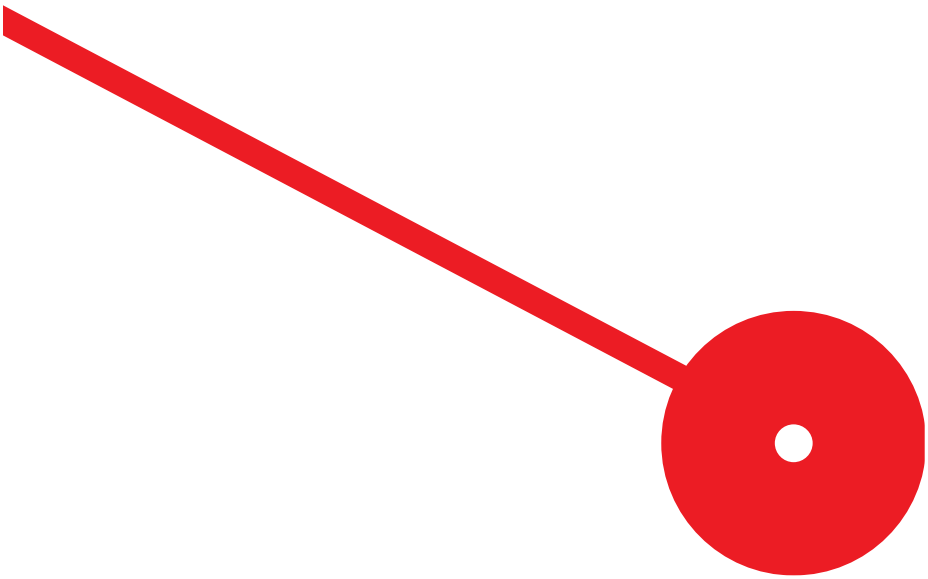
INSTITUTO
SUPERIOR
DE CONTABILIDADE
E ADMINISTRAÇÃO
DO PORTO
POLITÉCNICO
DO PORTO

M MESTRADO
BUSINESS INTELLIGENCE AND ANALYTICS

Hierarchical Time Series Forecasting with Deep Learning in Retail

José Carlos Guedes Gomes

2024



José Carlos Guedes Gomes. Hierarchical Time Series Forecasting with Deep Learning in Retail.
2024

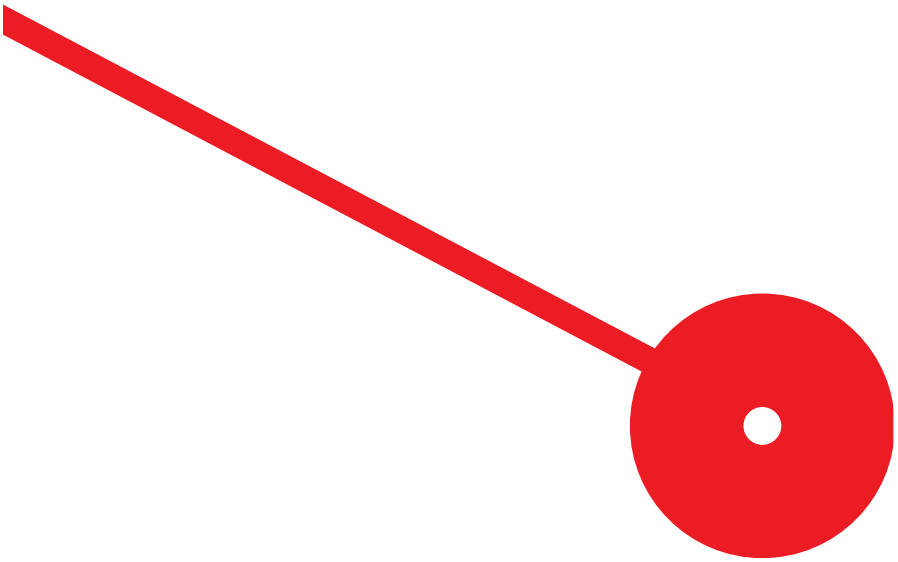
INSTITUTO
SUPERIOR
DE CONTABILIDADE
E ADMINISTRAÇÃO
DO PORTO
POLITÉCNICO
DO PORTO

M MESTRADO
BUSINESS INTELLIGENCE AND ANALYTICS

Hierarchical Time Series Forecasting with Deep Learning in Retail

José Carlos Guedes Gomes

Dissertação de Mestrado apresentada ao Instituto Superior de Contabilidade e Administração do Porto para a obtenção do grau de Mestre em Business Intelligence and Analytics, sob orientação da Doutora Patrícia Alexandra Gregório Ramos e do Doutor José Manuel Soares Oliveira.



José Carlos Guedes Gomes. Hierarchical Time Series Forecasting with Deep Learning in Retail.
2024

Acknowledgement

I wish to extend my deepest gratitude to my supervisors, Professor Patricia Alexandra Gregório Ramos and Professor José Manuel Soares Oliveira, for their invaluable dedication, time, and the insightful knowledge they imparted throughout this journey. Their guidance was instrumental to the successful development of this work, allowing me to explore and deepen my academic understanding.

I am also profoundly grateful to my family for their unwavering patience, affection, and unconditional support during the entirety of this master's degree.

To my professors, I express my appreciation for their commitment to delivering knowledge in an accessible and inspiring manner, and for their willingness to address my questions and assist me in overcoming the challenges faced.

I am thankful to my colleagues and friends who accompanied me throughout this journey, for their shared experiences, camaraderie, and the fruitful exchange of knowledge.

To all of you, my sincere thanks

Resumo:

Esta dissertação investiga a aplicação de modelos de *deep learning* para melhorar as metodologias de previsão hierárquica na indústria retalhista. Os métodos de previsão tradicionais muitas vezes falham na captura das complexas interdependências e estruturas hierárquicas presentes nas operações de retalho, levando a previsões subótimas e alocação ineficiente de recursos. Este estudo procura superar estas limitações, aproveitando a capacidade dos modelos de *deep learning*, como as redes *MultiLayer Peceptron* (MLP) e as arquiteturas baseadas em Transformer, para identificar padrões e dependências não lineares nos dados. O estudo centra-se no desenho e implementação de um sistema de previsão hierárquico capaz de prever com precisão a procura a vários níveis, desde produtos individuais até vendas agregadas em lojas e mercados. Ao integrar modelos de *deep learning* num quadro hierárquico, o estudo visa melhorar a precisão e eficiência das previsões em domínios críticos do retalho, como o planeamento da procura, a previsão de vendas e a gestão de inventário. O quadro proposto é concebido para ser adaptável e escalável, capaz de acomodar mudanças na estrutura organizacional, carteiras de produtos e dinâmica de mercado, garantindo a sua relevância e eficácia em diversos contextos empresariais. Este trabalho utiliza o conjunto de dados da competição M5, publicamente disponível, um *benchmark* para avaliar metodologias de previsão que envolve a geração de estimativas pontuais e intervalos probabilísticos para dados de séries temporais hierárquicas da Walmart. Este conjunto de dados consiste em dados de vendas diárias para 3.049 produtos em dez lojas em três estados dos EUA num período de 1.969 dias. Devido a restrições computacionais, o estudo utiliza uma versão reduzida do conjunto de dados com 1.288 séries temporais, preservando as características principais do conjunto de dados. A dissertação utiliza uma combinação de estruturas hierárquicas e agrupadas para representar com precisão as dependências dentro do conjunto de dados M5. Vários métodos de reconciliação, incluindo a agregação ascendente e a reconciliação MinTrace, são utilizados para alinhar as previsões de diferentes níveis e garantir a coerência ao longo da hierarquia. É realizada uma extensa otimização de hiperparâmetros utilizando a biblioteca Optuna para melhorar a precisão preditiva de modelos como o Vanilla Transformer, TFT, Informer, PatchTST, Autoformer, MLP, NBEATS e NHITS. A dissertação avalia tanto as previsões pontuais como as probabilísticas para capturar a incerteza nas previsões de vendas futuras. O Erro Médio Absoluto Escalado (MASE) é utilizado como métrica primária para as previsões pontuais, e o *Continuous Ranked*

Probability Score (CRPS) é utilizado para as previsões probabilísticas. Os resultados demonstram que os modelos de *deep learning*, particularmente as arquitecturas baseadas em Transformer, melhoram significativamente a precisão da previsão em comparação com métodos tradicionais como ARIMA e ETS, especialmente quando combinados com técnicas de reconciliação avançadas. O estudo destaca a eficácia da abordagem de reconciliação MinTrace, particularmente com variações ponderadas, no equilíbrio entre a precisão da previsão e a consistência hierárquica. Em conclusão, a dissertação fornece uma solução de previsão robusta e escalável para a indústria retalhista. A integração de modelos de *deep learning* com metodologias de previsão hierárquica permite às empresas tomar decisões baseadas em dados em todos os níveis da organização, levando a uma melhor alocação de recursos, maior eficiência operacional e, em última análise, melhores resultados empresariais.

Palavras chave: Aprendizagem Profunda, Previsão Hierárquica, Retalho, Otimização de Recursos

Abstract:

This dissertation investigates the application of deep learning models to enhance hierarchical forecasting methodologies in the retail industry. Traditional forecasting methods often fail to capture the complex interdependencies and hierarchical structures prevalent in retail operations, leading to suboptimal predictions and inefficient resource allocation. This research seeks to overcome these limitations by leveraging the ability of deep learning models, such as MultiLayer Peceptron (MLP) networks and Transformer architectures, to identify non-linear patterns and dependencies in data. The study focuses on designing and implementing a hierarchical forecasting system that can accurately predict demand at various levels, from individual product SKUs to aggregate sales across stores, chains, and markets. By integrating deep learning models within a hierarchical framework, the study aims to improve the accuracy and efficiency of predictions in critical retail domains such as demand planning, sales forecasting, and inventory management. The proposed framework is designed to be adaptable and scalable, capable of accommodating changes in organizational structure, product portfolios, and market dynamics, ensuring its relevance and effectiveness in diverse business contexts. This research utilizes the publicly available M5 competition dataset, a benchmark for evaluating forecasting methodologies that involves generating point estimates and probabilistic intervals for hierarchical time series data from Walmart. This dataset consists of daily sales data for 3,049 products across ten stores in three US states over a period of 1,969 days. Due to computational constraints, the study uses a reduced version of the dataset with 1,288 time series, preserving the dataset's key characteristics. The dissertation employs a combination of hierarchical and grouped structures to accurately represent dependencies within the M5 dataset. Several reconciliation methods, including bottom-up aggregation and MinTrace reconciliation, are used to align forecasts from different levels and ensure coherence across the hierarchy. Extensive hyperparameter optimization is performed using the Optuna library to enhance the predictive accuracy of models like Vanilla Transformer, TFT, Informer, PatchTST, Autoformer, MLP, NBEATS, and NHITS. The dissertation evaluates both point and probabilistic forecasts to capture uncertainty in future sales predictions. Mean Absolute Scaled Error (MASE) is used as the primary metric for point forecasts, and the Continuous Ranked Probability Score (CRPS) is used for probabilistic forecasts. The results demonstrate that deep learning models, particularly Transformer-based architectures, significantly improve

forecasting accuracy compared to traditional methods like ARIMA and ETS, especially when combined with advanced reconciliation techniques. The study highlights the effectiveness of the MinTrace reconciliation approach, particularly with weighted variations, in balancing forecast accuracy and hierarchical consistency. In conclusion, the dissertation provides a robust and scalable forecasting solution for the retail industry. The integration of deep learning models with hierarchical forecasting methodologies enables businesses to make data-driven decisions at every level of the organization, leading to improved resource allocation, enhanced operational efficiency, and ultimately, better business outcomes.

Keywords: Deep Learning, Hierarchical Forecasting, Retail Business, Resources Optimization

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ANNs: Artificial Neural Networks. ,
ARIMA: Autoregressive Integrated Moving Average.
BLEU: Bilingual Evaluation Understudy.
CA: California.
CNNs: Convolutional Neural Networks.
CRPS: Ranked Probability Score.
ETS: Error Trend and Seasonality, or Exponential Smoothing.
FSS: Forecasting Support Systems.
GDP: Gross Domestic Product.
GELU: Gaussian Error Linear Unit.
GLS: Generalized Least Squares.
GLUs: Gated Linear Units.
GRNs: Gated Residual Networks.
HPO: Hyperparameter Optimization.
LSTF: Long Sequence Time-Series Forecasting.
LSTMs: Long Short-Term Memory Networks.
MAPEs: Mean Absolute Percentage Errors.
MASE: Mean Absolute Scaled Error.
MLPs: multilayer perceptrons.
MSSE: Mean Squared Scaled Error.
N-BEATS: Neural Basis Expansion Analysis for Interpretable Time Series.
N-HiTS: Neural Hierarchical Interpolation for Time Series.
NNs: Neural Networks.
PatchTST: Patch Time Series Transformer.
ReLU: Rectified Linear Unit.
RNNs: Recurrent Neural Networks.
SGD: Stochastic Gradient Descent.
SIM: Spatial Interaction Model.
SKU: Stock Keeping Unit.
SLAM: Store Location Assessment Model.
SVMs: Support Vector Machines.
TFT: Temporal Fusion Transformer.
TX: Texas.
WI: Wisconsin.
WLS: Weighted Least Squares. ,
WMT 2014: collection of datasets used in shared tasks of the Ninth Workshop on Statistical Machine Translation.
WQL: Weighted Quantile Loss.

CHAPTER I – INTRODUCTION

1 Introduction

Hierarchical forecasting plays a pivotal role in business operations, especially in industries where demand planning, sales forecasting, and inventory management are critical for success. Traditional forecasting methods often struggle to capture the complex interdependencies and hierarchical structures present within organizations, leading to suboptimal predictions and inefficient resource allocation.

In the retail industry, accurate forecasting is crucial due to the nature of the business model. The sector is marked by intense competition, a trend that has been escalating in recent years due to consumerism and globalization. Retailers are tasked with predicting sales for a vast array of products across diverse categories. This is done to optimize their operations economically and financially. This includes logistics operations aimed at minimizing stock levels and marketing efforts directed towards achieving ultimate customer satisfaction.

This dissertation seeks to address these challenges by leveraging the capabilities of deep learning models, such as MultiLayer Peceptron (MLP) networks and Transformer-based architectures, to enhance hierarchical forecasting methodologies.

Deep learning models have demonstrated remarkable proficiency in capturing nonlinear patterns and dependencies in data, making them well-suited for the intricacies of hierarchical forecasting tasks.

At its core, the research aims to design and implement a hierarchical forecasting framework that seamlessly integrates deep learning architectures. The framework will be tailored to accommodate the hierarchical structures inherent in diverse business units and product categories. By doing so, it will provide a comprehensive forecasting solution capable of generating accurate predictions across multiple levels of the organizational hierarchy.

One of the key advantages of employing deep learning models in hierarchical forecasting is their ability to learn from historical data while simultaneously capturing temporal dependencies and hierarchical relationships.

The research will focus on three primary areas: demand forecasting, sales projections, and inventory management. These areas are fundamental to business operations and are directly impacted by the accuracy of forecasting models. By leveraging deep learning

models within a hierarchical framework, the study aims to enhance the accuracy and efficiency of predictions in these critical domains.

Furthermore, the proposed framework will be designed with adaptability and scalability in mind. It will be capable of accommodating changes in organizational structure, product portfolios, and market dynamics, ensuring its relevance and effectiveness across various business contexts. This adaptability is essential for businesses operating in dynamic environments where agility and responsiveness are paramount.

Creating accurate predictions from hierarchically structured time series is a complex task due to the challenge of deciphering the relationships between different time series. For decision-making to be coherent across various hierarchical levels, it's imperative that the forecast from each aggregated series aligns with the sum of the forecasts from its corresponding disaggregated series. The lower level of the hierarchy can encompass thousands of time series, resulting in substantial computational requirements (Taieb et al., 2017).

It's recommended to steer clear generating forecasts independently for each hierarchical level, as this can lead to discrepancies where the forecast at a higher level doesn't tally with the sum of the forecasts at a lower level, and the reverse is also true (Gene, 2001).

Ultimately, the goal of this dissertation is to provide businesses with a robust forecasting solution that empowers them to make informed decisions at every level of the organizational hierarchy. By leveraging deep learning models within a hierarchical framework, the research aims to revolutionize the way businesses approach forecasting, leading to improved resource allocation, enhanced operational efficiency, and ultimately, better business outcomes.

This dissertation is structured into five chapters. The first chapter serves as an introduction, followed by the second chapter which provides a comprehensive literature review on the current advancements in this specific field of study, highlighting key concepts and critically analyzing various contributions from existing literature. The third chapter briefly outlines the potential methodologies to be employed. The fourth chapter details the case study undertaken and the results derived from it. The final chapter, the fifth, draws conclusions from the study and discusses its limitations. The dissertation ends with a list of references.

2 Literature Review

2.1 Retail Forecasting

From a conceptual standpoint, retail can be delineated as the comprehensive process encompassing the marketing of goods and services directly to the ultimate consumers (Aras et al., 2017). The task of accurately predicting the sales of a specific good or service emerges as a paramount challenge in the administration of a retailer's supply chain. Inaccuracies in forecasts can precipitate stock shortages, thereby leading to customer dissatisfaction (Beutel and Minner, 2012).

Retailers need demand forecasts at various levels of aggregation to facilitate a multitude of decisions along the supply chain. For instance, they require forecasts at the store level to manage its inventory, while also needing regionally aggregated forecasts to administer the inventory of a specific distribution center (Kremer et al., 2016). In their endeavor to plan and manage their supply chain, companies predominantly establish a unit dedicated to forecasting.

The prevalent approach to demand forecasting in the retail sector entails the utilization of a computational forecasting system to generate initial forecasts, followed by a manual adjustment of these forecasts by the individual or department accountable for the company's sales forecasting. This is done to accommodate exceptional circumstances anticipated during the planning horizon (Fildes et al., 2009). Celia et al. (2003) characterizes forecasting as a probabilistic estimation of a future value.

The fundamental premise underpinning most forecasting methods is the reflection of patterns and behaviors observed in the past and in the future. In this context, and for the objectives of this dissertation, the prediction of a retailer's sales involves the derivation of potential future sales values via a statistical model, predicated on historical sales data. At present, research on sales forecasting has been significantly concentrated on integration.

Sales data at the product level are influenced by various factors, such as: Marketing mix, Seasonal events, Competitive behaviors. These characteristics make the data volatile and asymmetric, with high seasonality and explanatory variables of multiple dimensions.

In retail, the objective of sales forecasting is to generate projections for an extensive array of time series within a brief forecasting horizon. The precise prediction of demand for

each commodity in every store or distribution center is of paramount importance, given that numerous operational decisions, encompassing pricing strategies, allocation of space, order processing and inventory management are intrinsically linked to demand forecasting. In essence, inaccuracies in forecasting precipitate subpar services and escalate costs.

In retail, all forecasting hinges on a certain degree of aggregation (Figure 1) across product units, locations, or time intervals, contingent upon the objective of the forecasting activity. This section employs the term ‘aggregate retail sales forecasting’ to denote the total retail sales in a market, chain, store type (e.g., in-town), or individual store, as opposed to forecasts specific to a product (SKU/brand/category).

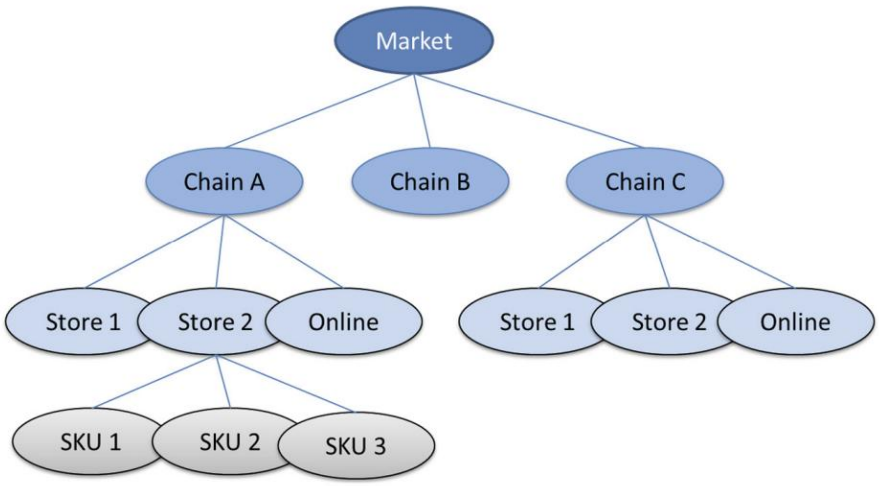


Figure 1 - Hierarchical retail sales from SKU to Store to Chain to Market.

In other words, we implicitly aggregate across products and promotions up to a specific granularity (e.g., weekly or monthly) over a defined time period. Typically, aggregate retail sales are quantified in terms of revenue rather than product units. However, since price increases due to inflation usually have minimal impact on demand, the effect on total revenue can be more readily captured by forecasting in units and then multiplying by price.

2.1.1 Market Level Demand

Market-level aggregate sales forecasting pertains to the total sales of a retail category, a channel, or the entire industry within a specific region or country. The time frame for these market-level forecasts could be monthly, quarterly, or yearly. These forecasts are

crucial for retailers, particularly large ones, to understand the changing market conditions and their potential impact on their total sales (Alon, Qi, & Sadowski, 2001). They play a central role in the strategic planning and operation of a retail business, as they help identify the growth potential of various business models and stimulate the development of new strategies to maintain their market position.

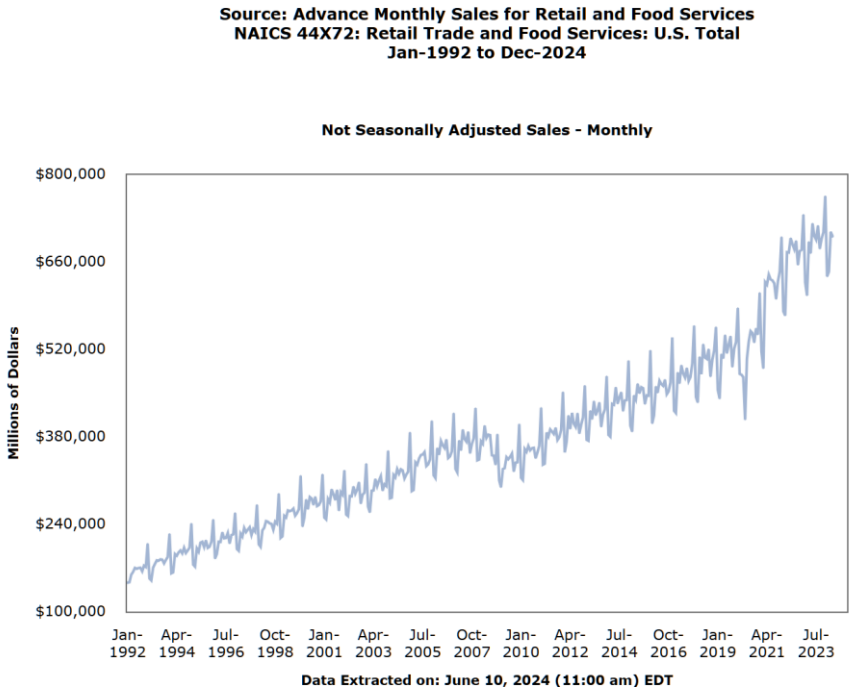


Figure 2 - US retail sales monthly series in millions of dollars.

Market-level aggregate retail sales data often display strong trends, seasonal variations, serial correlation, and regime shifts, as any long span in the data may include economic growth, inflation, and unexpected events (Figure 2). Time series models have long been applied for market-level aggregate retail sales forecasting as they provide solutions for capturing these characteristics (e.g., Alon et al., 2001; Bechter & Rutner, 1978; Schmidt, 1979; Zhang & Qi, 2005). Simple exponential smoothing and its extensions, along with Autoregressive Integrated Moving Average (ARIMA) models, have been the most frequently employed time series models for market-level sales forecasting.

However, researchers have found that traditional time series models are sometimes inadequate for approximating aggregate retail sales, identifying evidence of nonlinearity and volatility in market-level retail sales time series (e.g., Alon et al., 2001; Chu & Zhang, 2003; Kuvulmaz, Usanmaz, & Engin, 2005; Zhang & Qi, 2005). They therefore resorted

to nonlinear models, especially artificial neural networks. The results indicate that traditional time series models with a stochastic trend, such as exponential smoothing and ARIMA, performed well when macroeconomic conditions are relatively stable. However, when economic conditions are volatile (with rapid changes in economic conditions), artificial neural networks (ANNs) have been claimed to outperform the linear methods (Alon et al., 2001).

Unlike time series models, econometric models depend on the successful identification of predictable explanatory variables. Bechter and Rutner (1978) compared the forecasting performances of ARIMA, and econometric models designed for US retail sales. They used two explanatory variables in the economic model: personal income and nonfinancial personal wealth, as measured by an index of the price of common stocks. Past values of retail sales were also included in alternative models that mixed autoregressive and economic components. They found that ARIMA forecasts were usually no better and often worse than those generated by a simple single equation economic model, while the mixed model had a better record over the entire 30-month forecast period than any of the other three models.

In summary, no recent research has been found that uses current econometric methods to link retail sales to macroeconomic variables such as Gross Domestic Product (GDP) and evaluates their conditional and unconditional performances relative to time series approaches. The evidence on the performance of nonlinear models is limited, with too few series from too few countries, and no comparison with econometric models has been made.

2.1.2 Chain and Channel Level Demand

In retail, all forecasting hinges on a certain degree of aggregation across product units, locations, or time intervals, contingent upon the objective of the forecasting activity. This section employs the term ‘aggregate retail sales forecasting’ to denote the total retail sales in a market, chain, store type (e.g., in-town), or individual store, as opposed to forecasts specific to a product (SKU/brand/category). In other words, we implicitly aggregate across products and promotions up to a specific granularity (e.g., weekly or monthly) over a defined time period. Typically, aggregate retail sales are quantified in terms of revenue rather than product units. However, since price increases due to inflation usually have minimal impact on demand, the effect on total revenue can be more readily captured by

forecasting in units and then multiplying by price. We review existing research on three distinct levels: the aggregate retail sales in a market, a chain, and a store. While the forecasting of aggregate sales at these three levels shares many common issues, such as seasonality and trend, they pose different forecasting questions and have different objectives, data characteristics, and solutions.

2.1.3 Store Level Demand

Retailers, with their diverse store formats catering to various customer segments in different locations, rely heavily on sales forecasting. This forecasting can be categorized into two: (1) forecasting of existing store sales for distribution, target setting, viability, financial control, and workforce planning, and (2) forecasting of potential sales for new store site selection analysis.

Existing store sales forecasting employs both univariate time series models and regression models. Davies (1973) demonstrated the explanatory power of factor scores of individual stores on their sales performance levels using factor analysis. Geurts and Kelly (1986) concluded that univariate time series methods outperformed judgment or econometric models in forecasting monthly sales of a department store. However, the evidence on its comparative accuracy is not convincing.

The forecasting of store activity or footfalls in a shopping center or individual store, which was previously problematic due to data collection issues (Abrishami, Kumar, & Nienaber, 2017), can now utilize recently available 'big' data in the form of third-party mobile payment transactions (Ma & Fildes, 2018).

The forecasting of a new store's sales potential is a crucial yet challenging task. Traditional approaches can be classified into three categories: judgmental, analogue regression, and spatial interaction models (also called gravitational models). The success of judgmental approaches depends on the experience of the location analyst (Reynolds & Wood, 2010). Retailers often use a checklist to assess systematically the relative value of a given site compared to other potential sites in the area.

The analogue regression generates turnover forecasts for a new store by comparing the proposed site with analogous existing sites. Simkin (1989) reported the successful application of a regression-based store location assessment model (SLAM) in several of the UK's major retailers. However, the results on the predicted turnover suggested that

the use of regression equations was not sufficient to predict the potential performances of stores in new locations.

The spatial interaction model (SIM) (or gravity model) is used widely as a sophisticated retail location analysis tool. Unlike analogous regressions, which mainly rely on data from existing stores in the same chain, SIM uses data from various sources to improve the prediction accuracy. A SIM is based on the theory that expenditure flows and subsequent store revenue are driven by the store's comparative attractiveness and constrained by distance (Newing, Clarke, & Clarke, 2014).

Retailers typically operate multiple stores of varying formats, catering to different customer segments in diverse locations. These stores' sales are significantly influenced by factors such as location, local economy, competitive retailers, consumer demographics, promotions, weather, seasons, and local events. Forecasting store sales can be bifurcated into two categories: (1) forecasting existing store sales for distribution, target setting, financial control, and workforce planning, and (2) forecasting potential sales for new store site selection analysis.

Existing store sales forecasting employs both univariate time series models and regression models. Recently, Glaeser et al. (2019) provided a solution to optimally select locations for the pickup of online purchases, using a random forest that includes geographic and other predictors.

Spatial Interaction Models (SIMs) are usually validated on in-sample data. However, Birkin, Clarke & Clarke (2010) criticized this approach, emphasizing the importance of a hold-out sample. They showed that their model could be operationalized with a forecasting accuracy of around 10%, which was better than the company's performance.

Predictive models of store performance are only one element in supporting the location decision. Wood and Reynolds (2013) discuss how the models are combined with context-specific knowledge, the judgments of location analysts, and analogous information to produce final recommendations.

With the rapid changes on the high street in many countries showing increasing vacancy rates, these forecasting models will increasingly have a new use: for identifying shops to be closed. We speculate that multivariate time series models that include indicator

variables (for the store type), supplemented by local knowledge, should prove useful. However, this research remains to be done.

Note that any evaluation of new store forecasts (or of forecasts that might be used to determine which stores to close) needs to take a potential selection bias into account. The candidates selected for development may therefore see systematically lower sales than were forecasted, because of regression to the mean. This potential selection bias is rarely discussed in the literature.

2.1.4 Product Level Demand

In retail, product-level demand forecasting typically aims to generate predictions for a multitude of time series over a brief forecasting horizon. This contrasts with long-term forecasting, which focuses on one or a few time series at a more aggregate level. The ability to accurately forecast the demand for each item in each store or distribution center is vital for a retail chain's survival and growth. This is because many operational decisions, such as pricing, space allocation, listing/delisting, ordering, and inventory management for an item, are directly linked to its demand forecast. Forecast errors can directly lead to subpar service and increased costs. Order decisions need to ensure that the inventory level is not too high to avoid excessive inventory costs (including spoilage and obsolescence), nor too low to avoid stockouts and lost sales. As a result, forecasting for inventory control should place more emphasis on quantile forecasts, predictive densities, or prediction intervals. However, most of the existing research in this field still focuses on unbiased expectation forecasts or the minimization of Mean Absolute Percentage Errors (MAPEs).

2.1.4.1 Time Dimension of Product

Different temporal granularities are admitted for different management decisions: Operational decisions require more detailed forecasts, such as daily forecasts for store replenishment. Whereas strategic decisions may utilize weekly or monthly forecasts.

2.1.4.2 Product Dimension of Product

SKU (Stock Keeping Unit): The basic unit for forecasting in retail, used for daily stock and distribution planning. Brand: Includes several SKUs with different variations, such as types of packaging and sizes. Category: Contains several brands and SKUs with common attributes, used for budget planning and product mix decision.

2.1.4.3 Supply Dimension of Product

The typical supply chain includes manufacturers, intermediaries, distribution centers and stores. Forecasts at different levels of the supply chain are necessary for replenishment and distribution decisions.

2.1.5 Data Aggregation

The choice of data aggregation level depends on the demand generation process. Bottom-up: necessary when there are large differences in the time series of demand. Top-down: more accurate for homogeneous demand series and small samples. Middle-out or hierarchical combination: used to reconcile forecasts of all series in a hierarchy.

Recent advances in deep learning have shown great potential for improving the accuracy and robustness of hierarchical forecasting, especially in complex business contexts, namely in time series and specifically in retail. So, demand forecasting must consider three dimensions: level in the product hierarchy, position in the retail supply chain, time granularity.

2.2 Hierarchical Time Series Forecasting in Retail

2.2.1 Applications in General

Forecasting, defined as the task of extrapolating time series into the future, serves numerous crucial applications (Petropoulos et al., 2020), such as predicting demand for retail items (Bandara et al., 2019; Böse at al., 2017; Croston, 1972; Mukherjee et al., 2918; Salinas et al., 2020; Wen et al., 2017), traffic flow (Laptev et al., 2017; Li et al., 2018; Lv et al., 2014), energy demand and supply (Dimoulkas et al., 2019; Li et al., 2019; Saxena et al., 2019; Smyl and Hua, 2019), and financial metrics like covariance matrices, volatility, and long-tail distributions (Ballestra et al., 2019; Callot et al., 2019; Callot et al., 2017; Luo et al., 2018; Yoo & Kang, 2021).

It is a well-studied area (Hyndman & Athanasopoulos, 2018) with a dedicated research community. Additionally, machine learning, data science, systems, operations research, and application-specific research communities have extensively explored this problem (Faloutsos et al., 2018, 2019, 2020).

2.2.2 Use and Advancements

The history of NNs dates to 1957 (Rosenblatt, 1957), and their application in forecasting emerged in 1964 (Hu & Root, 1964).

Interest in neural networks (NNs) has fluctuated over the years, with peaks of attention attributed to breakthroughs. For instance, Rumelhart et al. (1985, 1986) popularized the training of multilayer perceptrons (MLPs) using back-propagation. Subsequent significant advances include the utilization of convolutional neural networks (CNNs) (LeCun & Bengio, 1995) and Long Short Term Memory (LSTM) cells, addressing issues with recurrent NNs' (RNNs) training (Hochreiter & Schmidhuber, 1997).

Despite these advancements, NNs remained challenging to train and work with. Alternative methods like Support Vector Machines (SVMs) (Boser et al., 1992) and Random Forests (Ho, 1995), developed in the 1990s, proved highly effective (LeCun et al., 1995). This diverted researchers' interest away from NNs, as reflected in a widely cited review (Zhang et al., 1998).

The breakthrough in 2006 by Hinton et al. (1978), demonstrating successful training of deep NNs with appropriately initialized weights, marked the dawn of the deep learning era.

The recent surge of attention on deep forecasting models is a significant development. Driven by the availability of extensive time series data, deep forecasting models, primarily based on NNs, have been exploited in applied industrial research divisions (Gasthaus et al., 2019; Laptev et al., 2017; Salinas et al. 2020; Wen et al., 2017).

The success of deep forecasting methods in competitions like the M4 competition (Smyl, 2020) has convinced even previously skeptical academics (Makridakis et al., 2018).

In the recent M5 competition, deep forecasting methods ranked second and third (Makridakis et al., 2021), despite tree-based methods like LightGBM (Ke et al., 2017) and XGBoost (Chen & Guestrin, 2016) dominating the competition (Januschowski et al., 2021).

2.2.3 Neural Networks in our Days

Modern software frameworks (Abadi et al., 2016; Chen et al., 2015; Paszke et al., 2019) have facilitated the development of NN models, with dedicated forecasting packages available (Alexandrov, 2020).

Contrary to traditional forecasting applications, modern scenarios often involve large sets of interrelated time series that require simultaneous forecasting (Januschowski & Kolassa, 2019).

Although these characteristics make them suitable for deep learning or neural networks (NNs), NNs were not always the standard approach for such problems, with historical views on their effectiveness being mixed (Zhang et al., 1998).

2.3 Probabilistic Forecasts for Hierarchical Time Series

2.3.1 Probabilistic Hierarchical Forecasting and Reconciliation

In recent years, the development of methods for hierarchical forecasting has been accompanied by growing interest in probabilistic forecasting, reflecting the increasing recognition of its importance. Consequently, various studies have sought to address the challenge of probabilistic hierarchical forecasting. Like point forecasting, methods for generating probabilistic hierarchical forecasts can be classified into bottom-up, top-down, and reconciliation approaches, though some algorithms integrate elements from multiple methodologies. Bayesian approaches for probabilistic forecast reconciliation are also discussed in the literature.

2.3.2 Bottom-up Approaches

The bottom-up methods for probabilistic hierarchical forecasting were initially proposed by Ben Taieb et al. (2017) and further developed by Ben Taieb et al. (2021). The core of this approach involves generating a Monte Carlo sample from each bottom-level variable's predictive distribution, ensuring independence by construction. These samples are then ranked and permuted to introduce dependence, with the permutations designed to ensure the empirical copula of the samples matches the copula of the residuals. This process exploits the hierarchical structure to avoid complications with high-dimensional copulas. The resulting samples are aggregated to yield a probabilistic forecast of the upper-level series.

An extension of this bottom-up approach incorporates top-level information by adjusting the mean of each series to align with a reconciled point forecast. For example, MinT reconciliation is used by Ben Taieb et al. (2021). One limitation of this approach is the need for equal-sized samples from both the predictive distribution and the training data, which can be restrictive in cases with limited training data. This issue is addressed by alternative approaches, such as quantile regression methods proposed by Panamtash and Zhou (2018) and Zhao et al. (2019), which estimate predictive quantiles directly, bypassing the need for Monte Carlo sampling.

2.3.3 Top-down Approaches

Panamtash and Zhou (2018) also introduced a top-down method for probabilistically coherent forecasts, wherein quantile forecasts are first generated for all series. These are then disaggregated based on the ratio between the forecast of a child node and its parent node. Another top-down method was proposed by Das et al. (2023), leveraging a combination of Long Short-Term Memory (LSTM) networks and multi-head self-attention to model proportions based on past data, generating samples from the predictive distribution of the top level, and disaggregating them using forecast proportions.

2.3.4 Two-step Reconciliation Approaches

Like point forecasting, progress has been made in two-step reconciliation methods for probabilistic forecasts. In temporal reconciliation, Jeon et al. (2019) proposed generating samples from each series' predictive distribution, stacking them into a matrix, and applying a projection matrix to reconcile the forecasts. One such method, the "ranked sample" approach, orders the observations from each distribution before reconciling quantiles - a concept previously explored by Shang and Hyndman (2017). This method performs well when there is high dependence between series.

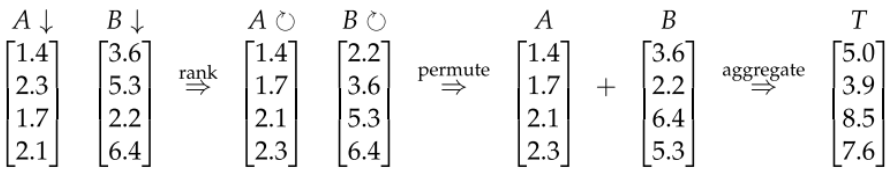


Figure 3 - Simplified example of the bottom-up approach for probabilistic forecasting.

The simplified example above illustrates the bottom-up approach for probabilistic forecasting in a basic 3-variable hierarchy where the total series TTT is composed of A+BA + BA+B. A sample of size K=4K = 4K=4 is drawn from the predictive distributions of the two bottom-level series, AAA and BBB. These samples are then ranked in ascending order and permuted to ensure that their empirical copula aligns with the copula of the residuals. Although the residuals are not explicitly shown, in this example, the smallest residual in series AAA corresponds to the second smallest in series BBB, and the second smallest in AAA corresponds to the smallest in BBB, and so forth. Finally, AAA and BBB are aggregated to generate a sample from the predictive distribution of the total series TTT.

2.3.5 Advances in Probabilistic Reconciliation

Recent contributions to probabilistic forecast reconciliation include the framework by Panagiotelis et al. (2023), which formalizes definitions of coherence and reconciliation for probabilistic forecasts. This framework allows the reconciliation of any set of base forecasts—univariate or multivariate—using any reconciliation method, with reconciliation weights trained via multivariate scoring rules. Rangapuram et al. (2021) further developed an end-to-end reconciliation method that optimizes probabilistic forecasts with respect to scoring rules, eliminating the two-step process.

Building upon this, cross-temporal reconciliation for probabilistic forecasts was explored by Girolimetto et al. (2023), who employed parametric and non-parametric methods to reconcile forecasts across time horizons. Despite recent advancements, there remain open questions regarding the theoretical underpinnings of reconciliation techniques in the probabilistic context. While the optimality of methods like MinT has been established for Gaussian distributions (Wickramasuriya, 2023), further research is required to determine the calibration and coverage properties of different reconciliation approaches in probabilistic forecasting.

2.3.6 Challenges of Forecasting in Retail

While the history of NNs in forecasting is rich, this work focuses on recent developments post the emergence of the term "deep learning." Retailers heavily rely on demand forecasts delivered through forecasting support systems (FSS) to plan, impacting organizational performance and efficiency along the retail supply chain (Fisher & Raman, 2018).

Forecast accuracy is particularly critical in low-margin, high-volume retailing, directly influencing profitability (Fisher & Raman, 2018). Forecasts serve as essential inputs for decision-making across various functional areas within organizations, including marketing, sales, production, purchasing, finance, and accounting. Additionally, forecasts underpin national, regional, and local distribution and replenishment plans.

The development and enhancement of forecasting models have received significant attention over the years, reflecting a shift in retail decision-making from intuition to data-driven approaches (Fisher & Raman, 2018). Forecasts play a pivotal role in supporting strategic, tactical, and operational decisions within retail organizations, each level focusing on different aspects while requiring compatibility.

Strategic decisions in retail encompass long-term planning within dynamic competitive and technological landscapes. Forecasting informs various elements of retail strategy, including market analysis, competitive positioning, and regulatory considerations (Levy et al., 2012). Tactical decisions, on the other hand, align with strategic objectives but pertain to more immediate concerns such as communication strategies, product offerings, and promotional activities at both chain and category levels.

At the category level, the focus shifts to maximizing profitability through pricing and promotional strategies tailored to category performance. Effective demand and supply planning processes are crucial for avoiding inventory issues, customer service disruptions, and unnecessary costs associated with obsolete products.

Successful forecasting enables retail managers to optimize staffing schedules based on anticipated customer activity and product demand, ensuring efficient store operations. Aggregate sales forecasting at the market level provides insights into changing market conditions, aiding retailers in strategic decision making and business model development to maintain market competitiveness (Alon et al., 2001).

2.3.7 Deep Learning in Forecasting for Retail

Deep learning has significantly impacted forecasting (Längkvist et al., 2014), with NNs becoming standard techniques (Hyndman & Athanasopoulos, 2018). New models, specifically designed for forecasting, leverage deep learning to enhance classical models or devise novel approaches.

Deep learning has emerged as a transformative approach in retail forecasting, particularly for hierarchical time series where predictions are required at various levels of aggregation, such as products, categories, or overall store performance. Traditional forecasting methods, like ARIMA, have been widely used due to their simplicity and effectiveness in certain contexts. However, these methods often struggle to model the complexity, nonlinearity, and interdependencies inherent in hierarchical retail data, especially when dealing with larger datasets or multiple aggregation levels (Crăcan, 2020).

Deep learning models have shown significant advantages in addressing these challenges, primarily due to their ability to handle complex patterns and scale effectively. For example, Transformer-based architectures such as the Temporal Fusion Transformer (TFT) and Informer have demonstrated superior performance by capturing intricate relationships and long-term dependencies in retail data. These models not only enhance prediction accuracy but also ensure coherence in forecasts across hierarchical levels, a critical requirement in retail operations (Li et al., 2023).

Another advantage of deep learning in this domain is its flexibility in integrating contextual information. Models like Long Short-Term Memory (LSTM) networks and Hierarchical Neural Additive Models (HNAM) excel at incorporating external variables, such as promotional activities, seasonal trends, and customer behavior. This integration enables forecasts to reflect real-world retail dynamics more accurately, thereby supporting better strategic and operational decisions (Crăcan, 2020; Taylor, Huang, & Kumar, 2024). Moreover, HNAMs offer the added benefit of interpretability, bridging the gap between the simplicity of traditional statistical models and the complexity of advanced neural networks. This is particularly valuable in retail contexts, where understanding the drivers of forecasted values is often as important as the predictions themselves (Taylor et al., 2024).

Hybrid approaches combining traditional statistical methods with deep learning have also been explored to maximize forecasting performance. For instance, ARIMA models can be used to generate baseline forecasts, while neural networks refine these predictions by capturing nonlinear patterns and interactions that ARIMA cannot model effectively. Such hybrid systems have been shown to outperform standalone models, leveraging the strengths of each method (Crăcan, 2020).

The practical applications of deep learning models in retail are extensive, ranging from demand forecasting to inventory management. By incorporating external factors like promotional schedules into their predictions, these models help retailers anticipate demand spikes more accurately, thereby reducing the costs associated with overstocking or stockouts. This enhanced accuracy has tangible benefits for optimizing supply chain operations and improving overall efficiency (Li et al., 2023).

Despite their advantages, deep learning models are not without challenges. They require significant computational resources and careful parameter tuning, which can limit their accessibility for smaller businesses or datasets with limited historical depth. Additionally, the complexity of these models may pose a steep learning curve for practitioners accustomed to traditional methods (Taylor et al., 2024).

In conclusion, deep learning has proven to be a powerful tool for hierarchical time series forecasting in retail, offering substantial improvements in accuracy, scalability, and flexibility. By capturing complex relationships and incorporating contextual information, these models enable retailers to make more informed decisions across the supply chain. When adopted thoughtfully, they represent a significant advancement over traditional methods, highlighting their growing importance in both academic research and practical applications (Crăcan, 2020; Li et al., 2023; Taylor et al., 2024).

3 Methodology

3.1 Hierarchical Time Series Forecasting

An hierarchical time series is a collection of multiple time series that are interconnected through known linear relationships, often representing aggregation structures. In such a hierarchy, higher-level series are sums of lower-level series, adhering to specific aggregation constraints. Consider the hierarchical structure depicted in Figure 4. At the top of the hierarchy (level 0) is the most aggregated series, referred to as the "Total" series. This series is subsequently disaggregated into two series, A and B, which comprise level 1. Each of these series is further divided into three sub-series, resulting in the bottom level of the hierarchy (level 2).

The value observed at time t for series i is denoted as $y_{i,t}$, where $t = 1,\ldots,T$. In this example, the total number of series is n, while the number of bottom-level series is m, for each time period t.

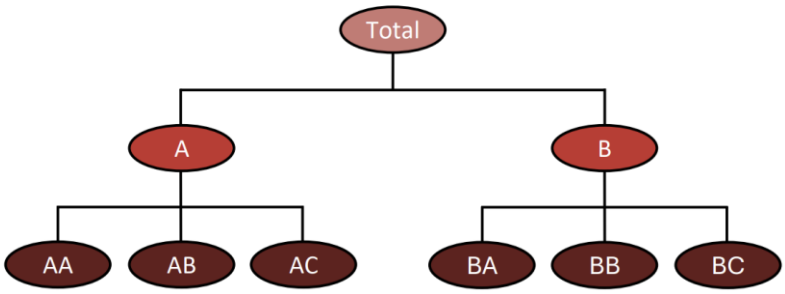


Figure 4 - 2-level hierarchical tree structure.

For each time period t, the observations in this hierarchical structure are related by the following aggregation constraints:

$$y_{Total,t} = y_{AA,t} + y_{AB,t} + y_{AC,t} + y_{BA,t} + y_{BB,t} + y_{BC,t},$$

$$y_{A,t} = y_{AA,t} + y_{AB,t} + y_{AC,t}, \quad y_{B,t} = y_{BA,t} + y_{BB,t} + y_{BC,t}.$$

Let y_t represent the vector containing the t-th observations of all series in the hierarchy and let b_t represent the vector containing the t-th observations of the bottom-level series.

Define S as the summing matrix of order $n \times m$, which shows how the bottom-level series aggregate to form higher-level series. So, the aggregation constraints can be expressed in matrix form as follows:

$$y_t = Sb_t.$$

For the structure illustrated in Figure 4, the aggregation equation can be expanded as:

$$\begin{bmatrix} y_{Total,t} \\ y_{A,t} \\ y_{B,t} \\ y_{AA,t} \\ y_{AB,t} \\ y_{AC,t} \\ y_{BA,t} \\ y_{BB,t} \\ Y_{BC,t} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} y_{AA,t} \\ y_{AB,t} \\ y_{AC,t} \\ y_{BA,t} \\ y_{BB,t} \\ y_{BC,t} \end{bmatrix}.$$

The objective is to generate coherent forecasts for each series within the hierarchy, ensuring that they are consistent with the hierarchical structure and satisfy the aggregation constraints.

Let $\tilde{y}_{t+h|t}$ represent the vector of h-step-ahead forecasts (where h =1, 2, …) for all series in the hierarchy, based on observations up to time t.To achieve coherence among forecasts, a reconciliation method must be applied:

$$\tilde{y}_{t+h|t} = SP\hat{y}_{t+h|t},$$

where P is an $m \times n$ matrix that maps the base forecasts $\hat{y}_{t+h|t}$ into reconciled forecasts at the bottom level. These forecasts are aggregated using the summing matrix S to generate coherent forecasts $\tilde{y}_{t+h|t}$. The choice of the matrix P depends on the selected reconciliation method. For the bottom-up method, $P = \left[0_{m \times (n-m)} | I_m\right]$, where $0_{m \times (n-m)}$ is the null matrix of order $m \times (n-m)$ and I_m is de identity matrix of order m. For the hierarchical structure depicted in Figure 4, the matrix is:

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

For the top-down method, is defined as $P = \left[p \,|\, 0_{m \times (n-1)} \right]$, where p is a vector of length m containing the proportions used to disaggregate the top-level forecast into bottom-level forecasts. These proportions are typically calculated using historical data. For the hierarchical structure depicted in Figure 4, this matrix P is:

$$P = \begin{bmatrix} p1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ p2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ p3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ p4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ p5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ p6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Traditional approaches for calculating disaggregation proportions, as proposed by Gross and Sohl (1985), use historical data to allocate forecasts. In the first method, each proportion p_i is computed as the average of the historical ratios of the bottom-level series $y_{i,t}$ to the top-level series $y_{Total,t}$ over the period:

$$p_i = \frac{1}{T} \sum_{t=1}^{T} \frac{y_{i,t}}{t_{Total,t}}, \; i = 1, \ldots, m.$$

In the second approach, each proportion p_i is determined by calculating the ratio of the mean historical values of the bottom-level series $y_{i,t}$ to the mean historical value of the top-level series $y_{Total,t}$ over the corresponding time period:

$$p_i = \frac{\frac{1}{T}\sum_{t=1}^{T} y_{i,t}}{\frac{1}{T}\sum_{t=1}^{T} y_{Total,t}}, i=1, \ldots, m.$$

These top-down approaches are appreciated for their simplicity; however, they are inherently static and do not incorporate potential temporal variations in the proportional relationships. As a result, these methods often yield less precise forecasts at the lower levels of the hierarchy in comparison to the bottom-up approach. To mitigate this limitation, Athanasopoulos et al. proposed an advanced top-down approach that utilizes forecast-derived proportions, thereby improving the adaptability and accuracy of the hierarchical forecasting process:

$$p_i = \prod_{l=0}^{k-1} \frac{\widehat{y}^{(l)}_{i,t+h|t}}{\widehat{S}^{(l+1)}_{i,t+h|t}}, \quad i=1,\ldots,m,$$

where k represents the number of levels in the hierarchy. In this formulation, $\widehat{y}^{(l)}_{i,t+h|t}$ denotes the h-step-ahead base forecast for the series corresponding to the node l levels above node i, and $\widehat{S}^{(l+1)}_{i,t+h|t}$ is the sum of the h-step-ahead base forecasts for the series associated with the nodes $l+1$ levels above node i. Hyndman et al. proposed an optimal reconciliation method based on a regression model:

$$\widehat{y}_{t+h|t} = S\beta_{t+h|t} + \varepsilon_h,$$

where $\beta_{t+h|t}$ denotes the vector of unknown means for the most disaggregated series, and ε_h represents the reconciliation error, which has a mean of zero and a covariance matrix $\sum_h$. Given that $\sum_h$ is known, the Generalized Least Squares (GLS) estimator of $\beta_{t+h|t}$ can be employed to produce the following reconciled forecasts:

$$\tilde{y}_{t+h|t} = S\widehat{\beta}_{t+h|t} = SP\widehat{y}_{t+h|t} = S\left(S'\sum_h^{-1}S\right)^{-1}S'\sum_h^{-1}\widehat{y}_{t+h|t}.$$

Hyndman et al. established that if the base forecasts, denoted as $\widehat{y}_{t+h|t}$, are unbiased, then the reconciled forecasts, represented as $\tilde{y}_{t+h|t}$, will also be unbiased, provided that the condition $SPS = S$ is satisfied. This condition holds true for both the optimal

reconciliation approach and the bottom-up method. However, no top-down method fulfills this requirement, indicating that top-down approaches inherently lead to biased reconciled forecasts. Wickramasuriya et al. further argued that the reconciliation approach proposed by Hyndman et al. is generally impractical, given that $\sum_h$ is typically unknown and cannot be reliably estimated. According to these authors, the covariance matrix of the h-step-ahead reconciled forecast errors can be expressed as follows:

$$Var\left[y_{t+h} - \tilde{y}_{t+h|t}\right] = SPW_h P'S',$$

where $W_h = Var\left[y_{t+h} - \hat{y}_{t+h|t}\right]$ is the variance-covariance matrix of the h-step-ahead base forecast errors.

Our objective is to determine the matrix P that minimizes the error variances of the reconciled forecasts, represented by the diagonal elements of $Var\left[y_{t+h} - \hat{y}_{t+h|t}\right]$. The optimal reconciliation approach, referred to as the MinT (Minimum Trace) method, was introduced by Wickramasuriya et al.. This method identifies the P matrix that minimizes the trace of $Var\left[y_{t+h} - \hat{y}_{t+h|t}\right]$, while ensuring that the condition $SPS = S$= S is satisfied. The resulting is given by:

$$P = \left(S'W_h^{-1}S\right)^{-1}S'W_h^{-1}.$$

Consequently, the reconciled forecasts obtained using the MinT approach are:

$$\tilde{y}_{t+h|t} = S\left(S'W_h^{-1}S\right)^{-1}S'W_h^{-1}\hat{y}_{t+h|t} \quad .$$

This optimal reconciliation approach still necessitates the estimation of W_h. Wickramasuriya et al. proposed several alternatives for achieving this estimation:

1. $W_h = k_h I_{n,} \forall h$ where $k_n > 0$. Under this condition, the estimator for $\beta_{t+h|t}$ corresponds to the Ordinary Least Squares (OLS) estimator. While this represents the simplest estimation method, the resulting P matrix is independent of the data, implying it does not consider variations in scale across hierarchical levels or the interrelationships between the series. This particular specification is known as OLS.

2. $W_h = k_h diag\left(\widehat{W}_1\right), \forall h$ where $k_h > 0$ and $\widehat{W}_1 = \frac{1}{T}\sum_{t=1}^{T} e_t e'_t$ represents the sample covariance estimator of the one-step-ahead base forecast errors. This approach scales the base forecasts based on the variance of the residuals e_t. The resulting MinT estimator is known as the Weighted Least Squares (WLS) (var) estimator.
3. $W_h = k_h \wedge, \forall h$ where $k_h > 0, \wedge = diag(S1)$ with 1 being a unit vector of dimension m. This approach assumes that the variance of the base forecast errors at the bottom level is k_h, and that these errors are uncorrelated across the nodes. Unlike other methods, this estimator is based purely on the structural aggregation constraints of the hierarchy rather than relying on data, which makes it especially suitable when residuals are not available. This technique is known as structural scaling and is represented as WLS (struct).
4. $W_h = k_h \widehat{W}_1, \forall h$, where $k_h > 0$, represents the sample covariance estimator for $h = 1$. While this estimator is relatively simple to calculate, it may be inappropriate when the number of bottom-level series m is greater than the number of time periods T. This approach is referred to as MinT (sample).
5. $W_h = k_h \widehat{W}_{1,D}, \forall h$, where $k_h > 0$, represents a shrinkage estimator. In this context, $\widehat{W}_{1,D} = \lambda_D \widehat{W}_{1,D} + \left(1 - \lambda_D\right)\widehat{W}_1$ is formulated to shrink the off-diagonal elements of $\widehat{W}_1$ towards zero while retaining the original diagonal values. Here, Wc1,D is a diagonal matrix containing the diagonal elements of Wc1, and λ_D represents the shrinkage intensity parameter. Under the assumption of constant variances, Schäfer and Strimmer proposed the following formula for calculating the shrinkage intensity parameter:

$$\widehat{\lambda}_D = \frac{\sum_{i \neq j} \widehat{Var\left(\hat{r}_{ij}\right)}}{\sum_{i \neq j} \hat{r}^2_{ij}},$$

where $\hat{r}_{ij}$ represents the ijth element of $\widehat{R}_1$, the sample correlation matrix of the one-step-ahead base forecast errors. This approach is referred to as MinT (shrink).

3.2 Time Series Deep Learning Models

3.2.1 MLP

Deep Learning algorithms represent a subset within the broader Machine Learning domain, modeled to resemble neural processes observed in the human brain. This approach employs multi-layered neural networks, which sequentially transform data to identify optimal representations for decision-making or predictive tasks. Learning within these networks occurs through iterative adjustments in the connections between neurons.

The Multi-Layer Perceptrons (MLPs) are a foundational type of deep neural network. These architectures consist of an input layer for data intake, hidden layers where learning occurs, and an output layer that produces the predictions. Each neuron in a given layer is fully connected to all neurons in both the preceding and following layers. The design complexity of MLPs, especially the number and arrangement of layers and neurons, directly affects the network's capacity to learn intricate patterns from data.

To enable non-linear modeling capabilities, activation functions are applied to the outputs of neurons after transformation. Among the most common are the Rectified Linear Unit (ReLU), which allows only positive values and mitigates vanishing gradients to optimize learning, the sigmoid function, which maps values between 0 and 1 and is often applied in probability-related tasks despite its susceptibility to vanishing gradients, and the hyperbolic tangent (tanh), which maps values between -1 and 1 to add another dimension of non-linearity.

In training, MLPs rely on adjusting parameters - weights and biases - to minimize prediction errors, quantified by a loss function such as Mean Squared Error (MSE). Stochastic Gradient Descent (SGD) is a popular optimization technique used here, wherein parameters are iteratively adjusted based on a randomly selected subset (minibatch) of data. Batch normalization often accompanies this process, stabilizing learning by normalizing inputs for each layer to improve efficiency.

Effective MLPs must generalize well to new data, balancing complexity to avoid both underfitting, where model complexity is insufficient, and overfitting, where adaptation to training data is excessive. Overfitting is commonly managed through regularization techniques like L1/L2 penalties, dropout, and early stopping, which help maintain model robustness across diverse datasets.

3.2.2 NBEATS

The N-BEATS model is a deep learning-based framework for univariate time series forecasting that offers both flexibility and interpretability, addressing key challenges associated with classical statistical methods. The model architecture, initially proposed by Oreshkin et al. (2020), is designed to be generic, making use of residual connections, a deep stack of fully connected layers, and no time-series-specific feature engineering, thus aiming to provide state-of-the-art performance across a wide range of forecasting tasks.

The N-BEATS architecture is built upon a stack of basic building blocks, each of which contains multiple fully connected layers. Each block is equipped with forward and backward residual links, enabling the model to efficiently approximate both the current state and future values of a time series. Specifically, the forward operation produces the forecast output, which estimates the future values, while the backward operation generates the backcast component, which aims to eliminate irrelevant parts of the input, facilitating the optimization of subsequent blocks (Oreshkin et al., 2020).

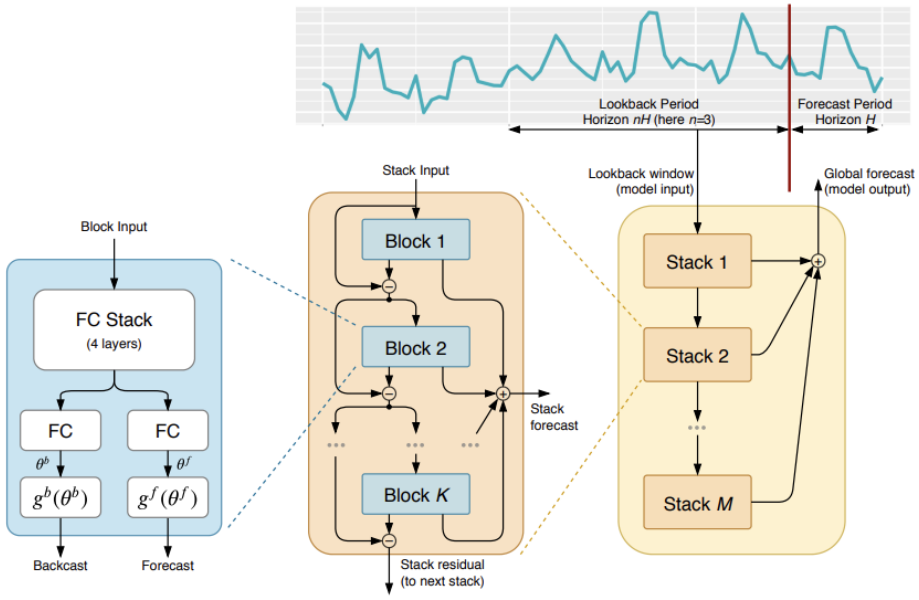


Figure 5 - N-BEATS model architecture.

The architecture as presented in figure 6, features a stacking mechanism in which blocks are connected in sequence to form stacks. This stacking strategy improves the model's overall capacity to represent complex time series data while leveraging both short-term

and long-term dependencies through residual connections (He et al., 2016). The residual stacking approach, inspired by the classical residual network designs, ensures efficient backpropagation of gradients, thereby mitigating issues related to vanishing gradients often encountered in deep architectures.

A significant feature of the N-BEATS model is its ability to produce interpretable outputs. This is achieved by incorporating structured inductive biases through specialized stacks that decompose the forecast into trend and seasonality components. The trend stack utilizes polynomial functions to capture slowly varying or monotonic behaviors, while the seasonality stack employs Fourier bases to approximate cyclical, recurring fluctuations in the data. This decomposition offers a transparent representation of the forecast, like classical decomposition methods, and enables practitioners to better understand the underlying temporal dynamics (Cleveland et al., 1990; Makridakis et al., 2018).

Empirical evaluations conducted on prominent datasets such as M3, M4, and TOURISM have demonstrated the efficacy of N-BEATS, with results indicating that the model consistently achieves superior performance compared to traditional statistical and hybrid methods. The model was shown to outperform both pure statistical methods and domain-adjusted hybrid approaches, reducing forecast error by up to 11% over statistical benchmarks and 3% over the M4 competition winner (Makridakis et al., 2020). Furthermore, the model's generic version achieved these results without relying on specialized pre-processing or domain knowledge, suggesting that deep learning primitives, when properly structured, are sufficient for solving a wide range of forecasting problems (Oreshkin et al., 2020).

N-BEATS architecture represents a significant advancement in time series forecasting by providing a model that is not only powerful in terms of predictive accuracy but also interpretable. The generic nature of its architecture allows it to be applied across multiple domains without requiring time-series-specific components, and the introduction of structured stacks enables it to provide outputs that align well with practitioner needs for transparency and interpretability.

3.2.3 N-HiTS

N-HiTS (Neural Hierarchical Interpolation for Time Series) is a novel model designed to improve both the accuracy and efficiency of long-horizon time series forecasting, which

typically faces challenges such as high computational complexity and prediction volatility (Challu et al., 2023). The N-HiTS model presents several unique characteristics that contribute to its superior performance.

N-HiTS employs a multi-rate data sampling approach that utilizes subsampling layers to significantly reduce memory usage and computational demands while maintaining the model's ability to capture long-range dependencies. This multi-rate data sampling approach ensures that each layer in the model is focused on specific scales of the time series, thereby allowing the model to effectively concentrate on both low and high frequency components within the data (Challu et al., 2023).

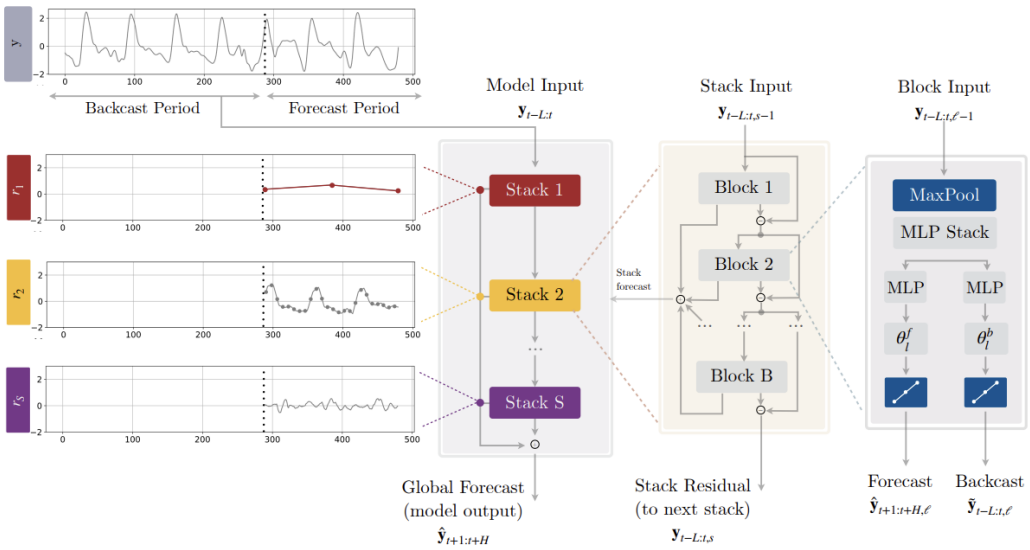


Figure 6 - N-HiTS model architecture.

Another critical aspect of the N-HiTS model is its hierarchical interpolation mechanism. This method allows for the efficient approximation of arbitrarily long horizons while controlling model expressiveness and computational load. By enforcing smoothness and decomposing the input signal into different frequency bands, the hierarchical interpolation technique allows the model to generate consistent and accurate predictions across a wide range of time scales (Challu et al., 2023). This innovative approach to interpolation facilitates a sequential construction of forecasts, addressing the common issue of volatility observed in long-term predictions.

N-HiTS builds upon the previously established N-BEATS framework, enhancing it through the incorporation of hierarchical and multi-scale techniques. The architecture is

composed of multiple stacks of blocks, each specialized in different frequency bands of the signal. This hierarchical organization of blocks enables the model to effectively specialize and aggregate its predictions, resulting in a system that is both interpretable and modular, capable of generating efficient and accurate forecasts (Oreshkin et al., 2020).

Empirical results demonstrate that N-HiTS provides state-of-the-art performance in long-horizon time series forecasting. When compared to Transformer-based approaches, N-HiTS achieves a 20% improvement in prediction accuracy while also reducing computational complexity by a factor of 50. This substantial improvement can be attributed to the model's ability to adapt its components dynamically to the frequency-specific features of the input data, ensuring precise and efficient long-term forecasts (Challu et al., 2023).

The effectiveness of N-HiTS has been validated across a diverse range of benchmark datasets, including electricity transformer temperature, exchange rates, traffic patterns, weather data, and influenza-like illness records. Its superior performance, particularly in long-horizon and multivariate time series contexts, positions N-HiTS as a promising solution for applications in infrastructure planning, health monitoring, and resource management (Churpek et al., 2016; Field et al., 2012).

3.2.4 Vanilla Transformer

The Transformer model, as articulated by Vaswani et al. (2017), represents a groundbreaking approach to sequence transduction tasks, such as machine translation and language modelling. Unlike conventional architectures that rely on recurrent neural networks (RNNs) or convolutional neural networks (CNNs), the Transformer architecture as presented in Figure 8 is based entirely on self-attention mechanisms.

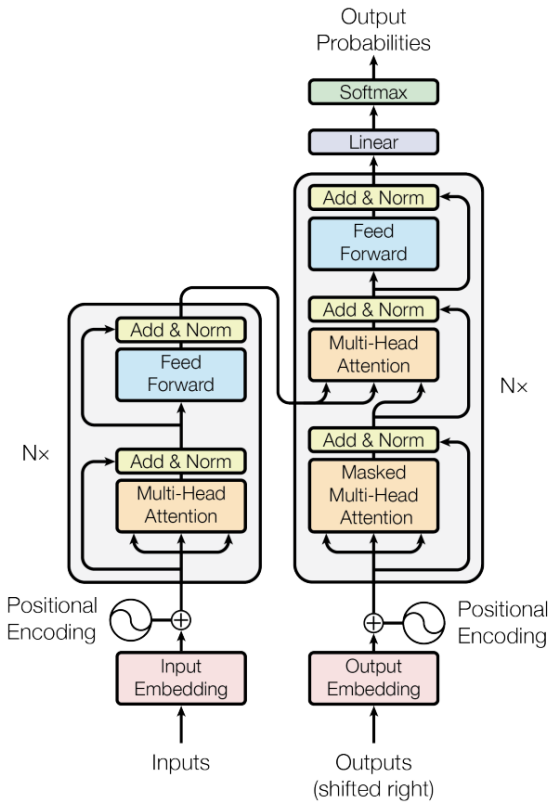


Figure 7 - Transformer model architecture.

This fundamental shift results in enhanced parallelization capabilities and significantly reduced training times. In comparative evaluations, this approach has demonstrated superior computational efficiency and performance, particularly when contrasted with previous encoder-decoder frameworks that incorporated attention mechanisms alongside recurrence or convolution.

The Transformer adheres to an encoder-decoder architecture. The encoder component transforms an input sequence into a continuous representation, while the decoder utilizes this representation to generate the corresponding output sequence. Both the encoder and decoder comprise N=6 identical layers, each including a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. Additionally, the decoder layers include a third sub-layer that executes multi-head attention over the encoder output, facilitating the integration of encoded information into the output sequence generation (Vaswani et al., 2017).

At the core of the Transformer is the scaled dot-product attention mechanism, wherein the elements are represented as queries, keys, and values. The output is computed as a

weighted sum of the values, with the weights determined by evaluating the compatibility of the query with the corresponding keys, scaled by the square root of their dimension. This scaling factor mitigates the risk of excessively large dot products, which could push the softmax function into regions characterized by very small gradients. Multi-head attention extends this concept by applying several distinct linear projections to the queries, keys, and values, which allows the model to attend to different segments of the input representation concurrently, thereby enhancing its capacity to capture various linguistic features (Vaswani et al., 2017).

Transformer also incorporates position-wise feed-forward networks in each encoder and decoder layer. These networks involve two linear transformations, separated by a ReLU activation, which are applied independently to each position within the sequence. Such architecture facilitates parallelization and improves computational efficiency. Residual connections (He et al., 2016) and layer normalization (Ba et al., 2016) are also employed in each sub-layer to ensure training stability and enhance model performance.

The absence of recurrence or convolution in the Transformer raises the challenge of providing the model with information about the sequence's positional structure. Vaswani et al. (2017) addressed this issue through the introduction of positional encodings, which are added to the input embeddings to supply the model with information regarding the relative or absolute positions of tokens within the sequence. These positional encodings are based on sine and cosine functions at varying frequencies, enabling the model to generalize effectively to longer sequences beyond those observed during training.

Empirical evaluations have demonstrated that the Transformer significantly outperforms preceding models in machine translation tasks. On the WMT 2014 English-to-German translation task, the Transformer achieved a BLEU score of 28.4, surpassing the existing state-of-the-art, including ensemble models, by over 2 BLEU. Furthermore, it established a new single-model state-of-the-art BLEU score of 41.8 on the WMT 2014 English-to-French translation task, after only 3.5 days of training on eight GPUs, highlighting both the effectiveness and efficiency of the architecture (Vaswani et al., 2017).

3.2.5 TFT

The Temporal Fusion Transformer (TFT) is a state-of-the-art deep learning model specifically designed to address the challenges associated with multi-horizon time series forecasting, while balancing both predictive accuracy and interpretability. Lim et al.

(2021) introduce TFT as an innovative solution that effectively integrates both static and time-varying covariates to generate forecasts across multiple time steps, enhancing its applicability across various domains such as retail, healthcare, and economics.

The TFT model leverages a combination of mechanisms to address the limitations of traditional deep learning models, particularly regarding their "black box" nature. The primary components of the TFT architecture include attention mechanisms, static covariate encoders, variable selection networks, gating mechanisms, and quantile regression. Each of these components contributes to improving the model's performance and interpretability.

The attention mechanism utilized in the TFT is an interpretable multi-head self-attention layer, which serves to capture long-term dependencies between temporal inputs. This mechanism facilitates the identification of the most relevant time steps for forecasting, enhancing the model's interpretability by providing insights into which features are most influential at specific points in time (Vaswani et al., 2017; Lim et al., 2021). Unlike conventional attention-based models, the TFT's attention mechanism is designed to provide more interpretable outputs by sharing values across different heads, thereby simplifying the analysis of temporal dependencies.

Static covariate encoders are employed within the TFT to effectively incorporate static metadata into the forecasting process. These encoders use Gated Residual Networks (GRNs) to generate context vectors, which are subsequently utilized throughout the model to condition the temporal dynamics. This approach enables the integration of static and time-varying features in a coherent manner, ultimately improving the model's ability to leverage diverse data sources for forecasting (Lim et al., 2021).

The TFT also incorporates instance-wise variable selection networks to dynamically select relevant input features at each time step. This capability ensures that only the most salient features are used in prediction, enhancing both the model's interpretability and its robustness by mitigating the effects of noisy or irrelevant inputs. Gating mechanisms based on Gated Linear Units (GLUs) are used to dynamically skip over unnecessary components within the network, enabling the model to adapt its complexity to different datasets. These components collectively contribute to the flexibility and efficiency of the TFT architecture.

To address the need for predictive intervals, the TFT utilizes quantile regression to provide interval forecasts at different quantiles, such as the 10th, 50th, and 90th percentiles. This approach enables the model to generate not only point forecasts but also uncertainty estimates, providing a more comprehensive understanding of the range of possible future outcomes (Wen et al., 2017). Such quantile-based forecasting is particularly useful in applications where risk assessment and decision-making are critical.

The temporal processing capabilities of the TFT combine sequence-to-sequence layers for capturing localized temporal patterns with self-attention layers for modeling long-term dependencies. This hybrid approach allows the model to effectively learn from both short-term and extended temporal relationships, providing a more nuanced understanding of the temporal dynamics inherent in the data. A static enrichment layer is also incorporated to enhance temporal features with static metadata, which improves the model's ability to account for interactions between static and dynamic inputs.

The empirical evaluation of the TFT demonstrates significant performance gains over existing benchmarks across various real-world datasets, including electricity consumption, traffic forecasting, retail sales, and financial volatility prediction. Specifically, the TFT exhibits an average reduction in prediction error of 7% for median predictions (P50) and 9% for upper confidence interval estimates (P90) when compared to the next best models (Lim et al., 2021). Furthermore, the interpretability of the TFT is showcased through several practical use cases, such as identifying important variables, visualizing persistent temporal patterns (e.g., seasonality), and detecting significant events indicative of regime changes.

The Temporal Fusion Transformer presents a comprehensive solution for multi-horizon forecasting, combining high predictive accuracy with interpretability. The integration of attention mechanisms, static covariate encoders, and variable selection networks within the model architecture makes it a versatile tool for complex temporal prediction tasks. The TFT's ability to provide insights into the temporal dynamics of the data, alongside its performance advantages, establishes it as a valuable model for practical forecasting applications.

3.2.6 Informer

The Informer model is a transformer-based architecture developed to tackle the distinct challenges inherent in Long Sequence Time-Series Forecasting (LSTF). Traditional

methods often face significant limitations in terms of computational complexity, memory consumption, and inference speed when dealing with long sequences. Informer introduces innovative solutions to address these limitations, making it an efficient and scalable choice for extended sequence forecasting.

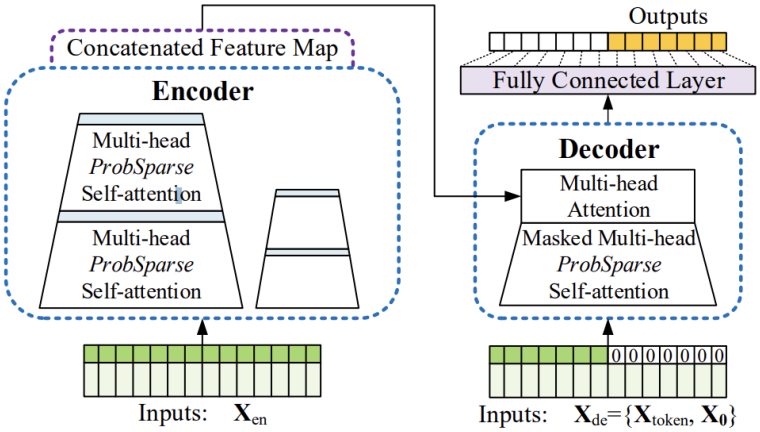


Figure 8 - Informer model architecture.

The Informer model utilizes ProbSparse self-attention, which replaces the standard self-attention mechanism found in traditional transformers. This mechanism reduces computational complexity by selectively focusing on the most important query-key pairs [Kitaev et al., 2019; Zhou et al., 2021]. By employing this selective attention mechanism, the model can process long sequences more efficiently while maintaining the ability to capture significant dependencies. This innovation results in a considerable reduction in computational burden, making Informer particularly well-suited for LSTF tasks.

To further optimize performance, Informer incorporates a self-attention distilling mechanism. This process emphasizes dominant attention scores while eliminating less relevant information, which results in cascading reductions of input dimensions across layers. Consequently, the overall space complexity is reduced significantly lowering the memory requirements [Zhou et al., 2021]. This reduction allows the model to efficiently handle long sequence inputs, which would otherwise be computationally prohibitive for standard transformer architectures.

Additionally, Informer employs a generative-style decoder, deviating from the conventional step-by-step decoding approach. Instead, the model predicts the entire output sequence in a single forward operation, drastically improving inference speed

[Sutskever et al., 2014; Zhou et al., 2021]. This ability to perform predictions in a single pass makes Informer particularly effective for real-time applications, as it minimizes the latency associated with generating long sequence forecasts.

3.2.7 PatchTST

The Patch Time Series Transformer (PatchTST) model, introduced by Nie et al. (2023), presents an efficient architecture for enhancing multivariate time series forecasting. The key innovations in this model are the implementation of patching and channel-independence, which collectively facilitate effective long-term forecasting with optimized computational resources.

PatchTST aims to forecast future time series values based on historical data, defined by a look-back window. The core of the model is a Transformer encoder, which effectively captures temporal dependencies across time steps (Vaswani et al., 2017). The model processes multivariate inputs by splitting them into separate univariate time series, and each channel is handled independently. This channel-independence reduces complexity and ensures that information from unrelated channels does not interfere, which enhances the learning process (Zeng et al., 2022).

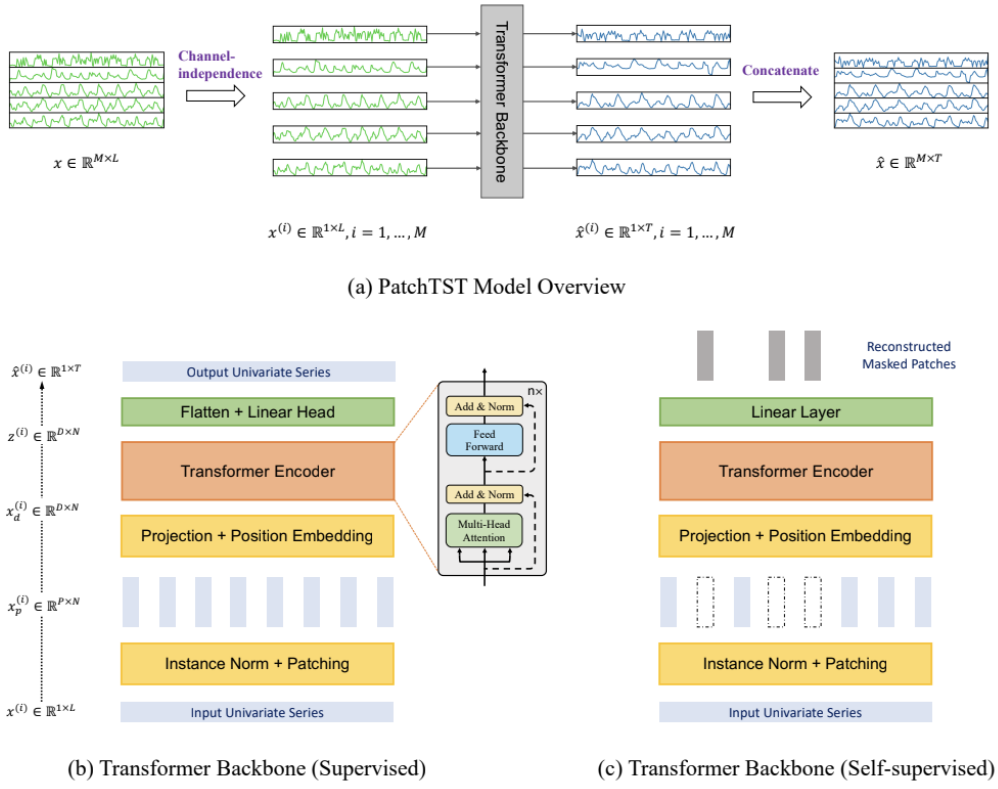


Figure 9 - Patch TST model architecture.

A significant innovation of the PatchTST model is the patching mechanism. The input time series is segmented into smaller patches, which helps retain local semantic information while significantly reducing the number of input tokens. This leads to decreased memory usage and computational demands, making it possible to utilize longer historical sequences effectively (Dosovitskiy et al., 2021). The Transformer encoder then processes these patches through multi-head self-attention to generate latent representations, which allows the model to capture relationships within the time series. Furthermore, a learnable position embedding is applied to maintain the temporal order of the patches (Vaswani et al., 2017).

The PatchTST model demonstrates superior performance in long-term forecasting compared to other state-of-the-art models. The combination of patching and channel-independence enables the model to achieve greater accuracy while maintaining computational efficiency, especially when using extended historical sequences (Nie et al., 2023). This balance between efficiency and performance makes PatchTST particularly suitable for applications requiring robust time series forecasting.

3.2.8 Benchmarks (ARIMA/ETS/Seasonal Naive/Naive)

The provided document, titled "Evaluating the Effectiveness of Time Series Transformers for Demand Forecasting in Retail" by Oliveira and Ramos (2024), explores the use of Transformer-based models for retail demand forecasting, specifically comparing newer models like the vanilla Transformer, Informer, Autoformer, PatchTST, and Temporal Fusion Transformer (TFT) with traditional forecasting methods like AutoARIMA and AutoETS.

The models were evaluated based on their performance using the mean absolute scaled error (MASE) and weighted quantile loss (WQL). The study employed the M5 competition dataset, consisting of 30,490 time series from 10 stores across the United States. The results demonstrated that Transformer-based models significantly outperform traditional models in both short-term and long-term forecasting scenarios. Specifically, Transformers improved MASE by 26% to 29% and reduced WQL by up to 34% compared to traditional seasonal Naïve models, excelling in accuracy for point and probabilistic forecasts.

While models like Autoformer and PatchTST also surpassed traditional models, their performance was slightly lower than that of vanilla Transformer, Informer, and TFT, indicating room for improvement through further tuning. The study highlights the potential trade-off between model complexity and computational efficiency, noting that although Transformer models are computationally intensive, they offer significant gains in forecasting accuracy compared to traditional methods like AutoARIMA, which are notably slower.

4 Empirical Study

4.1 Dataset

To ensure the robustness and generalizability of the research findings, it is essential that the results can be independently verified and compared with related studies. Therefore, this research utilizes the widely acknowledged and publicly available M5 competition dataset, providing a solid foundation for our analysis. The M5 competition served as a rigorous benchmark for evaluating forecasting methodologies, requiring the generation of both precise point estimates and probabilistic intervals for hierarchical time series data. Given its focus on Walmart, the global leader in retail revenue, the competition presented a particularly high-stakes and challenging context for methodological validation.

The M5 dataset is organized hierarchically and includes 3,049 distinct products categorized into three primary classes: Foods, Hobbies, and Household goods. These main categories are further divided into seven product departments: Foods1, Foods2, Foods3, Hobbies1, Hobbies2, Household1, and Household2. Sales data for these products were collected from ten retail stores across three U.S. states: California (CA), Texas (TX), and Wisconsin (WI). Specifically, California includes four stores (CA1, CA2, CA3, CA4), Texas includes three stores (TX1, TX2, TX3), and Wisconsin also comprises three stores (WI1, WI2, WI3). The dataset spans a period of approximately 5.4 years, encompassing daily sales records from January 29, 2011, to June 19, 2016, resulting in a total of 1,969 daily observations.

The full dataset is organized by aggregation and has a total of 42840 series. Due to the lack of computational power available for this study, the total of series was reduced to a manageable dimension of 1288 series. The reduction was made to preserve the characteristics of the dataset.

The empirical study was conducted in a development environment utilizing Google Colab and the NeuralForecast package by NIXTLA.

hierarchy_levels =		Reduced Number of series
[['Total'],		1
	['Total', 'State'],	3
	['Total', 'State', 'Store'],	10
	['Total', 'Category'],	3
	['Total', 'Category', 'Department'],	7
	['Total', 'State', 'Category'],	9
	['Total', 'State', 'Category', 'Department'],	21
	['Total', 'State', 'Store', 'Category'],	30
	['Total', 'State', 'Store', 'Category', 'Department'],	70
	['Total', 'Category', 'Department', 'Product']]	81
	['Total', 'State', 'Product']]	243
	['Total', 'State', 'Store', 'Category', 'Department', 'Product']]	810
		1288

Table 1 - M5 reduced Dataset - Hierarchical levels.

The preprocessing phase involved loading the raw data and adapting it to serve a hierarchical structure suitable for time series forecasting. Specifically, the dataset was first loaded into a pandas DataFrame, followed by structuring it into meaningful categorical variables that represented different levels of aggregation.

The main preprocessing step involved creating multiple hierarchical layers to encompass different perspectives in forecasting, including geographic and product attributes. These were used to define hierarchical relations such as "Total", "State", "Store", "Category", "Department", and "Product" identifiers. This structured the data into multiple levels, ensuring that forecasts generated for individual series could be reconciled appropriately across aggregated levels. The data preparation process also involved splitting the dataset into training and testing subsets, with a forecast horizon of 28 time periods.

This forecast horizon was critical as it defined the periods over which models were trained and subsequently evaluated. After preprocessing, the hierarchical data was saved in appropriate formats for further analysis and model fitting.

4.2 Hierarchical and Grouped Time Series

To accurately model the time series, the M5 dataset was structured into a hierarchical format that captures the inherent groupings within the data, reflecting different dimensions like geographic locations and product lines. Hierarchical forecasting allows for a coherent structure where the time series data can be forecasted at different levels of aggregation, ranging from the most granular product-level forecasts to the total aggregated sales at the highest level. This hierarchical representation was essential for

improving the accuracy of forecasts while maintaining the consistency of predictions across levels. Grouped time series allow for both hierarchical and non-hierarchical groupings, where each series may be aggregated by categories of interest, such as geographic divisions or product types, without strictly following a tree-like hierarchy. In this study, a combination of hierarchical and grouped structures was used, thereby capturing a more complete representation of dependencies within the dataset.

The construction of the hierarchical dataset involved applying aggregation and disaggregation techniques to create relationships among the time series, which was essential for consistent forecasting at different levels. Reconciliation methods were used for aligning the forecasts from different levels, ensuring that the bottom-level forecasts summed to match higher-level forecasts.

The reconciliation methods applied included bottom-up aggregation and MinTrace reconciliation (using ordinary least squares and other weight-based methods). These reconciliation methods were pivotal in reducing inconsistencies and maintaining a coherent structure across all levels.

4.3 Hyperparameter Tuning

The empirical study included extensive hyperparameter optimization using the Optuna library. Hyperparameter tuning is crucial to enhance the predictive capabilities of models by optimizing configurations such as learning rates, hidden layers, and window sizes. In this study, the models considered included advanced neural architectures like Transformer-based models (Vanilla Transformer, TFT, Informer, PatchTST, Autoformer), recurrent networks (N-BEATS), and other bespoke neural network architectures (such as MLP and NHITS). The hyperparameter tuning process aimed to identify the optimal configuration for each model, which was particularly necessary due to the inherent complexity of hierarchical time series forecasting.

Optuna's Tree-structured Parzen Estimator (TPE) was used as the search strategy to determine the best hyperparameters by exploring a wide range of values for each configuration. Parameters such as input window size, number of hidden units, number of layers, learning rate, batch size, and other training-specific hyperparameters were tuned through multiple iterations to achieve the best performance on the validation data. The models were trained over a fixed number of maximum steps, with checks for early stopping to prevent overfitting. Each model’s hyperparameters were tuned for both

individual point forecasts as well as probabilistic forecasts to ensure robustness across different forecast horizons and reconciliation scenarios. The hyperparameter optimization also included consideration of random seeds to maintain reproducibility of results.

The following Table 2 presents the hyperparameter search space utilized for Hyperparameter Optimization (HPO) of Transformer-based models. The hyperparameters considered include input size, hidden size, learning rate, batch size, and random seed. Each hyperparameter is associated with a defined range of values, which can either be discrete or continuous.

This table provides a concise overview of the hyperparameter search settings, highlighting the different ranges and types used to guide the optimization of Transformer models, thereby aiming to achieve an optimal model configuration with robust performance.

The input size hyperparameter is tested with discrete values, specifically 28, 28 × 2, and 28 × 3, representing different configurations to account for variations in the input representation. The hidden size hyperparameter also follows a discrete search space, with possible values of 32, 64, 128, and 256, which directly influence the capacity and complexity of the model. The learning rate is tuned within a continuous logarithmic range between 10-4 and 10-2, which allows for a systematic exploration of step sizes used during the optimization process, impacting the convergence behavior of the model. The batch size is selected from discrete values of 16, 32, 64, and 128, determining the number of samples processed simultaneously during training, which in turn affects the model's learning dynamics and computational efficiency. Lastly, the random seed parameter is assigned a discrete integer range between 1 and 20, ensuring reproducibility while allowing variability in training experiments.

Hyperparameter	**Range**	**Parameter Type**
Input size	{28, 28 × 2, 28 × 3}	Discrete
Hidden size	{32, 64, 128, 256}	Discrete
Learning rate	[10−4, 10−2]	Continuous (log)
Batch size	{16, 32, 64, 128}	Discrete
Random seed	[1, 20]	Discrete (int)

Table 2 - Model's hyperparameter search spaces used in HPO for Transformer-based models.

Table 3 outlines the hyperparameter search space specifically used for the Hyperparameter Optimization (HPO) of a Multi-Layer Perceptron (MLP) model. Similar to the previous Transformer-based example, this MLP model considers five key hyperparameters: input size, hidden size, learning rate, batch size, and random seed, each defined within discrete or continuous ranges to achieve optimal performance.

For the input size, discrete values are tested, reflecting different configurations of input layers that are compatible with the model architecture. The hidden size is similarly configured, testing specific values to control the model capacity and complexity in terms of the number of neurons within each hidden layer. The learning rate, as previously mentioned, is explored within a continuous logarithmic range to determine the best pace for model convergence, ensuring the training process balances between effective learning and avoiding divergence. The batch size is defined by discrete options, affecting the efficiency and stability of the model training process by determining how many data samples are processed at each step. The random seed is varied across discrete integer values, providing diversity in initialization to enhance the reproducibility of results.

The overall goal remains consistent with that of Transformer models, focusing on efficiently navigating through the hyperparameter space to identify the optimal configuration for improved performance of the MLP, while leveraging the same types of hyperparameters and optimization strategies.

Hyperparameter	**Range**	**Parameter Type**
Input size	{28, 28 × 2, 28 × 3}	Discrete
Hidden size	{16, 32, 64, 128, 256, 512, 1024}	Discrete
Number of layers	{1, 2, 3, 4, 5, 6, 7, 8}	Discrete
Learning rate	[10−4, 10−2]	Continuous (log)
Batch size	{16, 32, 64, 128}	Discrete
Random seed	[1, 20]	Discrete (int)

Table 3 - Model's hyperparameter search spaces used in HPO for MLP model.

The hyperparameter search space for the N-BEATS model is outlined in table 4, similar to the earlier Transformer and MLP models, covering input size, learning rate, batch size, and random seed.

Unique to the N-BEATS model, the hyperparameters also include the number of blocks and number of MLP units, each configured with discrete ranges to control the model architecture's depth and complexity. Input size and learning rate maintain the same types and ranges as previously explained, supporting flexibility in model tuning. The batch size and random seed are similarly varied, aiming to enhance both the training stability and reproducibility of results. The goal remains to identify the optimal combination of parameters for high model performance through systematic exploration.

Hyperparameter	**Range**	**Parameter Type**
Input size	{28, 28 × 2, 28 × 3}	Discrete
Number of blocks	{[1, 1, 1], [2, 2, 2], [4, 4, 4], [16, 16, 16]}	Discrete
Number of MLP units	{[[64, 64], [64, 64], [64, 64]], [[128, 128], [128, 128], [128, 128]], [[256, 256], [256, 256], [256, 256]], [[512, 512], [512, 512], [512, 512]], [[1024, 1024], [1024, 1024], [1024, 1024]]}	Discrete
Learning rate	[10−4, 10−2]	Continuous (log)
Batch size	{16, 32, 64, 128}	Discrete
Random seed	[1, 20]	Discrete (int)

Table 4 - Model's hyperparameter search spaces used in HPO for NBEATS model.

Table 5 describes the hyperparameter search space for the NHITS model, sharing several similarities with the N-BEATS model in terms of input size, number of blocks, number of MLP units, learning rate, batch size, and random seed, all of which use discrete or continuous ranges for systematic optimization.

Specific to NHITS are the number of pool kernel sizes and the number of frequency downsample, both of which are configured as discrete hyperparameters. These additional parameters provide flexibility for handling temporal features by varying pooling

operations and downsampling frequencies, crucial for optimizing performance in time-series forecasting.

Overall, the search strategy for NHITS extends the structure of N-BEATS, focusing on specific parameters that enhance the model's adaptability to sequential data, while maintaining common principles in hyperparameter tuning.

Hyperparameter	**Range**	**Parameter Type**
Input size	{28, 28 × 2, 28 × 3}	Discrete
Number of blocks	{[1, 1, 1], [2, 2, 2], [4, 4, 4], [16, 16, 16]}	Discrete
Number of MLP units	{[[64, 64], [64, 64], [64, 64]], [[128, 128], [128, 128], [128, 128]], [[256, 256], [256, 256], [256, 256]], [[512, 512], [512, 512], [512, 512]], [[1024, 1024], [1024, 1024], [1024, 1024]]}	Discrete
Number of pool kernelsize	{[1, 1, 1], [2, 2, 2], [4, 4, 4], [16, 8, 1]}	Discrete
Number of frequency downsample	{[168, 24, 1], [16, 8, 1], [24, 12, 1], [2, 1, 1], [1, 1, 1]}	Discrete
Learning rate	[10−4, 10−2]	Continuous (log)
Batch size	{16, 32, 64, 128}	Discrete
Random seed	[1, 20]	Discrete (int)

Table 5 - Model's hyperparameter search spaces used in HPO for NHITS model.

Table 6 outlines the hyperparameter optimization settings specifically employed for Transformer-based and MLP-based models, providing details on training control and validation procedures to ensure effective and efficient optimization of the models.

The maximum number of training steps is set to 5000, which determines the total number of iterations for which the model is allowed to train. This serves as a cap to avoid

excessive training and overfitting. The validation check steps are set at every 100 steps, which means that the model's performance on a validation set is evaluated periodically, helping to monitor progress and guide the optimization process towards better generalization.

The early stopping patience steps are set at 4, indicating that if there is no improvement in validation performance for four consecutive validation checks, the training will stop prematurely. This is a safeguard against overfitting and reduces unnecessary computational expenditure by halting training once the model ceases to improve. The number of trials, set to 30, refers to the total number of different configurations of hyperparameters tested during the optimization process, allowing exploration of the parameter space to determine an optimal configuration.

The validation function used is Mean Absolute Error (MAE), which serves as the evaluation metric to quantify model performance during validation. MAE measures the average magnitude of errors in predictions without considering their direction, making it a straightforward and interpretable metric, especially relevant when considering models like Transformers and MLPs that operate on potentially complex datasets.

These optimization settings ensure a balance between effective model training and computational efficiency, similar in principle to the hyperparameter tuning processes of N-BEATS and NHITS models. By structuring training with validation checks, early stopping, and multiple trials, these methods ensure robust exploration of hyperparameters while avoiding pitfalls like overfitting and excessive computation.

Parameter	**Value**
Maximum number of training steps	5000
Validation check steps	100
Early stopping patience steps	4
Number of trials	30
Validation function	MAE

Table 6 - Hyperparameter optimization settings for Transformer-based and MLP-based models.

The next table 7 presents parameter configurations for the Transformer-based models: Transformer, Temporal Fusion Transformer (TFT), Informer, PatchTST, and

Autoformer. These configurations aim to enhance time-series forecasting performance by optimizing each model's unique architecture.

For multi-head self-attention layers, Transformer, TFT, Informer, and Autoformer utilize four heads, whereas PatchTST employs sixteen to capture complex temporal dependencies. Most models use two or three encoder layers and a single decoder layer, providing effective processing of time-dependent features. The convolutional hidden size is consistently set to 32 for most models, supporting sufficient representational capacity, while Gaussian Error Linear Unit (GELU) is the preferred activation function for stability.

To prevent overfitting, dropout is used, with Transformer, Informer, and Autoformer adopting a rate of 0.05, while PatchTST uses 0.2 for more rigorous regularization. The decoder input size multiplier is consistently set at 0.5 for models that utilize it, and ProbSparse attention factor is set to 3 for TFT and Autoformer to improve attention efficiency.

PatchTST includes specific hyperparameters such as linear hidden size (256), patch length (16), and stride (8), reflecting its approach for temporal segmentation. The moving average window (25) aids in capturing long-term trends by smoothing out fluctuations.

All models use an inference window batch size of 1024, and scaling is noted as "Robust," as a preprocessing method to ensure stability. These configurations collectively optimize each model's capacity to manage temporal data effectively for time-series forecasting.

Table 8 presents the specific hyperparameter configurations used for the MLP, N-BEATS, and NHITS models, highlighting settings that facilitate their performance in time-series forecasting tasks. These configurations reflect the careful tuning of parameters to leverage each model's unique architecture and optimize its forecasting capabilities.

The window batch size is set uniformly at 1024 for all models, ensuring a consistent quantity of data is processed during each inference stage, thereby contributing to training stability. The scaling parameter for each model is configured as "Identity," implying that no additional scaling transformations are applied to the input data, preserving the original data scale for processing.

Parameter	Transformer	TFT	Informer	PatchTST	Autoformer
Multi-head self-attention layers	4	4	4	16	4
Encoder layers	2	--	2	3	2
Decoder layers	1	--	1	--	1
Convolutional hidden size	32	--	32	--	32
Activation function	GELU	--	GELU	GELU	GELU
Dropout	0.05	0.1	0.05	0.2	0.05
Attention layer dropout	--	0.0	--	0.0	--
Flatten head dropout	--	--	--	0.0	--
Linear layer dropout	--	--	--	0.2	--
Decoder input size multiplier	0.5	--	0.5	--	0.5
ProbSparse attention factor	--	--	3	--	3
Linear hidden size	--	--	--	256	--
Patch length	--	--	--	16	--
Stride	--	--	--	8	--
Moving average window	--	--	--	--	25
Inference windows batch size	1024				
Scaling	Robust				

Table 7 - Specific parameter configurations for each Transformer-based model.

The harmonic terms for seasonality and polynomial degree for trend are explicitly tuned in the N-BEATS model, both set to a value of 2. This indicates a systematic approach to capturing seasonal variations and trend components within the time-series data, reflecting N-BEATS' focus on interpretable component learning. Regarding the stack types, the N-BEATS model employs a combination of "identity," "trend," and "seasonality" stacks, whereas NHITS uses "identity" stacks exclusively. These stack configurations highlight

the differing strategies in model decomposition and emphasize the interpretability offered by N-BEATS.

The dropout probability theta is set to 0.0 for both N-BEATS and NHITS, indicating no dropout regularization is applied. This configuration choice is a reliance on maintaining full feature representation during training, which may be advantageous for models that focus on learning distinct components from the input data. The activation function chosen for both N-BEATS and NHITS is ReLU (Rectified Linear Unit), a common selection in deep learning due to its effectiveness in preventing vanishing gradients, thereby enabling stable and efficient training.

The learning rate decay is set to a factor of 3 for both N-BEATS and NHITS, allowing the learning rate to decrease progressively during training, which helps stabilize the convergence and adaptively reduce the step size as the model approaches optimality. The pooling mode is unique to NHITS, where MaxPool1d is employed to reduce the feature dimensionality while preserving the most critical information, enhancing the model's ability to generalize across temporal segments. Additionally, NHITS utilizes a linear interpolation mode to ensure smooth reconstruction of the forecasted time-series values, which contributes to maintaining temporal continuity in the predictions.

Overall, these parameter configurations have been designed to enhance each model's ability to capture and forecast temporal dynamics effectively, reflecting the distinct approaches that each architecture takes to model time-series data.

Table 9 presents the optimal hyperparameter configurations derived from Optuna for Transformer-based models, including Transformer, TFT, Informer, PatchTST, and Autoformer. The configurations include input size, hidden size, learning rate, batch size, and random seed, each carefully tuned to maximize model performance in time-series forecasting.

The input size varies among the models, with Transformer and PatchTST both employing an input size of 2×28, while TFT uses 3×28 and Informer and Autoformer use 28. The hidden size differs significantly between models: Transformer and TFT use a hidden size of 64, Informer and PatchTST have a larger hidden size of 256, and Autoformer uses 128. These differences in hidden size reflect the varying computational capacities and architectural complexities of each model.

Parameter	MLP	NBEATS	NHITS
Windows batch size	1024		
Scaling	Identity		
Harmonic terms for seasonality	--	2	--
Polynomial degree for trend	--	2	--
Stack types	--	["identity", "trend", "seasonality"]	["identity", "identity", "identity"]
Dropout prob theta	--	0.0	0.0
Activation function	--	ReLU	ReLU
Learning rate decays	--	3	3
Pooling mode	--	--	MaxPool1d
Interpolation mode	--	--	Linear

Table 8 - Specific parameter configurations for each MLP-based model.

The learning rate values also exhibit variation, indicating different optimization dynamics suited to each model's training behavior. Transformer uses a learning rate of 0.0051125, TFT 0.0022691, Informer 0.0003848, PatchTST 0.0002071, and Autoformer 0.0003815. These values reflect the specific learning dynamics needed to achieve convergence efficiently while avoiding instability during training.

The batch size is consistent across most models, set at 128 for Transformer, TFT, PatchTST, and Autoformer, while Informer uses a batch size of 64. The difference in batch size for Informer may be due to its higher hidden size, necessitating smaller batches to balance memory usage during training.

The random seed is another parameter that ensures reproducibility in model training. Different values are used for each model: Transformer (16), TFT (1), Informer (18), PatchTST (1), and Autoformer (1). These configurations allow for reproducibility of results while providing slight variability in initialization for model robustness.

Overall, the hyperparameter configurations presented in Table 9 reflect careful optimization to suit each model's architecture, balancing complexity, learning rate dynamics, and computational considerations to achieve optimal performance in time-series forecasting tasks.

Parameter	Transformer	TFT	Informer	PatchTST	Autoformer
Input size	2 x 28	3 x 28	28	2 x 28	28
Hidden size	64	64	256	256	128
Learning rate	0.0051125	0.0022691	0.0003848	0.0002071	0.0003815
Batch size	128	128	64	128	128
Random seed	16	1	18	1	1

Table 9 - Optimal hyperparameter configurations from Optuna for Transformers-based models.

Table 10 presents the optimal hyperparameter configurations from Optuna for MLP-based models, including MLP, N-BEATS, and NHITS. The configurations include input size, hidden size, number of layers, number of blocks, number of MLP units, number of pool kernel size, number of frequency downsample, learning rate, batch size, and random seed, each optimized to enhance model performance in time-series forecasting.

The input size for MLP and N-BEATS is set to 3 × 28, whereas NHITS uses an input size of 28, reflecting the different input structures required by each model's architecture. The hidden size is specified only for the MLP model, with a value of 512, while N-BEATS and NHITS do not utilize a hidden size in the same manner due to their block-based architectures. The number of layers is set to 2 for MLP, indicating a relatively shallow architecture compared to other deep learning models.

For the number of blocks, N-BEATS uses a configuration of [4, 4, 4], while NHITS employs [2, 2, 2], reflecting different strategies for model decomposition and hierarchical learning. The number of MLP units varies, with N-BEATS using [[1024, 1024], [1024, 1024], [1024, 1024]], while NHITS uses smaller units of [[128, 128], [128, 128], [128, 128]], indicating a difference in model capacity and focus on feature representation.

The number of pool kernel sizes and number of frequency downsample are unique to NHITS, with values of [2, 2, 2] for pooling kernel size and [168, 24, 1] for frequency downsampling. These configurations support the NHITS model in capturing long-term dependencies and effectively reducing dimensionality for temporal features.

The learning rate is set differently for each model, with MLP at 0.0001018, N-BEATS at 0.0001738, and NHITS at 0.0003492, reflecting the specific optimization needs of each model. The batch size is consistent at 128 across all models, ensuring comparable training dynamics. The random seed is set to different values for each model (MLP: 17, N-

BEATS: 9, NHITS: 10), ensuring reproducibility while providing slight variations in model initialization for robustness.

Overall, the hyperparameter configurations in Table 10 were tailored to the unique architectures and capabilities of each MLP-based model, balancing model complexity, learning rate, and capacity to optimize performance in time-series forecasting.

Parameter	**MLP**	**NBEATS**	**NHITS**
Input size	3 x 28	3 x 28	28
Hidden size	512	--	--
Number of layers	2	--	--
Number of blocks	--	[4, 4, 4]	[2, 2, 2]
Number of MLP units	--	[[1024, 1024], [1024, 1024], [1024, 1024]]	[[128, 128], [128, 128], [128, 128]]
Number of pool kernelsize	--	--	[2, 2, 2]
Number of frequency downsample	--	--	[168, 24, 1]
Learning rate	0.0001018	0.0001738	0.0003492
Batch size	128	128	128
Random seed	17	9	10

Table 10 - Optimal hyperparameter configurations from Optuna for MLP-based models.

Tables 2 to 10 present a comprehensive overview of the hyperparameter configurations employed across various model types, including Transformer-based, MLP-based, and specialized models like N-BEATS and NHITS, used in time-series forecasting tasks. Each table highlights how different hyperparameters were optimized to align with the distinct architectural features of each model, such as input size, hidden size, learning rate, batch size, and specific architectural components.

The configurations varied to accommodate the unique learning needs and computational considerations of each model. Transformer-based models, as shown in Tables 2 through 7, required specific adjustments in multi-head attention layers, encoder-decoder configurations, and other components to manage complex temporal dependencies. For the MLP-based and specialized models (Tables 8 to 10), configurations such as stack types, number of blocks, MLP units, and frequency handling played critical roles in

enhancing the models' interpretability and ability to capture seasonality and trend components.

Overall, the presented hyperparameter configurations reflect careful, model-specific tuning to achieve optimal performance in time-series forecasting. These settings were selected to balance model capacity, training efficiency, and the ability to generalize across different temporal contexts, ultimately providing robust and accurate forecasting outcomes.

4.4 Probabilistic Forecasting

In addition to point forecasts, the study also explored probabilistic forecasting to capture the uncertainty inherent in future sales predictions. Probabilistic forecasts are particularly relevant in a retail context, as they provide a distribution of possible future values, allowing decision-makers to assess risk and make more informed inventory and supply chain decisions. The probabilistic forecasting models were based on deep learning architectures and extended the capabilities of point forecasting models by incorporating stochastic elements in the predictions. The models used included variants like Vanilla Transformer, TFT, Informer, PatchTST, Autoformer, N-BEATS, N-HITS and MLP, all extended for probabilistic forecasting.

Each of these models utilized an advanced loss function, such as the quantile loss function, to generate multiple quantiles of future values rather than a single point estimate. The goal was to provide a richer representation of the future by giving probabilistic bounds that describe potential fluctuations in sales volumes. These probabilistic forecasts were further reconciled across different hierarchical levels using the same reconciliation techniques as in point forecasting. This added an additional layer of complexity, as it required ensuring not only consistency in the expected values across hierarchical levels but also coherence in the entire forecast distribution.

When the assumption of normally distributed residuals is deemed unrealistic, an alternative approach is to employ bootstrapping techniques. Bootstrapping requires only that the residuals are uncorrelated and exhibit constant variance. To illustrate this procedure, we will utilize a naïve forecasting methodology.

A one-step forecast error, denoted as $e_t = y_t - \hat{y}_{t|t-1}$. For a naïve forecasting method, $\hat{y}_{t|t-1} = y_{t-1}$, so it can be rewritten as

$$y_t = y_{t-1} + e_t.$$

Assuming that future forecast errors will resemble historical forecast errors, for $t > T$, the error term e_t can be replaced by resampling from previously observed errors (i.e., residuals). Thus, we can simulate the subsequent value of a time series using

$$y^{\cdot}_{T+1} = y_T + e^{\cdot}_{T+1},$$

where $e^{\cdot}_{T+1}$ is a randomly sampled residual from the historical error collection, and $y^{\cdot}_{T+1}$ denotes a potential future value that would occur given that specific sampled error. The notation $y^{\cdot}$ is used to indicate that this is not the observed value y_{t+1}, but rather one possible future scenario.

By incorporating this new simulated value into our data set, we can proceed iteratively to simulate the following observation:

$$y^{\cdot}_{T+2} = y_{T+1} + e^{\cdot}_{T+2}.$$

Where $e^{\cdot}_{T+2}$ is an independent sample drawn from the collection of residuals. This process can be continued iteratively to simulate an entire set of future time series values.

By repeating this procedure multiple times, a variety of potential future trajectories can be generated.

4.5 Performance Metrics

The evaluation of model performance was performed using a variety of metrics that reflected both the accuracy of point forecasts and the reliability of probabilistic forecasts. For point forecasts, Mean Absolute Scaled Error (MASE) was used as a primary metric. MASE is defined as:

$$MASE = \frac{\frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|}{\frac{1}{n-1}\sum_{i=2}^{n}\left|y_i - y_{i-1}\right|}.$$

Where y_i represents the actual values, and $\hat{y}_i$ represents the predicted values. The MASE provides a normalized version of the MAE, which is useful for comparing forecast

accuracy across different series by scaling the error relative to a naive benchmark (e.g., the in-sample one-step naive forecast).

For probabilistic forecasts, the Continuous Ranked Probability Score (CRPS) was used, which measures the accuracy of the entire forecast distribution. The CRPS is defined as:

$$CRPS(F,y) = \int_{-\infty}^{\infty} (F(z) - 1y \le z)^2 dz,$$

where $F(z)$ is the cumulative distribution function of the forecast, and y is the observed value. The CRPS metric assesses both the accuracy of the median forecast and the dispersion of the forecast distribution, making it particularly valuable in assessing probabilistic forecasting models.

In addition to MASE and CRPS, the models were also evaluated in terms of their reconciliation quality. Specifically, Mean Squared Scaled Error (MSSE) was used to assess the coherence between hierarchical levels, ensuring that aggregate forecasts were consistent with disaggregated series. MSSE is defined as:

$$MSSE = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\frac{1}{n-1}\sum_{i=2}^{n}\left(y_i - y_{i-1}\right)^2}$$

where y_i and $\hat{y}_i$ represent the actual and predicted values, respectively. MSSE provides a normalized version of the MSE, penalizing larger deviations more significantly and allowing for fairer comparisons across different time series.

The performance of reconciliation techniques was also measured by analyzing the difference between reconciled and unreconciled forecasts, where improved coherence was indicative of the success of methods like MinTrace. Lastly, the training time and computational efficiency of each model were recorded as secondary evaluation criteria, providing insights into the trade-off between forecast accuracy and computational cost, which is crucial for practical implementation in a retail environment.

4.6 Results and Discussion

4.6.1 Base Forecasts

The base forecasts provide an initial assessment of the model's capability to predict future sales across various hierarchical levels. The metrics in the provided results indicate the performance of benchmark models and help set the stage for comparison with more complex models. From the results, presented in Table 11, it can be observed that simple benchmark models, such as Naive and Seasonal Naive, provide a relatively high Mean Absolute Scaled Error (MASE) and Mean Squared Scaled Error (MSSE) values. This suggests that these models are less capable of accurately capturing the seasonality and trends within the hierarchical time series. The high errors are particularly evident in cases where sales variability is significant, underscoring the need for more sophisticated approaches that can effectively leverage the hierarchical relationships present in the dataset.

	metric	Naive	Seasonal Naive	ARIMA	ETS	MLP	NBEATS	NHITS	Vanilla Transformer	TFT	Informer	Patch TST	Autoformer
Total	**mase**	2,496	0,938	1,050	0,786	0,803	0,783	0,763	0,727	1,112	0,849	**0,722**	0,767
	msse	4,371	0,719	0,981	0,536	0,566	0,558	**0,532**	0,579	1,184	0,728	0,542	0,644
State	**mase**	1,943	1,004	0,917	0,822	0,822	0,805	0,788	**0,783**	1,057	0,930	0,819	0,872
	msse	2,951	0,962	0,816	0,617	0,636	0,617	**0,574**	0,636	1,054	0,839	0,653	0,744
Store	**mase**	1,706	0,992	0,853	0,751	0,759	0,743	**0,734**	0,761	0,946	0,822	0,747	0,824
	msse	2,206	0,910	0,660	0,547	0,564	0,560	**0,527**	0,570	0,830	0,658	0,559	0,664
Category	**mase**	2,209	1,043	0,940	0,848	0,805	0,800	0,768	0,794	1,068	0,784	**0,763**	0,801
	msse	3,871	1,042	0,836	0,601	0,590	0,613	**0,545**	0,638	1,057	0,667	0,582	0,660
Department	**mase**	1,949	1,177	0,962	0,966	0,957	0,931	0,927	0,938	1,056	**0,921**	0,937	0,950
	msse	3,221	1,344	0,867	0,854	0,859	0,812	0,828	**0,809**	1,030	0,815	0,813	0,879
State-Category	**mase**	1,801	1,109	0,910	0,860	0,858	0,870	**0,839**	0,878	1,049	0,912	0,871	0,888
	msse	2,975	1,279	0,836	0,712	0,733	0,755	**0,692**	0,772	1,029	0,857	0,737	0,782
State-Department	**mase**	1,710	1,198	0,999	0,940	0,937	0,929	**0,916**	0,935	1,013	0,957	0,930	0,943
	msse	2,729	1,427	0,961	0,851	0,878	0,856	**0,849**	0,856	0,996	0,901	0,854	0,892
Store-Category	**mase**	1,561	1,120	0,921	**0,871**	0,891	0,893	0,873	0,902	0,967	0,925	0,891	0,916
	msse	2,242	1,283	0,801	**0,716**	0,740	0,744	0,718	0,771	0,870	0,813	0,743	0,787
Store-Department	**mase**	1,662	1,283	0,999	0,968	0,951	0,955	**0,942**	0,964	0,986	0,972	0,976	0,985
	msse	2,383	1,524	0,916	**0,854**	0,874	0,880	0,860	0,898	0,910	0,893	0,900	0,906
Product	**mase**	2,518	1,843	1,505	1,488	1,436	1,438	**1,428**	1,447	1,469	1,458	1,453	1,472
	msse	3,926	2,204	1,436	1,405	1,334	1,336	**1,310**	1,359	1,407	1,370	1,363	1,415
State-Product	**mase**	2,205	1,844	1,501	1,490	**1,408**	1,425	1,424	1,424	**1,408**	1,431	1,451	1,450
	msse	2,973	2,066	1,266	1,229	1,246	1,249	**1,226**	1,243	1,248	1,272	1,248	1,272
Bottom	**mase**	2,443	2,010	1,713	1,727	**1,435**	1,465	1,459	1,442	**1,435**	1,459	1,569	1,524
	msse	2,802	2,138	1,198	**1,183**	1,323	1,313	1,297	1,342	1,339	1,351	1,259	1,313
All	**mase**	2,313	1,871	1,571	1,571	**1,368**	1,390	1,383	1,376	1,379	1,391	1,462	1,437
	msse	2,871	2,037	1,187	**1,158**	1,248	1,242	1,223	1,263	1,277	1,278	1,210	1,256

Table 11 - Base forecasts.

These base forecasts establish a benchmark to evaluate the performance of more advanced models like Transformer architectures and MLPs. The inability of these models to produce low error values highlights the complexity of retail time series, which typically

feature mixed frequencies, structural breaks, and hierarchical aggregation requirements. It is critical to compare these base results with more advanced methods to understand the effectiveness of deep learning techniques in improving forecast quality.

The Base Forecasts table provides a detailed evaluation of multiple benchmark models, including Naive, Seasonal Naive, ARIMA, ETS, as well as advanced models like MLP, NBEATS, NHITS, Vanilla Transformer, TFT, Informer, PatchTST, and Autoformer. The metrics analyzed include Mean Absolute Scaled Error (MASE) and Mean Squared Scaled Error (MSSE), which provide insights into the performance of each model across different hierarchical levels of the dataset.

The Naive model recorded a MASE of 2.496 and an MSSE of 4.371 at the Total level, indicating high error values and limited ability to capture patterns in the data. The Seasonal Naive model showed improved performance over the Naive model with a MASE of 0.938 and an MSSE of 0.719, indicating that incorporating seasonality reduces the overall error but still leaves significant room for improvement when compared to more sophisticated approaches. The ARIMA model achieved a MASE of 1.050 and an MSSE of 0.981 at the Total level, while the ETS model further reduced these errors, achieving a MASE of 0.786 and an MSSE of 0.536. These results suggest that statistical models like ARIMA and ETS are more capable of capturing the underlying structure of the time series compared to the naive models. However, they still fall short when handling complex hierarchical dependencies, as seen from higher errors at finer hierarchical levels.

Among the advanced models, MLP, NBEATS, and NHITS demonstrated further improvements. Notably, the NHITS model, with a MASE of 0.763 and an MSSE of 0.532 at the Total level, demonstrated effective handling of temporal patterns, outperforming the traditional statistical models. The NBEATS model also performed well, achieving similar metrics to NHITS, suggesting that these advanced methods can model non-linear relationships better than conventional approaches. The Transformer-based models—Vanilla Transformer, TFT, Informer, PatchTST, and Autoformer—demonstrated the most substantial improvements in forecasting accuracy. The PatchTST model recorded the lowest MASE of 0.722 at the Total level, which indicates its superiority in capturing both long-term and short-term dependencies present in the retail time series. The Autoformer model also performed well, with a MASE of 0.767 and an MSSE of 0.644, indicating robust performance across the dataset. These models leveraged attention

mechanisms that allow for the effective modeling of complex hierarchical structures, significantly reducing forecasting errors.

A detailed evaluation across different levels of the hierarchy, such as State, Store, Category, and Product, reveals that the advanced models generally outperform the traditional benchmarks. For example, at the Store level, ETS achieved a relatively low MASE of 0.751 and an MSSE of 0.547, but NHITS further reduced the MASE to 0.734 and MSSE to 0.527. PatchTST again showed excellent performance, with the lowest MASE of 0.763 and an MSSE of 0.582 at the Category level, highlighting its ability to consistently outperform both traditional and other deep learning models at varying granularities of the hierarchical structure. The Vanilla Transformer also demonstrated a balanced performance, with metrics comparable to NHITS, suggesting its reliability for practical implementation where a balance between accuracy and computational efficiency is required.

Overall, the Base Forecasts analysis underscores the significant advantage offered by advanced deep learning models, particularly those based on Transformer architectures. The PatchTST and Autoformer models consistently provided lower errors across different levels, demonstrating their superiority in handling hierarchical relationships and non-linear dependencies in the data. The Transformer-based models significantly outperformed traditional approaches like ARIMA and ETS, which, while more effective than naive models, failed to match the precision and adaptability required for complex retail forecasting scenarios. Furthermore, models like NHITS and NBEATS offered substantial improvements over traditional methods, highlighting the benefits of using deep learning techniques that can inherently handle non-linearity and capture complex temporal patterns.

4.6.2 Reconciled Point Forecasts

4.6.2.1 Benchmarks

The benchmark models for point forecasting exhibit limited capabilities in minimizing forecasting errors across all evaluated metrics (Table 12). The MASE values indicate a consistent level of error that benchmarks like ARIMA and ETS could not sufficiently reduce. This is particularly true for product-level forecasts, where the variance is greater, leading to an amplified error under these simple models.

The reconciliation quality, as evaluated by MSSE, remains insufficient in most cases, indicating that the forecasts do not maintain proper coherence across hierarchical levels. The benchmark models failed to effectively minimize the discrepancy between higher-level aggregated series and lower-level disaggregated series. This discrepancy could result in practical issues in inventory management, where inconsistency at different levels of aggregation impacts overall efficiency.

		BottomUp				MinTrace-OLS		MinTrace-WLS_struct		MinTrace-WLS_var		MinTrace-mint_shrin	
	metric	Naïve	Seasonal Naive	ARIMA	ETS	ARIMA	ETS	ARIMA	ETS	ARIMA	ETS	ARIMA	ETS
Total	**mase**	2,496	0,938	1,202	0,727	0,922	0,742	0,900	0,704	0,924	**0,699**	0,856	0,701
	msse	4,371	0,719	1,381	0,532	0,789	0,494	0,816	**0,472**	0,863	0,478	0,673	0,476
State	**mase**	1,943	1,004	1,109	0,796	0,899	0,833	0,900	0,797	0,919	0,794	0,842	**0,791**
	msse	2,951	0,962	1,154	0,630	0,789	0,619	0,790	**0,604**	0,820	0,608	0,711	0,605
Store	**mase**	1,706	0,992	0,941	0,752	0,854	0,776	0,826	0,752	0,838	0,752	0,760	**0,748**
	msse	2,206	0,910	0,832	0,558	0,684	0,547	0,641	0,537	0,658	0,542	0,575	**0,534**
Category	**mase**	2,209	1,043	1,111	**0,764**	0,863	0,853	0,887	0,797	0,919	0,787	0,826	0,802
	msse	3,871	1,042	1,156	0,580	0,734	0,604	0,764	0,566	0,813	**0,559**	0,670	0,570
Department	**mase**	1,949	1,177	1,036	**0,939**	1,140	1,026	0,978	0,967	0,988	0,949	0,967	0,981
	msse	3,221	1,344	1,026	**0,814**	1,284	0,941	0,872	0,869	0,894	0,833	0,845	0,884
State-Category	**mase**	1,801	1,109	1,044	**0,845**	0,906	0,882	0,909	0,858	0,938	0,852	0,876	0,857
	msse	2,975	1,279	1,028	0,728	0,825	0,729	0,816	0,715	0,846	**0,714**	0,777	0,725
State-Department	**mase**	1,710	1,198	0,988	0,936	1,041	0,968	0,950	0,947	0,952	0,939	**0,935**	0,954
	msse	2,729	1,427	0,952	**0,840**	1,083	0,892	0,874	0,865	0,882	0,850	0,861	0,877
Store-Category	**mase**	1,561	1,120	0,959	0,869	0,929	0,879	0,897	**0,866**	0,904	0,867	0,883	0,868
	msse	2,242	1,283	0,878	0,719	0,799	0,719	0,762	**0,705**	0,778	0,709	0,733	0,710
Store-Department	**mase**	1,662	1,283	1,001	0,977	1,085	0,997	0,981	0,979	0,976	0,976	**0,961**	0,983
	msse	2,383	1,524	0,936	0,887	1,098	0,889	0,897	0,881	0,889	0,877	**0,874**	0,887
Product	**mase**	2,518	1,843	1,503	**1,453**	1,623	1,643	1,492	1,530	1,485	1,465	1,471	1,476
	msse	3,926	2,204	1,442	**1,330**	1,638	1,658	1,408	1,460	1,398	1,351	1,370	1,372
State-Product	**mase**	2,205	1,844	1,475	**1,466**	1,556	1,579	1,489	1,516	1,477	1,477	1,470	1,484
	msse	2,973	2,066	1,227	**1,187**	1,348	1,310	1,225	1,236	1,220	1,202	1,209	1,211
Bottom	**mase**	2,443	2,010	**1,713**	1,727	1,805	1,850	1,751	1,786	1,725	1,734	1,723	1,743
	msse	2,802	2,138	1,198	**1,183**	1,293	1,243	1,206	1,207	1,194	1,186	1,192	1,190
All	**mase**	2,313	1,871	1,569	**1,564**	1,652	1,677	1,589	1,615	1,570	1,571	1,564	1,579
	msse	2,871	2,037	1,188	**1,147**	1,289	1,231	1,179	1,179	1,170	1,152	1,160	1,160

Table 12 - Reconciled point forecasts – benchmarks.

ARIMA and ETS models were further explored, specifically focusing on their integration with different reconciliation techniques. The ARIMA model recorded a MASE of 1.569 and an MSSE of 1.564 at the All level, indicating that while the model captures some temporal dependencies, it struggles to manage the complexity of the data when all levels of hierarchy are considered. The ETS model performed comparably with a MASE of 1.564 and an MSSE of 1.147, showing that ETS, particularly with appropriate reconciliation methods, provides a robust baseline for forecasting hierarchical time series.

MinTrace-OLS was shown to be effective in reducing overall forecast error, with the ETS model achieving a MASE of 1.564 and an MSSE of 1.147, demonstrating significant improvements over standalone models. Similarly, MinTrace-WLS_struct applied to ETS

produced further gains, resulting in the lowest MSSE of 1.231 and a MASE of 1.677 across all levels, suggesting that this reconciliation technique is particularly effective at enhancing the accuracy of hierarchical forecasts.

The MinTrace-WLS_var and MinTrace-mint_shrink reconciliation techniques also demonstrated strong performance. MinTrace-WLS_var with ETS achieved an MSSE of 1.152, which was the lowest recorded at the All level, indicating its strength in maintaining coherence while reducing overall forecast error. MinTrace-mint_shrink with ETS achieved a comparable MSSE of 1.160, further highlighting the effectiveness of these advanced reconciliation approaches in managing the trade-off between forecast accuracy and hierarchical consistency.

Overall, the results from the Point Forecasts - Benchmarks section emphasize the value of combining traditional forecasting models with advanced reconciliation techniques to improve both accuracy and coherence in hierarchical time series forecasting. The findings underscore the importance of model selection and the application of appropriate reconciliation methods to achieve high-quality forecasts that meet the demands of complex retail settings, where accurate predictions are required across different levels of the product and sales hierarchy.

4.6.2.2 Transformer-based Models

The performance of Transformer models, such as Autoformer, Informer, and Vanilla Transformer, shows a marked improvement compared to the benchmark models (Table 13). Notably, the use of attention mechanisms allows these models to capture temporal dependencies more effectively, as evidenced by the decrease in MASE and MSSE values across various levels of aggregation. For instance, the Autoformer appears to be particularly adept at reducing both MASE and MSSE, indicating its effectiveness in capturing not only short-term seasonal trends but also the longer-term variations.

The results further highlight the ability of Transformers to maintain coherence across hierarchical levels, which is evidenced by a substantial reduction in MSSE values. This improvement is critical for applications such as retail, where maintaining consistency between high-level and low-level forecasts ensures operational alignment. The performance improvement also underscores the flexibility of Transformer models in handling large, complex datasets with intricate hierarchical relationships, where benchmark models often struggle.

		BottomUp					MinTrace-OLS					MinTrace-WLS_struct					MinTrace-WLS_var					MinTrace-mint_shrink				
	metric	**Vanilla Transformer**	**TFT**	**Informer**	**Patch TST**	**Autoformer**	**Vanilla Transformer**	**TFT**	**Informer**	**Patch TST**	**Autoformer**	**Vanilla Transformer**	**TFT**	**Informer**	**Patch TST**	**Autoformer**	**Vanilla Transformer**	**TFT**	**Informer**	**Patch TST**	**Autoformer**	**Vanilla Transformer**	**TFT**	**Informer**	**Patch TST**	**Autoformer**
Total	**mase**	2,820	2,935	2,940	2,247	2,925	0,755	1,118	0,871	**0,717**	0,807	0,978	1,235	1,086	0,858	1,067	1,101	1,315	1,211	0,966	1,202	2,258	4,216	2,554	1,598	2,480
	msse	5,484	6,118	5,860	3,573	5,671	0,612	1,199	0,770	**0,549**	0,695	0,935	1,476	1,140	0,750	1,084	1,129	1,693	1,347	0,896	1,316	3,460	11,451	4,320	1,999	4,080
State	**mase**	2,242	2,330	2,330	1,803	2,327	**0,778**	1,047	0,873	0,813	0,845	0,927	1,123	1,008	0,868	1,000	1,010	1,193	1,084	0,912	1,086	1,839	3,361	2,068	1,335	2,004
	msse	3,726	4,093	3,939	2,500	3,879	**0,630**	1,025	0,737	0,644	0,721	0,838	1,196	0,974	0,757	0,974	0,965	1,334	1,109	0,849	1,127	2,519	7,583	3,110	1,582	2,940
Store	**mase**	1,743	1,808	1,784	1,382	1,784	**0,738**	0,934	0,796	0,746	0,791	0,819	0,979	0,875	0,779	0,880	0,882	1,031	0,938	0,826	0,949	1,380	2,439	1,500	1,073	1,497
	msse	2,346	2,591	2,460	1,586	2,436	**0,542**	0,800	0,625	0,552	0,629	0,662	0,894	0,750	0,615	0,759	0,751	0,991	0,842	0,679	0,861	1,579	4,215	1,869	1,061	1,842
Category	**mase**	2,295	2,412	2,363	1,809	2,370	0,790	1,076	0,839	**0,760**	0,802	0,906	1,143	0,971	0,793	0,928	0,974	1,201	1,041	0,850	1,015	1,667	3,054	1,834	1,237	1,788
	msse	3,842	4,352	4,013	2,531	3,936	0,632	1,074	0,729	**0,577**	0,656	0,842	1,246	0,964	0,701	0,897	0,973	1,392	1,107	0,792	1,058	2,418	7,040	2,841	1,487	2,647
Department	**mase**	2,048	2,108	2,110	1,530	2,049	0,957	1,058	**0,935**	0,965	0,952	0,964	1,053	0,995	0,964	0,966	0,977	1,083	1,029	0,962	1,010	1,306	2,153	1,376	1,120	1,375
	msse	3,164	3,424	3,366	2,008	3,131	0,850	1,042	**0,836**	0,870	0,871	0,880	1,084	0,941	0,865	0,909	0,941	1,160	1,031	0,878	1,005	1,615	4,145	1,855	1,178	1,751
State-Category	**mase**	1,772	1,852	1,809	1,435	1,828	**0,852**	1,035	0,881	0,868	0,873	0,912	1,065	0,943	0,893	0,936	0,958	1,099	0,997	0,930	0,994	1,374	2,305	1,482	1,114	1,439
	msse	2,516	2,794	2,606	1,777	2,616	0,729	0,985	0,791	**0,726**	0,736	0,844	1,076	0,919	0,788	0,880	0,927	1,167	1,012	0,849	0,988	1,672	4,100	1,930	1,190	1,800
State-Department	**mase**	1,556	1,593	1,584	1,270	1,544	0,937	0,999	**0,933**	0,950	0,939	0,937	0,996	0,950	0,945	0,945	0,957	1,011	0,976	0,956	0,975	1,176	1,689	1,226	1,064	1,199
	msse	2,065	2,182	2,125	1,481	2,028	**0,855**	0,964	0,858	0,897	0,909	0,871	0,973	0,903	0,888	0,896	0,920	1,020	0,963	0,908	0,956	1,298	2,550	1,439	1,084	1,373
Store-Category	**mase**	1,391	1,448	1,404	1,170	1,408	0,877	0,948	0,902	**0,875**	0,887	0,900	0,971	0,928	0,882	0,918	0,935	1,006	0,965	0,908	0,957	1,141	1,677	1,193	1,008	1,177
	msse	1,623	1,799	1,655	1,201	1,662	0,722	0,840	0,771	**0,713**	0,743	0,768	0,883	0,812	0,730	0,794	0,822	0,942	0,868	0,767	0,855	1,143	2,266	1,265	0,931	1,230
Store-Department	**mase**	1,274	1,289	1,274	1,122	1,259	0,976	1,019	0,988	0,988	0,997	**0,965**	1,003	0,981	0,977	0,983	0,985	1,011	0,997	0,979	1,000	1,088	1,364	1,128	1,031	1,110
	msse	1,441	1,495	1,445	1,156	1,407	**0,877**	0,917	0,895	0,899	0,910	0,878	0,917	0,894	0,890	0,895	0,927	0,962	0,937	0,906	0,937	1,058	1,593	1,120	0,973	1,101
Product	**mase**	1,956	1,959	1,942	1,636	1,884	1,515	1,529	1,494	1,584	1,501	**1,465**	1,488	1,473	1,529	1,474	1,479	1,495	1,502	1,468	1,512	1,668	2,004	1,705	1,572	1,694
	msse	2,354	2,377	2,329	1,721	2,209	1,434	1,448	1,381	1,548	1,434	**1,365**	1,394	1,368	1,467	1,394	1,457	1,496	1,494	1,419	1,515	1,726	2,565	1,823	1,544	1,808
State-Product	**mase**	1,559	1,552	1,565	1,481	1,554	1,511	1,516	1,500	1,548	1,520	1,468	1,473	1,471	1,509	1,483	1,434	**1,429**	1,442	1,444	1,445	1,572	1,725	1,590	1,519	1,576
	msse	1,623	1,629	1,617	1,367	1,553	1,222	1,240	1,229	1,281	1,254	**1,194**	1,209	1,210	1,246	1,220	1,256	1,267	1,276	1,243	1,279	1,393	1,724	1,448	1,310	1,410
Bottom	**mase**	1,442	**1,435**	1,459	1,569	1,524	1,822	1,827	1,807	1,835	1,808	1,774	1,783	1,770	1,788	1,764	1,576	1,573	1,581	1,625	1,586	1,765	1,819	1,769	1,723	1,758
	msse	1,342	1,339	1,351	1,259	1,313	1,207	1,205	1,231	1,246	1,221	1,186	**1,185**	1,209	1,222	1,196	1,223	1,221	1,241	1,217	1,224	1,279	1,397	1,316	1,254	1,278
All	**mase**	1,501	1,500	1,514	1,518	1,548	1,636	1,651	1,625	1,656	1,630	1,596	1,613	1,598	1,616	1,596	**1,470**	1,476	1,479	1,500	1,484	1,654	1,801	1,669	1,595	1,656
	msse	1,523	1,540	1,532	1,323	1,483	1,175	1,193	1,193	1,220	1,195	**1,157**	1,175	1,179	1,194	1,174	1,205	1,222	1,228	1,192	1,218	1,328	1,673	1,384	1,257	1,347

Table 13 - Reconciled point forecasts – Transformers.

The Point Forecasts for Transformer-based models include Vanilla Transformer, TFT, Informer, Patch TST, and Autoformer, each evaluated with different reconciliation techniques such as BottomUp, MinTrace-OLS, MinTrace-WLS_struct, MinTrace-WLS_var, and MinTrace-mint_shrink. The metrics analyzed were Mean Absolute Scaled Error (MASE) and Mean Squared Scaled Error (MSSE), allowing for an in-depth assessment of performance across multiple hierarchical levels.

The BottomUp reconciliation approach applied to the transformer-based models provided varied results. Patch TST, under the BottomUp strategy, recorded the lowest MASE of 1.382 and an MSSE of 1.586 at the Store level, highlighting its effectiveness in disaggregated contexts where the demand for accuracy is crucial. However, other models like TFT and Autoformer exhibited higher error rates, with a MASE of 2.935 and 2.925 respectively at the Total level, which suggests that the BottomUp approach may be less suited for maintaining accuracy across more aggregated levels.

The MinTrace-OLS reconciliation applied to transformer models provided significant improvements in forecast accuracy. Notably, Patch TST recorded the lowest MASE of 0.717 and an MSSE of 0.549 at the Total level, indicating its superiority in managing aggregated series effectively. Autoformer, under the MinTrace-OLS framework, achieved a MASE of 0.807 and an MSSE of 0.695, making it another strong contender. These results underline the effectiveness of OLS-based reconciliation in enhancing the performance of transformer models for hierarchical forecasting tasks.

MinTrace-WLS_struct showed promising outcomes, particularly for the Patch TST model, which achieved a MASE of 0.858 and an MSSE of 0.750 at the Total level. This suggests that the WLS-based structural reconciliation can significantly improve consistency while maintaining low forecast errors. Autoformer also performed well under this reconciliation framework, with a MASE of 1.067 and an MSSE of 1.084 at the Total level, indicating its robustness in handling both aggregate and disaggregate series.

The MinTrace-WLS_var reconciliation strategy was applied to the transformer-based models and demonstrated considerable success in reducing forecast errors. Patch TST recorded a MASE of 0.966 and an MSSE of 0.896 at the Total level, while Vanilla Transformer achieved a MASE of 1.101 and an MSSE of 1.129. These results highlight that variance-based reconciliation methods, particularly when applied to transformer

models, can provide effective solutions for managing hierarchical dependencies in time series data.

The MinTrace-mint_shrink reconciliation applied to transformers also indicated strong performance, particularly for Patch TST and Autoformer. Patch TST achieved a MASE of 1.598 and an MSSE of 1.999 at the Total level, while Autoformer performed even better at the Store level, with a MASE of 1.073 and an MSSE of 1.497. This demonstrates the ability of MinTrace-mint_shrink to maintain consistency across hierarchical levels while effectively reducing overall forecasting errors.

In summary, the analysis of Point Forecasts using Transformer-based models reveals that the choice of reconciliation technique significantly influences the model's effectiveness. Patch TST consistently demonstrated strong performance across different reconciliation frameworks, particularly under MinTrace-OLS and MinTrace-WLS_struct, achieving some of the lowest MASE and MSSE values. This indicates that Patch TST, when coupled with advanced reconciliation techniques, can provide highly accurate and reliable forecasts for hierarchical time series. Autoformer also showed consistent robustness across various levels, especially when applied with MinTrace-WLS_var and MinTrace-mint_shrink, which makes it another strong candidate for complex hierarchical forecasting tasks. The results emphasize the importance of selecting both the appropriate transformer model and reconciliation strategy to optimize forecasting performance in hierarchical retail contexts.

4.6.2.3 MLP-based Models

Multi-Layer Perceptron (MLP) models provide competitive point forecasts, although the performance is slightly behind that of Transformer-based models (Table 14). The MASE values obtained with the MLPs indicate that while these models are capable of reducing forecasting errors when compared to benchmarks, they still fall short of fully capturing the hierarchical dependencies inherent in the dataset. The performance of MLPs is dependent on the hyperparameter tuning, and although they can effectively model non-linear relationships, they do not inherently have the sequential capabilities offered by Transformer architectures.

However, MLP models were more computationally efficient compared to Transformer models. This makes them suitable for scenarios where faster results are prioritized over slightly improved accuracy. The MSSE values suggest that MLPs still face challenges in

maintaining hierarchical coherence, although they perform better than simpler benchmarks. These results point to MLP's utility as an alternative for computationally constrained environments but highlight the necessity of hierarchical reconciliation methods for maintaining aggregation consistency.

		BottomUp			MinTrace-OLS			MinTrace-WLS_struct			MinTrace-WLS_var			MinTrace-mint_shrink		
	metric	MLP	NBEATS	NHITS	MLP	NBEATS	NHITS	MLP	NBEATS	NHITS	MLP	NBEATS	NHITS	MLP	NBEATS	NHITS
Total	mase	2,534	2,463	2,036	0,697	0,682	0,685	0,853	0,686	**0,652**	0,965	0,737	0,682	1,918	1,794	2,072
	msse	4,365	4,090	3,129	0,530	0,433	**0,414**	0,746	0,431	0,440	0,886	0,504	0,501	2,577	2,237	2,910
State	mase	2,027	1,963	1,652	0,757	0,770	0,787	0,840	**0,719**	0,749	0,900	0,732	0,776	1,555	1,445	1,675
	msse	3,006	2,793	2,205	0,580	**0,502**	0,529	0,724	0,509	0,532	0,819	0,559	0,571	1,904	1,646	2,123
Store	mase	1,553	1,516	1,334	0,732	0,723	0,734	0,784	**0,699**	0,724	0,834	0,720	0,748	1,202	1,109	1,269
	msse	1,902	1,817	1,519	0,533	0,495	0,512	0,609	**0,494**	0,515	0,673	0,528	0,548	1,232	1,098	1,343
Category	mase	2,046	2,010	1,668	0,734	0,783	0,815	0,791	**0,717**	0,758	0,858	0,722	0,759	1,444	1,381	1,544
	msse	3,051	2,922	2,320	0,557	0,551	0,566	0,681	**0,519**	0,552	0,775	0,556	0,586	1,784	1,647	2,017
Department	mase	1,742	1,717	1,504	0,935	0,979	0,955	0,922	0,881	0,885	0,926	**0,851**	0,853	1,211	1,164	1,241
	msse	2,415	2,326	1,944	0,804	0,915	0,870	0,792	0,751	0,740	0,814	0,698	**0,697**	1,289	1,225	1,433
State-Category	mase	1,605	1,540	1,383	0,822	0,847	0,898	0,864	**0,809**	0,865	0,906	0,816	0,878	1,227	1,143	1,293
	msse	2,084	1,954	1,705	0,664	0,659	0,758	0,740	**0,644**	0,725	0,805	0,675	0,752	1,316	1,189	1,475
State-Department	mase	1,364	1,339	1,244	0,927	0,944	0,975	0,915	0,894	0,930	0,925	**0,891**	0,914	1,091	1,065	1,138
	msse	1,653	1,617	1,451	0,850	0,902	0,935	0,840	0,814	0,838	0,860	**0,802**	0,824	1,126	1,080	1,217
Store-Category	mase	1,271	1,252	1,179	0,858	0,874	0,901	0,876	**0,858**	0,885	0,902	0,871	0,896	1,048	1,011	1,099
	msse	1,394	1,358	1,248	0,686	0,704	0,748	0,718	**0,685**	0,725	0,759	0,708	0,748	0,978	0,929	1,063
Store-Department	mase	1,156	1,161	1,128	0,962	0,990	0,996	0,952	0,953	0,973	0,956	**0,949**	0,970	1,034	1,028	1,078
	msse	1,222	1,225	1,171	0,855	0,889	0,905	0,850	**0,846**	0,867	0,873	0,858	0,876	0,967	0,959	1,028
Product	mase	1,691	1,717	1,680	1,511	1,562	1,522	1,477	1,498	1,477	1,449	**1,423**	1,436	1,572	1,558	1,637
	msse	1,834	1,868	1,820	1,427	1,511	1,445	1,377	1,396	1,368	1,396	**1,340**	1,350	1,562	1,520	1,661
State-Product	mase	1,480	1,489	1,479	1,508	1,543	1,527	1,484	1,502	1,492	1,427	**1,422**	1,433	1,513	1,515	1,557
	msse	1,389	1,411	1,396	1,214	1,256	1,258	**1,195**	1,206	1,211	1,219	1,202	1,214	1,304	1,293	1,358
Bottom	mase	1,537	**1,516**	1,534	1,791	1,845	1,827	1,765	1,801	1,790	1,595	1,599	1,619	1,699	1,718	1,749
	msse	1,237	1,250	1,256	1,178	1,210	1,236	**1,167**	1,180	1,201	1,181	1,182	1,197	1,218	1,218	1,258
All	mase	1,511	1,499	1,496	1,614	1,660	1,645	1,591	1,616	1,609	1,474	**1,470**	1,488	1,584	1,592	1,634
	msse	1,341	1,351	1,332	1,152	1,188	1,204	**1,141**	1,148	1,163	1,160	1,146	1,161	1,240	1,230	1,293

Table 14 - Reconciled point forecasts – MLP.

The Point Forecasts analysis for Multi-Layer Perceptron (MLP) models, in combination with reconciliation strategies such as BottomUp, MinTrace-OLS, MinTrace-WLS_struct, MinTrace-WLS_var, and MinTrace-mint_shrink, presents a comprehensive evaluation of how these methods influence forecasting accuracy at different hierarchical levels. The metrics considered were Mean Absolute Scaled Error (MASE) and Mean Squared Scaled Error (MSSE).

The BottomUp approach applied to MLP, NBEATS, and NHITS models revealed notable discrepancies in performance. Specifically, NHITS outperformed the other models with a MASE of 2.036 and an MSSE of 3.129 at the Total level, indicating its relative effectiveness in forecasting aggregated data. The MLP and NBEATS models, while slightly higher in error, still performed comparably, with MASE values of 2.534 and 2.463, respectively, indicating that BottomUp is generally less effective at reducing forecasting errors in aggregated series compared to other reconciliation methods.

The MinTrace-OLS reconciliation strategy significantly enhanced the performance of MLP-based models. The MLP model under MinTrace-OLS achieved a MASE of 0.697 and an MSSE of 0.530 at the Total level, marking a substantial reduction in forecast errors compared to the BottomUp approach. The lowest MSSE recorded for MinTrace-OLS was 0.414 for NHITS, demonstrating the effectiveness of this approach in managing aggregated forecasts across hierarchical levels. This result highlights the benefit of leveraging OLS reconciliation, particularly for MLP and NHITS models, to achieve higher accuracy.

MinTrace-WLS_struct further improved forecast performance, particularly for NHITS, which recorded a MASE of 0.652 and an MSSE of 0.440 at the Total level. The MLP model also benefited under this strategy, achieving a MASE of 0.853 and an MSSE of 0.746, suggesting that WLS_struct provides a balanced approach for reconciling both aggregated and disaggregated series. NBEATS also showed competitive performance, though slightly higher in error compared to NHITS, indicating its robustness in diverse forecasting scenarios.

The MinTrace-WLS_var strategy yielded strong results for MLP models, particularly at more granular levels. For instance, MLP achieved a MASE of 0.965 and an MSSE of 0.886 at the Total level, which, while slightly higher than the structural reconciliation approach, still showed considerable accuracy improvements over BottomUp. NHITS outperformed both MLP and NBEATS, achieving an MSSE of 0.501, suggesting that variance-based reconciliation is particularly well-suited to NHITS for maintaining coherence across hierarchical levels.

The MinTrace-mint_shrink reconciliation approach demonstrated mixed results across models and levels. For the MLP model, a MASE of 1.918 and an MSSE of 2.577 at the Total level were recorded, indicating higher errors compared to other reconciliation strategies. However, NHITS demonstrated stronger performance, particularly at the Store level, where it achieved a MASE of 1.269 and an MSSE of 1.343, reflecting its robustness in disaggregated contexts. This result underscores that while MinTrace-mint_shrink may be effective for certain granular levels, its overall performance can vary significantly depending on the model and data hierarchy.

In summary, the Point Forecasts analysis for MLP models reveals that reconciliation strategies like MinTrace-OLS and MinTrace-WLS_struct consistently enhance forecast

accuracy across hierarchical levels. NHITS generally outperformed MLP and NBEATS, particularly under MinTrace-WLS_struct and MinTrace-WLS_var, achieving the lowest MASE and MSSE values at both aggregated and disaggregated levels. MLP also showed considerable improvements with the use of these reconciliation methods, suggesting their value in optimizing hierarchical time series forecasting. The findings indicate that careful selection of reconciliation methods is essential for maximizing the performance of MLP-based models, especially in complex retail forecasting scenarios where both aggregated and disaggregated forecasts are crucial for operational decision-making.

4.6.3 Reconciled Probabilistic Forecasts

4.6.3.1 Benchmarks

Probabilistic forecasts using benchmark models reveal significant variability, as shown by the high Continuous Ranked Probability Score (CRPS) (Table 15). This high CRPS value indicates that the benchmark models are unable to provide reliable bounds for the forecast distribution. The benchmark models, which are mostly deterministic, struggle to account for the inherent uncertainty in sales forecasts, leading to distributions that fail to capture the full spectrum of possible future sales scenarios.

The inability of the benchmark models to provide accurate probabilistic bounds can have real-world implications, such as overestimating or underestimating stock requirements. This highlights the necessity of using more sophisticated probabilistic models to better estimate uncertainty, especially in volatile retail environments where precise inventory levels are crucial to avoid either stockouts or overstocking.

		BottomUp		MinTrace-OLS		MinTrace-WLS_struct		MinTrace-WLS_var		MinTrace-mint_shrink	
	metric	ARIMA	ETS	ARIMA	ETS	ARIMA	ETS	ARIMA	ETS	ARIMA	ETS
Total	**scaled_crps**	0,104	0,068	0,084	0,069	0,082	0,066	0,084	0,066	0,079	**0,066**
State	**scaled_crps**	0,123	0,091	0,101	0,094	0,102	0,090	0,105	0,090	0,096	**0,090**
Store	**scaled_crps**	0,150	0,127	0,139	0,128	0,136	0,126	0,137	0,126	0,128	**0,125**
Category	**scaled_crps**	0,123	0,091	0,102	0,094	0,102	0,091	0,105	**0,090**	0,099	0,091
Department	**scaled_crps**	0,144	0,120	0,132	0,118	0,128	0,118	0,130	**0,117**	0,127	0,120
State-Category	**scaled_crps**	0,160	0,134	0,143	0,137	0,142	0,134	0,145	**0,134**	0,140	0,135
State-Department	**scaled_crps**	0,191	0,174	0,186	0,173	0,180	0,173	0,181	**0,172**	0,177	0,174
Store-Category	**scaled_crps**	0,218	0,201	0,211	0,201	0,207	**0,199**	0,208	0,200	0,204	0,200
Store-Department	**scaled_crps**	0,277	0,265	0,280	0,264	0,269	0,263	0,269	**0,263**	0,265	0,264
Product	**scaled_crps**	0,294	**0,281**	0,296	0,295	0,286	0,287	0,286	0,281	0,285	0,284
State-Product	**scaled_crps**	0,423	**0,418**	0,430	0,431	0,420	0,424	0,419	0,420	0,419	0,422
Bottom	**scaled_crps**	0,658	0,662	0,666	0,674	0,657	0,667	0,656	0,662	**0,656**	0,665
All	**scaled_crps**	0,239	0,219	0,231	0,223	0,226	0,220	0,227	**0,218**	0,223	0,220

Table 15 - Reconciled probabilistic forecasts – benchmarks.

The probabilistic forecast evaluation employed the Continuous Ranked Probability Score (CRPS), scaled to assess the accuracy of predicted probability distributions against the observed values. The CRPS evaluation included various reconciliation strategies, specifically applied to benchmark models ARIMA and ETS, such as BottomUp, MinTrace-OLS, MinTrace-WLS_struct, MinTrace-WLS_var, and MinTrace-mint_shrink. The performance was evaluated across different hierarchical levels, ranging from aggregated levels like Total to disaggregated levels such as State-Department and State-Product.

The BottomUp reconciliation strategy yielded distinct results for both ARIMA and ETS models. At the Total level, ETS achieved a scaled CRPS of 0.067, which was significantly lower than ARIMA's value of 0.104. This demonstrates ETS's better capability in handling aggregate series probabilistically. Throughout other hierarchical levels, such as State, Store, and Department, ETS consistently outperformed ARIMA. For instance, at the Store level, ETS recorded a scaled CRPS of 0.127 versus ARIMA’s 0.150, indicating that ETS provides better-calibrated probabilistic forecasts when using a direct aggregation approach.

For the MinTrace-OLS reconciliation strategy, substantial improvements in accuracy were observed for both models. At the Total level, ETS achieved a scaled CRPS of 0.069, outperforming ARIMA, which recorded 0.084. A similar trend was observed across other levels such as State and Category. At the Category level, ETS had a scaled CRPS of 0.094 compared to ARIMA's 0.102. This result highlights the efficiency of MinTrace-OLS reconciliation in improving the probabilistic accuracy of hierarchical forecasts, particularly benefiting ETS's inherent structural adaptability to seasonal and trend components.

MinTrace-WLS_struct reconciliation demonstrated favorable results, especially for ETS, which generally recorded lower scaled CRPS values compared to ARIMA. At the Total level, ETS achieved a CRPS of 0.066 versus ARIMA’s 0.082, showcasing a notable reduction in forecast uncertainty when the WLS_struct framework was applied. This reconciliation strategy also provided competitive results at granular levels. For example, at the Store level, ETS achieved a scaled CRPS of 0.126 compared to ARIMA’s 0.136, supporting the argument that WLS_struct is effective for enhancing coherence in probabilistic forecasts across different hierarchy levels.

MinTrace-WLS_var also proved to be highly effective in reducing forecast errors. ETS, under this reconciliation framework, recorded the lowest CRPS at the Total level, with a value of 0.066, while ARIMA had a higher CRPS of 0.084. This indicates that variance-based reconciliation enhances the sharpness and reliability of probabilistic forecasts, especially for ETS. Moreover, at detailed hierarchical levels such as State-Department and Store-Category, ETS continued to show lower CRPS values, demonstrating its robustness when capturing uncertainty and preserving consistency across the hierarchy.

The MinTrace-mint_shrink reconciliation strategy provided some of the best results in terms of probabilistic accuracy, particularly for ETS. At the Total level, ETS achieved a scaled CRPS of 0.066, which was the lowest CRPS observed among all reconciliation methods for both ARIMA and ETS models. This illustrates that MinTrace-mint_shrink effectively combines shrinkage techniques with reconciliation to significantly reduce forecast uncertainty. Furthermore, at the Product level, ETS recorded a scaled CRPS of 0.284, lower than ARIMA's 0.285, which again demonstrated the superiority of ETS under this reconciliation strategy.

In terms of lower-level categories, such as State-Product and Bottom, the CRPS values remained relatively high, reflecting a greater challenge in achieving accurate probabilistic forecasts at granular levels. For example, the State-Product level saw ETS with a CRPS of 0.418 under the BottomUp approach, while ARIMA recorded 0.423. Despite ETS's slight outperformance, the overall CRPS was high, highlighting the difficulty in forecasting highly variable, granular data. Similarly, the Bottom level produced the highest CRPS values, with ETS reaching 0.662 for BottomUp and 0.665 for MinTrace-mint_shrink. These results imply that although reconciliation enhances forecast quality, more refined approaches might still be required to handle disaggregated series.

In summary, the analysis of probabilistic forecasts benchmarks revealed that ETS consistently outperformed ARIMA in terms of CRPS values, particularly when combined with advanced reconciliation techniques like MinTrace-OLS, MinTrace-WLS_var, and MinTrace-mint_shrink. Among the reconciliation methods, MinTrace-WLS_var and MinTrace-mint_shrink demonstrated the lowest overall CRPS values, especially for the ETS model, indicating their effectiveness for hierarchical probabilistic forecasting. These findings underscore the critical role of selecting appropriate reconciliation frameworks to improve forecast accuracy, especially in contexts that require coherence across different hierarchical levels, such as retail demand forecasting. Furthermore, while reconciliation

methods significantly enhance forecast quality, achieving low uncertainty remains a challenge for more granular levels, thus necessitating careful model and reconciliation strategy selection for optimal performance in hierarchical time series forecasting.

4.6.3.2 Transformer-based Models

The Transformer-based models, including Autoformer and Informer, demonstrate substantial improvements in probabilistic forecasting capabilities (Table 16). The reduction in CRPS values compared to benchmarks indicates that these models are effective at capturing the full range of possible outcomes, which is essential in probabilistic forecasting. These models leverage self-attention mechanisms, which help in modeling complex temporal relationships and uncertainties inherent in retail sales data.

The lower CRPS scores demonstrate that Transformer models are effective in predicting not only point estimates but also in providing a reliable probability distribution. This enables better decision-making under uncertainty by allowing inventory and supply chain managers to plan for a range of outcomes. The probabilistic forecasts provided by Transformer models are more reliable, which makes them suitable for risk-averse strategies where decision-makers need to account for potential variability in demand.

The analysis of probabilistic forecasts using Transformer-based models was conducted across various reconciliation methods, including BottomUp, MinTrace-OLS, MinTrace-WLS_struct, MinTrace-WLS_var, and MinTrace-mint_shrink. These models were evaluated using the scaled Continuous Ranked Probability Score (scaled_CRPS), which measures the accuracy of probabilistic forecasts across different hierarchical levels.

The BottomUp reconciliation approach, when applied to transformer-based models, showed mixed performance. For instance, the Patch TST model exhibited the lowest scaled_CRPS at the "Total" level with a value of 0.091, indicating effective forecasting for aggregated levels. However, at more granular levels like "State" and "Store," the Vanilla Transformer yielded slightly better results with scaled_CRPS values of 0.122 and 0.150, respectively, compared to other models. This suggests that while Patch TST may perform well in aggregated contexts, other models, such as the Vanilla Transformer, may provide greater accuracy in more detailed disaggregated scenarios. Nonetheless, at the most disaggregated levels, such as "Product" and "Store-Category," the BottomUp approach produced higher error rates.

		BottomUp					MinTrace-OLS					MinTrace-WLS_struct					MinTrace-WLS_var					MinTrace-mint_shrink				
	metric	**Vanilla Transformer**	**TFT**	**Informer**	**Patch TST**	**Autoformer**	**Vanilla Transformer**	**TFT**	**Informer**	**Patch TST**	**Autoformer**	**Vanilla Transformer**	**TFT**	**Informer**	**Patch TST**	**Autoformer**	**Vanilla Transformer**	**TFT**	**Informer**	**Patch TST**	**Autoformer**	**Vanilla Transformer**	**TFT**	**Informer**	**Patch TST**	**Autoformer**
Total	**scaled_crps**	0,106	0,125	0,105	0,091	0,102	0,073	0,103	0,075	0,071	0,069	0,077	0,103	0,078	0,070	0,072	0,079	0,103	0,080	0,071	0,074	**0,069**	0,106	0,084	0,069	0,073
State	**scaled_crps**	0,122	0,139	0,122	0,108	0,120	0,093	0,122	0,095	0,096	0,093	0,096	0,122	0,097	0,093	0,095	0,098	0,122	0,098	0,093	0,096	**0,086**	0,123	0,100	0,089	0,094
Store	**scaled_crps**	0,150	0,166	0,150	0,138	0,150	0,125	0,152	0,129	0,128	0,129	0,127	0,152	0,130	0,125	0,129	0,129	0,152	0,131	0,126	0,130	**0,117**	0,148	0,130	0,120	0,126
Category	**scaled_crps**	0,120	0,138	0,121	0,107	0,118	0,095	0,121	0,095	0,095	0,092	0,097	0,121	0,097	0,092	0,094	0,098	0,121	0,098	0,092	0,095	**0,090**	0,123	0,104	0,091	0,094
Department	**scaled_crps**	0,136	0,152	0,139	0,129	0,135	0,122	0,142	0,121	0,121	0,121	0,122	0,141	0,122	0,120	0,119	0,121	0,139	0,121	0,118	0,117	**0,114**	0,141	0,132	0,119	0,119
State-Category	**scaled_crps**	0,154	0,169	0,154	0,143	0,153	0,135	0,160	0,136	0,138	0,135	0,136	0,158	0,136	0,135	0,135	0,137	0,157	0,137	0,134	0,136	**0,128**	0,157	0,140	0,131	0,132
State-Department	**scaled_crps**	0,183	0,196	0,185	0,178	0,182	0,173	0,191	0,172	0,174	0,173	0,172	0,188	0,171	0,172	0,170	0,171	0,187	0,171	0,171	0,169	**0,164**	0,186	0,179	0,170	0,168
Store-Category	**scaled_crps**	0,213	0,226	0,212	0,204	0,213	0,201	0,219	0,205	0,203	0,203	0,200	0,217	0,202	0,199	0,201	0,201	0,217	0,202	0,199	0,201	**0,192**	0,211	0,203	0,196	0,197
Store-Department	**scaled_crps**	0,268	0,278	0,268	0,265	0,268	0,264	0,279	0,266	0,266	0,267	0,261	0,275	0,263	0,263	0,263	0,262	0,274	0,263	0,262	0,262	**0,254**	0,269	0,269	0,260	0,260
Product	**scaled_crps**	0,285	0,293	0,290	0,281	0,285	0,281	0,294	0,282	0,285	0,282	0,278	0,289	0,280	0,282	0,280	0,277	0,287	0,279	0,279	0,278	**0,269**	0,284	0,291	0,278	0,278
State-Product	**scaled_crps**	**0,405**	0,409	0,410	0,408	0,408	0,415	0,423	0,415	0,421	0,419	0,410	0,417	0,412	0,416	0,414	0,409	0,414	0,410	0,412	0,411	0,406	0,410	0,422	0,413	0,411
Bottom	**scaled_crps**	**0,615**	0,615	0,619	0,639	0,630	0,657	0,659	0,656	0,664	0,661	0,647	0,649	0,648	0,657	0,653	0,640	0,641	0,643	0,651	0,645	0,653	0,644	0,666	0,656	0,653
All	**scaled_crps**	0,230	0,242	0,231	0,224	0,230	0,219	0,239	0,221	0,222	0,220	0,219	0,236	0,220	0,219	0,219	0,219	0,235	0,220	0,217	0,218	**0,212**	0,233	0,227	0,216	0,217

Table 16 - Reconciled probabilistic forecasts – Transformers.

Specifically, the scaled_CRPS values reached 0.285 for Vanilla Transformer at the "Product" level and 0.213 at the "Store-Category" level, highlighting the limitations of the BottomUp approach for finer details in hierarchical time series. The MinTrace-OLS reconciliation strategy significantly enhanced the accuracy of probabilistic forecasts for transformer models across most hierarchical levels. At the "Total" level, Patch TST achieved a scaled_CRPS of 0.071, which was notably the lowest among the models tested, indicating the effectiveness of OLS-based reconciliation in capturing hierarchical dependencies. Autoformer also performed well under MinTrace-OLS, achieving a scaled_CRPS of 0.069, further supporting its robustness for forecasting tasks involving complex hierarchies. Improvements were also seen in "State" and "Store" levels, where models such as Autoformer and Vanilla Transformer achieved scaled_CRPS values of 0.093 and 0.129, respectively. These improvements emphasize the advantage of using MinTrace-OLS to ensure more reliable forecasts.

MinTrace-WLS_struct also yielded strong results for transformer models, offering improved consistency across multiple levels. The Patch TST model demonstrated the lowest scaled_CRPS of 0.070 at the "Total" level and achieved a value of 0.092 at the "Category" level, reflecting the efficiency of WLS-based structural reconciliation in reducing forecast errors while maintaining coherence across different levels of aggregation. Vanilla Transformer also showed relatively low scaled_CRPS values, for example, 0.097 at the "Category" level, indicating that WLS_struct is a promising approach for enhancing forecast coherence in hierarchical contexts.

The MinTrace-WLS_var reconciliation strategy demonstrated excellent effectiveness in minimizing forecast errors for transformer-based models. Specifically, Patch TST achieved a scaled_CRPS of 0.071 at the "Total" level, which indicates strong performance in capturing hierarchical dependencies effectively. Autoformer also showed positive results under this strategy, yielding a scaled_CRPS of 0.074 at the "Total" level and 0.093 at the "State" level. This performance suggests that MinTrace-WLS_var is a versatile and effective reconciliation approach for both aggregated and disaggregated levels, thereby supporting consistent forecast quality across hierarchical structures.

The MinTrace-mint_shrink reconciliation produced notable results, particularly in minimizing uncertainty and enhancing forecast quality. Vanilla Transformer exhibited the lowest scaled_CRPS at the "Total" level, with a value of 0.069, indicating effective capturing of the overall hierarchical structure and reducing forecast uncertainty.

Similarly, Patch TST and Autoformer also delivered solid performances under MinTrace-mint_shrink, with scaled_CRPS values of 0.069 and 0.073, respectively, at the "Total" level. These values demonstrate that the mint_shrink reconciliation is effective in maintaining consistency across hierarchical levels while reducing forecasting errors, making it well-suited for contexts where detailed forecasting across aggregated and disaggregated levels is crucial.

The probabilistic forecasting analysis of transformer-based models clearly indicates that the choice of reconciliation strategy plays a significant role in determining the effectiveness of these models. Among the models analyzed, Patch TST consistently delivered strong performance, especially when applied with MinTrace-OLS and MinTrace-WLS_struct reconciliation methods, achieving some of the lowest scaled_CRPS values observed. Autoformer similarly demonstrated robustness across different hierarchical levels, particularly when applied with MinTrace-WLS_var and MinTrace-mint_shrink, positioning it as a strong candidate for hierarchical forecasting tasks that involve complex data structures. The findings emphasize that sophisticated reconciliation methods such as MinTrace-OLS, MinTrace-WLS_struct, and MinTrace-mint_shrink provide significant improvements in probabilistic forecasting performance over simpler approaches like BottomUp. This improvement is especially relevant in retail forecasting, where maintaining consistency across various levels of aggregation, such as products, categories, and overall sales, is essential for operational planning and decision-making. Thus, selecting appropriate transformer models and pairing them with effective reconciliation strategies is critical for optimizing forecasting accuracy and reliability in hierarchical retail environments.

4.6.3.3 MLP-based Models

MLP models also offer a significant improvement in probabilistic forecasts compared to benchmark models, though not as effective as Transformer-based models (Table 17). The CRPS values for MLPs are lower than those observed with traditional benchmarks but higher compared to Transformers, indicating a moderate ability to quantify uncertainty. While MLPs can model non-linear relationships effectively, they lack the sequence modeling capabilities inherent to Transformer models, which affects their ability to capture temporal dependencies across hierarchical levels.

Despite these limitations, MLPs still provide a reasonable estimate of uncertainty, which can be useful when computational efficiency is a priority. The reduced CRPS values, relative to benchmarks, show that MLPs are capable of providing useful probabilistic information, albeit less precise than that provided by Transformer architectures. This suggests that MLPs can still serve as a viable alternative in situations where quick turnaround times are required, though they should be supplemented with reconciliation methods to ensure coherence across hierarchical levels.

For the probabilistic forecasts of MLP models, a comprehensive analysis was conducted across multiple reconciliation methods: BottomUp, MinTrace-OLS, MinTrace-WLS_struct, MinTrace-WLS_var, and MinTrace-mint_shrink. The assessment was performed using the scaled Continuous Ranked Probability Score (scaled_CRPS), which quantifies the accuracy of probabilistic forecasts for different hierarchical levels.

Under the BottomUp reconciliation approach, the NHITS model recorded the lowest scaled_CRPS values for several levels, notably achieving a value of 0.090 at the "Total" level, outperforming both MLP and NBEATS, which had scaled_CRPS values of 0.099 and 0.096, respectively. At the "State" level, NHITS also performed well with a scaled_CRPS of 0.106, while MLP and NBEATS showed slightly higher error values of 0.113 and 0.112. These results indicate that NHITS is the more effective model under the BottomUp approach, particularly for aggregated hierarchical levels. At finer granularities, such as the "Store-Category" and "Store-Department" levels, the scaled_CRPS values for NHITS were 0.206 and 0.264, indicating that NHITS maintained its comparative advantage over the other models for more disaggregated data.

The MinTrace-OLS reconciliation method was effective in enhancing the forecast quality across all three models. For the "Total" level, the NBEATS model yielded the lowest scaled_CRPS of 0.06, indicating a significant improvement in accuracy compared to BottomUp. NHITS also performed comparably well, with a scaled_CRPS of 0.065, demonstrating its reliability under OLS reconciliation. At the "State" level, both NBEATS and NHITS continued to show strong performance with scaled_CRPS values of 0.088 and 0.089, respectively, compared to MLP's 0.090. This suggests that MinTrace-OLS is effective in improving forecast coherence across hierarchical levels, with NBEATS showing a slight edge in minimizing forecast errors.

		BottomUp			MinTrace-OLS			MinTrace-WLS_struct			MinTrace-WLS_var			MinTrace-mint_shrink		
	metric	MLP	NBEATS	NHITS	MLP	NBEATS	NHITS	MLP	NBEATS	NHITS	MLP	NBEATS	NHITS	MLP	NBEATS	NHITS
Total	**scaled_crps**	0,097	0,096	0,090	0,068	0,064	0,065	0,068	0,061	0,062	0,069	0,060	0,062	0,073	**0,056**	0,066
State	**scaled_crps**	0,113	0,112	0,106	0,090	0,088	0,089	0,089	0,083	0,085	0,090	0,082	0,085	0,091	**0,076**	0,084
Store	**scaled_crps**	0,142	0,141	0,137	0,124	0,121	0,124	0,124	0,118	0,121	0,124	0,117	0,121	0,122	**0,108**	0,116
Category	**scaled_crps**	0,113	0,110	0,106	0,090	0,089	0,092	0,089	0,084	0,088	0,090	0,083	0,087	0,092	**0,079**	0,090
Department	**scaled_crps**	0,130	0,129	0,124	0,117	0,115	0,116	0,114	0,110	0,112	0,113	0,107	0,110	0,118	**0,105**	0,115
State-Category	**scaled_crps**	0,146	0,144	0,143	0,130	0,132	0,139	0,129	0,126	0,134	0,129	0,125	0,133	0,130	**0,118**	0,129
State-Department	**scaled_crps**	0,176	0,175	0,173	0,168	0,169	0,174	0,165	0,163	0,168	0,164	0,162	0,166	0,167	**0,157**	0,167
Store-Category	**scaled_crps**	0,206	0,207	0,206	0,197	0,198	0,205	0,196	0,194	0,200	0,196	0,193	0,200	0,193	**0,183**	0,194
Store-Department	**scaled_crps**	0,261	0,263	0,264	0,259	0,261	0,267	0,256	0,256	0,262	0,256	0,254	0,261	0,256	**0,248**	0,259
Product	**scaled_crps**	0,277	0,276	0,278	0,281	0,279	0,280	0,277	0,274	0,277	0,274	0,271	0,277	0,275	**0,266**	0,275
State-Product	**scaled_crps**	0,402	**0,401**	0,407	0,415	0,418	0,422	0,410	0,411	0,416	0,408	0,407	0,414	0,409	0,404	0,414
Bottom	**scaled_crps**	0,624	**0,623**	0,636	0,652	0,659	0,666	0,645	0,650	0,659	0,640	0,641	0,652	0,646	0,643	0,655
All	**scaled_crps**	0,224	0,223	0,223	0,216	0,216	0,220	0,214	0,211	0,215	0,213	0,209	0,214	0,214	**0,204**	0,214

Table 17 - Reconciled probabilistic forecasts – MLP.

MinTrace-WLS_struct reconciliation provided further reductions in forecasting errors, particularly for the NBEATS model. At the "Total" level, NBEATS achieved a scaled_CRPS of 0.060, which was the lowest among all three models, indicating the benefit of WLS structural reconciliation in reducing forecast uncertainty. The MLP model also showed improved accuracy, achieving a scaled_CRPS of 0.068, though it was not as effective as NBEATS. At the "Department" level, NBEATS recorded a scaled_CRPS of 0.110, again outperforming both MLP (0.114) and NHITS (0.112). These findings underscore the effectiveness of the MinTrace-WLS_struct method, particularly when applied to NBEATS, in enhancing accuracy and coherence across the hierarchy.

The MinTrace-WLS_var approach was similarly effective, showing competitive results for both NBEATS and NHITS. At the "Total" level, NBEATS had a scaled_CRPS of 0.060, which was slightly better than NHITS at 0.062. MLP, while still benefiting from this approach, recorded a higher scaled_CRPS of 0.069. At the "State" level, the NBEATS model again had the lowest scaled_CRPS of 0.082, compared to MLP's 0.090 and NHITS's 0.085. The findings indicate that MinTrace-WLS_var helps maintain consistency while effectively reducing the overall error rate, making it particularly suitable for models like NBEATS that can leverage its variance-based reconciliation strategy.

MinTrace-mint_shrink reconciliation method exhibited notable effectiveness, especially for NBEATS. At the "Total" level, NBEATS achieved the lowest scaled_CRPS value of 0.056, outperforming both MLP (0.073) and NHITS (0.066). This pattern continued across other levels, with NBEATS achieving the lowest scaled_CRPS values at the "State" (0.076) and "Store" (0.108) levels as well. MLP also benefited from the MinTrace-mint_shrink reconciliation, but it did not reach the level of performance shown by NBEATS. For example, at the "State-Department" level, NBEATS recorded a scaled_CRPS of 0.157, which was lower compared to MLP's 0.167 and NHITS's 0.167, suggesting that MinTrace-mint_shrink provides significant advantages for reducing forecast errors across complex hierarchical structures.

The analysis of probabilistic forecasts using MLP, NBEATS, and NHITS models across different reconciliation strategies highlights the effectiveness of advanced reconciliation methods such as MinTrace-OLS, MinTrace-WLS_struct, and MinTrace-mint_shrink in improving forecasting accuracy. NBEATS consistently demonstrated superior performance across multiple reconciliation frameworks, particularly under MinTrace-

mint_shrink and MinTrace-WLS_struct, achieving the lowest scaled_CRPS values across different levels. NHITS also showed competitive results, especially under the BottomUp and MinTrace-OLS approaches, indicating its reliability for both aggregated and disaggregated forecasting tasks. MLP, while showing improvements, did not consistently outperform NBEATS or NHITS, indicating that the choice of reconciliation strategy plays a significant role in determining its effectiveness. Overall, these findings emphasize that pairing appropriate reconciliation methods with advanced forecasting models can significantly enhance accuracy and reliability in hierarchical probabilistic forecasting, particularly in complex retail settings where forecast coherence across multiple levels is crucial for effective planning and decision-making.

5 Conclusions

This dissertation develops a comprehensive framework to address the complexities of hierarchical forecasting in business operations, particularly in the context of demand planning, sales forecasting, and inventory management. By focusing on the unique challenges of the retail sector, where accurate forecasting is essential for optimizing logistics, minimizing inventory levels, and ensuring customer satisfaction, this research aimed to enhance traditional forecasting methods that often fail to capture the intricate hierarchical relationships within organizations.

The core objective of this research was to design a hierarchical forecasting system that could effectively integrate deep learning architectures while adapting to the varying structures of business units and product categories. To achieve this, the framework was constructed to accommodate different hierarchical levels, ensuring coherence across aggregated and disaggregated forecasts. This adaptability allows the framework to respond to changes in organizational structure, product portfolios, and market dynamics, making it a scalable and robust solution for businesses operating in dynamic environments.

The study specifically examines the potential of two groups of deep learning models: Transformer-based models (including the Vanilla Transformer, Autoformer, ETSFormer, Informer, NSTransformer, and Reformer) and MLP-based models (MLP, NBEATS, and NHITS), to capture complex temporal dependencies and hierarchical relationships within time series data. The research explores both point and probabilistic forecasting methodologies, with the goal of enhancing prediction accuracy and reliability across different levels of the hierarchy.

The comprehensive analysis of both point and probabilistic forecasts across various hierarchical forecasting models and reconciliation techniques reveals that the NBEATS and NHITS models, coupled with advanced reconciliation methods, consistently yield superior results in terms of accuracy and coherence across multiple hierarchical levels. Specifically, for point forecasts, the NBEATS model, particularly under the MinTrace-OLS and MinTrace-WLS_struct reconciliation methods, demonstrated the lowest error metrics with Mean Absolute Scaled Error (MASE) values as low as 0.682 and 0.660, and Mean Squared Scaled Error (MSSE) values reaching 0.433 and 0.414 at the Total level, respectively. These values reflect a significant improvement in forecast quality and

highlight NBEATS as a robust model for hierarchical time series forecasting when accuracy at various levels of aggregation is required.

NHITS also showed competitive performance, particularly under the MinTrace-mint_shrink reconciliation strategy, which was reflected in consistently low MASE and MSSE values across different hierarchical structures. NHITS achieved notable results at the Store-Department level, achieving MASE values of 0.942, demonstrating its efficacy in managing disaggregated forecast tasks while maintaining coherence across different levels. In probabilistic forecasting, NBEATS again emerged as the best performing model, with the MinTrace-mint_shrink reconciliation strategy producing the lowest scaled Continuous Ranked Probability Score (scaled_CRPS) values, such as 0.056 at the Total level. This indicates that NBEATS not only excels in point forecast accuracy but also in probabilistic forecast reliability, making it a suitable choice for environments where uncertainty quantification is vital.

The analysis also indicates that advanced reconciliation methods play a critical role in enhancing forecast performance. Specifically, MinTrace-WLS_struct and MinTrace-mint_shrink were found to be highly effective when applied to both NBEATS and NHITS, resulting in the lowest observed errors across hierarchical levels, particularly for more disaggregated data like the Store-Department and State-Category levels.

In contrast, the traditional methods, such as ARIMA and ETS, even when combined with reconciliation strategies, did not consistently outperform the more advanced neural network-based models, although ETS combined with MinTrace-WLS_var did yield competitive error rates in some scenarios.

In conclusion, NBEATS and NHITS, especially when paired with sophisticated reconciliation techniques like MinTrace-mint_shrink and MinTrace-WLS_struct, provide the most accurate and reliable forecasts for both point and probabilistic hierarchical time series tasks. These models demonstrated the lowest error metrics across the board, making them the preferred choice for complex hierarchical forecasting requirements where accuracy and consistency at different aggregation levels are paramount.

The results highlight the necessity of integrating advanced reconciliation methods with state-of-the-art forecasting models to achieve the best possible outcomes in hierarchical forecasting contexts, such as retail and supply chain management.

A promising direction for future research is the development of an integrated model that combines both base and reconciled predictions. This approach would leverage the strengths of individual forecasting models while ensuring consistency across hierarchical prediction levels. By integrating base models with a reconciliation mechanism, the model could achieve both localized accuracy and global coherence.

The integrated model could use hierarchical forecasting algorithms or deep learning-based reconciliation layers to align individual base predictions with aggregate predictions, thereby improving accuracy in hierarchical or grouped time-series scenarios. An adaptive approach, employing dynamic weighting or attention mechanisms, could further enhance this integration by prioritizing the best-performing models based on data characteristics.

Incorporating reconciled predictions into an integrated model offers a robust and scalable forecasting solution, particularly for complex datasets involving multiple aggregation levels. This approach bridges the gap between precision in local forecasts and overall consistency, providing a holistic solution to time-series forecasting challenges.

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., & Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*, 265–283.

Alon, I., Qi, M., & Sadowski, R. J. (2001). Forecasting aggregate retail sales: A comparison of artificial neural networks and traditional methods. *Journal of Retailing and Consumer Services, 8*(3), 147–156. https://doi.org/10.1016/S0969-6989(00)00012-1.

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *arXiv preprint* arXiv:1607.06450.

Ballestra, L. V., Guizzardi, A., & Palladini, F. (2019). Forecasting and trading on the VIX futures market: A neural network approach based on open to close returns and coincident indicators. *International Journal of Forecasting, 35*(4), 1250–1262.

Bandara, K., Shi, P., Bergmeir, C., Hewamalage, H., Tran, Q., & Seaman, B. (2019). Sales demand forecast in e-commerce using a long short-term memory neural network methodology. In T. Gedeon, K. Wong, & M. Lee (Eds.), *Neural Information Processing. ICONIP 2019. Lecture Notes in Computer Science* (Vol. 11955). Cham: Springer. https://doi.org/10.1007/978-3-030-36718-3_39.

Berrocal, V. J., Raftery, A. E., Gneiting, T., & Steed, R. C. (2010). Probabilistic weather forecasting for winter road maintenance. *Journal of the American Statistical Association, 105*(490), 522–537.

Böse, J., Flunkert, V., Gasthaus, J., Januschowski, T., Lange, D., Salinas, D., Schelter, S., Seeger, M., & Wang, Y. (2017). Probabilistic demand forecasting at scale. *Proceedings of the VLDB Endowment, 10*, 1694–1705. https://doi.org/10.14778/3137765.3137775.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory* (pp. 144–152). ACM.

Callot, L., Kock, A. B., & Medeiros, M. C. (2017). Modelling and forecasting large, realized covariance matrices and portfolio choice. *Journal of Applied Econometrics, 32*(1), 140–158.

Callot, L., Caner, M., Önder, A. Ö., & Ulaşan, E. (2019). Anodewise regression approach to estimating large portfolios. *Journal of Business & Economic Statistics*, 1–12.

Challu, C., Olivares, K. G., Oreshkin, B. N., Garza, F., Mergenthaler-Canseco, M., & Dubrawski, A. (2023). N-HiTS: Neural hierarchical interpolation for time series forecasting. *Association for the Advancement of Artificial Intelligence*.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM.

Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C., & Zhang, Z. (2015). *MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems.* In *Proceedings of the NeurIPS Workshop on Machine Learning Systems*.

Churpek, M. M., Adhikari, R., & Edelson, D. P. (2016). The value of vital sign trends for detecting clinical deterioration on the wards. *Resuscitation, 102,* 1–5.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A seasonal-trend decomposition procedure based on Loess. *Journal of Official Statistics, 6*(3), 3–73.

Crăcan, M. (2020). *LSTM vs ARIMA in retail forecasting: A comparative study* (Master's thesis). Erasmus University Rotterdam. Retrieved from https://thesis.eur.nl/pub/53546/.

Croston, J. D. (1972). Forecasting and stock control for intermittent demands. *Operational Research Quarterly, 23,* 289–303. http://dx.doi.org/10.2307/3007885.

Dimoulkas, I., Mazidi, P., & Herre, L. (2019). Neural networks for GEFCom2017 probabilistic load forecasting. *International Journal of Forecasting, 35*(4), 1409–1423.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations.*

Faloutsos, C., Flunkert, V., Gasthaus, J., Januschowski, T., & Wang, Y. (2019). Forecasting big time series: Theory and practice. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.*

Faloutsos, C., Flunkert, V., Gasthaus, J., Januschowski, T., & Wang, Y. (2020). Forecasting big time series: Theory and practice. In *Proceedings of the Companion Proceedings of the Web Conference 2020* (pp. 320–321). Association for Computing Machinery.

Faloutsos, C., Gasthaus, J., Januschowski, T., & Wang, Y. (2018). Forecasting big time series: Old and new. *Proceedings of the VLDB Endowment, 11*(12), 2102–21055.

Faloutsos, C., Gasthaus, J., Januschowski, T., & Wang, Y. (2019). Classical and contemporary approaches to big time series forecasting. In *Proceedings of the 2019 International Conference on Management of Data.* ACM, New York, NY.

Field, C. B., Barros, V., Stocker, T. F., & Dahe, Q. (2012). *Managing the risks of extreme events and disasters to advance climate change adaptation.* Cambridge University Press.

Fisher, M., & Raman, A. (2018). Using data and big data in retailing. *Production and Operations Management, 27*, 1665–1669.

Gasthaus, J., Benidis, K., Wang, Y., Rangapuram, S. S., Salinas, D., Flunkert, V., & Januschowski, T. (2019). Probabilistic forecasting with spline quantile function RNNs. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics* (pp. 1901–1910).

Gneiting, T., & Katzfuss, M. (2014). Probabilistic forecasting and statistical modeling.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*(7), 1527–1554.

Ho, T. K. (1995). Random decision forests. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition* (pp. 278–282). IEEE.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

Hong, T., Pinson, P., & Fan, S. (2014). GEFCOM2012 global energy forecasting competition 2012.

Hu, M. J. C., & Root, H. E. (1964). An adaptive data processing system for weather forecasting. *Journal of Applied Meteorology, 3*(5), 513.

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*. OTexts.

Januschowski, T., & Kolassa, S. (2019). A classification of business forecasting problems. *Foresight: The International Journal of Applied Forecasting, 52*, 36–43.

Januschowski, T., Wang, Y., Hasson, H., Erkkila, T., Torkkila, K., & Gasthaus, J. (2021). Forecasting with trees. *International Journal of Forecasting, 37*(3), 1158–1176.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems, 30*, 3146–3154.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint* arXiv:1312.6114.

Kitaev, N., Kaiser, L., & Levskaya, A. (2019). Reformer: The efficient transformer. *arXiv preprint* arXiv:2001.04451.

Zhou, H., Zhang, S., Peng, J., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems* (pp. 3104–3112).

Laptev, N., Yosinski, J., Li, L., & Smyl, S. (2017). Time-series extreme event forecasting with neural networks at Uber. In *Proceedings of the ICML Time Series Workshop*.

Längkvist, M., Karlsson, L., & Loutfi, A. (2014). A review of unsupervised feature learning and deep learning for time-series modelling. *Pattern Recognition Letters, 42*, 11–24.

LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. In M. A. Arbib (Ed.), *The Handbook of Brain Theory and Neural Networks* (p. 3361). MIT Press.

LeCun, Y., Jackel, L. D., Bottou, L., Brunot, A., Cortes, C., Denker, J. S., Drucker, H., Guyon, I., Muller, U. A., Sackinger, E., Simard, P., & Vapnik, V. (1995). Comparison of learning algorithms for handwritten digit recognition. In *Proceedings of the International Conference on Artificial Neural Networks* (pp. 53–60).

Levy, M., Weitz, B. A., & Grewal, D. (2012). *Retailing management* (8th ed.). Irwin.
Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data driven traffic forecasting. In *Proceedings of the International Conference on Learning Representations*.

Li, X., Shang, W., & Wang, S. (2019). Text-based crude oil price forecasting: A deep learning approach. *International Journal of Forecasting, 35*(4), 1548–1560.

Li, X., Zhang, Y., & Wang, H. (2023). Transformer-based models for Retail Forecasting. MDPI. Retrieved from https://www.mdpi.com/2227-7390/12/17/2728

Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting, 37*(4), 1748–1760.

Luo, R., Zhang, W., Xu, X., & Wang, J. (2018). A neural stochastic volatility model. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F.-Y. (2014). Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems, 16*(2), 865–873.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). The M4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting, 34*(4), 802–808.

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

Makridakis, S., Spiliotis, E., Assimakopoulos, V., Chen, Z., Gaba, A., Tsetlin, I., & Winkler, R. L. (2021). The M5 uncertainty competition: Results, findings, and conclusions. *International Journal of Forecasting*.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). The M4 Competition: Results, findings, conclusion, and way forward. *International Journal of Forecasting, 34*(4), 802–808.

Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2023). A time series is worth 64 words: Long-term forecasting with transformers. *International Conference on Learning Representations (ICLR)*.

Oliveira, J. M., & Ramos, P. (2024). Evaluating the effectiveness of time series transformers for demand forecasting in retail. *Mathematics, 12*(17), 2728. https://doi.org/10.3390/math12172728

Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). N-BEATS: Neural basis expansion analysis for time series forecasting. *International Conference on Learning Representations (ICLR)*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., & Antiga, L. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems, 32*.

Petropoulos, F., Kourentzes, N., Trapero, J. R., & Assimakopoulos, V. (2020). Forecasting: Theory and practice. *International Journal of Forecasting, 36*(4), 1347–1360.

Rosenblatt, F. (1957). The Perceptron, a perceiving and recognizing automaton (Report 85-60-1). Cornell Aeronautical Laboratory.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning internal representations by error propagation (Technical Report). University of California San Diego, La Jolla Institute for Cognitive Science.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533–536.

Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting, 36*(3), 1181–119.

Saxena, H., Aponte, O., & McConky, K. T. (2019). A hybrid machine learning model for forecasting a billing period's peak electric load days. *International Journal of Forecasting, 35*(4), 1288–1303.

Smyl, S., & Hua, N. G. (2019). Machine learning methods for GEFCom2017 probabilistic load forecasting. *International Journal of Forecasting, 35*(4), 1424–1431.

Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting, 36*(1), 75–85.

Taieb, S. B., Taylor, J. W., & Hyndman, R. J. (2020). Hierarchical probabilistic forecasting of electricity demand with smart meter data. *Journal of the American Statistical Association, 0*(0), 1–17.

Taylor, R., Huang, J., & Kumar, P. (2024). Hierarchical neural additive models for demand forecasting. *ArXiv*. Retrieved from https://arxiv.org/abs/2404.04070

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … & Polosukhin, I. (2017). Developed the multi-head attention mechanism, which serves as the foundation for capturing temporal dependencies.

Wickramasuriya, S. L., Athanasopoulos, G., & Hyndman, R. J. (2019). Optimal forecast reconciliation for hierarchical and grouped time series.

Wen, R., Torkkola, K., Narayanaswamy, B., & Madeka, D. (2017). A multi-horizon quantile recurrent forecaster. *arXiv preprint* arXiv:1711.11053.

Wen, R., Torkkola, K., Narayanaswamy, B., & Madeka, D. (2017). Applied quantile regression in time series forecasting, providing the basis for TFT's interval forecasting capabilities.

Yoo, J., & Kang, U. (2021). Attention-based autoregression for accurate and efficient multivariate time series forecasting. In *Proceedings of the 2021 SIAM International Conference on Data Mining* (pp. 531–539). SIAM.

Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2022). Are transformers effective for time series forecasting? *arXiv preprint* arXiv:2205.13504.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, *14*(1), 35–62.