University of Padova

Department of Mathematics "Tullio Levi-Civita"

Master Thesis in Data Science

Hierarchical Data Forecasting for the Business Sector

Supervisor
Prof. Mariangela Guidolin
University of Padova

Master Candidate
Endrit Sveçla (2041500)

Academic Year
2022-2023

To my parents, my brother, and everyone else who supported me throughout this journey.

Abstract

Data structures that are divided into multiple levels of aggregations are known as Hierarchical data. These structures are very common in the Big Data world we live in and forecasting methods that generate coherent forecasts for all levels are still being developed. This thesis contributes to the Hierarchical Data Forecasting studies, with an emphasis on the Supply Chain Management implementation of these methods. This study dives deep into the End-to-End methodology proposed by Ragnapuram et al. 2021 which presents a novel approach for hierarchical time series forecasting that produces coherent, probabilistic forecasts across all hierarchical levels. The method outperforms state-of-the-art models in accuracy and fit by learning from all the time series data and incorporating reconciliation in a single trainable model.

Further, the method's adaptability is tested in Supply Chain Management demand forecasting, where it manages hierarchical data to create probabilistic projections for strategic and operational planning. Next, the thesis acknowledges the methodology's advantages and limitations especially in model explainability issues for the business sector, in light of the EU's AI Act of 2023. It suggests improving model explainability or modifying the procedure by proposing the replacement of the main model with simpler multivariate alternative models. It also describes the difficulties of switching from point predictions to probabilistic forecasts, which require extensive recoding and the integration of multiple coding systems.

In summary, this thesis contributes to forecasting theory and gives realistic insights and tactics for utilizing End-to-End hierarchical forecasting in Supply Chain Management, balancing accuracy, coherence, and regulatory compliance.

Contents

Listing of figures

Listing of tables

Listing of acronyms

EU ..... European Union
AI ..... Artificial Intelligence
ADF ..... Augmented Dickey-Fuller
ACF ..... Autocorrelation Function
PACF ..... Partial Autocorrelation Function
CDF ..... Cumulative Distribution Function
PIT ..... Probability Integral Transform
CRPS ..... Continuous Ranked Probability Score
MinT ..... Minimum Trace
OLS ..... Ordinary Least Squares
WLS ..... Weighted Least Squares
DLC ..... Convex Optimization Layer
VAR ..... Vector Autoregressive
MA ..... Moving Averages
RNN ..... Recurrent Neural Network
FC ..... Fulfillment Center
SCM ..... Supply Chain Management
SC ..... Supply Chain
NWD ..... New Workable Demand
MSE ..... Mean Squared Error
AE ..... Absolute Error
MAPE ..... Mean Absolute Percentage Error
MAQL ..... Mean Absolute Quantile Loss
MWQL ..... Mean Weighted Quantile Loss
SHAP ..... Shapley Additive Explanations
HITL ..... Human In The Loop
VARIMA ..... Vector Autoregressive Integrating Moving Averages
VISTS ..... Vector Innovations Structural Time Series
ARIMA ..... Autoregressive Integrating Moving Averages
BU ..... Bottom-up

1 Introduction

Forecasting, the art of predicting or estimating future events or trends, is a crucial part of most professions in today's world including economics, meteorology, healthcare, finance, business, social, and many more. It is the process of utilizing past data to create a model that can accurately anticipate future results. Starting from the premise that there is no fully accurate forecast, then one must acknowledge that the precision, dependability, and interpretability of forecasts are of utmost importance, as they shape decisions that might have substantial consequences. It allows organizations to make well-informed strategic decisions, optimize operations, and reduce risks.

During my internship at Amazon's Supply Chain Department (EU), I was lucky enough to witness how a huge company handles its forecasts and more importantly how all these forecasts come together to serve the decision-making process. I contributed to an innovative platform that automates weekly supply chain plans on a warehouse level. These plans were highly complex as they had to consider a wide range of aspects like demand, sales, transportation, human resources, facility, and machine capacities, potential hiring changes, etc. but also, they needed to comply with linear constraints that were implied by higher level forecasts as cluster (all the warehouses within a regional boarder) and network (all the clusters within a continent) forecasts.

Each week, different teams in the department utilized the historical data, statistical techniques, and expert knowledge to forecast their respective variables in order to make these predictions available for the planners which would produce the final product. The forecasts that happened following a top-down approach would lead to teams waiting for other teams' forecasts and spending a lot of time aligning their results manually. In the end, the company used the platform that my team built to bring everything in one place and give an automated version of how the supply chain plan would look. These plans were revised weekly.

As I followed these processes amazed by all the work that goes into them, I could not help noticing the bottlenecks, but more importantly, I got to see the difficult task that hierarchical data forecasting presents. This potential inspired me to learn more about how these processes could be improved by utilizing an end-to-end approach to this problem.

Data often reflects inherent relationships and structures that are commonly found in numerous real-life situations. Many systems and concepts have nested or layered structures like natural language, biological and ecological systems, spatial organizations, etc. These structures arrange data in a hierarchical manner, usually in a fashion resembling a tree, where each level reflects a distinct aggregation or level of detail of the data[37]. Also known as Hierarchical Data Structures, this type of data is common and presents real-world phenomena. In my case, demand data was structured hierarchically, starting from the network level, and descending to clusters, and warehouses. In general, there is no limitation on the number of levels or disaggregation nodes per level.

In the past, the main emphasis in forecasting has primarily focused on point forecasting, which entails forecasting a singular value for each forthcoming data item, even though [52] describes the transition from point estimation to distribution estimation in the nineteenth century. Although it is the most used, especially in the business sector, and enables us to get to measurable and tangible plans, we often overlook the inherent uncertainty in future occurrences, resulting in excessive confidence in projections and potentially unsafe decision-making. Today, we are witnessing a paradigm shift, shown by a transdisciplinary transition from single-valued or point forecasts to distributional or probabilistic forecasts [28].

Moreover, conventional methodologies employed for predicting hierarchical data, such as bottom-up, top-down, and middle-out, have often disregarded the intrinsic organization of the data and therefore they do not make the best use of those data. These models frequently fail to consider the interconnections between various levels of the hierarchy, addressing each level separately [34]. The act of reconciling forecasts, which involves assuring coherence across several levels of hierarchy, is frequently regarded as an independent stage [47] which in a big setting such as Amazon is a very time-consuming process and prone to errors. This strategy may compromise the accuracy of the forecasts, as it fails to properly exploit the information contained in the hierarchical linkages, but instead, it gives priority to a certain level.

From my experience, there is an immense need to centralize the work done by different forecasting teams, into one big model for the entire hierarchy. This would save a lot of time in terms of technical work and manual alignment, optimize the storage and transfer of the results, and increase accuracy by utilizing the data interdependencies, which will lead to lowering operational costs and naturally increase profit. Lastly, for planners to have full visibility of their possible plans, it is very important to acquire probabilistic forecasts which will allow better risk management of different scenarios and a more complete interpretation of them.

An extensive literature review gave me an overview of the latest developments in this field and showed that these issues that I found will likely get more attention in the near future. Discussions with my professor also led me to find the innovative work of Rangapuram, S. S., Werner, L. D., Benidis, K., Mercado, P., Gasthaus, J., and Januschowski, T. (2021, July) on "End-to-End Learning of Coherent Probabilistic Forecasts for Hierarchical Time Series " [47], which is a direct extension of the exact models that are developed and used by Amazon (DeepAR)[6]. This paper represents a notable transformation in the field of hierarchical data forecasting by resolving the limitations of conventional models on incorporating the full forecasting process, which includes considering the hierarchical structure of the data and reconciling projections across different levels, under a cohesive framework. This approach exploits the interdependence of many levels in a hierarchy and provides a probabilistic forecast.

This thesis examines this End-to-End strategy, thoroughly analyzing its intricacies, benefits, and possible limitations. More precisely, this thesis will utilize the End-to-End approach in the context of predicting the demand for a supply chain, which is especially suitable for implementing such a strategy, considering the hierarchical struc-

ture of supply chains and the crucial significance of precise demand prediction in guaranteeing operational effectiveness and consumer contentment. Further, the thesis will investigate possible approaches to enhance the End-to-End technique. The primary objective is to support a foundation work for enhanced, dependable, and all-encompassing prediction methodologies that may effectively guide decision-making in diverse fields.

Chapter 2 lays the theoretical prerequisites to understand Hierarchical data forecasting, while Chapter 3 is dedicated to the details of the End-to-End approach by examining all the model components and phases. Next, in Chapter 4, the implementation of this model to forecast Supply Chain Demand is shown. Finally, in Chapter 5, you may find the discussion on the advantages and the limitations of this approach but also the proposed solutions to overcome these limitations.

2

Hierarchical Data Forecasting Literature Review

2.1 Time Series Forecasting

A forecast is a statement of what is judged likely to happen in the future, especially in connection with a particular situation. This futuristic way of thinking implies paramount importance to the time and the series of events in a defined time window. In this case, the aim is to estimate how the sequence will continue into the future.

Anything that is observed sequentially over time is a time series. It is a common technique in many fields, and it is used to anticipate future events or behaviors. The predictability of an event or a quantity depends on several factors including:

1. How well we understand the factors that contribute to it;
2. How much data is available; and
3. Whether the forecasts can affect the thing we are trying to forecast [32].

Even though some things are easier to forecast than others, as some great forecasters say, there is no fully accurate forecast. Nonetheless, a forecasting model is intended to capture the way things move, not just where things are, and a good forecasting model can capture these patterns that appear. Especially in the business sector, there are Short-term, Medium-term, and Long-term forecasts that are used for different purposes, mainly differentiating between operational and strategic planning. In today's world, an organization needs to develop a forecasting system that involves several approaches to predicting uncertain events, such as problem definition, gathering information, preliminary (exploratory) analysis, choosing and fitting models, and using and evaluating a forecasting model [32].

Time series data can exhibit a variety of patterns, and it is often helpful to split a time series into several components, each representing an underlying pattern category. Understanding this structure is crucial for effective analysis and forecasting. Typically, a time series can be broken down into the following components:

- Trend: This represents the long-term progression of the series. Trends can be upward, downward, or flat. For example, a steady increase in a company's sales over several years would reflect an upward trend.
- Seasonality: These are patterns that repeat at regular intervals. Seasonality is often observed in data with a fixed and known period, like daily temperature changes, monthly sales affected by holidays, or weekly traffic patterns.
- Cyclic: Cyclic patterns occur when data exhibits rises and falls that are not of a fixed frequency. These cycles are often influenced by economic conditions and are usually observed over longer time horizons than seasonal patterns. For instance, business cycles that span several years [32].
- Irregular or Random Component (Noise): This is the unpredictable, random fluctuation that is always present in a time series. Noise is the residual part of the time series after the other components are removed. It represents the randomness or unforeseen events that cannot be attributed to seasonal or cyclic components.
- Level: This refers to the baseline value for the series if it were a flat line, essentially the series' mean [13].

In practice, time-series data may contain some, all, or none of these components, depending on the nature of the data and the context. For instance, financial time series might have trends and noise, but not seasonality. The challenge in time series analysis is to identify these components and adjust for them to make accurate forecasts or derive insights [36].

A fundamental concept in time-series analysis is stationarity since many statistical forecasting methods assume or require it. A stationary time series is one whose properties do not depend on the time at which the series is observed. Thus, time series with trends, or with seasonality, are not stationary — the trend and seasonality will affect the value of the time series at different times. Stationary data is easier to model and results in more accurate results. A time series is said to be stationary if its statistical properties, such as mean, variance, and autocorrelation, are all constant over time. Mathematically stated, a time series is stationary if the following properties hold [32]:

$$E[X_t] = \mu \quad \forall t \in T \tag{2.1a}$$

$$E[X_t^2] < \infty \quad \forall t \in T \tag{2.1b}$$

$$\mathrm{Cov}(X_{t_1}, X_{t_2}) = \mathrm{Cov}(X_{t_1+h}, X_{t_2+h}) \quad \forall t \in \mathbb{N}, \forall h \in \mathbb{Z} \tag{2.1c}$$

Various techniques exist to test for stationarity, starting from visual inspection where you look for changes in mean, variance, or the presence of trends and seasonality, to statistical tests like the Dickey-Fuller (ADF) test, Phillip-Perron test, etc. If a time series is non-stationary, it can often be transformed into a stationary one through techniques like differencing, detrending, or transforming (e.g., taking the logarithm of the series)[40].

Just as correlation measures the extent of a linear relationship between two variables, autocorrelation measures the linear relationship between lagged values of a time series. There are several autocorrelation coefficients, corresponding to each panel in the lag plot. It can be expressed as in equation 2.2:

$$\rho_j = \text{corr}(X_j, X_{j-k}) = \frac{\text{cov}(X_j, X_{j-k})}{\sqrt{V(X_j)}\sqrt{V(X_{j-k})}} \tag{2.2}$$

The autocorrelation coefficient ρ ranges between -1 and 1, where a positive value indicates a positive relationship between the data point and its past value while a negative coefficient suggests a negative relationship. A value of 0 implies no autocorrelation. To evaluate the autocorrelation in a time series the autocorrelation function (ACF) is usually used. This is a plot that shows the autocorrelation coefficient at different lags and can reveal information about seasonality and/or trend [54].

Further, the partial autocorrelation concept complements that of the autocorrelation. It measures the degree of association between observations in a time series separated by various time lags, but, crucially, it does this while controlling for the influence of other time lags. For instance, the PACF at lag 3 would measure the direct relationship between observations three time periods apart, after removing the effect of their correlations at lags 1 and 2. There is also a partial autocorrelation function (PACF) plot that plays an important role in time series model selection and as a diagnostic tool that evaluates the residuals to assess the adequacy of the model. Time series that show no autocorrelation are called white noise [46].

Temporal and cross-temporal aggregations refer to methods of summarizing or combining time series data over time intervals (temporal) and across different time series (cross-temporal). Both aggregations are essential techniques for preprocessing time series data, making it suitable for analysis, modeling, and forecasting[51]. The choice of aggregation method and level depends on the specific goals of the analysis and the nature of the data. Temporal aggregation involves combining data points within a single time series at successive intervals to reduce the frequency of the data. They can alter the properties of a time series by smoothing out noise and revealing underlying trends or cycles, resulting in information loss, especially if higher-frequency fluctuations are significant, or it can affect stationarity. Cross-temporal aggregation involves combining data points across different time series at the same time intervals. This is often done to analyze relationships or to consolidate information [12].

2.2 Probabilistic Forecasting

Point forecasting refers to the process of predicting a single, specific value as a forecast for a future period in a time series. This type of forecasting is widely used in various fields such as finance, economics, weather prediction, inventory management, etc [2]. The conventional interpretation of this value is that it shows what to expect on average if the situation repeats itself many times. In the case of a pure additive model (such as linear regression), the point forecasts correspond to the conditional expectation (mean) from the model.

Taking a historical perspective, [52] describes the transition from point estimation to distribution estimation in the nineteenth century. Today, we are witnessing a paradigm shift, shown by a transdisciplinary transition from single-valued or point forecasts to distributional or probabilistic forecasts [28]. In a nutshell, probabilistic forecasts serve to quantify the uncertainty in a prediction, and they are an essential ingredient of optimal decision-making.

Although probability forecasts for binary events (e.g., an 80% chance of rain today, a 10% chance of a financial meltdown by the end of the year) have been commonly issued for the past several decades attention has been shifting toward probabilistic forecasts for more general types of variables and events. Critical problems of science and society have been driving this development; these problems include weather and climate prediction, flood risk assessment, seismic hazard prediction, predictions about the availability of renewable energy resources, economic and financial risk management, election outcome prediction, demographic and epidemiological projection, health care management, and predictive and preventative medicine. The need for advancement in the theory, methodology, and application of probabilistic forecasting is pronounced, and challenges and opportunities for statistical scientists to become involved and contribute abound [25].

2.2.1 Probabilistic Forecast Properties

A prediction space is a kind of mathematical framework that's used when we want to create and evaluate predictions that come with a measure of uncertainty. These predictions are not just single outcomes, but rather a whole range of possibilities each with its likelihood [27]. Instead of saying "It will rain," which is very certain and may not happen, you provide a range of outcomes with probabilities like "50% chance of rain, 30% chance of cloudy skies, and 20% chance of sunshine."[23].

There are two main components: The actual variable we observe or try to predict, like the weather, which is denoted by Y; and the probabilistic forecast, denoted by $\mathcal{F}$, which is our best guess of what $\mathcal{Y}$ will be, given all the information known at the time. This forecast is expressed as a cumulative distribution function (CDF), which shows all the probabilities of different outcomes (like the percentages of rain, clouds, etc.). The "prediction space" then looks at both $\mathcal{F}$ and $\mathcal{Y}$ together. It's like comparing your forecast ($\mathcal{F}$) to what the actual data ends up being ($\mathcal{Y}$). In more general terms, there might be multiple forecasts ($\mathcal{F}1$, $\mathcal{F}2$, ..., $\mathcal{F}\backslash$) and we are interested in how all of these relate to what actually happens ($\mathcal{Y}$). The prediction space keeps track of all these forecasts and the actual outcome together [25].

Calibration concerns the statistical compatibility between the probabilistic forecasts and the realizations - refers to the degree to which a model's predicted probabilities or values align with the actual outcomes. Essentially, the observations should be indistinguishable from random draws from the predictive distributions. On the other side, dispersion refers to the spread or variability in a set of values or predictions. It's a concept that helps in understanding how much uncertainty or variability is inherent in the predictions made by a model. Dispersion can be relevant in both the outcomes of the model (the predictions) and in the model's performance. Calibration and dispersion thus concern facets of the joint law of the probabilistic forecast and the observation. Since this is a crucial aspect of forecasting, it is critical to check if a forecast is well-calibrated[53].

One of the measures that are frequently utilized to check calibration is The Probability Integral Transform (PIT) [1], which is a statistical concept used to transform a random variable into a uniform distribution. It represents the value that the predictive CDF attains at the observation, with suitable adaptions at any points of discontinuity [20]. While with PIT we can check if there is good model calibration (or under/over-dispersion), it does not provide complete information about other aspects of forecast quality, such as sharpness.

Sharpness refers to the concentration of the predictive distributions. In the case of density forecasts for a real-valued variable, sharpness can be assessed in terms of the associated prediction intervals . The mean widths of these intervals should be as short as possible, subject to the empirical coverage being at the nominal level[25]. A

forecast with high sharpness has a narrow distribution, indicating a high level of confidence or precision in the prediction. Conversely, a less sharp forecast has a wider distribution, reflecting greater uncertainty. However, the value of sharpness must be weighed against the risk of being wrong; highly precise but inaccurate forecasts can lead to misguided decisions [22].

2.2.2 Proper Scoring Rules

Proper scoring rules provide summary measures of the predictive performance that allow for the joint assessment of calibration and sharpness. Generally, we take scores to be negatively oriented penalties that forecasters wish to minimize [28]. The role of scoring rules is to encourage the assessor to make careful assessments and to be honest. We consider a generic convex class $\mathcal{F}$ of probability distributions on $\mathbb{R}$, which we identify with their respective CDFs. A scoring rule assigns a numerical score $S(\mathcal{F}, y)$ to each pair $(\mathcal{F}, y)$, where $\mathcal{F} \in \mathcal{F}$ is a probabilistic forecast and $y \in \mathbb{R}$ is the realized value.

Proper scoring rules encourage forecasters to provide honest and careful quotes [26]. To give a formal definition of proper scoring rules, we write

$$S(\mathcal{F}, \mathcal{G}) = \mathbb{E}_{\mathcal{G}}[S(\mathcal{F}, Y)] \tag{2.3}$$

for the expected score under $\mathcal{G}$ when the probabilistic forecast is $\mathcal{F}$, for $\mathcal{F}, \mathcal{G} \in \mathcal{F}$, assuming tacitly that the expectation is well defined. The extended real line is denoted by $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$.

Definition: The scoring rule $S : \mathcal{F} \times \mathbb{R} \to \overline{\mathbb{R}}$ is *proper* relative to the class $\mathcal{F}$ if

$$S(\mathcal{G}, \mathcal{G}) \leq S(\mathcal{F}, \mathcal{G}) \tag{2.4}$$

For all $\mathcal{F}, \mathcal{G} \in \mathcal{F}$. It is strictly proper if Equation 2.3 holds with equality only if $\mathcal{F} = \mathcal{G}$.

Thus, a proper scoring rule is designed such that quoting the true distribution as the forecast distribution is an optimal strategy in expectation. This property is critically important, as the use of improper scoring rules can lead to grossly misguided inferences about predictive performance [24] [26].

In this work, an important score to mention is the Continuous ranked probability score (CRPS), defined by (Epstein 1969, Matheson and Winkler 1976)[53], which is used by the author of the End-to-End methodology to evaluate the accuracy of the forecasts but also compare point forecasts and probabilistic forecasts. CRPS is a strictly proper scoring rule [24], meaning a sample scores better (lower) when it is drawn from the true distribution. Following [41], given a univariate predictive CDF $\hat{F}_{t,i}$ for time series i, and a ground-truth observation $y_{t,i}$, CRPS can be defined as

$$\text{CRPS}(\hat{F}_{t,i}, y_{t,i}) := \sum_i \int_0^1 Q_S^q \left(\hat{F}_{t,i}^{-1}(q), y_{t,i} \right) dq, \tag{2.5}$$

where Q_S^q is the quantile score (or pin-ball loss) for the q-th quantile:

$$Q_S^q = 2 \left(\mathbb{I} \left\{ y_{t,i} \leq \hat{F}_{t,i}^{-1}(q) \right\} - q \right) \left(\hat{F}_{t,i}^{-1}(q) - y \right) \tag{2.6}$$

2.3 Hierarchical and Grouped Time Series

Time series data can often be broken down by various attributes of interest. Consider, for instance, the total revenue generated by a software company. The revenue can be disaggregated by product types, such as desktop software, mobile apps, cloud services, and enterprise solutions. Each of these categories can be further divided; for example, cloud services can be split into storage, computing, and database services. These subcategories are part of the larger category groups, forming a hierarchical aggregation structure in the time series data. Such a structure is known as **hierarchical time series**.

These hierarchical structures are also commonly observed in sales data segmented by geographic location [32]. The total revenue of the software company could be broken down by continent, then within each continent by country, within each country by state or province, and so forth, down to individual cities or sales outlets.

A hierarchical time series is a multivariate time series that satisfies linear aggregation constraints. Such aggregation constraints encode a tree hierarchy as we can notice in Figure 2.1:

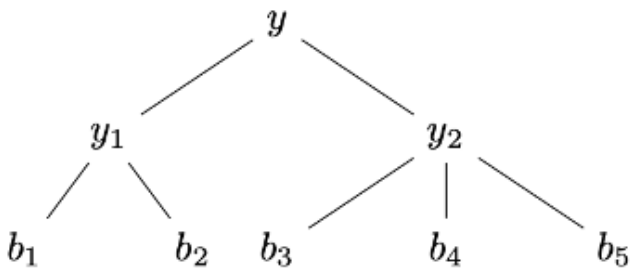

Figure 2.1: Example of hierarchical time series structure for n = 8 time series with m = 5 bottom and r = 3 aggregated time series. Source: Rangapuram et Al. 2021 [47]

To get to a formal definition of the structure, following the notation proposed in Hyndman et al. 2022 [33], we consider a time horizon $t = 1, \ldots, T$. Let $\mathbf{y}_t \in \mathbb{R}^n$ denote the values of a hierarchical time series at time t, with $y_{t,i} \in \mathbb{R}$ the value of the i-th (out of n) univariate time series. Assume that the index i of the individual time series is given by the level-order traversal of the hierarchical tree going from left to right at each level. Further, let $\mathbf{x}_{t,i} \in \mathbb{R}^k$ be time-varying covariate vectors associated with each univariate time series at time t, and $\mathbf{x}_t := [\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,n}] \in \mathbb{R}^{k \times n}$. The shorthand is used $\mathbf{y}_{1:T}$ to denote the sequence $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T\}$[47].

Referring to the time series at the leaf nodes of the hierarchy as bottom-level series and those of the remaining nodes as aggregated series. Also, let's call a given set of forecasts for all time series in the hierarchy that are generated without heeding the aggregation constraint as base forecasts (not to be confused with bottom-level). For notational convenience we split the vector of all series $\mathbf{y}_t$ into m bottom entries and r aggregated entries such that:

$$\mathbf{y}_t = \begin{bmatrix} \mathbf{a}_t \\ \mathbf{b}_t \end{bmatrix} \text{ with } \mathbf{a}_t \in \mathbb{R}^r \text{ and } \mathbf{b}_t \in \mathbb{R}^m. \tag{2.7}$$

Clearly $n = r + m$. For an individual hierarchy or grouping, an aggregation matrix $\mathbf{S} \in \{0,1\}^{n \times m}$ is defined and the $\mathbf{y}_t$, $\mathbf{b}_t$, and $\mathbf{S}$ satisfy:

$$\mathbf{y}_t = \mathbf{S}\mathbf{b}_t \quad \Leftrightarrow \quad \begin{bmatrix} \mathbf{a}_t \\ \mathbf{b}_t \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{\text{sum}} \\ \mathbf{I}_m \end{bmatrix} \mathbf{b}_t \tag{2.8}$$

for every t. $\mathbf{S}_{\text{sum}} \in \{0,1\}^{r \times m}$ is a summation matrix and $\mathbf{I}_m$ is the $m \times m$ identity matrix. It is also useful to equivalently represent as:

$$\mathbf{A}\mathbf{y}_t = 0 \tag{2.9}$$

where $\mathbf{A} := [\mathbf{I}_r | -\mathbf{S}_{\text{sum}}] \in \{0,1\}^{r \times n}$, 0 is an r-vector of zeros, and $\mathbf{I}_r$ is the $r \times r$ identity. The last formulation allows for a natural definition of forecast error[11].

We can illustrate our notation with the example in Figure 2.2. For this hierarchy,

$$\mathbf{a}_t = \begin{bmatrix} y_t \\ y_{1,t} \\ y_{2,t} \end{bmatrix} \in \mathbb{R}^3 \text{ and } \mathbf{b}_t = \begin{bmatrix} b_{1,t} \\ b_{2,t} \\ b_{3,t} \\ b_{4,t} \\ b_{5,t} \end{bmatrix} \in \mathbb{R}^5.$$

The aggregation matrix $\mathbf{S}$ is:

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{\text{sum}} \\ \mathbf{I}_5 \end{bmatrix} = \left[\begin{array}{ccccc} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ \hline & & \mathbf{I}_5 & & \end{array}\right]$$

In hierarchical time series forecasting, one is typically interested in producing forecasts for all the time series in the hierarchy for a given number τ of future time steps after the present time T. Here τ is the length of the prediction or forecast horizon. Since the forecasts are either point predictions or probabilistic in nature, in which case they can be represented as a set of Monte Carlo samples drawn from the forecast distribution. For $h \leq \tau$ we denote an h-period-ahead forecast sample by $\hat{\mathbf{y}}_{T+h}$, with the entire set of samples that comprises the probabilistic forecast written as $\{\hat{\mathbf{y}}_{T+h}\}$ [47].

Grouped time series reflect a complex aggregation structure, which expands and goes beyond the straightforward hierarchical layers. In such time series, the categorization does not follow a singular, top-down hierarchical division but rather involves a blend of factors that are both nested within each other and intersecting [32].

For instance, considering the same example of the software company, we could break down this data not only by geographical location, such as by continent, country, and city, but also by cloud services like storage, computing, and database services, so it does not have to be one or the other. Moreover, each of these categories can be cross-segmented; for instance, computing sales can be analyzed for each city within each country. In this case, the cloud categories are "crossed" with the geographic segmentation.

This crossed structure allows for a multidimensional analysis of time series data, where one could investigate a certain product's sales trends in each specific region or any combination of the subcategories. The complexity of

grouped time series lies in the fact that the disaggregation can occur along multiple dimensions that are not strictly hierarchical but interrelated. In figure 2.2 there is a visual example of these structures:

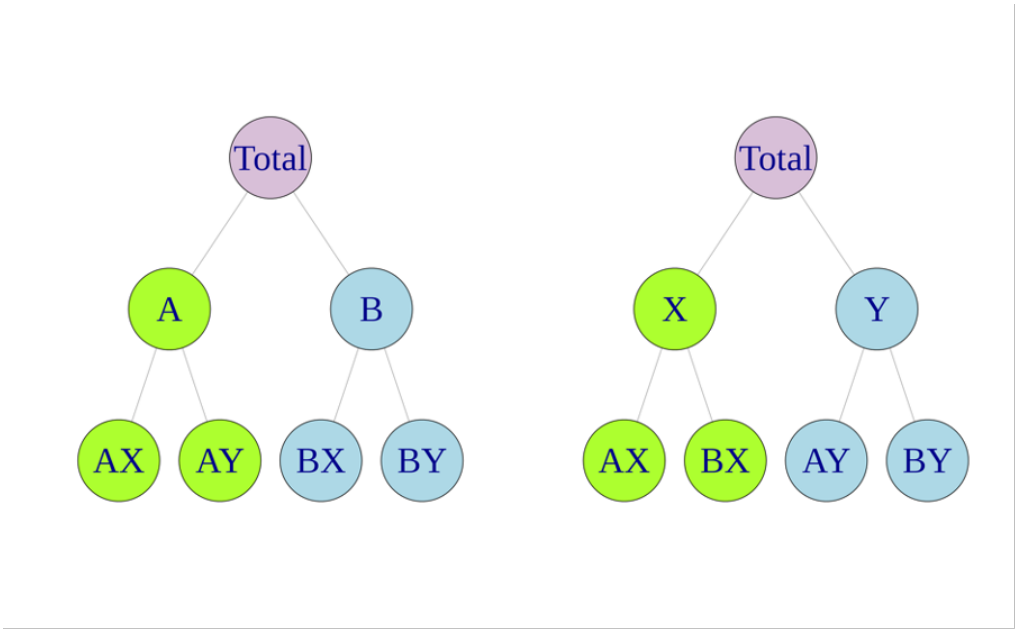

Figure 2.2: Alternative representations of a two level grouped structure. Source: [32]

2.4 Reconciliation

Reconciliation is the process of ensuring that forecasts at different levels of the hierarchy are coherent with each other. Coherence is a principle that ensures the predictions made at different levels of a hierarchy or grouped series add up correctly and are compliant with the forecasts of the other levels. In other terms, is the requirement that the forecasts generated respect the aggregation constraints implied by the structure. Coherence in forecasting is crucial for ensuring that all parts of a hierarchical structure are working together effectively and that strategic decisions are based on a reliable and consistent view of the future [11].

Before the development of forecast reconciliation, the focus was on forecasting a subset of variables at some selected level of aggregation and subsequently aggregating or disaggregating these to generate coherent forecasts for all series. So rather than generating forecasts for all the time series and then implying coherence, the forecasts that were generated were coherent by construction, these methods are known as Single Level Approaches.

A simple method for generating coherent forecasts is the **bottom-up** approach [9]. This approach involves first generating forecasts for each series at the bottom level, and then summing these to produce forecasts for all the series in the structure. An advantage of this approach is that we are forecasting at the bottom level of a structure, and therefore no information is lost due to aggregation. On the other hand, bottom-level data can be quite noisy and more challenging to model and forecast.

Top-down approaches only work with strictly hierarchical aggregation structures, and not with grouped structures. They involve first generating forecasts for the Total series y_t, and then disaggregating these down the hierarchy. We let $(p_1, p_2, \ldots, p_m)$ be a set of disaggregation proportions that dictate how the forecasts of the Total series are to be distributed to obtain forecasts for each series at the bottom level of the structure. The two most common top-down approaches specify disaggregation proportions based on the historical proportions of the data; they involve Average historical proportions, Proportions of the historical averages, and Forecast proportions [29].

The **middle-out** approach combines bottom-up and top-down approaches. First, a "middle level" is chosen and forecasts are generated for all the series at this level. For the series above the middle level, coherent forecasts are generated using the bottom-up approach by aggregating the "middle-level" forecasts upwards. For the series below the "middle level", coherent forecasts are generated using a top-down approach by disaggregating the "middle level" forecasts downwards[32].

An alternative approach emerged with [9] and [34] who recommended producing forecasts of all series (referred to as 'base' forecasts) and then adjusting, or 'reconciling', these forecasts to be coherent. These papers formulated reconciliation as a regression model, reconciling the base forecasts by projecting them onto a subspace for which aggregation constraints hold. Subsequent work has formulated reconciliation as an optimization problem where weights are chosen to minimize a loss, such as a weighted squared error [55], a penalized version thereof [15], or the trace of the forecast error covariance [57]. Other available methods in the literature either follow a projection matrix-based approach or an empirical copula-based reordering approach to revise the incoherent future sample paths to obtain reconciled probabilistic forecasts [32].

The popularity of forecast reconciliation methods can be attributed to several factors. Forecasts across different aggregation levels may be generated by different departments or 'silos' within an organization, as in the case of Amazon, using different sets of predictors, modeling approaches, or expert judgment. Potentially, these are viewed as optimal within these divisions. Reconciliation represents a way to combine information via the sharing of forecasts, thus breaking down these silos. Although it may be difficult to share forecasting processes and associated information across different parts of a large organization, the forecasts themselves are much easier to share and reconcile.

2.4.1 Mapping Matrices

Suppose we forecast all series independently ignoring the aggregation constraints and get the base forecasts, which we denote by $\hat{y}_h$ where h is the forecast horizon. They are stacked in the same order as the data y_t. Then all forecasting approaches for either hierarchical or grouped structures can be represented as

$$\tilde{y}_h = \mathbf{SG}\hat{y}_h \tag{2.10}$$

where $\mathbf{G}$ is a matrix that maps the base forecasts into the bottom-level, and the summing matrix $\mathbf{S}$ sums these up using the aggregation structure to produce a set of coherent forecasts $\tilde{y}_h$ [32].

The $\mathbf{G}$ matrix is defined according to the approach implemented. For example, if the bottom-up approach is used to forecast the hierarchy of Figure 2.1, then

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

G contains two partitions, the first three columns zero out the base forecasts of the series above the bottom-level, while the m-dimensional identity matrix picks only the base forecasts of the bottom-level. These are then summed by the **S** matrix which is constructed based on the structure of the hierarchy[32].

We can rewrite the previous equation as:

$$\tilde{y}_h = \mathbf{P}\hat{y}_h \tag{2.11}$$

where $\mathbf{P} = \mathbf{SG}$ is a "projection" or a "reconciliation matrix". That is, it takes the incoherent base forecasts $\hat{y}_h$, and reconciles them to produce coherent forecasts $\tilde{y}_h$ [31].

Thus far, the methods outlined did not involved reconciliation of multiple time series, since up to now the methods relied on forecasts from only one level of the aggregation structure, which are then either aggregated or disaggregated to provide forecasts at all subsequent levels. However, it is possible to utilize other **G** matrices, whereby **P** will combine and reconcile the underlying forecasts to generate consistent forecasts. Indeed, it is possible to determine the ideal **G** matrix that yields the most precise reconciled forecasts.

2.4.2 The Optimal Reconciliation Approach

Optimal forecast reconciliation will occur if we can find the **G** matrix which minimises the forecast error of the set of coherent forecasts. First, it is checked if the forecasts obtained are unbiased . If the base forecasts $\hat{y}_h$ are unbiased, then the coherent forecasts $\tilde{y}_h$ will be unbiased provided $\mathbf{SGS} = \mathbf{S}$ [30]. This provides a constraint on the matrix **G**. Interestingly, no top-down method satisfies this constraint, so all top-down methods are biased [31].

MinT

Next we find the error in our forecasts. Wickramasuriya et al. (2019) [57] show that the variance-covariance matrix of the h-step-ahead coherent forecast errors is given by:

$$\mathbf{V}_h = \text{Var}[\mathbf{y}_{T+h} - \tilde{\mathbf{y}}_h] = \mathbf{SGW}_h\mathbf{G}'\mathbf{S}' \tag{2.12}$$

where $\mathbf{W}_h = \text{Var}[(\mathbf{y}_{T+h} - \hat{\mathbf{y}}_h)]$ is the variance-covariance matrix of the corresponding base forecast errors. The objective is to find a matrix **G** that minimises the error variances of the coherent forecasts. These error variances are on the diagonal of the matrix $\mathbf{V}_h$, and so the sum of all the error variances is given by the trace of the matrix $\mathbf{V}_h$. Wickramasuriya et al. (2019) [57] show that the matrix **G** which minimises the trace of $\mathbf{V}_h$ such that $\mathbf{SGS} = \mathbf{S}$, is given by

$$\mathbf{G} = (\mathbf{S}'\mathbf{W}_h^{-1}\mathbf{S})^{-1}\mathbf{S}'\mathbf{W}_h^{-1}$$

Therefore, the optimal reconciled forecasts are given by:

$$\tilde{\mathbf{y}}_h = \mathbf{S}(\mathbf{S}'\mathbf{W}_h^{-1}\mathbf{S})^{-1}\mathbf{S}'\mathbf{W}_h^{-1}\hat{\mathbf{y}}_h \tag{2.13}$$

which is referred to as the MinT or Minimum Trace estimator.

OLS

Set $\mathbf{W}_h = k_h\mathbf{I}$ for all h, where $k_h > 0$. This is the most simplifying assumption to make, and means that $\mathbf{G}$ is independent of the data, providing substantial computational savings. The disadvantage, however, is that this specification does not account for the differences in scale between the levels of the structure, or for relationships between series. The weights here are referred to as OLS (ordinary least squares) because setting $\mathbf{W}_h = k_h\mathbf{I}$ in 2.13 gives the least squares estimator [32].

WLS

Set $\mathbf{W}_h = k_h\text{diag}(\hat{\mathbf{W}}_1)$ for all h, where $k_h > 0$

$$\hat{\mathbf{W}}_1 = \frac{1}{T}\sum_{t=1}^{T}\mathbf{e}_t\mathbf{e}_t'$$

and $\mathbf{e}_t$ is an n-dimensional vector of residuals of the models that generated the base forecasts stacked in the same order as the data. This specification scales the base forecasts using the variance of the residuals and it is therefore referred to as the WLS (weighted least squares) estimator using variance scaling [32].

Shrinkage estimator

Set $\mathbf{W}_h = k_h\mathbf{W}_1$ for all h, where $k_h > 0$. Here we only assume that the error covariance matrices are proportional to each other, and we directly estimate the full one-step covariance matrix $\mathbf{W}_1$. The most simple way would be to use the sample covariance. However, for cases where the number of bottom-level series m is large compared to the length of the series T, this is not a good estimator. Instead, a shrinkage estimator is used, which shrinks the sample covariance to a diagonal matrix [32].

3

The End-to-End Methodology

The End-to-End methodology is a novel approach to probabilistic forecasting of hierarchical time series that incorporates both learning and reconciliation into a single end-to-end model. Model parameters are learned simultaneously from all the time series in the hierarchy. The probabilistic forecasts from the model are guaranteed to be coherent without requiring any post-processing step. Two primary components comprise the approach:

1. A forecasting model that produces a multivariate forecast distribution over the prediction horizon; and
2. A sampling and projection step where samples are drawn from the forecast distribution, and are then projected onto the coherent subspace [47].

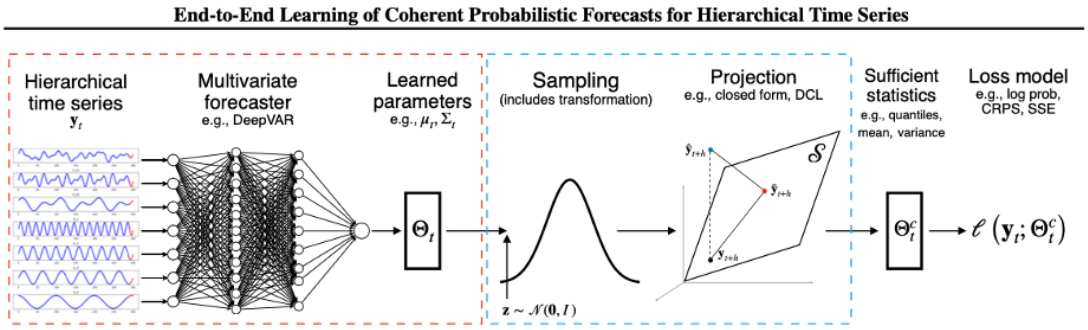

Figure 3.1: Model architecture. Hierarchical time series data is used to train a multivariate forecaster. Learned distribution parameters along with the reparameterization trick allow this distribution to be sampled during training. Optionally, a nonlinear transformation of the samples (e.g., normalizing flow) can account for data in a non-Gaussian domain. Samples are then projected to enforce coherency. From the empirical distribution represented by the samples, sufficient statistics Θct can be computed and used to define an appropriate loss. Source: Rangapuram et Al. 2021 [47]

The structural representation of this methodology can be found in Figure 3.1. The main insight behind the proposed method is the fact that when these components are amenable to autodifferentiation, they constitute a single global model whose parameters are learned end-to-end by minimizing a loss on the coherent samples directly. First, the differentiability of the sampling operation, thanks to the reparametrization trick (Kingma & Welling, 2013) [38], and second, the implementation of the reconciliation (projection) step on samples can be formed as a convex optimization layer (DCL) (Amos & Kolter, 2017; Agrawal et al., 2019a;b) [8]. In the setting of hierarchical and grouped time series, the optimization problem has a closed-form solution requiring only a matrix-vector multiplication (with a pre-computable matrix) and hence is trivially differentiable. However, the proposed approach can handle more sophisticated constraints than those imposed by hierarchical settings via DCL [47]. This allows one to combine typically independent components (generation of base forecasts, sampling, and reconciliation) into a single trainable model.

This methodology referred to as *DeepVARHierarchical* uses *DeepVAR* model as the base multivariate forecaster because of its proven performance and compatibility with the problems of this nature. This is a schematic representation of the DeepVARHierarchical and the way the data is handled in the methodology;

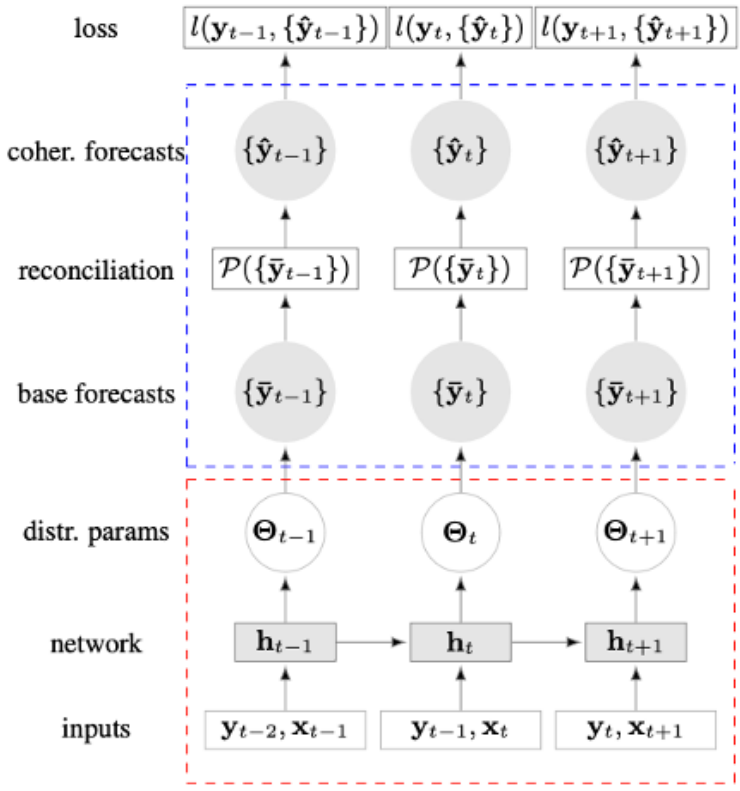

Figure 3.2: Specific instantiation of the approach with DeepVAR (Salinas et al., 2019) multivariate forecasting model (red boundary). Sampling and projection steps are highlighted by the blue boundary [50]. Source: Rangapuram et Al. 2021 [47]

In Figure 3.2 the red dashed line represents the multivariate forecasting model DeepVAR (described below)

and the blue dashed line highlights the sampling and projection steps. Once trained, the model produces coherent forecasts by construction [50] [47].

3.1 DeepVAR

DeepVAR is a multivariate, nonlinear generalization of classical autoregressive models (Salinas et al., 2019; 2020; Alexandrov et al., 2019)[50] [5]. It uses a recurrent neural network (RNN) to exploit relationships across the entire history of the multivariate time series and is trained to learn the parameters of the forecast distribution. More precisely, given a feature vector $\mathbf{x}_t$ and the multivariate lags $\mathbf{y}_{t-1} \in \mathbb{R}^n$ as inputs, DeepVAR assumes the predictive distribution at time step t is parameterized by Θ_t, which are the outputs of the RNN, also known as "Learned parameters":

$$\Theta_t = \Psi(\mathbf{x}_t, \mathbf{y}_{t-1}, \mathbf{h}_{t-1}; \Phi) \tag{3.1}$$

where Ψ is a recurrent function of the RNN whose global shared parameters are given by Φ and hidden state by $\mathbf{h}_{t-1}$ [47]. Typically, DeepVAR assumes that the forecast distribution is Gaussian in which case $\Theta_t = \{\mu_t, \Sigma_t\}$, where $\mu_t \in \mathbb{R}^n$ and $\Sigma_t \in \mathbb{S}_n^+$, although it can be extended to handle other distributions. The unknown parameters Φ are then learned by the maximum likelihood principle given the training data [47]. Note that for simplicity only one lag $\mathbf{y}_{t-1}$ is specified as the input to the recurrent function but in the implementation, lags are chosen from a lag set determined by the frequency of the time series [5].

In the hierarchical setting, the covariance matrix Σ_t captures the correlations imposed by the hierarchy as well as the relationships among the bottom-level time series. It is often found in industrial applications that the bottom-level time series are too sparse to learn any covariance structure let alone more complicated nonlinear relationships between them [47]. Given this, it is proposed by (Rangapuram et al. 2021) to learn a diagonal covariance matrix Σ_t when producing the initial base forecasts; also if more flexibility is needed to capture the nonlinear relationships then one could transform base forecasts using normalizing flows. The linear relationships between the aggregated and bottom-level time series are enforced via projection. Although we assume Σ_t is diagonal, this is not equivalent to learning independent models for each of the n time series in the hierarchy. In fact, the mean $\mu_{t,i}$ and the variance $\Sigma_{t,(i,i)}$ of the forecast distribution for each time series are predicted by combining the lags of all time series $\mathbf{y}_{t-1}$ and features $\mathbf{x}_t$ in a nonlinear way using shared parameters Φ [47].

3.2 Sampling and Projection

In broader terms, probabilistic coherence is defined as any forecast that assigns zero probability to events that do not meet the coherence condition [44]. In the same way that point forecast reconciliation begins with an incoherent forecast, in the probabilistic setting we begin with an incoherent probabilistic forecast. In the point forecasting setting, we can consider a (usually linear) function that takes an incoherent point and maps it to a coherent point. In the probabilistic setting, we consider the same types of functions but think about them as mapping sets of incoherent points to sets of coherent points. The probabilities assigned to these two sets are the same, giving us a general definition of the probabilistic forecast reconciliation coherence condition. The key

implication of this definition is that any existing point reconciliation method (e.g., OLS or MinT) can be extended to the probabilistic setting [44].

To generate coherent forecasts given distribution parameters $\Theta_t = \{\mu_t, \Sigma_t\}$ from the RNN, the method initially includes generating a set of N Monte Carlo samples from the predicted distribution, $\{\bar{y}_t \in \mathbb{R}^n\} \sim \mathcal{N}(\mu_t, \Sigma_t)$. As mentioned, the sampling step is differentiable with a simple reparameterization of $\mathcal{N}(\mu_t, \Sigma_t)$:

$$\bar{y} = \mu + \Sigma^{1/2} z, \quad \text{with} \quad z \sim \mathcal{N}(0, I). \tag{3.2}$$

[47]

That is, given the samples from the standard multivariate normal distribution, which are independent of the network parameters, the actual forecast samples are deterministic functions of μ_t and Σ_t.

Coherence is enforced on the transformed samples $\{\bar{y}_t\}$ obtained from the forecast distribution by solving the following optimization problem:

$$\hat{y}_t = \arg\min_{y \in \mathbb{R}^n} \quad ||y - \bar{y}_t||^2 \quad \text{s.t.} \quad A\mathbf{y} = 0. \tag{3.3}$$

[47]

Note that this is essentially a projection onto the null space of A which can be computed with a closed-form projection operator:

$$\mathbf{M} := \mathbf{I} - A^\top (AA^\top)^{-1} A. \tag{3.4}$$

In other words, we have:

$$\hat{y}_t = \mathbf{M}\bar{y}_t \in \mathcal{S}. \tag{3.5}$$

[47]

It should be noted that $\mathbf{A}\mathbf{A}^\top$ is invertible for the hierarchical setting. $\mathbf{M}$, which is time-invariant, can be computed offline once, prior to the start of training. In principle, the projection problem (3.5) can accommodate additional convex constraints. Although this precludes the possibility of a closed-form solution, the projection can be implemented with a differentiable layer within the DCL framework [4].

3.3 Training and Prediction

The training of the hierarchical forecasting model is similar to DeepVAR except that the loss is directly computed on the coherent predicted samples [47]. Given a batch of training series $\mathbf{Y} := \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T\}$, where $\mathbf{y}_t \in \mathbb{R}^n$, and associated time series features $\mathbf{X} := \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$, the likelihood of the shared parameters Φ is given by

$$l(\theta) = p(\mathbf{Y}; \mathbf{X}, \theta) = \prod_{t=1}^{T} p(\mathbf{y}_t | \mathbf{y}_{t-1}; \theta), \tag{3.6}$$

[47]

where Θ_t are the distribution parameters 3.1 predicted by the DeepVAR model.

In the hierarchical setting, the learnable parameters are still given by Φ but the model outputs coherent Monte Carlo samples $\{\hat{\mathbf{y}}_t\}$ at each time step t. We are then able to compute sufficient statistics $\Theta_{c,t}$ on $\{\hat{\mathbf{y}}_t\}$ and define the following likelihood model:

$$l_c(\theta) = \prod_{t=1}^{T} p(\mathbf{y}_t; \Theta_{c,t}). \tag{3.7}$$

[47]

The exact distribution $p(\mathbf{y}_t; \Theta_{c,t})$ can be chosen according to the data. One can then maximize the likelihood of estimating the parameters Φ. More importantly, it has the flexibility to estimate the parameters Φ by optimizing any other loss function such as quantile loss, continuous ranked probability, or any of the metrics typically preferred in the forecasting community. This is possible because any quantile of interest can be computed given sufficiently many samples (N large enough)[47].

Prediction is performed by unrolling the RNN step-by-step over the prediction horizon as shown in Figure 3.3 [50].

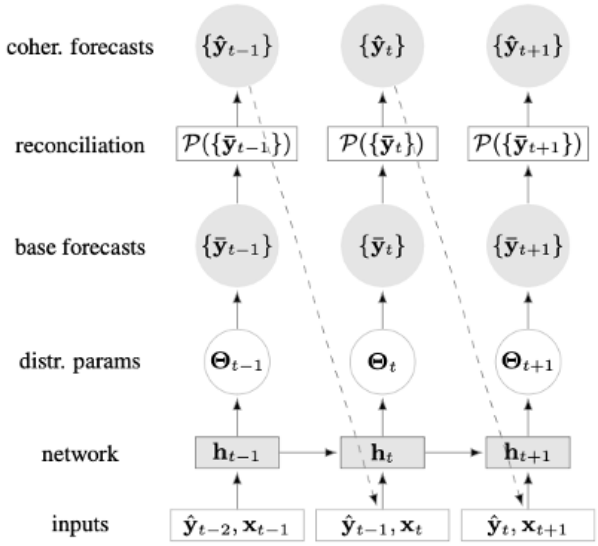

Figure 3.3: DeepVARHierarchical training and prediction.Source: Rangapuram et Al. 2021 [47]

Given an observed hierarchical time series $\{y_1, y_2, \ldots, y_T\}$, there is the need to predict its values for τ subsequent periods. Starting with $t = T + 1$, we obtain forecast distribution parameters Θ_{T+1} by unrolling the RNN for one time step using the last hidden state from training h_T, time series features $x_{1:T+1}$ and the observed lag values $y_{t-1}, t = 2, 3, \ldots, T + 1$. Then generate a set of sample predictions $\{\hat{y}_{T+1}\}$ by first taking Monte Carlo samples from parameters Θ_{T+1} and then projecting them with the same matrix $\mathbf{M}$ used in training. For each $t > T + 1$, a sample predicted in the previous step $\hat{y}_{t-1}$ is used as the lag input, shown as the dotted line in Figure 3.3, to generate prediction $\hat{y}_t$. We repeat this procedure for each of the N samples generated at the beginning of the prediction horizon $T + 1$. This way, a set of sample paths was obtained $\{\hat{y}_{T+1}, \ldots, \hat{y}_{T+\tau}\}$ that is coherent when the end of the prediction horizon is reached. These samples may then be used to generate point (mean) or probabilistic

forecasts by computing appropriate sample statistics (e.g., quantiles) [47].

The continuous ranked probability score (CRPS) is used to evaluate the accuracy of our forecast distributions, details of which can be found in Chapter 2.

4

End-to-End methodology in forecasting Demand for Supply Chain Management

Forecasting is an essential process in many fields but its importance is highlighted in the context of business, economics, and finance. Market research, expansion plans, revenue projections, cash flow management, inventory, sales, and risk management, are only a few departments in which forecasting is utilized frequently. It is difficult to find any department in big companies that do not use, generate, or leverage the forecasting results. One of the key business units that allow a business to run smoothly is Supply Chain Management, and accurately forecasting SC flows as Demand, Arrivals, or Transfer-In has a cardinal impact on the overall performance of a company.

In particular, Demand forecasting is a crucial process in supply chain management that predicts future demand for a product, enabling supply chain operations to be planned in order to reduce delivery times, stock levels, and operating costs. SCM needs to offer the right product at the right place, time, and price to satisfy customers' expectations. Accurate forecasting minimizes uncertainty, stabilizes the supply chain, increases financial savings, and enhances competitiveness.

On the other side, incorrect predictions can lead to excessive expenditures on procurement, shipping, human resources, service level, and inventories. However, demand forecasting is complex due to customer behavior fluctuations, economic growth, and advances in technology, making it difficult for organizations and predictors to conduct scientific forecasts [14]. Many firms know their predictions are unrealistic but lack the knowledge to fix them, leading to ignoring the issue and hoping to solve it later. Thus, improving demand forecasting accuracy and methodologies is critical for companies and supply chains [17].

4.1 The Problem

Demand Planning and Forecasting was one of the primary processes that were utilized in the process of developing Supply Chain Plans while I was doing my internship at Amazon. As far as the remainder of the SC plan was concerned, this section was the basis of the rest of the process. For example, it was responsible for defining transportation, employment, capacity utilization, warehouse placements, and everything else in the middle up to deliver a package to the final customer.

In Amazon, the forecast was developed using a hierarchical framework that was constructed with four tiers. On the highest level, which was the most aggregated, there was the so-called "network demand," which might encompass the demand for either Europe, the United Kingdom, North America, or any number of other territories. In the following step, each of these networks was broken down into "clusters," which included countries such as France, Italy, Spain, and others. These clusters showed geographically distinct groups of warehouses or "fulfillment centers" (FC), which were used to construct the third level. All of the product forecasts were located at the leaf nodes, which were the lowest level.

Our team's primary objective was to automate supply chain activities, but even for us, the results of demand forecasts were a core result in most of our products. However, there was a vast amount of time and work that was required from many different teams and assessments to arrive at the final demand projection, which proved to be a significant impediment in SCM. During that time, efforts were made to devise strategies that would improve the effectiveness of this process, since it was acknowledged that this stage had a substantial impact. Therefore, it was essential to make certain that accurate forecasts were made but also to lower costs significantly in terms of both time and labor.

In light of these considerations, it is apparent that there is a strong necessity for the implementation of an End-to-End method that is capable of generating these forecasts at all levels without the requirement of a separate reconciliation procedure. With the target of demand forecasting in supply chain management, this Chapter is going to demonstrate how the 'End-to-End Learning of Coherent Probabilistic Forecasts for Hierarchical Time Series' methodology may be put into practice.

4.2 Dataset and Methodology

The methodology implementation followed the setup details outlined by the authors of the End-to-End paper [47]. However, due to data sensitivity considerations, it was not feasible to use Amazon-specific data in this instance. Instead, in collaboration with my manager at Amazon, the methodology was applied to a publicly available dataset exhibiting behavior similar to real demand data.

The dataset, curated by Syama Sundar Rangapuram in 2022, is accessible at the following link: `https://gist.github.com/rshyamsundar/39e57075743537c4100a716a7b7ec047/`.

The summation matrix associated with the dataset can be found here: `https://gist.githubusercontent.com/rshyamsundar/17084fd1f28021867bcf6f2d69d9b73a/raw/`. Notably, the values in the dataset are normalized, likely as a result of anonymization and privacy standards.

The summation matrix **S** of this dataset is:

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

The dataset comprises 336 data points, recorded on two weeks, in hourly frequency. The dataset is formed into a format of *Gluon TS* by using the bottom level series and the summation matrix, which shows how the data should be aggregated into the higher levels. Check Figure 4.1 for an overview of the dataset.

	EU	Italy	Spain	Node1	Node2	Node3	Node4
2020-03-22 00:00	0.686671	0.156873	0.529798	0.056962	0.099911	0.039827	0.489971
2020-03-22 01:00	2.189128	0.669261	1.519866	0.246535	0.422727	0.763164	0.756702
2020-03-22 02:00	1.152853	0.582213	0.570640	0.314393	0.267820	0.169645	0.400996
2020-03-22 03:00	1.198889	0.653139	0.545750	0.609158	0.043981	0.235009	0.310741
2020-03-22 04:00	2.069197	0.678490	1.390707	0.380788	0.297702	0.898429	0.492278
...	...	...	...	...	...	...	...
2020-04-04 19:00	2.386645	1.351527	1.035118	0.708064	0.643463	0.716753	0.318365
2020-04-04 20:00	1.880544	0.743583	1.136961	0.218748	0.524835	0.938315	0.198646
2020-04-04 21:00	2.166601	0.611360	1.555241	0.341728	0.269632	0.567782	0.987459
2020-04-04 22:00	1.216177	0.436552	0.779625	0.387454	0.049097	0.745593	0.034032
2020-04-04 23:00	2.395706	1.165140	1.230566	0.664046	0.501094	0.839132	0.391434

336 rows × 7 columns

Figure 4.1: The dataset that was used to test the adaptability of the DeepVARHierarchical on forecasting demand in the SC context.

The data is structured in a three-level hierarchical structure, with two nodes on the second and four nodes at the bottom level, which accounts for seven time series in total (EU, Italy, Spain, Node1, Node2, Node3, Node4). The top node represents New Workable Demand (NWD) within the European Union's network. In this simplified reality version, the demand of the EU will be further broken down into two clusters, Italy and Spain, which constitute the second tier of the hierarchy. Each of these national segments is then divided into two additional sub-segments, forming the third tier, which represents the FCs (warehouses). A visual representation of this structure can be found in Figure 4.2.

The general code base for the method is found in the *Gluon TS* library [5]. The hyperparameters are also left at the default values set by the authors[47]. The details of the process that the End-to-End methodology follows may be found in Chapter 3.

In the experimental setup, the following hyperparameters are used:

- Prediction length: 24 hours
- Number of epochs: 10
- Learning rate: 1×10^{-3}
- Batch size: 32

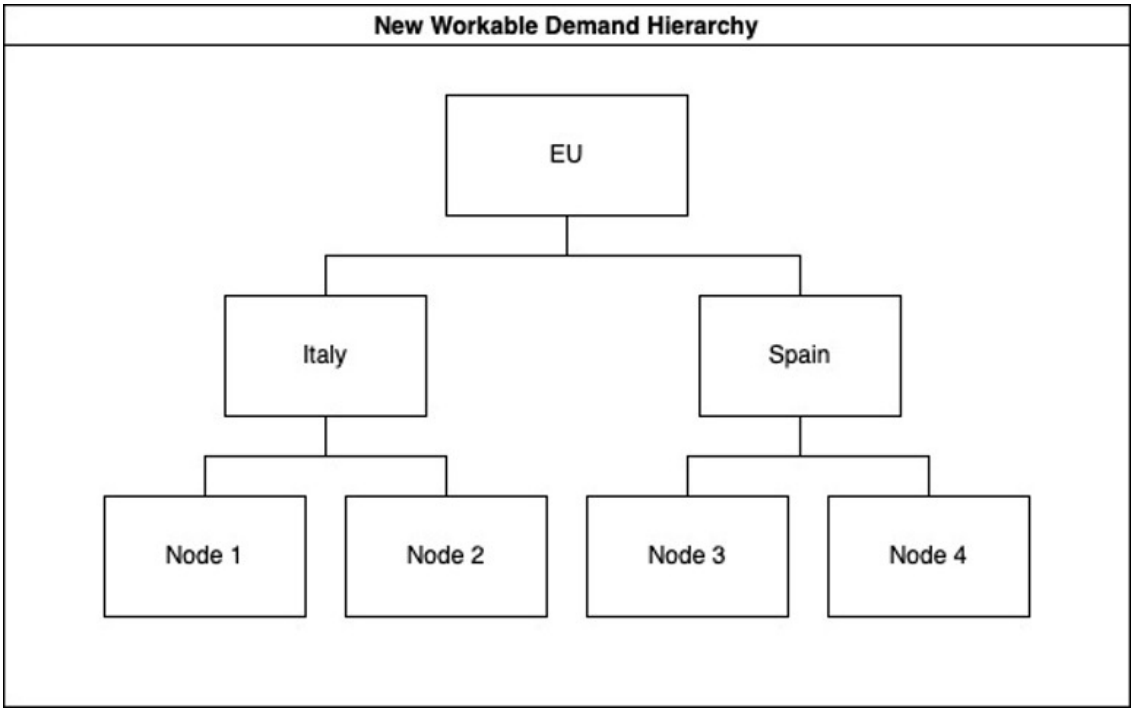

Figure 4.2: Hierarchy of the New Workable Demand from an Amazon example.

4.3 Exploratory Data Analysis

An initial exploratory data analysis was performed on the data to get a better comprehension of the dataset at hand. In Table 4.1, count, mean, standard deviation, and quantiles of the time series can be found. The analysis confirms that all the time series have the same count. Keeping in mind that the values are normalized the minimum value on the leaf node is 0 while the maximum is 1, with a similar mean and standard deviation.

The relationship between the time series was checked through the correlation matrix found in Figure 4.3. A mask is used to hide the upper triangle of the heatmap to avoid duplicating information since the correlation matrix is symmetrical.

	count	mean	std	25%	50%	75%
EU	336.0	1.986023	0.613545	1.554246	2.000046	2.408383
Italy	336.0	1.000742	0.416350	0.685174	0.998668	1.289633
Spain	336.0	0.985281	0.422882	0.645015	0.976069	1.292093
Node1	336.0	0.522583	0.295102	0.246390	0.529223	0.798044
Node2	336.0	0.478159	0.281454	0.251101	0.463864	0.730812
Node3	336.0	0.509392	0.301991	0.236183	0.524774	0.765462
Node4	336.0	0.475889	0.284407	0.235621	0.456849	0.721892

Table 4.1: Statistical summary of the time series.

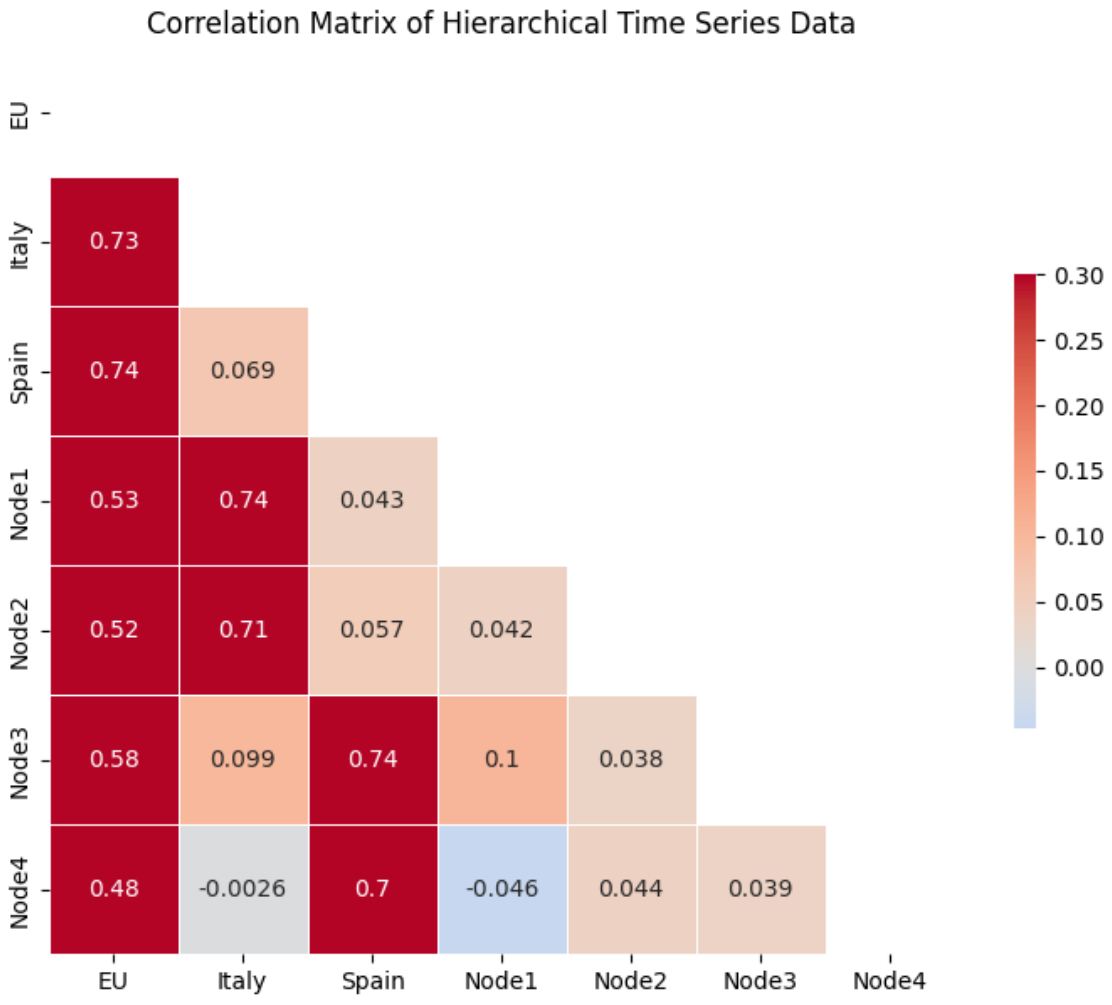

Figure 4.3: Correlation matrix of the Hierarchical Dataset

In the correlation matrix, the hierarchical structure is shown again, as we see the high correlation on the parent-node connected time series. While the leaf nodes have little to no correlation between them, Node 1 and Node 2 have a strong correlation to Italy, while Node 3 and Node 4 correlate with Spain. As expected, each of the other time series is correlated to the EU, but this correlation falls moving down the hierarchy. Showing that hierarchy levels that are not subsequent are less correlated.

Next, the distributions of the time series were graphed. From the distributional graphs in Figure 4.4, it is noticeable that the bottom level graphs do not show a clear distribution, with Nodes 1 and 3 appearing to be a somewhat bimodal shape and Nodes 2 and 4 slightly skewed to the right and left respectively. As it is known in a business context the leaf nodes usually do not contain a lot of clear behavior and their forecasts are used more for operational decision making. As we can see from the graphs, moving up in the hierarchy the data goes to a more Gaussian distribution. The higher levels of hierarchy are usually the ones that drive strategic decisions in the company. Further, the End-to-End methodology also assumes that the forecasted data will follow a Gaussian distribution but as we see this might not be always the case.

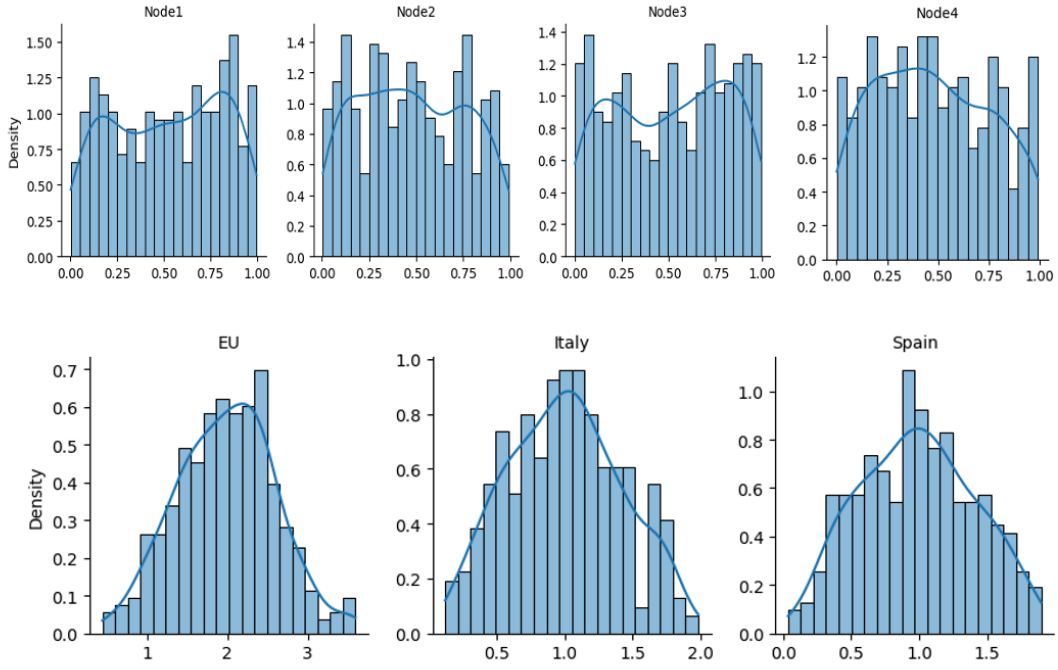

Figure 4.4: Distributions of the seven time series included in the dataset.

Finally, each time series were plotted to see if there was any visible behavior. In Figure 4.5 the blue line represents the actual hourly data points for the respective timeseries, the orange line depicts the 24-hour moving average, which smooths out the short-term fluctuations and highlights longer-term trends or cycles, and lastly the gray shaded area indicates the 24-hour moving standard deviation, which provides a sense of the variability or dispersion of the data around the moving average.

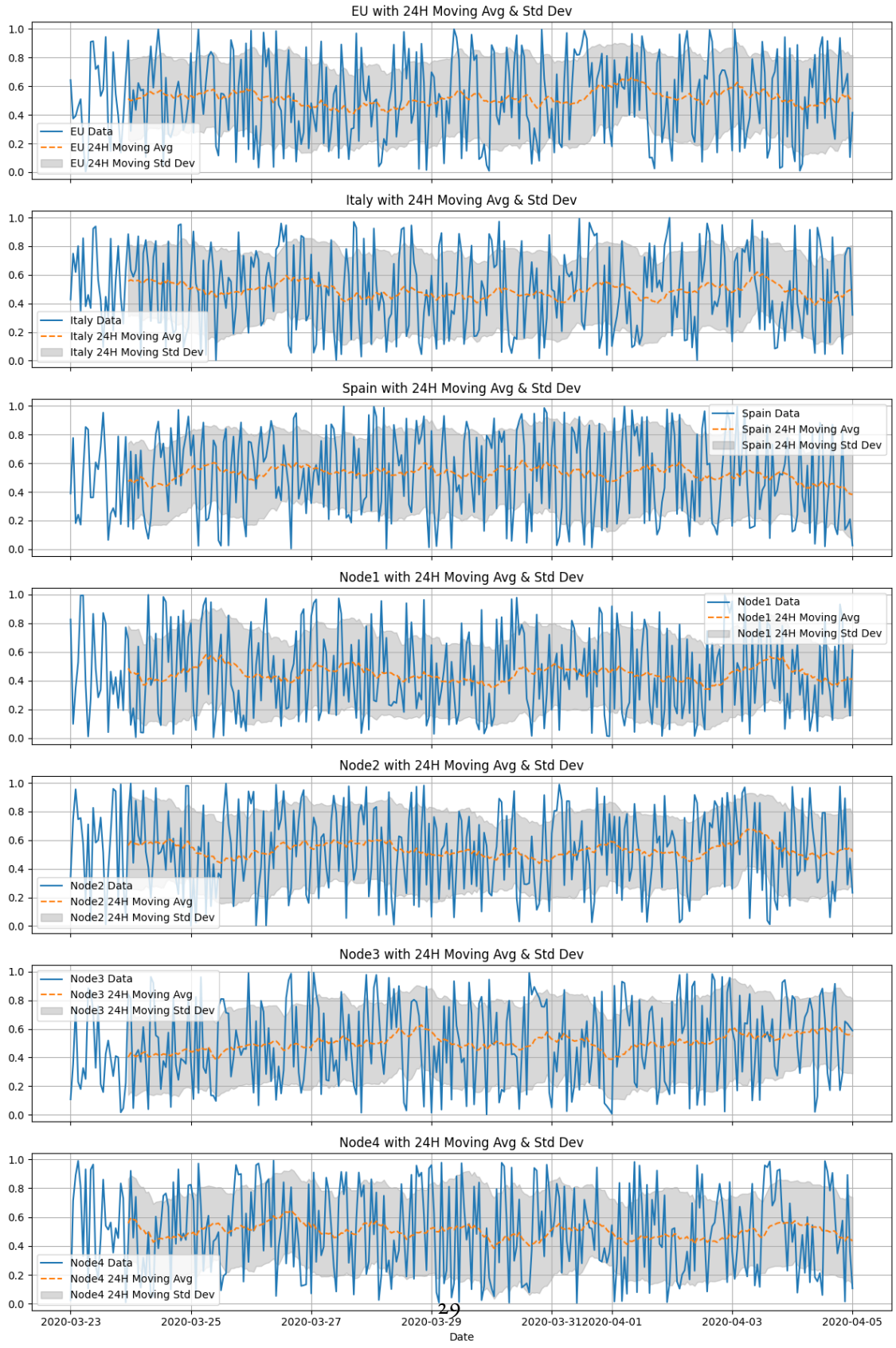

Figure 4.5: Line graphs of the time series.

We can say that in all the time series the values have a consistent variability, showing no clear trend, seasonality, or periodic patterns. No outliers were spotted even though the data contains occasional spikes. By performing the Augmented Dickey-fuller (ADF) test it was confirmed that all the series are stationary. No autocorrelation was found either.

4.4 Forecast Results

The forecast horizon is equal to the lead time of the decisions driven by the forecast [43]. In this case, considering the volume and the frequency of the data, the DeepVARHierarchical model generates forecasts that predict the range of possible outcomes for the upcoming 24 hours. These results include the mean (point) forecast as well as a bandwidth that shows where the lower 10% and upper 90% of outcomes are expected to fall, providing a probabilistic view of future values. It is worth remembering that the model takes into account the entirety of the time-series data to learn not only the patterns within each individual series but also the relationships between them. Another key feature of these results is that they already represent a probabilistically coherent forecast, meaning they respect the linear constraints dictated by the hierarchical data structure.

Moreover, Mean Squared Error (MSE), Absolute Error (AE), Mean Absolute Percentage Error (MAPE), Mean Absolute Quantile Loss (MAQL), and Mean Weighted Quantile Loss (MWQL), were used to provide an overview of the forecast performance in each level from which can then compare the model performance in different sections of the hierarchy.

Top-level: EU

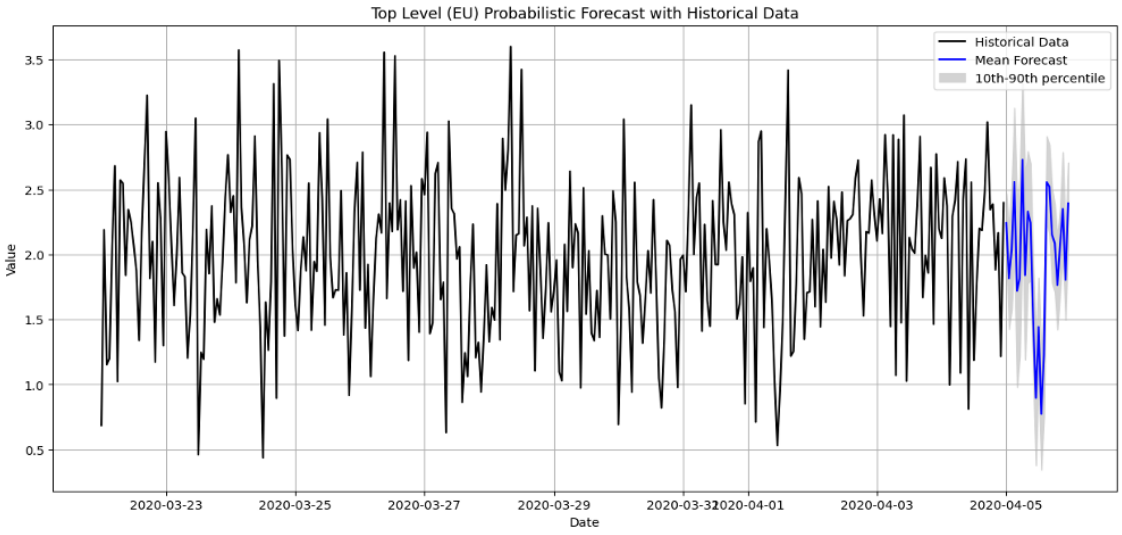

Figure 4.6: Forecasting result for the EU level demand.

In Figure 4.6 the black line represents the historical data (observed values) for the EU level, while the solid blue line is the mean forecast or point forecast. The shaded area represents the 10th-90th percentile range of the probabilistic forecast. From the graph, we can say that the model has captured the behavior of the data, which is also confirmed by the metrics in Table 4.2 – indicating a good fit and reliable probabilistic forecast.

Metric	MSE	AE	MAPE	MAQL	MWQL
EU	0.1533	7.1950	0.1979	5.6138	0.1115

Table 4.2: Evaluation metrics for the EU model.

Second level: Italy and Spain

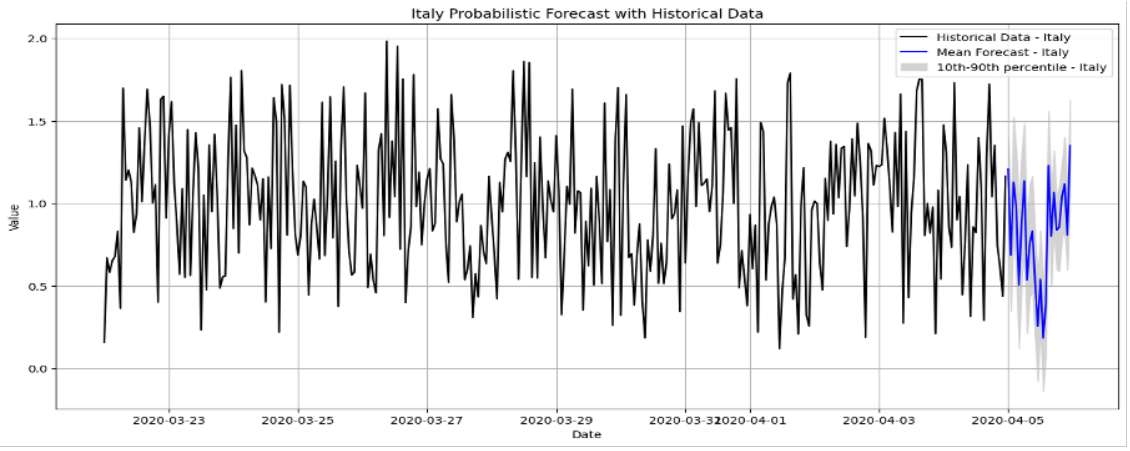

Figure 4.7: Forecasting result of the SC demand of the second level of the hierarchy - Italy.

The same graphic representation of the results is used for the other levels also. In Figures 4.7 and 4.8 both countries' forecasts show a similar pattern of historical data, with variability and no clear long-term trends. Italy's forecasts are generally more accurate than Spain's across all metrics provided. The probabilistic forecast ranges visually appear to capture the variability in the historical data well, with the actual data mostly residing within this range. Business operations in these markets would adjust their strategies accordingly, taking advantage of the predictability where possible and guarding against uncertainty where necessary, especially for Italy where the forecast is quite accurate, suggesting that decision-making for this region can be made with a higher degree of confidence. The metrics for these two forecasts can be found in Table 4.3.

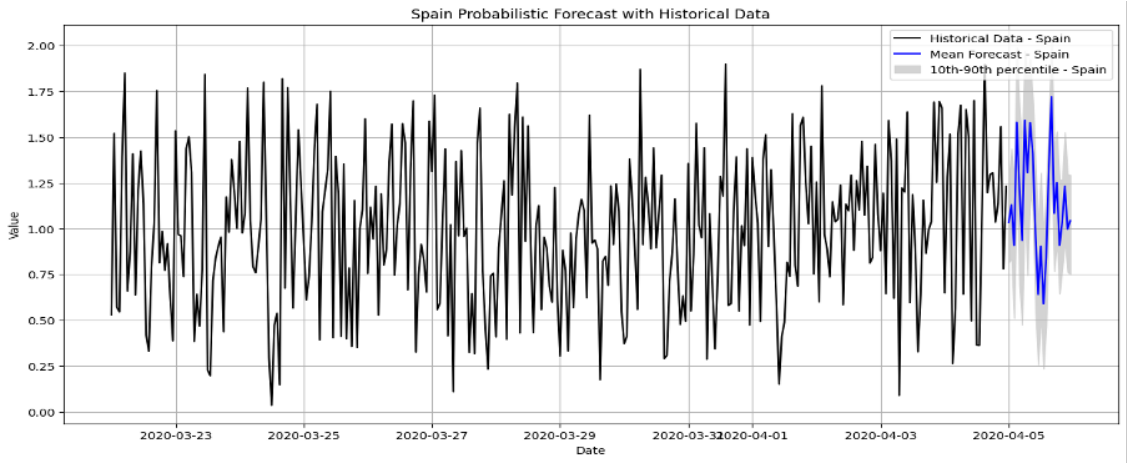

Figure 4.8: Forecasting result of the SC demand of the second level of the hierarchy - Spain.

Metric	MSE	AE	MAPE	MAQL	MWQL
Italy	0.0669	4.8850	0.3172	3.8554	0.1632
Spain	0.1587	7.8433	0.4197	6.2931	0.2356

Table 4.3: Evaluation metrics for the models in Italy and Spain.

Third level: Nodes 1-4

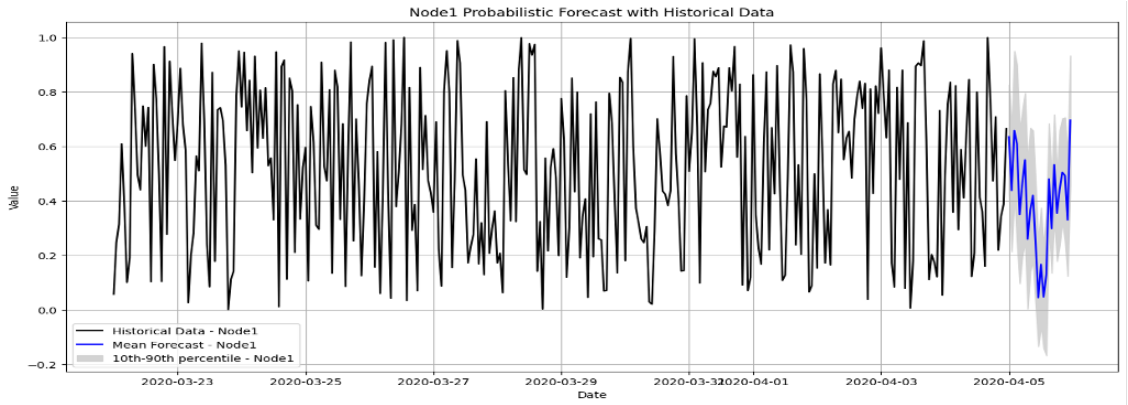

Figure 4.9: Forecasting result of the SC demand of the third level of the hierarchy - Node 1.

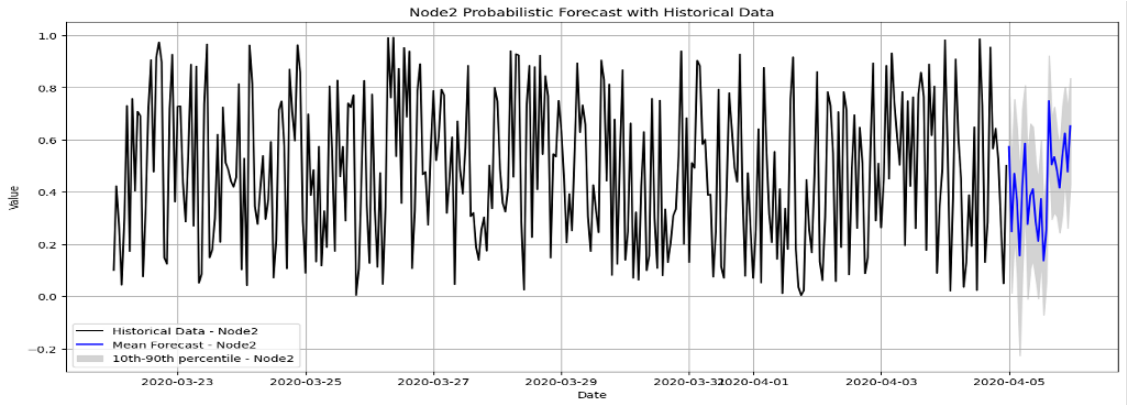

Figure 4.10: Forecasting result of the SC demand of the third level of the hierarchy - Node 2.

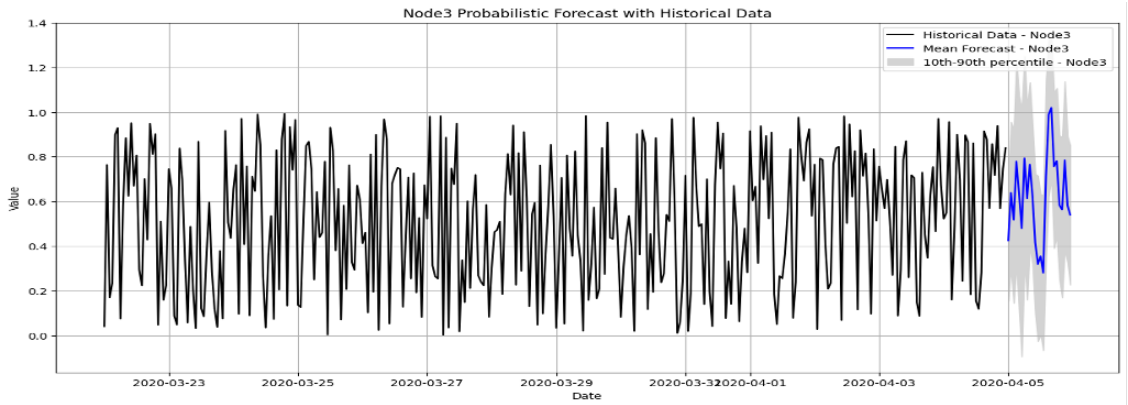

Figure 4.11: Forecasting result of the SC demand of the third level of the hierarchy - Node 3.

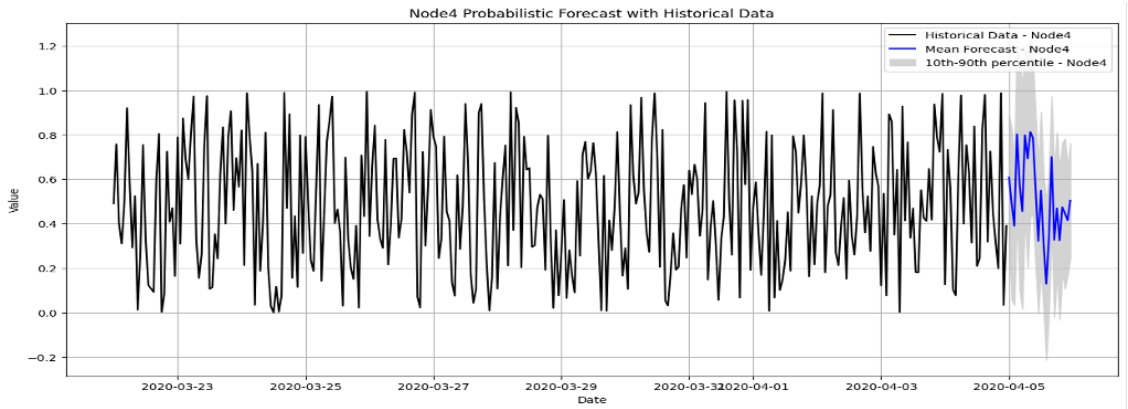

Figure 4.12: Forecasting result of the SC demand of the third level of the hierarchy - Node 4.

All nodes show a good MSE and AE which implies a good fit on the point forecast, but with a relatively high MAPE indicating that percentage errors may be significant. Quantile losses are moderate, suggesting reasonable distributional forecast accuracy, which is also backed up by the MAQL and MWQL values, refer to Table 4.4. Either way, forecasts for Node 2 and Node 4 are less reliable, which may require more conservative strategies or contingency plans. In Figures 4.9, 4.10, 4.11, and 4.12 the visual representation of these forecasts can be appreciated.

Metric	MSE	AE	MAPE	MAQL	MWQL
Node 1	0.0583	4.8207	0.5438	3.7794	0.2985
Node 2	0.0550	4.6742	2.4177	3.6694	0.3348
Node 3	0.0654	5.0784	0.5581	3.8584	0.2590
Node 4	0.0693	5.3299	1.0611	4.0026	0.3388

Table 4.4: Evaluation metrics for Nodes 1 to 4.

All Levels

The overall CRPS for the hierarchy is **0.2228** (Details in Chapter 2, under Proper Scoring). The model has room for improvement which can be done by tuning the hyperparameters rather than using the autotuned parameters, and potentially increasing training epochs. Nonetheless, for the sake of this implementation, it was aimed to present what the model is capable of doing in its basic mode. From the metrics below, we notice that the model had a good performance overall being able to catch the behavior of the series, but not in every level the model performed similarly. This was also evident in the End-to-End paper, which may require more attention in a business decision-making context. In Table 4.5 the average (or summed) metrics for all levels are represented.

Metric	MSE	AE	MAPE	MAQL	MWQL
All Levels	0.0896	39.8268	0.7879	31.0723	0.2058

Table 4.5: Evaluation metrics at all levels.

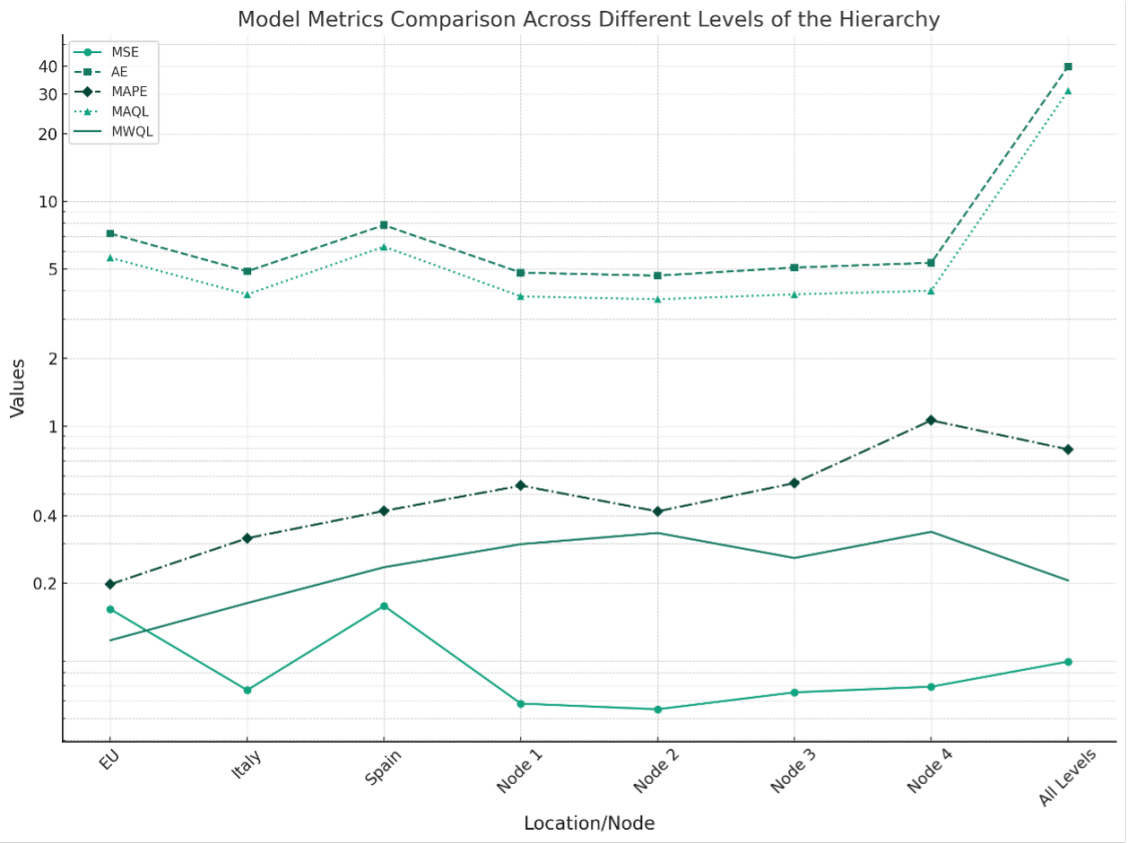

Figure 4.13: Model Metrics Comparison Across Different Levels of the Hierarchy.

The line chart in Figure 4.13 displays the metrics of the model across different hierarchy levels. MSE is generally low across all levels with EU and Spain having the highest error, which also corresponds with MAQL indicating the lower performance in these two levels of the point and probabilistic forecast when we look at each level independently. On the other hand, MAPE and MWQL show an increasing trend going from the top of the hierarchy to the lower levels which confirms the business experience that the lower levels will be generally sparser, and these forecasts will contribute more in operational decision-making, rather than strategical ones.

4.5 Comparing End-to-End and ARIMA-BU Methods

To provide a benchmark of the performance of the End-to-End methodology, the Autoregressive Integrated Moving Average (ARIMA) model including a Bottom-up Reconciliation (check Chapter 2 for details), was implemented to the dataset [16]. This implementation is twofold. First, it starts by using ARIMA to achieve a base forecast of the bottom level, particularly *auto.arima* function in *R* was used, which automatically chooses the best ARIMA configuration for the data at hand [35]. In Figure 4.14, the forecasts for Nodes 1-4 can be found. Without going into the metrics of these forecasts, it is visible that ARIMA could not capture properly the behaviour of this data. An assumption that can be made is that the current dataset might be too variable for ARIMA to capture the pattern and an increase of the data volume might be helpful.

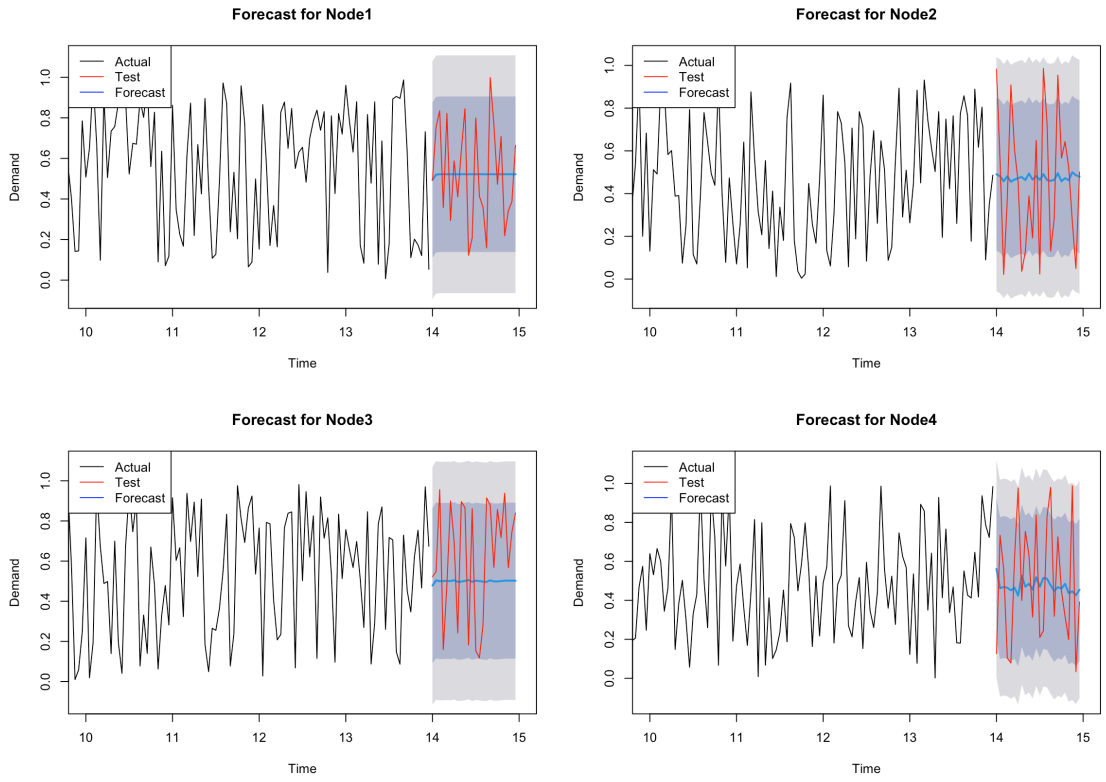

Figure 4.14: ARIMA forecasts for Nodes 1-4. The black lines represent the historical training data; The red lines represent the test data; The blue lines represent the forecasts generated by ARIMA. Lastly, the purple and the gray areas show the 80% and 95% probabilistic forecast bands respectively.

Next, the ARIMA forecasts and the reconciled forecasts ARIMA-Bottom-up were obtained for the second and first levels of the hierarchy. In Figure 4.15 the forecasts of Italy, Spain, and the EU are presented. The forecasts are better than the forecasts for the Nodes but still, the essence of the behavior and the variability of the data is not captured.

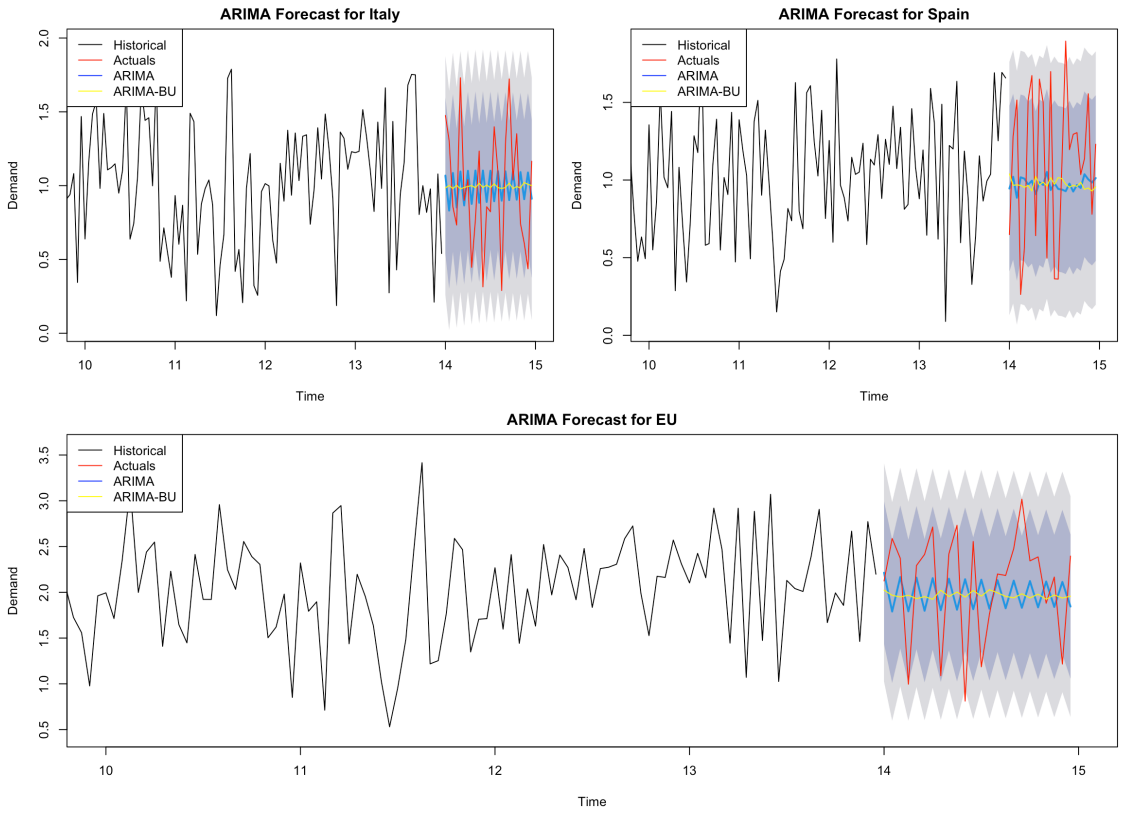

Figure 4.15: ARIMA forecasts for Italy, Spain, and EU. The black lines represent the historical training data; The red lines represent the actual test data; and The blue lines represent the forecasts generated by ARIMA. The yellow lines represent the forecasts generated by ARIMA-Bottom-up. Lastly, the purple and the gray areas show the 80% and 95% probabilistic forecast bands respectively.

To dive deeper into the metrics of these forecasts, with the intention of having an exact comparison, in Table 4.6, MSE, AE, MAPE, MAQL, and MWQL values are shown for the three methodologies. It should be noted that since ARIMA does not typically produce quantile forecasts, the MAQL and MWQL values for ARIMA and ARIMA-BU are not included.

Method	Metric	MSE	AE	MAPE	MAQL	MWQL
End-to-End	EU	0.1533	7.1950	0.1979	5.6138	0.1115
End-to-End	Italy	0.0669	4.8850	0.3172	3.8554	0.1632
End-to-End	Spain	0.1587	7.8433	0.4197	6.2931	0.2356
End-to-End	Node 1	0.0583	4.8207	0.5438	3.7794	0.2985
End-to-End	Node 2	0.0550	4.6742	2.4177	3.6694	0.3348
End-to-End	Node 3	0.0654	5.0784	0.5581	3.8584	0.2590
End-to-End	Node 4	0.0693	5.3299	1.0611	4.0026	0.3388
ARIMA	EU	0.4157	12.9327	33.0257	-	-
ARIMA	Italy	0.1756	8.3477	50.9420	-	-
ARIMA	Spain	0.2559	10.8025	57.8693	-	-
ARIMA	Node 1	0.0604	5.2021	62.5106	-	-
ARIMA	Node 2	0.0936	6.2413	300.8801	-	-
ARIMA	Node 3	0.0955	6.6621	70.6361	-	-
ARIMA	Node 4	0.0924	6.3987	136.4505	-	-
ARIMA-BU	EU	0.3702	12.7847	32.5035	-	-
ARIMA-BU	Italy	0.1597	8.0323	50.1093	-	-
ARIMA-BU	Spain	0.2521	10.8327	57.7579	-	-

Table 4.6: Combined Evaluation Metrics for End-to-End, ARIMA, and ARIMA-BU.

It is visible from the table that the End-to-End method has superior performance compared to the ARIMA and ARIMA-BU. This is proved first from the point forecast metrics (MSE, AE, MAPE) in which the End-to-End performs better than the other two methods in all the nodes and levels. Secondly, the End-to-End method provides a good probabilistic forecast which is showcased in the MAQL and MWQL and is visible also from the graphs of the results. While the ARIMA probability forecasts shown in the graphs do not seem to provide sharp and well-calibrated forecasts. On the other hand, the ARIMA-BU that is used for obtaining reconciled forecasts for Italy, Spain, and the EU, performs better than the independent ARIMA models. This performance can be attributed to the inclusion of data inter-dependencies from the End-to-End and ARIMA-BU methods.

Figure 4.16 presents a direct comparison of the average performance of the End-to-End, ARIMA, and ARIMA-BU methods across the EU, Italy, and Spain metrics. This visualization uses an average normalized score, calculated by averaging the normalized MSE, AE, and MAPE for each method and region. The figure visualizes again the advantages of the End-to-End method by showing the performance gap.

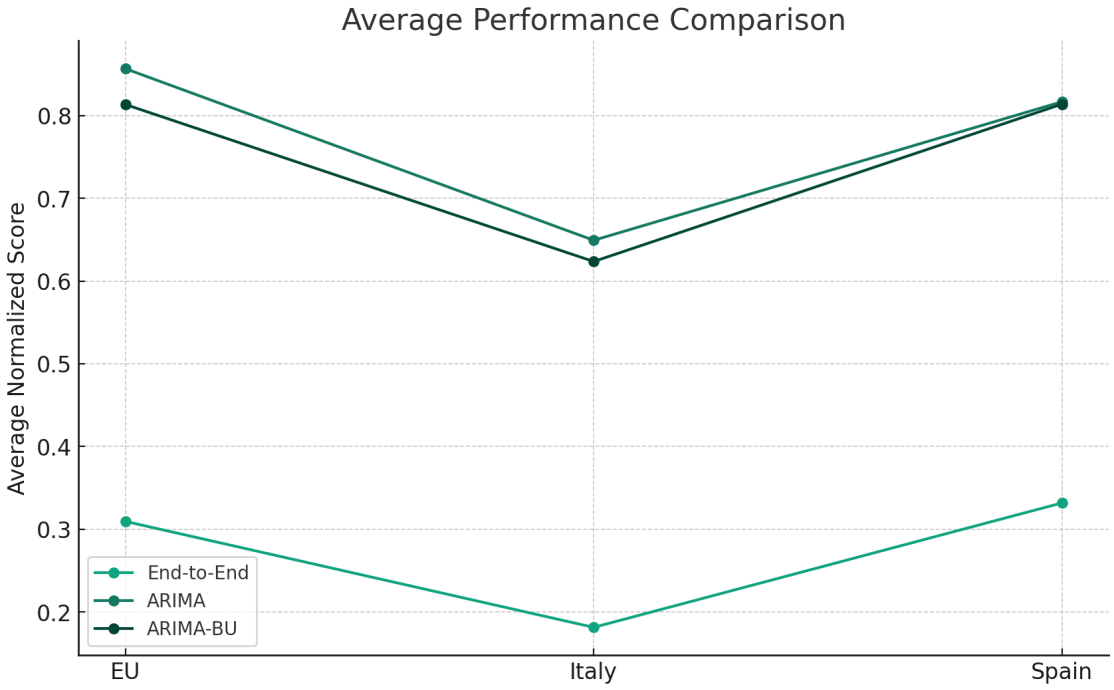

Figure 4.16: Average performance comparison of the MSE, AE, and MAPE, for the three methodologies.

5 Discussion

5.1 Advantages and limitations of the End-to-End approach

The primary advantages of this novel approach, when benchmarked against current state-of-the-art methods, are manifold and significant. Fundamentally, the method is engineered to inherently ensure the generation of coherent and probabilistic forecasts. It streamlines the forecasting process by integrating the reconciliation step directly into the model's architecture, eliminating the need for separate post-processing. By concurrently training on all the time series within a unified nonlinear model, it refines the accuracy for each series, resulting in superior precision. This end-to-end approach leverages the strengths of a multivariate, nonlinear autoregressive model, which simplifies the incorporation of newer multivariate forecasting models, negating the necessity for extensive modifications. The model's architecture is not only flexible in accommodating a variety of loss functions tailored to the application at hand but is also adept at managing more general structural constraints during the projection phase. While DeepVAR traditionally operates under the assumption of a Gaussian forecast distribution, the methodology presented here may be extended beyond this to embrace alternative distributions [47].

Despite the method's robust performance, it is not without its limitations that warrant consideration. The approach, as currently formulated, operates within certain distributional assumptions that, while extendable, may not fully capture the complex nature of real-world data distributions. The method's reliance on deep learning models, particularly Recurrent Neural Networks (RNNs), introduces a degree of opacity that can impede interpretability and explainability. This limitation is consequential; it creates difficulties not only in understanding and communicating the model's outputs to stakeholders who rely on transparency but also in achieving the rigorous requirements of the EU's AI Act, which underlines the need for clear and comprehensible AI systems. The capacity to interpret model outputs is becoming increasingly important in the field, and a lack of it in this method

could prevent greater acceptance and usage, especially in contexts where explainability is as vital as accuracy.

5.2 The Distribution Assumption

Typically, DeepVAR assumes that the forecast distribution is Gaussian in which case $\Theta_t = \{\mu_t, \Sigma_t\}$, where $\mu_t \in \mathbb{R}^n$ and $\Sigma_t \in \mathbb{S}_n^+$, although it can be extended to handle other distributions [47]. The unknown parameters Φ are then learned by the maximum likelihood principle given the training data. In the model at hand, the sampling step is differentiable as long as the distribution chosen allows for a suitable reparameterization where the random "noise" component of the distribution can be separated from the deterministic values of the parameters. This is the case for several distributions including Gaussian, Gamma, log-Normal, Beta, and Student-t [39] [49] [3]. Figurnov et al. (2018) present an alternative approach to compute reparameterization gradients showing broader applicability to Student-t, Dirichlet, and mixture distributions [21]. Although, empirically a multivariate Gaussian distribution performed well on the datasets considered, as future work, it would be very beneficial to explore the usage of nonlinear transformations like normalizing flows to better model non-Gaussian data.

5.3 Model Explainability and Interpretability

For AI methods, the terms interpretability and explainability are commonly interchangeable. It is important to distinguish the difference between them to help organizations determine an AI approach to meet their use case.

Interpretability — If a business wants high model transparency and wants to understand exactly why and how the model is generating predictions, it needs to observe the inner mechanics of the AI method. This leads to interpreting the model's weights and features to determine the given output. However, high interpretability typically comes at the cost of performance, as seen in the following figure. If a company wants to achieve high performance but still wants to have a general understanding of the model behavior, model explainability starts to play a larger role [7].

Explainability — Explainability is how to take an AI model and explain the behavior in human terms. With complex models (for example, black boxes), you cannot fully understand how and why the inner mechanics impact the prediction. However, through model agnostic methods (for example, partial dependence plots, SHapley Additive exPlanations (SHAP) dependence plots, or surrogate models) you can discover meaning between input data attributions and model outputs, which enables you to explain the nature and behavior of the model [7].

When starting a new project, it is needed to consider whether interpretability is required or how explainable your model needs to be. Explainability is essential for most AI systems for achieving several goals. It aims to establish transparency and trust in AI systems by demystifying their internal processes and decision-making mechanisms. Ensures algorithmic accountability by allowing developers, auditors, and regulators to examine the decision-making processes of the models, identify potential biases or errors, and assess their compliance with ethical guidelines and legal requirements[42].

Human-AI collaboration is also facilitated by explainability, as it provides interpretable insights and fosters a mutually beneficial partnership. Human experts can validate AI model decisions against their knowledge and experience, identifying potential errors or biases. Stakeholder communications, which is a crucial part of any

company are made smoother when the counterpart has a clear idea of what is impacting the decision or where are the recommendations coming from. Explainability also facilitates the integration of the Human-in-the-Loop (HITL) system, allowing humans to interact with the AI system, review and interpret its outputs, and provide feedback to refine and improve the model [18].

Fairness and bias mitigation are also addressed by explainability, enabling the detection and mitigation of biases in machine learning models. Certain techniques help identify and understand errors or inaccuracies in machine learning models, allowing developers to debug the models, improve accuracy, and reduce potential risks associated with incorrect decisions [7]. In Figure 5.1, a representation of performance and interpretability trade-offs in different models can be observed.

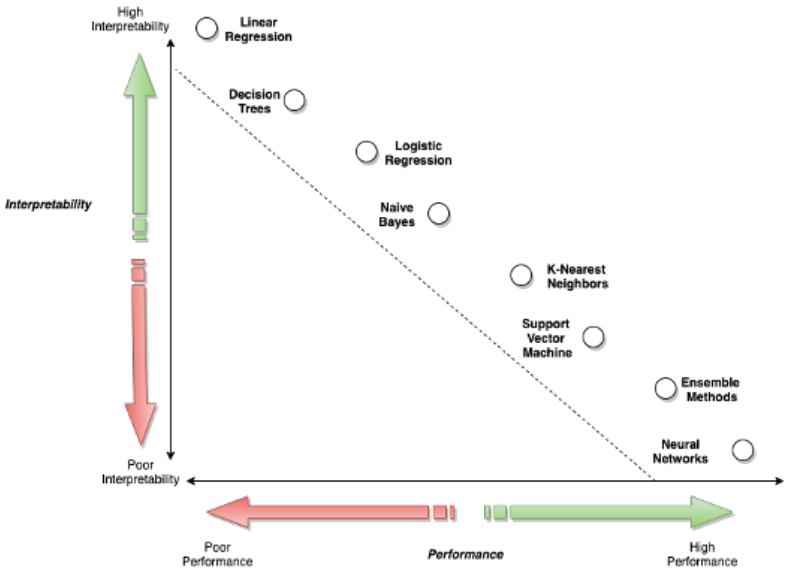

Figure 5.1: Representation of Performance and Interpretability trade-off in different models. Source: Amazon Web Services [7]

5.4 AI Act Compliance

On Friday, December 8, 2023, after months of intensive trialogue negotiations – the European Parliament and Council reached a political agreement on the European Union's Artificial Intelligence Act [45]. AI Act is a new EU regulatory framework for artificial intelligence (AI), which has a top priority to make sure that AI systems used in the EU are safe, transparent, traceable, non-discriminatory, and environmentally friendly. AI systems should be overseen by people, rather than by automation, to prevent harmful outcomes. This groundbreaking legislation aims to address the utilization of AI systems and their associated risks through a risk-based approach. The act categorizes AI systems into four risk levels - unacceptable, high, limited, and minimal risk, each with corresponding regulations and obligations. This pyramid of risks can be found in Figure 5.2. AI applications would be regulated only as strictly necessary to address specific levels of risk [19].

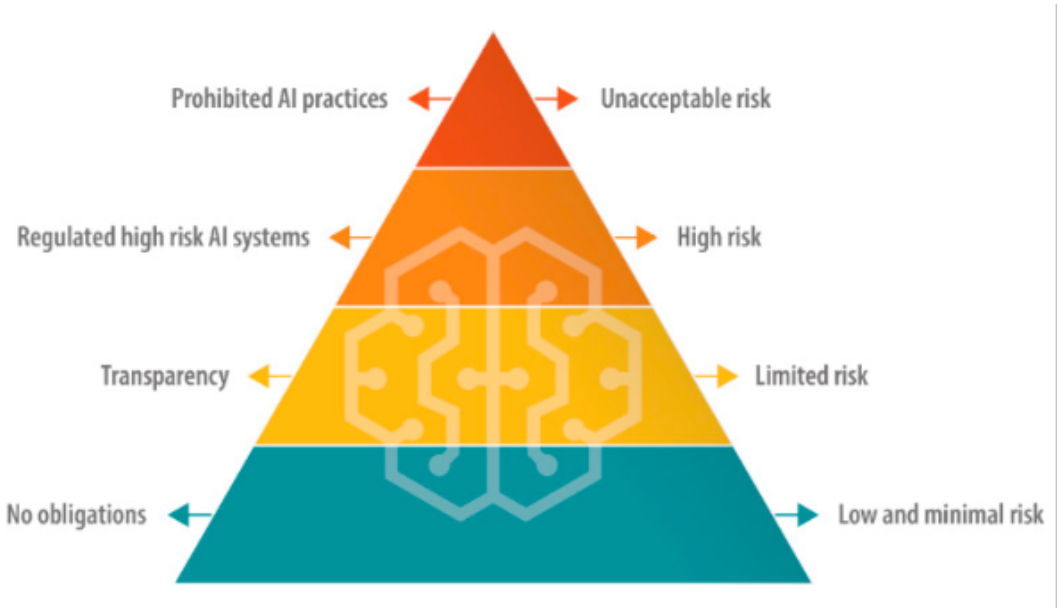

Figure 5.2: Pyramid of risks - used to classify the potential risk in AI system - part of the AI Act, 2023. Source: EU Commission [19]

The proposed AI act aims to ban harmful AI practices that pose a clear threat to people's safety, livelihoods, and rights. It prohibits the use of AI systems that deploy harmful manipulative techniques, exploit vulnerable groups, use by public authorities for social scoring purposes, and use 'real-time' remote biometric identification systems in public spaces for law enforcement purposes [19].

High-risk AI systems are distinguished between systems used as safety components of a product or falling under EU health and safety harmonization legislation. AI systems presenting 'limited risk', such as chatbots, emotion recognition systems, biometric categorization systems, and 'deepfakes', will be subject to a limited set of transparency obligations. All other AI systems presenting low or minimal risk can be developed and used in the EU without conforming to additional legal obligations [19].

What would AI in Supply Chain Management be categorized to? A system that has so much impact most likely would be categorized as a High-risk system or less likely in a Limited-risk system. Both these categories, at the very least, would be subjected to a set of transparency obligations.

In the context of a company such as Amazon, considering the impact on operational significance, critical infrastructure, and wide-reaching consequences, where any mishandling in the model could lead to widespread disruptions, affecting businesses, consumers, and potentially even safety and fundamental rights, a system that forecasts Demand of the SC would be a High-risk system. Independently of the technology at hand, a company would use, the risk is measured on the direct impact that the technology would have on people. Nonetheless, these systems must comply with various requirements, including risk management, testing, technical robustness, data training and governance, transparency, human oversight, and cybersecurity.

In any case, to be able to comply with the new rules that coming up, a system's interpretability and explainability are crucial. The AI Act places a strong emphasis on transparency. It requires these systems to be understandable

and interpretable by users. This means the system's sources, decisions, and the processes leading to these decisions, should be transparent and explainable. If a system's decision-making process is opaque, it becomes challenging to assess its risks accurately and ensure it operates reliably and safely. To effectively oversee an AI system, operators need to understand how it makes decisions. An interpretable and explainable AI system allows for more effective human intervention and decision-making. An interpretable and explainable AI system simplifies the process of demonstrating compliance with the AI Act's requirements and addressing ethical and legal concerns.

Two ways are highlighted in this thesis, on how to tackle this problem. First, a company can work on the explainability of the current methodology, or second, a company can choose an 'explainability-accuracy trade-off' by simplifying the methodology towards more explainable models.

5.5 DeepVAR - RNN Explainability

The convolutional neural networks are not the only deep learning methods that can perform time series classification. The recurrent neural networks, which are perfectly adapted to sequential data types, are also used to accomplish forecasts of time series [48]. DeepVAR is an RNN model chosen by the End-to-End methodology authors to predict the Learning Parameters of the probability forecasts (prior to sampling and projection). Even though the model has been shown to work well and be superior to the state-of-the-art machine learning models which lack explainable elements are classed as being a "black box" and run the risk of perpetuating computer-based discrimination and bias. There are a lot of models that can help in RNN explainability, which differ a lot depending on whether are they ante-hoc or post-hoc, which methodology are they based on (Backpropagation, Perturbation, Attention Mechanism, Fuzzy logic, etc.), model specific or agnostic, scope, and target audience [48].

Post-hoc explainability refers to methods that provide insights into the model's decisions after it has been trained. These are applied to models that are not inherently interpretable. Post-hoc XAI techniques are essential tools that can shed light on these models' decisions, but they do not necessarily change the fundamental nature of the model's interpretability. As part of post-hoc methods, there is also the possibility to explain recurrent models by using a model-agnostic explanation method. Kim et al. [51] use the SHapley Additive exPlanations (SHAP) algorithm [52], a common model-agnostic feature attribution method, to explain the output of a recurrent model [48].

On the other side, Ante-hoc explainability refers to inherently interpretable methods, meaning the model's structure and functioning are designed to be understandable. The transparency of a model and its interpretability are enhanced when ante-hoc methods are incorporated into the architecture, allowing for a certain level of explanation to be built in from the start as part of the model design [48].

Attention mechanisms are Ante-Hoc explainability methods that assign values corresponding to the importance of the different parts of the time series according to the model, see example in Figure 5.3. They are embedded in the structure of recurrent networks and the explicability they offer is available directly at the end of the learning phase.

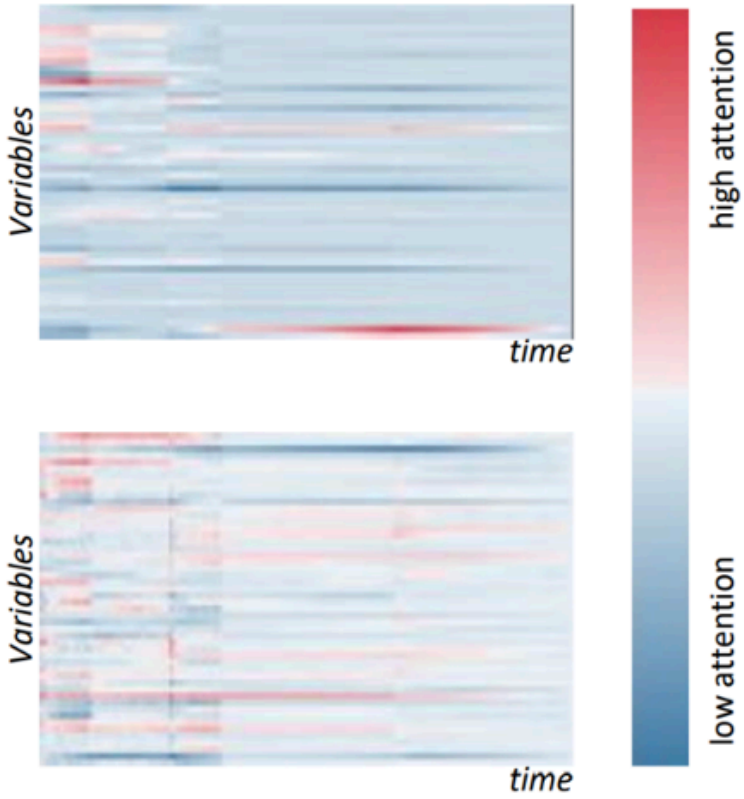

Figure 5.3: Utilization of Attention mechanisms in RNN's explainability. Source: Rojat et Al., 2021 [48]

5.6 Model Proposals

Even though the authors of the End-to-end methodology specify that the methodology will try to take advantage of the increasingly rich literature on neural networks, with a focus on a multivariate, probabilistic model, one of the important claims that they make is: "One could easily replace DeepVAR with any recently proposed multivariate forecasting model without requiring major changes.". This claim allowed to continue thinking of the possibilities of adjusting the methodology in favor of explainability to contribute to result communications with stakeholders and compliance with the new AI regulations.

The assumption that was made is that if one moves away from deep learning models, then the explainability would be able to increase but sacrificing the accuracy of the model. This 'explainability-accuracy trade-off' would give the companies a chance to simplify their procedures depending on the need of cases and not overshoot with a highly complex model that they do not understand.

But how can DeepVAR be replaced in the End-to-End methodology? First, it needs to be considered that any model to be proposed needs to deal with a multivariate forecasting problem, which means that a time series has more than one time series variable and the model has to produce forecasts for all of them. Secondly, it needs to be a global model which considers the relations between the different time series, so all variables affect each other. And third, it would be a fairly explainable model.

The model on which the DeepVARHierarchical is based is the VAR model – Vector Autoregressive model, therefore, VAR would be the first natural candidate to replace DeepVAR since it also fulfills all the above-mentioned conditions. The variables in this model are modeled as if they all influence each other equally. In more formal terminology, all variables are now treated as "endogenous". It generalizes the univariate autoregressive (AR) model for forecasting a vector of time series. Despite the criticism that VARs face for being atheoretical – not built in some economic theory that imposes a theoretical structure – they are very useful in several contexts as testing whether one variable is useful in forecasting another, impulse response analysis, forecast error variance decomposition, and as in the case of an End-to-End method to forecast a collection of related variables which would be part of a hierarchical structure [58].

Next, moving on the spectrum of this explainability-accuracy trade-off one can also utilize more complex models. An extension of VAR that is suitable to this problem is VARIMA – Vector Autoregressive Integrating Moving Averages, which handles non-stationary data by including the integration component and takes into account the impact of shocks at various time lags by including also Moving Averages (MA) component, potentially leading to more accurate forecasts. This increased complexity comes also with reduced explainability [56].

Furthermore, another proposal for these experimentations would be the VISTS framework – Vector Innovations Structural Time Series – which encapsulates exponential smoothing methods in a multivariate setting. It allows for the modeling of multiple time series simultaneously while accounting for structural components such as trends, seasons, cycles, and the influence of exogenous variables. While the primary output of VAR, VARIMA, and VISTS models may be point forecasts, they all can provide probabilistic forecasts [10].

All the proposed models would be worth pursuing in an experiment, and that is what was attempted to do during this thesis. Even though initially the task seemed straightforward, in the deep dives and experimentations that were conducted during this time, replacing DeepVAR with the proposed models, proved to be a complex challenge practically.

One of the main hurdles that were encountered was the complexity of transitioning from point forecasts to probabilistic forecasts. Considering that the second part of the End-to-End method relies on the distribution parameters of the probabilistic forecast as its inputs, this part of the task was crucial. Expect the conceptual and theoretical side of the challenge, the implementation of this part required structural changes of the codebase in a newly launched coding package as *Gluon TS* to modify the DeepVARHierarchical and major changes of the models themselves in other coding packages as *statsmodels*, where the integration of different packages was required. Both these changes needed to match each other both conceptually and practically.

6 Conclusions

This thesis has contributed to further studies of Hierarchical Data forecasting within the domain of Supply Chain Management. Through an examination of the End-to-End methodology and its practical implementation, this research has brought to light the significant benefits of this approach in generating coherent and probabilistic forecasts. By employing a single nonlinear model to train all the time series simultaneously and benefiting from the usage of their inter-dependencies, the method has demonstrated an improved fit for each series, enhancing the accuracy beyond the capabilities of the state-of-the-art models.

A central advantage identified in this work is the methodology's inherent design, which eliminates the need for independent reconciliation, ensuring the production of coherent forecasts. Moreover, the flexibility of the model training process, facilitated by the application-dependent loss functions, and its ability to incorporate structural constraints such as non-negativity, speaks to the robustness and adaptability of the End-to-End methodology.

The practical implementation of the End-to-End methodology within the context of demand forecasting in Supply Chain Management stands as one of the most impactful elements of this thesis. By applying this advanced forecasting approach, the research successfully navigated the complex hierarchy of supply chain data, producing probabilistic forecasts at all levels. This is particularly valuable in Supply Chain Management where decisions at every level—from strategic to operational—rely on accurate forecasts to manage inventory, allocate resources, and plan for future demand, where an End-to-End process would save a considerable amount of time and effort, consequentially lowering operational cost.

However, this thesis has not shied away from a critical analysis of the limitations present. The assumption of a Gaussian distribution, while standard, may not always encapsulate the true nature of the data. Furthermore, the limited model interpretability and the looming necessity to align with the new EU AI Act of 2023 pose challenges that the industry must proactively address. To this end, the thesis proposes two paths forward: enhancing the explainability of the current methodology or opting for an 'explainability-accuracy trade-off' by simplifying the methodology towards more interpretable models.

The practical challenges encountered during the implementation phase have underscored the complexity of

transitioning from point forecasts to probabilistic forecasts. The necessity to recode and restructure the base in GluonTS to accommodate modifications, along with integrating disparate coding packages like *statsmodels*, presented a substantial obstacle. The endeavor required not only a theoretical understanding but also a significant engineering effort to ensure conceptual and practical alignment.

This thesis stands as a new contribution to the field, offering both a methodological advancement in forecasting and a candid discussion of its potential and pitfalls. The insights garnered here lay a foundation for future research and development, guiding the pursuit of models that strike a balance between accuracy, coherence, and compliance with regulatory frameworks. As such, this work does not merely represent an academic exercise but a step forward in the practical application of End-to-End hierarchical forecasting in Supply Chain Management, with implications that extend to the broader landscape of forecasting.

References

[1] On the multivariate probability integral transformation. *Statistics Probability Letters*, 53(4):391–399, 2001.

[2] Forecasting: theory and practice. *International Journal of Forecasting*, 38(3):705–871, 2022.

[3] Navid Abiri and Mattias Ohlsson. Variational auto-encoders with student's t-prior. In *Proceedings, 27th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2019)*, pages 415–420, Bruges, 2019.

[4] Akshay Agrawal, Shane Barratt, Stephen Boyd, Enzo Busseti, and Walaa M. Moursi. Differentiating through a cone program. *arXiv preprint arXiv:1904.09043*, 2019.

[5] Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Vincent Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C. Maddix, Syama Sundar Rangapuram, David Salinas, Johann Schulz, et al. Gluonts: Probabilistic time series models in python. *Journal of Machine Learning Research*, 2019.

[6] Amazon Web Services. Amazon sagemaker deepar. (accessed: February 14, 2024).

[7] Amazon Web Services. Interpretability versus explainability in ai/ml, Latest.

[8] B. Amos and J. Z. Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pages 136–145. PMLR, 2017.

[9] George Athanasopoulos, Roman Ahmed, and Rob J. Hyndman. Hierarchical forecasts for australian domestic tourism. *International Journal of Forecasting*, 25(1):146–166, 2009.

[10] George Athanasopoulos and Ashton de Silva. Multivariate exponential smoothing for forecasting tourist arrivals. *Journal of Travel Research*, 51:640–652, 09 2012.

[11] George Athanasopoulos, Rob J. Hyndman, Nikolaos Kourentzes, and Anastasios Panagiotelis. Forecast reconciliation: A review. *International Journal of Forecasting*, 2023.

[12] George Athanasopoulos, Rob J. Hyndman, Nikolaos Kourentzes, and Fotios Petropoulos. Forecasting with temporal hierarchies. *European Journal of Operational Research*, 262(1):60–74, 2017.

[13] Australian Bureau of Statistics. Time series analysis: The basics. Accessed on February 14, 2024.

[14] M. Zied Babai, John E. Boylan, and Bahman Rostami-Tabar. Demand forecasting in supply chains: a review of aggregation and hierarchical approaches. *International Journal of Production Research*, 60(1):324–348, 2022.

[15] Souhaib Ben Taieb and Bonhyung Koo. Regularized regression for hierarchical forecasting without unbiasedness conditions. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1337–1347, 2019.

[16] George E. P. Box and Gwilym M. Jenkins. *Time series analysis: Forecasting and control*. Holden-Day, 1970.

[17] Joseph F. Coates. Why forecasts fail. *Research-technology Management*, 36:4, 1993.

[18] Comet. Explainability in ai and machine learning systems: An overview, Latest.

[19] European Commission. Regulatory framework for artificial intelligence. Digital Strategy, 2023.

[20] Francis X Diebold. *Elements of forecasting*. Citeseer, 1998.

[21] Michael Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. In Samy Bengio, Hanna Wallach, Hugo Larochelle, Kristen Grauman, Nicolo Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 441–452. Curran Associates, Inc., 2018.

[22] Paul H. Garthwaite, Joseph B. Kadane, and Anthony O'Hagan. Statistical methods for eliciting probability distributions. *Journal of the American Statistical Association*, 100(470):680–700, 2005.

[23] Gerd Gigerenzer, Ralph Hertwig, Eva Van Den Broek, Barbara Fasolo, and Konstantinos V Katsikopoulos. "a 30% chance of rain tomorrow": How does the public understand probabilistic weather forecasts? *Risk Analysis: An International Journal*, 25(3):623–629, 2005.

[24] Tilmann Gneiting. Making and evaluating point forecasts. *Journal of the American Statistical Association*, 106(494):746–762, 2011.

[25] Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1(1):125–151, 2014.

[26] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

[27] Tilmann Gneiting and Roopesh Ranjan. Combining predictive distributions. 2013.

[28] Tilmann Gneiting, Larissa I. Stanberry, Eric P. Grimit, Leonhard Held, and Nicholas A. Johnson. Assessing probabilistic forecasts of multivariate quantities, with applications to ensemble predictions of surface winds. *Test*, 17:211–264, 2008.

[29] C.W. Gross and J.E. Sohl. Disaggregation methods to expedite product line forecasting. *Journal of Forecasting*, 9:233–254, 1990.

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

[31] R.J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, Melbourne, Australia, 3rd edition, 2018. Accessed on December 23, 2023.

[32] R.J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, Melbourne, Australia, 3rd edition, 2021. Accessed on December 9, 2023.

[33] Rob J. Hyndman. Reconciliation notation. `https://robjhyndman.com/hyndsight/reconciliation-notation.html`.

[34] Rob J. Hyndman, Roman A. Ahmed, George Athanasopoulos, and Han Lin Shang. Optimal combination forecasts for hierarchical time series. *Computational Statistics & Data Analysis*, 55(9):2579–2589, 2011.

[35] Rob J Hyndman and Yeasmin Khandakar. auto.arima: Automatic arima modeling, 2021. Accessed: [Insert date here].

[36] M.E. Ibrahim, S.A. Metawae, and I.M. Aly. Statistical decomposition analysis of financial statements and prediction of bond rating changes. *Managerial Finance*, 16(1):7–15, 1990.

[37] H. V. Jagadish, L. V. S. Lakshmanan, and D. Srivastava. Hierarchical or relational? a case for a modern hierarchical data model. In *Proceedings 1999 Workshop on Knowledge and Data Engineering Exchange (KDEX'99)*, pages 3–10, Chicago, IL, USA, 1999.

[38] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[39] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[40] D. Kwiatkowski, P. C. B. Phillips, P. Schmidt, and Y. Shin. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of Econometrics*, 54(1-3):159–178, 1992.

[41] Francesco Laio and Stefania Tamea. Verification tools for probabilistic forecasts of continuous hydrological variables. *Hydrology and Earth System Sciences*, 11(4):1267–1277, 2007.

[42] Aniek F. Markus, Jan A. Kors, and Peter R. Rijnbeek. The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies. *Journal of Biomedical Informatics*, 113:103655, 2021.

[43] Konstantinos Nikolopoulos, Aris A Syntetos, and John E Boylan. An aggregate–disaggregate intermittent demand approach (adida) to forecasting: an empirical proposition and analysis. *Journal of the Operational Research Society*, 62:544–554, 2011.

[44] Anastasios Panagiotelis, Puwasala Gamakumara, George Athanasopoulos, and Rob J. Hyndman. Probabilistic forecast reconciliation: Properties, evaluation, and score optimization. *European Journal of Operational Research*, 306(2):693–706, 2023.

[45] European Parliament. Eu ai act: First regulation on artificial intelligence. European Parliament News, 2023.

[46] Fred L. Ramsey. Characterization of the partial autocorrelation function. *The Annals of Statistics*, 2(6):1296–1301, 1974.

[47] Syama Sundar Rangapuram, Leif D. Werner, Konstantinos Benidis, Pedro Mercado, Jan Gasthaus, and Tim Januschowski. End-to-end learning of coherent probabilistic forecasts for hierarchical time series. In *International Conference on Machine Learning*, pages 8832–8843. PMLR, July 2021.

[48] Thomas Rojat, Raphaël Puget, David Filliat, Javier Del Ser, Rodolphe Gelin, and Natalia Díaz-Rodríguez. Explainable artificial intelligence (xai) on timeseries data: A survey. *arXiv preprint arXiv:2104.00950*, 2021.

[49] Francisco R Ruiz, Michalis RC AUEB Titsias, and David Blei. The generalized reparameterization gradient. In Daniel Lee, Masashi Sugiyama, Ulrike Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 460–468. 2016.

[50] David Salinas, Michael Bohlke-Schneider, Laurent Callot, Ricardo Medico, and Jan Gasthaus. High-dimensional multivariate forecasting with low-rank gaussian copula processes. In *Advances in Neural Information Processing Systems*, volume 32, pages 6827–6837, 2019.

[51] Andrea Silvestrini and David Veredas. Temporal aggregation of univariate and multivariate time series models: A survey. *Journal of Economic Surveys*, 22(3):458–497, March 2008.

[52] Stephen M. Stigler. The transition from point to distribution estimation. *Bulletin of the International Statistical Institute*, 46:332–340, 1975.

[53] Hristos Tyralis and Georgia Papacharalampous. A review of probabilistic forecasting and prediction with machine learning. *Journal Name*, Volume(Issue):Page Range, 2022.

[54] Anthony Unwin. *Graphical Data Analysis with R*. Chapman & Hall/CRC, 2015.

[55] Tim Van Erven and Jairo Cugliari. Game-theoretically optimal reconciliation of contemporaneous hierarchical time series forecasts. In *Modeling and Stochastic Learning for Forecasting in High Dimensions*, pages 297–317. Springer, 2015.

[56] Ky M Vu. *The ARIMA and VARIMA time series: Their modelings, analyses and applications*. AuLac Technologies Inc., 2007.

[57] S.L. Wickramasuriya, G. Athanasopoulos, and R.J. Hyndman. Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, 114(526):804–819, 2019.

[58] Eric Zivot and Jiahui Wang. Vector autoregressive models for multivariate time series. *Modeling financial time series with S-PLUS®*, pages 385–429, 2006.

Acknowledgments

This journey has been a rollercoaster of emotions, filled with challenges that taught me a lot about Data Science but also about myself. I continue to be a strong believer that academic and personal growth should go hand in hand. This unforgettable experience would not be the same without the people who surrounded me throughout.

I thank my all professors, with special thanks to my thesis advisor, Prof. Mariangela Guidolin, for her inspiring lectures, continuous support, guidance, and patience, from the first moment that I asked her to supervise this thesis.

Next, I want to thank Mr. Oliver Roch, my manager during my internship at Amazon. Your support, lessons, guidance, and motivation enabled me to finally be confident in my knowledge and skills.

To all my friends in Padova, you turned Padova into a second home. To the friends I hold dearest, independently from the geographical proximity, you empower and motivate me to be who and where I am today. I am eternally grateful and I love you a lot.

Lastly, the most important ones, to my parents, my brother, and the rest of my family, this is your success as much as it is mine. Thank you!