through Model Performance Insights: In addition to forecasting across various levels of the semiconductor market, this study offers valuable guidance for forecasting practitioners. Out-of-sample error measures for each statistical and machine learning model are presented for all 110 product categories. These performance insights allow practitioners to assess the accuracy and applicability of different models in forecasting specific segments of the semiconductor market. To the best of our knowledge, providing such detailed model performance data for a wide range of product categories is a novel contribution.

3. Used data-driven methods

This section gives a brief introduction to the data-driven models used in the subsequent analysis. It starts with the description of traditional models based on statistical time series analysis (Section 2.1), continues with ML methods (Section 2.2), and concludes with a brief note on ensemble methods (Section 2.3).

The selection of the models presented here is partially influenced by their performance in the Makridakis Competitions (M-Competitions), which are renowned forecasting competitions conducted on diverse and realistic datasets (Hyndman, 2020). The results based on the M3

Table 2
Selection of related works in semiconductor cycle prediction including goals, response variables as well as utilized forecasting methods and indicators.

Authors	Goal	Methods	Response variable	Indicators[g]
Aubry and Renou-Maissant (2014)	Identification of best model for the semiconductor cycle prediction	ARMA, VAR[a], BVAR[b], VECM[c], MRSM[d], SF[e], ES[f]	WSTS global semiconductor sales (Jan 1991–Jun 2010)	SOX, NI, TI, BOOK
Aubry and Renou-Maissant (2013)	Prediction and description of the global semiconductor industry cycle	VECM[c]	WSTS global semiconductor sales (Jan 1991–Jun 2010)	SOX, NO, TI, IP, BOOK
Chow and Choy (2006)	Identification of leading indicators of semiconductor sales to predict the global semiconductor cycle	VAR[a], BVAR[b], BECM[c]	World semiconductor sales (Feb 1992–Jan 2005)	NASDAQ, NO, SI, PPI
Liu and Chyi (2006)	Prediction of the semiconductor cycle turning points	MRSM[d]	WSTS global semiconductor sales growth (Jan 1990–Aug 2003)	–
Liu (2005)	Identification of explanatory factors for the semiconductor cycle	VAR[a]	WSTS global semiconductor sales growth (Jan 1990–Dec 2001)	IP, FF, CS, SOX, NO, TI, UTL, EQO, CAP, PPI, SIP, VS

[a] Vector autoregression.
[b] Bayesian vector autoregression.
[c] Vector error correction models.
[d] Markov regime switching model.
[e] Spectral forecasting.
[f] Exponential smoothing.
[g] SOX: Philadelphia Semiconductor Index, IP: U.S. Industrial Production, FF: Federal Funds Rate, CS: U.S. Consumer Sentiment, NO: New Orders, TI: Total Inventories, UTL: Capacity Utilization, EQO: North American Equipment Orders, CAP: Capacity, PPI: Producer Price Index, SIP: Industry Production Index, VS: Value of Shipments, BOOK: Global Bookings of N.A. Semiconductor Equipment Producers, NASDAQ: NASDAQ Stock Index, SI: U.S. Shipments to Inventories Ratio.

competition are of particular interest: the dataset comprised 3003 individual time series with 14 to 126 observations featuring various levels of seasonality (Hyndman, 2020; Makridakis & Hibon, 2000).

Furthermore, this dataset has served as a benchmark for evaluating popular data-driven forecasting methods. Both Ahmed et al. (2010) and Makridakis et al. (2018) utilized a subset of the M3 dataset (consisting of 1045 time series) with a minimum length of 81 observations for their analyses.

3.1. Statistical models

Comparisons based on the M3-Competition data have generally favored statistical models over ML approaches (Hyndman, 2020; Makridakis et al., 2018). One suggested reason for this trend is the relatively short length of the time series involved. Unfortunately, this limitation is common in forecasting applications and reflects a realistic constraint. The time series examined in this paper, ranging from 92 to 392 monthly observations (further details in Section 4.1), are longer compared to those studied in the papers based on the M3-Competition, where lengths typically spanned from 81 to 126 observations. Nevertheless, these lengths are still comparable, especially when contrasted with datasets such as the M5 competition, which feature significantly longer time series, reaching up to approximately 2000 observations (Makridakis et al., 2022b).

3.1.1. SARIMA

The **s**easonal **a**uto**r**egressive **i**ntegrated **m**oving **a**verage (ARIMA) model is a traditional statistical time series model. Its advantages are its interpretability, its wide spread use and that many of its mathematical properties are well known (Brockwell & Davis, 2002). According to Brockwell and Davis (2002), a time series $X = \{X_t\}$ is said to be a SARIMA$(p,d,q)\times(P,D,Q)$ process with period s if

$$Y_t := (1-B)^d\,(1-B^s)^D\,X_t$$

is an causal ARMA process defined as

$$\phi(B)\Phi(B^s)\,Y_t = \theta(B)\Theta(B^s)\,Z_t,$$

where B is the back-shift operator defined as $BY_t = Y_{-1}$, $\phi(z) = 1-\phi_1 z-\cdots-\phi_p z^p$, $\Phi(z) = 1-\Phi_1 z-\cdots-\Phi_P z^P$, $\theta(z) = 1+\theta_1 z+\cdots+\theta_q z^q$, $\Theta(z) = 1+\Theta_1 z+\cdots+\Theta_Q z^Q$, and $Z = \{Z_t\}$ being a white noise process.

Generally, an ARMA(p,q) process $Y = \{Y_t\}$ is characterized as

$$Y_t - \sum_{i=1}^{p}\phi_i Y_{t-i} = Z_t + \sum_{i=1}^{q}\theta_i Z_{t-i},$$

with $Z = \{Z_t\}$ being a white noise process. The left-hand side of this equation is the autoregressive part, while the right-hand side is the moving average part (moving average of the error process Z). More details on the SARIMA model can be found in Brockwell and Davis (2002).

These models are often used to describe and to generate data of a wide range of processes. But they can also be used as a predictive model when the parameters are estimated. To this end, we use the `auto.arima` function of the `forecast` library in R (Hyndman & Khandakar, 2008). A similar implementation for Python is available through the `StatsForecast` library (Garza et al., 2022).

The inclusion of the SARIMA model in this work is motivated by its ubiquity in time series analysis and its strong performance on the M3-Competition dataset (Makridakis et al., 2018).

3.1.2. Simple exponential smoothing

Exponential smoothing models range back to the 1950's (Gardner, 1985). Despite their simplicity, they often achieve high predictive performance (Hyndman, 2001; Satchell & Timmermann, 1995). Simple exponential smoothing (SES) only requires two quantities: the initial forecast $\hat{X}_0$ and the smoothing constant α. Consecutive forecasts can then be calculated via

$$\hat{X}_t = (1-\alpha)\,\hat{X}_{t-1} + \alpha X_{t-1},$$

where $\hat{X}_t$ denotes the one-step forecast for X_t based upon the history up to X_{t-1}. An R implementation is available with the SES function of the `forecast` library (Hyndman & Khandakar, 2008). A similar implementation for Python is available through the `StatsForecast` library (Garza et al., 2022).

SES' simplicity, ease of implementation, and computational efficiency make it a popular forecasting tool for practitioners. Additionally, the model performed well on the M3 and M5 Competition datasets (Makridakis et al., 2018, 2022a).

3.1.3. Error, trend, and seasonality

Error, Trend, and Seasonality (ETS) approaches are a flexible class of exponential smoothing models that go beyond SES (see above). As their name suggests, they are capable of modeling time series with trends and seasonality (Hyndman & Athanasopoulos, 2018). ETS was the best performing model in the (Makridakis et al., 2018) comparison based on the M3-Competition data. It is also implemented as part of the `forecast` library in R (Hyndman & Khandakar, 2008). As for the previous two models, a similar implementation for Python is available through the `StatsForecast` library (Garza et al., 2022).

3.2. ML models

This subsection introduces the used ML models. Makridakis et al. (2018) found that ML methods performed worse than classical statistical models for relatively short time series - a finding that was confirmed by Cerqueira et al. (2022). This is particularly the case for artificial neural networks and deep learning models, which are well known to require large sample sizes to produce the desired results (Goodfellow et al., 2016). This was also verified by the NN3-Competition, which extended the M3-Competition to include neural network approaches (Crone et al., 2011; Hyndman, 2020). Therefore, following (Cerqueira et al., 2022), this paper does not discuss neural network models despite their considerable popularity in recent years. Likewise, boosting models are not included.

3.2.1. Random forest

Random forests (RF) are a bagging algorithm, a specific kind of ensemble learning, which combines the outputs of multiple decision trees (Breiman, 2001). Ahmed et al. and Makridakis et al. included Categorization and Regresssion Trees (CART), which generate single decision trees for regression or classification purposes (Breiman, 1984). However, since its introduction, RF has proven to be an incredibly versatile and successful model for both regression and classification (Biau & Scornet, 2016; Grinsztajn et al., 2022; Huang et al., 2020). We use the `ranger` implementation of this model as provided by Wright and Ziegler (2017). An alternative Python implementation is available through the `skranger` library. During each CV-step (see below), a grid search was conducted to tune the three hyper-parameters (Probst et al., 2019):

$$\begin{aligned} mtry &\in \{2, 7, 12, 16, 23\} \\ min.node.size &\in \{5, 7, 10\} \\ splitrule &\in \{variance, extratrees\}. \end{aligned}$$

3.2.2. Extremely randomized trees

Extremely Randomized Trees (ExtraTrees, also referred to as ET for brevity) is a model similar to Random Forest (see above). The difference lies in randomizing the splitting point and the feature to split on during training. It is computationally more efficient and promises greater accuracy on a range of problems (Geurts et al., 2006). The `ranger` library is also used for this model (Wright & Ziegler, 2017). However, in contrast to the RF model, the parameter `splitrule` remained fixed as `extratrees`. A Python implementation is available through `sklearn` (Pedregosa et al., 2011). During each CV-step, a grid search was conducted to tune the hyper-parameters (Probst et al., 2019):

$$\begin{aligned} mtry &\in \{2, 7, 12, 16, 23\} \\ min.node.size &\in \{5, 7, 10\}. \end{aligned}$$

3.2.3. Gaussian processes regression

Gaussian Processes Regression (GPR) is a probabilistic regression model incorporating Bayesian ideas: A prior distribution of possible regression functions is narrowed down as evidence (observed data points) are incorporated to yield a posterior distribution (Wang, 2023). It has been shown that GPR can be viewed as a limit of many artificial neural network designs and ARMA processes can be viewed as a Gaussian process under the right conditions (Williams & Rasmussen, 1995). Furthermore, due to their probabilistic nature, GPR easily provides uncertainty quantification for the forecasts in terms of prediction intervals. GPR showed a promising performance on the M3-Competition dataset (Ahmed et al., 2010; Makridakis et al., 2018). This analysis uses the GPR implementation of the `kernlab` R library (Karatzoglou et al., 2004). The model is also implemented for Python in the `sklearn` library (Pedregosa et al., 2011).

3.2.4. K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a popular non-parametric model classification and regression model. It bases estimates on the K nearest neighbors in the covariate space (Cover & Hart, 1967; Fix & Hodges, 1989). K represents a hyper-parameter to be tuned. For regression, a mean of these K nearest neighbors is usually used as the predictor in regression. This model can be employed with a kernel. In the following, the kernel is automatically chosen. While KNN has not been among the best performing models on the M3-Competition data (Ahmed et al., 2010; Makridakis et al., 2018), it is nevertheless popular as simple a non-parametric model. Here, the implementation of the `kknn` library is used (Schliep & Hechenbichler, 2016). KNN is implemented for Python as the `NearestNeighbors` model in the `sklearn` library (Pedregosa et al., 2011). During each CV-step, a grid search was conducted to tune hyper-parameters among:

$$\begin{aligned} K &\in \{1, 2, 3, 4, 5, 7, 9\} \\ distance &\in \{L1, L2, L3\}, \end{aligned}$$

where the $L1, L2, L3$ distances are given by

$$d_{L^p}(x, y) := \|x - y\|_p$$

given that $\|x\|_p := \left(\sum_{i=1} x_i^p\right)^{1/p}$ is the $\mathcal{L}^p$ norm.

3.2.5. Support vector regression

Support Vector Regression (SVR) was proposed as an extension to the classical Support Vector Machine (SVM) for classification (Cortes & Vapnik, 1995) to tackle regression problems (Drucker et al., 1996). Instead minimizing all residuals, such as in ordinary least square regression, the distance of observations outside a margin of error to this margin of error (the ϵ-insensitive tube) are minimized. Analogous to SVMs, these observations are called support vectors, because the regressor only depends on these observations. To model non-linear dependencies, kernels can be used (Awad & Khanna, 2015). We use the radial kernel to add another non-linear method. The used model is implemented in the `kernlab` library for R or the `sklearn` library for Python (Karatzoglou et al., 2004; Pedregosa et al., 2011). A grid search was conducted to tune hyper-parameters among:

$$\begin{aligned} \sigma &\in \left\{\frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1\right\} \\ C &\in \left\{\frac{1}{4}, \frac{1}{2}, 1, 2, 4\right\}. \end{aligned}$$

3.3. Ensemble

Ensemble models consist of several individual models which are combined to produce a single output (Opitz & Maclin, 1999). In addition to the tree-based ensembles Random Forests and ExtraTrees, this paper also analyzes a simple ensemble of all the employed data-driven models by taking the median prediction.

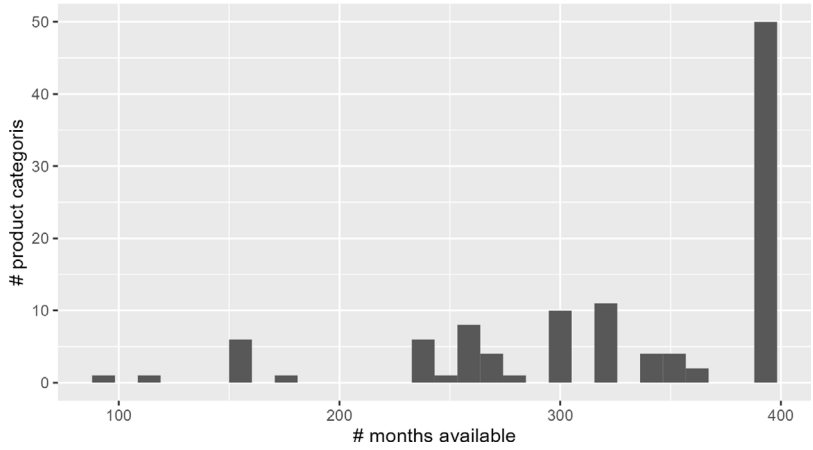


Fig. 1. Histogram of time series lengths.

4. Experimental setup

This section provides an overview over the data used in the analysis (Section 4.1) and the methodology behind the time series training and model evaluation (Section 4.2).

4.1. Data

This paper analyzes time series aggregated sales of 110 WSTS product categorizations, which were reported monthly and measured in USD. The data is accessible through a WSTS membership[3] or a subscription.[4] A major challenge was the consistency of the data:

Given the dynamic nature of the semiconductor market, product categorizations changed over time. The historical consistency of the current product categorizations was investigated and resolved dating back to Jan 2010 by merging the C7a and C7b product categorizations to C7 (Field Effect General Purpose Power Transistors), the P51 and P52 categorizations into P5 (Automotive and General Purpose MCU), and the L7a, L7b, L7c, L7d and L7f categorizations into L7a/b/c/d/f (Wireless Communication Total). The newest category (F10) dates back to Jan 2016. Fig. 1 provides an overview of the months of history of the categorizations as used in the analysis. These are the categories which are consistent until August 2023 and they provide a comprehensive overview of the semiconductor market. We refer to WSTS.org (2024) for an exact description of the market and each category.

Generally, the categories positioned higher in the hierarchy exhibit greater consistency. Fig. 2 illustrates the hierarchical structure of the product categorizations (starting with T99 as the highest aggregation). To effectively incorporate seasonal components, model fitting necessitated a minimum of 24 months (two seasonal cycles worth) of training data. The shortest time series comprised 92 monthly data points, thus leaving 68 months (or about 17.4% of the complete time series from 1991) as a test set for the first step of the rolling time series cross-validation (CV, see below). Hence, the first training set spanned all data from January 1991 (or whenever available) to December 2017. Consequently, the test set spanned the time frame from January 2018 to August 2023.

Official forecasts from WSTS were released quarterly from Q1 2016 to Q3 2023 (midway through the first forecasted quarter). Expert forecasts are consolidated during a global WSTS meeting twice a year – each May and November. WSTS additionally issues forecast updates semiannually, in February and August, derived from the preceding meeting's expert forecasts and updated algorithmically with new data reported for the prior quarter. For example, the forecast for Q2 2024 relies on the upcoming global WSTS meeting scheduled for May 20th–23rd, while the February 2024 forecast update drew upon forecasts from the November 2023 meeting and the reported data from Q4 2023. Thus, 11 expert forecasts and 12 updated forecasts were considered in the analyzed time span from January 2018 to August 2023.

4.2. Methodology

Training and evaluating the ML models: As described in the last subsection, each time series is split into first training and test sections. Time series with a longer available history consequently have more data points for training than shorter time series, i.e. newer product categories.

To obtain reliable forecast performance estimates for all of them, rolling time series cross-validation (CV) as in Hyndman and Athanasopoulos (2018) is performed on each time series and for each model. This is illustrated in Fig. 3, which also shows the most extreme training periods for the different time series (only 24 months for the first forecast of category F10 up to 390 months for the last forecast of T99).

Within each iteration, the training data is automatically transformed with the Box–Cox transformation (Box & COX, 1964) given by

$$X_t^{(\lambda)} = \begin{cases} (X_t^{\lambda} - 1)/\lambda & \lambda \neq 0, \\ \log(X_t), & \lambda = 0. \end{cases}$$

This transformation is incorporated and automatically estimated by the used libraries "forecast" and "caretForecast" (Akay, 2022; Hyndman & Khandakar, 2008). Applying the Box–Cox transformation is standard practise, especially when residual distributions are skewed, and when non-negative forecasts are desired (Hyndman & Athanasopoulos, 2018). The considered time series report aggregated sales in USD. Thus, there is unlimited upside potential whereas the lower bound is always zero since all time series are positive. Note that λ can always be chosen close to one if the transformation is not particularly helpful. Automatically applying it to all cases therefore does not hurt. This is also the default setting in the `forecast` library (Hyndman & Khandakar, 2008).

Additionally, hyper-parameters of all data-driven models introduced in Section 3 are optimized using the default grid search setting of the "caret" and "caretForecast" libraries (Akay, 2022; Kuhn, 2008) if no hyper-parameter optimization is conducted through the learning algorithm (one example where hyper-parameters are tuned automatically is the SARIMA model using the "auto.arima" function). After each training iteration, a forecast is generated up to three months in advance. These forecasts can be compared against the reported numbers, providing an estimate of the performance of the model, and the WSTS forecasts.

[3] https://www.wsts.org/61/membership.
[4] https://www.wsts.org/61/subscription.

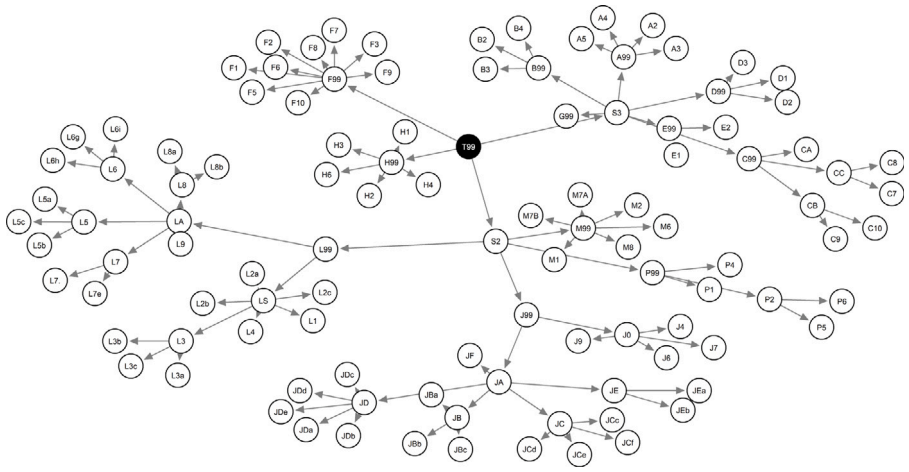


Fig. 2. WSTS product categorization hierarchy. The highest aggregation level is T99, the node colored in black with white print, slightly to the top of the center of the illustration. The arrows point to the subsumed product categories.

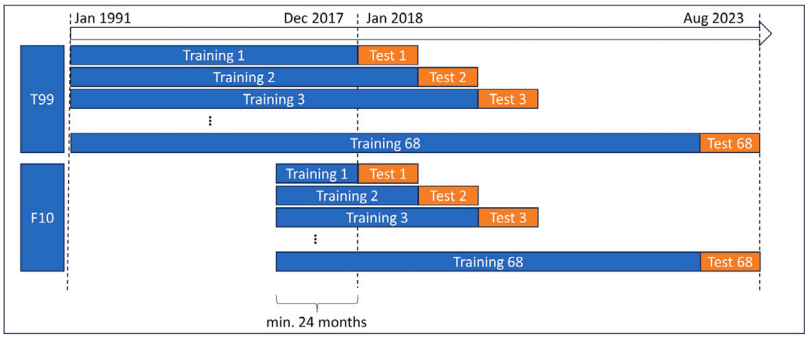


Fig. 3. Illustration of time series cross validation for two product categories with differing lengths: T99 has a much longer history of reported aggregate sales than F10.

Evaluation and comparison with WSTS' forecasts: The forecasts provided by WSTS are evaluated by type ("meeting" corresponding to the two WSTS expert forecast per year, "alg. update" corresponding to the two WSTS algorithmic forecasts per year and "overall" for all four WSTS forecasts per year) and compared to the corresponding data-driven forecasts. The first comparison evaluates all forecasts with a forecasting horizon of $h = 3$ months. This corresponds to the information timestamp available to WSTS' updating algorithm and the industry experts.

At the same time, it is safe to assume that the industry experts have access to the sales data (among many more) of the first month of each quarter when the meeting convenes (the meeting is held in about the middle of the first quarter to be forecasted). Additionally, the forecasts are disclosed at the same time as the first month's results are published. Hence, the information time stamp of the forecast and the first month's results is the same. Utilizing this information reduces the required forecast horizon to $h = 2$ months. This is analyzed in a second step to investigate whether using more of the available information might boost the forecasting accuracy.

Similar to Hyndman and Koehler (2006) and Pauly and Kuhlmann (2023), the forecasting accuracy is evaluated using the mean squared error (MSE), mean absolute percentage error (MAPE), and the mean absolute error (MAE). These are given by the following equations

$$MSE = \frac{1}{T}\sum_{t=1}^{T}(\hat{X}_t - X_t)^2$$

$$MAPE = \frac{1}{T}\sum_{t=1}^{T}\left|\frac{\hat{X}_t - X_t}{X_t}\right|$$

$$MAE = \frac{1}{T}\sum_{t=1}^{T}|\hat{X}_t - X_t|,$$

where again $\hat{X}_t$ is the one-step forecast for the tth observation X_t of the test set and $T = 68$ is the number of evaluated forecasts. We note that the MAPE is applicable as the values of all time series are positive.

Table 3 provides a brief overview over the key methods which are employed in this study and provides references to motivating studies and deeper methodological discussions.

5. Results

As discussed in Section 4.2, first, the quarterly forecasts ($h = 3$ months) are discussed in Section 5.1 to examine the first research hypothesis (H1) that expert forecasts exhibit higher accuracy compared to data-driven forecasts. This is followed by a comparison with the model performance when an additional month of available information ($h = 2$ months) is incorporated in Section 5.2, addressing the second research hypothesis (H2). Lastly, the results are contrasted by time series length

5.1. Quarterly forecast performance

WSTS' forecasts are provided on a quarterly basis. Each quarter, the previous quarter's numbers are known when the forecasts are compiled. Therefore, as a first step, the data-driven models are compared against WSTS' forecasts on a forecasting horizon of $h = 3$ months, i.e. one quarter.

Table 4 presents the average performance of data-driven forecasts across 110 product categories, relative to the average performance of forecasts provided by the World Semiconductor Trade Statistics (WSTS). Hence, the first data column (for WSTS) always reads 1. Each row represents a different error measure, organized according to

Table 3
Overview of used methods: Purposes and motivations for chosen techniques, with references.

Methods	Purpose and motivation	Authors
SARIMA, SES, ETS, RF, ET, GPR, KNN, SVR	Purpose: generating point forecasts. Motivation: the selection of these models was largely motivated by studies based on the M3-Competition dataset. These models achieved outstanding performance on time series similar to the ones in this study.	Ahmed et al. (2010) and Makridakis et al. (2018)
Cross-Validation	Purpose: estimating out-of-sample performance estimates. Motivation: The application of time series cross validation is a standard procedure.	Cerqueira et al. (2020)
MSE, MAPE, MAE	Purpose: measures for prediction accuracy. Motivation: these are standard measures commonly employed to assess the accuracy of (time series) models with continuous targets	Hyndman and Koehler (2006)

Table 4
Average performance of the data-driven forecasts across all 110 product categories and relative to the World Semiconductor Trade Statistics' (WSTS). Each row refers to a different error measure, sorted by WSTS' forecast type: algorithmic update, meeting (expert forecast), and overall. Lower values are preferable. The best value per row is bold and italic.

		WSTS	SARIMA	ETS	ET	GPR	KNN	RF	SES	SVM	Ensemble
Alg. Update	MSE	1.00	***0.34***	0.55	1.55	0.37	4.47	1.49	0.94	3.20	0.64
	MAPE	***1.00***	1.01	1.01	1.20	1.01	1.71	1.18	1.06	1.69	1.04
	MAE	1.00	***0.75***	0.81	1.34	0.77	2.14	1.29	1.08	1.94	0.95
Meeting	MSE	***1.00***	1.61	1.57	2.78	1.33	7.23	2.49	1.79	6.91	1.77
	MAPE	***1.00***	1.14	1.09	1.32	1.13	1.76	1.28	1.10	1.82	1.14
	MAE	***1.00***	1.22	1.22	1.53	1.17	2.32	1.47	1.24	2.32	1.28
Overall	MSE	1.00	0.73	0.86	1.93	***0.66***	5.31	1.79	1.20	4.33	0.98
	MAPE	***1.00***	1.08	1.05	1.26	1.07	1.73	1.23	1.08	1.75	1.09
	MAE	1.00	0.97	1.00	1.42	***0.95***	2.22	1.37	1.15	2.11	1.10

WSTS' forecast types: algorithmic update, meeting (expert forecast), and overall. Lower values are preferable in all cases.

For WSTS' "Algorithmic Update" forecasts, three error measures are reported: Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), and Mean Absolute Error (MAE) – see Section 4.2. Among these measures, the data-driven forecasts outperform WSTS in terms of MSE and MAE, with the best-performing model indicated by bold and italic formatting. The SARIMA model exhibits the lowest MSE (0.34 relative to WSTS) and MAE (0.75), suggesting superior performance in this category. Almost as good was the GPR model (relative MSE of 0.37 and MAE of 0.77). In term of MAPE, WSTS performed slightly better than several data driven forecast methods: SARIMA, ETS, and GPR each scored 1% worse. Overall, these results indicate potential for improvement of WSTS' algorithmic update protocol.

Similarly, for the "Meeting" forecasts, the same three error measures are provided. Consistent with the first research hypothesis (H1), industry experts demonstrated superior performance across all three error measures. However, among the data-driven models, the best performers were GPR, exhibiting a 33% increase in MSE compared to expert forecasts, while ETS and SES displayed 9% and 10% higher MAPE values respectively. Additionally, GPR, again, was the best performing data-driven model in terms of MAE, showing a 17% higher MAE relative to the expert forecasts.

Table 4 concludes with the "Overall" forecasts, providing the average performance of all models relative to the average performance of the combined algorithmic and expert forecasts by WSTS (from 2 meeting and 2 algorithmic forecast per year). Once more, the GPR model emerges as the top-performing data-driven model, showing a 34% improvement in MSE and a 5% enhancement in MAE compared to WSTS. However, WSTS outperforms all models in terms of MAPE, with ETS, the top-performing data-driven model, recording a 5% higher error than WSTS.

Table 5 provides insights into the mean ranks of the various forecasting models across all 110 product categories, structured similarly to Table 4. Contrary to Table 4, the columns in Table 7 are not standardized by the WSTS column. A rank of 1 indicates the model performed the best for that product category based on the respective error measure, thus lower mean ranks are preferred.

In general, the observations from Table 7 are similar to those from Table 6. The forecasts provided by WSTS demonstrate excellence across most error metrics and scenarios. However, there is an exception with WSTS' Algorithmic Update forecasts, where SARIMA and GPR achieved slightly lower average ranks (3.8 vs. 3.9 for WSTS). Considering that the average MSE for the forecasts based on the SARIMA model was 66% lower than the MSE of WSTS' algorithmic updates, it suggests SARIMA's strong performance is primarily driven by specific product categories where it outperforms WSTS (Section 5.4 contains a detailed discussion on individual product categories). Although several data-driven models exhibited a superior average error in terms of MSE and MAE for the overall forecasts (for example, SARIMA and GPR) this superiority does not necessarily translate to lower average ranks, highlighting WSTS' robust performance across product categories.

Comparing only the data-driven forecasts, GPR emerges as the most accurate data-driven model in the Algorithmic Update and Overall scenarios, while Simple Exponential Smoothing (SES) demonstrates the best performance among data-driven models during quarters where WSTS provided expert forecasts (Meeting), particularly in terms of MSE and MAE. ETS achieved a slightly lower average rank in terms of Mean Absolute Percentage Error (MAPE).

One plausible explanation for the strong performance of the industry experts is their access insider information. In contrast, the data-driven models rely solely on aggregated historical sales data from the specific product category being forecasted. Another factor to consider is the timing of the WSTS meetings where expert forecasts are consolidated. These meetings typically occur in the middle of the forecasted quarter, implying that experts are likely aware of their first-month figures. Moreover, these figures are available simultaneously with the official forecast release. In contrast, the bi-quarterly algorithmic updates do not integrate this information, though they could potentially benefit from it. Consequently, data-driven models that incorporate this timely information, necessitating forecasts with a horizon of only $h = 2$ months, are evaluated in the subsequent subsection.

5.2. Forecast performance with additional information

Considering the timing of the WSTS meetings in the middle of the quarter, it is reasonable to assume that industry experts take into account the numbers and internal information pertaining to the first month when formulating their forecasts for the first quarter. Moreover, the consolidated results of the first month are released simultaneously with the forecasts by WSTS. Therefore, utilizing all available information for forecasts seems appropriate. This approach allows for an

Table 5
Mean ranks of the forecasts with horizon h = 3 months across all 110 product categories. Each row refers to a different error measure, sorted by WSTS' forecast type: algorithmic update, meeting (expert forecast), and overall. Lower values are preferable. The best value per row is bold and italic.

		WSTS	ARIMA	ETS	ET	GPR	KNN	RF	SES	SVM	Ensemble
Alg. Update	MSE	3.9	***3.8***	4.2	6.6	***3.8***	8.7	6.2	4.4	8.8	4.5
	MAPE	***3.6***	4.1	4.1	6.6	3.9	8.8	6.1	4.5	8.7	4.5
	MAE	***3.6***	4.1	4.2	6.6	3.9	8.8	6.1	4.4	8.9	4.5
Meeting	MSE	***2.9***	4.5	3.9	7.0	4.1	8.9	6.4	3.8	8.9	4.6
	MAPE	***2.8***	4.3	4.0	7.0	4.2	8.9	6.4	4.1	8.8	4.5
	MAE	***2.9***	4.3	4.0	7.2	4.2	8.9	6.4	3.8	8.8	4.5
Overall	MSE	***3.2***	4.4	4.2	6.9	3.7	9.0	6.3	4.1	8.9	4.4
	MAPE	***2.8***	4.1	3.9	7.0	3.9	9.1	6.5	4.2	8.9	4.5
	MAE	***2.9***	4.2	4.0	7.0	4.0	9.1	6.4	4.1	8.8	4.5

Table 6
Average performance of the data-driven forecasts with horizon $h = 2$ months across all 110 product categories and relative to WSTS. Each row refers to a different error measure, sorted by WSTS' forecast type: algorithmic update, meeting (expert forecast), and overall. Lower values are preferable. The best value per row is bold and italic.

		WSTS	SARIMA	ETS	ET	GPR	KNN	RF	SES	SVM	Ensemble
Alg. Update	MSE	1.00	***0.10***	0.18	0.52	***0.10***	1.31	0.49	0.46	1.29	0.32
	MAPE	1.00	0.62	0.62	0.76	***0.61***	1.09	0.75	0.68	1.08	0.65
	MAE	1.00	***0.42***	0.51	0.81	0.43	1.31	0.79	0.77	1.24	0.63
Meeting	MSE	1.00	0.68	0.66	1.41	***0.64***	3.65	1.25	0.96	3.18	0.91
	MAPE	1.00	***0.61***	0.62	0.82	0.65	1.15	0.79	0.65	1.15	0.68
	MAE	1.00	***0.72***	0.73	1.05	0.75	1.54	1.00	0.88	1.46	0.85
Overall	MSE	1.00	0.28	0.32	0.80	***0.27***	2.03	0.72	0.61	1.86	0.50
	MAPE	1.00	***0.61***	0.62	0.79	0.63	1.12	0.77	0.66	1.12	0.66
	MAE	1.00	***0.56***	0.61	0.92	0.58	1.41	0.89	0.82	1.34	0.73

additional month of data to forecast the quarterly result, rendering a forecasting horizon of $h = 2$ months sufficient. Considering that one out of three months' numbers do not require estimation, a plausible anticipation would be to observe approximately a 33% reduction in MSE (assuming an unbiased estimator and uncorrelated errors). Such a decrease could already position several data-driven methods as competitive alternatives to WSTS, which is examined in this section.

The average performance of these forecasts relative to WSTS' is presented in Table 6, structured equivalently to Table 4. Table 6 presents the average performance of data-driven forecasts with a horizon of $h = 2$ months across 110 product categories, relative to the forecasts provided by WSTS. Each row represents a different error measure, categorized by WSTS' forecast types: algorithmic update, meeting (expert forecast), and overall. As before, lower values indicate better performance, with the best value per row highlighted in bold and italic.

Contrasting these results with those presented in previous table (Table 4), several notable differences emerge. Firstly, upon a cursory glance of the results, it becomes evident that the data-driven approaches have exhibited markedly superior performance in this context. Whereas WSTS' expert forecasts (Meeting) previously outperformed the data-driven forecasts in terms of MSE, MAPE, and MAE, the tables have now turned, with the data-driven forecasts consistently showcasing superior forecast accuracy in the new scenario (with the exception of SVM and KNN for all as well as ET and RF for the Meeting MSE and MAE comparisons). Specifically, the GPR model achieved a 36% lower MSE than WSTS' expert forecasts, followed by ETS (34% lower) and SARIMA (32% lower). In terms of MAPE, SARIMA outperformed WSTS' experts by 39%, followed by ETS (38% lower), and GPR and SES (both 35% lower). Additionally, SARIMA demonstrated the best performance in terms of MAE (28% lower than WSTS), trailed by ETS (27% lower) and GPR (25% lower). Even the simple ensemble, which even incorporates the forecasts of the worse performing models, surpassed WSTS' experts by 9% in MSE, 32% in MAPE, and 15% in MAE. This suggests that data-driven models incorporating the latest available information are highly effective in forecasting outcomes within a shorter horizon. Furthermore, WSTS' expert forecasts attained superior average performance in terms of MAPE across all three forecast types with a forecasting horizon of $h = 3$ months. However, with a reduced horizon of $h = 2$ months, the top-performing data-driven forecasts now outperform WSTS by up to 39%. ARIMA and ETS emerged as the top performers, closely followed by GPR.

Secondly, concerning the algorithmic updates, according to Table 6, SARIMA and GPR once again emerge as one of the top-performing models. In MSE, both SARIMA and GPR achieved errors 90% lower than WSTS. Additionally, in terms of MAE, SARIMA attained a 58% lower error, closely followed by GPR with a 57% reduction. While in terms of MAPE, where WSTS previously outperformed data-driven models, GPR achieved a 39% lower error, with SARIMA and ETS achieving a 38% lower MAPE.

Table 7 offers insights into the mean ranks of the various forecasting models across all 110 product categories, organized similarly to Tables 4 and 6. It is important to note that, in contrast to Tables 4 and 6, the columns in Table 7 are not standardized by the WSTS column. A rank of 1 indicates the model performed the best for that product category based on the respective error measure, thus lower mean ranks are preferred. Overall, the observations from Table 7 parallel those from Table 6. Models such as SARIMA, ETS, and GPR consistently garnered high ranks. Notably, most data-driven methods outperformed WSTS' expert forecasts, except for KNN and SVM, which exhibited poorer performance.

Finally, Fig. C.4 in the Appendix illustrates the frequency of the best-performing forecasts across 110 WSTS product categories, with colors indicating performance metrics: red for Mean Squared Error (MSE), blue for Mean Absolute Percentage Error (MAPE), and green for Mean Absolute Error (MAE). The left panel reflect the $h = 3$ months forecast while $h = 2$ is presented on the right side. It is evident that WSTS forecasts rarely emerge as the top performers within any given product category for $h = 2$. While WSTS' expert forecasts generally outperform its algorithmic updates, data-driven models consistently outshine both. When comparing against WSTS' expert forecasts (second row), SARIMA emerges as the best performer between 20% (MSE) and 30% (MAE) of the time, followed by ETS between 19% (MSE) and 27% (MAPE). Additionally, GPR and SES each excel between 13% (MAE and MAPE) and 17% and 18% (MSE), respectively. In contrast, WSTS' expert forecasts demonstrate the best performance between 7% (MAE and MAPE) and 13% (MSE) of the time.

Table 7
Mean ranks of the forecasts with horizon h = 2 months across all 110 product categories. Each row refers to a different error measure, sorted by WSTS' forecast type: algorithmic update, meeting (expert forecast), and overall. Lower values are preferable. The best value per row is bold and italic.

		WSTS	SARIMA	ETS	ET	GPR	KNN	RF	SES	SVM	Ensemble
Alg. Update	MSE	8.0	3.6	3.7	5.8	***3.3***	8.4	5.7	3.9	8.7	3.7
	MAPE	8.2	***3.4***	3.7	5.8	3.6	8.6	5.6	4.1	8.4	3.5
	MAE	8.0	***3.4***	3.8	5.9	3.5	8.6	5.7	4.1	8.5	3.6
Meeting	MSE	6.9	3.5	***3.3***	6.5	3.7	8.9	5.9	3.5	8.5	4.2
	MAPE	7.2	***2.9***	3.4	6.7	3.4	8.8	6.2	3.7	8.6	4.2
	MAE	7.0	***2.9***	3.5	6.8	3.5	8.8	6.2	3.7	8.5	4.1
Overall	MSE	7.8	***3.3***	3.4	6.3	***3.3***	8.8	5.9	3.6	8.7	3.8
	MAPE	8.0	***2.9***	3.2	6.4	3.3	9.0	5.9	3.8	8.7	3.7
	MAE	7.8	***2.9***	3.3	6.4	3.4	9.0	6.0	3.8	8.6	3.8

Table 8
Overview of the time series lengths by category.

	Count	Av. Length	Min. Length	Max. Length
Long	50	392	392	392
Medium	31	324	296	359
Short	29	222	92	284
All	110	328	92	392

These findings corroborate the trends observed in the preceding analyses of this section, suggesting the reliability and effectiveness of certain data-driven models over expert forecasts in near-term forecasting scenarios when additional available information is incorporated, supporting the second research hypothesis (H2).

5.3. Impact of time series length on forecast performance

This section aims to investigate the possibility that expert forecasts perform better on shorter time series due to the limited data available for training data-driven models (Hypothesis (H3)). To explore this third hypothesis, the 110 product categories are divided into long, medium, and short categories based on the length of available observations. An overview of this categorization is presented in Table 8.

In total, the time series had an average length of 328 monthly observations. Among these, the 50 product categories with available monthly data points for the entire examined time period were classified as "long" time series. The "medium" category comprised 31 time series with observations ranging from 296 to 359, averaging 324 available data points. The remaining 29 product categories were designated as "short" time series, with observations ranging from 92 to 284, averaging 222 available data points.

Similar to Table 6, Table 9 provides an overview of the performance metrics Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), and Mean Absolute Error (MAE) for the 2-month forecasts. These metrics are segmented based on time series lengths: Short, Medium, and Long, as delineated in Table 8, and further categorized by WSTS' forecast types: algorithmic update, meeting (expert forecast), and overall.

For short time series, GPR demonstrated superior performance in terms of both MSE and MAPE, showcasing reductions of 76% and 43% respectively compared to WSTS' combined forecasts. Regarding MAPE, ETS exhibited the lowest error rate (37% lower than WSTS), trailed by SES (36% lower), SARIMA (34% lower), and GPR (33% lower). When contrasting the data-driven forecasts with WSTS' expert forecasts, ETS emerged as the top-performing model, tying with GPR in terms of MSE (both 43% lower than expert forecasts) and outperforming WSTS' experts by 32% in terms of MAE. However, SES marginally outperformed ETS in terms of MAPE, showing reductions of 35% and 34% respectively. When comparing data-driven forecasts solely to algorithmic forecasts, GPR emerged as the most accurate model across all metrics, boasting reductions of 93% in MSE, 43% in MAPE, and 58% in MAE. This explains the robust performance of GPR relative to WSTS' combined forecasts.

It is also noteworthy to highlight the disparity in the performance of data-driven methods for short time series depending on whether quarters with expert forecasts or algorithmic updates were used for the benchmarks. Given that these are derived from the same time series, a consistent ranking of data-driven methods might have been expected. The best-performing models for medium and long time series remain largely consistent, regardless of whether they are evaluated using algorithmic forecasts or expert forecasts. Even in cases where there are differences, the margin is minimal.

In the Appendix, Table A.10 outlines the mean ranks of the analyzed models, once again categorized by Mean Squared Error (MSE), Mean Average Percentage Error (MAPE), and Mean Absolute Error (MAE), and segmented by time series lengths. These rankings are further delineated by WSTS forecast type, mirroring the structure of Table 9.

Consistent with the findings in Table 9, GPR demonstrated the most favorable performance in terms of mean ranks across MSE (2.8), MAPE (2.8), and MAE (2.9) for short time series when compared to WSTS' algorithmic forecasts. However, while ETS excelled in average MSE and MAE, and SES in MAPE, this outcome is reversed when considering mean ranks: SES exhibited the best performance in terms of mean ranks for MSE and MAE, while ETS performed best for MAPE. It is worth noting that the differences between them are minimal in each case. Furthermore, ETS attained the lowest mean rank when all quarters were taken into account (overall), suggesting that exponential smoothing models perform well when data is limited. Furthermore, the high accuracy of the data-driven forecasts compared to WSTS' expert forecasts is evidence contrary to the third research hypothesis (H3) that expert forecasts would outperform data-driven forecast in the context of short time series.

In fact, a slight increase in error relative to the expert forecasts is observed for the best forecasts when medium-length time series are considered in terms of MSE (0.58 vs. 0.57). Likewise, the data-driven forecasts fared slightly worse in terms of MSE (0.64 vs. 0.57) and MAE (0.72 vs. 0.68) for long time series. Given the additional information which was utilized in training the models, the opposite might have been expected. This was the case when the forecasts were evaluated by MAPE (0.57 medium and long time series vs. 0.65 for short ones).

For medium time series, SARIMA generated the most reliable forecasts in terms of relative average error measures – 42% more accurate than the experts polled by WSTS in terms of MSE, 43% more accurate in terms of MAPE, and 36% more accurate in terms of MAE, followed by ETS. This is also reflected in the mean ranks of the SARIMA forecasts (2.9, 2.2, and 2.8 vs. 7.1, 7.8, and 7.5 respectively). Similarly, the SARIMA forecasts demonstrated superior performance in comparison to the algorithmic updates and when evaluated overall.

In the case of the long time series, the most accurate forecasts resulted from the GPR and SARIMA models (Table 9). GPR resulted in 36% lower MSE and a 43% lower MAPE compared to the experts. SARIMA excelled in terms of MAE – 38% lower than WSTS. A similar

Table 9
Average performance of the data-driven forecasts with horizon $h = 2$ months across the different time series lengths and relative to WSTS. Each row refers to a different error measure, sorted by WSTS' forecast type: algorithmic update, meeting (expert forecast), and overall. Lower values are preferable. The best value per row is bold and italic.

			WSTS	SARIMA	ETS	ET	GPR	KNN	RF	SES	SVM	Ensemble
Short	Alg. Update	MSE	1.00	0.12	0.11	0.25	***0.07***	0.58	0.22	0.29	0.60	0.14
		MAPE	1.00	0.63	0.61	0.73	***0.57***	1.08	0.72	0.63	1.07	0.63
		MAE	1.00	0.49	0.49	0.70	***0.42***	1.09	0.68	0.67	1.11	0.56
	Meeting	MSE	1.00	0.74	***0.57***	0.86	***0.57***	1.35	0.77	0.65	1.26	0.66
		MAPE	1.00	0.69	0.66	0.88	0.77	1.23	0.85	***0.65***	1.24	0.74
		MAE	1.00	0.77	***0.68***	0.89	0.74	1.23	0.85	0.72	1.20	0.76
	Overall	MSE	1.00	0.33	0.27	0.46	***0.24***	0.85	0.41	0.42	0.83	0.32
		MAPE	1.00	0.66	***0.63***	0.80	0.67	1.15	0.78	0.64	1.16	0.68
		MAE	1.00	0.63	0.58	0.79	***0.57***	1.16	0.76	0.70	1.16	0.65
Medium	Alg. Update	MSE	1.00	***0.14***	0.17	0.42	***0.14***	1.33	0.39	0.27	1.23	0.27
		MAPE	1.00	***0.53***	0.57	0.76	0.54	1.08	0.75	0.61	1.06	0.63
		MAE	1.00	***0.45***	0.48	0.81	***0.45***	1.33	0.79	0.63	1.32	0.64
	Meeting	MSE	1.00	***0.58***	0.61	1.05	0.66	2.17	0.93	0.73	2.21	0.78
		MAPE	1.00	***0.57***	0.65	0.83	0.64	1.20	0.79	0.66	1.15	0.67
		MAE	1.00	***0.64***	0.67	0.97	0.72	1.45	0.92	0.76	1.48	0.81
	Overall	MSE	1.00	***0.31***	0.34	0.66	0.34	1.66	0.60	0.45	1.61	0.47
		MAPE	1.00	***0.55***	0.61	0.79	0.59	1.14	0.77	0.63	1.10	0.65
		MAE	1.00	***0.54***	0.57	0.89	0.58	1.39	0.85	0.69	1.39	0.72
Long	Alg. Update	MSE	1.00	***0.10***	0.18	0.54	***0.10***	1.34	0.50	0.47	1.31	0.32
		MAPE	1.00	***0.67***	0.68	0.78	0.68	1.09	0.77	0.76	1.11	0.68
		MAE	1.00	***0.41***	0.52	0.83	0.43	1.34	0.81	0.80	1.25	0.64
	Meeting	MSE	1.00	0.68	0.66	1.45	***0.64***	3.80	1.28	0.98	3.29	0.93
		MAPE	1.00	0.58	0.58	0.78	***0.57***	1.07	0.75	0.64	1.09	0.64
		MAE	1.00	***0.72***	0.75	1.08	0.76	1.60	1.03	0.93	1.50	0.88
	Overall	MSE	1.00	***0.27***	0.33	0.81	***0.27***	2.08	0.74	0.63	1.91	0.51
		MAPE	1.00	***0.62***	0.63	0.78	***0.62***	1.08	0.76	0.70	1.10	0.66
		MAE	1.00	***0.55***	0.63	0.94	0.58	1.46	0.91	0.86	1.36	0.75

picture arises when mean ranks (Table A.10) are considered. An exception is the algorithmic update category, where forecasts based on the ensemble method achieved the lowest mean ranks.

Fig. C.5 in the Appendix illustrates the frequency of the best-performing forecasts with a forecast horizon of $h = 2$ months. The chart is divided into 3 columns corresponding to the time series lengths: short, medium, and long. Rows are arranged by WSTS forecast type (algorithmic update, meeting (expert), and overall). The colors differentiate between Mean Squared Error (MSE) in green, Mean Absolute Percentage Error (MAPE) in blue, and Mean Absolute Error (MAE) in red. In concordance with the analysis of Tables 9 and A.10, GPR shows the highest frequency of top model performance for short sime series when compared to the algorithmic updates. Overall and for the expert forecasts ETS and SES had the highest frequencies of highest accuracy. For the medium and long time series, SARIMA, GPR, and the exponential smoothing models excelled most often.

Additional details pertaining to the overall performance of the various data-driven models for each product category categorized by time series lengths is available in Table B.12.

5.4. Additional results

In addition the aggregated results presented in Sections 5.1–5.3, this Section elaborates the results on a product category level.

Tables B.11, B.12, and B.13 in the Appendix, provide the Root Mean Squared Errors (RMSE), which is the square root of the Mean Squared Error (MSE) described in Section 4.2 and was chosen for readability here, mean absolute percentage errors (MAPE), and Mean Absolute Errors (MAE) respectively for various forecasting models across 110 different product categories. Each time, these are organized by time series lengths, consistent with the categorization in Section 5.3. The error measures for the forecasts with horizon $h = 3$ months are presented in the first half of each table and those of forecasts with horizon $h = 2$ months are presented in the second halves. Lower RMSE, MAPE, and MAE values indicate higher forecasting accuracy, with the best performing model per product category printed in bold and italic in each row of each table.

In the 3-month forecast category, the results are consistent with those discussed in Section 5.1. As can be seen in Tables B.11 and B.13, data-driven methods, particularly GPR and SARIMA, are overall able to outperform WSTS in terms of RMSE and MAE. Table B.11 reveals that this is in large part due to the strong performance of the GPR ans SARIMA forecasts in terms of RMSE for a few product categories such as M99 (37.72 for GPR vs 46.77 for WSTS), T99 (56.83 for GPR vs. 71.85 for WSTS), and S2 (49.17 for SARIMA vs 70.79 for WSTS). Similarly, scrutinizing Table B.13 indicates that among all forecasts, WSTS' was the most reliable for most product categories in terms of MAE. But GPR, SARIMA, and ETS produced forecasts which excelled for specific product categories, such as T99 (53.76 for WSTS vs. 46.24 for SARIMA, 47.35 for GPR, and 50.85 for ETS), resulting in a higher average performance for long time series. In terms of MAPE, Table B.12 illustrates WSTS' strong performance on the $h = 3$ month horizon across all time series lengths. Nevertheless, upon scrutinizing Table B.12, it becomes apparent that data-driven models outperform WSTS' combined expert and algorithmic update forecasts in terms of MAPE for specific product categories. For instance, in categories such as J99 and L8a, the SARIMA model produces the lowest MAPE, demonstrating its effectiveness in forecasting these particular products. In some categories like L1, where the time series might exhibit unique patterns or complexities, traditional statistical models such as SARIMA and ETS perform inadequately compared to specific machine learning models, in this case: RF.

Consistent with the observations in Section 5.2, these results are reversed when the forecasts with a horizon of $h = 2$ months are considered. In this scenario, data-driven forecasts excelled across the vast majority of product categories in terms of RMSE (Table B.11), MAPE (Table B.12), and MAE (Table B.13). Nevertheless, WSTS' forecasts were superior for some select product categories with long histories in terms of RMSE, such as C7 (1.18 vs. 1.32 for the best performing data-driven model: GPR), CC (1.29 vs. 1.47 for the best performing

data-driven model: GPR), and S3 (2.77 vs. 2.94 for the best performing data-driven model: SES).

Moreover, despite the dominance of the SARIMA, GPR, and ETS among the data-driven models, it is noteworthy that other models with poorer average performance still yielded in strong forecasts for select product categories: ET was amongst the top performing models for the A5 product category in terms of RMSE and MAPE, and SVM performed best for the JCd product category in terms of RMSE and MAE. This finding is underlined by Fig. C.4 located in Appendix. The bar chart visualizes the frequency of the best-performing forecasts across 110 WSTS product categories. Differentiated by colors, green signifies the best performance based on Mean Squared Error (MSE), blue represents Mean Absolute Percentage Error (MAPE), and red indicates Mean Absolute Error (MAE). The chart is divided into two sections: the left side displays outcomes for forecasts with a horizon of $h = 3$ months, while the right side portrays forecasts with a horizon of $h = 2$ months (to be discussed in the subsequent section). Rows are arranged by algorithmic update, expert forecasts (meeting), and overall performance.

Finally, Fig. C.4 in the Appendix illustrates the frequency of the best-performing forecasts across 110 WSTS product categories, with colors indicating performance metrics: red for Mean Squared Error (MSE), blue for Mean Absolute Percentage Error (MAPE), and green for Mean Absolute Error (MAE). The left panel reflect the $h = 3$ months forecast while $h = 2$ is presented on the right side.

For $h = 3$ months the superior performance of the WSTS models is evident, particularly their Meeting and Overall forecasts. Depending on the error measure, WSTS' experts outperformed all data driven models between 46% and 48% of all product categories. The most frequent best performing data driven models were SARIMA, GPR and the exponential smoothing models, ETS and SES. In aggregate, the forecasts provided by WSTS (Overall row) achieved a top performance in 38% to 39% of the cases. In contrast, the playing field was more even when only the algorithmic updates are considered. WSTS achieved a top performance for 21% to 23% of all product categories, followed by SARIMA with 17% to 23% and GPR with 13% to 19% of all product categories.

It is evident that WSTS forecasts rarely emerge as the top performers within any given product category for $h = 2$ months. While WSTS' expert forecasts generally outperform its algorithmic updates, data-driven models consistently outshine both. When comparing against WSTS' expert forecasts (second row), SARIMA emerges as the best performer between 20% (MSE) and 30% (MAE) of the time, followed by ETS between 19% (MSE) and 27% (MAPE). Additionally, GPR and SES each excel between 13% (MAE and MAPE) and 17% and 18% (MSE), respectively. In contrast, WSTS' expert forecasts demonstrate the best performance between 7% (MAE and MAPE) and 13% (MSE) of the time.

This highlights that there is no one model for all situations but that the model choice should depend on the individual product category and the error measure with the greatest business relevance.

6. Discussion

6.1. Summary

Section 2 made the case that the semiconductor industry plays a crucial role in the broader economy and stressed the importance of reliable forecasts for operational and strategic decision making. Furthermore, the rapidly evolving technologies, complicated geopolitical considerations, and complex supply chains exposing the industry to the bullwhip effect make data-driven forecasting more challenging. Concurrently, industry insiders, such as those queried by the World Semiconductor Trade Statistics (WSTS) – a leading provider of semiconductor market data and forecasts, promise reliable forecasts based on a wealth of quantitative and qualitative insider information.

This motivated the first research hypothesis (H1) that expert forecasts exhibit higher accuracy compared to data-driven forecasts. This hypothesis was extensively examined in Section 5.1 for a forecast horizon of $h = 3$. The analysis of the benchmark concluded that the bi-quarterly expert forecasts indeed demonstrated superior accuracy on a quarterly horizon in terms of Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), and Mean Absolute Error (MAE). In contrast to the superior performance of the expert forecasts, it was also found that the bi-quarterly algorithmic forecasts provided by WSTS showed potential for further improvement.

Furthermore, it was noted that industry insiders may have access to additional information owing to the timing of WSTS meetings, during which the forecasts are formulated. This observation prompted the formulation of the second research hypothesis (H2) that the additional information yields competitive data-driven forecasts, which was investigated in Section 5.2. It was found that the additional data and the shorter horizon of $h = 2$ months significantly improved forecasts based on quantitative models. Forecasts based on the SARIMA, GPR, ETS, and SES models consistently demonstrated superior accuracy. As a consequence, it is recommended that practitioners complement expert forecasts with data-driven methods to enhance the forecasts reliability.

General wisdom holds that experts excel in situations with limited historical data (Hyndman & Athanasopoulos, 2018). Consequently, the third research hypothesis (H3), which postulates that industry insiders outperform in short time series due to restricted quantitative data available for model training, was examined in Section 5.3. The analysis revealed that data-driven forecasts exhibited superiority across all examined time series lengths. Nonetheless, different models showcased higher accuracy under varying circumstances. Specifically, exponential smoothing models attained the highest accuracy for short time series, whereas SARIMA dominated in the medium-length scenario. Conversely, GPR outperformed for longer time series. This implies that several diverse models should be evaluated and the one that aligns most effectively with the given circumstances should be selected.

6.2. Implications, limitations, and future directions

This paper has presented the first comprehensive comparative analysis of expert semiconductor market forecasts. Long production lead times of semiconductors means that fulfilled orders have to be placed and the production has to be planned months in advance. As a result, industry experts must be very accurate when it comes to short-term sales forecasts. WSTS consolidates such industry forecasts to derive market forecast. Hence, we hope that the strong performance of the data driven methods motivates analysts and industry practitioners to employ data-driven methods to enhance existing forecasts – even when time series are short and the models are simple.

Additionally, this study shows that comprehensive multi-granularity modeling of the semiconductor market is feasible. Therefore, our hope is that this paper presents a first step in the direction of reconciling the fields of semiconductor cycle prediction, which assumes a higher-level view, and semiconductor sales and demand forecasting, which is much more granular: (1) The semiconductor market consists of diverse products from processors and memory chips to switches and sensors. Therefore, a multi-granularity study of the semiconductor market could yield insights into possible sub-cycles. (2) Conversely, demand and sales forecasts could benefit from granular and reliable market forecasts as an indicator for future sales. The same holds for semiconductor cycle indicators. Both approaches remain open for future study.

Nevertheless, this study has several limitations. The first relating to the **short time series lengths**, which had several consequences.

1. *Model simplicity.* We observed that simpler models such as SARIMA, ETS, and GPR performed stronger than more complex ML models such as RF. Furthermore, we abstained from the use of even more complex models such as Long Short Term Memory (LSTM) and boosting algorithms such as XGBoost.
2. *Purely autoregressive modeling.* We did not incorporate explanatory variables and relied solely on autoregressive modeling to keep the number of parameters at a minimum.

Table A.10
Mean ranks of the forecasts with horizon h = 2 months across the different time series lengths. Each row refers to a different error measure, sorted by WSTS' forecast type: algorithmic update, meeting (expert forecast), and overall. Lower values are preferable. The best value per row is bold and italic.

			WSTS	SARIMA	ETS	ET	GPR	KNN	RF	SES	SVM	Ensemble
Short	Alg. Update	MSE	8.1	3.4	3.3	5.9	***2.8***	8.9	5.9	4.0	8.9	3.9
		MAPE	8.2	3.6	3.5	6.1	***2.8***	8.8	5.8	4.0	8.5	3.7
		MAE	7.9	3.3	3.4	6.1	***2.9***	8.9	5.9	4.1	8.7	3.9
	Meeting	MSE	7.2	3.8	2.8	6.6	4.0	9.1	5.9	***2.7***	8.6	4.2
		MAPE	7.2	3.3	***2.8***	6.8	3.8	8.9	6.3	3.0	8.7	4.3
		MAE	7.0	3.3	3.0	6.7	4.1	8.9	6.1	***2.9***	8.8	4.2
	Overall	MSE	8.2	3.4	***3.0***	6.2	3.3	9.1	5.9	3.3	8.8	3.8
		MAPE	8.2	3.2	***2.8***	6.4	3.4	9.1	5.8	3.2	9.0	3.9
		MAE	7.9	3.2	***3.0***	6.4	3.4	9.2	5.8	3.2	8.9	4.0
Medium	Alg. Update	MSE	8.1	***2.8***	3.5	6.5	2.9	8.4	6.5	3.7	8.5	4.1
		MAPE	8.4	***2.8***	3.4	6.2	3.1	8.6	6.2	3.7	8.6	4.0
		MAE	8.1	***2.8***	3.5	6.4	3.0	8.6	6.3	3.9	8.5	3.9
	Meeting	MSE	7.1	***2.9***	3.6	6.6	3.8	8.7	6.0	4.1	8.2	4.0
		MAPE	7.8	***2.2***	3.7	6.6	3.7	8.7	5.9	4.0	8.2	4.1
		MAE	7.5	***2.4***	3.5	6.8	3.7	8.7	6.2	3.9	8.2	4.1
	Overall	MSE	8.1	***2.3***	3.4	6.7	3.1	8.8	6.4	3.7	8.5	4.1
		MAPE	8.4	***2.1***	3.5	6.4	3.1	9.1	6.2	3.9	8.4	3.9
		MAE	8.2	***2.0***	3.4	6.5	3.1	9.0	6.4	4.0	8.3	4.0
Long	Alg. Update	MSE	8.0	4.3	4.2	5.3	3.9	8.2	5.1	4.1	8.6	***3.3***
		MAPE	8.1	3.7	4.1	5.4	4.3	8.5	5.1	4.3	8.3	***3.1***
		MAE	7.9	3.8	4.2	5.5	4.2	8.4	5.1	4.3	8.4	***3.2***
	Meeting	MSE	6.7	3.7	3.5	6.4	***3.4***	8.9	5.9	3.7	8.6	4.2
		MAPE	6.8	***3.1***	3.5	6.7	***3.1***	8.8	6.2	3.9	8.7	4.1
		MAE	6.7	***3.1***	3.7	6.8	***3.1***	8.7	6.1	4.0	8.6	4.1
	Overall	MSE	7.4	3.9	3.7	6.2	***3.5***	8.7	5.6	3.7	8.7	3.6
		MAPE	7.7	***3.3***	***3.3***	6.4	***3.3***	8.9	5.8	4.0	8.8	3.5
		MAE	7.5	***3.3***	3.4	6.3	3.6	8.8	5.9	3.9	8.7	3.6

3. *Short forecast horizon.* The focus on quarterly forecasts comparisons limited the forecast horizon under study.

The use of simple models and purely autoregressive modeling could be viewed as strengthening our arguments in the context of the comparative analysis: they provide a natural benchmark, and they can be easily implemented by practitioners. The use of more complex models and the incorporation of indicators are possible extensions of this work. Future contributions could also emphasize longer forecast horizons.

The **hierarchical nature of product categorizations** was not considered in this study. This means that forecasts may not be coherent, i.e., forecasts of sub-groups may not sum up to forecasts of higher-level groups. Forecast coherence is often desired because it enables decisions which are based on these forecasts to be consistent across hierarchical levels. Furthermore, studies have shown that forecast reconciliation can further enhance forecast accuracy (Wickramasuriya et al., 2019).

Finally, our study focuses on **point forecasts** to evaluate forecast accuracy. Probabilistic forecasts (e.g., see Gneiting and Katzfuss (2014)) or uncertainty quantification in the form of prediction intervals may provide a better gauge of risks and opportunities to practitioners. This perspective was not the primary goal of this study, but prediction intervals can easily be obtained from many of the models that were employed here. A study of their accuracy may be of interest to analysts and industry practitioners.

CRediT authorship contribution statement

Louis Steinmeister: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. **Markus Pauly:** Conceptualization, Funding acquisition, Methodology, Supervision, Writing – review & editing.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used the ChatGPT 3.5 model from OpenAI for minor language edits, aiming to enhance readability. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Louis Steinmeister reports financial support was provided by Graduate School of Logistics at TU Dortmund University. This paper was supported through a research collaboration between Infineon Technologies AG and TU Dortmund University through the Graduate School of Logistics. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This paper was supported through a research collaboration between Infineon Technologies AG and TU Dortmund University through the Graduate School of Logistics.

Appendix A. Mean ranks by time series lengths

See Table A.10.

Appendix B. Error measures by product category

Table B.11
Root Mean Squared error (RMSE), the square root of the Mean Squared Error (MSE), for the 110 examined WSTS product categories, which are organized by time series length and presented for the 3-month forecasts (first) and 2-month forecast (second). Averages are contained in the long, medium, short, 3-months, and 2-months rows respectively. The best result in each row, i.e. for each time series, is printed bold and italic. The RMSE was chosen instead of the MSE for readability purposes in this table.

ID	WSTS	SARIMA	ETS	ET	GPR	KNN	RF	SES	SVM	Ensemble
3 months	4.14	3.87	4.07	5.51	***3.78***	8.93	5.36	4.50	8.61	4.27
Long	6.87	6.24	6.71	9.39	***6.06***	15.55	9.11	7.59	14.76	7.08
A2	0.17	0.19	***0.15***	0.25	0.17	0.38	0.23	0.16	0.51	0.19
A5	***0.11***	0.18	0.18	0.13	0.15	0.22	0.13	0.15	0.20	0.15
A99	***0.48***	0.56	0.53	0.80	***0.48***	1.29	0.82	0.53	1.16	0.59
B2	***0.18***	0.20	0.21	0.28	0.24	0.45	0.27	0.20	0.53	0.19
B4	***0.16***	0.24	0.22	0.21	0.21	0.24	0.21	0.19	0.20	0.19
B99	***0.33***	0.48	0.45	0.37	0.42	0.43	0.38	0.37	0.40	0.38
C7	***1.18***	1.58	1.27	2.03	1.30	3.75	1.81	1.34	4.04	1.70
C99	2.11	2.24	2.04	3.57	***1.95***	6.62	3.15	2.22	7.30	2.78
CA	0.33	0.32	0.31	0.33	***0.30***	0.44	0.34	0.35	0.43	0.33
CC	***1.29***	1.69	1.37	2.14	1.41	4.16	1.86	1.46	4.35	1.79
D1	***0.22***	0.23	0.23	0.28	0.23	0.35	0.27	0.24	0.33	0.23
D2	***0.18***	0.23	0.21	0.29	0.23	0.41	0.30	0.19	0.48	0.24
D3	0.16	***0.15***	0.17	0.23	0.17	0.38	0.22	***0.15***	0.50	0.20
D99	***0.42***	0.59	0.48	0.70	0.58	1.08	0.65	0.46	1.29	0.55
E1	***0.10***	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.14	0.12
E2	***0.11***	***0.11***	0.12	0.13	0.13	0.17	0.13	0.12	0.17	0.12
E99	***0.15***	***0.15***	0.16	***0.15***	***0.15***	0.18	***0.15***	0.17	0.18	0.16
F1	0.11	0.10	0.09	0.11	***0.08***	0.13	0.11	0.09	0.13	0.09
F2	1.59	1.53	1.52	1.50	1.68	1.93	1.50	***1.36***	2.10	1.43
F3	0.32	0.39	***0.30***	0.44	0.35	0.68	0.42	0.31	0.79	0.33
F5	5.80	5.94	5.69	6.18	5.85	6.68	6.23	7.84	7.36	***5.37***
F6	0.99	0.86	***0.84***	0.87	0.90	0.99	0.87	0.88	0.86	***0.84***
F8	0.21	0.15	0.17	0.18	0.16	0.28	0.18	***0.14***	0.19	0.16
F99	6.65	6.97	8.32	8.14	7.94	9.08	8.17	9.36	11.53	***6.53***
G99	0.11	0.11	0.11	0.11	0.11	***0.10***	0.11	0.11	0.13	0.11
J0	***3.75***	3.82	4.74	6.72	4.08	10.73	6.79	4.26	10.54	5.13
J4	***0.48***	0.50	0.55	0.71	0.49	1.15	0.63	0.50	1.06	0.57
J6	***2.73***	3.22	3.52	4.85	2.84	7.93	4.74	3.21	7.38	3.69
J7	0.72	0.95	0.74	0.93	0.73	1.32	0.88	***0.68***	1.10	0.80
J9	***0.49***	0.60	0.62	0.84	0.61	1.39	0.78	0.53	1.27	0.64
J99	9.10	***8.75***	10.23	15.80	9.52	27.52	16.16	10.53	29.76	12.28
JA	5.98	***5.41***	6.37	9.89	6.37	17.01	9.46	7.04	19.10	7.71
L1	0.03	0.03	0.03	0.03	***0.02***	***0.02***	***0.02***	***0.02***	0.03	***0.02***
L2a	0.35	0.40	0.41	0.34	***0.32***	0.39	0.35	0.35	0.36	0.33
L2b	0.04	0.04	0.04	0.04	0.04	0.04	0.04	***0.03***	0.05	***0.03***
L2c	3.20	3.20	***3.05***	5.47	3.71	10.27	4.70	3.83	10.54	4.24
L99	21.83	***15.97***	20.19	27.02	18.43	54.18	25.59	22.41	53.28	21.82
LA	18.64	***13.17***	17.50	21.95	16.86	44.46	21.00	18.75	40.56	17.96
LS	***4.37***	5.93	5.51	7.91	5.90	16.21	7.18	4.96	17.44	6.40
M1	28.93	35.32	27.90	33.67	27.67	53.12	34.56	30.86	52.37	***26.74***
M2	0.13	0.13	0.13	0.15	0.13	0.16	0.14	***0.11***	0.15	***0.11***
M8	0.43	0.44	0.50	***0.40***	0.49	0.41	0.49	0.66	0.44	0.46
M99	46.77	49.26	40.48	49.73	***37.72***	75.78	53.49	45.91	75.55	39.99
P1	***10.39***	10.71	12.80	13.02	10.59	14.24	12.71	13.42	14.83	11.98
P2	***2.83***	3.55	3.63	4.78	3.54	8.08	4.52	3.03	9.06	3.97
P5	***2.55***	3.17	3.24	4.39	3.46	7.82	4.26	2.77	9.48	3.72
P99	10.93	11.10	12.96	13.92	***10.73***	19.96	12.74	13.19	16.38	12.28
S2	70.79	***49.17***	65.42	99.61	53.28	166.49	94.91	77.94	146.56	68.48
S3	***2.77***	3.83	3.49	4.86	3.12	9.10	3.92	2.97	10.56	3.65
T99	71.85	57.77	66.25	113.07	***56.83***	189.21	106.86	83.20	164.85	76.28
Medium	***1.77***	1.78	1.81	2.27	1.81	3.51	2.22	1.86	3.60	1.92
A3	0.26	0.23	0.29	0.41	***0.22***	0.60	0.36	0.29	0.51	0.27
A4	0.09	***0.08***	***0.08***	***0.08***	***0.08***	0.09	***0.08***	***0.08***	***0.08***	***0.08***
B3	***0.08***	0.10	***0.08***	0.10	0.09	0.15	0.09	0.09	0.17	0.09
C10	0.84	0.98	***0.72***	1.15	0.76	1.97	1.08	0.87	2.03	0.94
C8	0.22	***0.20***	0.21	0.31	***0.20***	0.50	0.28	0.22	0.52	0.25
C9	0.37	0.35	0.33	0.57	***0.31***	0.90	0.53	0.33	1.11	0.41
CB	1.11	1.22	***0.91***	1.61	***0.91***	2.90	1.42	1.04	3.05	1.24

(*continued on next page*)

Table B.11 (*continued*).

F7	0.23	0.21	***0.18***	0.27	0.19	0.32	0.29	0.22	0.32	0.21
H99	***2.11***	2.41	2.49	3.28	2.41	6.23	3.21	2.42	6.95	2.66
JB	0.82	0.94	0.86	0.83	0.82	***0.77***	0.83	0.97	0.89	0.79
JBa	0.56	0.56	***0.52***	0.57	0.55	0.59	0.58	0.59	0.60	0.55
JBb	***0.36***	0.50	0.49	0.60	0.46	0.83	0.65	0.50	0.74	0.46
JC	0.58	0.76	0.72	0.57	0.71	0.70	***0.54***	0.73	0.58	0.58
JCc	***0.15***	0.20	0.22	0.25	0.23	0.30	0.24	0.17	0.30	0.21
JCd	0.49	0.66	0.59	0.48	0.62	0.50	0.47	0.58	***0.45***	0.52
JCe	0.12	0.13	0.13	0.13	***0.11***	0.20	0.12	0.13	0.12	0.12
JD	3.90	4.08	3.95	5.57	***3.89***	9.52	5.55	4.36	10.23	4.49
JE	***2.19***	2.42	2.29	3.64	2.51	5.91	3.61	2.44	6.81	3.07
JF	***0.48***	0.66	0.60	0.81	0.58	1.60	0.73	0.59	1.83	0.69
L3	3.28	2.99	2.99	3.72	3.16	6.59	3.92	***2.96***	6.13	3.26
L5	3.13	2.88	3.17	4.03	3.29	7.02	4.45	***2.64***	6.00	3.38
L6	9.68	***8.82***	9.54	10.84	9.19	16.44	10.50	9.64	16.51	9.75
L6g	8.76	***8.17***	9.36	10.05	9.13	13.19	9.67	8.90	12.70	9.23
L6h	1.16	***1.13***	1.25	1.49	1.16	2.81	1.47	1.24	2.72	1.31
L6i	0.79	***0.69***	0.84	0.85	0.77	1.04	0.82	0.74	0.88	0.76
L7	***9.72***	10.91	10.09	13.75	10.12	20.27	13.06	11.63	22.29	10.43
L8	1.06	***1.05***	1.09	1.55	1.14	3.52	1.48	1.17	4.16	1.32
L9	0.58	***0.54***	0.63	1.13	1.07	1.43	1.06	0.57	1.50	0.90
M6	0.23	***0.21***	0.28	0.22	***0.21***	0.28	0.23	0.23	0.26	0.22
M7A	0.98	0.56	0.59	0.83	***0.50***	0.81	0.76	0.75	0.66	0.63
P4	***0.53***	0.66	0.69	0.65	0.66	0.80	0.63	0.61	0.64	0.61
Short	1.98	2.02	***1.92***	2.28	1.97	3.31	2.26	1.97	3.34	1.95
F10	***0.82***	1.27	0.94	1.38	1.86	1.57	1.40	1.15	1.44	1.16
F9	0.41	0.41	0.43	0.52	***0.37***	1.29	0.52	0.46	1.21	0.48
H1	0.85	0.85	***0.80***	0.93	0.82	1.30	0.89	0.85	1.34	0.85
H2	0.62	0.73	0.73	0.67	0.64	0.83	0.70	0.72	0.87	***0.60***
H3	***0.56***	0.63	0.70	0.81	0.62	0.89	0.81	0.65	0.98	0.67
H4	***0.46***	0.58	0.48	0.55	0.50	0.79	0.51	0.47	0.91	0.49
H6	***1.77***	1.84	2.13	2.37	2.04	3.06	2.37	1.99	5.25	2.10
JBc	***0.02***	0.03	0.03	***0.02***	0.03	0.03	***0.02***	0.03	0.03	***0.02***
JCf	***0.06***	0.07	0.07	0.08	0.07	0.11	0.08	0.07	0.09	0.07
JDa	***3.23***	3.27	3.34	4.40	3.44	6.83	4.33	3.58	6.96	3.69
JDb	***0.43***	0.47	0.50	0.60	0.46	1.46	0.57	0.44	1.84	0.51
JDc	***0.58***	0.67	0.76	0.83	0.78	1.17	0.80	0.70	1.09	0.68
JDd	0.63	0.88	***0.61***	0.77	0.77	0.93	0.76	0.68	1.06	0.72
JDe	0.40	0.37	0.38	0.37	0.36	0.70	0.35	0.35	***0.33***	0.34
JEa	***0.27***	0.34	0.41	0.47	0.40	0.72	0.42	0.40	0.94	0.42
JEb	***1.99***	2.13	2.58	3.57	3.48	5.08	3.52	2.49	5.39	3.14
L3a	1.53	1.45	1.42	1.28	1.30	1.61	1.36	1.42	***1.25***	1.31
L3b	***1.94***	2.13	2.09	2.37	2.11	3.07	2.68	2.08	2.86	2.23
L3c	0.55	1.23	0.47	0.62	0.55	1.41	0.62	***0.43***	1.50	0.54
L4	1.02	0.96	1.56	1.25	***0.81***	2.07	1.24	0.82	1.96	0.88
L5a	1.97	1.85	1.94	1.86	1.91	3.14	1.84	***1.79***	2.33	1.80
L5b	1.27	1.34	1.27	1.61	1.94	3.15	1.56	***0.94***	2.60	1.43
L5c	0.23	0.23	***0.20***	0.26	0.22	0.38	0.26	0.22	0.26	0.21
L7a/b/c/d/f	***9.28***	10.65	9.41	12.60	9.58	16.70	11.97	10.93	18.00	9.96
L7e	3.90	3.45	***3.10***	3.92	3.28	7.63	3.84	3.81	9.17	3.80
L8a	0.69	***0.59***	0.86	1.03	0.77	2.46	1.02	0.78	2.54	0.91
L8b	***0.52***	0.58	0.57	0.84	0.65	1.25	0.82	0.56	1.34	0.69
M7B	20.62	18.75	17.11	19.15	16.66	25.32	19.42	17.74	22.21	***16.17***
P6	0.73	0.73	0.68	0.98	0.81	1.04	0.93	***0.64***	1.21	0.81
2 months	4.14	2.33	2.44	3.51	***2.31***	5.70	3.39	3.06	5.60	2.87
Long	6.87	3.87	4.12	6.01	***3.83***	9.87	5.80	5.33	9.66	4.89
A2	0.17	0.13	0.12	0.14	0.12	0.20	0.13	***0.10***	0.31	***0.10***
A5	0.11	0.14	0.13	***0.10***	0.13	0.16	***0.10***	***0.10***	0.12	0.12
A99	0.48	0.47	0.42	0.51	0.44	0.80	0.51	***0.37***	0.67	0.40
B2	***0.18***	0.24	0.21	0.21	0.24	0.29	0.20	0.19	0.36	0.19
B4	0.16	0.13	0.16	0.13	0.16	0.16	0.14	0.13	0.13	***0.12***
B99	0.33	0.42	0.37	0.29	0.39	0.32	0.30	0.29	***0.28***	0.29
C7	***1.18***	1.68	1.39	1.57	1.32	2.54	1.45	1.48	2.71	1.51
C99	2.11	2.28	2.06	2.64	***1.86***	4.50	2.42	2.47	4.77	2.48
CA	0.33	0.25	0.23	0.23	***0.21***	0.28	0.22	0.24	0.28	0.22
CC	***1.29***	1.85	1.57	1.72	1.42	2.82	1.58	1.66	2.94	1.67
D1	0.22	0.17	0.16	0.18	0.17	0.20	0.18	***0.15***	0.24	0.16
D2	0.18	0.21	0.18	0.20	0.19	0.30	0.19	***0.17***	0.32	0.18
D3	0.16	***0.12***	***0.12***	0.17	***0.12***	0.26	0.15	0.14	0.32	0.14
D99	0.42	0.44	0.40	0.47	0.43	0.75	0.46	***0.39***	0.87	0.42

(*continued on next page*)

Table B.11 (*continued*).

E1	0.10	0.07	0.08	0.08	0.07	0.09	0.07	***0.06***	0.09	0.07
E2	0.11	***0.07***	0.08	0.08	0.08	0.09	0.09	***0.07***	0.12	0.08
E99	0.15	***0.09***	0.10	0.10	***0.09***	***0.09***	***0.09***	0.10	0.12	***0.09***
F1	0.11	0.06	***0.05***	0.06	***0.05***	0.08	0.06	***0.05***	0.06	***0.05***
F2	1.59	0.68	0.74	0.85	0.85	1.12	0.85	***0.65***	1.35	0.74
F3	0.32	0.26	0.27	0.31	0.26	0.50	0.29	***0.23***	0.50	0.26
F5	5.80	***3.52***	4.01	4.63	4.04	4.76	4.64	4.25	5.33	3.77
F6	0.99	0.55	0.64	0.51	0.64	0.57	0.55	0.52	***0.49***	0.50
F8	0.21	***0.06***	0.08	0.10	0.08	0.18	0.11	0.08	0.10	0.08
F99	6.65	4.47	4.88	5.16	4.68	6.25	5.09	5.24	7.26	***4.43***
G99	0.11	***0.06***	***0.06***	0.07	***0.06***	0.09	0.07	***0.06***	0.08	0.07
J0	3.75	3.02	2.42	3.56	2.46	6.68	3.55	***2.32***	6.51	2.46
J4	0.48	0.24	0.25	0.36	0.26	0.59	0.35	***0.23***	0.54	0.29
J6	2.73	1.93	1.94	2.72	1.80	4.60	2.49	***1.71***	4.76	1.90
J7	0.72	0.47	0.41	0.48	0.43	0.82	0.45	0.47	0.63	***0.37***
J9	0.49	0.45	0.38	0.47	0.41	0.86	0.48	0.34	0.76	***0.32***
J99	9.10	5.07	5.08	9.07	***4.79***	18.55	9.15	5.31	18.52	6.11
JA	5.98	***2.69***	2.99	5.80	2.97	11.38	5.64	3.89	12.17	4.22
L1	0.03	***0.02***	0.03	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***	0.03	***0.02***
L2a	0.35	0.25	0.24	0.22	0.26	0.24	0.23	***0.18***	0.23	0.20
L2b	0.04	***0.02***	***0.02***	0.03	***0.02***	0.03	***0.02***	***0.02***	0.04	***0.02***
L2c	3.20	1.45	***1.41***	3.19	1.61	6.72	2.99	2.23	6.41	2.46
L99	21.83	10.68	12.23	17.65	***10.63***	37.43	16.77	14.80	33.61	14.83
LA	18.64	***8.92***	11.23	14.76	10.11	30.80	14.38	12.43	26.03	12.73
LS	4.37	2.90	***2.76***	4.68	2.80	10.37	4.71	3.07	10.53	3.57
M1	28.93	19.44	***18.65***	20.98	19.17	32.64	22.87	25.83	35.41	19.31
M2	0.13	0.10	0.10	0.11	0.10	0.14	0.11	***0.09***	0.13	***0.09***
M8	0.43	***0.29***	0.30	0.30	0.34	0.32	0.32	0.50	0.31	0.30
M99	46.77	28.79	28.31	33.60	***26.37***	48.21	32.72	36.18	51.81	28.73
P1	10.39	***6.56***	7.12	8.19	6.64	10.02	7.64	8.46	10.63	7.66
P2	2.83	***1.88***	1.97	3.14	1.99	5.39	3.02	2.13	5.65	2.45
P5	2.55	1.70	***1.66***	2.85	1.87	5.30	2.77	2.04	6.14	2.25
P99	10.93	***5.97***	6.47	8.48	6.42	13.13	8.14	8.51	12.01	8.01
S2	70.79	***32.89***	37.13	64.12	33.11	101.47	59.85	54.16	94.54	49.90
S3	***2.77***	3.53	3.15	3.36	3.12	5.99	3.03	2.94	6.80	3.15
T99	71.85	***35.79***	41.38	72.01	35.88	114.29	68.42	59.43	108.95	55.01
Medium	1.77	***0.96***	1.01	1.43	1.01	2.32	1.37	1.14	2.28	1.19
A3	0.26	0.22	0.24	0.27	***0.21***	0.44	0.27	0.22	0.33	0.23
A4	0.09	***0.02***	0.03	0.04	0.03	0.05	0.05	***0.02***	0.04	0.03
B3	0.08	***0.05***	***0.05***	0.07	***0.05***	0.11	0.06	***0.05***	0.11	0.06
C10	0.84	***0.46***	0.50	0.84	0.47	1.26	0.80	0.69	1.31	0.74
C8	0.22	0.19	0.19	0.26	***0.18***	0.37	0.25	0.23	0.38	0.23
C9	0.37	0.32	0.29	0.41	***0.26***	0.62	0.40	0.30	0.75	0.35
CB	1.11	0.65	0.65	1.13	***0.60***	1.79	1.01	0.91	2.01	0.98
F7	0.23	0.13	***0.11***	0.16	***0.11***	0.22	0.16	***0.11***	0.19	0.12
H99	2.11	***1.01***	1.18	1.92	1.17	4.28	1.75	1.49	4.44	1.48
JB	0.82	0.28	***0.26***	0.44	0.30	0.40	0.44	0.37	0.43	0.31
JBa	0.56	***0.17***	***0.17***	0.29	0.18	0.28	0.29	0.20	0.30	0.18
JBb	0.36	0.25	***0.24***	0.34	***0.24***	0.49	0.34	0.26	0.48	0.28
JC	0.58	0.37	0.36	0.37	0.37	0.50	0.38	0.40	0.39	***0.35***
JCc	0.15	0.14	0.21	0.16	0.16	0.16	0.17	***0.13***	0.19	0.15
JCd	0.49	0.32	0.29	0.30	0.29	0.31	0.32	0.36	***0.28***	0.31
JCe	0.12	***0.08***	0.10	0.09	0.10	0.09	0.09	***0.08***	0.10	***0.08***
JD	3.90	2.37	***2.13***	3.46	2.20	6.78	3.48	2.51	6.28	2.72
JE	2.19	***0.78***	0.98	2.27	1.05	3.96	2.21	1.48	4.03	1.68
JF	0.48	0.40	***0.37***	0.48	0.41	0.90	0.45	0.41	1.10	0.40
L3	3.28	1.60	1.71	2.20	1.99	3.74	2.10	***1.59***	3.94	1.83
L5	3.13	1.22	1.49	1.77	1.49	3.49	1.89	***0.97***	3.00	1.36
L6	9.68	***5.61***	6.09	7.13	5.75	11.39	6.85	6.68	10.71	6.51
L6g	8.76	***5.29***	5.78	6.61	5.68	9.12	6.04	6.13	8.36	6.10
L6h	1.16	0.66	0.69	0.92	***0.65***	1.88	0.90	0.72	1.84	0.77
L6i	0.79	***0.40***	0.45	0.56	0.43	0.81	0.56	0.41	0.51	0.43
L7	9.72	5.00	***4.82***	9.04	5.23	13.81	8.62	6.53	14.40	6.92
L8	1.06	0.57	***0.46***	0.94	0.50	2.40	0.86	0.71	2.75	0.77
L9	0.58	0.26	0.45	0.61	0.31	0.80	0.62	***0.25***	0.77	0.42
M6	0.23	0.15	0.19	0.15	0.19	0.19	0.15	0.16	0.16	***0.14***
M7A	0.98	***0.34***	0.37	0.52	0.36	0.62	0.50	0.46	0.43	0.37
P4	0.53	0.45	0.48	0.48	0.47	0.64	0.48	***0.43***	0.58	0.45

(*continued on next page*)

Table B.11 (*continued*).

Short	1.98	1.14	***1.06***	1.43	1.07	2.12	1.39	1.22	2.17	1.20
F10	0.82	1.09	0.84	0.82	1.31	0.95	0.82	0.80	0.97	***0.73***
F9	0.41	0.30	0.33	0.41	***0.27***	0.88	0.41	0.36	0.82	0.38
H1	0.85	0.31	***0.28***	0.47	0.35	0.82	0.46	0.29	0.83	0.35
H2	0.62	0.31	***0.27***	0.43	***0.27***	0.48	0.43	0.33	0.55	0.33
H3	0.56	***0.28***	0.31	0.50	***0.28***	0.52	0.49	0.40	0.54	0.39
H4	0.46	0.33	0.31	0.37	0.33	0.56	0.36	***0.30***	0.60	0.33
H6	1.77	***0.81***	0.92	1.31	0.86	2.08	1.29	1.02	3.45	1.13
JBc	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***
JCf	0.06	***0.04***	0.05	0.05	0.06	0.07	0.05	***0.04***	0.06	0.05
JDa	3.23	1.73	***1.48***	2.41	1.70	3.95	2.43	1.77	4.57	2.01
JDb	0.43	0.25	0.29	0.37	***0.22***	0.99	0.36	0.26	1.21	0.31
JDc	0.58	0.59	0.59	0.55	0.54	0.82	0.55	0.70	0.74	***0.53***
JDd	0.63	0.42	***0.39***	0.43	***0.39***	0.59	0.47	0.45	0.55	0.41
JDe	0.40	0.32	0.35	***0.28***	0.30	0.45	***0.28***	0.33	***0.28***	***0.28***
JEa	0.27	***0.21***	0.22	0.29	0.22	0.42	0.28	0.26	0.60	0.27
JEb	1.99	1.47	1.48	2.14	***1.14***	3.38	2.25	1.56	3.41	1.79
L3a	1.53	0.84	0.74	0.83	0.84	1.03	0.78	***0.73***	1.01	0.79
L3b	1.94	1.35	1.34	1.55	1.52	2.10	1.58	***1.29***	2.00	1.44
L3c	0.55	0.66	***0.20***	0.33	0.24	0.64	0.34	0.22	0.92	0.25
L4	1.02	0.55	0.55	0.65	0.47	1.41	0.66	***0.45***	1.27	0.49
L5a	1.97	0.72	0.82	0.86	0.78	1.65	0.80	***0.65***	1.05	0.70
L5b	1.27	0.46	0.65	0.86	0.98	1.72	0.87	***0.38***	1.37	0.64
L5c	0.23	0.13	0.14	0.16	0.14	0.22	0.16	***0.11***	0.18	0.13
L7a/b/c/d/f	9.28	4.29	***3.98***	7.51	4.31	10.11	7.23	4.84	11.17	5.43
L7e	3.90	2.34	***2.09***	2.89	2.23	5.30	2.96	2.71	6.12	2.77
L8a	0.69	***0.35***	0.37	0.53	0.37	1.67	0.56	0.39	1.61	0.46
L8b	0.52	0.42	***0.36***	0.59	0.37	0.89	0.57	0.44	0.94	0.51
M7B	20.62	12.21	10.98	13.22	***10.02***	17.07	12.23	13.84	15.39	11.33
P6	0.73	0.38	***0.35***	0.58	0.39	0.61	0.55	0.40	0.68	0.46

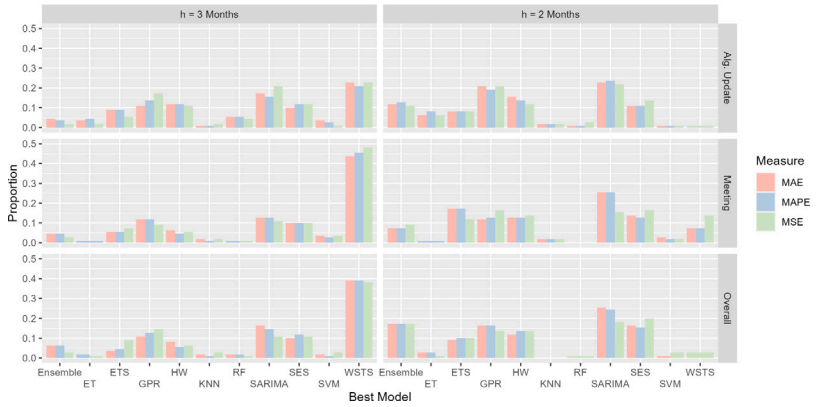


Fig. C.4. Bar chart illustrating the frequency of the best-performing forecasts based on Mean Squared Error (MSE) in red, Mean Absolute Percentage Error (MAPE) in blue, and Mean Absolute Error (MAE) in green, for the 110 WSTS product categories. The left side of the chart represents the outcomes of forecasts with a horizon of $h = 3$ months, while the right side represents those with a forecast horizon of $h = 2$ months. The rows are organized by algorithmic update, expert forecasts ("Meeting"), and overall performance, mirroring the structure of Tables 4 and 6.

Table B.12
Mean absolute percentage errors (MAPE) for the 110 examined WSTS product categories, which are organized by time series length and presented for the 3-month forecasts (first) and 2-month forecast (second). Averages are contained in the long, medium, short, 3-months, and 2-months rows respectively. The best result in each row, i.e. for each time series, is printed bold and italic.

ID	WSTS	SARIMA	ETS	ET	GPR	KNN	RF	SES	SVM	Ensemble
3 months	***8.4%***	9.0%	8.8%	10.6%	9.0%	14.5%	10.3%	9.0%	14.7%	9.1%
Long	***7.7%***	8.1%	8.1%	9.6%	8.0%	12.7%	9.5%	8.5%	13.3%	8.3%
A2	***5.5%***	6.6%	5.9%	8.3%	7.3%	12.7%	7.7%	6.1%	15.5%	6.8%
A5	***4.6%***	7.1%	6.5%	5.4%	6.3%	9.1%	5.7%	6.5%	8.3%	5.9%
A99	***4.2%***	5.2%	5.5%	7.6%	5.0%	12.3%	7.5%	5.4%	10.3%	5.8%
B2	5.2%	***4.9%***	5.2%	9.1%	6.1%	13.4%	8.8%	5.7%	16.3%	6.4%
B4	***6.0%***	8.5%	7.4%	7.6%	7.5%	9.7%	7.8%	6.8%	7.8%	7.3%
B99	***4.6%***	6.5%	5.8%	5.2%	5.6%	5.7%	5.5%	5.8%	5.3%	5.3%
C7	***4.0%***	4.8%	4.3%	7.1%	4.5%	13.0%	6.1%	4.8%	14.1%	6.0%
C99	***3.4%***	4.0%	3.6%	6.9%	3.7%	12.0%	6.0%	4.0%	13.4%	5.4%
CA	5.3%	5.2%	***4.6%***	5.3%	5.1%	7.5%	5.4%	5.7%	6.9%	5.4%
CC	***4.1%***	4.6%	4.2%	7.0%	4.6%	13.1%	6.2%	5.1%	13.5%	5.9%
D1	***4.1%***	5.1%	4.8%	5.2%	5.3%	6.8%	5.0%	4.9%	6.2%	5.0%
D2	***3.7%***	4.8%	4.8%	6.3%	5.0%	8.6%	6.5%	4.1%	9.5%	5.1%
D3	5.8%	5.5%	6.4%	8.8%	6.5%	14.6%	8.2%	***5.4%***	20.2%	7.5%
D99	***3.1%***	4.8%	4.0%	5.8%	4.7%	8.1%	5.3%	3.8%	10.3%	4.5%
E1	***8.2%***	9.2%	10.1%	9.8%	8.4%	9.4%	9.9%	9.3%	11.5%	8.6%
E2	***9.9%***	10.1%	10.3%	12.5%	11.4%	14.3%	12.7%	10.7%	14.8%	11.2%
E99	***6.6%***	6.9%	7.0%	6.9%	6.7%	8.2%	6.7%	7.9%	8.0%	7.2%
F1	8.3%	6.9%	6.2%	8.3%	***6.1%***	9.9%	8.5%	6.3%	11.0%	6.4%
F2	4.5%	4.0%	3.9%	4.0%	4.6%	5.2%	4.1%	***3.6%***	5.6%	3.9%
F3	5.5%	5.9%	5.5%	7.7%	5.5%	10.4%	7.3%	***5.3%***	12.3%	5.8%
F5	11.1%	11.7%	11.6%	10.1%	11.6%	11.3%	10.6%	15.7%	12.7%	***9.7%***
F6	11.8%	10.0%	10.7%	9.9%	11.2%	11.8%	10.4%	10.2%	9.9%	***9.7%***
F8	9.3%	7.3%	7.8%	9.4%	7.9%	14.5%	8.7%	***7.1%***	8.7%	7.8%
F99	***5.2%***	5.6%	7.2%	6.5%	6.8%	7.6%	6.6%	8.3%	9.8%	5.6%
G99	54.8%	53.4%	55.2%	56.6%	54.4%	***52.4%***	58.4%	55.9%	66.9%	55.3%
J0	4.8%	***4.7%***	5.7%	7.9%	5.0%	11.5%	7.7%	5.5%	11.1%	6.0%
J4	6.3%	6.3%	6.7%	7.8%	***6.1%***	11.5%	7.1%	6.9%	10.0%	6.8%
J6	***5.5%***	6.1%	6.9%	9.3%	5.7%	13.8%	9.1%	6.6%	13.4%	7.2%
J7	5.7%	7.5%	***5.4%***	7.6%	6.0%	9.5%	7.4%	6.0%	8.3%	6.6%
J9	***4.2%***	5.2%	5.0%	7.0%	4.9%	10.1%	6.4%	4.3%	10.3%	4.9%
J99	4.5%	***4.1%***	4.9%	7.8%	4.6%	12.5%	8.0%	5.5%	13.0%	6.1%
JA	4.9%	***4.4%***	4.9%	8.0%	4.9%	13.5%	7.8%	6.0%	14.0%	6.2%
L1	11.7%	13.5%	12.8%	11.6%	11.6%	11.7%	***10.5%***	12.6%	14.3%	11.0%
L2a	8.2%	8.4%	9.2%	8.0%	***6.9%***	9.7%	7.9%	7.8%	8.3%	7.5%
L2b	16.2%	18.5%	18.5%	18.7%	17.6%	23.5%	18.7%	***15.7%***	27.4%	16.8%
L2c	5.8%	***5.5%***	5.6%	9.5%	6.5%	14.1%	8.4%	6.4%	15.4%	7.3%
L99	4.8%	***3.5%***	4.2%	6.4%	4.2%	11.3%	6.0%	5.4%	11.4%	5.1%
LA	5.3%	***3.7%***	4.6%	6.5%	4.7%	12.3%	5.8%	5.6%	11.3%	5.0%
LS	***4.7%***	6.7%	6.0%	8.1%	6.5%	13.5%	7.5%	5.8%	15.8%	6.8%
M1	12.0%	14.1%	12.1%	16.1%	***11.0%***	22.9%	16.4%	14.0%	17.6%	12.6%
M2	11.1%	11.5%	10.1%	13.7%	10.5%	14.8%	13.1%	***9.2%***	15.0%	10.6%
M8	26.1%	24.3%	26.3%	***21.6%***	23.1%	24.1%	25.1%	32.8%	24.2%	23.1%
M99	11.4%	11.9%	9.6%	14.0%	***9.2%***	20.8%	14.4%	12.1%	17.3%	11.1%
P1	***6.7%***	7.5%	8.8%	8.4%	7.8%	9.2%	8.1%	9.1%	10.4%	7.9%
P2	***4.9%***	6.0%	6.0%	8.2%	5.9%	11.3%	7.8%	5.7%	13.6%	6.8%
P5	***4.8%***	6.0%	6.1%	8.8%	6.8%	12.3%	8.5%	5.5%	16.3%	7.2%
P99	***4.8%***	5.1%	6.1%	6.4%	5.0%	8.4%	5.9%	6.3%	6.9%	5.8%
S2	5.2%	***4.1%***	4.9%	7.9%	4.2%	11.4%	7.6%	6.4%	9.9%	5.5%
S3	***3.0%***	4.3%	4.1%	5.4%	3.7%	10.4%	4.5%	3.7%	11.9%	4.4%
T99	4.4%	***3.7%***	4.1%	7.5%	3.8%	11.0%	7.2%	5.8%	9.1%	5.1%
Medium	***8.0%***	8.2%	8.2%	10.1%	8.3%	14.1%	9.7%	8.3%	14.0%	8.6%
A3	5.4%	4.9%	6.5%	8.9%	***4.5%***	12.5%	7.8%	6.3%	9.9%	6.1%
A4	6.4%	6.7%	6.1%	7.1%	6.8%	9.4%	6.8%	***5.7%***	7.4%	6.2%
B3	10.0%	12.6%	***9.7%***	13.9%	11.3%	22.9%	13.1%	10.5%	26.1%	12.0%
C10	***5.1%***	6.8%	5.4%	7.8%	5.5%	13.5%	7.5%	5.8%	15.1%	5.8%
C8	8.8%	9.0%	***8.5%***	11.8%	9.0%	15.3%	10.7%	8.7%	16.1%	9.5%
C9	6.4%	6.0%	5.6%	9.4%	***5.5%***	16.8%	8.6%	5.8%	20.3%	6.8%
CB	4.6%	5.6%	***4.5%***	7.6%	***4.5%***	14.5%	6.6%	4.7%	16.9%	5.5%

(*continued on next page*)

Table B.12 (*continued*).

F7	6.6%	6.3%	6.0%	8.2%	***5.8%***	10.2%	8.4%	6.9%	9.9%	6.4%
H99	***3.9%***	4.1%	4.6%	5.5%	4.2%	10.1%	5.3%	4.3%	12.4%	4.6%
JB	10.2%	12.1%	10.9%	11.4%	***9.9%***	10.4%	11.0%	12.3%	11.4%	10.2%
JBa	12.5%	12.9%	***11.5%***	13.3%	11.8%	13.5%	12.9%	13.4%	13.5%	12.2%
JBb	***11.5%***	13.6%	13.0%	15.8%	13.0%	20.1%	15.9%	14.4%	17.2%	13.3%
JC	6.8%	7.2%	8.2%	7.1%	7.8%	8.9%	***6.5%***	8.0%	7.2%	6.9%
JCc	10.5%	11.2%	13.4%	14.8%	14.6%	14.5%	13.7%	***10.4%***	17.5%	12.5%
JCd	***8.5%***	10.8%	9.3%	10.1%	10.5%	11.9%	9.6%	9.6%	9.5%	9.5%
JCe	***7.1%***	8.3%	8.2%	8.7%	7.7%	13.9%	8.7%	8.7%	7.9%	8.1%
JD	6.3%	6.2%	6.1%	8.7%	***5.6%***	14.0%	9.0%	6.8%	13.7%	6.9%
JE	***6.0%***	6.4%	6.4%	10.5%	6.9%	16.1%	10.3%	6.7%	19.5%	8.9%
JF	***5.0%***	5.8%	5.7%	8.0%	5.1%	14.0%	7.4%	6.3%	14.7%	6.7%
L3	***13.2%***	14.1%	14.3%	15.7%	14.9%	21.8%	16.4%	14.0%	20.1%	14.9%
L5	7.4%	6.9%	8.0%	9.2%	8.3%	13.3%	10.5%	***6.2%***	13.2%	8.2%
L6	***7.5%***	***7.5%***	7.7%	10.8%	8.0%	16.5%	9.9%	8.6%	15.4%	9.0%
L6g	8.6%	***7.8%***	8.8%	12.6%	9.6%	16.4%	11.5%	9.4%	15.6%	10.4%
L6h	6.2%	***5.6%***	6.1%	8.2%	6.0%	13.2%	8.0%	5.9%	12.5%	6.7%
L6i	11.1%	***10.5%***	11.9%	13.6%	11.5%	17.8%	13.3%	***10.5%***	13.0%	11.6%
L7	5.8%	6.7%	5.7%	7.3%	5.7%	9.7%	6.6%	6.8%	10.7%	***5.3%***
L8	***5.6%***	***5.6%***	6.4%	9.5%	6.3%	20.2%	8.7%	6.8%	24.9%	7.8%
L9	13.4%	12.7%	13.3%	15.5%	17.7%	20.1%	14.4%	***12.1%***	17.9%	14.8%
M6	5.7%	5.8%	7.6%	5.9%	5.6%	7.7%	5.9%	***5.4%***	7.2%	5.5%
M7A	14.0%	6.4%	8.1%	9.2%	***6.2%***	9.7%	8.7%	8.6%	8.8%	7.4%
P4	***6.4%***	8.0%	8.2%	7.8%	8.4%	9.4%	7.6%	7.2%	7.2%	7.4%
Short	***9.9%***	11.4%	10.4%	12.7%	11.4%	18.0%	12.5%	10.6%	17.8%	11.0%
F10	***15.8%***	24.8%	19.3%	26.0%	31.8%	26.8%	26.9%	23.6%	26.6%	22.8%
F9	5.6%	5.3%	5.7%	6.6%	***4.7%***	13.6%	6.5%	6.2%	16.0%	5.9%
H1	9.3%	***8.4%***	8.9%	11.5%	9.6%	18.6%	11.3%	8.9%	20.9%	9.7%
H2	7.5%	9.5%	9.0%	7.9%	8.1%	10.1%	7.8%	8.6%	10.8%	***7.1%***
H3	***5.2%***	6.3%	6.6%	8.3%	6.0%	9.3%	8.1%	6.3%	10.6%	6.4%
H4	***5.8%***	8.3%	6.4%	7.1%	6.8%	9.4%	6.6%	6.6%	11.5%	6.3%
H6	***7.1%***	***7.1%***	7.5%	9.1%	7.7%	10.9%	9.4%	8.2%	18.5%	7.8%
JBc	20.6%	***20.4%***	20.6%	25.5%	21.5%	31.1%	24.7%	23.2%	27.8%	21.7%
JCf	20.7%	23.4%	***19.5%***	28.8%	21.2%	39.9%	28.5%	23.4%	34.8%	25.3%
JDa	***8.1%***	8.2%	8.5%	10.8%	8.4%	15.4%	10.5%	9.1%	15.6%	9.3%
JDb	6.7%	6.9%	7.5%	10.0%	***6.2%***	22.6%	10.0%	7.1%	27.6%	8.0%
JDc	***7.1%***	8.4%	8.9%	9.2%	9.5%	13.3%	8.8%	8.4%	12.5%	7.5%
JDd	21.9%	25.2%	***18.2%***	21.6%	22.6%	24.8%	20.8%	20.4%	27.5%	19.8%
JDe	6.7%	5.9%	6.2%	6.7%	***5.4%***	10.9%	6.4%	5.6%	5.7%	5.6%
JEa	***6.7%***	8.3%	10.1%	11.6%	10.7%	15.5%	10.4%	9.2%	18.0%	10.2%
JEb	***6.0%***	6.3%	8.0%	11.5%	10.1%	16.0%	11.3%	7.9%	17.4%	10.3%
L3a	17.5%	14.8%	14.7%	15.7%	14.7%	17.8%	16.3%	14.7%	***14.1%***	15.3%
L3b	***22.1%***	23.6%	24.7%	23.5%	23.6%	33.4%	25.4%	24.3%	28.0%	24.3%
L3c	7.8%	22.2%	7.7%	11.6%	11.0%	19.3%	10.8%	***6.9%***	20.9%	9.5%
L4	7.4%	7.2%	9.6%	9.0%	***5.6%***	13.8%	8.8%	6.3%	12.4%	6.5%
L5a	7.1%	7.7%	7.6%	7.1%	7.6%	11.9%	7.2%	6.9%	9.4%	***6.4%***
L5b	9.7%	10.8%	10.8%	13.6%	16.7%	22.7%	12.1%	***7.9%***	21.6%	11.4%
L5c	8.9%	9.0%	***8.2%***	11.2%	9.1%	16.9%	10.7%	***8.2%***	11.2%	8.6%
L7a/b/c/d/f	6.8%	8.6%	6.8%	7.9%	6.8%	11.1%	7.5%	8.2%	11.0%	***6.5%***
L7e	6.6%	6.7%	***6.2%***	8.2%	6.5%	15.7%	7.7%	7.4%	16.1%	7.3%
L8a	5.6%	***5.0%***	7.1%	8.2%	6.5%	16.9%	8.2%	6.3%	20.2%	7.3%
L8b	***8.1%***	11.4%	9.8%	16.5%	13.0%	22.8%	15.9%	9.4%	24.8%	13.0%
M7B	13.1%	13.1%	11.7%	14.3%	11.8%	20.1%	14.0%	12.5%	14.6%	***11.2%***
P6	6.5%	7.3%	***6.3%***	9.1%	7.0%	10.6%	8.7%	7.0%	10.3%	8.0%
2 months	8.4%	***5.1%***	5.2%	6.6%	5.3%	9.4%	6.4%	5.5%	9.3%	5.5%
Long	7.7%	***4.8%***	***4.8%***	6.0%	***4.8%***	8.3%	5.8%	5.4%	8.5%	5.1%
A2	5.5%	4.6%	4.3%	5.0%	3.9%	7.0%	4.7%	***3.8%***	8.9%	4.0%
A5	4.6%	5.2%	5.2%	***4.3%***	5.4%	6.3%	4.4%	4.4%	4.8%	4.7%
A99	4.2%	3.4%	3.9%	4.9%	3.7%	7.7%	4.9%	***3.3%***	5.9%	3.9%
B2	5.2%	5.4%	5.4%	5.8%	5.7%	8.4%	5.7%	***4.9%***	10.7%	***4.9%***
B4	6.0%	***4.2%***	5.2%	4.5%	5.7%	6.1%	4.8%	4.7%	4.8%	4.4%
B99	4.6%	4.8%	4.6%	***3.5%***	4.9%	4.1%	4.0%	3.6%	3.7%	***3.5%***
C7	4.0%	4.0%	3.3%	5.1%	***3.1%***	8.4%	4.5%	4.8%	9.0%	4.7%
C99	3.4%	3.2%	3.2%	4.8%	***2.5%***	8.2%	4.4%	4.4%	8.7%	4.4%
CA	5.3%	4.0%	3.5%	3.5%	***3.2%***	4.4%	3.5%	3.7%	4.3%	3.3%
CC	4.1%	4.1%	3.4%	5.0%	***3.1%***	8.6%	4.5%	5.0%	9.0%	4.7%
D1	4.1%	3.7%	3.6%	3.9%	3.5%	4.2%	3.9%	***3.2%***	4.6%	3.4%
D2	3.7%	3.6%	3.4%	4.0%	3.3%	6.1%	3.9%	***3.2%***	6.2%	3.4%

(*continued on next page*)

Table B.12 (*continued*).

D3	5.8%	4.6%	***4.4%***	6.7%	4.5%	10.2%	5.8%	5.3%	12.9%	5.5%
D99	3.1%	3.1%	3.3%	3.7%	3.1%	5.5%	3.6%	***2.9%***	6.6%	3.0%
E1	8.2%	5.7%	6.1%	6.6%	6.4%	6.4%	6.1%	***5.3%***	8.2%	5.8%
E2	9.9%	6.5%	***6.3%***	7.3%	7.3%	8.7%	7.5%	6.6%	9.9%	6.4%
E99	6.6%	***3.6%***	***3.6%***	4.2%	3.7%	4.1%	4.1%	4.1%	5.0%	***3.6%***
F1	8.3%	4.5%	4.4%	4.8%	4.2%	5.4%	4.4%	4.4%	4.7%	***4.0%***
F2	4.5%	1.8%	2.0%	2.3%	2.4%	3.1%	2.4%	***1.7%***	3.3%	2.1%
F3	5.5%	***3.3%***	4.2%	5.4%	3.7%	7.7%	5.2%	3.6%	8.1%	4.3%
F5	11.1%	5.8%	5.8%	7.0%	6.1%	7.9%	7.0%	6.6%	9.6%	***5.5%***
F6	11.8%	***4.1%***	5.7%	6.0%	5.6%	6.9%	6.2%	4.7%	6.1%	5.1%
F8	9.3%	***3.1%***	3.5%	4.9%	3.7%	8.2%	5.3%	3.7%	4.6%	4.0%
F99	5.2%	3.7%	3.5%	3.5%	3.5%	5.2%	3.7%	3.6%	6.1%	***3.0%***
G99	54.8%	18.4%	19.0%	30.2%	***18.1%***	38.7%	27.5%	20.4%	43.1%	23.8%
J0	4.8%	3.6%	2.8%	4.4%	***2.7%***	6.7%	4.2%	2.9%	6.7%	2.9%
J4	6.3%	3.2%	***3.0%***	4.3%	3.5%	6.4%	4.1%	3.2%	5.8%	3.5%
J6	5.5%	3.1%	3.5%	5.3%	***2.9%***	8.0%	5.0%	3.2%	8.7%	3.6%
J7	5.7%	4.4%	3.5%	3.9%	3.8%	5.6%	3.7%	3.9%	4.8%	***2.8%***
J9	4.2%	3.6%	3.1%	4.0%	3.0%	6.5%	4.0%	3.0%	5.8%	***2.6%***
J99	4.5%	2.3%	2.3%	4.6%	***2.1%***	7.9%	4.6%	2.6%	7.5%	3.0%
JA	4.9%	***2.0%***	2.3%	4.9%	2.2%	8.3%	4.8%	3.3%	8.8%	3.5%
L1	11.7%	10.8%	10.3%	10.0%	***8.7%***	10.5%	9.1%	9.7%	11.5%	9.0%
L2a	8.2%	5.3%	5.5%	5.3%	5.6%	5.7%	5.6%	***4.1%***	5.3%	4.7%
L2b	16.2%	9.5%	9.4%	12.7%	8.5%	16.7%	12.6%	***8.1%***	18.8%	10.6%
L2c	5.8%	***2.3%***	***2.3%***	5.0%	2.7%	8.4%	4.8%	3.6%	8.4%	3.8%
L99	4.8%	***2.2%***	2.4%	4.0%	2.4%	7.8%	3.8%	3.4%	7.1%	3.2%
LA	5.3%	***2.3%***	2.9%	4.3%	2.7%	8.5%	4.0%	3.6%	7.3%	3.5%
LS	4.7%	***3.2%***	***3.2%***	4.7%	3.4%	8.7%	4.6%	3.6%	9.1%	4.0%
M1	12.0%	9.1%	***8.7%***	9.2%	9.1%	13.4%	9.3%	12.6%	12.2%	9.1%
M2	11.1%	9.6%	9.9%	10.9%	9.0%	12.6%	9.5%	9.0%	11.2%	***8.3%***
M8	26.1%	15.7%	15.2%	***14.7%***	18.3%	19.1%	16.1%	27.1%	16.4%	15.5%
M99	11.4%	7.8%	8.0%	9.0%	***7.4%***	11.8%	8.5%	10.4%	10.6%	7.6%
P1	6.7%	***4.3%***	4.7%	5.1%	4.5%	6.4%	4.9%	5.5%	7.3%	4.9%
P2	4.9%	***3.4%***	***3.4%***	5.0%	3.5%	6.8%	4.8%	3.7%	7.6%	3.7%
P5	4.8%	3.5%	***3.1%***	5.4%	3.7%	7.7%	5.2%	4.2%	9.9%	4.0%
P99	4.8%	***2.7%***	2.8%	3.9%	3.0%	5.6%	3.8%	3.9%	5.0%	3.7%
S2	5.2%	***2.0%***	2.8%	5.1%	2.4%	7.4%	4.9%	4.5%	5.9%	3.7%
S3	3.0%	2.7%	3.0%	3.7%	***2.6%***	6.4%	3.3%	3.2%	7.4%	3.1%
T99	4.4%	***1.9%***	2.6%	4.7%	2.1%	6.6%	4.6%	4.0%	5.9%	3.3%
Medium	8.0%	***4.4%***	4.8%	6.3%	4.7%	9.1%	6.1%	5.0%	8.7%	5.2%
A3	5.4%	***4.2%***	4.7%	6.2%	***4.2%***	9.4%	5.9%	4.6%	6.9%	4.9%
A4	6.4%	3.0%	4.3%	4.5%	4.6%	6.5%	4.4%	***2.6%***	4.6%	3.6%
B3	10.0%	6.2%	6.2%	9.4%	6.6%	15.9%	8.7%	***5.7%***	17.4%	7.4%
C10	5.1%	***3.3%***	3.7%	5.9%	***3.3%***	8.5%	5.6%	4.7%	9.7%	5.3%
C8	8.8%	***6.4%***	6.6%	8.8%	6.5%	10.6%	8.2%	8.1%	10.0%	7.2%
C9	6.4%	5.0%	4.8%	7.0%	***4.4%***	11.2%	7.1%	5.4%	13.4%	6.1%
CB	4.6%	***3.1%***	***3.1%***	5.8%	***3.1%***	8.7%	5.1%	4.5%	11.0%	5.0%
F7	6.6%	4.2%	3.3%	4.8%	***3.1%***	6.5%	4.8%	3.8%	6.1%	3.7%
H99	3.9%	***1.9%***	2.0%	3.4%	***1.9%***	7.2%	3.1%	2.8%	8.0%	2.6%
JB	10.2%	3.4%	***3.0%***	5.9%	3.4%	5.3%	6.0%	4.8%	5.8%	4.1%
JBa	12.5%	***3.5%***	***3.5%***	7.0%	4.0%	7.1%	6.7%	4.9%	7.0%	4.0%
JBb	11.5%	***6.3%***	7.1%	8.8%	6.7%	12.9%	9.1%	7.1%	11.8%	7.5%
JC	6.8%	4.1%	4.6%	4.1%	4.5%	5.7%	4.4%	5.1%	4.4%	***4.0%***
JCc	10.5%	8.5%	12.0%	8.2%	8.9%	9.3%	8.7%	8.0%	9.3%	***7.7%***
JCd	8.5%	5.6%	***5.5%***	6.0%	***5.5%***	7.1%	6.1%	7.0%	6.0%	5.9%
JCe	7.1%	***4.7%***	6.2%	5.2%	6.2%	6.6%	5.2%	5.2%	5.3%	4.9%
JD	6.3%	3.5%	***3.0%***	5.6%	***3.0%***	9.5%	5.6%	4.0%	8.0%	4.3%
JE	6.0%	***2.2%***	2.7%	6.8%	2.6%	10.4%	6.5%	4.2%	11.2%	4.9%
JF	5.0%	3.0%	***2.7%***	4.5%	3.0%	7.8%	4.2%	3.9%	8.8%	3.6%
L3	13.2%	***7.0%***	7.8%	9.9%	8.6%	12.4%	9.3%	***7.0%***	13.1%	8.0%
L5	7.4%	3.2%	4.5%	4.2%	4.5%	6.3%	4.3%	***2.7%***	7.3%	3.6%
L6	7.5%	***4.3%***	4.4%	7.2%	4.5%	11.1%	6.6%	5.8%	10.1%	5.9%
L6g	8.6%	***4.3%***	5.0%	8.0%	5.2%	11.2%	7.4%	6.2%	10.0%	6.7%
L6h	6.2%	3.6%	3.6%	5.2%	***3.5%***	8.7%	5.0%	3.8%	8.1%	4.2%
L6i	11.1%	***5.7%***	6.9%	8.9%	6.5%	12.8%	8.9%	5.8%	7.5%	6.7%
L7	5.8%	3.0%	***2.9%***	4.9%	***2.9%***	6.6%	4.7%	4.0%	7.4%	3.8%
L8	5.6%	2.9%	2.7%	5.3%	***2.5%***	13.3%	4.7%	3.6%	16.2%	4.0%
L9	13.4%	5.6%	6.9%	7.7%	6.1%	11.1%	7.4%	***5.3%***	9.7%	6.5%
M6	5.7%	4.1%	4.5%	3.7%	4.8%	5.2%	3.9%	4.7%	4.1%	***3.4%***
M7A	14.0%	***4.5%***	5.3%	6.3%	5.0%	8.2%	6.1%	5.7%	5.9%	5.0%
P4	6.4%	***5.1%***	5.5%	5.5%	5.5%	7.8%	5.7%	***5.1%***	7.0%	5.2%

(*continued on next page*)

Table B.12 (*continued*).

Short	9.9%	6.5%	***6.3%***	8.0%	6.6%	11.5%	7.8%	***6.3%***	11.5%	6.8%
F10	15.8%	20.7%	15.0%	15.3%	18.9%	15.6%	15.1%	14.0%	17.5%	***13.7%***
F9	5.6%	3.4%	4.0%	5.2%	***3.0%***	9.3%	5.1%	4.2%	11.2%	4.5%
H1	9.3%	4.1%	3.9%	6.1%	5.2%	11.5%	6.3%	***3.8%***	12.5%	4.5%
H2	7.5%	3.7%	***3.3%***	5.0%	3.4%	6.2%	5.2%	3.9%	6.9%	4.1%
H3	5.2%	***2.5%***	2.9%	5.2%	3.0%	5.3%	5.2%	3.9%	5.6%	4.0%
H4	5.8%	4.4%	***4.1%***	5.0%	4.3%	6.7%	4.9%	***4.1%***	7.7%	4.2%
H6	7.1%	***3.4%***	3.9%	5.3%	3.5%	7.5%	5.3%	4.2%	12.4%	4.4%
JBc	20.6%	14.0%	14.8%	17.5%	13.8%	24.0%	16.3%	***12.6%***	18.5%	13.7%
JCf	20.7%	***14.1%***	17.1%	21.2%	19.5%	26.3%	19.0%	14.2%	23.6%	17.8%
JDa	8.1%	4.2%	***3.6%***	6.1%	4.3%	9.3%	6.4%	4.6%	10.2%	5.2%
JDb	6.7%	***3.9%***	5.3%	6.6%	4.0%	15.7%	6.3%	4.4%	18.0%	5.6%
JDc	7.1%	6.6%	7.2%	***5.5%***	6.3%	9.0%	5.6%	8.2%	8.5%	5.8%
JDd	21.9%	***11.7%***	12.2%	13.5%	12.2%	16.6%	14.2%	13.1%	14.8%	12.3%
JDe	6.7%	***4.6%***	5.8%	4.9%	4.8%	8.4%	4.8%	5.3%	4.9%	4.7%
JEa	6.7%	***4.5%***	5.4%	6.7%	5.0%	9.5%	6.4%	6.0%	11.3%	6.2%
JEb	6.0%	4.2%	3.8%	7.1%	***3.0%***	10.1%	7.4%	5.0%	10.7%	5.9%
L3a	17.5%	8.5%	7.8%	9.2%	9.1%	10.9%	8.6%	***7.7%***	10.0%	8.5%
L3b	22.1%	15.1%	14.7%	16.7%	16.4%	21.6%	17.4%	***14.5%***	21.2%	16.3%
L3c	7.8%	9.8%	***2.8%***	5.7%	4.3%	9.2%	5.5%	3.4%	13.0%	4.2%
L4	7.4%	4.0%	4.2%	4.8%	3.7%	9.1%	4.7%	***3.5%***	7.7%	3.7%
L5a	7.1%	***2.9%***	4.0%	3.8%	3.5%	6.0%	3.7%	***2.9%***	5.2%	3.3%
L5b	9.7%	3.9%	5.2%	7.5%	9.0%	13.4%	6.9%	***3.1%***	11.3%	5.5%
L5c	8.9%	4.8%	4.4%	6.4%	4.9%	9.5%	6.4%	***4.3%***	7.5%	4.8%
L7a/b/c/d/f	6.8%	3.2%	3.0%	5.1%	***2.7%***	6.7%	4.9%	3.9%	6.9%	3.7%
L7e	6.6%	4.5%	***3.9%***	5.8%	4.4%	11.4%	5.6%	5.0%	11.2%	5.2%
L8a	5.6%	3.4%	***3.1%***	4.5%	3.2%	11.3%	4.6%	3.3%	12.9%	3.7%
L8b	8.1%	6.2%	***5.7%***	10.2%	6.3%	14.7%	10.0%	6.1%	16.4%	8.3%
M7B	13.1%	9.2%	7.7%	9.2%	***6.7%***	11.9%	8.4%	10.1%	9.3%	7.5%
P6	6.5%	4.3%	***4.1%***	5.8%	4.3%	6.4%	5.7%	4.7%	6.0%	4.9%

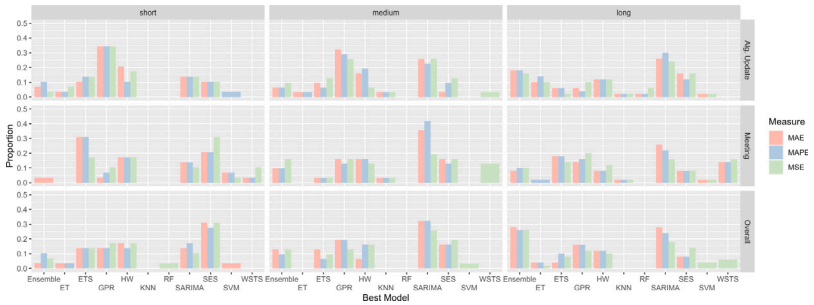


Fig. C.5. The bar chart provides insight into the comparative performance of forecasting methods across different time series lengths and evaluation metrics. It depicts the distribution of the best-performing forecasts across 110 WSTS product categories, categorized by Mean Squared Error (MSE) shown in red, Mean Absolute Percentage Error (MAPE) in blue, and Mean Absolute Error (MAE) in green. Each column represents a different time series length category: the left column represents short time series, the middle column represents medium-length series, and the rightmost column represents long series (those with full available history). The rows of the chart are arranged by algorithmic update, expert forecasts ("Meeting"), and overall performance, following the structure of Tables 4 and 6.

Table B.13
Mean absolute errors (MAE) for the 110 examined WSTS product categories, which are organized by time series length and presented for the 3-month forecasts (first) and 2-month forecast (second). Averages are contained in the long, medium, short, 3-months, and 2-months rows respectively. The best result in each row, i.e. for each time series, is printed bold and italic.

ID	WSTS	SARIMA	ETS	ET	GPR	KNN	RF	SES	SVM	Ensemble
3 months	3.12	3.01	3.10	4.43	***2.97***	6.93	4.27	3.59	6.59	3.44
Long	5.24	4.92	5.18	7.63	***4.83***	12.03	7.35	6.18	11.27	5.82
A2	***0.11***	0.13	***0.11***	0.16	0.14	0.26	0.15	0.12	0.34	0.13
A5	***0.09***	0.14	0.13	0.10	0.12	0.18	0.11	0.12	0.16	0.11
A99	***0.34***	0.43	0.44	0.62	0.40	1.03	0.62	0.43	0.90	0.48
B2	0.14	***0.13***	0.14	0.24	0.17	0.36	0.23	0.15	0.44	0.17
B4	***0.13***	0.20	0.17	0.16	0.16	0.21	0.17	0.15	0.17	0.16
B99	***0.25***	0.36	0.33	0.28	0.32	0.32	0.30	0.31	0.28	0.29
C7	***0.94***	1.15	0.98	1.65	1.04	3.16	1.41	1.10	3.45	1.40
C99	***1.58***	1.79	1.59	3.08	1.62	5.64	2.66	1.79	6.34	2.41
CA	0.26	0.26	***0.23***	0.26	0.25	0.37	0.26	0.28	0.35	0.27
CC	***1.03***	1.20	***1.03***	1.76	1.12	3.45	1.55	1.25	3.60	1.50
D1	***0.15***	0.19	0.18	0.20	0.20	0.26	0.19	0.18	0.24	0.19
D2	***0.14***	0.18	0.18	0.24	0.19	0.33	0.25	0.15	0.38	0.19
D3	0.12	***0.11***	0.13	0.19	0.13	0.33	0.17	***0.11***	0.45	0.16
D99	***0.30***	0.47	0.38	0.56	0.46	0.81	0.51	0.36	1.04	0.43
E1	***0.07***	0.08	0.09	0.09	***0.07***	0.08	0.09	0.08	0.10	0.08
E2	***0.09***	***0.09***	***0.09***	0.11	0.10	0.13	0.11	***0.09***	0.14	0.10
E99	***0.12***	0.13	0.13	0.13	***0.12***	0.15	***0.12***	0.14	0.15	0.13
F1	0.08	0.06	0.06	0.07	***0.05***	0.09	0.07	0.06	0.10	0.06
F2	1.34	1.18	1.17	1.20	1.36	1.56	1.22	***1.08***	1.71	1.17
F3	0.27	0.29	0.26	0.36	0.27	0.52	0.34	***0.25***	0.62	0.28
F5	5.02	5.27	5.23	4.78	5.20	5.32	4.91	7.17	5.98	***4.40***
F6	0.69	0.57	0.62	0.55	0.65	0.68	0.59	0.59	0.55	***0.54***
F8	0.15	***0.11***	0.13	0.15	0.13	0.23	0.14	***0.11***	0.14	0.12
F99	***5.27***	5.67	7.28	6.71	6.83	7.80	6.68	8.42	10.16	5.71
G99	0.07	0.07	0.08	0.07	0.08	***0.06***	0.07	0.07	0.08	0.07
J0	3.24	***3.17***	3.86	5.55	3.41	8.51	5.41	3.70	8.21	4.17
J4	***0.41***	0.42	0.44	0.54	***0.41***	0.82	0.48	0.44	0.74	0.47
J6	***2.34***	2.58	2.92	4.02	2.37	6.35	3.91	2.73	6.02	3.07
J7	0.54	0.71	***0.52***	0.77	0.58	0.98	0.74	0.60	0.87	0.66
J9	***0.39***	0.51	0.49	0.68	0.48	1.04	0.62	0.42	1.05	0.48
J99	7.54	***6.82***	8.06	13.25	7.55	22.57	13.55	9.18	23.72	10.31
JA	4.82	***4.34***	4.90	8.05	4.83	14.19	7.79	5.86	15.31	6.22
L1	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***
L2a	0.29	0.30	0.34	0.28	***0.25***	0.33	0.28	0.28	0.29	0.26
L2b	***0.03***	***0.03***	***0.03***	***0.03***	***0.03***	0.04	***0.03***	***0.03***	0.04	***0.03***
L2c	2.54	***2.37***	2.41	4.33	2.86	7.14	3.73	2.89	7.68	3.31
L99	16.95	***12.38***	15.58	22.30	14.32	41.44	20.87	18.34	41.94	17.83
LA	14.34	***9.99***	13.15	17.50	12.99	35.16	15.81	14.65	31.89	13.93
LS	***3.63***	5.08	4.45	6.10	4.92	11.26	5.63	4.20	13.37	5.24
M1	22.04	24.78	20.41	27.80	***19.65***	42.46	28.39	23.18	37.20	21.69
M2	0.10	0.10	0.09	0.12	0.09	0.13	0.12	***0.08***	0.13	0.09
M8	0.36	0.34	0.37	0.31	0.33	***0.30***	0.36	0.49	0.32	0.32
M99	34.16	36.70	28.63	40.73	***27.69***	63.70	41.84	34.76	58.82	32.47
P1	***8.12***	9.11	10.55	10.41	9.40	11.84	9.95	10.80	13.19	9.88
P2	***2.38***	2.98	2.98	3.95	2.99	6.11	3.70	2.64	7.30	3.35
P5	***2.05***	2.69	2.69	3.73	3.03	6.01	3.55	2.25	7.88	3.14
P99	***8.74***	9.41	11.14	11.83	9.16	15.93	10.90	11.38	13.06	10.75
S2	52.17	***41.34***	49.89	80.01	43.02	123.26	76.37	63.37	107.91	56.24
S3	***2.16***	3.13	2.86	3.78	2.57	7.51	3.11	2.52	8.91	3.11
T99	53.76	***46.24***	50.85	91.83	47.35	141.27	87.40	69.58	119.65	63.31
Medium	***1.29***	1.33	1.33	1.79	1.36	2.74	1.71	1.40	2.82	1.47
A3	0.21	0.19	0.25	0.34	***0.17***	0.49	0.30	0.24	0.40	0.23
A4	***0.04***	***0.04***	***0.04***	***0.04***	***0.04***	0.05	***0.04***	***0.04***	***0.04***	***0.04***
B3	***0.06***	0.08	***0.06***	0.08	0.07	0.13	0.08	0.07	0.14	0.07
C10	***0.60***	0.76	***0.60***	0.90	0.63	1.56	0.85	0.64	1.74	0.67
C8	0.15	0.15	***0.14***	0.22	***0.14***	0.30	0.20	0.15	0.32	0.17
C9	0.28	0.27	0.25	0.42	***0.24***	0.75	0.38	0.25	0.93	0.30

(*continued on next page*)

Table B.13 (*continued*).

CB	0.76	0.92	***0.72***	1.21	***0.72***	2.32	1.05	0.75	2.67	0.88
F7	0.17	***0.16***	***0.16***	0.22	***0.16***	0.27	0.22	0.18	0.27	0.17
H99	***1.65***	1.72	1.98	2.45	1.84	4.70	2.36	1.87	5.53	2.05
JB	0.58	0.71	0.64	0.64	***0.56***	0.61	0.61	0.67	0.63	0.58
JBa	0.40	0.42	***0.37***	0.41	0.38	0.42	0.40	0.41	0.41	0.38
JBb	***0.28***	0.34	0.33	0.43	0.34	0.58	0.43	0.36	0.51	0.35
JC	0.45	0.47	0.56	0.48	0.54	0.59	***0.44***	0.54	0.46	0.47
JCc	***0.12***	0.14	0.17	0.19	0.19	0.20	0.18	0.13	0.23	0.16
JCd	***0.33***	0.43	0.37	0.39	0.42	0.45	0.37	0.38	0.36	0.37
JCe	***0.08***	0.10	0.10	0.10	0.09	0.15	0.10	0.10	0.09	0.10
JD	3.16	3.17	3.14	4.53	***2.92***	7.57	4.59	3.45	7.69	3.57
JE	1.57	1.56	***1.54***	2.87	1.71	4.85	2.81	1.65	5.79	2.36
JF	***0.43***	0.53	0.51	0.69	0.46	1.31	0.64	0.53	1.45	0.60
L3	***2.54***	2.65	2.65	3.22	2.83	4.97	3.38	2.57	4.63	2.92
L5	2.20	2.00	2.40	2.95	2.50	4.45	3.35	***1.77***	4.31	2.55
L6	***6.43***	6.57	6.72	9.13	6.93	14.40	8.43	7.44	13.71	7.78
L6g	5.61	***5.38***	6.04	8.25	6.58	11.14	7.61	6.31	10.80	7.05
L6h	0.90	***0.83***	0.90	1.20	0.88	2.05	1.18	0.87	1.93	0.99
L6i	0.54	***0.49***	0.60	0.66	0.57	0.86	0.63	0.51	0.62	0.55
L7	7.77	8.77	7.49	10.36	7.60	14.48	9.36	8.99	15.96	***7.48***
L8	0.78	***0.76***	0.86	1.26	0.86	2.95	1.16	0.91	3.61	1.04
L9	0.48	***0.42***	0.48	0.72	0.80	0.96	0.65	0.43	0.96	0.64
M6	0.16	0.17	0.22	0.16	0.16	0.22	0.16	***0.15***	0.20	***0.15***
M7A	0.76	0.38	0.45	0.56	***0.37***	0.58	0.52	0.51	0.52	0.44
P4	***0.44***	0.55	0.57	0.54	0.57	0.62	0.52	0.49	0.48	0.50
Short	***1.42***	1.52	***1.42***	1.73	1.47	2.62	1.68	1.48	2.56	1.44
F10	***0.64***	1.04	0.77	1.16	1.33	1.28	1.18	0.99	1.25	1.00
F9	0.34	0.32	0.34	0.42	***0.30***	0.94	0.41	0.38	1.01	0.38
H1	0.49	***0.44***	0.48	0.65	0.50	1.06	0.64	0.48	1.16	0.54
H2	0.50	0.62	0.59	0.53	0.53	0.68	0.53	0.57	0.72	***0.47***
H3	***0.39***	0.46	0.50	0.63	0.45	0.72	0.61	0.46	0.83	0.48
H4	***0.31***	0.44	0.34	0.39	0.36	0.53	0.36	0.34	0.65	0.34
H6	***1.32***	1.36	1.46	1.69	1.51	2.05	1.73	1.50	3.81	1.49
JBc	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***	***0.02***
JCf	***0.05***	***0.05***	***0.05***	***0.05***	***0.05***	0.09	0.06	***0.05***	0.06	***0.05***
JDa	***2.53***	2.57	2.67	3.49	2.68	5.30	3.43	2.84	5.45	2.99
JDb	0.33	0.34	0.38	0.48	***0.30***	1.15	0.47	0.34	1.46	0.40
JDc	***0.44***	0.53	0.57	0.63	0.61	0.91	0.59	0.53	0.86	0.50
JDd	0.52	0.64	***0.45***	0.59	0.57	0.70	0.57	0.52	0.81	0.53
JDe	0.30	0.27	0.28	0.30	0.25	0.52	0.28	0.26	***0.24***	0.26
JEa	***0.21***	0.25	0.32	0.37	0.34	0.54	0.33	0.30	0.68	0.33
JEb	1.39	***1.33***	1.84	2.81	2.41	4.25	2.76	1.78	4.57	2.48
L3a	1.19	***1.02***	***1.02***	1.11	1.03	1.28	1.11	***1.02***	1.04	1.04
L3b	***1.63***	1.75	1.71	1.88	1.80	2.73	2.11	1.70	2.38	1.89
L3c	0.36	0.79	0.34	0.52	0.46	0.88	0.49	***0.32***	1.07	0.42
L4	0.72	0.69	0.99	0.87	***0.53***	1.45	0.84	0.60	1.36	0.62
L5a	1.26	1.32	1.33	1.26	1.35	2.23	1.27	1.18	1.71	***1.12***
L5b	0.91	0.98	1.01	1.32	1.64	2.35	1.13	***0.71***	2.12	1.10
L5c	0.18	0.18	***0.16***	0.21	0.18	0.31	0.20	***0.16***	0.22	0.17
L7a/b/c/d/f	7.20	8.80	7.26	8.78	7.14	12.73	8.21	8.61	12.72	***7.06***
L7e	2.35	2.29	***2.16***	2.72	2.24	5.55	2.62	2.53	6.08	2.54
L8a	0.53	***0.45***	0.66	0.78	0.58	1.80	0.77	0.59	2.07	0.69
L8b	***0.36***	0.44	0.42	0.71	0.55	1.07	0.68	0.39	1.17	0.55
M7B	14.08	14.25	12.70	15.10	12.47	21.94	14.56	13.18	17.81	***11.64***
P6	0.50	0.53	***0.47***	0.73	0.54	0.82	0.71	0.52	0.84	0.63
2 months	3.12	***1.74***	1.92	2.87	1.80	4.41	2.76	2.56	4.18	2.28
Long	5.24	***2.87***	3.28	4.95	3.02	7.64	4.77	4.51	7.13	3.90
A2	0.11	0.10	0.09	0.10	0.08	0.14	0.09	***0.07***	0.20	0.08
A5	0.09	0.10	0.10	***0.08***	0.11	0.12	0.09	***0.08***	0.09	0.09
A99	0.34	0.29	0.33	0.40	0.32	0.66	0.41	***0.27***	0.51	0.32
B2	0.14	0.15	0.15	0.16	0.16	0.23	0.15	***0.13***	0.29	***0.13***

(*continued on next page*)

Table B.13 (*continued*).

B4	0.13	***0.10***	0.12	***0.10***	0.13	0.13	***0.10***	0.11	***0.10***	***0.10***
B99	0.25	0.28	0.27	***0.20***	0.29	0.24	0.22	0.21	0.21	***0.20***
C7	0.94	1.04	0.84	1.21	***0.79***	2.05	1.06	1.16	2.21	1.12
C99	1.58	1.53	1.48	2.23	***1.18***	3.85	2.02	2.04	4.09	2.04
CA	0.26	0.20	0.17	0.18	***0.16***	0.22	0.17	0.18	0.22	***0.16***
CC	1.03	1.16	0.94	1.32	***0.84***	2.27	1.16	1.30	2.37	1.23
D1	0.15	0.14	0.13	0.15	0.13	0.16	0.15	***0.12***	0.18	0.13
D2	0.14	0.14	0.13	0.15	0.13	0.24	0.15	***0.12***	0.25	0.13
D3	0.12	***0.10***	***0.10***	0.14	***0.10***	0.23	0.13	0.12	0.29	0.12
D99	0.30	0.31	0.33	0.36	0.31	0.55	0.35	***0.29***	0.67	0.30
E1	0.07	***0.05***	0.06	0.06	0.06	0.06	***0.05***	***0.05***	0.07	***0.05***
E2	0.09	***0.06***	***0.06***	0.07	0.07	0.08	0.07	***0.06***	0.10	***0.06***
E99	0.12	***0.07***	***0.07***	0.08	***0.07***	0.08	0.08	0.08	0.10	***0.07***
F1	0.08	0.04	0.04	0.04	0.04	0.05	0.04	0.04	0.04	***0.03***
F2	1.34	***0.54***	0.61	0.71	0.72	0.94	0.74	***0.54***	1.03	0.63
F3	0.27	***0.17***	0.20	0.26	0.18	0.38	0.25	***0.17***	0.40	0.21
F5	5.02	2.83	2.86	3.44	2.94	3.75	3.47	3.26	4.49	***2.74***
F6	0.69	***0.28***	0.38	0.36	0.37	0.41	0.38	0.31	0.34	0.32
F8	0.15	***0.05***	0.06	0.08	0.06	0.14	0.09	0.06	0.08	0.07
F99	5.27	3.87	3.74	3.75	3.68	5.45	3.86	3.89	6.32	***3.21***
G99	0.07	***0.04***	***0.04***	0.05	0.05	0.06	0.05	***0.04***	0.06	0.05
J0	3.24	2.53	1.93	3.08	***1.84***	5.02	2.97	1.95	4.96	2.08
J4	0.41	0.20	***0.19***	0.29	0.22	0.45	0.27	0.20	0.41	0.23
J6	2.34	1.41	1.52	2.35	***1.30***	3.68	2.17	1.40	3.88	1.59
J7	0.54	0.40	0.31	0.38	0.35	0.58	0.36	0.37	0.49	***0.27***
J9	0.39	0.35	0.30	0.38	0.31	0.66	0.38	0.29	0.59	***0.25***
J99	7.54	4.04	3.93	7.90	***3.64***	14.52	7.88	4.48	13.99	5.18
JA	4.82	***2.01***	2.36	4.98	2.27	8.97	4.85	3.32	9.58	3.61
L1	0.02	0.02	0.02	0.02	***0.01***	0.02	***0.01***	0.02	0.02	***0.01***
L2a	0.29	0.19	0.20	0.18	0.20	0.19	0.20	***0.15***	0.18	0.16
L2b	0.03	0.02	0.02	0.02	***0.01***	0.03	0.02	***0.01***	0.03	0.02
L2c	2.54	***1.04***	1.05	2.38	1.21	4.38	2.25	1.67	4.37	1.81
L99	16.95	***7.76***	8.77	14.06	8.26	29.01	13.48	11.80	26.07	11.43
LA	14.34	***6.40***	8.29	11.67	7.47	24.51	11.02	9.63	20.64	9.71
LS	3.63	2.41	***2.32***	3.58	2.43	7.35	3.51	2.64	7.82	3.02
M1	22.04	16.54	***15.55***	17.62	16.82	25.79	18.30	23.26	24.94	16.91
M2	0.10	0.08	0.08	0.09	0.08	0.11	0.08	0.08	0.10	***0.07***
M8	0.36	0.22	0.22	***0.21***	0.26	0.24	0.23	0.41	0.22	0.22
M99	34.16	24.34	24.28	28.90	***23.20***	38.08	27.67	32.93	36.52	24.03
P1	8.12	***5.24***	5.66	6.43	5.42	8.13	6.08	6.82	9.24	6.19
P2	2.38	1.59	***1.58***	2.46	1.62	3.77	2.36	1.82	4.20	1.86
P5	2.05	1.47	***1.31***	2.34	1.53	3.80	2.26	1.83	4.86	1.82
P99	8.74	***4.96***	5.21	7.19	5.41	10.51	7.00	7.23	9.49	6.98
S2	52.17	***21.15***	29.51	52.70	25.16	79.43	50.38	46.24	65.43	39.02
S3	2.16	2.10	2.24	2.72	***2.01***	4.73	2.44	2.32	5.53	2.31
T99	53.76	***23.59***	33.75	59.66	27.24	85.43	56.83	49.92	78.19	42.75
Medium	1.29	***0.70***	0.74	1.15	0.74	1.79	1.09	0.89	1.80	0.93
A3	0.21	***0.16***	0.18	0.24	***0.16***	0.37	0.22	0.17	0.28	0.19
A4	0.04	***0.02***	***0.02***	0.03	0.03	0.04	0.03	***0.02***	0.03	***0.02***
B3	0.06	***0.04***	***0.04***	0.06	***0.04***	0.09	0.05	***0.04***	0.09	0.05
C10	0.60	0.38	0.40	0.68	***0.37***	0.99	0.64	0.54	1.12	0.61
C8	0.15	***0.12***	***0.12***	0.17	***0.12***	0.22	0.16	0.15	0.21	0.14
C9	0.28	0.23	0.21	0.32	***0.19***	0.51	0.32	0.24	0.61	0.27
CB	0.76	0.50	0.49	0.92	***0.47***	1.43	0.80	0.72	1.74	0.79
F7	0.17	0.11	***0.08***	0.13	***0.08***	0.18	0.13	0.10	0.16	0.10
H99	1.65	***0.81***	0.89	1.55	0.85	3.32	1.41	1.21	3.56	1.14
JB	0.58	0.21	***0.18***	0.34	0.21	0.30	0.34	0.29	0.32	0.24
JBa	0.40	***0.12***	***0.12***	0.22	0.14	0.22	0.21	0.17	0.21	0.13
JBb	0.28	***0.16***	0.18	0.25	0.17	0.37	0.25	0.18	0.34	0.20
JC	0.45	***0.27***	0.30	***0.27***	0.30	0.38	0.30	0.34	0.29	***0.27***
JCc	0.12	***0.10***	0.16	***0.10***	0.12	0.12	0.11	***0.10***	0.12	***0.10***
JCd	0.33	0.23	***0.21***	0.23	***0.21***	0.27	0.24	0.28	0.23	0.23
JCe	0.08	***0.06***	0.07	***0.06***	0.07	0.07	***0.06***	***0.06***	0.07	***0.06***
JD	3.16	1.75	***1.55***	2.88	***1.55***	5.25	2.88	2.00	4.52	2.23
JE	1.57	***0.62***	0.77	1.90	0.76	3.20	1.82	1.22	3.41	1.43
JF	0.43	0.28	***0.25***	0.39	0.30	0.72	0.36	0.34	0.86	0.32
L3	2.54	***1.28***	1.43	1.98	1.66	2.76	1.87	***1.28***	3.01	1.57
L5	2.20	0.92	1.30	1.27	1.27	2.09	1.32	***0.81***	2.30	1.09
L6	6.43	***3.92***	4.01	6.11	4.06	9.71	5.72	5.14	9.01	5.20
L6g	5.61	***3.08***	3.59	5.35	3.66	7.60	4.97	4.31	6.91	4.60
L6h	0.90	0.52	0.53	0.75	***0.50***	1.35	0.73	0.55	1.26	0.60
L6i	0.54	***0.30***	0.36	0.44	0.35	0.62	0.44	0.31	0.38	0.34
L7	7.77	4.04	***3.87***	6.92	3.98	9.69	6.68	5.48	10.84	5.33
L8	0.78	0.41	0.36	0.74	***0.34***	1.97	0.65	0.52	2.36	0.57
L9	0.48	0.21	0.31	0.35	0.23	0.55	0.35	***0.20***	0.49	0.27
M6	0.16	0.11	0.13	0.10	0.13	0.14	0.11	0.13	0.12	***0.09***

(*continued on next page*)

Table B.13 (*continued*).

M7A	0.76	***0.26***	0.29	0.38	0.28	0.48	0.37	0.33	0.34	0.29
P4	0.44	***0.36***	0.39	0.39	0.39	0.52	0.40	***0.36***	0.48	0.37
Short	1.42	0.89	0.83	1.12	***0.81***	1.64	1.08	0.99	1.64	0.92
F10	0.64	0.90	0.66	0.67	0.82	0.72	0.66	0.64	0.81	***0.60***
F9	0.34	0.22	0.25	0.32	***0.20***	0.64	0.31	0.26	0.70	0.28
H1	0.49	0.22	***0.21***	0.35	0.28	0.66	0.36	***0.21***	0.70	0.25
H2	0.50	0.25	***0.22***	0.34	0.23	0.41	0.35	0.26	0.47	0.27
H3	0.39	***0.20***	0.23	0.41	0.23	0.41	0.40	0.32	0.44	0.31
H4	0.31	0.24	***0.22***	0.28	0.23	0.38	0.27	***0.22***	0.43	0.23
H6	1.32	***0.61***	0.70	0.96	0.64	1.41	0.94	0.75	2.52	0.79
JBc	0.02	***0.01***	***0.01***	***0.01***	***0.01***	0.02	***0.01***	***0.01***	***0.01***	***0.01***
JCf	0.05	***0.03***	0.04	0.04	0.05	0.06	0.04	***0.03***	0.05	0.04
JDa	2.53	1.34	***1.18***	1.99	1.37	3.13	2.07	1.46	3.56	1.69
JDb	0.33	0.18	0.24	0.30	***0.17***	0.78	0.29	0.19	0.95	0.25
JDc	0.44	0.43	0.46	***0.37***	0.42	0.62	***0.37***	0.53	0.58	0.38
JDd	0.52	***0.29***	0.30	0.35	0.30	0.47	0.37	0.33	0.42	0.32
JDe	0.30	0.21	0.26	0.21	0.21	0.37	0.21	0.24	***0.20***	***0.20***
JEa	0.21	***0.15***	0.17	0.22	0.17	0.33	0.21	0.20	0.43	0.20
JEb	1.39	0.92	0.98	1.79	***0.79***	2.78	1.86	1.28	2.88	1.51
L3a	1.19	0.66	0.59	0.67	0.68	0.79	0.61	***0.58***	0.75	0.63
L3b	1.63	1.12	1.07	1.34	1.24	1.77	1.38	***1.06***	1.74	1.25
L3c	0.36	0.35	***0.13***	0.26	0.19	0.44	0.25	0.16	0.66	0.19
L4	0.72	0.38	0.42	0.48	0.36	0.98	0.48	***0.35***	0.86	0.37
L5a	1.26	0.51	0.69	0.65	0.62	1.10	0.63	***0.50***	0.89	0.56
L5b	0.91	0.36	0.49	0.70	0.80	1.33	0.64	***0.29***	1.09	0.50
L5c	0.18	0.10	***0.09***	0.13	0.10	0.19	0.13	***0.09***	0.15	0.10
L7a/b/c/d/f	7.20	3.29	3.15	5.63	***2.98***	7.67	5.34	4.06	8.02	4.09
L7e	2.35	1.52	***1.37***	1.94	1.51	3.96	1.91	1.72	4.14	1.78
L8a	0.53	***0.29***	***0.29***	0.42	0.30	1.21	0.42	0.30	1.32	0.35
L8b	0.36	0.29	***0.26***	0.47	0.29	0.71	0.46	0.29	0.78	0.38
M7B	14.08	10.31	9.02	10.82	***7.99***	13.65	9.91	12.01	11.40	8.90
P6	0.50	0.31	***0.29***	0.46	0.31	0.49	0.45	0.34	0.48	0.37

B.1. RMSE

See Table B.11.

B.2. MAPE

See Table B.12.

B.3. MAE

See Table B.13.

Appendix C. Best model frequencies

See Figs. C.4 and C.5.

Data availability

World Semiconductor Trade Statistics (WSTS) data and forecasts are available from wsts.org.

References

Agrawal, A., Khavkin, M., & Slonim, J. (2020). Bringing a real-world edge to forecasting: To improve the accuracy of corporate forecasts, build in the physical parameters from company operations. URL https://www.mckinsey.com/capabilities/strategy-and-corporate-finance/our-insights/bringing-a-real-world-edge-to-forecasting (Accessed 24 January 2024).

Ahmed, N. K., Atiya, A. F., Gayar, N. E., & El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, *29*(5–6), 594–621. http://dx.doi.org/10.1080/07474938.2010.481556, URL http://www.tandfonline.com/doi/abs/10.1080/07474938.2010.481556.

Akay, R. (2022). Caretforecast: Conformal time series forecasting using state of art machine. URL https://CRAN.R-project.org/package=caretForecast.

Alsop, T. (2024a). Semiconductor market revenue worldwide from 1987 to 2024. URL https://www.statista.com/statistics/266973/global-semiconductor-sales-since-1988 (Accessed 19 February 2024).

Alsop, T. (2024b). Semiconductors - statistics & facts. URL https://www.statista.com/topics/1182/semiconductors (Accessed 19 February 2024).

Armstrong, J. S. (2008). Methods to elicit forecasts from groups: Delphi and prediction markets compared. *SSRN Electronic Journal*, http://dx.doi.org/10.2139/ssrn.1153124, URL http://www.ssrn.com/abstract=1153124.

Atanasov, P., Rescober, P., Stone, E., Swift, S. A., Servan-Schreiber, E., Tetlock, P. E., Ungar, L., & Mellers, B. (2015). Distilling the wisdom of crowds: Prediction markets versus prediction polls. *Academy of Management Proceedings*, *2015*(1), 15192. http://dx.doi.org/10.5465/ambpp.2015.15192abstract.

Aubry, M., & Renou-Maissant, P. (2013). Investigating the semiconductor industry cycles. *Applied Economics*, *45*(21), 3058–3067. http://dx.doi.org/10.1080/00036846.2012.697123, URL http://www.tandfonline.com/doi/abs/10.1080/00036846.2012.697123.

Aubry, M., & Renou-Maissant, P. (2014). Semiconductor industry cycles: Explanatory factors and forecasting. *Economic Modelling*, *39*, 221–231. http://dx.doi.org/10.1016/j.econmod.2014.02.039, https://linkinghub.elsevier.com/retrieve/pii/S0264999314000832.

Awad, M., & Khanna, R. (2015). *Efficient learning machines*. Berkeley, CA: A Press, http://dx.doi.org/10.1007/978-1-4302-5990-9, URL http://link.springer.com/10.1007/978-1-4302-5990-9.

Aytac, B., & Wu, S. D. (2013). Characterization of demand for short life-cycle technology products. *Annals of Operations Research*, *203*(1), 255–277. http://dx.doi.org/10.1007/s10479-010-0771-5, URL http://link.springer.com/10.1007/s10479-010-0771-5.

Biau, G., & Scornet, E. (2016). A random forest guided tour. *TEST*, *25*(2), 197–227. http://dx.doi.org/10.1007/s11749-016-0481-7, URL https://link.springer.com/article/10.1007/s11749-016-0481-7.

Box, G. E. P., & COX, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *26*(2), 211–243. http://dx.doi.org/10.1111/j.2517-6161.1964.tb00553.x.

Breiman, L. (1984). *Classification and regression trees*. The Wadsworth & Brooks/Cole.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32.

Brockwell, P. J., & Davis, R. A. (2002). *Introduction to time series and forecasting*. Springer.

Cerqueira, V., Torgo, L., & Mozetič, I. (2020). Evaluating time series forecasting models: an empirical study on performance estimation methods. *Machine Learning*, *109*(11), 1997–2028. http://dx.doi.org/10.1007/s10994-020-05910-7, https://link.springer.com/10.1007/s10994-020-05910-7. arXiv:1905.11744.

Cerqueira, V., Torgo, L., & Soares, C. (2022). A case study comparing machine learning with statistical methods for time series forecasting: size matters. *Journal of Intelligent Information Systems*, *59*(2), 415–433. http://dx.doi.org/10.1007/s10844-022-00713-9, https://link.springer.com/10.1007/s10844-022-00713-9.

Chen, Y.-J., & Chien, C.-F. (2018). An empirical study of demand forecasting of non-volatile memory for smart production of semiconductor manufacturing. *International Journal of Production Research*, *56*(13), 4629–4643. http://dx.doi.org/10.1080/00207543.2017.1421783, https://www.tandfonline.com/doi/full/10.1080/00207543.2017.1421783.

Chien, C.-F., Chen, Y.-J., & Peng, J.-T. (2013). Manufacturing intelligence for semiconductor demand forecast based on technology diffusion and product life cycle. *International Journal of Production Economics*, *128*(2), 496–509. http://dx.doi.org/10.1016/j.ijpe.2010.07.022, URL https://linkinghub.elsevier.com/retrieve/pii/S092552731000263X.

Chien, C.-F., & Lin, K.-Y. (2012). Manufacturing intelligence for Hsinchu Science Park semiconductor sales prediction. *Journal of the Chinese Institute of Industrial Engineers*, *29*(2), 98–110. http://dx.doi.org/10.1080/10170669.2012.660200, http://www.tandfonline.com/doi/abs/10.1080/10170669.2012.660200.

Chow, H. K., & Choy, K. M. (2006). Forecasting the global electronics cycle with leading indicators: A Bayesian VAR approach. *International Journal of Forecasting*, *22*(2), 301–315. http://dx.doi.org/10.1016/j.ijforecast.2005.07.002.

Corder, G., Lafond, E., & Lorenzen, C. (2024). *Worldwide semiconductor market expected to recover strongly and hit $611 billion in 2024*: *Technical Report*, European Semiconductor Industry Association, https://www.eusemiconductors.eu/sites/default/files/ESIA_WSTS_Spring_Forecast2024.pdf (Accessed 01 October 2024).

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. http://dx.doi.org/10.1007/BF00994018, URL http://link.springer.com/10.1007/BF00994018.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, *13*(1), 21–27. http://dx.doi.org/10.1109/TIT.1967.1053964.

Crone, S. F., Hibon, M., & Nikolopoulos, K. (2011). Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. *International Journal of Forecasting*, *27*(3), 635–660.

Linstone, H. A. (Ed.), (1975). *The Delphi method: Techniques and applications*. Reading, Mass.: Addison-Wesley.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A., & Vapnik, V. (1996). Support vector regression machines. In M. Mozer, M. Jordan, & T. Petsche (Eds.), *Vol. 9*, *Advances in neural information processing systems* (pp. 155–161). MIT Press, URL https://proceedings.neurips.cc/paper_files/paper/1996/file/d38901788c533e8286cb6400b40b386d-Paper.pdf.

European Comission (2023). European chips act: Online factpage. URL https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/europe-fit-digital-age/european-chips-act_en (Accessed 19 February 2024).

Fix, E., & Hodges, J. L. (1989). Discriminatory analysis. Nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale de Statistique*, *57*(3), 238. http://dx.doi.org/10.2307/1403797.

Foster, R. N. (1986). Working the S-curve: Assessing technological threats. *Research Management*, *29*(4), 17–20. http://dx.doi.org/10.1080/00345334.1986.11756976.

Gardner, E. S. (1985). Exponential smoothing: The state of the art. *Journal of Forecasting*, *4*(1), 1–28. http://dx.doi.org/10.1002/for.3980040103.

Garza, F., Canseco, M. M., Challú, C., & Olivares, K. G. (2022). StatsForecast: Lightning fast forecasting with statistical and econometric models. URL https://github.com/Nixtla/statsforecast. PyCon Salt Lake City, Utah, US 2022.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42. http://dx.doi.org/10.1007/s10994-006-6226-1.

Gneiting, T., & Katzfuss, M. (2014). Probabilistic forecasting. *Annual Review of Statistics and Its Application*, *1*(1), 125–151. http://dx.doi.org/10.1146/annurev-statistics-062713-085831, URL https://www.annualreviews.org/doi/10.1146/annurev-statistics-062713-085831.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? In *Thirty-sixth conference on neural information processing systems datasets and benchmarks track* (pp. 507–520).

Huang, H., Pouls, M., Meyer, A., & Pauly, M. (2020). Travel time prediction using tree-based ensembles. In *Computational logistics: 11th international conference, ICCL 2020, enschede, the netherlands, September 28–30, 2020, proceedings 11* (pp. 412–427). Springer.

Hyndman, R. J. (2001). It's time to move from "what" to "why". *International Journal of Forecasting*, *17*(4), 567–570.

Hyndman, R. J. (2020). A brief history of forecasting competitions. *International Journal of Forecasting*, *36*(1), 7–14. http://dx.doi.org/10.1016/j.ijforecast.2019.03.015, https://linkinghub.elsevier.com/retrieve/pii/S016920701930086X.

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice* (2nd ed.). Melbourne: OTexts, URL https://otexts.com/fpp2/.

Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software*, *27*(3), 1–22. http://dx.doi.org/10.18637/jss.v027.i03, URL https://www.jstatsoft.org/article/view/v027i03.

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679–688. http://dx.doi.org/10.1016/j.ijforecast.2006.03.001, URL https://linkinghub.elsevier.com/retrieve/pii/S0169207006000239.

Kapur, P., Panwar, S., & Singh, O. (2019). Modeling two-dimensional technology diffusion process under dynamic adoption rate. *J. Model. Manage.*, *14*(3), 717–737. http://dx.doi.org/10.1108/JM2-06-2018-0088, URL https://www.emerald.com/insight/content/doi/10.1108/JM2-06-2018-0088/full/html.

Karatzoglou, A., Smola, A., Hornik, K., & Zeileis, A. (2004). Kernlab - an S4 package for kernel methods in r. *Journal of Statistical Software*, *11*(9), http://dx.doi.org/10.18637/jss.v011.i09.

Kuhn, M. (2008). Building predictive models in r using the caret package. *Journal of Statistical Software*, *28*(5), http://dx.doi.org/10.18637/jss.v028.i05.

Lee, H. L., Padmanabhan, V., & Whang, S. (1997). The bullwhip effect in supply chains. *MIT Sloan Management Review*, *38*(3), 93–102.

Liu, W.-H. (2005). Determinants of the semiconductor industry cycles. *Journal of Policy Modeling*, *27*(7), 853–866. http://dx.doi.org/10.1016/j.jpolmod.2005.05.009, URL https://linkinghub.elsevier.com/retrieve/pii/S0161893805000852.

Liu, W.-H., & Chyi, Y.-L. (2006). A Markov regime-switching model for the semiconductor industry cycles. *Economic Modelling*, *23*(4), 569–578. http://dx.doi.org/10.1016/j.econmod.2006.02.007, URL https://linkinghub.elsevier.com/retrieve/pii/S0264999306000204.

Lv, S., Kim, H., Zheng, B., & Jin, H. (2018). A review of data mining with big data towards its applications in the electronics industry. *Applied Sciences*, *8*(4), 582. http://dx.doi.org/10.3390/app8040582.

Macher, J. T. (2006). Technological development and the boundaries of the firm: A knowledge-based examination in semiconductor manufacturing. *Management Science*, *52*(6), 826–843. http://dx.doi.org/10.1287/mnsc.1060.0511.

Makridakis, S., & Hibon, M. (2000). The M3-competition: results, conclusions and implications. *International Journal of Forecasting*, *16*(4), 451–476. http://dx.doi.org/10.1016/S0169-2070(00)00057-1, URL https://linkinghub.elsevier.com/retrieve/pii/S0169207000000571.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. In A. R. Hernandez Montoya (Ed.), *PLoS One*, *13*(3), Article e0194889. http://dx.doi.org/10.1371/journal.pone.0194889, URL https://dx.plos.org/10.1371/journal.pone.0194889{%}0A https://dspace.lboro.ac.uk/dspace-jspui/handle/2134/25091 https://dx.plos.org/10.1371/journal.pone.0194889.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022a). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, *38*(4), 1346–1364. http://dx.doi.org/10.1016/j.ijforecast.2021.11.013, https://linkinghub.elsevier.com/retrieve/pii/S0169207021001874.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022b). The M5 competition: Background, organization, and implementation. *International Journal of Forecasting*, *38*(4), 1325–1336. http://dx.doi.org/10.1016/j.ijforecast.2021.07.007, https://linkinghub.elsevier.com/retrieve/pii/S0169207021001187.

Modis, T. (1994). Life cycles: forecasting the rise and fall of almost anything. *The Futurist*, *28*(5), 20–25.

Modis, T. (1999). A second lease on life for technological forecasting. *Technological Forecasting and Social Change*, *62*(1–2), 29–32. http://dx.doi.org/10.1016/S0040-1625(99)00036-0.

Nagao, M. (2019). The semiconductor trend from the front-end view. In *2019 pan Pacific microelectronics symposium (pan Pacific)* (pp. 1–6). IEEE, http://dx.doi.org/10.23919/PanPacific.2019.8696919, URL https://ieeexplore.ieee.org/document/8696919/.

Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, *11*, 169–198. http://dx.doi.org/10.1613/jair.614.

Pauly, M., & Kuhlmann, L. (2023). A dynamic systems model for an economic evaluation of sales forecasting methods. *Tehnički Glasnik*, *17*(3), 397–404. http://dx.doi.org/10.31803/tg-20230511175500.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Ben Taieb, S., Bergmeir, C., Bessa, R. J., Bijak, J., Boylan, J. E., Browell, J., Carnevale, C., Castle, J. L., Cirillo, P., Clements, M. P., Cordeiro, C., Cyrino Oliveira, F. L., De Baets, S., Dokumentov, A., ... Ziel, F. (2022). Forecasting: theory and practice. *International Journal of Forecasting*, *38*(3), 705–871. http://dx.doi.org/10.1016/j.ijforecast.2021.11.001, arXiv:2012.03854.

Probst, P., Wright, M. N., & Boulesteix, A. L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *9*(3), Article e1301. http://dx.doi.org/10.1002/widm.1301, URL https://onlinelibrary.wiley.com/doi/full/10.1002/widm.1301?casa_token=hyjhezUnLksAAAAA%3AvXjQPCMqR3ltQIm6WsK5Pk8y_xXwIosdDRnv9aOTF1KDIiFNUU4SH7UR-4hesN3wdjmy90AYMrvU9eQb. arXiv:1804.03515.

Rajan, G. (2024). OpenAI's Altman in talks to raise funds for chips, AI initiative - WSJ. URL https://www.reuters.com/technology/openais-altman-talks-raise-funds-chips-ai-initiative-wsj-2024-02-09/ (Accessed 19 February 2024).

Satchell, S., & Timmermann, A. (1995). On the optimality of adaptive expectations: Muth revisited. *International Journal of Forecasting*, *11*(3), 407–416. http://dx.doi.org/10.1016/0169-2070(95)00588-7.

Schliep, K., & Hechenbichler, K. (2016). Kknn: Weighted k-nearest neighbors. URL https://CRAN.R-project.org/package=kknn R package version 1.3.1.

Simons, C. (2024). *2024 semiconductor industry outlook: Trends and predictions for a cyclical industry*: *Technical Report*, Deloitte, URL https://www2.deloitte.com/us/en/pages/technology-media-and-telecommunications/articles/semiconductor-industry-outlook.html (Accessed 01 October 2024).

Steinmeister, L., Ramosaj, B., Schröter, L., & Pauly, M. (2023). Testing the limits : A robustness analysis of logistic growth models for life cycle estimation during the COVID-19 pandemic. *Vol. 2*, In *Proceedings of the conference on production systems and logistics: CPSL* (pp. 33–44). http://dx.doi.org/10.15488/15265.

Taylor, M. (2023). The US CHIPS and science act of 2022. *MRS Bulletin*, *48*(9), 874–879. http://dx.doi.org/10.1557/s43577-023-00581-w.

Wang, J. (2023). An intuitive tutorial to Gaussian processes regression. *Computing in Science & Engineering*, 1–8. http://dx.doi.org/10.1109/MCSE.2023.3342149.

Wang, C.-H., & Chen, J.-Y. (2019). Demand forecasting and financial estimation considering the interactive dynamics of semiconductor supply-chain companies. *Computers & Industrial Engineering*, *138*(June), Article 106104. http://dx.doi.org/10.1016/j.cie.2019.106104, https://linkinghub.elsevier.com/retrieve/pii/S036083521930573X.

Wang, W., He, N., Chen, M., & Jia, P. (2024). Freight rate index forecasting with prophet model based on multi-dimensional significant events. *Expert Systems with Applications*, *249*, Article 123451. http://dx.doi.org/10.1016/j.eswa.2024.123451, URL https://www.sciencedirect.com/science/article/pii/S0957417424003166.

Wickramasuriya, S. L., Athanasopoulos, G., & 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. http://dx.doi.org/10.1080/01621459.2018.1448825, https://www.tandfonline.com/doi/full/10.1080/01621459.2018.1448825.

Williams, C., & Rasmussen, C. (1995). Gaussian processes for regression. In D. Touretzky, M. Mozer, & M. Hasselmo (Eds.), *Vol. 8*, *Advances in neural information processing systems* (pp. 514–520). MIT Press, URL https://proceedings.neurips.cc/paper_files/paper/1995/file/7cce53cf90577442771720a370c3c723-Paper.pdf.

World Bank (2024). GDP (current US$) - United States, European union, world. URL https://data.worldbank.org/indicator/NY.GDP.MKTP.CD?locations=US-EU-1W (Accessed 19 February 2024).

Wright, M. N., & Ziegler, A. (2017). Ranger : A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, *77*(1), http://dx.doi.org/10.18637/jss.v077.i01.

WSTS. org (2024). More than 35 years authentic market monitoring by WSTS. URL https://www.wsts.org/ (Accessed 24 January 2024).

Wu, J. Z., & Chien, C. F. (2008). Modeling strategic semiconductor assembly outsourcing decisions based on empirical settings. *OR Spectrum*, *30*(3), 401–430. http://dx.doi.org/10.1007/s00291-007-0120-5.

Wu, G., Hu, Y.-C., Chiu, Y.-J., Jiang, P., & Chi, R. (2024). Forecast combination using grey relational analysis and choquet fuzzy integral for container throughput forecasting. *Expert Systems with Applications*, *252*, Article 124170. http://dx.doi.org/10.1016/j.eswa.2024.124170, URL https://www.sciencedirect.com/science/article/pii/S0957417424010364.

Xu, Q., & Sharma, V. (2017). Ensemble sales forecasting study in semiconductor industry. In P. Perner (Ed.), *Advances in data mining. applications and theoretical aspects* (pp. 31–44). Springer, http://dx.doi.org/10.1007/978-3-319-62701-4_3, URL http://link.springer.com/10.1007/978-3-319-62701-4{_}3.

Yu, Y., Hui, C.-L., & Choi, T.-M. (2012). An empirical study of intelligent expert systems on forecasting of fashion color trend. *Expert Systems with Applications*, *39*(4), 4383–4389. http://dx.doi.org/10.1016/j.eswa.2011.09.153, URL https://www.sciencedirect.com/science/article/pii/S095741741101476X.

Louis Steinmeister was awarded his MS degree in applied mathematics from Missouri S&T, where he was a member of the Applied Computational Intelligence Lab. His research interests include machine learning, statistics, and their application to business and financial problems. Currently, he is pursuing a Ph.D. in Statistics through the Graduate School of Logistics at TU Dortmund University, Germany.

Markus Pauly received his diploma degree in mathematics from the Heinrich-Heine University Düsseldorf (2005), where he also completed his Ph.D. (2008) and his habilitation (2013). From 2014 to 2019 he was full professor for Statistics at Ulm University. Since 2019 he is full professor at the Statistics Department at TU Dortmund University, where he investigates a wide range of statistical and machine learning topics.

Article 3

Steinmeister, L., & Pauly, M. (2024). Degenerate Hierarchical Time Series Reconciliation With The Minimum Trace Algorithm in R. In D. Herberger & M. Hübner (Eds.), Proceedings of the Conference on Production Systems and Logistics: CPSL. publish-Ing. doi.org/10.15488/17729

6th Conference on Production Systems and Logistics

Degenerate Hierarchical Time Series Reconciliation With The Minimum Trace Algorithm in R

Louis Steinmeister[1,2,3], Markus Pauly[1,2,4]

[1]*Department of Statistics, TU Dortmund University, Dortmund, Germany*
[2]*Graduate School of Logistics, Dortmund, Germany*
[3]*Infineon Technologies AG, Neubiberg, Germany*
[4]*Research Center Trustworthy Data Science and Security, UA Ruhr, Dortmund, Germany*

Abstract

Many authors have highlighted the importance of reliable forecasts in industry – be it for demand and capacity planning or corporate strategy. Hierarchical time series, where higher level time series are formed by aggregating lower ones, can occur in many fields and industries. The most basic approach ensuring consistency between forecasts on different hierarchy levels is to forecast on the lowest level and to aggregate the results on the higher levels. This method is known as the bottom-up approach. However, this approach ignores information of higher-level time series. Additionally, bottom-level time series may be more volatile, possibly making them more difficult to forecast. To make optimal use of the information of all available time series, [1] propose the Trace Minimization (MinT) algorithm. This algorithm has been implemented in the "hts" [2] and "fable" [3] libraries in R. However, both implementations explicitly assume an equal hierarchical depth. As we illustrate with an example at the end of the paper, this is not always the case in practice.

The objective of this paper is to provide an adapted implementation of the MinT algorithm based on the "hts" library in R which allows the user to handle degenerate hierarchical structures without introducing a bias. This algorithm is applied to our example from the World Semiconductor Trade Statistics (WSTS), a premier provider of semiconductor market data.

Keywords

MinT; Forecasting; Semiconductor; Time Series; Algorithm; Sales; Market Trend; Power Transistors

1. Introduction to hierarchical time series

Whenever time series occur in a hierarchical structure such that time series on lower levels are aggregated to time series on higher ones, we speak of hierarchical time series. A popular textbook example is the Australian domestic tourism dataset where the hierarchies are given by different Australian regions and travel purposes [4]. This kind of structure is common in many domains: different aggregation levels for forecasts are often needed to make decisions in production and supply chain management [5–7] and technologies are often categorized hierarchically [8]. Additional applications are in actuarial science [9] and economics [10–12], to name a few, highlighting importance of hierarchical time series forecasting to practitioners.

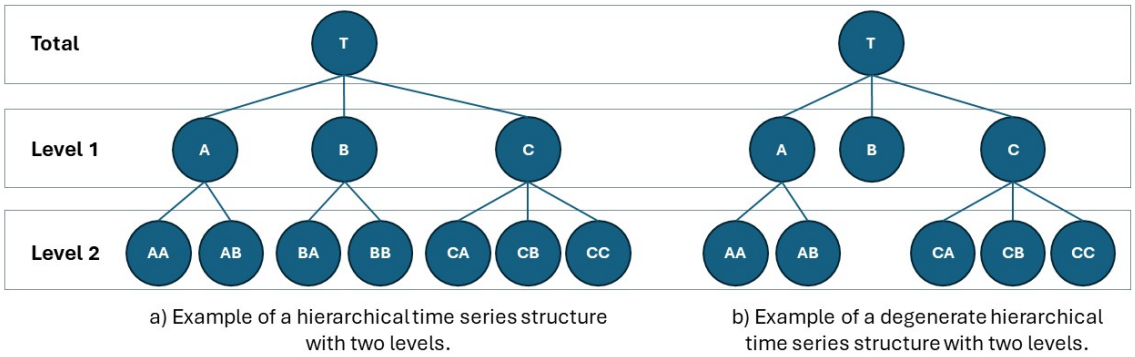
The remainder of this section summarizes the idea and basic notation of hierarchical time series for the reader based on Hyndman and Athanasopoulos' "Forecasting: Principles and Practice" [13].

A time series is generally a sequence of random variables indexed by time [14]:

DOI: https://doi.org/10.15488/17729
ISSN: 2701-6277

$$Y = (Y_t)_{t \in T}, \tag{1}$$

where we assume that T is a discrete set. Hierarchical time series arise, when we observe several time series that are organized hierarchically, i.e. summed up to higher levels. Figure 1 illustrates two example hierarchies. Hierarchy a) consists of 11 time series, where $Y_A = Y_{AA} + Y_{AB}$, etc. Hierarchy b) is very similar but degenerate: time series B on level 1 does not have any corresponding nodes on level 2. Thus, the hierarchical depth differs depending on which path is chosen through the hierarchy.



a) Example of a hierarchical time series structure with two levels.

b) Example of a degenerate hierarchical time series structure with two levels.

Figure 1: Examples of hierarchical structures of time series.

The hierarchical structure of such time series can be expressed in matrix form:

$$\boldsymbol{Y}_t = \boldsymbol{S}\boldsymbol{b}_t, \tag{2}$$

where $\mathbf{b}_t$ refers to the time series on the leaf nodes and $\mathbf{S}$ is an appropriate summation matrix. In the example hierarchy of Figure 1 a), this equation reduces to

$$\begin{bmatrix} Y_{T,t} \\ Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \\ Y_{AA,t} \\ Y_{AB,t} \\ \vdots \\ Y_{CC,t} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ & & & \boldsymbol{I}_7 & & & \end{bmatrix} \begin{bmatrix} Y_{AA,t} \\ \vdots \\ Y_{CC,t} \end{bmatrix}, \tag{3}$$

where $\boldsymbol{I}_7$ is the 7-dimensional identity matrix. A challenge with hierarchical time series arises when forecasting: how to ensure that the produced forecasts are coherent within the hierarchical structure? The naïve approach to this problem is probably the bottom-up approach. Instead of producing forecasts for every time series, only the time series on the leaf nodes $\mathbf{b}_t$ (e.g., Level 2 in Figure 1 a)) are considered. Coherent forecasts on each level can easily be obtained by substituting the bottom-level forecasts into Equation (2). Another simple approach, which we will not cover in more depth, is to disaggregate a top-level forecast. Hyndman et al. showed that such reconciled forecasts can be represented as

$$\widetilde{\boldsymbol{Y}}_t = \boldsymbol{S}\boldsymbol{G}\widehat{\boldsymbol{Y}}_t, \tag{4}$$

where $\widetilde{\mathbf{Y}}_t$ is the vector of reconciled forecasts at time t, $\widehat{\mathbf{Y}}_t$ is the vector of unreconciled forecasts at time t, and $\mathbf{G}$ is an appropriate matrix [15]. In the case of bottom-up forecasts, Equation (4) becomes

$$\tilde{\boldsymbol{Y}}_t = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ & & & I_7 & & & \end{bmatrix} [\mathbf{0} \quad \boldsymbol{I}_7]\hat{\boldsymbol{Y}}_{\boldsymbol{t}}, \tag{5}$$

where $\hat{\boldsymbol{Y}}_{\boldsymbol{t}} = \left[\hat{Y}_{T,t}, \hat{Y}_{A,t}, \cdots, \hat{Y}_{CC,t}\right]^{\boldsymbol{T}}$ is the vector of the individual base forecasts. However, it has been pointed out that neither the bottom-up nor the top-down approaches are optimal as they don't use all available information [16,13]. This is evident, for example, from the **0** part of the **G**-matrix in Equation (5): all the higher-level forecasts are discarded. Furthermore, bottom-level forecasts can be more volatile and therefore more difficult to forecast [17,13].

Outline: To solve this problem, the trace minimization (MinT) algorithm was proposed by [1], which is discussed in the next section. In particular, we further motivate and justify the presented approach to reconciling degenerate hierarchical time series with MinT. Section 3 provides a demo of the software providing the altered MinT implementation based on the "hts" library. Section 4 rounds the paper off by providing a case study with an example use case from the semiconductor industry.

2. The trace minimization algorithm

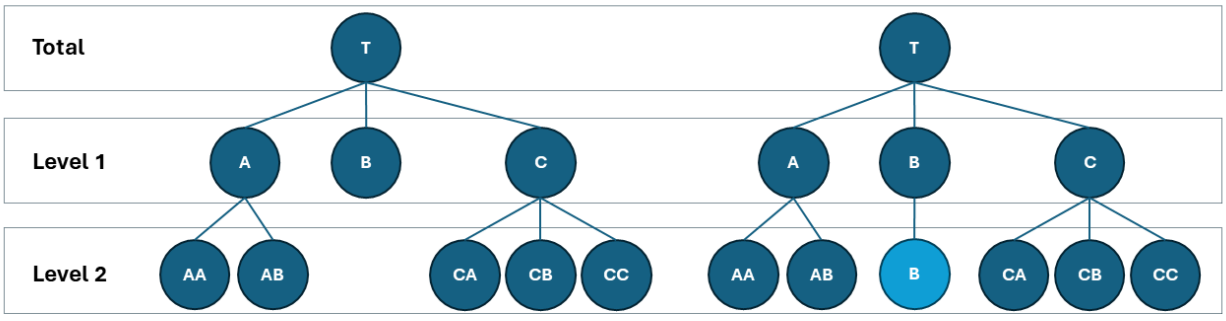
This section provides a summary of the trace minimization algorithm (MinT) based on Wickramasuriya et al.'s original paper [1]. As the name suggests, the objective is to minimize the trace of the covariance matrix $\boldsymbol{W} = Var(\boldsymbol{Y}_t - \tilde{\boldsymbol{Y}}_t)$, in other words the sum of the variances of the residuals of the reconciled forecasts across all time series. Wickramasuriya et al. found that the solution to this minimization problem is given by

$$\boldsymbol{G} = (\boldsymbol{S}^T\boldsymbol{W}^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}^T\boldsymbol{W}^{-1}. \tag{6}$$

For computational efficiency, Equation (4) and (6) can be combined and rewritten to

$$\tilde{\boldsymbol{Y}}_t = [\, \boldsymbol{I_n} - \boldsymbol{W}\boldsymbol{C}^T(\boldsymbol{C}\boldsymbol{W}^{-1}\boldsymbol{C}^T)^{-1}\boldsymbol{C}\,]\hat{\boldsymbol{Y}}_{\boldsymbol{t}}, \tag{7}$$

where $\mathbf{C} = \mathbf{I}_{n_q} - \mathbf{A}$. **A** refers to the aggregation matrix (the part of $\boldsymbol{S}$ without the identity matrix). This allows inverting a potentially much smaller $n_q \times n_q$ matrix instead of the full $n \times n$ matrix [16,1]. Both Equations (6) and (7) depend on the covariance matrix $\boldsymbol{W} = Var(\boldsymbol{Y}_t - \tilde{\boldsymbol{Y}}_t)$. In practice this can be an issue when T is not sufficiently larger than n. Wickramasuriya et al. therefore suggest estimating $\boldsymbol{W}$ with Schäfer and Strimmer's shrinkage estimator [18,1]. MinT with the option to employ this shrinkage estimator is



a) Example of a degenerate hierarchical time series structure with two levels.

b) Example of the same degenerate hierarchical time series structure with two levels and the quick fix.

Figure 2: Degenerate time series hierarchy and the corresponding quick fix.

implemented in the "hts" [2] and "fable" [3] libraries in R. Similar implementations are available for Python with the "pyhts" and "hierarchicalforecast" libraries [19].

The implementations available for R do not allow the reconciliation of forecasts in a degenerate hierarchical structure, such as the hierarchy illustrated in Figure 1 b) and Figure 2 a). The function signature of "hts" requires either the "nodes" or the "characters" argument from which the hierarchical structure is established [2]. Both assume non-degenerate hierarchies. Similarly, this issue was raised on Stackoverflow where Hyndman suggested the equivalent of the quick fix illustrated in Figure 2 for use with fable [20]. Essentially, this approach extends shorter branches of the hierarchical tree with single nodes until the full depth is reached. This is illustrated by the light blue node on level 2 (B) in Figure 2 b).

However, this approach implies a singular covariance matrix $\boldsymbol{W}$, which causes problems in Equations (6) and (7). Additionally, thinking about the minimization of the trace, it results in a higher weight of the duplicated nodes, which skews the reconciliation. In our example, node B would effectively occur twice, giving it twice the weight it would otherwise have. Hyndman pointed out that this approach is not optimal and that it might result in bias, which might be somewhat reduced by employing the shrinkage estimate for $\boldsymbol{W}$ [20]. Furthermore, the Equations (6) and (7) are dependent on the hierarchical structure only through the summation matrix $\boldsymbol{S}$. The constraint that the hierarchy must be non-degenerate is therefore not necessary for the algorithm. This implicit assumption makes interfacing with the libraries more convenient when the hierarchy is non-degenerate, which is the case for some problems like the text book examples given in [4,13], because the user does not need to specify the matrix $\boldsymbol{S}$. However, not all problems meet this constraint as the question on Stackoverlow [20] and our example in Section 4 show. Therefore, this paper suggests an adaptation of the "MinT" function of the "hts" library, which allows the user to directly supply the matrix $\boldsymbol{S}$ instead of supplying either the "nodes" or the "characters" argument to compute reconciled forecasts in the presence of degenerate time series hierarchies.

3. Library demo

The following lines of code will install and load the library containing the adapted implementation of the MinT algorithm. The "devtools" library is assumed to already be installed.

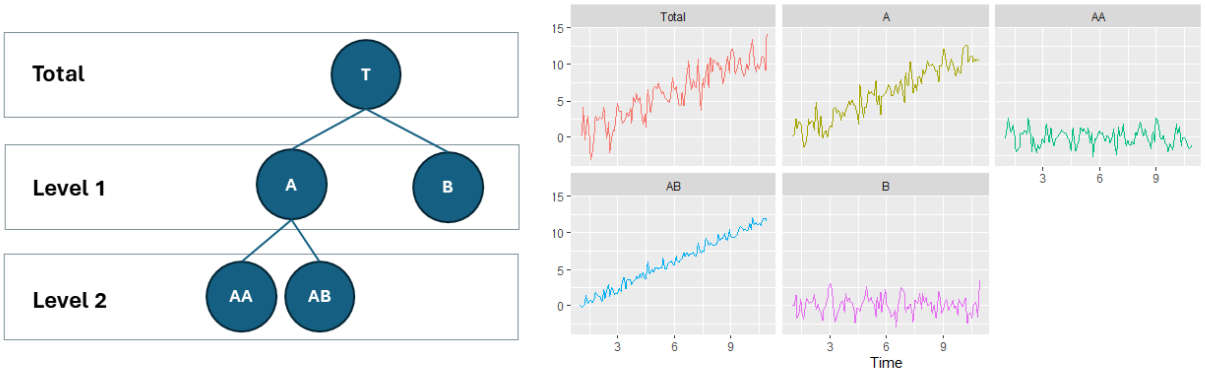
```
devtools::install_github("lsteinmeister/htsDegenerateR")
library(htsDegenerateR)
```

Time series in a small degenerate structure, see Figure 3 a), are considered. The $\boldsymbol{S}$ matrix in this case is given by

$$\boldsymbol{S} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ & \boldsymbol{I}_3 & \end{bmatrix}. \tag{8}$$

The following R code generates the desired time series with 120 monthly observations. The trajectories are plotted with the time in years in Figure 3 b).

```
# Set the seed for reproducibility
set.seed(123)
# Create a sequence of 120 numbers
x <- seq(1, 120)
# Generate the columns
AA <- sin(x*pi/6) + rnorm(120, 0, 1)  # Sine component with random noise
AB <- 0.05*x + rnorm(120, 0, 0.5)   # Linear component
B  <- cos(x*pi/6)+ rnorm(120, 0, 1)   # Cosine component
# Combine the columns into a matrix
matrix <- cbind(AA, AB, B)
hts = ts(matrix, frequency = 12)
# Define S matrix
S <- rbind(c(1,1,1), c(1,1,0), diag(1,3))
rownames(S) <-c("Total", "A", "AA", "AB", "B")
colnames(S) <- c("AA", "AB", "B")
# Aggregate hts on all levels
hts.complete <- ts(t(S %*% t(hts)), frequency = 12)
```



a) Structure of the simulated hierarchical time series.

b) Trajectories of the simulated hierarchical time series.

Figure 3: Structure and trajectories of the simulated degenerate hierarchical time series.

The forecasts and the reconciliation are computed with the following lines.

```
# Fit a model to the time series
hts.models = lapply(hts.complete, function(c.ts) forecast::ets(c.ts))
# Generate predictions based on this model
hts.forecasts = sapply(hts.models, function(mdl) forecast::forecast(mdl, h = 1)$mean)
# Extract residuals
hts.residuals = sapply(hts.models, function(mdl) mdl$residuals)
# Compute reconciled forecasts
MinT(fcasts = hts.forecasts, Smat = S, residual = hts.residuals, algorithms = "chol")
```

The resulting forecasts are rounded to the nearest first decimal and presented in Table 1. Note that the simple base forecasts are not coherent: AA and AB add up to 12.9, not 12.6 (A).

Table 1: Resulting base forecasts and the respective MinT-reconciled forecasts.

	Total	A	AA	AB	B
Base forecasts	13.1	12.6	0.9	12	0.4
MinT forecasts	13.2	12.8	0.8	12	0.4

An application to a real-world data set with an evaluation of the forecast performance follows in the next section.

4. WSTS case study

WSTS is the premier provider of semiconductor market data [21]. In this case study, the WSTS hierarchy of power transistors, which are used to control the flow of electrical power in a circuit, is considered as an industry example of a degenerate hierarchical time series. Monthly data for 8 product categories was obtained from January 1991 to August 2023. An overview of these product categories is provided in Table 2. For historical consistency the C7a and C7b product categorizations were combined into C7 (Field Effect General Purpose Power Transistors). The hierarchy of this categorization is illustrated in Figure 4. Bipolar and other power transistors (CA) in contrast to the other level 1 categories (CB, CC) are not categorized further. Therefore, the hierarchy is degenerate, like the example discussed in Section 2 and illustrated in Figure 2.

Table 2: Overview of the considered product categories and the time periods (in months), they were observed.

	Description	Available From	Available Months
C99	Power Transistors	Jan 1991	392
CB	Insulated Gate Bipolar Transistors	Jan 1995	344
CC	MOSFET Power Transistors	Jan 1991	392
CA	Bipolar and Other Power Transistors	Jan 1991	392
C10	Insulated Gate Bipolar Transistor Modules	Jan 1999	296
C9	Insulated Gate Bipolar Transistors	Jan 1995	344
C8	Field Effect General Purpose Power Transistor Modules	Jan 1999	296
C7	Field Effect General Purpose Power Transistors	Jan 1991	392

To showcase the applicability of MinT to different base forecasts, several statistical and machine learning models were trained on the individual time series from Jan 1991 through Dec 2017. The remaining 68 months were used as a test set. The resulting forecasts are produced by time series cross validation (CV) [22]. The evaluated base forecast models encompass the ARIMA [14], the Error Trend and Seasonality (ETS) model [22], ExtraTrees (ET) [23], Gaussian Process Regression (GPR) [24], Holt-Winters (HW) [25], KNN [26], Random Forest (RF) [27], and Simple Exponential Smoothing (SES) [22]. These models are selected due to their strong performance on the 3-Competition data [28,30,29], which has similar characteristics to the WSTS data considered here. The statistical models are fitted with help of the "forecast" [31] library and the machine learning models are trained with the "caret" [32] and "caretForecast" [33] libraries. Here, the "ranger" library [34] is used to implement RF and ET, and "kernlab" [35] is used for GPR. A grid search is conducted for the machine learning models to tune hyper-parameters.

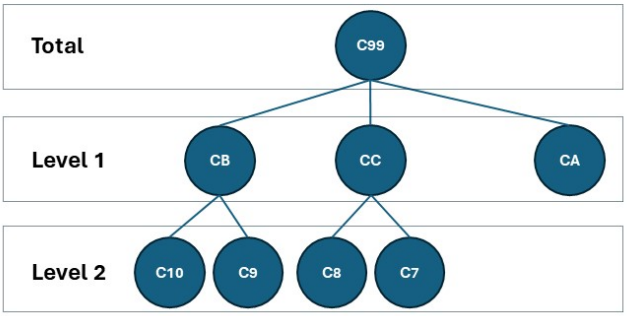


Figure 4: Illustration of the WSTS power transistor hierarchy.

Table 3: Improvement of reconciled forecasts over unreconciled forecasts for the WSTS power transistor hierarchy. The counts and percentages are based on the number of time series and the number of evaluated error metrics: RMSE, MAPE, MAE, MASE. For example, if the forecast accuracy for a single time series improved in terms of all four error measures, it is counted as 4 in this table. Similarly, if the forecast accuracy of one timer series improved in terms of two error measures and another in terms of three error measures, they count as 5. The percentages relate these scores to the total possible score on each hierarchy level.

Model	Total	Level 1	Level 2	Overall
ARIMA	4 (100%)	8 (66.67%)	11 (68.75%)	23 (71.88%)
ETS	4 (100%)	6 (50%)	11 (68.75%)	21 (65.63%)
ET	4 (100%)	5 (41.67%)	12 (75%)	21 (65.63%)
GPR	4 (100%)	11 (91.67%)	16 (100%)	28 (93.75%)
HW	4 (100%)	5 (41.67%)	13 (81.25%)	22 (68.75%)
RF	4 (100%)	7 (58.33%)	16 (100%)	27 (84.38%)
SES	0 (0%)	5 (41.67%)	13 (81.25%)	18 (56.25%)

The resulting forecasts for the eight product categories are reconciled in similar fashion to the demo example in Section 3 for each CV step. Afterwards, the out of sample performance of the base forecasts and the MinT reconciled forecasts is measured in terms of RMSE, MAPE, MAE, and MASE (for details about these error measures, see [36]). The performance measures relative to the best performing model are detailed in Table 5 in the appendix. Overall, the GPR model performed best, followed by HW, ETS, and ARIMA.

Table 3 provides an overview of how often the reconciled forecasts had a higher CV accuracy than the base forecasts according to the four performance measures and averaged across each time series. Generally, MinT not only provides coherent forecast by reconciliation as seen in Table 1 of Section 3 but also has the potential to improve the predictive performance of the forecasts. For example, all top-level forecasts improved by all error measures with the exception of the forecasts based on the SES model. This behaviour is desired and expected (though not guaranteed), since MinT minimizes the sum of the individual time series prediction errors by definition. Table 4 in the appendix confirms this finding and quantifies the improvement. The GPR model improved most from reconciliation and ET improved least when averaged across all time series and all error measures. Generally, forecasts generated by models with a good fit (ARIMA, ETS, GPR, HW) were improved more through reconciliation than those generated by models with a higher average error measure (ET, RF, SES).

5. Discussion

Section 1 gave an overview of the common framing and notation of hierarchical time series. In Section 2, the trace minimization algorithm was introduced, and it was shown that the algorithm does not require non-degenerate hierarchies despite this implicit assumption in popular R implementations. Section 3 provided a demonstration of the altered MinT algorithm, which was originally implemented in the "hts" library [2]. It walks prospective users through the required steps to download and install the "htsDegenerateR" library and use it for the reconciliation of forecasts in a degenerate hierarchical structure. The steps were illustrated with a reproducible simulated example, and it was shown that the algorithm successfully reconciled the example forecasts. Section 4 provided an industry case study. The WSTS power transistor product categorization is an example of a degenerate hierarchical time series structure. The "htsDegenerateR" library was successfully used to reconcile forecasts using several statistical and machine learning algorithms. Furthermore, it was shown that the forecasting performance for a majority of the time series increased through this reconciliation approach. Thus, a working solution to a real-world problem has been provided.

Acknowledgements

This paper was supported by a research cooperation between Infineon Technologies AG and TU Dortmund University through the Graduate School of Logistics.

Appendix

Table 4: Percentage improvement of forecasting accuracy through reconciliation for the different base forecast methods ARIMA, Error Trend and Seasonality (ETS) model, ExtraTrees (ET) , Gaussian Process Regression (GPR), Holt-Winters (HW), KNN, Random Forest (RF) and Simple Exponential Smoothing (SES) by error measure. Positive values indicate an improvement. Larger values are preferred.

	ARIMA										
	Total	**Level 1**				**Level 2**					**All**
	C99	**CB**	**CC**	**CA**	**Overall**	**C10**	**C9**	**C8**	**C7**	**Overall**	
MAE	4.4%	7.8%	2.1%	-0.1%	3.8%	8.7%	6.3%	0.8%	-1.2%	2.7%	3.6%
MAPE	3.2%	8.0%	0.8%	-0.8%	2.7%	8.7%	7.0%	0.7%	-0.6%	3.9%	3.4%
MASE	4.4%	7.8%	2.1%	-0.1%	3.3%	8.7%	6.3%	0.8%	-1.2%	3.9%	3.7%
RMSE	4.6%	6.1%	2.7%	-4.5%	3.0%	8.6%	4.5%	-0.4%	-1.7%	1.8%	3.0%
	ETS										
	Total	**Level 1**				**Level 2**					**All**
	C99	**CB**	**CC**	**CA**	**Overall**	**C10**	**C9**	**C8**	**C7**	**Overall**	
MAE	8.1%	-0.8%	-0.4%	4.3%	0.1%	-0.5%	3.9%	6.4%	-0.1%	0.8%	2.8%
MAPE	8.0%	-1.5%	-0.4%	3.4%	0.7%	1.4%	5.3%	9.5%	0.5%	5.2%	4.0%
MASE	8.1%	-0.8%	-0.4%	4.3%	1.1%	-0.5%	3.9%	6.4%	-0.1%	2.3%	2.6%
RMSE	4.9%	0.4%	0.3%	4.2%	0.8%	0.9%	3.9%	2.1%	-0.5%	0.6%	1.9%
	ET										
	Total	**Level 1**				**Level 2**					**All**
	C99	**CB**	**CC**	**CA**	**Overall**	**C10**	**C9**	**C8**	**C7**	**Overall**	
MAE	0.5%	-0.7%	1.4%	-7.9%	-0.1%	0.0%	1.7%	-0.6%	2.5%	1.5%	0.6%
MAPE	2.0%	-0.2%	1.5%	-7.1%	-1.5%	-0.1%	0.6%	-2.8%	2.3%	-0.2%	-0.4%
MASE	0.5%	-0.7%	1.4%	-7.9%	-1.7%	0.0%	1.7%	-0.6%	2.5%	0.9%	0.0%
RMSE	0.8%	0.1%	1.0%	-8.4%	-0.1%	1.0%	1.7%	0.2%	2.0%	1.6%	0.8%
	GPR										
	Total	**Level 1**				**Level 2**					**All**
	C99	**CB**	**CC**	**CA**	**Overall**	**C10**	**C9**	**C8**	**C7**	**Overall**	
MAE	5.0%	1.9%	0.8%	3.1%	1.5%	1.7%	1.3%	1.2%	1.1%	1.3%	2.5%
MAPE	5.0%	-0.2%	1.1%	2.6%	1.3%	2.0%	1.9%	2.0%	1.3%	1.8%	2.0%
MASE	5.0%	1.9%	0.8%	3.1%	2.0%	1.7%	1.3%	1.2%	1.1%	1.3%	2.0%
RMSE	5.1%	1.2%	1.7%	3.4%	1.7%	0.3%	1.1%	-0.1%	1.5%	1.0%	2.5%
	HW										
	Total	**Level 1**				**Level 2**					**All**
	C99	**CB**	**CC**	**CA**	**Overall**	**C10**	**C9**	**C8**	**C7**	**Overall**	
MAE	5.3%	-1.1%	0.2%	-0.2%	-0.3%	2.1%	6.2%	4.4%	-0.3%	1.6%	2.0%
MAPE	4.8%	-2.0%	-0.6%	-1.0%	-1.2%	3.2%	7.5%	6.0%	-0.8%	4.6%	2.7%
MASE	5.3%	-1.1%	0.2%	-0.2%	-0.4%	2.1%	6.2%	4.4%	-0.3%	3.1%	2.1%
RMSE	4.0%	2.4%	2.4%	1.9%	2.3%	3.1%	3.6%	2.0%	0.9%	1.9%	2.7%
	RF										
	Total	**Level 1**				**Level 2**					**All**
	C99	**CB**	**CC**	**CA**	**Overall**	**C10**	**C9**	**C8**	**C7**	**Overall**	
MAE	2.0%	0.1%	2.1%	-6.0%	0.7%	2.9%	3.8%	1.9%	2.6%	2.8%	1.8%
MAPE	2.4%	0.6%	1.7%	-4.9%	-0.5%	3.0%	3.2%	1.6%	2.5%	2.5%	1.5%
MASE	2.0%	0.1%	2.1%	-6.0%	-0.6%	2.9%	3.8%	1.9%	2.6%	2.8%	1.5%
RMSE	1.6%	-0.1%	1.2%	-7.7%	0.0%	2.7%	3.7%	1.3%	1.6%	2.2%	1.2%

	SES										
	Total	Level 1				Level 2					All
	C99	CB	CC	CA	Overall	C10	C9	C8	C7	Overall	
MAE	-0.8%	-0.1%	0.8%	-0.7%	0.4%	1.3%	-0.6%	0.7%	0.7%	0.7%	0.1%
MAPE	-0.9%	-0.1%	1.1%	-0.7%	0.1%	1.6%	-0.6%	1.5%	0.9%	0.9%	0.5%
MASE	-0.8%	-0.1%	0.8%	-0.7%	0.1%	1.3%	-0.6%	0.7%	0.7%	0.6%	0.2%
RMSE	-0.4%	0.0%	0.3%	-0.7%	0.1%	1.2%	1.2%	0.1%	0.2%	0.5%	0.1%

Table 5: Error measures relative to the best performing model by hierarchy level among forecast methods ARIMA, Error Trend and Seasonality (ETS) model, ExtraTrees (ET), Gaussian Process Regression (GPR), Holt-Winters (HW), KNN, Random Forest (RF) and Simple Exponential Smoothing (SES). The values indicate the percentage of the error measure by which the considered model was worse than the best performing model, e.g., A value of 7% indicates that the given model was 7% worse than the best model in that category and by that error measure.

		Total	Level 1	Level 2	All			Total	Level 1	Level 2	All
MAE	ARIMA	5%	9%	11%	9%	MASE	ARIMA	5%	5%	10%	8%
	ETS	7%	3%	5%	5%		ETS	7%	3%	7%	5%
	ET	63%	54%	58%	58%		ET	63%	38%	56%	50%
	GPR	0%	0%	0%	0%		GPR	0%	0%	0%	0%
	HW	3%	1%	4%	3%		HW	3%	2%	6%	4%
	RF	57%	47%	54%	52%		RF	57%	32%	50%	44%
	SES	74%	59%	59%	64%		SES	74%	48%	51%	52%
MAPE	ARIMA	1%	4%	7%	5%	RMSE	ARIMA	6%	10%	11%	9%
	ETS	6%	2%	8%	5%		ETS	3%	4%	5%	4%
	ET	61%	35%	46%	43%		ET	52%	45%	49%	48%
	GPR	0%	0%	0%	0%		GPR	1%	0%	0%	0%
	HW	2%	1%	7%	5%		HW	0%	4%	5%	3%
	RF	55%	29%	41%	39%		RF	47%	39%	44%	43%
	SES	74%	49%	51%	53%		SES	66%	50%	49%	54%

References

[1] Wickramasuriya, S.L., Athanasopoulos, G., 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.

[2] Rob Hyndman, Alan Lee, Earo Wang, Shanika Wickramasuriya, 2021. hts: Hierarchical and Grouped Time Series. https://CRAN.R-project.org/package=hts.

[3] Mitchell O'Hara-Wild, Rob Hyndman, Earo Wang, 2023. fable: Forecasting Models for Tidy Time Series. https://CRAN.R-project.org/package=fable.

[4] Athanasopoulos, G., Ahmed, R.A., Hyndman, R.J., 2009. Hierarchical forecasts for Australian domestic tourism. International Journal of Forecasting 25 (1), 146–166.

[5] Abolghasemi, M., Hyndman, R.J., Tarr, G., Bergmeir, C., 2019. Machine learning applications in time series hierarchical forecasting. http://arxiv.org/pdf/1912.00370.pdf.

[6] Mancuso, P., Piccialli, V., Sudoso, A.M., 2021. A machine learning approach for forecasting hierarchical time series. Expert Systems with Applications 182, 115102.

[7] Moon, S., Hicks, C., Simpson, A., 2012. The development of a hierarchical forecasting method for predicting spare parts demand in the South Korean Navy—A case study. International Journal of Production Economics 140 (2), 794–802.

[8] Steinmeister, L., Ramosaj, B., Schröter, L., Pauly, M., 2023. Testing The Limits: A Robustness Analysis Of Logistic Growth Models For Life Cycle Estimation During The COVID-19 Pandemic. Hannover : publish-Ing, 33 pp.

[9] Li, H., Li, H., Lu, Y., Panagiotelis, A., 2019. A forecast reconciliation approach to cause-of-death mortality modeling. Insurance: Mathematics and Economics 86, 122–133.
[10] Athanasopoulos, G., Gamakumara, P., Panagiotelis, A., Hyndman, R.J., Affan, M., 2020. Hierarchical Forecasting, in: Glaeser, Fuleky, P. (Eds.), Macroeconomic Forecasting in the Era of Big Data, vol. 52. Springer International Publishing, Cham, pp. 689–719.
[11] Capistrán, C., Constandse, C., Ramos-Francia, M., 2010. Multi-horizon inflation forecasts using disaggregated data. Economic Modelling 27 (3), 666–677.
[12] Lila, M.F., Meira, E., Cyrino Oliveira, F.L., 2022. Forecasting unemployment in Brazil: A robust reconciliation approach using hierarchical data. Socio-Economic Planning Sciences 82, 101298.
[13] Hyndman, R.J., Athanasopoulos, G., 2018. Forecasting: Principles and practice, 2nd edition ed. OTexts, Melbourne.
[14] Brockwell, P.J., Davis, R.A., 2002. Introduction to time series and forecasting. Springer.
[15] Hyndman, R.J., Ahmed, R.A., Athanasopoulos, G., Shang, H.L., 2011. Optimal combination forecasts for hierarchical time series. Computational Statistics & Data Analysis 55 (9), 2579–2589.
[16] Athanasopoulos, G., Hyndman, R.J., Kourentzes, N., Panagiotelis, A., 2023. Forecast reconciliation: A review. International Journal of Forecasting.
[17] Grunfeld, Y., Griliches, Z., 1960. Is Aggregation Necessarily Bad? The Review of Economics and Statistics 42 (1), 1–13.
[18] Schäfer, J., Strimmer, K., 2005. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. Statistical applications in genetics and molecular biology 4, Article32.
[19] Olivares, K.G., Garza, F., Luo, D., Challú, C., Mergenthaler, M., Taieb, S.B., Wickramasuriya, S.L., Dubrawski, A., 2022. HierarchicalForecast: A Reference Framework for Hierarchical Forecasting in Python. http://arxiv.org/pdf/2207.03517.pdf.
[20] Hyndman, R.J., 2020. Can fable reconcile hierarchical time series, where the hierarchy has different depth at different nodes? https://stackoverflow.com/questions/64012331/can-fable-reconcile-hierarchical-time-series-where-the-hierarchy-has-different. Accessed 27 February 2024.
[21] World Semiconductor Trade Statistics. More Than 35 Years Authentic Market Monitoring by WSTS. World Semiconductor Trade Statistics. https://www.wsts.org/. Accessed 24 January 2024.
[22] Hyndman, R.J., Athanasopoulos, G., 2018. Forecasting: Principles and Practice, 2nd edition ed. OTexts, Melbourne.
[23] Geurts, P., Ernst, D., Wehenkel, L., 2006. Extremely randomized trees. Machine Learning 63 (1), 3–42.
[24] Williams, C., Rasmussen, C., 1995. Gaussian Processes for Regression, in: Advances in Neural Information Processing Systems. MIT Press.
[25] Winters, P.R., 1960. Forecasting Sales by Exponentially Weighted Moving Averages. Management Science 6 (3), 324–342.
[26] Cover, T., Hart, P., 1967. Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13 (1), 21–27.
[27] Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.
[28] Ahmed, N.K., Atiya, A.F., Gayar, N.E., El-Shishiny, H., 2010. An Empirical Comparison of Machine Learning Models for Time Series Forecasting. Econometric Reviews 29 (5-6), 594–621.
[29] Makridakis, S., Spiliotis, E., Assimakopoulos, V., 2018. Statistical and Machine Learning forecasting methods: Concerns and ways forward. PLoS ONE 13 (3), e0194889.
[30] Makridakis, S., Hibon, M., 2000. The M3-Competition: results, conclusions and implications. International Journal of Forecasting 16 (4), 451–476.
[31] Hyndman, R.J., Khandakar, Y., 2008. Automatic Time Series Forecasting: The forecast Package for R. Journal of Statistical Software 27 (3), 1–22.
[32] Kuhn, M., 2008. Building Predictive Models in R Using the caret Package. Journal of Statistical Software 28 (5).
[33] Akay, R., 2022. caretForecast: Conformal Time Series Forecasting Using State of Art Machine.
[34] Wright, M.N., Ziegler, A., 2017. ranger : A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. Journal of Statistical Software 77 (1).
[35] Karatzoglou, A., Smola, A., Hornik, K., Zeileis, A., 2004. kernlab - An S4 Package for Kernel Methods in R. Journal of Statistical Software 11 (9).

[36] Hyndman, R.J., Koehler, A.B., 2006. Another look at measures of forecast accuracy. International Journal of Forecasting 22 (4), 679–688.

Biography

Louis Steinmeister was awarded his MS degree in applied mathematics from Missouri S&T, where he was a member of the Applied Computational Intelligence Lab. His research interests include machine learning, statistics, and their application to business and financial problems. Currently, he is pursuing a Ph.D. in Statistics through the Graduate School of Logistics at TU Dortmund University, Germany.

Markus Pauly received his diploma degree in mathematics from the Heinrich-Heine University Düsseldorf (2005), where he also completed his Ph.D (2008) and his habilitation (2013). From 2014 to 2019 he was full professor for Statistics at Ulm University. Since 2019 he is full professor at the Statistics Department at TU Dortmund University, where he investigates a wide range of statistical and machine learning topics.

Article 4

Steinmeister, L., Pauly, M., 2024, Iterative Trace Minimization for the Reconciliation of Very Short Hierarchical Time Series. doi.org/10.48550/arXiv.2409.18550

Iterative Trace Minimization for the Reconciliation of Very Short Hierarchical Time Series

Louis Steinmeister[a,b,*], Markus Pauly[a,b,c]

[a]Department of Statistics, TU Dortmund University, Vogelpothsweg 78, Dortmund, 44227, Germany
[b]Graduate School of Logistics, Leonhard-Euler-Straße 5, Dortmund, 44227, Germany
[c]Research Center Trustworthy Data Science and Security, Joseph-von-Fraunhofer-Straße 25, Dortmund, 44227, Germany

Abstract

Time series often appear in an additive hierarchical structure. In such cases, higher-level time series are the sums of their subordinate time series. This hierarchical structure places a natural constraint on forecasts. However, univariate forecasting techniques are incapable of ensuring this forecast coherence. An obvious solution is to forecast only bottom time series and obtain higher level forecasts through aggregation. This approach is also known as the bottom-up approach. In their seminal paper, [1] propose an optimal reconciliation approach named MinT. It tries to minimize the trace of the underlying covariance matrix of all forecast errors. The MinT algorithm has demonstrated superior performance to the bottom-up and other approaches and enjoys great popularity. This paper provides a simulation study examining the performance of MinT for very short time series and larger hierarchical structures. This scenario makes the covariance estimation required by MinT difficult. A novel iterative approach is introduced which significantly reduces the number of estimated parameters. This approach is capable of improving forecast accuracy further. The application of MinTit is also demonstrated with a case study at the hand of a semiconductor dataset based on data provided by the World Semiconductor Trade Statistics (WSTS), a premier provider of semiconductor market data.

Keywords:
Prediction, Market Trend, Coherence, Forecasting, Machine Learning, Statistical Learning

1. Executive Summary

The main **objective** of this paper is to introduce iterative versions (MinTit) of the Trace Minimization (MinT) algorithm, which was introduced by Wickramasuriya et al. [1]. The goal of the iterative algorithms is to simplify covariance estimation for large hierarchical structures, especially in the case of time series with very few observations.

The **motivation** stems from the difficulty of estimating large residual covariance matrices with very few observations. In the case study presented in Section 5, a total of 110 dimensions implies the estimation of 6160 parameters with only 24 observations in the most extreme case. Athanasopoulos et al. [2] point out that this challenge of dimensionality can lead to "volatile performance of MinT, especially for short time series".

Methods: A simulation study resembling Wickramasuriya et al. [1] for extremely short time series is conducted to assess the performance of MinT in this case and determine whether an iterative approach can improve upon the performance of MinT and existing alternatives such as structural scaling. Here, performance is measured by a reduction in root mean squared error (RMSE) compared to the base forecasts. For the latter we considered three common forecast models: Gaussian process regression (GPR), Error, Trend, and Seasonality (ETS) as well as auto.ARIMA models. The latter automatically seeks to find an optimal parameter combination including potential seasonal components as described in Hyndman and Khandakar [3]. In addition, a case study is presented which showcases the application of various reconciliation methods on a real-world dataset.

Results: In many of the simulated scenarios, MinTit, the iterative adaptations of MinT, achieved a higher reduction in average RMSE than any other reconciliation method. Nevertheless, MinT proved remarkably robust – even in the presence of extremely short time series. It was generally the best performing reconciliation method for forecasts based on GPR. However, the forecasts based on the ETS and ARIMA models were much more accurate in most of the simulated cases. For these two base forecasts, the new MinTit approaches were often more favorable.

Conclusion: MinTit reconciliation has the potential to improve forecast accuracy for extremely short time series and when time series lengths differ across the hierarchical structure. Ideally, the reconciliation method should be chosen based on a cross validation when the length of the considered time series permits, as performance may differ depending on the structure of the data and the employed forecasting algorithm.

As a rule of thumb, MinT may be the reconciliation method of choice when there is reason to assume strong seasonal effects: the ETS and auto.ARIMA models (both seek an optimal model, including models incorporating seasonal components) struggled to adequately identify seasonality with short time series, which appeared to have negatively affected the performance of the iterative reconciliation methods. Otherwise, MinTit may be a good choice with the potential to achieve a lower RMSE. MinTit

*Corresponding author: Louis Steinmeister
Email address: louis.steinmeister@tu-dortmund.de (Louis Steinmeister)

also allows for efficient use of information when time series lengths differ, which can provide an additional edge.

2. Brief Introduction

Accurate forecasts play an increasingly important role in a more and more interconnected and digitized world. In a corporate setting, forecasts routinely inform investment decisions, capital allocation, production, strategy, and help companies anticipate technological change [4, 5, 6, 7, 8]. In many settings in practice, time series occur in a hierarchical context. A textbook example is the Australian tourism data set [9]. Here, quarterly tourism demand is broken down into three levels: by purpose of travel, states, and nights spent in the regional capital or other cities. Kuhlmann et al. [10] provide a corporate example of the occurrence of hierarchical time series where sales numbers are broken down by product and sales channel. Furthermore, the case study in Section 5 consists of time series in a hierarchical structure with market revenues aggregated by semiconductor product categories. Forecasts at different levels of detail are crucial for operational planning. However, ensuring these forecasts and plans are coherent is essential for a unified operational strategy.

The paper is structured as follows: The above brief introduction motivates the discussion of hierarchical time series reconciliation methods, including the popular trace minimization algorithm Wickramasuriya et al. [1]. The latter is explained in Section 3, which also introduces new iterative versions of trace minimization (Section 3.1). Similar to Wickramasuriya et al. [1], a simulation study to compare the performance of various reconciliation algorithms is conducted in Section 4. There, we explore the effect of correlation, the impact of the smoothing effect of aggregation, the effects of seasonality, different time series lengths, or hierarchical degeneracy, as well as the effect of a 'larger' hierarchy. Section 5 presents a case study of a semiconductor market data set: The World Semiconductor Trade Statistics (WSTS). Finally, Section 6 provides a discussion of the overall findings.

3. Hierarchical Time Series Reconciliation

A hierarchical time series is defined as a set of (bottom) time series which can be summed to higher level time series. An illustration of a simple such hierarchy is provided in Figure 1. In this example, the bottom time series encompass

$$X^{(bottom)} = \left(X^{(AA)}, X^{(AB)}, X^{(BA)}, X^{(BB)}, X^{(BC)}\right)^{\intercal}.$$

These are progressively summed according to the overall hierarchical structure:

$$\begin{aligned} X^{(A)} &= X^{(AA)} + X^{(AB)} \\ X^{(B)} &= X^{(BA)} + X^{(BB)} + X^{(BC)} \\ X^{(T)} &= X^{(A)} + X^{(B)} = X^{(AA)} + X^{(AB)} + X^{(BA)} + X^{(BB)} + X^{(BC)}. \end{aligned}$$

Note that this set of equations can be summarized in a simple matrix multiplication: $X = \mathbf{S}X^{(bottom)}$, where $X = (X^{(T)}, X^{(A)}, X^{(B)}, \ldots, X^{(BC)})^{\intercal}$ are the pooled time series from all levels in hierarchical order and $\mathbf{S}$ is an appropriate structure matrix. In the example of Figure 1, clearly

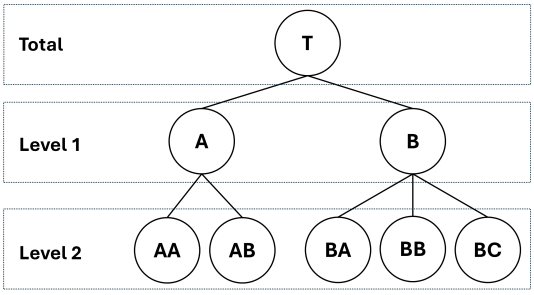


Figure 1: Illustration of a simple time series hierarchy with two levels.

$$\mathbf{S} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The goal of hierarchical time series reconciliation is to ensure the coherence of base forecasts, which are obtained through appropriate forecasting models like ARIMA, ETS, or other forecasting approaches. This means that forecasts should be constrained by the underlying hierarchical structure, i.e. the appropriate bottom time series forecasts should add up to the corresponding higher level forecasts.

Probably the most intuitive method to achieve this goal is called the bottom-up approach (BU). Here, forecasts are only generated on the bottom time series. Higher level forecasts are derived by summing the respective bottom forecasts. However, this approach has a significant drawback: information on higher aggregation levels is omitted. To make matters worse, higher aggregation levels are observed to often feature a preferable signal-to-noise-ratio compared to lower levels, potentially making the forecasting process more difficult and less accurate.

The opposite approach is called the top-down approach (TD). As the name suggests, it consists of disaggregating a single, top-level forecast to arrive at lower level forecasts. However, this implies that all information of lower level time series is omitted in the process. Furthermore, Hyndman et al. [11] and Panagiotelis et al. [12] showed that a top-down approach necessarily introduces bias.

Hyndman et al. [11] showed that any of the existing methods can be expressed as

$$\tilde{X} = \mathbf{S}\mathbf{G}\hat{X}, \tag{1}$$

for an appropriate matrix $\mathbf{G}$, where $\tilde{X}$ are the reconciled forecasts based on the base forecasts $\hat{X}$. For instance, the bottom-up approach for the example hierarchy from Figure 1 can be expressed with

$$\mathbf{G} = [\mathbf{0}_{5\times 3}, \mathbf{I}_5].$$

Ideally, all available forecasts should be utilized in a way that optimally combines the information on all hierarchical levels. This motivated the introduction of the trace minimization algorithm (MinT) by Wickramasuriya et al. [1]. Denote by $\mathbf{V}$ the covariance matrix of all forecast errors (from all levels). As the name suggests, MinT aims to minimize the trace of $\mathbf{V}$. In other words, MinT is a solution of the minimization problem that minimizes the total sum of forecast errors under the constraint that the forecasts have to obey the hierarchical structure. Applying Equation (1), this minimization problem can be written as

$$\min_{\mathbf{G}} \{\mathbf{SGWG}^\intercal\mathbf{S}^\intercal\} \tag{2}$$

under the constraint that the reconciled forecasts are unbiased or, as Hyndman et al. [11] showed, under the condition that $\mathbf{SGS} = \mathbf{S}$. Here, $\mathbf{W}$ is defined as the covariance matrix of the base forecast errors. Wickramasuriya et al. [1] further show that the solution to the minimization problem in Equation (2) has the closed form solution

$$\mathbf{G} = \left(\mathbf{S}^\intercal\mathbf{W}^{-1}\mathbf{S}\right)^{-1}\mathbf{S}^\intercal\mathbf{W}^{-1}. \tag{3}$$

This algorithm is accessible through the `hts`[13] and `fable`[14] packages in R or the `pyhts` and `hierarchicalforecast` libraries in Python [15]. Steinmeister and Pauly [16] provide an adapted version of the `hts` implementation of MinT, allowing for degenerate hierarchical structures.

Note that the solution in Equation (3) requires the inversion of the covariance matrix of base forecast errors $\mathbf{W}$. With a total of p time series, the estimation of $\mathbf{W}$ thus requires the estimation of $\frac{p(p+1)}{2}$ variables. The inversion of this matrix can cause further issues, particularly when $\mathbf{W}$ is badly conditioned. This may amplify estimation errors and lead to an unstable solution. Therefore, in the case of hierarchies with a large number of time series and a relatively small number of observations, Wickramasuriya et al. [1] propose the use of a shrinkage estimator introduced by Schäfer and Strimmer [17]. This shrinkage estimator finds a combination of the standard covariance estimator and a diagonal matrix containing only the base forecast variances. The result is a covariance estimator where the off-diagonal entries are shrunk towards zero. While this approach significantly stabilizes the algorithms, we propose an iterative approach in the following section which additionally reduces the number of estimated parameters significantly. This approach is compared against common reconciliation methods in an extensive simulation study in Section 4.

3.1. Iterative Trace Minimization

In the spirit of divide and conquer, we propose breaking a large hierarchy down into smaller sub-hierarchies to mitigate the stability issues with MinT. An illustration of the steps that the algorithm passes through in each iteration for the example hierarchy in Figure 1 is provided in Figure 2. The iterative process views each 1-level sub-hierarchy as an optimization problem in itself. The intuition is that this allows for the estimation of several small covariance matrices instead of one large one. Iterating this process allows information from the higher levels to be propagated to lower levels and vice versa, similar to back-propagation in neural networks [18].

The total number of parameters is greatly reduced by this method. For illustration, assume a hierarchy with D levels and $w > 1$ sub-nodes each. Then the total number of nodes, or time series, is given by

$$p = \sum_{k=0}^{D} w^k = \frac{w^{D+1} - 1}{w - 1}. \tag{4}$$

This implies that $\frac{w^{D+1}-1}{w-1}\frac{w^{D+1}-w-2}{w-1}\frac{1}{2} = O\left(w^{2D}\right)$ parameters have to be estimated to obtain the covariance matrix. Thus, the complexity is polynomial of order $2D$ in terms of the hierarchy's width and of exponential order in terms of the hierarchy's depth.

In contrast, the iterative approach goes through $\frac{w^D-1}{w-1}$ steps, each requiring the estimation of $w(w+1)/2$ parameters. Thus, yielding a polynomial order of $D+1$ degrees in terms of the hierarchy's width. The order of complexity for the depth remains exponential, but with a (significantly) smaller base. Overall, the complexity is given by $O\left(w^{D+1}\right)$. Table 1 illustrates the impact that this difference in complexity can have in practice. Particularly when hierarchies are large, it can be seen that the iterative algorithm, as detailed in Algorithm 1, has a much smaller complexity in terms of internal parameters that need to be estimated. However, this advantage usually does not translate into faster runtime. The iterative approach requires several iterations, usually ranging from high double digits to several hundreds. Furthermore, more iterations may be required when the hierarchy is large, as it may take longer for the information to propagate.

Table 1: Comparison of the number of parameters required to estimate the covariance matrix for simple hierarchies each with constant widths of 2 and 3 and depths ranging from 2 to 6.

Width	2			3		
Depth	MinT	Iterative	Fraction	MinT	Iterative	Fraction
2	28	9	0.32	91	24	0.26
3	120	21	0.18	820	78	0.10
4	496	45	0.09	7381	240	0.03
5	2016	93	0.05	66430	726	0.01
6	8128	189	0.02	597871	2184	0.00

4. Monte Carlo Simulation

To evaluate the performance of the iterative algorithm and compare it against established methods, we conduct Monte Carlo experiments with 5000 iterations in Section 4.3, unless stated otherwise.

The compared reconciliation methods are summarized in Section 4.1 and an overview over the utilized forecasting methods is given in Section 4.2. Results are presented in root mean squared error (RMSE) relative to the baseline forecasts.

4.1. Reconciliation Methods

The following reconciliation methods are compared: the bottom-up approach (BU), trace minimization (MinT) and two iterative versions: MinTit$_l$ and MinTit$_g$. They are all briefly explained below:

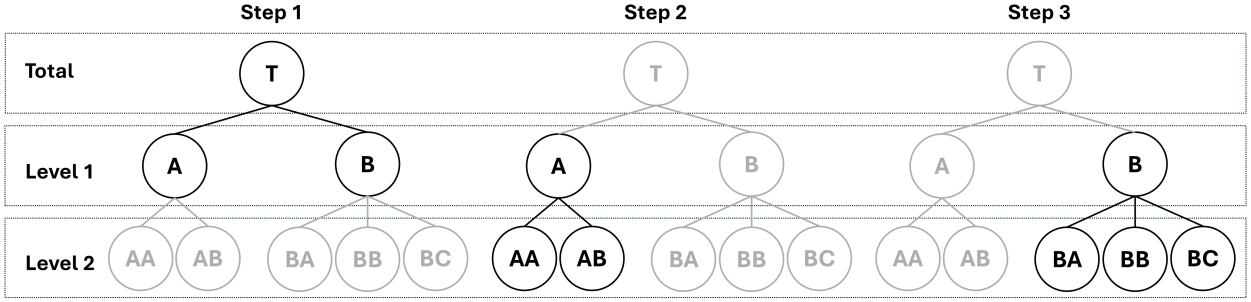


Figure 2: The steps with their respective sub-hierarchies in each iteration.

Algorithm 1 Iterative Trace Minimization

Input: A hierarchical structure with levels $L = \{L_0, ..., L_N\}$ each containing the time series $TS_{L_i} = \{TS^0_{L_i}, ..., TS^{K_{L_i}}_{L_i}\}$ with respective constituents $C(TS^{k_{L_i}}_{L_i})$. Forecasts are denoted $\widehat{TS}$ and $\widehat{C(TS)}$ and residuals are denoted $R(TS)$ and $R(C(TS))$. Define $\epsilon > 0$ as the convergence threshold.
for $k \in \{1, ..., maxit\}$ **do**
  Save vector of all forecasts in $forecasts_{old}$
  for $l \in L$ **do**
    for $TS \in l$ **do**
      $\mathbf{S} \leftarrow$ structure matrix containing TS and $C(TS)$
      $\mathbf{\Sigma} \leftarrow$ covariance matrix* of $[TS, C(TS)]$
      $\left[\widehat{TS}, \widehat{C(TS)}\right] \leftarrow \text{MinT}\left(\left[\widehat{TS}, \widehat{C(TS)}\right], \mathbf{\Sigma}, \mathbf{S}\right)$
    end for
  end for
  Save vector of all updated forecasts in $forecasts_{new}$
  if $\|forecasts_{new} - forecasts_{old}\| < \epsilon$ **then**
    return $forecasts_{new}$
  end if
end for

* The covariance matrix can either be locally estimated (using only the sub-hierarchy) or globally by using the global shrinkage estimator proposed by Schäfer and Strimmer [17] and subsetting the global matrix appropriately. The former will be denoted as MinTit_l and the latter as MinTit_g

The **bottom-up** (BU) approach requires forecasts only for the bottom time series. These forecasts are then aggregated according to the specified hierarchical structure. This is the simplest approach in the comparison – but nevertheless commonly used as it presents a canonical way of ensuring coherence.

Trace minimization (MinT) functions by minimizing the sum of expected forecasting errors. A closed form solution is given by Equation (3). This approach has been shown to excel in many situations [1].

Iterative trace minimizations (MinTit) is introduced in Section 3 and iteratively reconciles sub-hierarchies until forecasts converge, as illustrated in Figure 2 and defined in Algorithm 1. It was inspired by the idea that it may not be necessary to specify the complete covariance matrix as in MinT, potentially leading to higher performance in settings with large hierarchical structures and very short time series or time series of varying lengths. Two options for estimation of the covariance matrix in the sub-hierarchy are compared: either the shrinkage estimator [17] can be computed globally and subdivided according to the respective sub-hierarchies (MinTit_g), or the shrinkage estimator can be computed locally, leading to locally differing shrinkage factors (MinTit_l). The latter has the advantage that more data can be incorporated when available on the sub-hierarchy-level.

Additionally, two weighted least squares methods are compared, which differ from MinT mainly in assumptions regarding the form of the covariance matrix $\mathbf{W}$:

Structural Scaling (WLS_s) as in Athanasopoulos et al. [19] assumes equal and uncorrelated residual variances of the reconciled bottom time series. This results in higher variances on higher levels in the hierarchy, proportional to the number of subsumed bottom time series. This assumption can be expressed as

$$\mathbf{W} = k\mathbf{\Lambda},$$

where $\mathbf{\Lambda} = diag(\mathbf{S1})$ is a diagonal matrix with the row sums of the structure matrix $\mathbf{S}$ on the diagonal.

Variance Scaling (WLS_v) as in Athanasopoulos et al. [19] assumes that reconciled errors of all time series (also including higher hierarchical levels) are uncorrelated. Thus, it uses

$$\mathbf{W} = diag\left(\hat{\mathbf{W}}\right),$$

where $\hat{\mathbf{W}}$ is the sample covariance matrix of the observed forecast errors.

4.2. Forecasting Methods

Base forecasts are generated with three common models: Error Trend and Seasonality (ETS), ARIMA, and Gaussian Process Regression (GPR). ETS and ARIMA are standard time series models, which are, e.g., explained in more detail in Hyndman and Athanasopoulos [20].

This study uses the R implementations of the models available in the `forecast` package [21]. For ARIMA, we utilize the `auto.arima` function, which searches for an optimal model, including models with seasonality. The exact process is explained in Hyndman and Khandakar [3]. In all evaluations and result discussions, we use the abbreviation ARIMA for ease of presentation.

GPR is explained in more detail in Wang [22]. It was one of the strongest performing forecasting models on studies with the M3-Competition dataset Ahmed et al. [23], Makridakis et al.

[24] and performed very well on a previous study involving the WSTS dataset [25]. Here, the GPR implementation in the `kernlab` R package [26] is used.

4.3. Experiments

The design of our numeric experiments is inspired by Wickramasuriya et al. [1], where similar Monte Carlo simulations were run to assess the performance of the MinT algorithm. In contrast to Wickramasuriya et al. [1], which examined time series of length $T = 60$, 180, and 300 observations, the simulation study in this paper focuses the challenging situation of shorter time series with $T = 15$, 30, and 60 observations with the final $h = 4$, 4, and 8 observations withheld respectively for out-of-sample evaluation.

To give detailed recommendations for such shorter time series, we study several important scenarios. In particular, we investigate

1. the impact of correlated time series (Section 4.3.1) in a setup similar to Section 3.2 of Wickramasuriya et al. [1],
2. the effect of smoothing, which is supposed to replicate higher signal-to-noise-ratios often observed in practice (Section 4.3.2) based on the experimental setup in Section 3.3 of Wickramasuriya et al. [1],
3. reconciliation in the presence of seasonality (Section 4.3.3), similar to Section 3.4 in Wickramasuriya et al. [1],
4. the effect of differing time series lengths (Section 4.3.4),
5. the effect of degeneracy on the choice of the reconciliation algorithm (Section 4.3.5), and
6. reconciliation for a large hierarchy (Section 4.3.6) in a setup similar to Section 3.5 of Wickramasuriya et al. [1].

The concrete scenario settings and the resulting performances of all approaches are discussed below.

4.3.1. Impact of Correlation

Wickramasuriya et al. [1] considered a two level hierarchy with 7 total time series. Because the iterative version proposed in Section 3.1 is intended for larger hierarchies with short histories, a slightly larger hierarchy of level 3 with 15 total time series, which is illustrated in Figure 3, is considered.

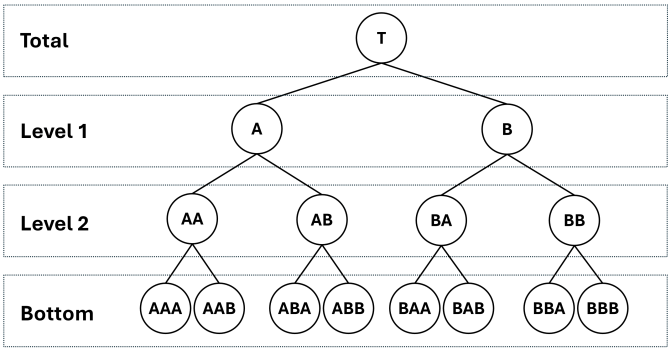


Figure 3: Hierarchical structure for the simulations in Section 4.3.1.

The bottom time series follow a similar data generating process as in Wickramasuriya et al. [1]: Each time series on the bottom level were generated from an ARIMA(p, d, q) process, where p and q were uniformly drawn from $\{0, 1, 2\}$ and d from $\{0, 1\}$. For each time series, the ARIMA coefficients were drawn from different uniform distributions as detailed in Table 2. The given boundaries are the same as in Wickramasuriya et al. [1] to ensure better comparability.

Table 2: Overview of the sampling ranges of the ARIMA parameters, analogous to Wickramasuriya et al. [1].

AR coefficients		
p	coefficient	sampling range
1	ϕ_1	$[0.5, 0.7]$
2	ϕ_1	$[\phi_2 - 0.9, 0.9 - \phi_2]$
	ϕ_2	$[0.5, 0.7]$
MA coefficients		
q	coefficient	sampling range
1	θ_1	$[0.5, 0.7]$
2	θ_1	$[-\frac{0.9+\theta_2}{3.2}, \frac{0.9+\theta_2}{3.2}]$
	θ_2	$[0.5, 0.7]$

Moreover, the bottom time series were generated with contemporaneous errors following a predefined covariance matrix to introduce correlation of time series throughout the hierarchy. However, since the larger hierarchy calls for more bottom time series, the covariance matrix used here is a larger version of the one used in Wickramasuriya et al. [1] and given by $\mathbf{\Sigma}$ as defined in Equation (5).

$$\mathbf{\Sigma} = \begin{bmatrix} 5 & 3 & 2 & 1 & 1 & 1 & 1 & 1 \\ 3 & 4 & 2 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 5 & 3 & 2 & 1 & 1 & 1 \\ 1 & 1 & 3 & 4 & 3 & 2 & 1 & 1 \\ 1 & 1 & 2 & 3 & 5 & 3 & 2 & 1 \\ 1 & 1 & 1 & 2 & 3 & 4 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 & 2 & 5 & 3 \\ 1 & 1 & 1 & 1 & 1 & 1 & 3 & 4 \end{bmatrix} \tag{5}$$

Similarly to the simulation in Section 3.2 of Wickramasuriya et al. [1], this structure creates more strongly correlated errors for time series under the same parent node in the hierarchical structure and generally more moderately correlated errors for time series of more distant parent nodes. The thus generated bottom time series were then summed according to the defined hierarchical structure as illustrated in Section 4.3.1.

Results for ETS forecasts. Table 3 summarizes the simulation results for the ETS-generated forecasts. The table is divided into three column groups containing the results for time series of total lengths $T = 15$, $T = 30$, and $T = 60$, respectively. Each of them contains columns of relative RMSE changes for forecasting horizons of $h = 1$, $h = 1 : 2$ (the average over the 1-step and 2-step ahead forecasts), and $h = 1 : 4$ in the first two cases ($T = 15$ and $T = 30$) . In the case of $T = 60$, the forecast horizons are $h = 1$, $h = 1 : 4$, and $h = 1 : 8$. The last column provides row-wise averages.

The rows contain groups relating to the different hierarchical levels. Each of these groups provides details of the relative RMSE of the compared reconciliation method. Negative values

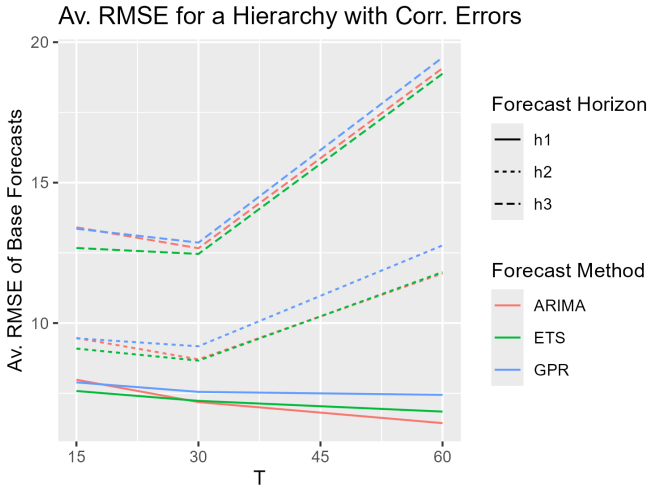


Figure 4: Average RMSE for the various base forecasts in Section 4.3.1 (ARIMA in red, ETS in green, and GPR in blue) by total length of the time series. A forecast horizon of $h1$ corresponds to $h = 1$, $h2$ to $h = 1:2$ in the case of $T = 15, 30$ and $h = 1:4$ for $T = 60$. Similarly, $h3$ corresponds to the longer forecasting horizons of $h = 1:4$ and $h = 1:8$, respectively.

indicate a %-improvement over the unreconciled base forecasts and positive values indicate a %-increase in RMSE. Hence, smaller values indicate better forecast performance.

The introduced contemporaneous covariance structure gives rise to lower SNRs on higher hierarchical levels. This explains how the simple BU approach on average achieves relatively large error reductions.

However, on average, the MinTit reconciliation methods lead to the highest reduction in RMSE for forecasts generated by ETS. This largely stems from its superior performance on higher hierarchical levels, with substantial improvements over the forecasts reconciled with MinT or the WLS approaches. However, the performance of MinT and MinTit was much closer on the bottom-level: both MinT and MinTit$_g$ achieving a 0.8% improvement for the ETS forecasts.

The *results for ARIMA forecasts* (see Table 4) are similar to those for ETS. MinTit performed best on average – leading to average improvements in RMSE of 10% (compared to 9.1% for MinT). As was observed with the ETS forecasts, this strong performance largely stemmed from the larger RMSE reductions on higher hierarchical levels (up to 11.9%). On the bottom level, MinT outperformed MinTit, achieving a 5.3% improvement against 5.1%.

Results for GPR forecasts. MinT and MinTit result in similar improvements for the GPR-generated forecasts (Table 5) with MinT leading to slightly more accurate reconciled forecasts. However, the base forecast performance of GPR was worse than for ETS and ARIMA, see Figure 4. Reconciliation resulted in a maximum average error reduction of 7.3% using the MinT reconciliation method as compared to a 10% reduction in RMSE for ARIMA forecasts using MinTit$_g$ reconciliation. Thus, these GPR forecasts are less accurate, putting the comparative advantage of MinT against MinTit into perspective.

Table 3: Out-of-sample forecast performance for the hierarchy with contemporaneously correlated time series in Section 4.3.1 for ETS-generated forecasts relative to the ETS-generated base forecasts.

	ETS									
	T = 15			T = 30			T = 60			
	h=1	1:2	1:4	h=1	1:2	1:4	h=1	1:4	1:8	Av.
	Top-Level									
MinT	-6.8	-5.3	-3.5	-9.1	-8.5	-7.6	-8.9	-6.2	-4	-6.7
WLS$_s$	-7.4	**-6.7**	**-6.5**	-8.8	-9.3	**-9.7**	-7.3	-7.2	**-6.6**	-7.7
WLS$_v$	-7.4	**-6.7**	**-6.5**	-8.8	-9.3	**-9.7**	-7.3	-7.2	**-6.6**	-7.7
BU	-7.3	-5.6	-4.8	**-10.5**	-9.6	-8.3	-9.5	-6.4	-3.6	-7.3
MinTit$_g$	**-7.7**	-6.6	-5.8	-10.1	**-10**	**-9.7**	-9.3	**-7.7**	-6.2	**-8.1**
MinTit$_l$	**-7.7**	-6.5	-5.6	-10	-9.8	-9.3	**-9.6**	-7.6	-6	-8.0
	Level 1									
MinT	-6.7	-5.6	-3.8	-8.5	-7.8	-7	-8	-6.1	-4.2	-6.4
WLS$_s$	-6.9	**-6.5**	**-6.2**	-7.5	-8	-8.3	-6.1	-6.5	**-6.2**	-6.9
WLS$_v$	-6.8	-6.4	-6.1	-7.5	-8	-8.2	-6.1	-6.5	-6.1	-6.9
BU	-6.9	-5.4	-4.5	**-9.5**	-8.3	-7	-8.4	-5.9	-3.3	-6.6
MinTit$_g$	**-7.1**	-6.2	-5.2	-9.1	**-8.8**	**-8.4**	-8.3	**-7.2**	-5.9	**-7.4**
MinTit$_l$	**-7.1**	-6	-5	-9	-8.5	-8	**-8.5**	-7.1	-5.7	-7.2
	Level 2									
MinT	**-4.3**	-3.6	-2.4	-5.2	-4.9	-4.2	-4.6	-3.1	-2.4	-3.9
WLS$_s$	-4	**-3.8**	**-3.9**	-4	-4.6	-4.9	-2.8	-3.3	**-3.7**	-3.9
WLS$_v$	-3.7	-3.6	-3.6	-3.8	-4.5	-4.8	-2.6	-3.2	-3.6	-3.7
BU	-4	-2.8	-2.2	**-5.8**	-4.9	-3.8	-4.8	-2.7	-1.1	-3.6
MinTit$_g$	-4.1	-3.5	-2.9	-5.3	**-5.3**	**-5**	-4.7	**-3.9**	-3.5	**-4.2**
MinTit$_l$	-4.1	-3.3	-2.6	-5.2	-4.9	-4.5	**-4.9**	-3.6	-3.2	-4.0
	Bottom-Level									
MinT	**-0.8**	**-1.3**	-0.8	**-0.1**	-0.6	-1.1	**-0.5**	-0.8	-1.6	**-0.8**
WLS$_s$	0	-0.7	**-1.1**	1.4	0.2	-0.8	1.5	-0.4	-1.7	-0.2
WLS$_v$	0.8	-0.1	-0.5	2.1	0.6	-0.5	2.2	-0.2	-1.5	0.3
BU	0	0	0	0	0	0	0	0	0	0.0
MinTit$_g$	-0.2	-0.9	-0.9	0.1	**-0.7**	**-1.4**	-0.2	**-1.2**	**-2.2**	**-0.8**
MinTit$_l$	-0.4	-0.8	-0.8	0.2	-0.4	-1.1	-0.4	-1	-2	-0.7
	Average									
MinT	-5.4	-4.5	-2.9	-7	-6.6	-5.8	-6.8	-4.8	-3.3	-5.2
WLS$_s$	-5.6	**-5.2**	**-5**	-6.2	-6.7	-7.1	-5	-5.2	**-5.1**	-5.7
WLS$_v$	-5.3	-5	-4.9	-6.1	-6.7	-7	-4.8	-5.2	**-5.1**	-5.6
BU	-5.5	-4.1	-3.4	**-8**	-7	-5.8	-7.1	-4.6	-2.4	-5.3
MinTit$_g$	-5.7	-5	-4.2	-7.6	**-7.5**	**-7.2**	-7	**-5.9**	-4.9	**-6.1**
MinTit$_l$	**-5.8**	-4.9	-4	-7.5	-7.2	-6.8	**-7.2**	-5.7	-4.7	-6.0

4.3.2. *Impact of the Smoothing Effect of Aggregation*

As Wickramasuriya et al. [1] note, aggregated time series display a higher signal-to-noise-ratio (SNR) compared to their component time series. Borrowing the same design, which was based on the work of van Erven and Cugliari [27], this section studies the performance of the different reconciliation methods when the SNR decreases with the hierarchical level on the 3-level hierarchy used in Section 4.3.1 and illustrated in Figure 3. Wickramasuriya et al. [1] used the following design to generate a total of four bottom time series for their 2-level hierarchy:

$$y_{AA,t} = z_{AA,t} - \eta_t - 0.5\omega_t,$$
$$y_{AB,t} = z_{AB,t} + \eta_t - 0.5\omega_t,$$
$$y_{BA,t} = z_{BA,t} - \eta_t + 0.5\omega_t,$$
$$y_{BB,t} = z_{BB,t} + \eta_t + 0.5\omega_t,$$

where z_{AA}, z_{Ab}, z_{bA} and z_{BB} are simulated as independent ARIMA processes, otherwise as specified in Section 4.3.1, with standard normal distributed error terms and additional errors, which cancel out hierarchically, η_t $N(0, 10)$ and ω_t $N(0, 6)$. Note that the representation in terms of added errors (η and ω) can equally be achieved through the representation with a single, appropriately

Table 4: Out-of-sample forecast performance for the hierarchy with contemporaneously correlated time series in Section 4.3.1 for ARIMA-generated forecasts relative to the ARIMA-generated base forecasts.

	ARIMA									
	T = 15			T = 30			T = 60			
	h=1	1:2	1:4	h=1	1:2	1:4	h=1	1:4	1:8	Av.
	Top-Level									
MinT	-16.2	-13.7	-10.1	-11.6	-10.8	-9.6	-8.6	-7.2	-6.7	-10.5
WLS_s	-13.9	-12.5	-10.8	-11.2	-10.6	-10.5	-8.9	-8.2	-8.5	-10.6
WLS_v	-13.9	-12.5	-10.8	-11.2	-10.6	-10.5	-8.9	-8.2	-8.5	-10.6
BU	-14.6	-12.1	-9.4	-8.3	-8.5	-9.3	-9.8	-8.5	-7.7	-9.8
$MinTit_g$	-16.4	**-13.9**	**-10.9**	**-13.5**	**-12.5**	**-11.6**	**-10.7**	-9.2	**-8.8**	**-11.9**
$MinTit_l$	**-16.5**	**-13.9**	-10.7	**-13.5**	**-12.5**	-11.5	-10.6	**-9.3**	-8.7	**-11.9**
	Level 1									
MinT	**-15.3**	**-12.7**	-9.2	-13	-12.3	-9.6	-7.8	-6.6	-5.7	-10.2
WLS_s	-12.3	-11	-9.4	-10.9	-10.7	-9.5	-7.3	-6.8	-6.8	-9.4
WLS_v	-12.3	-11	-9.3	-11	-10.7	-9.5	-7.4	-6.9	-6.8	-9.4
BU	-12.8	-10.4	-7.8	-7.3	-7.9	-7.8	-7.9	-7	-5.7	-8.3
$MinTit_g$	-15.1	-12.5	**-9.5**	**-14.4**	**-13.5**	**-11**	**-9.4**	**-8**	**-7**	**-11.2**
$MinTit_l$	-15.1	-12.4	-9.3	**-14.4**	**-13.5**	-10.8	**-9.4**	**-8**	-6.8	-11.1
	Level 2									
MinT	**-11.2**	**-9**	**-6.3**	-11.4	-9.8	-7.3	-5.5	-4.4	-3.7	-7.6
WLS_s	-7.4	-6.5	-5.9	-7.1	-6.5	-6	-4.3	-4.1	-4.1	-5.8
WLS_v	-7.3	-6.4	-5.7	-7.8	-6.8	-6.2	-4.6	-4.3	-4.2	-5.9
BU	-7.8	-5.9	-4.4	-2.7	-2.8	-3.9	-4.3	-4.2	-3.3	-4.4
$MinTit_g$	-10.3	-8.3	-6.2	-12	**-10.2**	**-7.9**	**-6.6**	**-5.3**	**-4.5**	**-7.9**
$MinTit_l$	-10.2	-8.1	-5.9	**-12.1**	-10.1	-7.6	-6.5	-5.2	-4.1	-7.8
	Bottom-Level									
MinT	**-4.7**	**-3.9**	**-2.3**	-13.9	-11	-5.9	-3.5	-1.3	-1.1	**-5.3**
WLS_s	0.3	-0.4	-1	-3.1	-2.5	-1.5	0	0.1	-0.5	-1.0
WLS_v	0.2	-0.3	-0.5	-8.8	-6.7	-3.6	-2.2	-0.7	-0.9	-2.6
BU	0	0	0	0	0	0	0	0	0	0.0
$MinTit_g$	-3.3	-2.7	-1.6	-14.1	-10.9	**-6**	**-4.4**	**-1.8**	**-1.4**	-5.1
$MinTit_l$	-3.3	-2.6	-1.5	**-14.4**	**-11.1**	-5.9	-4.2	-1.7	-1.2	-5.1
	Average									
MinT	**-13.5**	**-11.1**	-7.9	-12.3	-11.1	-8.6	-7.2	-5.7	-4.9	-9.1
WLS_s	-10.3	-9.2	-7.9	-9.1	-8.6	-7.9	-6.5	-5.9	-5.9	-7.9
WLS_v	-10.3	-9.2	-7.8	-10.1	-9.3	-8.3	-6.9	-6	-6	-8.2
BU	-10.9	-8.7	-6.6	-5.7	-5.9	-6.3	-7	-6	-5.1	-6.9
$MinTit_g$	-13.1	-10.9	**-8.2**	-13.5	**-12.1**	**-9.8**	**-8.8**	**-7.1**	**-6.3**	**-10.0**
$MinTit_l$	-13.2	-10.8	-8	**-13.6**	**-12.1**	-9.7	-8.7	**-7.1**	-6.1	-9.9

Table 5: Out-of-sample forecast performance for the hierarchy with contemporaneously correlated time series in Section 4.3.1 for GPR-generated forecasts relative to the GPR-generated base forecasts.

	GPR									
	T = 15			T = 30			T = 60			
	h=1	1:2	1:4	h=1	1:2	1:4	h=1	1:4	1:8	Av.
	Top-Level									
MinT	-13.6	**-12.7**	-9.4	-10.1	**-9.2**	-8	**-7.9**	**-6.3**	-5.7	-9.2
WLS_s	-10.9	-10	-9.3	-7.7	-6.9	-6.9	-5.3	-4.6	-4.9	-7.4
WLS_v	-10.9	-10	-9.3	-7.7	-6.9	-6.9	-5.3	-4.6	-4.9	-7.4
BU	**-13.7**	-12.1	-10.1	-8.9	-7.1	-6.7	-5.9	-4	-4.3	-8.1
$MinTit_g$	-13.4	-12.4	**-10.5**	-10	-8.8	-8.1	-7.5	-5.8	-5.9	-9.2
$MinTit_l$	-13.5	-12.4	-10.2	**-10.2**	-9	**-8.3**	-7.5	-6.2	**-6.4**	**-9.3**
	Level 1									
MinT	**-12.3**	**-11**	-8.9	**-10.3**	**-8.8**	**-6.8**	**-9.1**	**-5.8**	-4.5	**-8.6**
WLS_s	-8.9	-7.9	-8.5	-7.1	-6	-5.3	-5.7	-3.8	-3.5	-6.3
WLS_v	-8.8	-7.8	-8.5	-7.1	-5.9	-5.2	-5.8	-3.8	-3.4	-6.3
BU	-11.9	-10	-9	-9	-6.7	-5.4	-7.2	-3.7	-3.2	-7.3
$MinTit_g$	-11.5	-10.1	**-9.4**	-9.8	-8.1	-6.6	-8.4	-5.2	-4.3	-8.2
$MinTit_l$	-11.5	-10.1	-9	-9.8	-8.2	-6.7	-8.4	-5.4	**-4.7**	-8.2
	Level 2									
MinT	**-8.2**	**-7.1**	**-4.9**	**-7.4**	**-6.2**	**-4.4**	**-7**	**-4.1**	**-2.7**	**-5.8**
WLS_s	-4.5	-3.9	-4.2	-4	-3.2	-2.6	-3.6	-1.9	-1.6	-3.3
WLS_v	-4.2	-3.6	-4	-3.8	-3.1	-2.6	-3.5	-1.9	-1.6	-3.1
BU	-7.3	-5.8	-4.6	-5.9	-4.1	-2.9	-5.2	-2.1	-1.5	-4.4
$MinTit_g$	-6.8	-5.8	-4.8	-6.4	-5.1	-3.7	-6.1	-3.3	-2.4	-4.9
$MinTit_l$	-6.8	-5.8	-4.4	-6.4	-5.1	-3.6	-6	-3.4	**-2.7**	-4.9
	Bottom-Level									
MinT	**-1.9**	**-2.1**	**-1**	**-2.2**	**-2.7**	**-2**	**-2.4**	**-2.4**	**-1.6**	**-2.0**
WLS_s	2.2	1.4	0.3	1.5	0.6	0.2	1.2	0.1	-0.1	0.8
WLS_v	3	2	0.7	1.9	0.8	0.2	1.4	-0.1	-0.2	1.1
BU	0	0	0	0	0	0	0	0	0	0.0
$MinTit_g$	0.2	-0.3	-0.4	-0.7	-1.1	-0.9	-1.2	-1.4	-1.1	-0.8
$MinTit_l$	0.1	-0.4	-0.3	-0.8	-1.2	-1	-1.1	-1.6	-1.3	-0.8
	Average									
MinT	**-10.7**	**-9.6**	-7.1	**-8.6**	**-7.6**	**-6**	**-7.3**	**-5.1**	-4	**-7.3**
WLS_s	-7.4	-6.6	-6.6	-5.6	-4.8	-4.5	-4.2	-3.1	-3	-5.1
WLS_v	-7.2	-6.5	-6.5	-5.5	-4.8	-4.5	-4.1	-3.1	-3	-5.0
BU	-10.2	-8.6	-7.2	-7.2	-5.4	-4.5	-5.3	-2.9	-2.7	-6.0
$MinTit_g$	-9.8	-8.8	**-7.5**	-8	-6.7	-5.7	-6.6	-4.4	-3.9	-6.8
$MinTit_l$	-9.9	-8.8	-7.2	-8.1	-6.9	-5.8	-6.6	-4.7	**-4.3**	-6.9

correlated error term. In the above case, this can be simplified to

$$\begin{bmatrix} y_{AA,t} \\ y_{AB,t} \\ y_{BA,t} \\ y_{BB,t} \end{bmatrix} = \begin{bmatrix} z_{AA,t} \\ z_{AB,t} \\ z_{BA,t} \\ z_{BB,t} \end{bmatrix} + \zeta_t,$$

where ζ_t are multivariate normal errors with zero mean and covariance

$$\begin{bmatrix} 11.5 & -8.5 & 8.5 & -11.5 \\ -8.5 & 11.5 & -11.5 & 8.5 \\ 8.5 & -11.5 & 11.5 & -8.5 \\ -11.5 & 8.5 & -8.5 & 11.5 \end{bmatrix}.$$

This results in an additional error variance of 6 on level 1 and 11.5 on level 2.

The time series for the 3-level time series are simulated analogously with multivariate normal errors with covariance

$$\begin{bmatrix} 11.75 & -8.25 & 8.75 & -11.25 & 11.25 & -8.75 & 8.25 & -11.75 \\ -8.25 & 11.75 & -11.25 & 8.75 & -8.75 & 11.25 & -11.75 & 8.25 \\ 8.75 & -11.25 & 11.75 & -8.25 & 8.25 & -11.75 & 11.25 & -8.75 \\ -11.25 & 8.75 & -8.25 & 11.75 & -11.75 & 8.25 & -8.75 & 11.25 \\ 11.25 & -8.75 & 8.25 & -11.75 & 11.75 & -8.25 & 8.75 & -11.25 \\ -8.75 & 11.25 & -11.75 & 8.25 & -8.25 & 11.75 & -11.25 & 8.75 \\ 8.25 & -11.75 & 11.25 & -8.75 & 8.75 & -11.25 & 11.75 & -8.25 \\ -11.75 & 8.25 & -8.75 & 11.25 & -11.25 & 8.75 & -8.25 & 11.75 \end{bmatrix}.$$

resulting in added error variances of 4, 7, and 11.75 on levels 1 to 3, respectively. For the time series of the totals, these errors cancel out.

Similarly to Section 4.3.1, the results are presented in three tables for the three respective base forecast methods: Table 6 contains the changes in RMSE for the ETS generated forecasts, Table 7 those for the ARIMA generated forecasts, and Table 8 those for the GPR generated forecasts.

Results for Arima and ETS forecasts. In the case of the ETS and ARIMA forecasts, the MinTit methods and MinT achieved a comparable average decrease in RMSE leading the other methods. MinTit performed better for time series lengths of $T = 15$ and $T = 30$ observations and lower hierarchical levels while MinT showed stronger results for time series of length $T = 60$ and for higher levels of aggregation. The ARIMA and ETS forecasts were comparable in RMSE with ETS having a slight edge, especially for the very short time series ($T = 15$), as shown in Figure 5.

Similarly to Section 4.3.1, MinT excelled with forecasts generated by GRP. However, these in turn had a much higher base RMSE than ETS and ARIMA – particularly for $T = 30$ and $T = 60$, as can be seen in Figure 5.

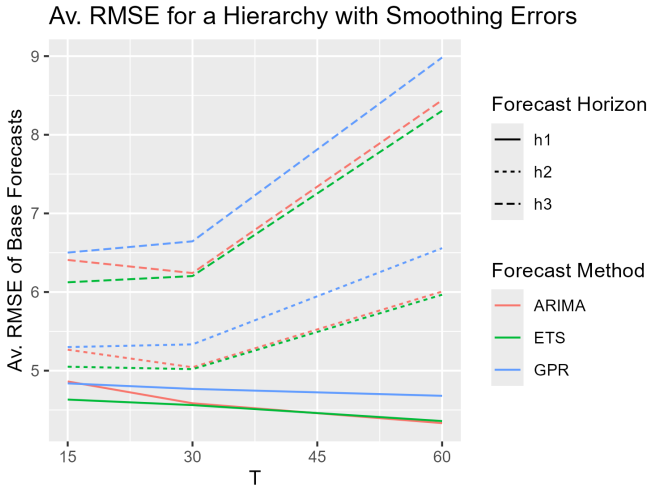


Figure 5: Average RMSE for the various base forecasts in Section 4.3.2 (ARIMA in red, ETS in green, and GPR in blue) by total length of the time series. A forecast horizon of $h1$ corresponds to $h = 1$, $h2$ to $h = 1:2$ in the case of $T = 15, 30$ and $h = 1:4$ for $T = 60$. Similarly, $h3$ corresponds to the longer forecasting horizons of $h = 1:4$ and $h = 1:8$, respectively.

4.3.3. *Impact of Seasonality*

This subsection studies the effect seasonality has on the choice of the reconciliation method. For comparability, the bottom time series $\mathbf{b}_t$ are simulated analogous to the structural time series model consisting of a trend component μ_t, a quarterly seasonal component γ_t, and an error component η_t as used in Wickramasuriya et al. [1]:

$$\begin{aligned}
\mathbf{b}_t &= \mu_t + \gamma_t + \eta_t, \\
\mu_t &= \mu_{t-1} + \nu_t + \varrho_t, & \varrho_t &\sim MN(\mathbf{0}, 2\mathbf{I_8}) \\
\nu_t &= \nu_{t-1} + \zeta_t, & \zeta_t &\sim MN(\mathbf{0}, 0.007\mathbf{I_8}) \\
\gamma_t &= -\sum_{i=1}^{3} \gamma_{t-i} + \omega_t, & \omega_t &\sim MN(\mathbf{0}, 7\mathbf{I_8}),
\end{aligned}$$

with independently distributed initial values μ_0, ν_0, γ_0, γ_1, and $\gamma_2 \sim MN(\mathbf{0}, \mathbf{I}_8)$, where MN denotes the multivariate normal distribution. η_t is simulated as an ARIMA process similar to Section 4.3.1 and as detailed in Table 2 but with d held at zero and p and q only ranging from zero to one. The resulting time series are summed according to the hierarchical structure as illustrated in Figure 3.

Results for ETS forecasts. The performance of the base forecasts can be obtained from Figure 6. In contrast to Section 4.3.1 and Section 4.3.2, it is noticeable that ETS performed poorly in this scenario. This is likely due to an inability of the model to capture and identify both the seasonal and trend components with few observations. Especially a failure to capture trend components could explain the worsening performance as the number of observations increases.

In terms of improvements through reconciliation, MinT achieves superior performance (average 2.6% improvements for MinT vs. a 2.2% improvement for MinTiT) followed by the WLS approaches (1.9% and 1.8% improvements). BU fares worst with only a marginal improvement of 0.1%.

Table 6: Out-of-sample forecast performance for the hierarchy with hierarchically smoothing time series in Section 4.3.2 for ETS-generated forecasts relative to the ETS-generated base forecasts.

	ETS									
	T = 15			T = 30			T = 60			
	h=1	1:2	1:4	h=1	1:2	1:4	h=1	1:4	1:8	Average
	Top-Level									
MinT	**-1.9**	**-1.2**	**-1.5**	**-3.1**	**-3.2**	-4.9	**-2.3**	**-4.3**	-5.7	**-3.1**
WLS_s	2.8	2.7	-0.4	2.5	1.7	-3.9	3.6	-2.6	**-6.8**	0.0
WLS_v	2.8	2.7	-0.4	2.5	1.7	-3.9	3.6	-2.6	**-6.8**	0.0
BU	39.3	32.5	18.4	45.1	37.9	20.3	44.6	19.3	7.1	29.4
$MinTit_g$	-1.3	-0.5	-1.1	-2.7	-2.7	-4.9	-2.1	**-4.3**	-6.1	-2.9
$MinTit_l$	-1.4	-0.6	-1.3	-2.8	-2.8	**-5**	-2.1	**-4.3**	-6.1	-2.9
	Level 1									
MinT	**-3.5**	-2.9	-2.2	-3.7	-4.1	-4.4	**-4.3**	-4.8	-4.8	-3.9
WLS_s	-1.8	-1.6	**-2.4**	-2.1	-2.5	-4.8	-2.1	-4.2	-5.9	-3.0
WLS_v	-1.8	-1.6	**-2.4**	-2.1	-2.6	-4.9	-2.1	-4.3	**-6**	-3.1
BU	20	17.6	11.4	22	19.6	12.3	20.5	11	4.9	15.5
$MinTit_g$	**-3.5**	-2.9	-2.2	**-4.1**	-4.3	-5	**-4.3**	**-4.9**	-5.2	-4.0
$MinTit_l$	**-3.5**	**-3**	**-2.4**	**-4.1**	**-4.4**	**-5.1**	-4.2	-4.8	-5.1	**-4.1**
	Level 2									
MinT	-3.3	-3.3	-2.4	-3.8	-4.1	-3.7	**-4.3**	**-4**	-3.5	-3.6
WLS_s	-3.2	-3.1	**-3.1**	-3.8	-4.2	-4.7	-3	-3.6	-4.3	**-3.7**
WLS_v	-3.2	-3.1	**-3.1**	-3.9	-4.2	**-4.8**	-3.1	-3.7	**-4.4**	**-3.7**
BU	6.8	6.4	4.9	6.4	6.2	4.6	6.1	4.4	2.4	5.4
$MinTit_g$	**-3.5**	**-3.4**	-2.7	**-4.1**	**-4.4**	-4.3	-3.7	-3.8	-3.8	**-3.7**
$MinTit_l$	**-3.5**	**-3.4**	-2.8	**-4.1**	**-4.4**	-4.3	-3.6	-3.8	-3.8	**-3.7**
	Bottom-Level									
MinT	-2.4	-2.4	-1.9	-2.2	-2.3	-1.9	**-3.5**	**-3.1**	-2	-2.4
WLS_s	-2.2	-2.3	-2.2	-2.3	-2.5	-2.7	-2	-2.2	-2.3	-2.3
WLS_v	-2.4	-2.5	**-2.4**	**-2.5**	**-2.7**	**-2.9**	-2.3	-2.6	**-2.8**	-2.6
BU	0	0	0	0	0	0	0	0	0	0.0
$MinTit_g$	**-2.6**	**-2.6**	-2.3	**-2.5**	**-2.7**	-2.8	-2.6	-2.9	**-2.8**	-2.6
$MinTit_l$	**-2.6**	**-2.6**	**-2.4**	**-2.5**	**-2.7**	-2.8	-2.6	-2.9	**-2.8**	**-2.7**
	Average									
MinT	-2.7	-2.6	-2	-2.9	-3.1	-3.2	**-3.7**	**-3.8**	-3.6	-3.1
WLS_s	-2	-1.9	-2.2	-2.3	-2.5	-3.7	-1.9	-3	-4.4	-2.7
WLS_v	-2.1	-2	**-2.3**	-2.4	-2.7	**-3.9**	-2	-3.2	**-4.6**	-2.8
BU	7.8	7.7	5.8	8.6	8.6	6.4	7.9	5.8	3	6.8
$MinTit_g$	-2.8	-2.6	-2.2	**-3.1**	**-3.4**	-3.8	-3	-3.7	-4.2	**-3.2**
$MinTit_l$	**-2.9**	**-2.7**	**-2.3**	**-3.1**	**-3.4**	**-3.9**	-3	-3.6	-4.1	**-3.2**

The *results for the ARIMA forecasts* are similar. ARIMA also struggled with the identification of an appropriate model (Figure 6). However, the RMSE for the ARIMA model stabilizes at $T = 60$ observations. With regard to the reconciliation performance, for which Table 10 provides an overview, the results mirror those of the ETS forecasts but with much larger improvements (except in the case of BU which leads to an increase in RMSE).

Results for GPR forecasts. The GPR forecasts fared significantly better than its ETS and ARIMA counterparts in this scenario and showcase a steady improvement in accuracy as the time series lengths increase (Figure 6). WLS reconciliation achieved by far the best results with an RMSE improvement of 6% ($MinTit_g$: 4.2%, $MinTit_l$, MinT: 3.2%).

4.3.4. *Impact of Differing Time Series Lengths*

This section investigates the impact of differing time series lengths throughout the hierarchical structure. Differing time series lengths in a hierarchical context often occur in practice due to several reasons: Product categories may get more refined over time as markets grow and products become more differentiated or the organizational structure becomes more refined as

Table 7: Out-of-sample forecast performance for the hierarchy with hierarchically smoothing time series in Section 4.3.2 for ARIMA-generated forecasts relative to the ARIMA-generated base forecasts.

	ARIMA									
	T = 15			T = 30			T = 60			
	h=1	1:2	1:4	h=1	1:2	1:4	h=1	1:4	1:8	Average
	Top-Level									
MinT	**-7**	**-6.4**	**-6.5**	**-5**	**-5**	**-5.7**	**-3.7**	**-4.9**	-5.9	**-5.6**
WLS_s	0.4	-0.8	-5.8	4.4	2.3	-3.3	4.4	-2.6	**-6.9**	-0.9
WLS_v	0.4	-0.8	-5.8	4.4	2.3	-3.3	4.4	-2.6	**-6.9**	-0.9
BU	51.1	41	21	63.2	48.4	24.3	55.8	21.1	5.7	36.8
$MinTit_g$	-6	-5.4	-6.1	-4.9	-4.7	-5.6	-3.3	-4.7	-6.1	-5.2
$MinTit_l$	-6.1	-5.5	-6.2	-4.8	-4.7	-5.6	-3.3	-4.7	-6.1	-5.2
	Level 1									
MinT	**-8.2**	**-7.8**	-7	-5.9	**-6**	**-5.8**	-5.3	-4.9	-4.8	**-6.2**
WLS_s	-5.4	-5.7	-7.3	-2.5	-2.9	-4.7	-2.9	-4.1	**-5.8**	-4.6
WLS_v	-5.4	-5.7	**-7.4**	-2.5	-2.9	-4.7	-3	-4.2	**-5.8**	-4.6
BU	24.8	20.7	12	29.8	24.9	15	24.7	12.8	4.5	18.8
$MinTit_g$	-8	-7.6	-6.9	**-6.2**	**-6**	**-5.8**	**-5.5**	**-5**	-5	**-6.2**
$MinTit_l$	-8	-7.6	-6.9	**-6.2**	**-6**	**-5.8**	-5.4	**-5**	-4.9	**-6.2**
	Level 2									
MinT	**-6.3**	**-6.1**	-4.9	**-5.8**	**-5.7**	**-5**	**-5.1**	**-4.5**	-3.5	**-5.2**
WLS_s	-5.7	-5.5	-5.6	-4.5	-4.3	-4.4	-3.9	-3.9	-4.1	-4.7
WLS_v	-5.7	-5.5	**-5.7**	-4.6	-4.4	-4.5	-4	-4.1	**-4.2**	-4.7
BU	9	8.2	5.9	9.9	9.2	6.6	7.3	4.8	2.3	7.0
$MinTit_g$	**-6.3**	-6	-5	**-5.8**	-5.6	**-5**	-4.5	-4.3	-3.6	-5.1
$MinTit_l$	**-6.3**	-6	-5.1	**-5.8**	-5.6	**-5**	-4.5	-4.2	-3.6	-5.1
	Bottom-Level									
MinT	-3.7	-3.6	-2.8	**-4**	-3.9	-3.4	**-3.7**	**-3.3**	-2	-3.4
WLS_s	-3.4	-3.4	-3.3	-3.3	-3.3	-3.1	-2.5	-2.5	-2.2	-3.0
WLS_v	-3.8	-3.7	**-3.6**	-3.6	-3.6	-3.5	-2.7	-2.9	**-2.6**	-3.3
BU	0	0	0	0	0	0	0	0	0	0.0
$MinTit_g$	**-4**	**-3.9**	-3.5	-3.9	**-4**	**-3.8**	-3	-3.1	**-2.6**	**-3.5**
$MinTit_l$	-3.9	**-3.9**	-3.5	-3.8	**-4**	**-3.8**	-3	-3.1	**-2.6**	**-3.5**
	Average									
MinT	**-5.2**	**-5.1**	-4.5	**-4.8**	**-4.7**	-4.5	**-4.2**	**-4.1**	-3.7	**-4.5**
WLS_s	-3.9	-4	-4.9	-3	-3	-3.7	-2.4	-3.1	-4.3	-3.6
WLS_v	-4.1	-4.2	**-5.1**	-3.1	-3.2	-3.9	-2.6	-3.3	**-4.5**	-3.8
BU	10.5	9.7	6.7	11.8	11	7.9	9.4	6.5	2.6	8.5
$MinTit_g$	**-5.2**	**-5.1**	-4.8	-4.7	**-4.7**	**-4.7**	-3.7	-3.9	-4	**-4.5**
$MinTit_l$	**-5.2**	**-5.1**	-4.8	-4.7	**-4.7**	**-4.7**	-3.7	-3.9	-4	**-4.5**

Table 8: Out-of-sample forecast performance for the hierarchy with hierarchically smoothing time series in Section 4.3.2 for GPR-generated forecasts relative to the GPR-generated base forecasts.

	GPR									
	T = 15			T = 30			T = 60			
	h=1	1:2	1:4	h=1	1:2	1:4	h=1	1:4	1:8	Average
	Top-Level									
MinT	**-3.5**	**-3.6**	**-3.9**	**-4.2**	**-3.1**	**-3.5**	**-3.8**	**-1.6**	**-1.9**	**-3.2**
WLS_s	6.4	6.2	2	7.6	9.8	5.5	6.6	7.9	3.7	6.2
WLS_v	6.4	6.2	2	7.6	9.8	5.5	6.6	7.9	3.7	6.2
BU	48.4	39.4	25.7	56.3	52.7	36.6	56.7	42.1	26.6	42.7
$MinTit_g$	-1.8	-1.8	-2.9	-2.5	-1.4	-2.4	-2.1	-0.4	-1.4	-1.9
$MinTit_l$	-1.9	-1.9	-3	-2.6	-1.6	-2.6	-2.2	-0.6	-1.5	-2.0
	Level 1									
MinT	**-5.3**	**-6.1**	**-6.2**	**-5.2**	**-4.9**	**-4.5**	**-5.5**	**-3.8**	**-3.1**	**-5.0**
WLS_s	-0.4	-0.5	-2.1	0.7	2.7	2.1	0.2	3.2	1.9	0.9
WLS_v	-0.4	-0.5	-2.2	0.7	2.8	2.1	0.2	3.2	1.9	0.9
BU	25.8	21.4	15.3	29.1	29.9	24.3	28.9	27.4	19.8	24.7
$MinTit_g$	-4.9	-5.3	-5.6	-4.4	-3.8	-3.5	-4.1	-2.7	-2.4	-4.1
$MinTit_l$	-4.9	-5.3	-5.6	-4.4	-3.8	-3.6	-4.1	-2.8	-2.5	-4.1
	Level 2									
MinT	**-4.8**	**-5.6**	**-5.4**	**-5.2**	**-6.2**	**-6.1**	**-6.8**	**-6.4**	**-5.4**	**-5.8**
WLS_s	-3	-2.7	-2.7	-2.4	-1.9	-1.6	-2.2	-1	-0.9	-2.0
WLS_v	-2.9	-2.8	-2.9	-2.4	-2	-1.7	-2.3	-1.1	-1	-2.1
BU	10.1	8.4	7.3	10.6	11.5	10.7	10	12.2	10.1	10.1
$MinTit_g$	-4.6	-5	-4.9	-4.3	-5	-5	-4	-4.4	-4	-4.6
$MinTit_l$	-4.6	-5	-4.9	-4.3	-5	-5	-3.9	-4.4	-4	-4.6
	Bottom-Level									
MinT	-3.1	-3.5	-3.8	**-5.1**	**-6.1**	**-6.9**	**-7.4**	**-8.4**	**-7.9**	**-5.8**
WLS_s	-3.1	-2.8	-2.9	-3.1	-3.4	-3.6	-2.8	-3.9	-3.8	-3.3
WLS_v	-3.2	-3.1	-3.4	-3.4	-3.9	-4.3	-3.3	-4.5	-4.5	-3.7
BU	0	0	0	0	0	0	0	0	0	0.0
$MinTit_g$	**-3.5**	**-3.8**	**-4.1**	-3.9	-5	-5.8	-4.1	-6.1	-6.4	-4.7
$MinTit_l$	**-3.5**	**-3.8**	**-4.1**	-3.9	-5	-5.8	-4.1	-6.1	-6.4	-4.7
	Average									
MinT	**-3.9**	**-4.4**	**-4.7**	**-5**	**-5.7**	**-5.8**	**-6.8**	**-6.2**	**-5.3**	**-5.3**
WLS_s	-1.9	-1.5	-2	-1.5	-0.8	-0.9	-1.6	-0.4	-0.7	-1.3
WLS_v	-2	-1.7	-2.3	-1.7	-1.1	-1.2	-1.9	-0.7	-0.9	-1.5
BU	10.6	9.8	8.3	12.1	13.5	12.5	11.4	13.9	11.4	11.5
$MinTit_g$	-3.8	-4.1	-4.4	-4	-4.5	-4.7	-3.9	-4.4	-4.1	-4.2
$MinTit_l$	-3.8	-4.1	-4.4	-4	-4.5	-4.8	-3.9	-4.4	-4.2	-4.2

a company grows, leading to more differentiated reporting of revenues over time. Incorporating the available information can allow for more precise forecasts, especially when the time series are already very short.

As in Section 4.3.2, time series according to the hierarchical structure illustrated in Figure 3 are simulated to produce higher SNRs on higher hierarchical levels. The resulting time series were then cut to the respective lengths as detailed in Table 12, which ensures that hierarchically higher time series have at least as many observations as their constituents.

In this scenario, $MinTit_l$ should have an advantage: Because the covariance matrices are calculated separately for each sub-hierarchy, more observations can be used in some cases potentially leading to more accurate estimates. For example, overall, there are only 15 observations without missing data. But the local algorithm can utilize 90 observations to estimate the covariance matrix encompassing the top-level and level 1 time series.

The *results* of this simulation are presented in Table 13. As in Section 4.3.2, it is not surprising that the BU approach performed poorly, leading to an increase in average RMSE of a factor between 9% and over 90% compared to the base forecasts. The WLS approaches achieved a slight error reduction, which was always exceeded by MinT. As expected, the best performing reconciliation method for ARIMA and ETS was $MinTit_l$ with reductions in average RMSE up to over 20%. It performed slightly better than its global variant in this context. When the forecasts were generated with the GPR model, MinT again performed on-par or slightly better than the iterative approaches.

4.3.5. Impact of Degeneracy

This section examines the performance of the reconciliation methods under a degenerate hierarchical structure, such as illustrated in Figure 7. The time series are generated in the same way as in Section 4.3.1 but the bottom time series y_{BBA} and y_{BBB} are deleted after aggregation whereby the illustrated hierarchical structure is created. It should be noted that the convenient interface provided by the R packages `hts` [13] and `fable` is unable to handle this case. However, an adapted implementation of the algorithm found in Hyndman et al. [13] is capable of handling degenerate hierarchies and is provided in Steinmeister and Pauly [16].

Results. In contrast to Section 4.3.3 and in line with the other sections, ARIMA and ETS forecasts proved more accurate in

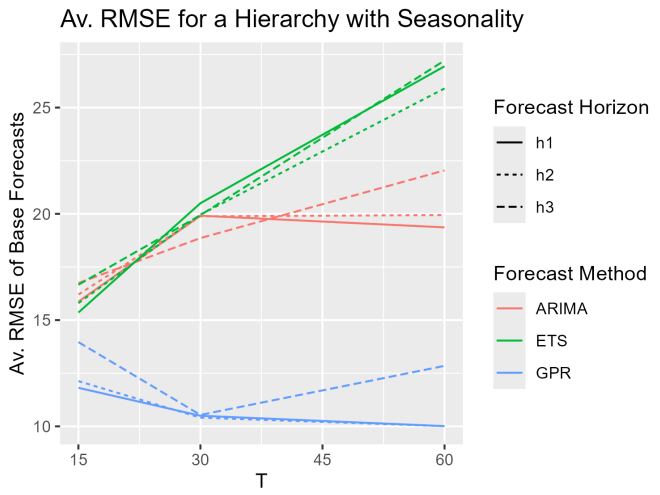


Figure 6: Average RMSE for the various base forecasts in Section 4.3.3 (ARIMA in red, ETS in green, and GPR in blue) by total length of the time series. A forecast horizon of $h1$ corresponds to $h = 1$, $h2$ to $h = 1:2$ in the case of $T = 15, 30$ and $h = 1:4$ for $T = 60$. Similarly, $h3$ corresponds to the longer forecasting horizons of $h = 1:4$ and $h = 1:8$, respectively.

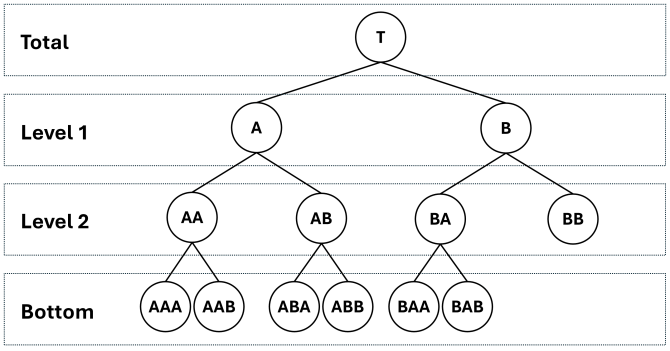


Figure 7: Hierarchical structure for the simulations in Section 4.3.5.

this setting than GPR forecasts, see Figure 8. Similarly, MinT performed best for the GPR forecasts with an average 6.8% RMSE reduction (compared to 6.1% for WLS and 4.6% for MinTit) but MinTit excelled for the ETS and ARIMA forecasts. In the case of ETS base forecasts, MinTit achieved a slightly higher reduction in average RMSE than WLS and MinT: 5.6% against 5.2% and 4.8%, respectively. For the ARIMA base forecasts, the case was a bit more clear cut: MinTit achieved an average RMSE reduction of 9.4% against 8.8% for MinT and 7.5% for WLS.

4.3.6. A "large" Hierarchy

Finally, this section explores the performance of the reconciliation methods on a much larger hierarchy – a scenario which motivated the formulation of the iterative trace minimization algorithm. This hierarchy is made up of 6 times series on the first hierarchical level, followed by 4 component time series each on the following three levels respectively. This results in a total of 6, 24, 96, 384 time series on levels 1 through 4, respectively. The total number of time series is 510. Here, only 500 Monte Carlo iterations were used (i.e. 500 such time series hierarchies

Table 9: Out-of-sample forecast performance for the hierarchy with seasonal time series in Section 4.3.3 for ETS-generated forecasts relative to the ETS-generated base forecasts.

	ETS									
	T = 15			T = 30			T = 60			
	h=1	1:2	1:4	h=1	1:2	1:4	h=1	1:4	1:8	Average
	Top-Level									
MinT	**-4.1**	**-3.7**	**-4.5**	**-2.4**	**-2.9**	**-3.8**	-1.3	**-2.4**	**-3.1**	**-3.1**
WLS_s	-3	-2.6	-2.6	-2	-2.5	-3.1	**-1.5**	-2.1	-2.9	-2.5
WLS_v	-3	-2.6	-2.6	-2	-2.5	-3.1	**-1.5**	-2.1	-2.9	-2.5
BU	-0.8	-0.3	-0.2	0	-0.2	-0.4	-0.3	0	0	-0.2
$MinTit_g$	-3.5	-3.1	-3.4	-2.3	-2.8	-3.5	**-1.5**	-2.2	-3	-2.8
$MinTit_l$	-3.4	-3	-3.3	-2.3	-2.7	-3.4	**-1.5**	-2.2	-3	-2.8
	Level 1									
MinT	**-3.5**	**-3.6**	**-4.4**	-2.1	**-2.5**	**-3.4**	-1.3	**-2.2**	**-3**	**-2.9**
WLS_s	-2.3	-2.5	-2.5	-1.7	-2.1	-2.6	**-1.5**	-1.9	-2.7	-2.2
WLS_v	-2.3	-2.5	-2.5	-1.7	-2.1	-2.6	-1.4	-1.9	-2.7	-2.2
BU	-0.2	-0.3	-0.2	0.1	0.2	0.1	-0.2	0	0	-0.1
$MinTit_g$	-2.8	-3	-3.3	-2	-2.4	-3.1	**-1.5**	-2.1	-2.9	-2.6
$MinTit_l$	-2.8	-3	-3.2	**-2.2**	**-2.5**	-3.2	**-1.5**	-2.1	-2.9	-2.6
	Level 2									
MinT	**-3.2**	**-3.2**	**-3.8**	**-2**	**-2.2**	**-3.2**	-0.8	**-2**	**-2.6**	**-2.6**
WLS_s	-2.1	-2	-1.9	-1.5	-1.6	-2.2	-0.9	-1.6	-2.2	-1.8
WLS_v	-2.1	-2	-2	-1.5	-1.6	-2.3	-0.9	-1.6	-2.2	-1.8
BU	-0.2	-0.1	0	0	0.4	0.1	0.2	0	0	0.0
$MinTit_g$	-2.4	-2.4	-2.6	-1.8	-2	-2.7	**-1**	-1.8	-2.4	-2.1
$MinTit_l$	-2.4	-2.4	-2.5	**-2**	-2.1	-2.8	**-1**	-1.8	-2.4	-2.2
	Bottom-Level									
MinT	**-2.3**	**-2.3**	**-3**	**-1.6**	**-2**	**-2.6**	-0.6	**-1.5**	**-1.8**	**-2.0**
WLS_s	-0.9	-0.9	-1	-0.7	-1	-1.1	-0.6	-0.8	-1.1	-0.9
WLS_v	-0.9	-0.9	-1	-0.9	-1.1	-1.4	-0.6	-0.9	-1.2	-1.0
BU	0	0	0	0	0	0	0	0	0	0.0
$MinTit_g$	-1.1	-1.2	-1.4	-1.2	-1.5	-1.8	**-0.7**	-1.1	-1.4	-1.3
$MinTit_l$	-1.1	-1.2	-1.4	-1.3	-1.6	-1.9	-0.6	-1.1	-1.4	-1.3
	Average									
MinT	**-3.3**	**-3.2**	**-3.9**	**-2**	**-2.4**	**-3.2**	-1	**-2**	**-2.6**	**-2.6**
WLS_s	-2.1	-2	-2	-1.5	-1.8	-2.3	-1.1	-1.6	-2.2	-1.8
WLS_v	-2.1	-2	-2	-1.5	-1.8	-2.4	-1.1	-1.6	-2.3	-1.9
BU	-0.3	-0.2	-0.1	0	0.1	0	-0.1	0	0	-0.1
$MinTit_g$	-2.4	-2.4	-2.7	-1.8	-2.2	-2.8	**-1.2**	-1.8	-2.4	-2.2
$MinTit_l$	-2.4	-2.4	-2.6	-1.9	-2.2	-2.8	**-1.2**	-1.8	-2.4	-2.2

were simulated) due to the high number of time series already contained in this setup. Similarly, Wickramasuriya et al. [1] only used 200 Monte Carlo iterations in their simulation of a large hierarchy.

The 384 bottom time series are simulated similarly to Section 4.3.2. Error terms with variances of 0.4, 1/40, and 1/640 which cancel out levels three, two, and one, respectively. This results in additional variances of about 0.427 on the bottom level, 0.425 on level 3, and 0.4 on level 2.

The *results for ETS forecasts* can be assessed in Table 17. On average, the iterative MinT methods, particularly $MinTit_l$, produced the largest reduction in RMSE (3.6%). The second highest RMSE-reduction was achieved by variance scaling and structural scaling (WLS_v and WLS_s). MinT led to a slight increase in forecasting error (av. 0.7 %), especially on lower hierarchical levels (7.5%).

Results for ARIMA forecasts (Table 18). The reconciliation results for the forecasts based on ARIMA are similar to those based on ETS. $MinTit_l$ and $MinTit_g$ achieved the largest accuracy gains (4.2% and 4% respectively), followed by WLS (3.4% and 3.3%). Again, MinT led to an increase in forecasting error (av. 2.2 %, bottom-level: 10.5%). BU generated an average increase

Table 10: Out-of-sample forecast performance for the hierarchy with seasonal time series in Section 4.3.3 for ARIMA-generated forecasts relative to the ARIMA-generated base forecasts.

	ARIMA									
	T = 15			T = 30			T = 60			
	h=1	1:2	1:4	h=1	1:2	1:4	h=1	1:4	1:8	Average
	Top-Level									
MinT	**-12.7**	**-10.3**	**-10.9**	**-30.3**	**-27.4**	**-23.6**	**-36.8**	**-32.4**	**-26.2**	**-23.4**
WLS_s	-8.1	-6.7	-6.6	-12	-10.8	-10.5	-16.4	-12.6	-9.6	-10.4
WLS_v	-8.1	-6.7	-6.6	-12	-10.8	-10.5	-16.4	-12.6	-9.6	-10.4
BU	-0.2	0.1	-0.1	1.8	0.9	0.9	0.3	-0.1	-0.1	0.4
$MinTit_g$	-10.2	-8.6	-8.9	-21.2	-20.4	-19.3	-28.4	-25.3	-20.1	-18.0
$MinTit_l$	-10.4	-8.7	-9	-21.5	-20.7	-19.2	-28.8	-25.7	-20.4	-18.3
	Level 1									
MinT	**-11.7**	**-9.9**	**-10.2**	**-30**	**-26.2**	**-22.8**	**-35.1**	**-31.3**	**-25.4**	**-22.5**
WLS_s	-6.9	-6.4	-6	-12	-10	-10.1	-14.2	-11.7	-9	-9.6
WLS_v	-7	-6.4	-6	-12.1	-10.1	-10.2	-14.4	-11.8	-9.1	-9.7
BU	0.4	-0.2	0	0.5	0.6	0.1	1.3	-0.3	-0.2	0.2
$MinTit_g$	-9.2	-8.2	-8.3	-21.6	-19.8	-19.2	-27	-25.1	-20.3	-17.6
$MinTit_l$	-9.2	-8.2	-8.3	-22.2	-20.4	-19.3	-28	-25.7	-20.9	-18.0
	Level 2									
MinT	**-10.8**	**-8.7**	**-9**	**-27.8**	**-24**	**-20.1**	**-33.2**	**-28.9**	**-23.5**	**-20.7**
WLS_s	-6.4	-5.4	-5.1	-10	-8.5	-8.2	-11.9	-9.8	-7.6	-8.1
WLS_v	-6.5	-5.5	-5.2	-10.5	-8.9	-8.7	-12.8	-10.6	-8.3	-8.6
BU	-0.5	-0.3	-0.1	0.3	0.1	0.2	0.3	-0.4	-0.3	-0.1
$MinTit_g$	-8	-6.8	-6.9	-19	-17.4	-16.3	-24.7	-22.8	-18.6	-15.6
$MinTit_l$	-8.1	-6.9	-6.9	-19.1	-17.3	-16.1	-25.7	-23.3	-19.1	-15.8
	Bottom-Level									
MinT	**-7.8**	**-6.1**	**-6.4**	**-23.4**	**-19.3**	**-15.7**	**-28.2**	**-23.4**	**-19**	**-16.6**
WLS_s	-2.9	-2.5	-2.5	-5	-4.2	-4.1	-5.9	-4.6	-3.6	-3.9
WLS_v	-3.3	-2.9	-2.8	-7.5	-6.3	-6.3	-10	-8	-6.4	-5.9
BU	0	0	0	0	0	0	0	0	0	0.0
$MinTit_g$	-4.4	-3.7	-3.8	-13.5	-12	-11.2	-18.2	-16.3	-13.4	-10.7
$MinTit_l$	-4.4	-3.7	-3.8	-13.4	-11.7	-10.8	-18.2	-16.1	-13.2	-10.6
	Average									
MinT	**-10.7**	**-8.7**	**-9.1**	**-27.8**	**-24.2**	**-20.5**	**-33.2**	**-28.9**	**-23.5**	**-20.7**
WLS_s	-6.1	-5.2	-5	-9.7	-8.3	-8.2	-12	-9.6	-7.4	-7.9
WLS_v	-6.2	-5.3	-5.1	-10.5	-9	-8.9	-13.3	-10.7	-8.3	-8.6
BU	-0.1	-0.1	-0.1	0.7	0.4	0.3	0.5	-0.2	-0.2	0.1
$MinTit_g$	-7.9	-6.8	-7	-18.8	-17.3	-16.5	-24.4	-22.3	-18.1	-15.5
$MinTit_l$	-8	-6.9	-7	-19	-17.5	-16.3	-25	-22.6	-18.3	-15.6

Table 11: Out-of-sample forecast performance for the hierarchy with seasonal time series in Section 4.3.3 for GPR-generated forecasts relative to the GPR-generated base forecasts.

	GPR									
	T = 15			T = 30			T = 60			
	h=1	1:2	1:4	h=1	1:2	1:4	h=1	1:4	1:8	Average
	Top-Level									
MinT	-6.3	-5	-4.3	-6.2	-6	-5.8	**-5**	**-4.2**	**-5**	-5.3
WLS_s	**-12.8**	**-13.3**	**-18**	**-6.5**	**-6.4**	**-6.4**	-4.5	-3.8	-4.7	**-8.5**
WLS_v	**-12.8**	**-13.3**	**-18**	**-6.5**	**-6.4**	**-6.4**	-4.5	-3.8	-4.7	**-8.5**
BU	-1.6	-0.3	0.9	0.4	-0.3	-0.4	-0.3	0	0	-0.2
$MinTit_g$	-9.7	-9	-10.7	-6.2	-6.1	-6	-4.5	-3.8	-4.7	-6.7
$MinTit_l$	-7.8	-6.7	-8.2	-5.7	-5.6	-5.5	-4.4	-3.8	-4.6	-5.8
	Level 1									
MinT	-3.5	-3.2	-0.5	-5.7	-5.4	-4.9	**-4.3**	**-3.7**	**-4.4**	-4.0
WLS_s	**-9.8**	**-11.3**	**-14**	**-6.1**	**-5.8**	**-5.5**	-3.7	-3.3	-4.2	**-7.1**
WLS_v	**-9.8**	**-11.3**	**-14**	**-6.1**	**-5.8**	-5.4	-3.7	-3.3	-4.2	**-7.1**
BU	1.1	0.9	4.2	0.1	-0.3	0	0.6	0.3	0.3	0.8
$MinTit_g$	-6.2	-6.6	-6.4	-5.7	-5.4	-5	-3.7	-3.4	-4.2	-5.2
$MinTit_l$	-4.1	-4.3	-3.4	-5.2	-4.9	-4.5	-3.6	-3.3	-4	-4.1
	Level 2									
MinT	-1.8	-1.6	0.8	-3.8	-3.9	-3.9	**-3.7**	**-3.4**	**-3.9**	-2.8
WLS_s	**-8**	**-9.4**	**-11.7**	**-4.2**	**-4.2**	**-4.3**	-3.1	-2.8	-3.5	**-5.7**
WLS_v	-7.9	-9.3	**-11.7**	**-4.2**	**-4.2**	**-4.3**	-3.1	-2.8	-3.5	**-5.7**
BU	0.3	0.4	3.4	1	0.5	0.2	0.2	0	0	0.7
$MinTit_g$	-4	-4.2	-3.5	-3.7	-3.8	-3.9	-3.1	-2.9	-3.5	-3.6
$MinTit_l$	-1.6	-1.5	0	-3.1	-3.2	-3.4	-3.1	-2.8	-3.4	-2.5
	Bottom-Level									
MinT	1.8	1.9	2.7	**-2.7**	**-2.5**	**-2.3**	**-2.6**	**-2.2**	**-2.5**	-0.9
WLS_s	**-4.1**	**-4.8**	-7	-2.5	-2.3	-2.2	-1.6	-1.4	-1.7	**-3.1**
WLS_v	-3.7	-4.5	**-7.1**	-2.4	-2.2	-2.2	-1.6	-1.3	-1.7	-3.0
BU	0	0	0	0	0	0	0	0	0	0.0
$MinTit_g$	0	-0.2	-0.5	-2.2	-2	-2	-1.7	-1.5	-1.8	-1.3
$MinTit_l$	1.9	2	2.3	-1.8	-1.7	-1.6	-1.7	-1.5	-1.7	-0.4
	Average									
MinT	-2.4	-1.9	-0.3	-4.6	-4.4	-4.2	**-3.9**	**-3.4**	**-3.9**	-3.2
WLS_s	**-8.7**	**-9.6**	-12.5	**-4.8**	**-4.7**	**-4.6**	-3.2	-2.8	-3.5	**-6.0**
WLS_v	-8.5	-9.5	**-12.6**	**-4.8**	**-4.7**	**-4.6**	-3.2	-2.8	-3.5	**-6.0**
BU	-0.1	0.2	2.1	0.4	0	-0.1	0.1	0.1	0.1	0.3
$MinTit_g$	-5	-4.9	-5.2	-4.5	-4.3	-4.2	-3.2	-2.9	-3.5	-4.2
$MinTit_l$	-2.9	-2.6	-2.2	-4	-3.8	-3.8	-3.2	-2.8	-3.4	-3.2

in RMSE of 11%.

Result for GPR forecasts (Table 19). In contrast to the ETS and ARIMA forecasts, and similar to other subsections, MinT achieved the largest error reduction as measured in RMSE for forecasts based on GPR (5.3% vs. MinTit: 4.3%, WLS: 1.5% and 1.2%). However, GPR had a worse average base-forecast performance than ETS and ARIMA as illustrated in Figure 9. Mirroring the results in Section 4.3.2, the BU approach was the worst performing reconciliation method (with an 11.6% RMSE increase) due to its inability to incorporate the more accurate forecasts of time series on higher hierarchical levels.

5. Case Study: The World Semiconductor Trade Statistics

Over the past decades, the semiconductor market has increasingly been identified as an economic driver. Semiconductors have become virtually ubiquitous in everyday life: they power applications from electronic toothbrushes, smart LEDs, computers, and washing machines to cars, power generators, and data centers. The industry's strategic importance was identified by the US and the European Commission, which allocated $280 billion and 43 billion Euros in subsidies respectively [28, 29]. In addition, the CEO of OpenAI, Sam Altman, is seeking an additional $5-7 trillion in investments to boost fabrication capacity of AI chips [30]. This highlights the importance of the industry and motivates a strong interest in predicting its development.

Ensuring that forecasts be coherent across hierarchical levels allows for consistent planning by executives, investors, and market analysts. Additionally, the simulation results in Section 4 suggest that ensuring adherence to the constraint implied by the hierarchical structure has potential to increase forecast accuracy, allowing practitioners to plan more certainty.

The World Semiconductor Trade Statistics (WSTS) is a premier provider of semiconductor market data [31]. Starting from January 1991, they report monthly worldwide market sizes of semiconductor products (here considered in units of thousand USD), which are hierarchically aggregated into 116 product categories. These are detailed on WSTS' website[1]. A 2021 version is also available from semiconductors.org[2] without login. The semiconductor market is highly dynamic, meaning that these cat-

[1] wsts.org

[2] www.semiconductors.org/wp-content/uploads/2021/02/Product_Classification_2021.pdf

Table 12: Time series lengths for the simulation in Section 4.3.4.

Time Series	**Obs.**	**Time Series**	**Obs.**
Top-Level		Bottom-Level	
T	120	AAA	60
Level 1		AAB	60
A	120	ABA	60
B	90	ABB	60
Level 2		BAA	45
AA	120	BAB	45
AB	90	BBA	15
BA	90	BBB	15
BB	90		

Table 13: Average out-of-sample forecast performance for the hierarchy with differing time series lengths in Section 4.3.4.

	ARIMA			ETS			GPR		
	h=1	1:2	1:4	h=1	1:2	1:4	h=1	1:3	1:6
MinT	-19.6	-19.4	-19.1	-2.8	-3	-3.2	**-4.2**	**-4.3**	**-4.4**
WLS_s	-3.5	-3.5	-3.3	-1.5	-1.6	-2.1	-2.2	-1.4	-1
WLS_v	-2.6	-2.6	-2.5	-1.6	-1.7	-2.2	-2.5	-1.7	-1.4
BU	90.4	90	88.8	10.1	10.4	9.1	12.8	14.6	14.6
MinTit_g	-19.9	-19.8	-19.4	-2.7	-3	-3.2	-4	-4.1	-4.2
MinTit_l	**-20.3**	**-20.2**	**-19.8**	**-2.9**	**-3.1**	**-3.5**	**-4.2**	-4.2	-4.3

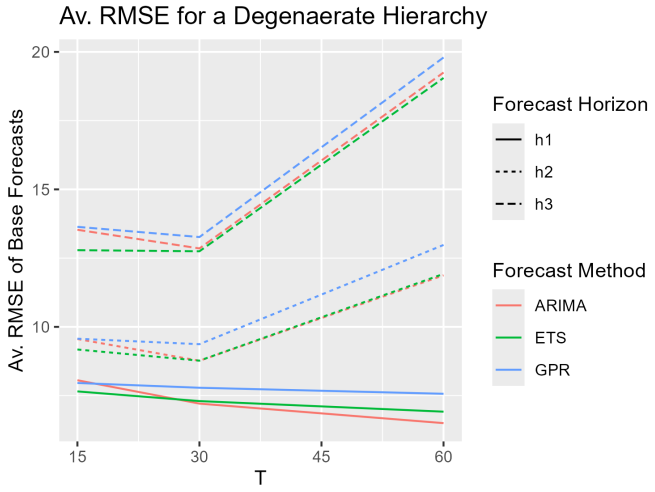


Figure 8: Average RMSE for the various base forecasts in Section 4.3.5 (ARIMA in red, ETS in green, and GPR in blue) by total length of the time series. A forecast horizon of $h1$ corresponds to $h = 1$, $h2$ to $h = 1 : 2$ in the case of $T = 15, 30$ and $h = 1 : 4$ for $T = 60$. Similarly, $h3$ corresponds to the longer forecasting horizons of $h = 1 : 4$ and $h = 1 : 8$, respectively.

egorizations are subject to change as the market develops. Some of the newer ones are aggregated into larger categories to have sufficient historical data to fit quantitative models, leaving a total of 110 product categories following the procedure described in [25]. All data is available from WSTS[3].

5.1. Experimental Design

The semiconductor sales data from January 1991 to August 2023 is obtained from WSTS and split into a training- and a test-set: the training set comprised all data from January 1991 to December 2017 to allow for a minimum of two years of history for model training.

The compared models for the base forecasts are ARIMA obtained with the `auto.arima` function of the `forecast` R-package, Error Trend and Seasonality (ETS), which is also implemented in the `forecast` package[21], and Gaussian Process Regression as implemented in `kernlab`[26]. These are the same models which were compared in the simulation in Section 4.

These models are then successively evaluated on the remaining 68 months of data spanning from January 2018 to August 2023 and comprising roughly 17.4% of the complete data set. A 1-step ahead cross-validation is used to obtain 68 out-of-sample forecasts for each product category. These forecasts are reconciled using the different methods detailed in Section 4.1 and out-of-sample RMSE estimates are computed.

5.2. Results

The results are presented in Table 20. For each hierarchical level, it details the percentage improvements over the respective base forecasts and the average RMSE of the base forecasts on that level. Note that, while the RMSE numbers look quite small, the semiconductor market size is vast. For example, in the case of the average top level forecast error for ARIMA, an RMSE of 1.719 billion USD only amounts to about 4% of the total semiconductor market in August 2023. For the base forecasts, ARIMA achieved the lowest forecast error (average RMSE of 308, 621) across all hierarchical levels, closely followed by GPR (310, 638).

The average RMSE of the ETS base forecasts was about 11% higher than that of the ARIMA model. However, the ETS model performed better on lower hierarchical levels. This mirrors the observations of Steinmeister and Pauly [25], as lower hierarchical levels tend to contain shorter time series.

The bottom-up approach generally fared poorly, suggesting that time series on higher hierarchical levels benefit from some smoothing of aggregation as was explored in Section 4.3.2. Though, BU did achieve a small reduction of average RMSE – even on the highest levels of aggregation – for the ETS forecasts.

For the forecasts generated by ARIMA and ETS, MinT achieved the largest reduction in RMSE: 1.9% and 4.9%, respectively. MinT also resulted in the most consistent improvements in this case study. However, when it comes to the GPR-generated forecasts, MinTit_l excelled with an average reduction of 6% when it converged. Overall, this yields the most accurate forecasts, with an estimated average RMSE of 292, 000.

6. Discussion

The Monte Carlo simulation in Section 4 showcases the potential for additional reductions in expected RMSE in various scenarios by utilizing the iterative reconciliation method introduced in Section 3.1.

MinTit achieved the lowest RMSE values for forecasts based on ARIMA and ETS for correlated time series in Figure 3, time series of differing lengths in Section 4.3.4, in the presence of

[3] wsts.org/61/subscription

Table 14: Out-of-sample forecast performance for the degenerate hierarchical structure in Section 4.3.5 for ETS-generated forecasts relative to the ETS-generated base forecasts.

	ETS									
	T = 15			T = 30			T = 60			
	h=1	1:2	1:4	h=1	1:2	1:4	h=1	1:4	1:8	Average
	Top-Level									
MinT	-6.2	-4.8	-3	-8.1	-7.4	-6.5	-8.2	-6.2	-4.2	-6.1
WLS_s	-7.1	**-6.5**	**-6.3**	-8.3	-8.8	**-8.9**	-6.8	-6.9	**-6.5**	-7.3
WLS_v	-7	-6.4	-6.2	-8.2	-8.6	-8.8	-6.6	-6.8	-6.4	-7.2
BU	-7	-5.4	-4.6	**-9.9**	-8.9	-8	**-8.7**	-6.2	-3.6	-6.9
$MinTit_g$	**-7.2**	-6.1	-5.2	-9.2	**-9**	-8.6	-8.5	**-7.5**	-6.2	**-7.5**
$MinTit_l$	-7.1	-5.9	-5	-9.2	-8.8	-8.3	**-8.7**	**-7.5**	-6.1	-7.4
	Level 1									
MinT	-5.9	-4.9	-3.2	-7.7	-7	-6.1	-7.1	-5.9	-4.4	-5.8
WLS_s	**-6.4**	**-6**	**-5.8**	-7.3	-7.7	**-7.9**	-5.5	-6.1	**-5.8**	-6.5
WLS_v	-6.1	-5.8	-5.6	-7	-7.4	-7.6	-5.3	-6	**-5.8**	-6.3
BU	-6.2	-4.9	-4	**-9**	**-7.9**	-7	**-7.5**	-5.6	-3.2	-6.1
$MinTit_g$	-6.3	-5.5	-4.5	-8.2	**-7.9**	-7.6	-7.3	**-6.8**	**-5.8**	**-6.7**
$MinTit_l$	-6.2	-5.2	-4.2	-8.2	-7.7	-7.3	**-7.5**	**-6.8**	-5.6	-6.5
	Level 2									
MinT	**-3.2**	-2.7	-1.7	-3.9	-3.8	-3.3	-3.3	-2.6	-2.3	-3.0
WLS_s	-3	**-3**	**-3.1**	-3	-3.7	**-4.1**	-1.8	-2.5	-3	-3.0
WLS_v	-2.6	-2.6	-2.7	-2.7	-3.4	-3.7	-1.5	-2.3	-2.9	-2.7
BU	-2.9	-2	-1.6	**-4.5**	**-4**	-3.3	**-3.4**	-2	-0.8	-2.7
$MinTit_g$	-3	-2.6	-2.1	-3.9	**-4**	-4	-3.3	**-3.1**	**-3.1**	**-3.2**
$MinTit_l$	-2.9	-2.3	-1.7	-3.7	-3.7	-3.6	-3.3	-2.9	-2.8	-3.0
	Bottom-Level									
MinT	**-0.8**	**-1.2**	-0.7	0.1	-0.4	-0.7	**-0.8**	-1	-1.8	**-0.8**
WLS_s	0.1	-0.7	**-1.1**	1.6	0.5	-0.5	1.5	-0.3	-1.6	-0.1
WLS_v	0.8	-0.2	-0.5	1.9	0.5	-0.3	1.8	-0.3	-1.5	0.2
BU	0	0	0	**0**	0	0	0	0	0	0.0
$MinTit_g$	-0.4	-1	-1	0.2	**-0.5**	**-1.1**	-0.3	**-1.2**	**-2.2**	**-0.8**
$MinTit_l$	-0.4	-0.9	-0.8	0.4	-0.2	-0.8	-0.4	-1.1	-1.9	-0.7
	Average									
MinT	-4.9	-4	-2.5	-6.3	-5.8	-5.1	-6.2	-4.8	-3.5	-4.8
WLS_s	-5.2	**-4.9**	**-4.8**	-6	-6.4	**-6.7**	-4.6	-5	**-4.9**	-5.4
WLS_v	-5	-4.7	-4.6	-5.7	-6.2	-6.4	-4.4	-4.9	-4.8	-5.2
BU	-5.2	-3.9	-3.2	**-7.6**	**-6.7**	-5.8	**-6.4**	-4.5	-2.4	-5.1
$MinTit_g$	**-5.3**	-4.6	-3.8	-6.9	**-6.7**	-6.5	-6.3	**-5.7**	**-4.9**	**-5.6**
$MinTit_l$	-5.2	-4.4	-3.5	-6.8	-6.5	-6.2	**-6.4**	-5.6	-4.7	-5.5

Table 15: Out-of-sample forecast performance for the degenerate hierarchical structure in Section 4.3.5 for ARIMA-generated forecasts relative to the ARIMA-generated base forecasts.

	ARIMA									
	T = 15			T = 30			T = 60			
	h=1	1:2	1:4	h=1	1:2	1:4	h=1	1:4	1:8	Average
	Top-Level									
MinT	-15	-12.9	-9.5	-11.1	-10.7	-10.1	-8.5	-7.2	-6.3	-10.1
WLS_s	-13.3	-12	**-10.7**	-9.7	-9.8	-10	-8	-7.7	-7.7	-9.9
WLS_v	-13.1	-11.9	-10.5	-9.4	-9.6	-9.8	-7.9	-7.5	-7.6	-9.7
BU	-12.8	-10.7	-8.4	-8.6	-8.8	-9.4	-9	-8.2	-7	-9.2
$MinTit_g$	**-15.3**	**-13.2**	-10.5	**-12.4**	**-12**	**-11.4**	**-10.1**	**-8.8**	**-8.2**	**-11.3**
$MinTit_l$	-15.2	-13	-10.3	-12.2	-11.7	-11.1	-9.9	-8.6	-8	-11.1
	Level 1									
MinT	**-13.1**	**-11**	-7.8	-12.2	-11.1	-9.3	-9.1	-7.9	-6.8	-9.8
WLS_s	-10.6	-9.4	**-8.3**	-9.6	-9.2	-8.5	-7.6	-7.5	-7.3	-8.7
WLS_v	-10.5	-9.2	-8	-9.5	-9	-8.3	-7.7	-7.4	-7.2	-8.5
BU	-10.1	-8.2	-6.2	-7.6	-7.2	-7.2	-8.8	-8.2	-6.7	-7.8
$MinTit_g$	-12.9	-10.8	-8.2	**-12.9**	**-11.8**	**-10.1**	**-10.4**	**-8.9**	**-7.9**	**-10.4**
$MinTit_l$	-12.8	-10.6	-8	-12.7	-11.4	-9.6	-10.2	-8.8	-7.7	-10.2
	Level 2									
MinT	**-8.8**	**-7.2**	**-4.8**	-10.3	-8.8	-6.6	-6.3	-4.8	-3.7	-6.8
WLS_s	-5.4	-4.8	-4.4	-6.2	-5.5	-5	-3.8	-3.7	-3.6	-4.7
WLS_v	-5.3	-4.6	-4.1	-6.6	-5.7	-4.9	-4.2	-3.7	-3.5	-4.7
BU	-4.7	-3.5	-2.4	-3.6	-3.3	-3.6	-4.7	-4.2	-3	-3.7
$MinTit_g$	-8	-6.4	-4.6	**-10.4**	**-8.9**	**-6.8**	**-7.1**	**-5.4**	**-4.4**	**-6.9**
$MinTit_l$	-7.8	-6.1	-4.2	-10	-8.4	-6.1	-6.8	-5.2	-4.1	-6.5
	Bottom-Level									
MinT	**-5.1**	**-4.3**	**-2.6**	**-12.6**	**-9.9**	**-5.7**	-2.6	-1.1	-0.9	**-5.0**
WLS_s	-0.3	-0.9	-1.4	-2.3	-1.9	-1.1	1.1	0.8	-0.2	-0.7
WLS_v	-1.1	-1.4	-1.4	-8.1	-6.3	-3.5	-0.6	0.1	-0.6	-2.5
BU	0	0	0	0	0	0	0	0	0	0.0
$MinTit_g$	-3.9	-3.3	-2.1	-12.5	-9.8	**-5.7**	**-3.3**	**-1.5**	**-1.4**	-4.8
$MinTit_l$	-3.9	-3.1	-1.8	-12.3	-9.5	-5.2	-3.1	-1.3	-1.2	-4.6
	Average									
MinT	**-12.2**	**-10.3**	-7.2	-11.4	-10.3	-8.6	-7.7	-6.2	-5.2	-8.8
WLS_s	-9.6	-8.6	**-7.6**	-8.1	-7.9	-7.5	-6.2	-5.9	-5.8	-7.5
WLS_v	-9.5	-8.5	-7.4	-8.7	-8.2	-7.6	-6.4	-5.9	-5.7	-7.5
BU	-9.1	-7.3	-5.5	-6.3	-6.2	-6.4	-7.2	-6.5	-5.2	-6.6
$MinTit_g$	-11.9	-10	**-7.6**	**-12.1**	**-11**	**-9.4**	**-8.9**	**-7.3**	**-6.5**	**-9.4**
$MinTit_l$	-11.9	-9.8	-7.3	-11.9	-10.7	-9	-8.8	-7.2	-6.2	-9.2

hierarchical degeneracy (Section 4.3.5), and for particularly large hierarchical time series structures in Section 4.3.6.

Both, Section 4.3.2 and Section 4.3.6 incorporate a dependence structure, which leads to higher signal-to-noise-ratios (SNR) on higher hierarchical levels. This is also referred to as the smoothing effect of aggregation. While MinTit outperformed MinT for the ARIMA and the ETS forecasts in the context of the larger hierarchy (Section 4.3.6), this effect was less noticeable in the context of Section 4.3.2, where both approaches achieved similar RMSE reductions.

At the same time, MinT proved remarkably robust in most situations. Considering the challenge of estimating large covariance matrices, MinT with the shrinkage estimator performed surprisingly well even when time series histories were very short.

Furthermore, MinT appears to excel in severe model misspecification as its strong performance for the GPR forecasts and the scenario with seasonality in Figure 6 suggests. Meanwhile, MinTit appears to be more susceptible when forecasts are of poor quality. Possibly, outliers in the residuals are more problematic due to the iterative approach.

As a rule of thumb, MinT may be preferable when the data can be assumed to contain a strong seasonal component – especially when time series are very short and the models have insufficient data to adequately capture this effect. Otherwise, MinTit may improve upon MinT and other existing reconciliation methods for very short time series, large hierarchical structures, or time series of differing lengths.

Acknowledgements

This paper was supported by a research cooperation between Infineon Technologies AG and TU Dortmund University through the Graduate School of Logistics.

Table 16: Out-of-sample forecast performance for the degenerate hierarchical structure in Section 4.3.5 for GPR-generated forecasts relative to the GPR-generated base forecasts.

	GPR									
	T = 15			T = 30			T = 60			
	h=1	1:2	1:4	h=1	1:2	1:4	h=1	1:4	1:8	Average
	Top-Level									
MinT	-12.1	-11	-7.8	**-9.9**	**-8.9**	**-7.7**	**-8.7**	**-5.7**	**-4.8**	-8.5
WLS_s	**-12.8**	**-13.3**	**-18**	-6.5	-6.4	-6.4	-5.7	-4.3	-4.3	**-8.6**
WLS_v	**-12.8**	**-13.3**	**-18**	-6.5	-6.4	-6.4	-5.6	-4.2	-4.3	**-8.6**
BU	-1.6	-0.3	0.9	0.4	-0.3	-0.4	-7.1	-3.9	-4	-1.8
$MinTit_g$	-9.7	-9	-10.7	-6.2	-6.1	-6	-8.1	-5.3	**-4.8**	-7.3
$MinTit_l$	-7.8	-6.7	-8.2	-5.7	-5.6	-5.5	-8.2	-5.4	-4.7	-6.4
	Level 1									
MinT	**-11.3**	-9.7	-6.2	**-9.6**	**-8.1**	**-6.7**	**-8.7**	**-5.1**	**-3.7**	**-7.7**
WLS_s	-9.8	**-11.3**	**-14**	-6.1	-5.8	-5.5	-5.4	-3.4	-3	-7.1
WLS_v	-9.8	**-11.3**	**-14**	-6.1	-5.8	-5.4	-5.2	-3.3	-2.9	-7.1
BU	1.1	0.9	4.2	0.1	-0.3	0	-7.3	-3.5	-2.9	-0.9
$MinTit_g$	-6.2	-6.6	-6.4	-5.7	-5.4	-5	-8	-4.5	-3.5	-5.7
$MinTit_l$	-4.1	-4.3	-3.4	-5.2	-4.9	-4.5	-8.1	-4.7	-3.5	-4.7
	Level 2									
MinT	-7.1	-5.8	-2.4	**-6.4**	**-5.3**	-3.8	**-5.7**	**-3.3**	**-2.2**	-4.7
WLS_s	**-8**	**-9.4**	**-11.7**	-4.2	-4.2	**-4.3**	-2.4	-1.4	-1.3	**-5.2**
WLS_v	-7.9	-9.3	**-11.7**	-4.2	-4.2	**-4.3**	-2.1	-1.2	-1.1	-5.1
BU	0.3	0.4	3.4	1	0.5	0.2	-4	-1.6	-1.3	-0.1
$MinTit_g$	-4	-4.2	-3.5	-3.7	-3.8	-3.9	-4.6	-2.3	-1.6	-3.5
$MinTit_l$	-1.6	-1.5	0	-3.1	-3.2	-3.4	-4.6	-2.3	-1.4	-2.3
	Bottom-Level									
MinT	-2.5	-2.7	-0.7	-2.3	**-2.6**	-1.9	**-2.3**	**-2.1**	**-1.3**	-2.0
WLS_s	**-4.1**	**-4.8**	-7	**-2.5**	-2.3	**-2.2**	1.6	0.2	0	**-2.3**
WLS_v	-3.7	-4.5	**-7.1**	-2.4	-2.2	**-2.2**	1.7	0	-0.1	**-2.3**
BU	0	0	0	0	0	0	0	0	0	0.0
$MinTit_g$	0	-0.2	-0.5	-2.2	-2	-2	-0.9	-1	-0.6	-1.0
$MinTit_l$	1.9	2	2.3	-1.8	-1.7	-1.6	-1	-1.2	-0.7	-0.2
	Average									
MinT	**-9.9**	-8.6	-5.4	**-8.3**	**-7.2**	**-5.9**	**-7.4**	**-4.6**	**-3.5**	**-6.8**
WLS_s	-8.7	**-9.6**	-12.5	-4.8	-4.7	-4.6	-4.2	-2.9	-2.7	-6.1
WLS_v	-8.5	-9.5	**-12.6**	-4.8	-4.7	-4.6	-4	-2.8	-2.7	-6.0
BU	-0.1	0.2	2.1	0.4	0	-0.1	-5.8	-2.8	-2.6	-1.0
$MinTit_g$	-5	-4.9	-5.2	-4.5	-4.3	-4.2	-6.6	-3.9	-3.2	-4.6
$MinTit_l$	-2.9	-2.6	-2.2	-4	-3.8	-3.8	-6.7	-4	-3.1	-3.7

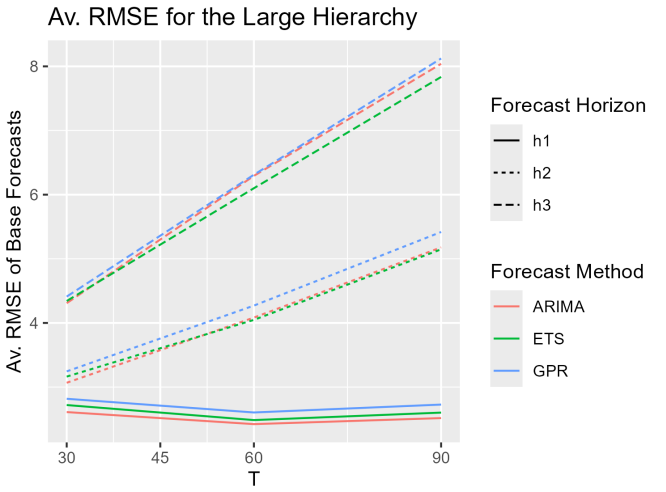


Figure 9: Average RMSE for the various base forecasts in Section 4.3.6 (ARIMA in red, ETS in green, and GPR in blue) by total length of the time series. A forecast horizon of $h1$ corresponds to $h = 1$, $h2$ to $h = 1:2$ in the case of $T = 15, 30$ and $h = 1:3$ for $T = 60$. Similarly, $h3$ corresponds to the longer forecasting horizons of $h = 1:4$ and $h = 1:6$, respectively.

Table 17: Out-of-sample forecast performance for the large hierarchy in Section 4.3.6 for ETS-generated forecasts relative to the ETS-generated base forecasts.

	ETS									
	T = 30			T = 60			T = 90			
	h=1	1:2	1:4	h=1	1:3	1:6	h=1	1:4	1:8	Average
	Top-Level									
MinT	-1.8	-5.7	-5.7	-2.5	-3.6	-1.3	-1.6	-2.7	-1.9	-3.0
WLS_s	-5.1	-6.9	**-9.1**	-1.6	-6.2	**-8.5**	-4.4	**-6.5**	**-6.6**	-6.1
WLS_v	-5.1	-6.9	**-9.1**	-1.6	-6.2	**-8.5**	-4.4	**-6.5**	**-6.6**	-6.1
BU	33.9	25.2	11.2	48	19.2	6	25.2	6.6	1.4	19.6
$MinTit_g$	-5.2	-6.5	-7.4	**-6**	-7.6	-8.1	-4.5	-6.3	-6.1	-6.4
$MinTit_l$	**-6**	**-7**	-7.8	-5.9	**-7.8**	-8.3	**-4.7**	-6.2	-6	**-6.6**
	Level 1									
MinT	-2.4	**-5.8**	-5.9	0.6	-2.3	-2.3	-0.9	-1.8	-1.5	-2.5
WLS_s	-2.7	-3.8	**-7.6**	1.2	-4.3	**-8.6**	0	-4.7	**-6.3**	-4.1
WLS_v	-2.7	-3.8	**-7.6**	1.2	-4.3	**-8.6**	0	-4.7	**-6.3**	-4.1
BU	32.8	27	12.3	42.1	18.1	3.9	37.5	10.4	2.8	20.8
$MinTit_g$	-3.9	-4.7	-6.9	-2.6	-5.3	-7.5	-3.1	-4.8	-5.6	-4.9
$MinTit_l$	**-4.5**	-5.3	-7.4	**-2.8**	**-5.6**	-7.8	**-3.3**	**-5**	-5.7	**-5.3**
	Level 2									
MinT	-1.5	**-3.9**	-3.2	-2	-1.9	0	-1.9	-1.2	0.7	-1.7
WLS_s	-1.4	-1.6	**-4.5**	0.5	-2.8	**-6**	0.4	-3.5	**-4.8**	-2.6
WLS_v	-1.4	-1.6	**-4.5**	0.5	-2.8	**-6**	0.4	-3.5	**-4.8**	-2.6
BU	24.3	21.4	11.5	28	14.5	4.7	27.4	8.6	3	15.9
$MinTit_g$	-2.6	-2.6	-4	-2.2	-3.9	-5.4	-2.4	-3.9	-4.3	-3.5
$MinTit_l$	**-3.1**	-3	**-4.5**	**-2.4**	**-4.1**	-5.6	**-2.6**	**-4.1**	-4.5	**-3.8**
	Level 3									
MinT	-4.4	**-5.3**	-2.2	-5.9	-1.9	3.1	-4	2	5.9	-1.4
WLS_s	-1.3	-1.1	**-2.4**	-0.6	-1.6	**-3.4**	0.9	-1	**-2.2**	-1.4
WLS_v	-1.3	-1.1	**-2.4**	-0.6	-1.7	**-3.4**	0.9	-1.1	**-2.2**	-1.4
BU	10.7	10.8	6.9	11	7.3	2.8	13.7	6.6	3	8.1
$MinTit_g$	-1.7	-1.5	-2	-1.7	-1.9	-2.5	-0.8	-1.4	-1.7	-1.7
$MinTit_l$	**-1.9**	-1.8	**-2.3**	**-1.8**	**-2.2**	-2.7	**-0.9**	**-1.6**	-1.9	**-1.9**
	Bottom-Level									
MinT	3.2	3.2	8.4	1.1	5.8	15	1.2	10.3	19.5	7.5
WLS_s	-1.6	-1.8	-1.8	-1.5	-1.7	-1.5	-1.7	-1.7	-1.5	-1.6
WLS_v	-1.8	-2	-2	-1.7	-2	-2	-1.9	-2.1	-1.9	-1.9
BU	0	0	0	0	0	0	0	0	0	0.0
$MinTit_g$	-1.9	-2.1	-2	**-2.2**	-2.5	-2	**-2.4**	-2.7	-2.2	-2.2
$MinTit_l$	-1.9	-2.2	-2.1	**-2.2**	**-2.6**	**-2.1**	**-2.4**	**-2.8**	**-2.3**	**-2.3**
	Average									
MinT	-0.2	-1.9	-0.2	-1.2	0.2	3.5	-0.8	2.1	4.7	0.7
WLS_s	-2	-2.5	-4.4	-0.8	-2.9	-5.3	-1	-3.3	-4.2	-2.9
WLS_v	-2.1	-2.6	**-4.5**	-0.9	-3	**-5.4**	-1.1	-3.4	**-4.3**	-3.0
BU	13.6	12.6	7.3	15.9	10	3.3	14	6.1	2.1	9.4
$MinTit_g$	-2.5	-2.9	-3.9	-2.4	-3.8	-4.9	-2.4	-3.7	-4	-3.4
$MinTit_l$	**-2.8**	**-3.2**	-4.3	**-2.5**	**-4**	-5.1	**-2.5**	**-3.8**	-4.1	**-3.6**

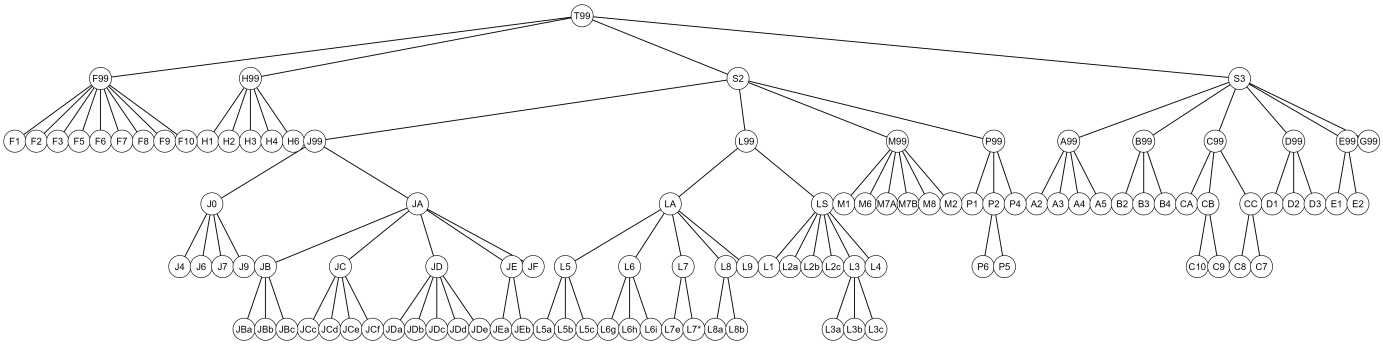


Figure 10: Illustration of the WSTS product classification Hierarchy. L* refers to L7a/b/c/d/f.

Table 18: Out-of-sample forecast performance for the large hierarchy in Section 4.3.6 for ARIMA-generated forecasts relative to the ARIMA-generated base forecasts.

	ARIMA									
	T = 30			T = 60			T = 90			
	h=1	1:2	1:4	h=1	1:3	1:6	h=1	1:4	1:8	Average
	Top-Level									
MinT	1.2	-3.1	-5	3	-3.8	-3.9	3.9	-0.2	-1.1	-1.0
WLS_s	-4.2	-5.8	**-8**	-3.2	-8	**-10.6**	-4.3	**-5.9**	**-6.4**	-6.3
WLS_v	-4.2	-5.8	**-8**	-3.2	-8	**-10.6**	-4.3	**-5.9**	**-6.4**	-6.3
BU	48.1	32.6	14.4	52.8	19	4.1	33.4	7.6	1	23.7
$MinTit_g$	-6.5	-6.6	-7.3	-7.2	-9	-10.1	-4.6	-5.6	-5.6	-6.9
$MinTit_l$	**-6.8**	**-6.9**	-7.7	**-7.4**	**-9.3**	-10.3	**-5**	-5.6	-5.8	**-7.2**
	Level 1									
MinT	0.4	-2.2	-4.5	4.4	-1.7	-3.7	3.1	-2.9	-3.8	-1.2
WLS_s	-2.9	-3.1	-6.7	0.3	-5.7	**-9.9**	0.5	-5.6	**-7.7**	-4.5
WLS_v	-2.9	-3.1	-6.7	0.3	-5.7	**-9.9**	0.5	-5.6	**-7.7**	-4.5
BU	47.8	38	17.7	49.3	18	2.4	46.4	10.4	1.1	25.7
$MinTit_g$	-6.2	-5.4	-6.7	-4.5	-6.5	-8.9	-3.9	-6.2	-7	-6.1
$MinTit_l$	**-6.6**	**-5.8**	**-7.1**	**-4.6**	**-6.6**	-9	**-4.2**	**-6.3**	-7.2	**-6.4**
	Level 2									
MinT	0.7	-2	-3.8	0.7	-0.7	-0.7	0.7	-0.8	0	-0.7
WLS_s	-2.2	-2.1	-5	-0.2	-3.2	**-6.1**	0.2	-3.4	**-5.3**	-3.0
WLS_v	-2.2	-2.1	-5	-0.2	-3.2	**-6.1**	0.2	-3.4	**-5.3**	-3.0
BU	33.1	27.4	13.9	32.7	15.6	4.6	32.4	9.8	2.2	19.1
$MinTit_g$	-4.8	-4.3	-5.4	-3.4	-4.1	-5.4	-3	-4	-4.8	-4.4
$MinTit_l$	**-5.3**	**-4.7**	**-5.7**	**-3.7**	**-4.3**	-5.7	**-3.2**	**-4.2**	-4.9	**-4.6**
	Level 3									
MinT	**-4.2**	**-4.9**	-2.9	**-3.5**	-1.3	1.9	0.6	2.4	5.5	-0.7
WLS_s	-1.6	-1.5	-2.5	-1.2	-1.9	-3.3	0.3	-1.4	-2.9	-1.8
WLS_v	-1.6	-1.5	-2.6	-1.2	-2	**-3.4**	0.3	-1.4	**-3**	-1.8
BU	14.6	13.2	8.2	12.8	7.9	2.9	15.4	6.8	2	9.3
$MinTit_g$	-3.2	-3	-2.9	-2.4	-2.3	-2.6	-1.5	-1.9	-2.4	-2.5
$MinTit_l$	-3.4	-3.2	**-3.1**	-2.5	**-2.5**	-2.8	**-1.6**	**-2**	-2.6	**-2.6**
	Bottom-Level									
MinT	3.4	4.3	7.7	2.2	11.4	20.4	4.6	15.4	25.1	10.5
WLS_s	-2.2	-2.2	-2	-1.8	-1.9	-1.6	-1.9	-1.8	-1.4	-1.9
WLS_v	-2.5	-2.6	-2.4	-2	-2.3	-2	-2.2	-2.3	-1.9	-2.2
BU	0	0	0	0	0	0	0	0	0	0.0
$MinTit_g$	**-2.9**	**-3.1**	**-2.8**	-2.4	-2.7	-2	-2.5	-2.8	-2.1	-2.6
$MinTit_l$	**-2.9**	**-3.1**	**-2.8**	**-2.5**	**-2.8**	**-2.1**	**-2.6**	**-2.9**	**-2.2**	**-2.7**
	Average									
MinT	0.9	-0.2	-0.3	1	2.4	3.6	3	3.8	5.3	2.2
WLS_s	-2.3	-2.5	-4.3	-1.4	-3.6	-6	-1.2	-3.4	-4.8	-3.3
WLS_v	-2.4	-2.7	-4.4	-1.5	-3.7	**-6.1**	-1.3	-3.5	**-4.9**	-3.4
BU	18.4	16.1	9.3	17.6	10.3	2.7	16.6	6.5	1.3	11.0
$MinTit_g$	-3.9	-3.9	-4.5	-3.1	-4.3	-5.5	-2.7	-3.9	-4.4	-4.0
$MinTit_l$	**-4.1**	**-4.1**	**-4.8**	**-3.2**	**-4.5**	-5.7	**-2.8**	**-4**	-4.5	**-4.2**

Table 19: Out-of-sample forecast performance for the large hierarchy in Section 4.3.6 for GPR-generated forecasts relative to the GPR-generated base forecasts.

	GPR									
	T = 30			T = 60			T = 90			
	h=1	1:2	1:4	h=1	1:3	1:6	h=1	1:4	1:8	Average
	Top-Level									
MinT	**-3.5**	**-3.6**	**-3.9**	**-4.2**	**-3.1**	**-3.5**	**-4.5**	**-2.4**	**-2.2**	**-3.4**
WLS_s	6.4	6.2	2	7.6	9.8	5.5	6.7	9.4	6	6.6
WLS_v	6.4	6.2	2	7.6	9.8	5.5	6.7	9.4	6	6.6
BU	48.4	39.4	25.7	56.3	52.7	36.6	54	47	32.8	43.7
$MinTit_g$	-1.8	-1.8	-2.9	-2.5	-1.4	-2.4	-2.6	-0.6	-1.1	-1.9
$MinTit_l$	-1.9	-1.9	-3	-2.6	-1.6	-2.6	-2.7	-0.7	-1.2	-2.0
	Level 1									
MinT	**-5.3**	**-6.1**	**-6.2**	**-5.2**	**-4.9**	**-4.5**	**-5.5**	**-4.8**	**-4.1**	**-5.2**
WLS_s	-0.4	-0.5	-2.1	0.7	2.7	2.1	-0.1	2.9	2.2	0.8
WLS_v	-0.4	-0.5	-2.2	0.7	2.8	2.1	-0.1	2.9	2.2	0.8
BU	25.8	21.4	15.3	29.1	29.9	24.3	27.6	28.5	22.5	24.9
$MinTit_g$	-4.9	-5.3	-5.6	-4.4	-3.8	-3.5	-4.4	-3.6	-3.1	-4.3
$MinTit_l$	-4.9	-5.3	-5.6	-4.4	-3.8	-3.6	-4.4	-3.7	-3.2	-4.3
	Level 2									
MinT	**-4.8**	**-5.6**	**-5.4**	**-5.2**	**-6.2**	**-6.1**	**-6.2**	**-6.7**	**-6.2**	**-5.8**
WLS_s	-3	-2.7	-2.7	-2.4	-1.9	-1.6	-2	-1.1	-0.9	-2.0
WLS_v	-2.9	-2.8	-2.9	-2.4	-2	-1.7	-2.1	-1.2	-1.1	-2.1
BU	10.1	8.4	7.3	10.6	11.5	10.7	10.7	12.5	11.1	10.3
$MinTit_g$	-4.6	-5	-4.9	-4.3	-5	-5	-3.9	-4.5	-4.5	-4.6
$MinTit_l$	-4.6	-5	-4.9	-4.3	-5	-5	-3.8	-4.5	-4.5	-4.6
	Level 3									
MinT	**-4.8**	**-5.6**	**-5.4**	**-5.2**	**-6.2**	**-6.1**	**-6.2**	**-6.7**	**-6.2**	**-5.8**
WLS_s	-3	-2.7	-2.7	-2.4	-1.9	-1.6	-2	-1.1	-0.9	-2.0
WLS_v	-2.9	-2.8	-2.9	-2.4	-2	-1.7	-2.1	-1.2	-1.1	-2.1
BU	10.1	8.4	7.3	10.6	11.5	10.7	10.7	12.5	11.1	10.3
$MinTit_g$	-4.6	-5	-4.9	-4.3	-5	-5	-3.9	-4.5	-4.5	-4.6
$MinTit_l$	-4.6	-5	-4.9	-4.3	-5	-5	-3.8	-4.5	-4.5	-4.6
	Bottom-Level									
MinT	-3.1	-3.5	-3.8	**-5.1**	**-6.1**	**-6.9**	**-6.5**	**-7.9**	**-8.2**	**-5.7**
WLS_s	-3.1	-2.8	-2.9	-3.1	-3.4	-3.6	-2.9	-3.7	-3.9	-3.3
WLS_v	-3.2	-3.1	-3.4	-3.4	-3.9	-4.3	-3.4	-4.3	-4.6	-3.7
BU	0	0	0	0	0	0	0	0	0	0.0
$MinTit_g$	**-3.5**	**-3.8**	**-4.1**	-3.9	-5	-5.8	-4	-5.8	-6.5	-4.7
$MinTit_l$	**-3.5**	**-3.8**	**-4.1**	-3.9	-5	-5.8	-4	-5.8	-6.5	-4.7
	Average									
MinT	**-3.9**	**-4.4**	**-4.7**	**-5**	**-5.7**	**-5.8**	**-6.1**	**-6.5**	**-6**	**-5.3**
WLS_s	-1.9	-1.5	-2	-1.5	-0.8	-0.9	-1.6	-0.5	-0.4	-1.2
WLS_v	-2	-1.7	-2.3	-1.7	-1.1	-1.2	-1.8	-0.8	-0.7	-1.5
BU	10.6	9.8	8.3	12.1	13.5	12.5	11.4	14	12.6	11.6
$MinTit_g$	-3.8	-4.1	-4.4	-4	-4.5	-4.7	-3.9	-4.5	-4.5	-4.3
$MinTit_l$	-3.8	-4.1	-4.4	-4	-4.5	-4.8	-3.9	-4.6	-4.6	-4.3

Table 20: Out-of-sample forecast performance for ARIMA, ETS, and GPR forecasts on the WSTS case study data in Section 5. The numbers for the reconciliation methods are percentage changes relative to the respective base forecasts. In addition, the last row of each section provides the average baseline RMSE of forecasts of the respective model and on the respective level. The last section, which presents the average results, includes a row indicating the average RMSE of forecasts with the respective model and the best average reconciliation method in terms of average RMSE reduction.

	ARIMA	ETS	GPR	ARIMA	ETS	GPR
	Top-Level			Level 3		
MinT	**3.1**	**-6.5**	0.5	**-7**	**-4.7**	-7.1
WLS_s	4.1	-5.3	1.8	-0.6	-1	-2.4
WLS_v	3.2	-5.2	1.3	-5.8	-3.7	-5.8
BU	19.3	-2.4	7.8	1.4	0	-1.9
$MinTit_g$	6.2	-5.4	1.9	-5.5	-4	-6.2
$MinTit_l$	9.3	-5.2	**-1.3**	-3.6	-2.6	**-11.4**
Base	**1,719,047**	2,062,625	1,754,233	**256,173**	259,774	257,230
	Level 1			Level 4		
MinT	3.8	**-5.2**	1	**-9.1**	**-4.8**	-1.7
WLS_s	6	-3.6	3.1	-2.8	-0.2	0.8
WLS_v	**3.6**	-3.6	2.1	-6.2	-2.1	0.8
BU	19.9	-0.7	7.9	-0.1	0.5	0.9
$MinTit_g$	7.3	-4	2.5	-5.7	-3.6	-0.3
$MinTit_l$	11.1	-3.8	**-4.9**	0.2	-2.4	**-8.4**
Base	**802,795**	960,035	839,987	88,835	**87,637**	90,870
	Level 2			Level 5		
MinT	**-8.9**	**-2.3**	-2.3	**-9.7**	**-4.2**	-2.1
WLS_s	-3.1	0	0.6	-2.1	-0.7	-0.2
WLS_v	-7.2	-1.1	-1.3	-7.6	-1.6	-0.2
BU	0.8	2.9	3.6	0	0	0
$MinTit_g$	-7	-1.3	-1.4	-6.9	-3.3	-1.1
$MinTit_l$	-5.1	-0.3	**-8.8**	-1.3	-2.3	**-11.9**
Base	338,426	334,742	**322,153**	75,200	**73,083**	76,204
				Average		
MinT				**-1.9**	**-4.9**	-1.4
WLS_s				1.7	-2.9	1.1
WLS_v				-1.2	-3.6	-0.3
BU				11.2	-0.4	5
$MinTit_g$				0.8	-3.9	-0.2
$MinTit_l$				3.6	-3.3	**-6**
Base				**308,621**	341,970	310,638
Best				302,757	325,213	**292,000**

References

[1] S. L. Wickramasuriya, G. Athanasopoulos, R. J. Hyndman, Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization, Journal of the American Statistical Association 114 (2019) 804–819. URL: https://doi.org/10.1080/01621459.2018.1448825https://www.tandfonline.com/doi/full/10.1080/01621459.2018.1448825. doi:10.1080/01621459.2018.1448825.

[2] G. Athanasopoulos, R. J. Hyndman, N. Kourentzes, A. Panagiotelis, Forecast reconciliation: A review, International Journal of Forecasting (2023).

[3] R. J. Hyndman, Y. Khandakar, Automatic time series forecasting: The forecast package for r, Journal of Statistical Software 27 (2008). URL: http://www.jstatsoft.org/v27/i03/. doi:10.18637/jss.v027.i03.

[4] A. Agrawal, M. Khavkin, J. Slonim, Bringing a real-world edge to forecasting: To improve the accuracy of corporate forecasts, build in the physical parameters from company operations., 2020. URL: https://www.mckinsey.com/capabilities/strategy-and-corporate-finance/our-insights/bringing-a-real-world-edge-to-forecasting, accessed: 24 January 2024.

[5] T. Modis, Life cycles: forecasting the rise and fall of almost anything, Futurist 28 (1994) 20–25.

[6] F. Petropoulos, D. Apiletti, V. Assimakopoulos, M. Z. Babai, D. K. Barrow, S. Ben Taieb, C. Bergmeir, R. J. Bessa, J. Bijak, J. E. Boylan, J. Browell, C. Carnevale, J. L. Castle, P. Cirillo, M. P. Clements, C. Cordeiro, F. L. Cyrino Oliveira, S. De Baets, A. Dokumentov, J. Ellison, P. Fiszeder, P. H. Franses, D. T. Frazier, M. Gilliland, M. S. Gönül, P. Goodwin, L. Grossi, Y. Grushka-Cockayne, M. Guidolin, M. Guidolin, U. Gunter, X. Guo, R. Guseo, N. Harvey, D. F. Hendry, R. Hollyman, T. Januschowski, J. Jeon, V. R. R. Jose, Y. Kang, A. B. Koehler, S. Kolassa, N. Kourentzes, S. Leva, F. Li, K. Litsiou, S. Makridakis, G. M. Martin, A. B. Martinez, S. Meeran, T. Modis, K. Nikolopoulos, D. Önkal, A. Paccagnini, A. Panagiotelis, I. Panapakidis, J. M. Pavía, M. Pedio, D. J. Pedregal, P. Pinson, P. Ramos, D. E. Rapach, J. J. Reade, B. Rostami-Tabar, M. Rubaszek, G. Sermpinis, H. L. Shang, E. Spiliotis, A. A. Syntetos, P. D. Talagala, T. S. Talagala, L. Tashman, D. Thomakos, T. Thorarinsdottir, E. Todini, J. R. Trapero Arenas, X. Wang, R. L. Winkler, A. Yusupova, F. Ziel, Forecasting: theory and practice, International Journal of Forecasting 38 (2022) 705–871. doi:10.1016/j.ijforecast.2021.11.001. arXiv:2012.03854.

[7] L. Steinmeister, B. Ramosaj, L. Schröter, M. Pauly, Testing the limits : A robustness analysis of logistic growth models for life cycle estimation during the covid-19 pandemic, in: Proceedings of the Conference on Production Systems and Logistics: CPSL, volume 2, 2023, pp. 33–44. doi:doi.org/10.15488/15265.

[8] M. Pauly, L. Kuhlmann, A dynamic systems model for an economic evaluation of sales forecasting methods, Tehnički glasnik 17 (2023) 397–404. doi:10.31803/tg-20230511175500.

[9] G. Athanasopoulos, R. A. Ahmed, R. J. Hyndman, Hierarchical forecasts for australian domestic tourism, International Journal of Forecasting 25 (2009) 146–166. URL: http://dx.doi.org/10.1016/j.ijforecast.2008.07.004https://linkinghub.elsevier.com/retrieve/pii/S0169207008000691. doi:10.1016/j.ijforecast.2008.07.004.

[10] L. Kuhlmann, F. Fesca, L. Steinmeister, M. Pauly, Hierarchical sales forecasting in multichannel distribution considering marketing campaigns, in: M. Herberger, D.; Hübner (Ed.), Proceedings of the Conference on Production Systems and Logistics: CPSL, Hannover, 2024, pp. 527–538.

[11] R. J. Hyndman, R. A. Ahmed, G. Athanasopoulos, H. L. Shang, Optimal combination forecasts for hierarchical time series, Computational Statistics and Data Analysis 55 (2011) 2579–2589. URL: http://dx.doi.org/10.1016/j.csda.2011.03.006. doi:10.1016/j.csda.2011.03.006.

[12] A. Panagiotelis, G. Athanasopoulos, P. Gamakumara, R. J. Hyndman, Forecast reconciliation: A geometric view with new insights on bias correction, International Journal of Forecasting 37 (2021) 343–359. URL: https://doi.org/10.1016/j.ijforecast.2020.06.004https://linkinghub.elsevier.com/retrieve/pii/S0169207020300911. doi:10.1016/j.ijforecast.2020.06.004.

[13] R. Hyndman, A. Lee, E. Wang, S. Wickramasuriya, hts: Hierarchical and Grouped Time Series, 2021. URL: https://cran.r-project.org/package=hts.

[14] M. O'Hara-Wild, R. Hyndman, E. Wang, fable: Forecasting Models for Tidy Time Series, 2023. URL: https://cran.r-project.org/package=fable.

[15] K. G. Olivares, F. Garza, D. Luo, C. Challú, M. Mergenthaler, S. B. Taieb, S. L. Wickramasuriya, A. Dubrawski, Hierarchicalforecast: A reference framework for hierarchical forecasting in python, 2022. URL: http://arxiv.org/pdf/2207.03517.pdf.

[16] L. Steinmeister, M. Pauly, Degenerate hierarchical time series reconciliation with the minimum trace algorithm in r, in: D. Herberger, M. Hübner (Eds.), Proceedings of the Conference on Production Systems and Logistics: CPSL, publish-Ing, 2024, pp. 380–390.

[17] J. Schäfer, K. Strimmer, A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics, Statistical Applications in Genetics and Molecular Biology 4 (2005). URL: https://www.degruyter.com/document/doi/10.2202/1544-6115.1175/html. doi:10.2202/1544-6115.1175.

[18] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533–536. doi:10.1038/323533a0.

[19] G. Athanasopoulos, R. J. Hyndman, N. Kourentzes, F. Petropoulos, Forecasting with temporal hierarchies, European Journal of Operational Research 262 (2017) 60–74. URL: https://www.sciencedirect.com/science/article/pii/S0377221717301911. doi:https://doi.org/10.1016/j.ejor.2017.02.046.

[20] R. J. Hyndman, G. Athanasopoulos, Forecasting: Principles and practice, 2nd edition ed., OTexts, Melbourne, 2018. URL: https://otexts.com/fpp2/.

[21] R. J. Hyndman, Y. Khandakar, Automatic time series forecasting: The forecast package for r, Journal of Statistical Software 27 (2008) 1–22. URL: https://www.jstatsoft.org/article/view/v027i03. doi:10.18637/jss.v027.i03.

[22] J. Wang, An intuitive tutorial to gaussian processes regression, Computing in Science & Engineering (2023) 1–8. doi:10.1109/MCSE.2023.3342149.

[23] N. K. Ahmed, A. F. Atiya, N. E. Gayar, H. El-Shishiny, An Empirical Comparison of Machine Learning Models for Time Series Forecasting, Econometric Reviews 29 (2010) 594–621. URL: http://www.tandfonline.com/doi/abs/10.1080/07474938.2010.481556. doi:10.1080/07474938.2010.481556.

[24] S. Makridakis, E. Spiliotis, V. Assimakopoulos, Statistical and Machine Learning forecasting methods: Concerns and ways forward, PLOS ONE 13 (2018) e0194889. URL: https://dx.plos.org/10.1371/journal.pone.0194889{%}0Ahttps://dspace.lboro.ac.uk/dspace-jspui/handle/2134/25091https://dx.plos.org/10.1371/journal.pone.0194889. doi:10.1371/journal.pone.0194889.

[25] L. Steinmeister, M. Pauly, Human vs. machines: Who wins in semiconductor market forecasting?, 2024. arXiv:2404.09334.

[26] A. Karatzoglou, A. Smola, K. Hornik, A. Zeileis, kernlab - an S4 package for kernel methods in R, Journal of Statistical Software 11 (2004). doi:10.18637/jss.v011.i09.

[27] T. van Erven, J. Cugliari, Game-theoretically optimal reconciliation of contemporaneous hierarchical time series forecasts, in: A. Antoniadis, J. Poggi, X. Brossat (Eds.), Modeling and Stochastic Learning for Forecasting in High Dimensions, volume 217 of *Lecture Notes in Statistics*, Springer, 2015, pp. 297–317. URL: https://link.springer.com/10.1007/978-3-319-18732-7{_}15. doi:10.1007/978-3-319-18732-7_15.

[28] M. Taylor, The US chips and science act of 2022, MRS Bulletin 48 (2023) 874–879. doi:10.1557/s43577-023-00581-w.

[29] European Comission, European chips act: Online factpage, 2023. URL: https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/europe-fit-digital-age/european-chips-act_en, accessed: 19 February 2024.

[30] G. Rajan, OpenAI's altman in talks to raise funds for chips, AI initiative - WSJ, 2024. URL: https://www.reuters.com/technology/openais-altman-talks-raise-funds-chips-ai-initiative-wsj-2024-02-09/, accessed: 19 February 2024.

[31] WSTS.org, More than 35 years authentic market monitoring by WSTS, 2024. URL: https://www.wsts.org/, accessed: 24 January 2024.