FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

# Metalearning for Enhanced Algorithm Selection in Time Series Decomposition-Based Forecasting

**José Luís Barbosa de Araújo**


Master's Thesis in Informatics and Computing Engineering

Supervisors: Moisés Santos and Carlos Soares

July 31, 2025

# Metalearning for Enhanced Algorithm Selection in Time Series Decomposition-Based Forecasting

**José Luís Barbosa de Araújo**

Master's Thesis in Informatics and Computing Engineering

Approved in oral examination by the committee:

President: Prof. Ricardo Cruz
Examiner: Prof. Diego Silva
Supervisor: Prof. Moisés Santos

July 31, 2025

# Resumo

A previsão de séries temporais desempenha um papel crucial em vários domínios, como as finanças, os cuidados de saúde e a gestão de cadeias de abastecimento, onde previsões exatas são essenciais para potenciar atividades de tomada de decisões. Os métodos tradicionais de previsão, embora sejam eficazes em determinados cenários, não são ideais para detetar dinâmicas complexas presentes em séries temporais do mundo real. Os métodos de previsão baseados em decomposição de séries temporais melhoram a exatidão, isolando a tendência, a sazonalidade e os componentes residuais, mas encontrar um algoritmo adequado para cada componente continua a ser um problema. Esta tese aborda as limitações das atuais técnicas de metalearning baseadas na decomposição de séries temporais, particularmente as da framework METAFORE, introduzindo uma estratégia mais detalhada, baseada em componentes, para a seleção de algoritmos. Usando metalearning, o sistema proposto determina automaticamente se deve decompor uma série temporal e recomenda modelos de previsão adaptados a cada componente resultante. Uma metodologia sistemática e a avaliação do desempenho em 5000 séries temporais de frequência mensal do conjunto de dados de benchmarking do M4 competition demonstram a eficácia desta abordagem em relação às estratégias convencionais e de modelo único. Os resultados destacam a utilidade das recomendações de algoritmos por componente para melhorar o desempenho das previsões, o que tem implicações significativas para aplicações práticas e investigação futura em sistemas de previsão automatizados.

# Abstract

Time series forecasting plays a crucial role in a number of fields such as finance, healthcare, and supply chain management, where accurate forecasts are critical to inform decision-making activities. Traditional forecasting methods, while being effective in particular scenarios, are not optimal for detecting complex dynamics present in real-world time series. Decomposition-based forecasting methods improve the accuracy by isolating a series' components, but finding a suitable algorithm for each component remains an issue. This thesis addresses the limitations of current time series decomposition-based metalearning techniques, particularly those of the METAFORE framework, by introducing a more detailed, component-based strategy for the selection of algorithms. Using metalearning, the proposed system automatically determines whether to decompose a time series and recommends customized forecasting models for each resulting component. A systematic methodology and evaluation of performance on 5000 time series of monthly frequency from the M4 competition benchmarking dataset demonstrate the efficacy of this approach over conventional and single-model strategies. These findings highlight the utility of component-wise algorithm recommendations in improving forecast performance, which has significant implications for practical applications and future research in automated forecasting systems.

# United Nations Sustainable Development Goals

This Master's dissertation, while technical in its focus, directly contributes to several of the United Nations (UN) Sustainable Development Goals (SDGs). The development of an advanced, automated, and accurate forecasting framework provides a great tool that can help industries and organizations operate more efficiently and sustainably. By improving the ability to predict future demand, consumption, and other key variables, this work creates a direct path to better resource management, waste reduction, and more resilient infrastructure. This section analyses how the research presented in this thesis aligns with three key SDGs, which are presented below:

- **SDG 9 (Industry, Innovation, and Infrastructure)**

  This goal is focused on developing reliable infrastructure, promoting fair and environmentally friendly industrial growth, and supporting new ideas. It aims to build quality systems that support economic activity and people's well being. At the same time, it pushes for innovation to create better and cleaner industrial solutions for the future.

- **SDG 12 (Responsible Consumption and Production)**

  This goal is about changing the way we make and use things to be more sustainable. It pushes for greater efficiency so that we can use fewer natural resources. The main idea is to reduce waste and pollution in order to improve quality of life while protecting the environment.

- **SDG 13 (Climate Action)**

  This goal calls for immediate and united efforts to fight climate change. It has two main parts, with the first being to lower the gas emissions that cause global warming, and the second being to help communities better cope with the climate effects that are already happening. It also highlights the need to include climate planning in all national policies.

The Table 1 details the specific goals, contributions, and performance indicators that connect this dissertation to the global sustainability agenda.

Table 1: Contribution to the UN Sustainable Development Goals.

| SDG | Target | Contribution | Performance Indicators and Metrics |
|---|---|---|---|
| **9** | 9.4 | This framework creates an advanced forecasting process that allows industries to optimize operations. More accurate forecasts lead to a significant **increase in resource-use efficiency**, such as reducing excess raw materials and energy in manufacturing or improving electricity grid management. | **Indicator:** Improved operational efficiency.<br><br>**Metric:** The **reduction in forecasting error (e.g., Mean Absolute Scaled Error (MASE), Normalized Mean Absolute Error (NMAE))** compared to baselines. Lower error directly reduces the need for operational buffers (like safety stock), improving resource efficiency. |
| **12** | 12.2 | Improved forecasting enables the **efficient use of natural resources** by helping companies better align production with true demand. This prevents the unnecessary consumption of raw materials and energy. | **Indicator:** More efficient use of resources in production.<br><br>**Metric:** The **reduction in forecasting error (e.g., Root Mean Squared Error (RMSE))**. Lower error minimizes the need for buffer resources, contributing to more sustainable management. |
| | 12.5 | By accurately predicting demand, the framework directly supports the **reduction of waste generation**. It minimizes overproduction, which is a major source of wasted finished goods, materials, and energy. | **Indicator:** Reduced waste from overproduction.<br><br>**Metric:** The framework's **per-series win count**. Each win represents an instance of optimal planning, directly preventing waste that results from inaccurate forecasts. |
| **13** | 13.2 | The efficiency gains from better forecasting directly lead to a **lower carbon footprint** through reduced energy waste. This framework enables better climate planning and strategies, such as improving the integration of renewable energy into power grids by more accurately predicting supply and demand. | **Indicator:** Reduced energy-related emissions.<br><br>**Metric:** The framework's **improved forecast reliability (lower error standard deviation)**. Higher reliability reduces the need for carbon-intensive backup systems used to manage forecast uncertainty, enabling more efficient climate strategies. |

# Acknowledgments

This thesis represents the culmination of a long and challenging journey, one that has been as much about personal growth as it has been about academic achievement. The experiences I have lived and the people I have had the privilege of knowing throughout this time have shaped the person I am today and will undoubtedly influence the person I become tomorrow. This work would not have been possible without the support, guidance, and encouragement of many people, to whom I wish to express my deepest gratitude.

I would first like to express my sincerest thanks to my supervisors, Moisés and Carlos. Their guidance and mentorship were crucial in addressing the complexities of this project, and this thesis is as much a product of their wisdom and direction as it is of my own work. Beyond their professional support, I am grateful for their friendship, for the insightful conversations, and for the invaluable wisdom they shared so generously.

To my family, my mother, my father, my sister, my aunt, and my grandparents, I owe my deepest thanks. Your love and support have been my foundation. Thank you for always believing in me and for supporting my academic and personal journey in every way possible. I could not have done this without you.

A special and heartfelt thank you goes to my girlfriend, Catarina. Your love, your patience, and your constant presence have been my greatest support through every challenge and celebration. The moments we have shared have been invaluable, and you truly make all the difference in my life. Thank you for being you.

I also wish to thank my friends. Your presence, the good moments we have shared, and your friendship have been a constant source of strength and have helped me overcome the many difficulties of this journey. Your support has been invaluable.

To everyone mentioned here, and to the many others who have been a part of this journey, thank you. This achievement is not mine alone, but a reflection of the collective support, friendship, and love I have been so fortunate to share.

José Araújo

*"The fear of being different prevents most people from seeking new ways to solve their problems."*

Robert Kiyosaki, *Rich Dad Poor Dad*

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Time series forecasting is a critical aspect in diverse domains, such as finance, healthcare, and supply chain management, where accurate predictions are essential for strategic decision-making [1]. Industries like healthcare or supply chain management, for instance, rely on accurate forecasts to optimize resource allocation and anticipate demand for services, while financial institutions use forecasts to manage risk and make informed investment decisions [2].

Traditional forecasting methods, ranging from basic statistical techniques to complex machine learning models, often struggle to capture intricate patterns within the data [2]. Decomposition-based methods like Seasonal Trend Decomposition using Loess (STL) have found growing support by breaking time series into simpler components, facilitating better modeling of trends, seasonality, and residuals [3, 4, 5, 6, 7, 8]. However, the challenge lies in selecting the most appropriate forecasting algorithm for each component, as manually testing different models can be computationally expensive and time-consuming. Metalearning emerges as a solution, optimizing algorithm selection by leveraging knowledge from previous tasks, as demonstrated by METAFORE [3], a pioneering framework that enhances forecasting performance by using decomposition methods and tailoring model selection to time series characteristics. Through automating and optimizing the algorithm selection process, metalearning has the potential to revolutionize time series forecasting in various domains.

By addressing the limitations of traditional methods and leveraging the power of metalearning, it is possible to achieve further potential of time series forecasting, leading to more accurate predictions and better informed decisions.

Despite the successes of METAFORE in leveraging metalearning for algorithm selection, its current approach falls short by applying a seasonal naive method to the seasonality component and recommending a single model for the combined trend and residual components. This limitation overlooks the unique patterns inherent in each component, potentially reducing forecasting accuracy.

The motivation for this research stems from the opportunity to refine METAFORE's framework by introducing a more granular, component-specific algorithm selection process. This extension not only promises improved accuracy but also holds significant potential for real-world applications in sectors where precise forecasting is critical, such as financial markets, where more accurate forecasts of stock prices, exchange rates, and other financial indicators can help investors make informed decisions and reduce financial risks, medical prognosis, where improved forecasting of disease outbreaks, patient demand, and resource utilization can optimize healthcare resource allocation and enhance patient care, and supply chain management, where more precise forecasts of demand and supply can enhance the

inventory management, reduce costs, and prevent stockouts or overstocking. By addressing the current limitations and introducing a more sophisticated component-specific approach, it could be possible to achieve more robust and adaptable forecasting models, benefiting both researchers and practitioners.

The primary limitation of existing automatic model selection methods for time series forecasting lies in their treatment of the entire dataset as a single entity, without considering its decomposition into distinct components. This simplification increases the complexity of pattern identification, leading to potentially lower forecasting accuracy. While METAFORE addresses this issue by applying decomposition-based methods, it restricts algorithm selection to a seasonal naive model for the seasonality component and a single model for both trend and residual components. This approach neglects the distinct characteristics of each component, potentially undermining the accuracy of the forecasts.

The aim of this study is to use metalearning to build a component-specific algorithm recommendation system, expanding the capabilities of the current METAFORE framework for time series forecasting. This expansion attempts to take advantage of the utilization of these individual components to improve forecasting accuracy. Time series decomposition has shown to be efficient in decomposing complex data into more simple components. This work will address many important concerns to understand the effectiveness and efficiency of the suggested approach in order to assess the impact of this extension. To guide this investigation and structure our work, we pose the following research questions:

- **RQ1:** What is the advantage of leveraging the metalearner for choosing between individual models and decomposition-based methods for the formation of forecasting combinations?
- **RQ2:** What is the advantage of leveraging this approach in comparison with individual models?

The primary focus of RQ1 is to investigate the advantage of leveraging a metalearning approach for selecting between individual forecasting models and decomposition-based strategies in the formation of forecast combinations. It seeks to identify whether the metalearner can reliably and sensibly pick the best performing model from a pool of candidates, allowing for improved predictability in a substantial proportion of cases. Furthermore, the application of decomposition methods shall be expected to break down challenging time series into simpler components, presumably making them less challenging to predict, which can be anticipated to provide improved performance in the forecasting task. This question is particularly relevant to the problem setting, where the goal is to explore methods and techniques for constructing effective and computationally efficient forecasting pipelines. If decomposition proves to enhance model performance by reducing series complexity and if the metalearner can exploit this in selecting optimal models based on their performance, it directly supports the aim of achieving higher forecast quality through intelligent model combination strategies.

RQ2, by contrast, is interested in comparing the independent performance of the proposed framework based on metalearning against each of the individual baseline models encompassed within it. The objective is to assess whether the metalearner-based selection process results in better overall performance than any of the constituent forecasting models when used in isolation. As the metalearner aims to estimate the theoretical best model for a specific instance from a time series, achieving greater, if not equal, overall performance is expected compared to the best performing individual model. This is a critical question in determining the utility of the framework in practice, as being able to beat single-model baselines consistently would imply that the metalearner is able to leverage data-driven cues to make model choices, and thus justify its use in real-world forecasting tasks.

## 1.1 Document Structure

In addition to this introductory chapter, this thesis is organized into five additional chapters. Chapter 2 reviews other studies and explains the key ideas and current knowledge needed to understand this work. Chapter 3 describes the proposed solution, explaining its structure and how it works in two main steps, which are collecting metadata and then recommending forecasting models. Chapter 4 covers how the experiments were done, detailing the data used and the methods for evaluating performance. Chapter 5 presents the results from these experiments, providing answers to the research questions, and takes a closer look at specific parts of the system to see how they affect performance. Chapter 6 concludes the thesis by summarizing the main results, discussing the contributions and limitations of the research, and suggesting ideas for future work. Finally, an appendix is included at the end to provide extra details and figures.

# Chapter 2

# Literature Review

Following the PRISMA (Preferred Reporting Items for Systematic reviews and Meta-Analyses) methodology, the literature review phase includes a review of the discovered studies in order to assess and place current research in the selected field. In this stage, researchers evaluate the quality, content, conclusions, and approaches of the chosen literature, highlighting important concepts and patterns that show up in the publications. The extensive examination contributes to a better understanding of the subject at hand, by identifying gaps and topics for additional research in addition to offering an overview of the existing state of knowledge.

To ensure a comprehensive review but also guide and ease the whole process, the Parsifal [9] software was employed to manage the screening, selection, and eligibility processes, enhancing the rigor and transparency of this literature review. Before presenting the findings of this review, a brief background will be provided on the core concepts of time series forecasting, decomposition and metalearning.

## 2.1 Background

This section underlines some of the key concepts that will be central to this work which are time series analysis, decomposition techniques, and the use of metalearning for algorithm selection in forecasting.

### 2.1.1 Time Series Analysis

A time series is a sequence of data points collected and ordered at consistent time intervals, $Y_t = \{y_t\}_{t=1}^{T}$, where $T$ denotes the length of the series. Time series can be categorized as univariate or multivariate, which means observing only one and multiple variables respectively, all associated with the same phenomenon. These series are further distinguished by frequency, such as daily, monthly, or yearly observations, with higher frequencies involving finer intervals like hours or seconds.

The components of a time series are described as follows:

- **Trend ($T_t$):** A low-frequency component and long-term progression, representing the underlying direction of the series.
- **Seasonality ($S_t$):** Cyclical patterns that repeat at fixed intervals, such as yearly or quarterly fluctuations.

- **Residuals ($R_t$):** Random or irregular components capturing noise or deviations unexplained by $T_t$ or $S_t$.

These components can be combined additively or multiplicatively. For a time series $Y_t$, the additive and multiplicative formulations are, respectively, $Y_t = T_t + S_t + R_t$ and $Y_t = T_t * S_t * R_t$.

Decomposing a time series involves reversing this relationship to estimate $T_t$, $S_t$ and $R_t$:

$$f(x) = (T_t, S_t, R_t) \tag{2.1}$$

where $f$ represents a decomposition function.

### 2.1.2 Multiple Seasonal-Trend decomposition using Loess (MSTL)

Many different techniques exist in regard to time series decomposition, but for reasons explained later, the MSTL decomposition technique will be adopted.

Multi-Seasonal-Trend Decomposition using Loess (MSTL) generalizes STL to handle the case of multiple overlapping seasonal patterns. In most real-world applications, time series data often contains more than one periodic behavior. For instance, electricity demand data can have both daily fluctuations depending on human activities and yearly patterns depending on seasonal weather changes. MSTL explicitly addresses these complexities by decomposing the time series into multiple seasonal components in addition to trend and residual elements.

Mathematically, MSTL models a time series $Y_t$ as the addition or product of its constituent series:

$$Y_t = T_t + \sum_{i=1}^{k} S_t^{(i)} + R_t \quad \text{(additive form)} \tag{2.2}$$

or:

$$Y_t = T_t \cdot \prod_{i=1}^{k} S_t^{(i)} \cdot R_t \quad \text{(multiplicative form)} \tag{2.3}$$

where $T_t$ is the overall trend, $S_t^{(i)}$ represents the $i$-th seasonal component, $R_t$ accounts for residual noise, and $k$ is the total number of seasonal patterns distinguished within this data. Similar to STL, MSTL applies Loess smoothing, a non-parametric regression technique, iteratively to extract components. However, MSTL differs in that it includes mechanisms that detect and handle multiple periodicities simultaneously. It detects the periodicities corresponding, for example, to weekly, monthly, or even hourly cycles by spectral analysis or autocorrelation methods. Each seasonal component is then extracted and smoothed individually. It should be robust regarding outliers and able to adapt to complex seasonal trend patterns evolving over time. Thus, MSTL enables modeling of complex temporal patterns not captured by simple decomposition techniques. Besides, thanks to the automatic detection of seasonality, the use of MSTL is appropriate when it comes to large-scale automated forecasting systems with impossible manual identification of periodicity. This can, in turn, significantly enhance the accuracy of those forecasting methods based on decomposition, thanks to the capture and modeling of multiple seasonalities. Combining MSTL with the latest forecasting frameworks, including metalearning-based approaches, shows great promise in handling high complexities for most of today's time series.

### 2.1.3 Time Series Forecasting

Time series forecasting aims to predict future values $Y_{T+h}$ for a time series $Y_t = \{y_t\}_{t=1}^{T}$, where $h$ represents the forecast horizon. The task can be expressed as finding a function $f$:

$$f(t) = \hat{y}_{t+h} \approx y_{t+h}, \quad \forall t+h > T, \tag{2.4}$$

where $\hat{y}_{t+h}$ denotes the forecast value. A forecasting model $f$ is a mapping from the input historical observations to predicted outputs:

$$A : \{y_t\}_{t=1}^{T} \rightarrow f \tag{2.5}$$

Forecasting models typically assume the existence of a probability measure $\mu(y_{T+h}|\{y_t\}_{t=1}^{T})$. Single step forecasting $(h = 1)$ predicts one period ahead, while multi-step forecasting $(h > 1)$ predicts multiple future periods.

### 2.1.4 Metalearning for Algorithm Selection

Algorithm selection is pivotal for time series forecasting due to the diversity of available models and the complexity of time series data. The Algorithm Selection Problem (ASP) can be mathematically represented as finding an optimal algorithm $L$ from a portfolio $A$, applied to datasets $D$ within a task space $P$:

$$S : P \rightarrow A \tag{2.6}$$

where $S$ maps tasks $d \in P$ to algorithms $L \in A$, optimising the performance measure $p : A \times P \rightarrow R^n$. Exploring all possible algorithm-dataset pairs $S$ is computationally expensive. Metalearning mitigates this by leveraging knowledge from previously solved tasks to predict effective algorithms for new tasks. Metalearning relies on meta-features, extracted as functions $f : D \rightarrow R^k$ , that encode properties of time series datasets. For instance, meta-features might include autocorrelation coefficients, statistics of trends or seasonality, and distribution measures such as kurtosis or skewness.

The relationship between meta-features and model performance is learnt via meta-targets, often defined as rankings, predicted performance, or specific algorithms. This process is depicted in the following pipeline:

1. **Dataset characterisation:** Extract meta-features $f(d_i)$.
2. **Algorithm evaluation:** Estimate $p(L, d_i)$, the performance of algorithms $L$ on datasets $d_i$.
3. **Metadata construction:** Combine meta-features and performance metrics.
4. **Meta-model training:** Train a model to map meta-features to algorithm recommendations.

Given a new time series $Y$, metalearning recommends a model $L$ for forecasting its components. By combining meta-knowledge with decomposition techniques, this approach addresses the critical challenge of algorithm selection while reducing reliance on domain expertise.

## 2.2 State of the Art

This section provides an overview of the current landscape of time series forecasting. It begins by discussing the established techniques that are foundational to the field. The discussion then covers the various decomposition methods employed in modern approaches. Finally, it examines the different metalearning strategies that have been adopted to enhance forecasting accuracy.

### 2.2.1 Decomposition methods

Decomposition methods in time series analysis are applied to break down a complicated time series into its different components, typically trend, seasonality, and residual or irregular components. This separation enables one to understand the underlying patterns and dynamics driving the time series. Valuable insights from such data would be much easier to perceive in long-term behaviour, periodic fluctuations, and random noise. They enable superior predictions, the identification of more interesting anomalies, and provide insight into the data-generating process.

#### 2.2.1.1 Basic techniques

Basic methods are the ones in which time series are decomposed into trend, seasonality, and remainder. They can be straightforward to perform and understand, hence providing a very basic sense about the patterns of such tasks as simple forecasting. They are not able to capture complex dynamics.

- **Seasonal Trend decomposition using Loess (STL):** A filtering method that decomposes a time series into three components: seasonal, trend, and remainder. It uses Loess (locally estimated scatterplot smoothing) to smooth the data and identify these patterns. This helps isolate different aspects of the time series for analysis or forecasting [3, 4, 5, 6, 7, 8].
- **Empirical Mode Decomposition (EMD):** A data-driven method that decomposes a time series into a set of Intrinsic Mode Functions (IMFs). These IMFs represent the different oscillatory modes embedded in the data, from high frequency to low frequency. Think of it like sifting the signal to separate out different scales of oscillations [10, 11, 12, 13, 14, 15].
- **Wavelet Transform (WT):** A powerful tool for analyzing time series data at different frequencies and resolutions. It uses wavelet functions to decompose the signal, providing information about both time and frequency simultaneously. This is like using a magnifying glass with adjustable zoom to examine the signal at different scales [10, 12, 13, 14, 15, 16].

#### 2.2.1.2 Advanced and Ensemble Techniques

Advanced techniques are capable of digesting difficult patterns and shifts in seasonality. Ensemble techniques simply combine outputs from multiple methods. Such techniques come in handy when faced with complex seasonality, nonlinear trends, or irregular fluctuations.

- **Time-varying filter empirical mode decomposition:** This is a variation of EMD that uses time-varying filters to improve the decomposition accuracy, especially for non-stationary signals where the frequency content changes over time [17].

- **Improved Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (ICEEMDAN):** An improved version of EMD that addresses some of its limitations by adding adaptive noise during the decomposition process. This helps to obtain more accurate and reliable IMFs [13, 18].
- **Empirical Wavelet Transform (EWT):** A more adaptive version of the wavelet transform that derives the basis functions (wavelets) directly from the data itself, making it more suitable for non-stationary signals [12].
- **Ensemble Empirical Mode Decomposition (EEMD):** An enhancement of EMD that addresses the "mode mixing" issue by adding white noise to the signal and averaging the results of multiple decompositions. This leads to more stable and meaningful IMFs [10, 11, 12, 14].
- **Fast Ensemble Empirical Mode Decomposition (FEEMD):** A computationally efficient version of EEMD that reduces the processing time while maintaining the benefits of ensemble averaging [10].

#### 2.2.1.3 Variational Mode Decomposition Techniques

Variational Mode Decomposition (VMD) methods decompose time series into intrinsic mode functions, which is extremely useful for catching nonlinear and nonstationary features. Compared with some older methods, such as EMD, VMD exhibits better performance in handling noise and mode separation.

- **Variational Mode Decomposition (VMD):** A non-recursive decomposition method that decomposes a signal into a set of band-limited modes. It uses a variational framework to estimate the center frequency and bandwidth of each mode, making it more robust to noise and sampling variability [10, 19].
- **Multivariate Variational Mode Decomposition (MVMD):** An extension of VMD that can handle multivariate time series data, allowing for the analysis of relationships between different channels or variables [11].
- **Optimized Variational Mode Decomposition (OVMD):** Refers to various techniques that aim to optimize the parameters of VMD (like the number of modes) to improve the decomposition performance for specific applications [20].

#### 2.2.1.4 Other Important Techniques

- **Singular Spectrum Analysis (SSA):** A powerful method for analyzing time series based on eigenvalue decomposition of the lag-covariance matrix. It can extract trends, oscillations, and noise from the data and is particularly useful for identifying periodicities [10, 21, 22].
- **Wavelet Decomposition (WD):** A general term referring to the process of decomposing a signal using wavelet transforms. It often involves breaking down the signal into different frequency sub-bands (like low-pass and high-pass components) [21].

- **Wavelet Soft Threshold Denoising (WSTD):** A denoising technique that uses wavelet transforms to identify and remove noise from a signal. It involves thresholding the wavelet coefficients to suppress those associated with noise [10].
- **Phase Space Reconstruction (PSR):** A technique used to reconstruct the dynamics of a system from a single time series. It involves creating a multi-dimensional space from lagged versions of the original signal, which can reveal hidden patterns and attractors [10].
- **Maximal Overlap Discrete Wavelet Transform (MODWT):** A variation of the discrete wavelet transform that provides a redundant representation of the signal, improving time-frequency analysis and reducing boundary effects [16].

### 2.2.2 Metalearning

Metalearning for model selection can be defined as the ability of knowledge learnt from past experiences to automatically choose the best forecasting model for a new time series dataset. Other than manual selection or trial-and-error approaches, metalearning applies algorithms that learn from the performances of different models on various datasets. It extracts meta-features (characteristics of the datasets) and learns how those are related to the best models. The knowledge obtained in this way can then be used to suggest the most suitable model for some new, previously unseen dataset. In fact, it becomes quite useful when one has to deal with thousands of datasets or complex combinations of forecasts, on which manual model selection is just impractical.

#### 2.2.2.1 One-model metalearner

One-model metalearners rely on a single model, which learns the relationship from metafeatures of datasets, along with the performance of various models. This type of metalearners is simple and efficient, hence, it can be preferred in the case of an exploratory study or in the presence of limited computation resources.

**Random Forest (RF)**

The Random Forest [23] is a family of machine learning algorithms that performs so as to realise ensemble learning, wherein, instead of relying on just one model for performance, an elicited result comes out of many models. All the trees, or tree-like structures, that are formed by partitioning the data across a variety of features combine through an output to make the whole of a Random Forest. By combining the predictions of many trees, it reduces overfitting (when a model performs well on training data but badly on new data) and produces robust and generalised results. Due to these points, Random Forests can turn out to be strong metalearners:

- Pattern capturing: Each tree in the forest learns different relationships in the data. This diversity is important for metalearning because the metalearner has to generalise across a variety of tasks, each with possibly different underlying structures.
- Handling high dimensionality: Random Forests are effective at handling high-dimensional datasets, which is common in metalearning tasks because the metalearner receives metadata—information about the tasks.

- Non-linear relationships: It can learn non-linear relationships between metadata and the performance of various learning algorithms. This is important since the relation between task characteristics and the best algorithm for a particular task may be quite intricate.

Among the selected documents in the Literature Review phase, seven of them [3, 5, 7, 8, 24, 25, 26] made use of Random Forest model as a metalearner.

**Artificial Neural Network (ANN)**

The Artificial Neural Network (ANN) is a computing system inspired by the human brain, comprising a large number of interconnected nodes or "neurons" in layers. Such layers include, but are not limited to, an input layer for receiving initial data, hidden layers performing intermediate computation in pattern and feature extraction, and an output layer with a final result or prediction. In any given neuron connection, each is related to some weighted input that indicates the strength of a signal continuing further through the network. During the training of a network, this weight factor is continuously varied to teach the network various relationships of data in order to tune its results.

ANNs are particularly suited to applications as metalearners, supported by the following inherent strengths:

- Representation learning: ANNs are good at representation learning, the process of automatically discovering features of interest from the input, to then be used on some downstream task. So in metalearning, effectively capturing complex relationships between task characteristics - metadata - and algorithm performance provides the best results.
- Function approximation: ANN is a universal function approximator. In principle, it could learn any continuous function that describes how properties of tasks may be mapped to the optimum learning strategy.
- Adaptability: ANNs can adapt themselves to new tasks and other data distributions, thanks to their capability for learning and generalization. This is one of the most important conditions for metalearners, who have to face a wide diversity of tasks.
- Handling High-Dimensionality: Like Random Forests, ANN can handle high-dimensional metadata common in metalearning.

Among the selected documents in the Literature Review phase, three of them [24, 25, 26] made use of Artificial Neural Networks as a metalearner.

**Multi-output regression-based metalearning method**

Multi-output regression-based metalearning, given an overall task, the goal is to predict multiple outputs in one go. Instead of just predicting one value, such as the accuracy of a single algorithm, the metalearner will predict several performance metrics or related values.

This multi-output regression metalearner will be most fitting for the metalearner task due to:

- Holistic performance evaluation: The ability to predict multiple outputs provides a fuller picture of how different algorithms perform on a given task, enabling better decision-making when selecting among algorithms.

- Capturing relationships between outputs: The metalearner can identify correlations and dependencies between outputs. For instance, it might discover that higher accuracy is often associated with longer training times for time series models.
- Efficiency: Training a single model for multiple outputs is more efficient than training separate models for each output.

Among the selected documents in the Literature Review phase, one of them [27] made use of a multi-output regression-based meta-learning method as a metalearner.

#### 2.2.2.2 Multi-model/Hybrid metalearner

Multi-model or hybrid metalearners combine the strengths of multiple machine learning models to increase robustness and accuracy in metalearning. Through capturing greater variability in patterns and relationships in meta-data, hybrid methods are designed so that it enables the assignment of different models for a particular task in the metalearning process. This will further allow customisation in the nature of the data or according to demand from the task of forecasting.

**Energy Meta Learning System (EMLS)**

EMLS uses the strengths of various metalearners to enhance the overall performance and robustness of the metalearning system. It makes use of an ensemble of RF and ANN models as metalearners. This will help overcome the limitations of the different metalearners and thus provide more accurate and reliable recommendations for forecasting models.

Among the documents selected during the Literature Review phase, one of them [28] used this hybrid metalearner approach.

**K-Nearest Neighbours (KNN), Decision Tree (DT), Random Forest (RF), Multi-Layer Perceptron (MLP)**

The technique involves an ensemble of several metalearners, such as K-Nearest Neighbours, Decision Tree, Random Forest, and Multi-Layer Perceptron. A key feature is the ease of working with an arbitrary variety of metalearners, each one having various strengths and weaknesses. In such a way, this system selects the best performance for any metalearner regarding any particular task to ensure that overall accuracy could be further improved.

Among the selected documents in the Literature Review phase, this hybrid metalearner approach was used by the author [29].

#### 2.2.2.3 Meta-features

The meta-features extracted from a time series dataset transform raw time series into a rich set of representative features. These features then act as informative metadata for metalearners. These features can be utilized by metalearners to learn relationships between time series characteristics and the performance of different forecasting models. In such a way, it can recommend the best forecasting model for a new, unseen time series, considering its extracted features.

- **Tsfresh**

  Tsfresh is a Python package that can automatically compute a large variety of features from time series data, including measures of trend, seasonality, autocorrelation, and so on.

One of the documents from the Literature Review phase [3] used this package.

- **PyMFE (Python Meta-Feature Extractor)**

  PyMFE is a Python package for meta-feature extraction, providing one-stop access to extracting and computing meta-features of time series. Being such, it provides a common, wide meta-feature ground that covers a wide collection of time series features with specific statistics, information-theoretic measures, and model-dependent estimations. It equally grants fine control over all of the feature extraction logic in terms of choosing the only relevant groups of meta-features needed, personalising statistics computed, and hyperparameter tuning at each measure.

  This package was used by one of the documents produced during the Literature Review phase [27].

- **TSFeatures**

  TSFeatures [30] is an package used to perform time series feature extraction. Its function is to systematically calculate a wide range of characteristics from a time series. These calculated features include general properties such as the strength of trend and seasonality, autocorrelation values, spectral entropy, and statistical test results like the ARCH.LM statistic.

  In the paper [26] it is made use of this package to extract 42 meta-features.

- **Custom Approaches**

  Some of the techniques used to extract meta-features were not related to any library or specific method, as they were custom-made. Some of those techniques will be covered in this part.

  - In the research done in the paper [24], the authors used a proprietary method of defining meta-features. They defined 128 different features and grouped them into seven categories: simple, statistical, information-theoretic (IT), time series (TS), DSTMF, all without DSTMF, and all. The very broad definition of meta-features allowed the authors to represent more characteristics of datasets for their metalearning model.
  - In the research made in the paper [7], several time series characteristics were calculated for each time series. Those characteristics include information about the time series itself, statistical measures, characteristics proposed by Wang et al., characteristics proposed by Lemke and Gabrys, and characteristics proposed in this work.
  - In the research made in the paper [31], 10 features are selected to describe the time series: Shannon entropy, Spectral entropy, Singular value decomposition entropy, Fisher information, Kurtosis, Skewness, Gaussianity of the differences, The Hurst Exponent, Detrended fluctuation analysis, and Stationarity.
  - In the research made in the paper [29], a custom approach is employed for defining meta-features, utilizing basic statistics, statistical tests, autocorrelation, and frequency domain analysis.
  - In the research made in the paper [25], a custom approach is employed for defining meta-features, including "time" meta-features, such as sampling frequency, stationarity, seasonality components, and "statistical" meta-features, including kurtosis and skewness.

– In the research made in the paper [32], the meta-features used are the model parameters.
– In another attempt [27], the authors take a different line of thinking by introducing a set of new meta-features to take into consideration the relationships presented in multivariate time series datasets together with the domain information they exhibit. Altogether, 155 such meta-features have been proposed.

### 2.2.3 Forecasting methods

Time series forecasting methods are those methods that have the ability to make future values in a time series, depending upon the pattern and trend followed previously by the time series data. These methods, if accurately used, find real applications in finance, economics, engineering, and environmental sciences for informed decisions and resource allocations. The selection of the method to be used would appropriately be made based on characteristics that define the time series data, desired horizon of forecasting, and the level of accuracy desired.

#### 2.2.3.1 Traditional / Statistical Methods

Traditional/statistical methods analyse patterns in historical data relying on statistical principles. They are interpretable, well understood, and extremely efficient for stable patterns. However, they may not work as well with complex or nonlinear data. In the Figure 2.1, it is possible to observe the distribution of traditional models leveraged across papers.

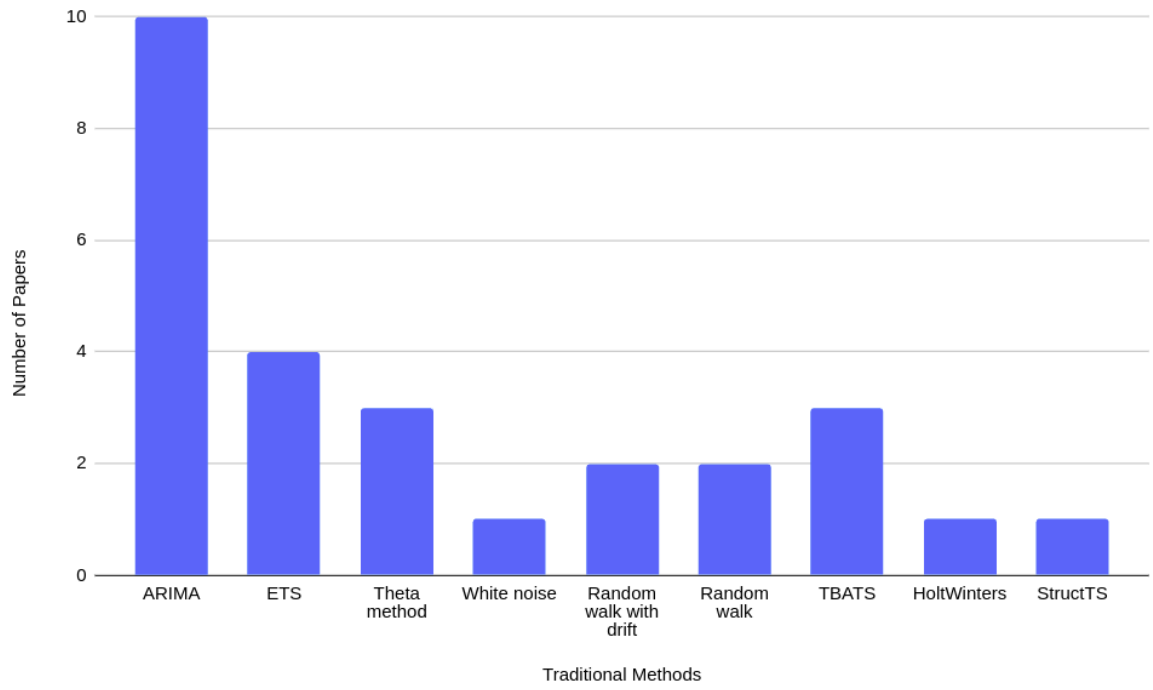



Figure 2.1: Distribution of used traditional methods.

**Autoregressive Integrated Moving Average (ARIMA)**

The ARIMA model, which stands for Autoregressive Integrated Moving Average [33], is one of the most popular and powerful statistical methods in time series forecasting. It incorporates three parts:

autoregression (AR), which models the relationship between an observation and a certain number of lagged observations, integrated (I), accounting for the degree of differencing, which may be necessary to make the time series stationary, and moving average (MA), which models the dependency between an observation and a residual error from a moving average model applied to the lagged observations. The ARIMA models can be versatile toward many time series patterns and hence are applicable in such data, which have been subjected to trends, seasonal variations, or other forms of temporal dependencies. Their strength in capturing both short and long term dependencies in data quite often results in accurate and reliable predictions, hence, it is valuable in many domains.

Ten papers [5, 10, 11, 13, 16, 29, 34, 35, 36, 37] leveraged this model for time series forecasting.

**Exponential Smoothing (ETS)**

Exponential Smoothing (ETS) [38, 39] finds its application in univariate time series data prediction, basically used for trend and seasonality predictions. It assigns weights similar to that of exponential smoothing, giving high importance to the recent observations. This model works quite well for shifts in the mean of the time series over time. ETS is a family of models that account for different types of variations in trends and seasonality, including Simple Exponential Smoothing, Holt's Linear Trend, and Holt-Winters' Seasonal method. The specific ETS model chosen depends on the characteristics of the time series data. ETS has become a staple in many different forecasting applications due to its simplicity, efficiency, and ability to adapt to changes in the data.

Four papers [5, 29, 34, 37] made use of this forecasting model.

**Theta method**

The Theta model, which is also known as a simple exponential smoothing model with drift, is a method of forecasting for time series data that exhibit a trend. This approach is based on the concept of giving exponentially weighted weight to the previous observations while making a forecast for the future, with an added drift term to account for the trend. The Theta model is particularly useful when one is dealing with data exhibiting a consistent upward or downward trend. Some of its advantages are simplicity, easiness of implementation, and adaptability to changes in trend. However, it is not suitable for a situation where the data has an intricate seasonal pattern or when it is highly volatile.

Three papers [34, 35, 36] leveraged this model for time series forecasting purposes.

**White noise**

The White Noise model is one of the most simple models, and it assumes each data point is a random and independent draw from some normal distribution with zero mean and constant variance. It basically means that the series doesn't have a pattern to follow and any fluctuation noted is just random. Now, it might sound dumb to use such a simple model, but the White Noise model is a necessary benchmark in which other, more complex methods are checked against its performance in a forecast. If it can't outcompete the White Noise benchmark, then it simply fails to pick up any meaningful trends in the data.

One paper [34] made use of the White Noise model for time series forecasting purposes.

**Random walk**

The Random Walk model is a simple but effective time-series forecasting method that assumes the future value of a series is equal to the current value plus a random error term. In other words, the optimal forecast of the series one period hence is its current level plus some random error. In this respect, it assumes that the series will continue along its present course with some randomness. This

model often serves as a benchmark for evaluating other, more sophisticated forecasting methods. If a sophisticated model cannot outperform a Random Walk model, it means that the model is not capturing any meaningful patterns in the data. Amazingly accurate, this simplistic model of the Random Walk can be for some forms of time series data-those kinds of data that show a lot of randomness or are naturally unpredictable.

Two papers [34, 36] make use of the Random Walk model for time series forecasting purposes.

**Trigonometric, Box-Cox transform, ARMA errors, Trend, and Seasonal components (TBATS)**

TBATS, which stands for Trigonometric, Box-Cox transform, ARMA errors, Trend, and Seasonal components [40], is a forecasting technique that was designed to handle even the most challenging time series data with multiple seasonal patterns. Unlike the traditional methods, which struggle to deal with complex seasonality, TBATS successfully models and forecasts series displaying weekly, monthly, and yearly seasonal cycles and will, therefore, be a great tool in many specific domains such as retail, tourism, and energy studies. Its flexibility and being able to grasp intricate seasonal patterns often ensure more accurate predictions compared to simpler alternatives. Its complexity also carries the implication that estimation in TBATS models is usually more computationally burdensome.

Three papers [5, 8, 36] made use of TBATS model for time series forecasting purposes.

**Holt Winters**

The Holt-Winters model, also known as triple exponential smoothing, is a powerful forecasting technique used for time series data that exhibits both trend and seasonality. It extends simple exponential smoothing by adding components to capture such patterns. The model contains three smoothing equations: one for the level, one for the trend, and one for the seasonality. This allows it to accommodate the changes in the series and make accurate forecasts when the underlying patterns are changing. Benefits of the Holt-Winters model are that it handles complex time series data, is relatively simple compared with more advanced methods like ARIMA, and effectively captures both trend and seasonal variations.

One paper [8] resorted to the Holt Winters model for time series forecasting purposes.

**StructTS**

StructTS is a powerful statistical modelling methodology which, besides being able to analyse, could also be used for time series data predictions driven by complex underlying patterns. It decomposes the time series into its main components: trend, seasonality, cycle, and irregular. This gives further details on the driving forces of data and allows for better forecasting. StructTS models are particularly useful when one is dealing with data that contains long-term trends, cyclical fluctuations, and seasonal patterns. They also handle outliers and missing values rather well.

One paper [8] resorted to StructTS for time series forecasting purposes.

#### 2.2.3.2 Machine Learning Methods

Machine learning methods are techniques that learn from data for the purpose of prediction. They are able to capture complex relationships and often adapt to changing patterns, thus being suitable for tough forecasting tasks. However, they can be less interpretable than traditional/statistical forecasting methods and may require careful tuning. In the Figure 2.2, it is possible to observe the distribution of Machine Learning models leveraged across papers, with special emphasis for the most used ones.

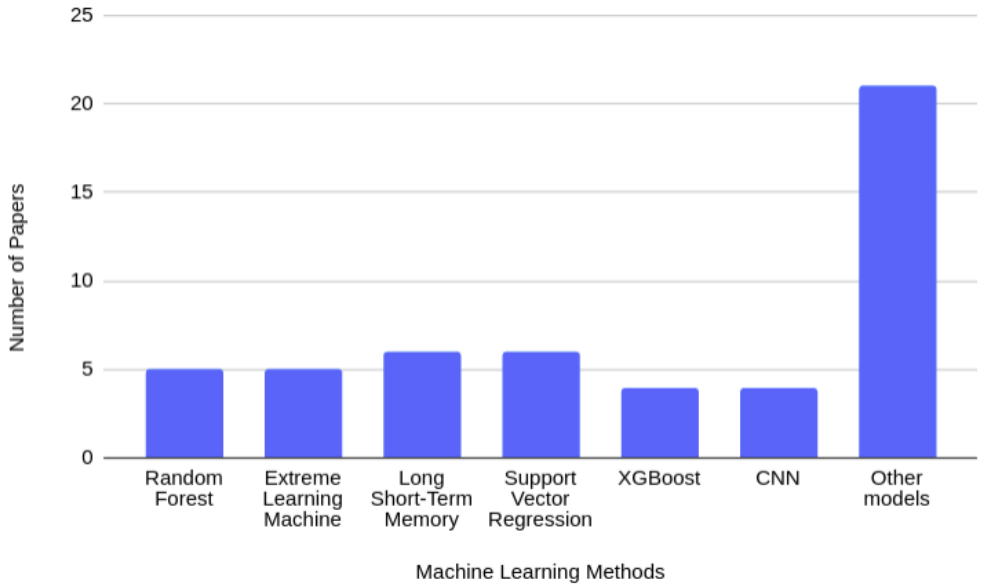



Figure 2.2: Distribution of used Machine Learning methods.

**Random Forest (RF)**

Random Forest is a well-known and established model, whose definition and advantages have been covered in the section above in the document regarding the metalearner methods.

Five papers [3, 7, 26, 37, 41] have used Random Forest for time series forecasting purposes.

**Extreme Learning Machine (ELM)**

The Extreme Learning Machine (ELM) is a feed-forward neural network with an extremely fast learning speed with a strong generalization capability. Unlike other neural networks, with iterative weight tuning, in ELM, the input layer weights are randomly initiated, while the weights of the output layer are determined analytically. None of these use backpropagation hence reduce training time. ELM has been applied to many fields of machine learning, including the area of time series forecasting. It has quite a number of advantages: it is efficient, simple, and is able to handle nonlinear data patterns. However, random initialization of input weights can cause a variation in performance.

Five papers [11, 13, 14, 20, 21] resorted to Extreme Learning Machine for time series forecasting purposes.

**Long Short-Term Memory (LSTM)**

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) designed to perceive the long-term dependencies of sequential data. Because of this, it is appropriate for time series forecasting. Unlike other RNNs, which suffer from the vanishing gradient problem, LSTMs can process information with the special cell and gates. These gates further help in the partial retaining or forgetting of information, hence allowing LSTMs to learn both short and long-term dependencies. They work very well in time series prediction problems where there is a complex temporal pattern, like financial data, weather forecasting, and traffic flow.

Six papers [11, 12, 14, 15, 21, 37] resorted to Long Short-Term Memory model for time series forecasting purposes.

**Support Vector Regression (SVR)**

Support Vector Regression (SVR) is a kind of machine learning used for the estimation of continuous variables. Actually, SVR is very effective in time series prediction since it can afford to handle nonlinear relationships and high dimensionality. The method will find the best-fitting line or, in three dimensions, the best-fitting hyperplane which maximizes the margin between the line and the examples, allowing a few errors to occur up to a certain threshold. That avoids overfitting and promotes better generalization. Because of its robustness, it is resistant to outliers and can grasp nonlinear trends, being well suited for time series analysis.

Six papers [3, 5, 6, 7, 21, 37] resorted to Support Vector Regression model for time series forecasting purposes.

**Extreme Gradient Boosting (XGBoost)**

XGBoost (Extreme Gradient Boosting) is a highly efficient and used machine learning algorithm for solving different classification and regression tasks. Recently, it has gained special popularity in solving time series forecasting tasks because it can capture complex patterns and give state-of-the-art performance. XGBoost, or the gradient boosting approach, works by iteratively training a set of decision trees where each corrects the errors of others. Efficient even more due to its regularization feature, parallelization, and effective handling of missing values for high performance and scalability.

Four papers [7, 18, 26, 37] made use of XGBoost for time series forecasting purposes.

**Convolutional Neural Network (CNN)**

CNN (Convolutional Neural Network) is a deep learning architecture originally designed for image recognition but has proved helpful in time series forecasting. They might automatically extract features from sequential data through convolutional filters. These filters scan the time series to capture local patterns and dependencies. Further, a fully connected layer provides the prediction based on the feature extracted. CNN models can operate directly with raw time series data with limited preprocessing, learning features automatically, and capturing temporal dependencies at multiple scales. Still, they are potentially computationally expensive, especially for long-term time series.

Four papers [5, 19, 34, 37] made use of CNN models for time series forecasting purposes.

### 2.2.4 Evaluation metrics

Performance metrics are essential tools for evaluating the accuracy and effectiveness of forecasting models, providing insights into how well a model's predictions align with the actual observed values. In the Figure 2.3, it is possible to observe the distribution of performance metrics leveraged across papers, with special emphasis for the most used ones.

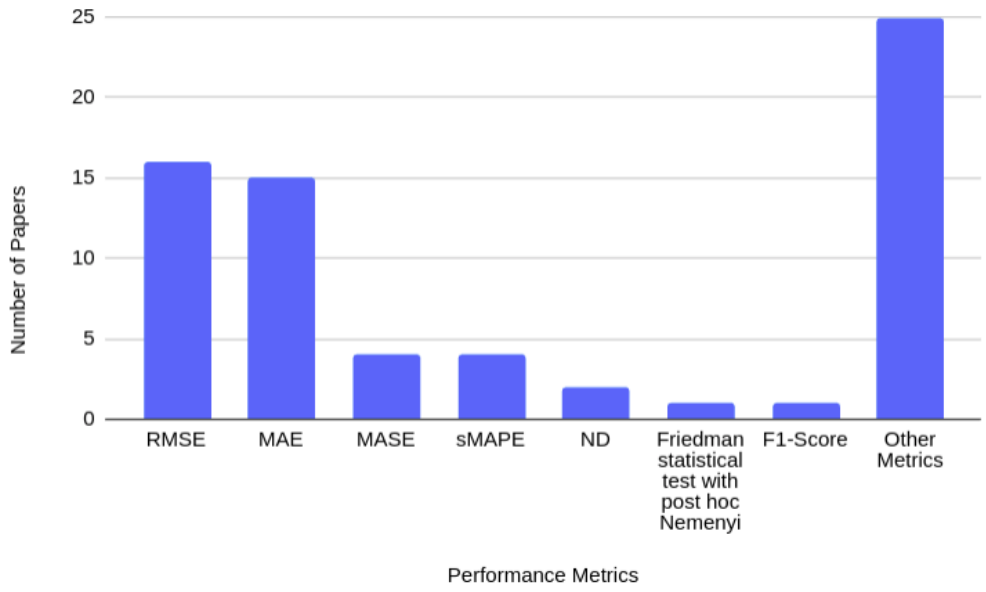



Figure 2.3: Distribution of used Performance Metrics.

**Root Mean Squared Error (RMSE)**

Root Mean Squared Error (RMSE) measures the square root of the average of the square of the differences between forecasted and actual values. Because it gives a greater weight to larger errors, it is sensitive to outliers. The lower the RMSE, the better the accuracy, having 0 as the perfect fit.

Several papers [5, 10, 11, 13, 16, 17, 24, 34, 35, 36, 37, 42, 43, 44, 45, 46] made use of this performance metric.

**Mean Absolute Error (MAE)**

Mean Absolute Error (MAE) calculates the average of the absolute differences between the predictions and the actual values. As opposed to the RMSE, MAE treats every error equally there. Therefore, MAE could give a better measure for overall accuracy. A lower MAE is better, having 0 as the perfect fit.

Several papers [5, 10, 11, 13, 16, 17, 24, 34, 35, 36, 37, 42, 44, 45, 46] made use of this performance metric.

**Mean Absolute Scaled Error (MASE)**

Mean Absolute Scaled Error (MASE) is a relative measure of performance that generally will be benchmarking against the model accuracy, typically a random walk or seasonal naive model. It scales the MAE by the MAE of the naive forecast. The resulting measure is scaled and comparable across different data sets and models. If less than 1, it is comfortably outperforming the naive forecast.

Four papers [29, 35, 37, 42] made use of this performance metric.

**Symmetric Mean Absolute Percentage Error (sMAPE)**

Symmetric Mean Absolute Percentage Error (sMAPE) computes symmetric Mean Absolute Percentage Error between forecasted and actual value. This is a measure that gives the measure of accuracy as a percent, hence, more interpretable and comparable across data sets. Lower values of sMAPE are related to better accuracy, while 0 corresponds to a perfect fit.

Four papers [17, 29, 34, 37] made use of this performance metric.

**Normalized Deviation (ND)**

Normalized Deviation (ND) uses the average of the actual value to calculate a normalized average absolute difference between forecast and actual values. It is a dimensionless accuracy measure and hence applicable for comparisons across different data sets having different scales. The lower the ND, the better the accuracy, having 0 as the perfect fit.

Two papers [37, 43] made use of this performance metric.

**Friedman statistical test with post hoc Nemenyi**

Friedman statistical test with post hoc Nemenyi is a non-parametric statistical test used to compare the performance of multiple forecasting models on multiple datasets. This test ranks the models on each dataset and then compares the average rank in trying to find out if their performances are significantly different or not. The post hoc Nemenyi test was conducted to find which pairs of models differ significantly.

One paper [3] made use of this performance metric.

**F1-Score**

F1-score is a measure that accounts for how well the model is performing, considering both precision and recall. Though it is for classification tasks, it is also extended to time series forecasting problems in terms of checking the accuracy of forecasting regarding specific events or patterns. F1 score is the harmonic mean of precision and recall, and it gives a symmetric measure of the latter two. A higher F1 score implies a better accuracy, with the best possible precision and recall having the value of 1.

One paper [27] made use of this performance metric.

### 2.2.5 Further works

There are 15 documents that state further development would be good, 0 documents that state that there should not be developed further works, and 26 documents that don't mention further works.

**Limitations in Current Metalearning Frameworks:** Existing frameworks lack versatility and automation. While Jati et al. (2013) [47] identifies the need for automated ML pipelines (e.g., preprocessing and hyperparameter tuning), Baratsas et al. (2024) [5] highlights insufficient coverage of time series characteristics.

**Advancements in Metalearning Models:** Current approaches fail to capture nonlinear meta-feature relationships. Bauer et al. (2020) [7] demonstrates the absence of deep meta-learning models, and Zhang et al. (2020) [25] shows limited algorithm diversity.

**Meta-Feature Engineering Challenges:** Bahrpeyma et al. (2024) [31] and Zhang et al. (2020) [25] reveal two critical issues, which are oversimplified meta-features and lack of domain-specific adaptations.

**Diversification of Forecasting Techniques:** Three studies [11, 14, 20] concur that reliance on single decomposition methods (e.g., EMD) limits performance.

**Diversification of Forecasting Techniques:** Three studies, Hao et al. (2024), Lin et al. (2024) and Wang et al. (2023) [11, 14, 20], concur that reliance on single decomposition methods (e.g., EMD) limits performance.

**Deep Learning Architectures:** Li et al. (2024) [15], Gao et al. (2022) [18], and Maya et al. (2021) [48] identify three shortcomings: rigid architectures, suboptimal preprocessing, and inadequate data augmentation.

**Metalearning Algorithm Selection:** Shahoud et al. (2021) [28], Taghiyeh et al. (2020) [41], and Cerqueira et al. (2022) [49] show inconsistent algorithm-preprocessing combinations degrade performance.

**Ensemble Methods:** Sayed et al. (2024) [26] criticizes static ensemble weighting for ignoring temporal dependencies.

# Chapter 3

# Solution

This chapter covers the architecture of the proposed solution, which was developed to address the research gaps identified in the previous Chapter 2. The goal is to create an automated and adaptive forecasting system that is able to select the best forecasting strategy for any given time series. The following sections will now explain each of these components in better detail.

## 3.1 Approach

In this section, the approach to be followed will be covered, which builds upon the foundational concepts and techniques explored in the background phase. The approach can be divided into two distinct phases: the metadata collection step and, afterwards, the models recommendation step, which will be covered in this section.

### 3.1.1 Metadata collection step

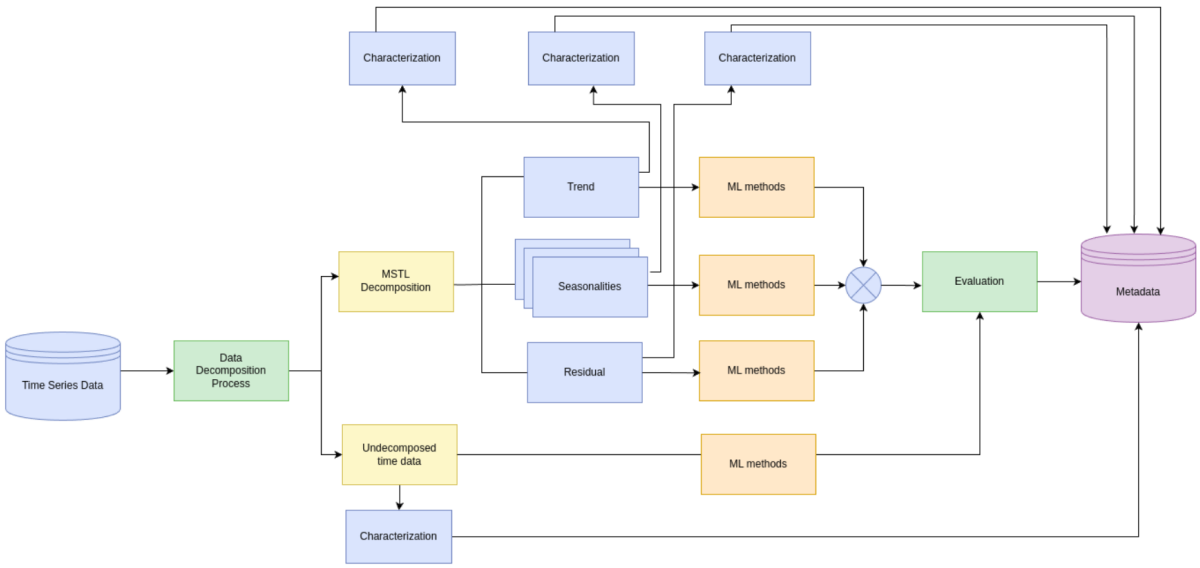



Figure 3.1: Pipeline of the metadata collection step.

The metadata collection step consists of the initial steps of analysing the time series data as a preparatory step to model recommendation. As shown in the Figure 3.1, this step starts with an input time series dataset $Y = \{Y_0, Y_1, ..., Y_n\}$. Each series undergoes a first decomposition decision that decides whether a series shall be decomposed or not. The basis for the decision depends on the nature of the distribution of the residual component, as a result of the findings by Santos et al. (2021) [4]. Time series residuals of normal distribution proceed with the decomposition using Multi-Seasonal-Trend decomposition using Loess (MSTL), whose use and explanation will be addressed in the following chapter in more detail. MSTL does the decomposition of the trend, seasonality/seasonalities, and residuals of the time series data. All time series that do not go through the decomposition process due to their residual's distribution skip such a process.

Component-specific modelling and meta-feature extraction then follow. Meta-features summarise statistical properties, structural attributes, and other relevant characteristics of each time series. After the predictions of each model available in the roster of the framework are made, the error of the prediction is extracted by comparing the forecast with the actual values. The errors of the models for a given series are then stored in the metadata repository along with the characteristics of each series, building a knowledge base that will become crucial to drive the next step of model recommendation. In essence, this repository of features and their corresponding model errors constitutes the training dataset for the metalearner, enabling it to learn the relationship between a time series' characteristics and the most effective forecasting model, which is the model with the lowest forecasting error.

### 3.1.2 Models recommendation step

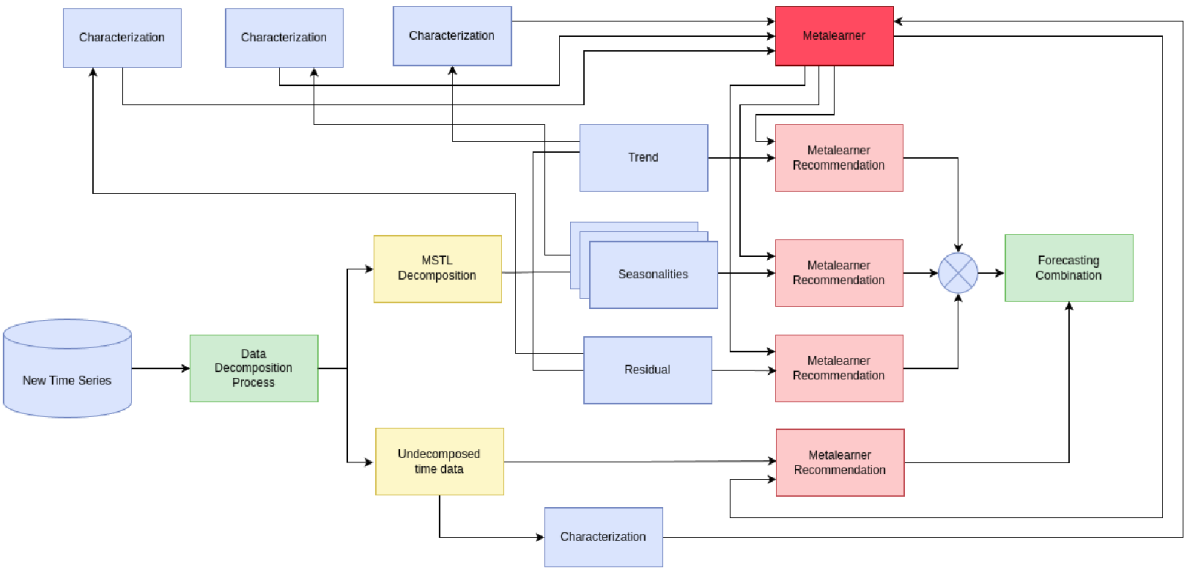



Figure 3.2: Pipeline of the models recommendation step.

During this second step, the model recommendation, metadata produced in the first step is used to generate predictions for the new time series $Y_{new}$. As illustrated in the Figure 3.2, the processes start by characterising the new series through computing their meta-features and matching them against the metadata repository.

The new series follows the same process for decomposition decisions. If the distribution of residuals suggests normality, then MSTL is used for decomposing $Y_{new}$ into its components: trend, seasonality/seasonalities, and residual. For each component, metadata will be analysed by the metalearner, which in turn will suggest a machine learning model, taking into account the characteristics of that component. In cases where time series are not decomposed, the metalearner will point to one model for the forecast of the whole series.

Forecasts from the individual component's models are then combined, or the single model in the case of non-decomposed series, to produce the final forecast. In cases of decomposed series, the forecasted trend, seasonality/seasonalities, and residuals are combined to provide the overall forecast. This will ensure that each component is optimally modelled, while the approach for an undecomposed series is simpler but quite effective, where there is no need to decompose.

# Chapter 4

# Experimental Procedures

This chapter details the experimental setup used to assess the performance of the proposed forecasting framework. The key tools and methods used are described, including the dataset, the set of forecasting models, the feature extraction libraries and the metalearning algorithms. The evaluation is done at two different levels, which are the meta-level, where it is checked how well the metalearner makes its decisions, and at the base-level, where the final forecasting accuracy of the system as a whole is measured. Defining this process clearly is important for understanding the results presented in the next chapter.

## 4.1 Materials and Methods

In this section, the data and methodological framework serving as a foundation to the approach to metalearning-based forecasting are described. It is detailed the time series datasets used, the decomposition strategy employed and the structure of the metalearning system, covering aspects such as the set of meta-features employed, the suite of base-learners, and more.

### 4.1.1 Time series data

The experiments conducted in this study utilize the M4 competition dataset [50], selected for its strong representation of real-world time series data. Unlike earlier benchmarks such as the M3 dataset, which includes a mix of synthetic and less diverse series, the M4 dataset is composed of genuine, application-driven time series, making it particularly suitable for evaluating practical forecasting performance.

To evaluate the robustness of the proposed approach across varying temporal structures, the focus is on two frequency subsets of the M4 dataset: monthly and quarterly series. This selection enables the assessment of forecasting effectiveness across different time horizons. The selected subsets of the M4 dataset includes a total of 72,000 time series, drawn from a diverse array of real-world domains, including micro and macro economic indicators, industry metrics, financial markets, demographic trends, and other sectors.

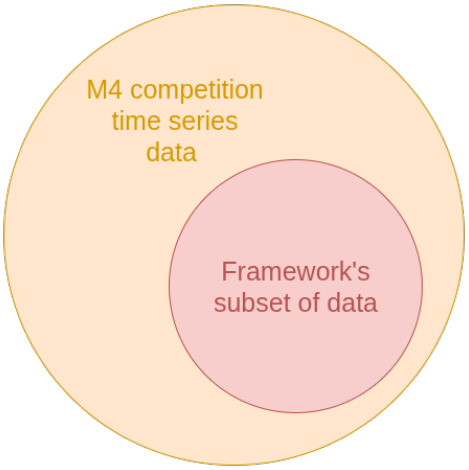



Figure 4.1: Relationship between the entire M4 competition data and the subset used in the proposed framework.

However, due to computational constraints and the intensive nature of model training within the proposed framework, processing the entire set of 72,000 series was not feasible. To strike a balance between representativeness and tractability, and as illustrated in the Figure 4.1, 5,000 time series from each of the two frequency groups (monthly and quarterly) were sampled. This subset was chosen to retain substantial diversity in series characteristics while ensuring that experiments could be conducted within available resource limits. Each of these 5,000 series sets was subsequently partitioned using an 80/20 train-test split, allowing sufficient data for both model training and evaluation. This sampling strategy provided a practical compromise that preserved generalisability while maintaining computational feasibility.

### 4.1.2 Decomposition

To address the complexity of real-world time series, this work adopts the Multi-Seasonal-Trend decomposition using Loess (MSTL) technique. MSTL extends the widely used STL method by supporting multiple overlapping seasonal patterns, a common characteristic in practical datasets such as electricity demand or economic indicators.

MSTL decomposes a time series $Y_t$ into trend, multiple seasonal components, and residuals. It supports both additive and multiplicative formulations, whose equations are presented in 2.2 and 2.3 respectively, where $T_t$ is the trend component, $S_t^{(i)}$ represents the $i$-th seasonal component, $R_t$ is the residual, and $k$ is the number of detected seasonalities.

The extraction of these components is performed iteratively using Loess, a non-parametric local regression technique. To ensure the decomposition is as accurate as possible, our approach first identifies the relevant seasonal periods within each time series. This is achieved by using the Fast Fourier Transform (FFT), a powerful form of spectral analysis that decomposes a signal into its constituent frequencies, as explored by Santos et al. (2021) [51]. By analyzing the resulting spectrum, we can identify the dominant periodicities (such as, daily, weekly, and yearly). These empirically detected seasonal periods are then explicitly provided to the MSTL algorithm, guiding it to extract the correct and most significant seasonal patterns from the data.

When examining time series with several overlapping seasonal patterns, MSTL's superiority over conventional STL can be demonstrated in practice. The same bi-seasonal time series is used in the figures that follow to demonstrate this comparison.

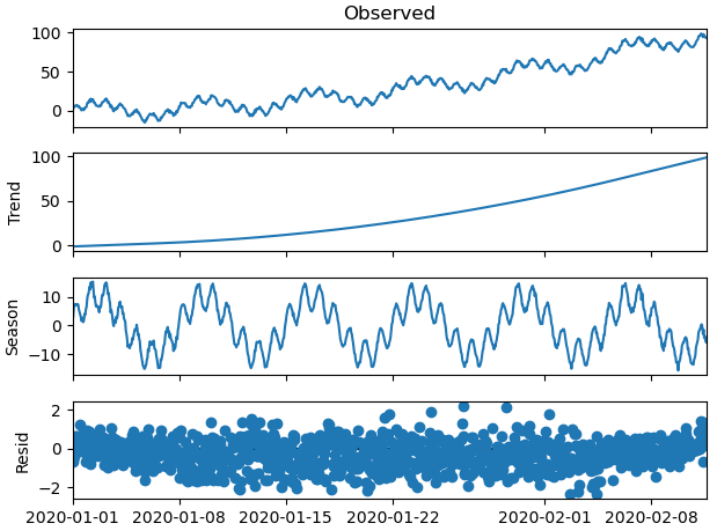



Figure 4.2: STL decomposition of bi-seasonal time series.

As can be observed in the Figure 4.2, the seasonal component still clearly possesses another seasonality that has not been isolated since the STL decomposition only records one seasonal component, while other seasonal components, if existant, remain.

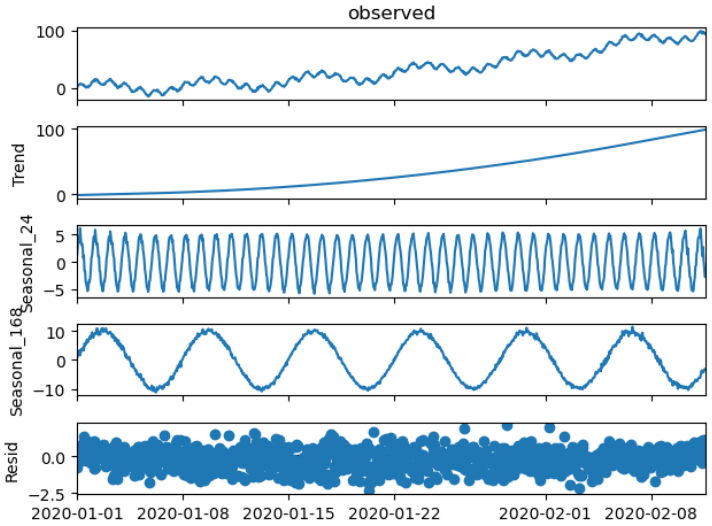



Figure 4.3: MSTL decomposition of bi-seasonal time series.

Conversely, and as can be noted from analysing the Figure 4.3, MSTL clearly decomposes the two seasonal factors and cleanly separated the weekly and daily cycles with the result consisting of more understandable factors. Such a decomposition highlights MSTL's great potential in forecasting pipelines, especially in contrast with STL, by demonstrating the ability to provide an improved representation of complex real-world scenarios, with greater automation and flexibility.

### 4.1.3 Metalearning

In this subsection, the metalearning component of the proposed framework is described. The meta-features which have been extracted to characterize each time series are described, followed by the usage of the base forecasting models to obtain performance data and generate forecasts. Then, a description of how the meta-target was defined in preparation for training the metalearner is conducted, finishing off by describing the metalearning strategy employed to predict model performance and enable algorithm selection.

#### 4.1.3.1 Meta-features

To characterize each time series and enable the metalearning process, the meta-features are extracted from two widely used time series feature extraction libraries: Catch22 [52], and TSFeatures [30]. The idea was to understand the effect of different sets of statistical and structural features on the performance and robustness of the forecasting system, as well as which meta-feature set allows the metalearner to provide the most accurate model recommendations.

The Catch22 package computes 22 carefully selected time-series features that are designed to cover a wide set of statistical properties with minimal redundancy. It provides a lightweight but informative feature set ideal for fast, interpretable analysis. In contrast, TSFeatures focuses on features that are widely used for forecasting tasks such as trend strength, seasonality, autocorrelation, and spectral entropy. This method is intended to capture general time series patterns and is widely used in forecasting competitions.

Every one of the two feature extractors were used separately on the dataset to generate distinct metadata representations. These were utilised to train individual instances of the metalearner, allowing for a comparison of the effect of different feature sets on model selection quality. This approach allowed to assess the adaptability and generalization capability of the framework under varying descriptive characteristics of time series data.

#### 4.1.3.2 Base-learners

The algorithms that were leveraged as base-learners are present in the Table 4.1, as well as the configurations for the search of the optimal parameters. First, the models were chosen based on their prevalence and strong performance as reported in the existing literature. Second, a key objective was to create a diverse and comprehensive suite of models for the metalearner to choose from. To achieve this, models from different families were included: traditional statistical methods (Simple Exponential Smoothing (SES), Trigonometric, Box-Cox transform, ARMA errors, Trend, and Seasonal components (TBATS), and Seasonal Naive (sNaive)), classical machine learning models (Random Forest (RF), Light Gradient-Boosting Machine (LGBM), and Linear Regression (LR)), and modern deep learning architectures (such as Temporal Convolutional Networks (TCN), Neural Hierarchical Interpolation for Time Series (NHITS), Deep Autoregressive (DeepAR) and Multilayer Perceptron (MLP)). This variety ensures the framework has a wide range of models at its disposal, making it capable of handling many different types of time series patterns.

In the initial phase, where forecasting model performance was assessed to build the metadata for training the metalearner, simplified versions of the models were used. The objective at this stage was not

to achieve optimal forecasting accuracy but rather to obtain a representative estimate of each model's relative performance across different types of time series. As such, no parameter optimisation techniques, such as grid search, were applied during this phase. This decision was made to control computational cost, which would have increased significantly had been performed hyperparameter tuning for each model across the entire dataset. Instead, model tuning was deferred to a later stage, after the metalearner recommended the most suitable forecasting model for a given time series. At that point, grid search was employed to fine-tune the selected model and maximise its predictive performance on the final forecasts.

Table 4.1: Base-learners and their configurations.

| **Model** | **Hyperparameters** |
|---|---|
| TCN | max_steps=5000, input_size=2*horizon, scalar_type="robust", random_seed=14, val_check_steps=100 |
| DeepAR | max_steps=10000, input_size=2*horizon, scalar_type="robust", random_seed=15, val_check_steps=100 |
| NHITS | max_steps=4000, input_size=2*horizon, scalar_type="robust", random_seed=16, val_check_steps=100 |
| MLP | max_steps=2500, input_size=2*horizon, scalar_type="robust", random_seed=17, val_check_steps=100 |
| LR | AutoLinearRegression via AutoMLForecast, num_samples=30, season_length={season_length}, freq=1 |
| LGBM Regressor | AutoLightGBM via AutoMLForecast, num_samples=30, season_length={season_length}, freq=1 |
| RF Regressor | RandomForestRegressor with AutoModel, n_estimators=[50,100], max_depth=[3, 15], max_samples=[0.3, 1.0], num_samples=30 |
| TBATS | season_length={season_length} |
| SES | alpha=0.5 |
| Seasonal Naive | season_length={season_length} |

For models such as Linear Regression, LGBM Regressor, and Random Forest Regressor, it was made use of the AutoMLForecast [53] and AutoModel [54] framework to perform automated hyperparameter tuning based on parameters defined in the Table 4.1. Models like TBATS, SES, and Seasonal Naive do not require or benefit from hyperparameter tuning, given that they are either fit by internal optimization routines (TBATS, SES) or are parameter-free heuristics (Seasonal Naive). Across some models, it was made use of a parameter called `season_length`. This value depends on the frequency of the dataset that is being used. In the event that the monthly subset of data is used, the value used is `12`, while if the quarterly subset of data is used, the value used is `4`, both to use lags representing a year time period depending on the data frequency.

#### 4.1.3.3 Meta-target definition

For the definition of the meta-target, the lowest value of the Mean Absolute Error (MAE) was used, as defined in 4.1. MAE was selected due to its interpretability, scale invariance, and consistent treatment of all errors, avoiding the over-penalization of outliers inherent to squared-error metrics. Moreover, a significant portion of the related works reviewed [5, 10, 11, 13, 16, 17, 24, 34, 35, 36, 37, 42, 44, 45, 46]

also adopted MAE as an evaluation metric, reinforcing its relevance and facilitating comparison with established benchmarks.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.1}$$

#### 4.1.3.4 Metalearner

The metalearner employed in this work is a multi-output regression model, where each output corresponds to the estimated Mean Absolute Error (MAE) of a specific forecasting algorithm. Since it performs consistently and doesn't require a lot of hyperparameter tweaking, a Random Forest Regressor was first selected as the base estimator for every output. The metalearner can anticipate the performance of several candidate models at once thanks to the multi-output structure, which makes it easier to choose a model based on predicted forecasting accuracy.

As a backup basis model, a LGBM Regressor was also constructed and assessed to make sure that the metalearner selection did not constitute a performance bottleneck. LGBM is renowned for its great accuracy and speed, especially when working with big, high-dimensional datasets. The prediction performance of the two metalearning models could be directly compared because they were trained and evaluated in the same way. Based on validation criteria, the top-performing metalearner was chosen to be included in the final forecasting pipeline.

The literature review findings, which support the use of multi-output regression [27] in metalearning situations and identifies Random Forest [5, 8] as a strong candidate, were also one of the reasons that influenced and served as the basis for the choice of the metalearner. As the focus of this work is not on optimising the metalearning algorithm itself but to grant that it would not be impacting the results negatively, only a few hyperparameters were changed, while all others were kept at their default values. The values used for the metalearners are covered in the Table 4.2.

Table 4.2: Metalearning models and their configurations.

| **Model** | **Hyperparameters** |
|---|---|
| RF Regressor | n_estimators=100, random_state=0 |
| LGBM Regressor | boosting_type='gbdt', objective='regression', n_estimators=1000, learning_rate=0.01, num_leaves=31, max_depth=-1, min_child_samples=20, subsample=0.8, colsample_bytree=0.8, reg_alpha=1.0, reg_lambda=1.0, random_state=42 |

A distinct metalearner of each model was trained for each frequency (monthly and quarterly) to account for the differing temporal dynamics inherent to each. For instance, monthly time series often exhibit finer-grained seasonal and trend variations compared to quarterly series, which may reflect broader economic or structural patterns. By training models based on each frequency, the metalearners can better capture the statistical and structural properties unique to each resolution.

To train the metalearner, a metadata table was constructed by associating time series characteristics (meta-features) with the forecasting performance of all the models. Each row in the table corresponds to a single time series and includes both its extracted meta-features and the corresponding MAE values obtained from applying different forecasting models to that series.

Table 4.3: Metadata table example with feature values and MAE results for each time series.

| **Time Series** | **f1** | **f2** | **f3** | ... | **m1** | **m2** | **m3** | ... |
|---|---|---|---|---|---|---|---|---|
| ts_1 | f1_v1 | f2_v1 | f3_v1 | ... | mae_11 | mae_12 | mae_13 | ... |
| ts_2 | f1_v2 | f2_v2 | f3_v2 | ... | mae_21 | mae_22 | mae_23 | ... |
| ts_3 | f1_v3 | f2_v3 | f3_v3 | ... | mae_31 | mae_32 | mae_33 | ... |

This structure allows the metalearner to learn the relationship between the input features and the expected error for each model. A simplified representation of this metadata is shown in the Table 4.3, where each column $f_i$ corresponds to a meta-feature and each column $m_j$ represents the MAE of model $j$ on the corresponding time series.

## 4.2 Evaluation

This section outlines the evaluation methodology used to assess model performance at both the base and meta levels. The set of metrics was specifically chosen to the distinct nature of the task at each level. For the base-level evaluation of time series forecasts, the Root Mean Squared Error (RMSE), Normalised Mean Absolute Error (NMAE), and Mean Absolute Scaled Error (MASE) were used. For the meta-level, which constitutes a regression task, performance was assessed using RMSE and NMAE. The exclusion of MASE from the meta-level evaluation is due to its specific design for time series. MASE achieves scale-independent comparison by normalizing errors against a naive time-series forecast, a process that requires sequential data and is therefore unsuitable for standard regression tasks where data points have no inherent order.

Table 4.4: Evaluation metrics used to assess the metalearner's performance.

| **Metric** | **Description** | **Purpose** |
|---|---|---|
| RMSE | Measures the average magnitude of the forecast errors. It gives significantly more weight to large errors. | Assesses overall accuracy where large errors are particularly undesirable. |
| MASE | Compares the model's average error to the average error of a simple, naive forecast. | Compares forecast accuracy across time series that have different scales or units. |
| NMAE | Expresses the average absolute error as a percentage of the average actual value. | Provides an intuitive, percentage-based error that is easy to interpret and compare. |

Table 4.4 summarises the evaluation metrics used throughout this evaluation phase covered in the subsections 4.2.1 and 4.2.2, along with their definitions and respective roles in performance analysis.

### 4.2.1 Meta-level Evaluation

At the meta-level, the performance of the metalearner was estimated using a 10-fold cross-validation approach. This method was chosen to provide a reliable and robust estimate of predictive accuracy without requiring a dedicated hold-out test set. Since the metalearner serves a supporting role within the broader forecasting framework, rather than being the primary focus of this work, the aim was to obtain a well-founded indication of its effectiveness in guiding model selection, instead of an exhaustive

evaluation of its standalone generalisation performance. After splitting the dataset into ten folds the metalearner cycled through every possible combination by iteratively training on nine folds and testing on the remaining one. Beyond providing insight into how stable and variable the predictions of the metalearner are on data partitions, the procedure also helps in preventing the risk of overfitting. An illustration of the 10-fold cross-validation procedure is presented in Figure 4.4.


Figure 4.4: Process of 10-fold cross validation.

The metalearner was also provided with a collection of basic forecasting models and was requested to forecast the Mean Absolute Error (MAE) values in each fold. The forecasts provided by the metalearner were then compared against the true forecast errors. The regression evaluation metrics listed in Table 4.4, which are RMSE and NMAE, were calculated in order to measure the performance of the metalearner's forecasts. These metrics as a whole provide a general representation of the model's performance in terms of explanatory power dimensions, variability, and error magnitude.

The per-fold performance measure on each of the ten folds was preserved, as well as an average of performance across the ten folds. This enabled to investigate how well the metalearner's predictions held up across various data segments. This also allowed to get a better understanding of how stable and robust the model was by considering the per-fold and the aggregate metrics. The consideration at two levels enabled to identify whether there were certain folds, possibly representative of more challenge or atypical samples of data, whose impact on the aggregate evaluation was skewed.

Two distinct regression models were investigated, Random Forest (RF) and Light Gradient-Boosting Machine (LGBM), as potential metalearners to make sure that the selection of a metalearner would not become a limiting element in the overall forecasting framework. These models were chosen due to their shown performance in tabular regression tasks, as well as their capacity to manage feature interactions and non-linear connections. Evaluating both RF and LGBM enabled to investigate the possible influence of model selection on the meta-level predictions, even though the main objective was not to fully optimise the metalearner itself. The purpose of this comparative arrangement was to confirm that the metalearner component was suitably constructed to support the larger system and to guarantee that inadequate meta-level modelling would not limit the forecasting performance. More generally, adding a metalearner made it possible to choose the best performing forecasting model from a group

of candidates in a flexible and computationally efficient way. This is in line with the goal of increasing accuracy while preserving scalability in real-world forecasting applications.

### 4.2.2 Base-level Evaluation

At the base level, the evaluation aimed to assess the forecasting performance of the proposed metalearning-based model selection framework in comparison to a range of well defined baselines. All analyses were conducted using the full set of evaluation metrics introduced at the beginning of this section in the Table 4.4. These metrics were selected to provide a robust and multi-dimensional view of predictive performance, capturing different facets such as average error magnitude, sensitivity to large deviations, and overall explanatory strength.

Forecasts were generated using the proposed framework across two different time series characterization strategies: `Catch22`, and `TSFeatures`. This setup enabled to investigate how the choice of feature representation influenced the accuracy and robustness of the forecasting pipeline. These predictions were then compared against those produced by a set of baseline models.

The baselines used in this evaluation consisted of all the individual forecasting models that the proposed metalearning framework was designed to select from, all of them specified in the Section 4.1.3.2. Each baseline model was executed independently to generate forecasts across the exact same set of time series as the proposed framework, ensuring strict comparability. Once all predictions were collected, the full suite of evaluation metrics was computed by comparing the forecasts with the corresponding ground truth values. This allowed for a consistent, side-by-side comparison of performance across all methods.

To deepen the analysis, a variety of statistical and visual tools was applied, including statistical significance heatmaps, consistency score plots, and more, all of which will be introduced and explained in further detail in the next chapter. These analyses provided insight into not just mean performance, but also variability and robustness across different forecasting approaches.

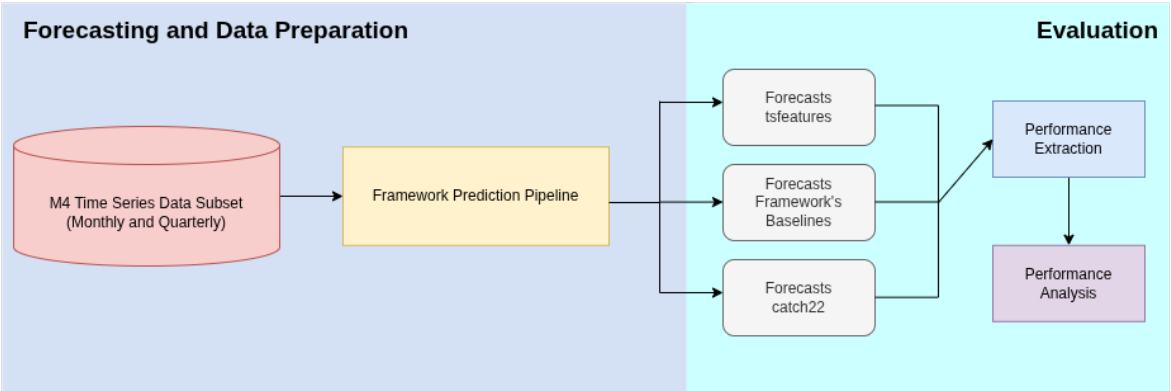



Figure 4.5: Base-level evaluation pipeline.

Figure 4.5 illustrates the complete evaluation workflow for the base-level analysis. The subset selection of the M4 competition data is employed at the start of the forecasting and data preparation block. The proposed framework processes this subset through a prediction pipeline that generates forecasts based on two alternative feature extraction strategies, `Catch22` and `TSFeatures`. Parallelly, the baseline models generate their own forecasts on the same subset, but they skip the decomposition steps, generating forecasts on the time series as a whole. The idea behind skipping the decomposition in the

baselines alone is to understand clearly what the performance advantage of the proposed framework would be in comparison with forecasts made uniquely by the baselines without any interference.

All forecast outputs are then passed into an evaluation block, where performance metrics are computed (Performance Extraction), and then analyzed using various statistical and visualization tools (Performance Analysis). This modular and comparative structure ensures transparency and reproducibility in assessing how different methods perform under identical conditions.

The adoption of this evaluation strategy allowed to assess the relative strengths and weaknesses of the proposed framework, explore the effect of different feature sets, and understand the advantage of leveraging the proposed metalearning approach over single baseline models in making time series forecasts.

# Chapter 5

# Results

This Chapter presents and interprets the experimental results from the evaluation procedure detailed in Chapter 4. The analysis is structured to provide evidence-based answers to the guiding research questions. To this end, the Chapter first addresses each research question sequentially, followed by a series of ablation studies that investigate the impact of specific framework components. The primary analysis is conducted on monthly time series data, while results from quarterly data are discussed in a separate section, where the reasons for its distinct treatment will be discussed.

## 5.1 Results for RQ1: Metalearner for Decomposition and Model Selection

This section presents and explores the performance results of the proposed metalearning framework. The primary objective is to evaluate the effectiveness of the approach as a whole of allowing the metalearner to choose between a single forecasting model and a decomposition-based approach over a similar metalearning approach that skips the decomposition step. The evaluation is framed to directly address the research question:

*What is the advantage of leveraging the metalearner for choosing between individual models and decomposition-based methods for the formation of forecasting combinations?*

The performance of four distinct approaches was measured:

- **Catch22:** Metalearner using the Catch22 feature set with the ability to select decomposition.
- **TSFeatures:** Metalearner using the TSFeatures library with the ability to select decomposition.
- **Catch22_und:** Metalearner using Catch22 features but restricted to selecting only individual, non-decomposed models.
- **TSFeatures_und:** Metalearner using TSFeatures but restricted to selecting only individual, non-decomposed models.

The results are presented and visualised in following section so that it is possible to analyse and understand them, drawing insights that allow to answer the research question being addressed. The metrics analysed are the ones mentioned in Section 4.2 referent to the base-level.

### 5.1.1 Presentation of results

The empirical results from the experiments are presented in this subsection. Firstly, it will be introduced a detailed table of performance metrics, followed by visualizations that graphically represent the performance and consistency of the evaluated methods. The goal is to provide a comprehensive and clear illustration of the collected data before proceeding to a more in depth analysis and drawing conclusions from data.

#### 5.1.1.1 Quantitative Performance Metrics

The core performance metrics used to evaluate the performance of all the four methods are summarized in Table 5.1, which contains the mean, standard deviation (in the format mean ± std), and the median (md) for the MASE, NMAE, and RMSE for each of the four methods being analysed. For each of these metrics, it is reported the mean, the median, and the standard deviation, with the goal of understanding the average performance, the performance but less sensitive to extreme values and outliers, and the performance consistency of the methods. A lower standard deviation indicates that a method's performance is more stable and reliable across different time series.

Table 5.1: Comparison of performance metrics across feature extraction methods.

| Model | MASE | MASE md | NMAE | NMAE md | RMSE | RMSE md |
|---|---|---|---|---|---|---|
| Catch22 | 3.50 ± 6.59 | 1.83 | 0.17 ± 0.21 | 0.10 | **993.64 ± 1503.74** | 477.93 |
| Catch22_und | 3.30 ± 6.26 | **1.78** | 0.17 ± 0.25 | **0.09** | 1057.47 ± 1818.52 | **453.23** |
| TSFeatures | **2.73 ± 2.52** | 1.80 | **0.16 ± 0.20** | **0.09** | 1005.13 ± 1570.46 | 487.96 |
| TSFeatures_und | 3.54 ± 5.50 | 1.96 | 0.19 ± 0.28 | 0.11 | 1077.16 ± 1884.04 | 510.47 |

A preliminary analysis of the table reveals that the methods that leverage decomposition, which are TSFeatures and Catch22, generally outperform their non-decomposition counterparts. The TSFeatures approach, in particular, achieves the best average performance, recording the lowest mean value of MASE and mean value of NMAE. However, the most interesting finding lies in the standard deviation metrics. For nearly every metric, the decomposition-based methods show substantially lower standard deviations. For instance, the standard deviation of RMSE for TSFeatures and Catch22 is lower than the one of TSFeatures non-decomposed and Catch22 non-decomposed, meaning a more reliable and stable performance. Interestingly, an analysis of the median values shows that Catch22 non-decomposed records the lowest median value of RMSE, suggesting it performs very well on the typical time series but fails on more complex ones, leading to the poor average scores and high volatility.

#### 5.1.1.2 Statistical Significance Heatmap

A statistical significance heatmap is presented in Figure 5.1 providing a pairwise comparison of the four methods, assessing whether the observed performance differences are statistically significant or not. This matrix was generated using the methodology detailed by Ismail-Fawaz et al. (2023) [55]. It is structured to compare every method against every other method using the Normalized Mean Absolute Error (NMAE) as the main metric. NMAE was given preference for this analysis due to its percentage-based nature, which makes the performance differences more straightforward to interpret. The methods

are ordered along both the x and y-axes according to their overall performance, from lowest to highest mean value of NMAE, starting from the top-left corner.

Each cell in the heatmap provides a comparison between the method in its row ($r$) and the method in its column ($c$). The top number in each cell shows the mean difference in NMAE, as expressed in the equation 5.1, with the cell's color visually representing this same value, having blue indicate the row model performed better, in other words, a negative difference, while red means the column model was superior, which is the same as saying a positive difference. The intensity of each color is indicative of the magnitude of the difference.

$$(NMAE_r - NMAE_c) \tag{5.1}$$

It is also important to mention that the fraction-like numbers (for instance, the value "505 / 11 / 484" in the row "tsfeatures" and column "catch22") detail the exact count of time series where the row model won, tied, or lost against the column model, respectively. The value at the bottom of the cell is the p-value from the Wilcoxon signed-rank test, which assesses whether the observed performance difference between the two paired models is statistically significant.

An ideal result for a given method is to have its corresponding row dominated by blue cells, signifying superior performance against others. On the other hand, the worst outcome is a row dominated by red cells. The most important indicator is the formatting, where cells with text in bold indicate that the p-value is less than 0.05. This means the observed performance difference is statistically significant and not likely due to random chance. Non-bold cells suggest that while there might be a difference in mean performance, it is not statistically significant. This plot goes further than a simple comparison of averages by providing statistical evidence of the superiority of one method over another.

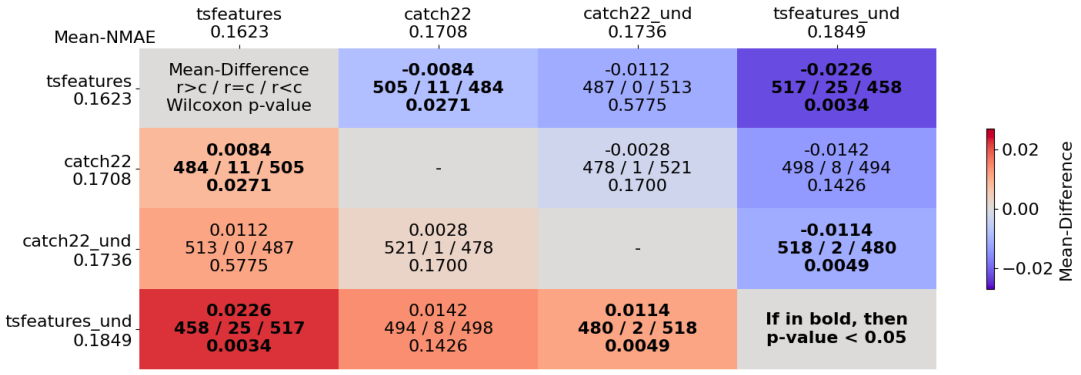



Figure 5.1: Heatmap of pairwise NMAE comparison with Wilcoxon signed-rank test of decomposed and non-decomposed approaches.

This statistical test provides validation for the initial observations from the performance metrics table. The heatmap clearly identifies TSFeatures as the best performing method, as it is not only ranked first but is also revealed to be statistically better than its closest competitor, Catch22 (p = 0.0271). More importantly, it demonstrates a highly significant advantage over its non-decomposed counterpart (p = 0.0034), providing strong evidence that the inclusion of decomposition is beneficial within the framework. The bottom row confirms that TSFeatures non-decomposed is the worst performer, as it is statistically outperformed by both TSFeatures and Catch22 non-decomposed. Interestingly, the analysis

also reveals that the performance difference between Catch22 and Catch22 non-decomposed is not statistically significant, indicating that the benefits of decomposition are more evident for the TSFeatures set.

#### 5.1.1.3 Consistency Score Plot

Figure 5.2 displays the consistency of each method for the NMAE and MASE metrics. This score is calculated as the ratio of the mean to the standard deviation, having the formula present in the Equation 5.2. A higher ratio indicates a more desirable performance, as it implies the method's average performance is high relative to its variability.

$$\text{Consistency Score} = \frac{\text{Mean value}}{\text{Standard Deviation}} \tag{5.2}$$

The plot is a 2D space where the x-axis represents MASE consistency and the y-axis represents NMAE consistency. The ideal outcome for a method is to be located in the top-right corner. A point in that region signifies high consistency across both scaled and normalized error metrics, making the method both effective and reliable. On the other hand, the worst outcome is a point near the origin in the bottom-left corner, which indicates low consistency and therefore erratic and unpredictable performance.

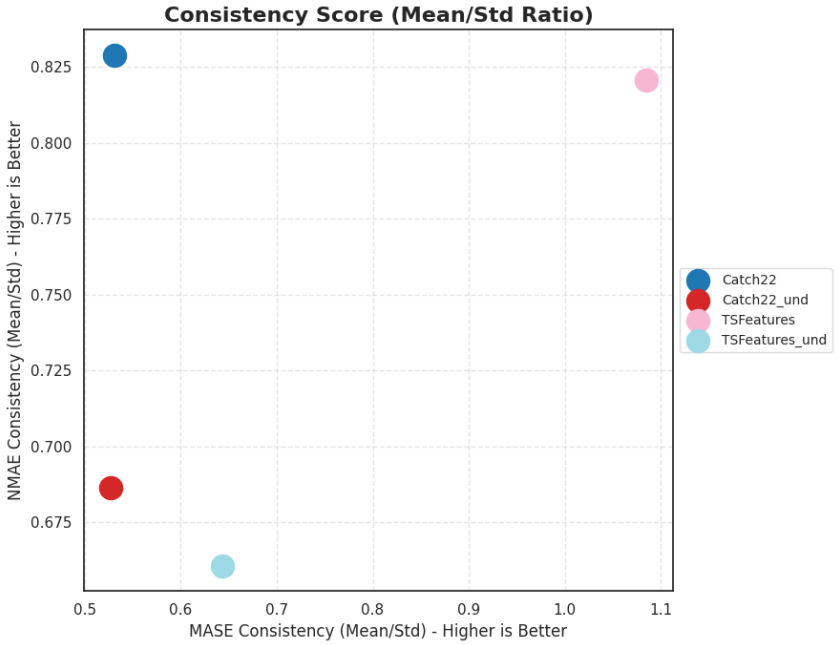



Figure 5.2: Consistency Score plot of the four approaches.

The plot provides visual confirmation of the performance differences, clearly separating the methods into two different groups. The decomposition-based methods, TSFeatures (pink) and Catch22 (dark blue), are placed in the top half of the plot, indicating their better consistency. TSFeatures, in particular, is positioned in the ideal top-right quadrant, demonstrating the highest consistency on the MASE metric and strong consistency on NMAE. In contrast, the non-decomposed methods, Catch22 (red) and TSFeatures (light blue) non-decomposed, are positioned in the bottom-left region. Their position signifies substantially lower consistency scores on both axes, reinforcing the conclusion from the previous table that these methods are less reliable and produce more erratic forecasts.

#### 5.1.1.4 RMSE-based Performance Space

The Figure 5.3 offers a different perspective, focusing on the Root Mean Squared Error (RMSE). This plot maps the average performance (RMSE mean value) against the performance volatility (RMSE standard deviation). This visualization is key in understanding the trade-off between the magnitude of the error and its consistency.

Regarding the plot, the axes represent two dimensions of performance, the mean value of the RMSE in the x-axis and its standard deviation value in the y-axis. The ideal outcome for any method is to be in the bottom-left corner, meaning high accuracy combined with high consistency. On the other hand, the worst outcome is to be situated in the top-right corner, which represents a method that is both inaccurate on average, due to high error value, and highly inconsistent, given its high error variance.

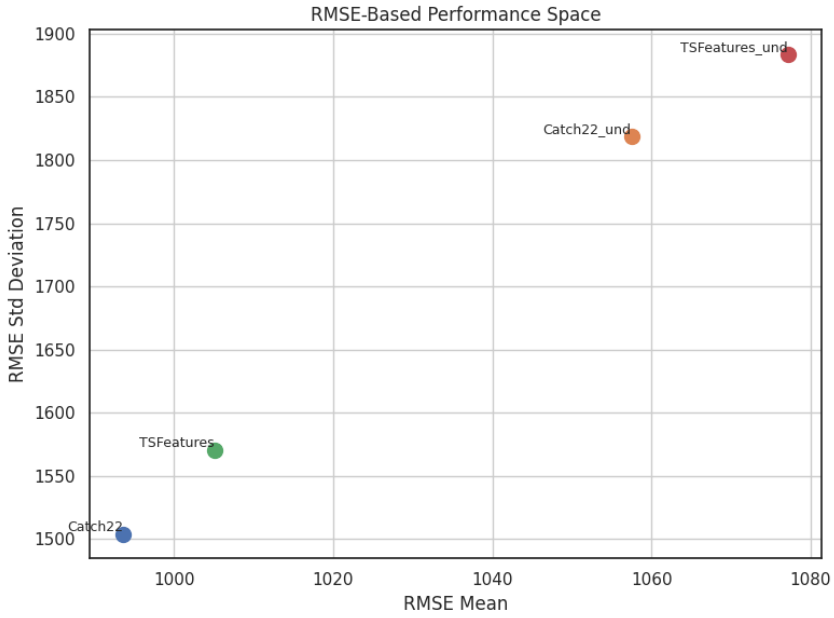



Figure 5.3: RMSE-based Performance Space plot of the four approaches.

By analysing the plot, there is a clear separation into two clusters. The decomposition-based methods, Catch22 and TSFeatures, are grouped together in the ideal bottom-left corner. This positioning means that they both achieve a lower mean RMSE and a lower standard deviation, confirming they are not only more accurate on average but also significantly more consistent. In contrast, the non-decomposed methods are isolated in the undesirable top-right corner. Their position indicates that they suffer from both higher average error and much greater performance volatility, branding them as inaccurate and unreliable in comparison. This plot provides clear evidence that the decomposition-based approaches are better in terms of both accuracy and reliability when evaluated with the RMSE metric.

### 5.1.2 Analysis and Discussion

This subsection provides an in-depth analysis of the experimental results presented previously. The discussion is structured to analyse the performance of the different methods, focusing on the concept of decomposition. By examining the mean, median, and standard deviation of the error metrics, it is

possible to extract insights into the accuracy and reliability of each approach, ultimately addressing the research question.

The most evident conclusion from the results is the positive impact of leveraging decomposition-based methods paired with a metalearner to produce forecasts. A direct comparison between the decomposition-based approaches (TSFeatures, Catch22) and their counterparts without decomposition methods (TSFeatures_und, Catch22_und) reveals a clear superiority across multiple dimensions of performance.

First, considering average performance, the methods with decomposition generally achieve lower mean errors. For instance, the TSFeatures approach records a mean MASE value of 2.73, a significant improvement over the 3.54 from TSFeatures_und. This demonstrates that, on average, the forecasts produced when decomposition is an option are substantially more accurate.

However, the most critical advantage is revealed by the standard deviation. The variability of errors for the non-decomposition methods is dramatically higher. The value of the standard deviation of MASE for TSFeatures is 2.52, which is less than half that of TSFeatures_und (5.50). Similarly, the value of the standard deviation of RMSE for Catch22 (1503.7) is significantly lower than for Catch22_und (1818.5). This is visually confirmed by the RMSE Performance Space plot (present in Figure 5.3), where TSFeatures_und and Catch22_und are isolated in the top-right quadrant, signifying both high average error and high performance volatility. This vast reduction in standard deviation means that the metalearner with the decomposition option is not just more accurate on average, but is more reliable and trustworthy, successfully avoiding the forecast failures that the non-decomposition methods suffer from.

This conclusion regarding improved reliability is further reinforced by the Consistency Score plot (present in Figure 5.2). This plot illustrates in a 2D space the ratio of the mean to the standard deviation for both MASE and NMAE, where a higher score signifies more stable and dependable performance. As can be inferred from it, the decomposition-based methods, TSFeatures and Catch22, occupy the desirable top-right quadrant, confirming their high consistency. In contrast, their non-decomposed counterparts are positioned in the bottom-left, highlighting their erratic performance. This visual evidence supports the finding that leveraging decomposition provides a key advantage in forecast reliability.

The statistical significance heatmap in Figure 5.1 provides the statistical evidence for these conclusions. It moves further from comparing averages and confirms that the best performing method, TSFeatures, is not just better than its non-decomposed counterpart, but is statistically significantly better, with a p-value of 0.0034. The win/loss numbers in the heatmap show that TSFeatures outperformed its non-decomposition version on 517 of the time series, compared to 458 losses. This validates that there is a performance advantage from leveraging decomposition in the TSFeatures feature set. The heatmap also reveals a more nuanced picture for the Catch22 feature set. While the decomposition version wins against its counterpart on more series (521 to 478), the difference in mean NMAE is not statistically significant, with a p-value of 0.1700. This suggests that while the decomposition strategy is highly effective and statistically verifiable for the TSFeatures set, its benefit is less pronounced for the Catch22 set.

Interestingly, an analysis of the median values introduces a valuable nuance. The value of the median of RMSE for Catch22_und (453.2) is the lowest of all four methods. This suggests that for the time series located in the median of it, the approach can perform well. However, the high mean and standard deviation of this same method indicate that it fails severely on more complex series. These difficult series produce extremely large errors that, while not affecting the median, drastically worsen the

average performance and destroy any notion of reliability. Therefore, the strength of the decomposition-based approaches lies in their robustness, given that they manage these difficult series, preventing the large errors that make the non-decomposition methods impractical and less well-suited for real-world application.

## 5.2 Results for RQ2: Comparison with Individual Models

This section presents and discusses the results of a performance analysis designed to compare the proposed metalearning framework against a comprehensive suite of individual forecasting models. The primary objective is to evaluate the advantages of the proposed approach when compared to relying on single, static forecasting methods. The evaluation is framed to directly address the research question:

*What is the advantage of leveraging this approach in comparison with individual models?*

The performance of the proposed framework, using the TSFeatures feature set due to reasons outlined later, is compared against ten individual baseline models. The baselines models are all models used within the framework, with the goal of understanding whether the metalearning approach achieves better performance than every single one of the baseline models. The full list of competitors is as follows:

- **TSFeatures:** The proposed metalearning framework using the TSFeatures library.
- **Baseline Models:** DeepAR, LGBM (Light Gradient Boosting Machine), LR (Linear Regression), MLP (Multilayer Perceptron), NHITS (Neural Hierarchical Interpolation for Time Series), RF (Random Forest), sNaive (Seasonal Naive), SES (Simple Exponential Smoothing), TBATS, and TCN (Temporal Convolutional Network).

The results are presented through several visualizations and tables in the following section. This data is then analysed to draw insights and conclusions that directly answer the research question being explored. The metrics values used are the ones referred in Section 4.2 regarding the base-level evaluation.

### 5.2.1 Presentation of results

This subsection presents the empirical results from the comparative experiments. Firstly, a statistical significance heatmap will be used, which provides a statistical ranking and comparison of all models. This is followed by two plots that visualize the performance and consistency of all evaluated methods and a table detailing per-series wins. The goal is to provide a diverse and clear illustration of the collected data before proceeding to a more in-depth analysis. The values of the quantitative performance metrics being analysed are present in the Table A.2.

#### 5.2.1.1 Statistical Significance Heatmap

To statistically validate the performance differences between the models for this research question, the significance heatmap presented in Figure 5.4 will be used. The results in that figure constitute a resumed version of the full one, which is A.1. As covered in more detail in Section 5.1.1.2, this plot provides

a pairwise comparison of models using the Wilcoxon signed-rank test, where results in bold indicate a statistically significant difference ($p < 0.05$). This analysis allows to confirm whether the observed differences in performance are due to random chance or represent an actual superiority of one method over another.

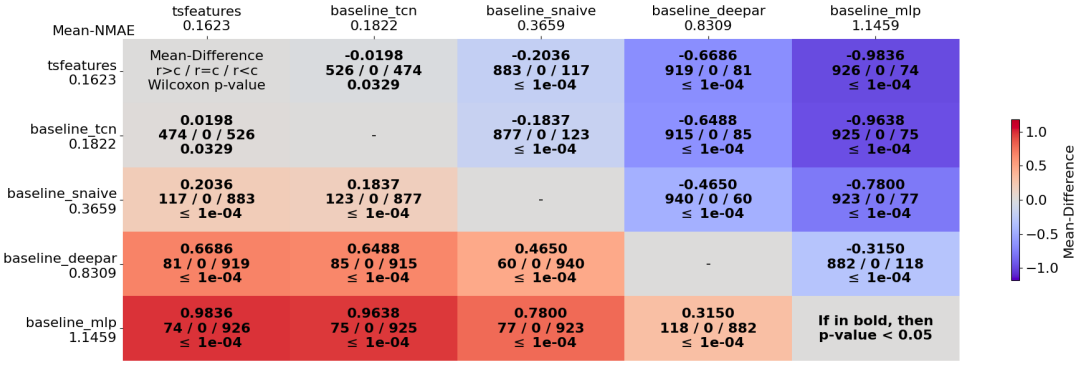



Figure 5.4: Heatmap of pairwise NMAE comparison with Wilcoxon signed-rank test of TSFeatures and some of the baseline models.

The results from the heatmap clearly show that the TSFeatures model outperforms all other methods by a significant margin. It achieves the lowest mean NMAE (0.1623) and consistently performs better in pairwise comparisons, with every comparison showing statistically significant improvements over the other models. The second best performer is the TCN baseline model, which also indicates strong performance, although it is slightly and significantly outperformed by TSFeatures. These two models form the top tier in terms of forecasting accuracy.

In contrast, methods such as DeepAR and MLP perform the worst, with high mean NMAE values (0.8309 and 1.1459 respectively) and consistent losses in head-to-head comparisons against almost every other model. The middle group, composed by Seasonal Naive, SES, TBATS, Random Forest, and LGBM, displays moderate performance, generally outperforming the worst models but falling short against the top two. In addition, the statistical tests indicate that most of these differences are not only large in magnitude, but also statistically significant, reinforcing the robustness of the performance rankings across the entire dataset.

#### 5.2.1.2 Consistency Score Plot

The Figure 5.5 displays the consistency of each method for the NMAE and MASE metrics, calculated as the ratio of the mean to the standard deviation. As covered in 5.1.1.3, a higher score indicates a more desirable and reliable performance, with the ideal outcome for a method to be located in the top-right corner of the plot.

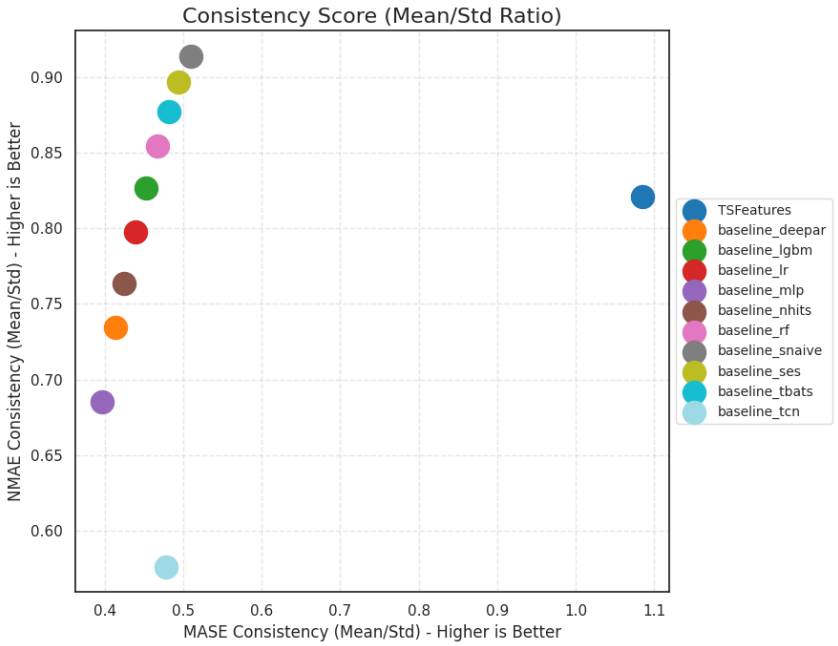



Figure 5.5: Consistency Score plot of all the methods.

The consistency plot reveals the overall performance landscape, with the TSFeatures model positioned in a category of its own. It is located alone in the ideal top-right region, demonstrating by far the highest MASE consistency and simultaneously one of the highest NMAE consistency scores. This placement identifies TSFeatures as the most reliable and stable model overall. In contrast, nearly all other baseline models are clustered in the top-left quadrant, indicating that while many of these models, such as LGBM and RF, achieve high consistency on the NMAE metric they all share a common weakness of poor MASE consistency. The plot therefore provides a powerful visual argument for the superior reliability of the TSFeatures approach, as it does not suffer from the consistency problems that characterizes the other methods.

#### 5.2.1.3 RMSE-based Performance Space

The Figure 5.6 utilizes the RMSE-based Performance Space plot, which was explained in detail above in 5.1.1.4. As a brief reminder, this plot visualizes the trade-off between a model's average error, with RMSE mean value in the x-axis, and its consistency, with RMSE standard deviation in y-axis. The most desirable outcomes are found in the bottom-left corner, representing models with both low and stable error rates.

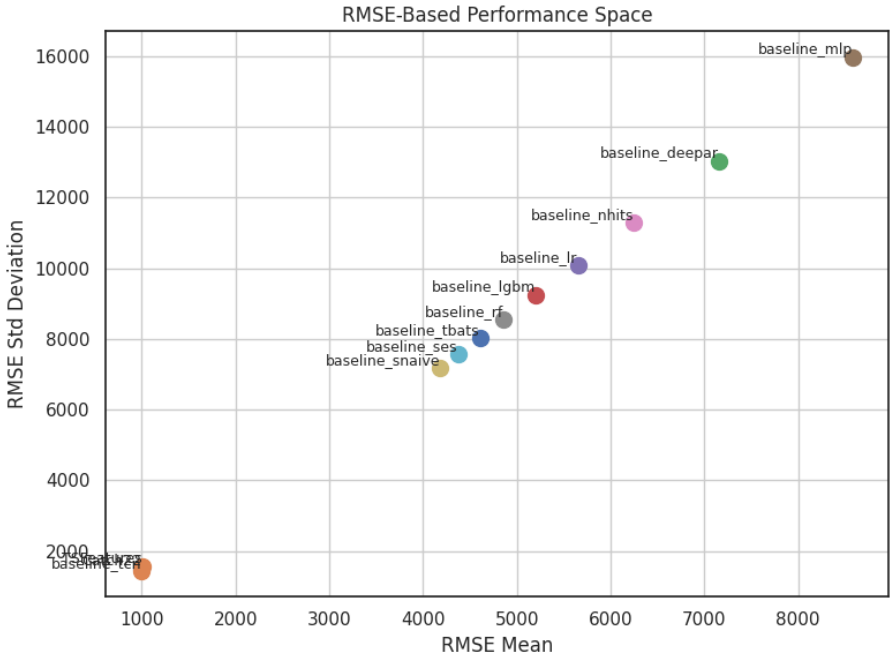



Figure 5.6: RMSE-based Performance Space plot of all the methods.

This performance space provides a powerful visualization of the models' effectiveness, separating them into two dramatically different performance tiers. The most evident conclusion is the great performance of the TSFeatures and TCN models. These two models are grouped together in the ideal bottom-left corner, isolating them from all other methods. This position means that they are vastly superior, achieving both the lowest average error and the highest level of consistency by a substantial margin.

In contrast, the majority of the other baseline models form a clear diagonal line stretching towards the top-right. This pattern is highly informative, as it illustrates a predictable trade-off, which is that among these methods, as the average error increases, so does the performance volatility. At the extreme end of this trend is the MLP model, which is isolated in the top-right corner as the least accurate and least reliable model. The plot makes it clear that TSFeatures and TCN are in a class of their own, escaping the accuracy-consistency trade-off that defines the other baseline methods.

#### 5.2.1.4 Per-series Performance Analysis

To complement the analysis of average error metrics, the Table 5.4 provides a win to loss perspective on model performance. This table changes the focus from average performance to dominance, counting the number of times each model achieved the single lowest error for an individual time series across the entire dataset. A win for a given metric, for instance, MASE, means that the model produced the best possible forecast out of all competitors for one specific series. This analysis is crucial for understanding the difference between a model that is consistently good and a model that is frequently the absolute best. A high number in this table indicates a model is often the optimal choice for a given task.

Table 5.2: Number of per-series wins (lowest error) per model for MASE, NMAE, and RMSE metrics.

| Model | MASE Wins | NMAE Wins | RMSE Wins |
|---|---|---|---|
| TSFeatures | **526** | **508** | **494** |
| Baseline_DeepAR | 0 | 8 | 5 |
| Baseline_LGBM | 0 | 1 | 1 |
| Baseline_LR | 0 | 0 | 1 |
| Baseline_MLP | 0 | 22 | 16 |
| Baseline_NHITS | 0 | 8 | 8 |
| Baseline_RF | 0 | 2 | 2 |
| Baseline_sNaive | 0 | 6 | 4 |
| Baseline_SES | 0 | 1 | 1 |
| Baseline_TBATS | 0 | 9 | 6 |
| Baseline_TCN | 474 | 435 | 462 |

This table reveals a hierarchy of model performance by counting the number of wins for each method. The results show that top-tier performance is concentrated in just two models, which are TSFeatures and TCN. TSFeatures is the clear leader, securing the best score more than 500 times for the MASE and NMAE metrics, which shows its remarkable ability to frequently produce the optimal forecast.

A particularly interesting finding is the performance of several well-established baseline models. For instance, LGBM and SES show identical, extremely low win counts. This trend is even more pronounced for Linear Regression, which secures only a single win in total. This suggests these are models whose strength lies in their ability to generalize and provide consistent performance, but they are rarely the single best choice for any individual time series. This contrasts with models like TSFeatures and TCN, which this table shows are highly effective at achieving the top rank in hundreds of individual cases.

### 5.2.2 Analysis and Discussion

This subsection provides an in-depth analysis of the experimental results, integrating the findings from the statistical and visual tools presented. The discussion is structured to have an understanding of the proposed framework's performance when compared with a suite of individual baseline models, focusing on the dimensions of statistical significance, performance consistency, and dominance per-series.

The results establish a distinct performance hierarchy, with the proposed TSFeatures framework and the TCN model consistently appearing as the top two performers. The statistical significance heatmap provides evidence of this, showing that TSFeatures not only achieves the lowest average NMAE but also highlights its superiority over all other models, including TCN, with statistical significance. This finding is visualized in the RMSE-Based Performance Space, where TSFeatures and TCN are isolated in the ideal bottom-left corner. Their position means that they exist in a separate, superior tier of performance, having successfully escaped the accuracy-consistency trade-off that the other baseline models possess, which form a diagonal line of increasing error and volatility.

While both TSFeatures and TCN demonstrate top-tier performance, the Consistency Score plot reveals a unique advantage of the proposed framework. It shows TSFeatures occupying a highly desirable spot in the top-right quadrant, indicating great consistency across both MASE and NMAE metrics. In contrast, nearly all baseline models, including the strong TCN contender, exhibit a relative weakness in MASE consistency. This highlights a key advantage of the framework, which is its reliability is robust

and balanced, making it less sensitive to the specific error metric being used for evaluation and therefore more generally dependable.

Furthermore, the analysis of per-series wins provides a critical perspective on dominance. The results in the Table 5.4 show that top-tier performance is concentrated in just two models, which are TSFeatures and TCN. The TSFeatures framework asserts itself as the clear leader, securing the single best score more than 500 times for both MASE and NMAE, having this high win count demonstrate its superior ability to frequently produce the optimal forecast. This analysis also highlights a key difference between being consistently competitive and being frequently optimal. Models like LGBM, SES, and LR have almost no wins, suggesting that while they may generalize relatively well, they are rarely the best choice for any specific task. This is in contrast to the TSFeatures framework, which is proven to be not only a top performer on average but also the most frequent winner in head-to-head comparisons.

In summary, the evidence paints the clear picture that the proposed framework, TSFeatures, operates in a superior performance tier, delivering statistically significant accuracy and exceptional reliability. It distinguishes itself from its strongest competitor, TCN, through a more balanced consistency profile and, most importantly, by its proven ability to be the single best performing model more frequently than any other candidate.

## 5.3 Ablation Studies

The previous sections of this work focused on presenting the results that directly answer the main research questions. This section explores other important aspects of the framework that, while beyond the scope of those questions, are relevant for a complete understanding of its behavior. Through a series of component analyses, often called ablation studies, specific parts of the proposed system will be looked at to better understand their individual impact and to support the final design choices.

To do this, the investigation will focus on three key areas. Firstly, the impact of the meta-feature set will be analysed, comparing the TSFeatures and Catch22 libraries to see which one provides better information for the metalearner. Secondly, the metalearner's performance will be evaluated by comparing the Random Forest and LGBM models. Although the metalearner itself is not the main focus of this work, it is still important to choose the better performing model so that the choice does not limit the framework's overall performance. Finally, the analysis will be extended to look at the performance of the proposed framework specifically on time series with a quarterly frequency, which was not explored in the main results.

Through these further analysis and investigation, this section aims to provide an understanding of the framework's behavior and confirm that the final design choices are effective and well justified.

### 5.3.1 Meta-feature Evaluation

This section of the ablation study focuses on determining the most effective set of meta-features for the proposed framework. The quality of the features used to describe the time series is determining, as it directly influences the metalearner's ability to make accurate and reliable model selections. The performance of the two feature sets used in the research is compared, which are the TSFeatures and the Catch22 libraries.

To perform this comparison, the results presented in the first research question's analysis will be revisited. While those plots and tables also include the non-decomposed approaches, the focus in this specific analysis will be solely on the direct comparison between the TSFeatures and Catch22 frameworks to justify the selection of one as the reference for the experiments.

#### 5.3.1.1 Individual Analysis of Results

Each piece of evidence from the previous Section 5.1.1 will be analysed, focusing only on the comparison between the TSFeatures and Catch22 models.

**Quantitative Performance Metrics**

As shown previously in Table 5.1, a quantitative comparison reveals a clear trade-off. For the scaled and normalized metrics, TSFeatures demonstrates better performance. Its mean MASE is lower (2.73 vs. 3.50), and its standard deviation is significantly better (2.52 vs. 6.59), indicating much higher reliability. It also holds a consistent edge in NMAE. In contrast, Catch22 shows a slight advantage on the absolute error metric, achieving a lower mean, median, and standard deviation for RMSE.

**Statistical Significance Heatmap**

The heatmap previously presented in Figure 5.1 provides statistical validation for the NMAE metric. The direct comparison between TSFeatures and Catch22 shows a mean difference in favor of TSFeatures. With a p-value of 0.0271, which is below the 0.05 threshold, it is possible to conclude that the better NMAE performance of the TSFeatures set is statistically significant.

**Consistency Score Plot**

The Consistency Score plot, shown earlier in Figure 5.2, offers an important visual summary of model reliability. In the plot, the point representing TSFeatures is located significantly further to the right and slightly lower than the point for Catch22. This position in the more desirable top-right area confirms that the TSFeatures set leads to a much more stable and consistent performance profile, particularly regarding the MASE metric.

**RMSE-based Performance Space**

Finally, the RMSE-Based Performance Space in Figure 5.3 visualizes the performance on the absolute error metric. This plot confirms the findings from the table, showing both methods located in the ideal bottom-left quadrant. However, it visually clarifies that Catch22 holds a slight edge, as its point is positioned marginally lower and to the left, representing a slightly better combination of mean error and error volatility for RMSE.

#### 5.3.1.2 Analysis and Discussion

Synthesizing the findings from each analysis provides a clear basis for selecting the optimal feature set. The evidence consistently shows that the choice between TSFeatures and Catch22 involves a trade-off. Catch22 demonstrates a small but consistent advantage when performance is measured by the absolute error metric, RMSE.

However, TSFeatures shows a clear and, in some cases, dramatic superiority on the scaled and normalized metrics, respectively, MASE and NMAE. Its advantage is not just in the average error but, more critically, in the reliability of the forecasts. The drastic reduction in the standard deviation for MASE and its dominant position in the Consistency Score plot indicate that the TSFeatures library provides the metalearner with information that leads to a much more stable and trustworthy system.

For a general purpose forecasting framework designed to handle series of varying scales, performance on scaled metrics and overall system reliability are arguably more important than a small gain in absolute error. The significant improvement in consistency offered by the TSFeatures set makes it the more robust and desirable choice. Therefore, based on its superior and more stable performance across scaled and normalized metrics, TSFeatures is selected as the reference feature set for the analyses conducted in this thesis.

### 5.3.2 Metalearner Performance

In this section, an evaluation is performed to determine which is the best performing metalearner for the proposed framework. As outlined in the introduction to the ablation studies, this analysis compares two models, which are Light Gradient Boosting Machine (LGBM) and Random Forest (RF). The goal is to select the algorithm that provides the most accurate recommendations for the base-level models, ensuring the metalearner itself is as effective as possible.

The comparison is performed in two stages. Firstly, a meta-level analysis is conducted by observing the performance of both metalearners using 10-fold cross-validation. If this analysis is not conclusive, a base-level evaluation will be the following step, which examines the final time series forecasting errors produced by the entire system when using each metalearner. To ensure a fair comparison, both models were trained and tested on the exact same data, using the TSFeatures set, which was previously identified as the superior feature set.

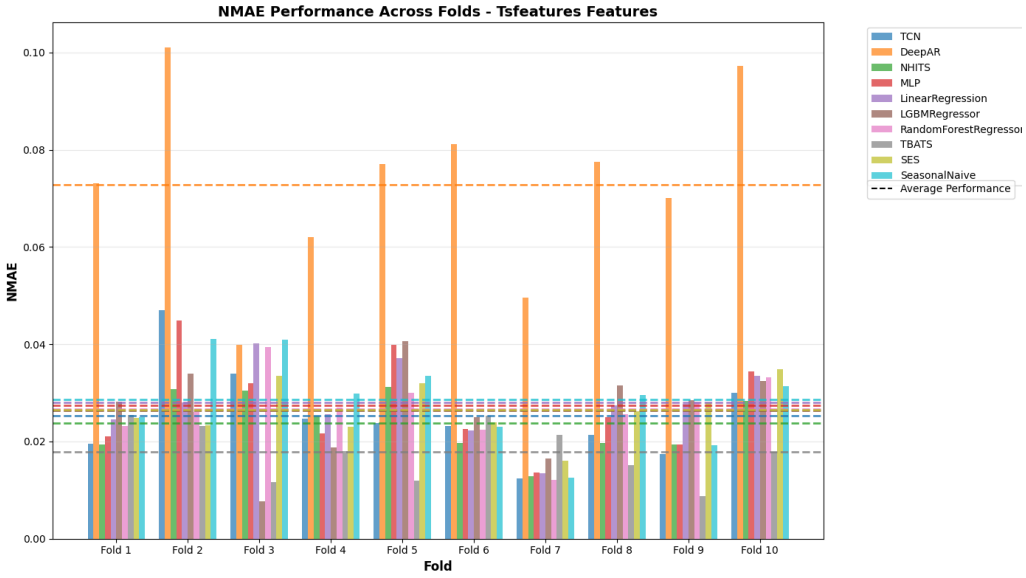
#### 5.3.2.1 Meta-level Evaluation

The meta-level analysis directly evaluates the metalearner's ability to accurately predict the performance of each individual forecasting model based on the characteristics of a time series. To get a reliable estimate of its performance, a 10-fold cross-validation process is used, as explained in the Section 4.2.1.

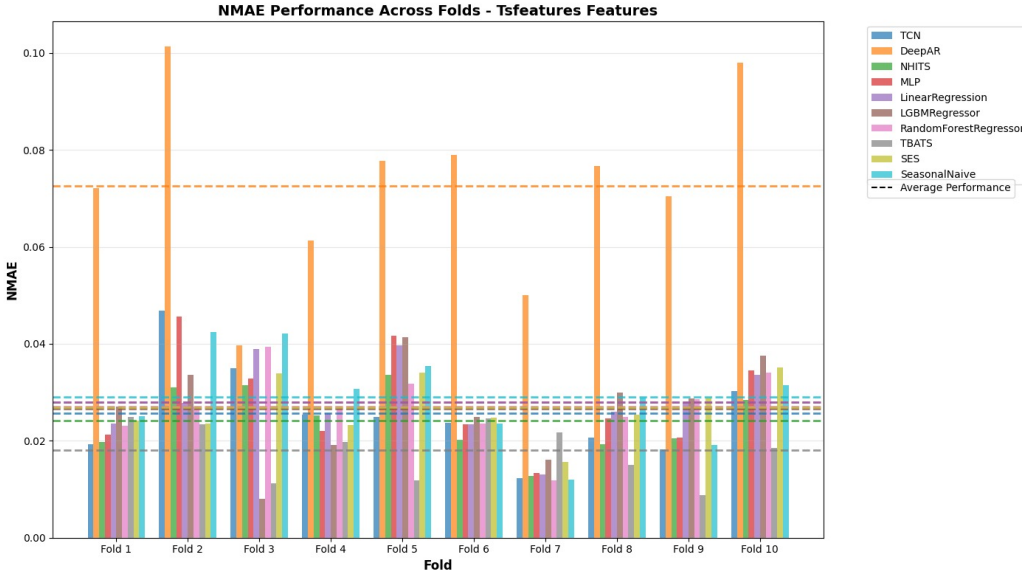
The plots in the Figure 5.7 allow the visualization of the cross-validation process, and they are particularly important not just for comparing overall averages, but also for understanding the consistency of the metalearner's performance. In these plots, the x-axis represents each of the 10 folds, while the y-axis shows the metalearner's prediction error. The colored bars within each fold show the error for predicting the performance of a specific base model. The colored dashed lines running horizontally show the average prediction error for that specific model across all 10 folds.

This visualization is particularly valuable because it allows for a direct comparison between the performance in each individual fold and the overall average performance, allowing the assessment of the stability of the metalearner. If the performance across all folds is consistent and close to the average, it suggests the model is robust and has generalized well to different subsets of the data. On the other hand, if any single fold shows a performance that is far from the average, it could indicate a sensitivity to the specific types of time series within that fold, highlighting a potential weakness or instability in the model's predictive ability.

The values for the NMAE metric are present in the Figures 5.7a and 5.7b, while the values for the RMSE metric are highlighted in the Figures A.2 and A.3.




(a) NMAE mean and across folds values of the LGBM based metalearner.




(b) NMAE mean and across folds values of the Random Forest based metalearner.

Figure 5.7: Comparison of NMAE for LGBM and Random Forest based metalearners.

When the plots for the LGBM and Random Forest metalearners are compared, it is possible to see that their performance profiles are highly similar. The patterns of the bars within each fold, as well as the positions of the average performance lines across folds, are very close in both sets of plots for NMAE and RMSE. Also, one conclusion that is possible to derive from both plots is that, while the bars in each fold may be further or closer from the average, no bar is extremely far from the average dashed lines in all the models, which is a good signal that indicates the metalearner generalized well to different subsets of the data. While there are minor variations, there is no clear or consistent visual evidence to suggest that one metalearner is definitively better than the other. Since this meta-level comparison is inconclusive, the base-level analysis must be the next move to make a final and more informed decision.

#### 5.3.2.2 Base-level Evaluation

The base-level analysis offers a different and more decisive evaluation, as it assesses the actual impact of the metalearner's choices on the final forecasting accuracy. The Table 5.3 presents the performance of the complete forecasting system, showing the final MASE, NMAE, and RMSE metrics when either LGBM or Random Forest are used as the metalearner. The table includes the mean, standard deviation (in the format mean ± std), and the median (md) values for each metric.

Table 5.3: Comparison of the metalearner models' performance metrics at base-level.

| Model | MASE | MASE md | NMAE | NMAE md | RMSE | RMSE md |
|---|---|---|---|---|---|---|
| LGBM | **2.73 ± 2.52** | **1.80** | **0.16 ± 0.20** | **0.09** | 1005.13 ± 1570.46 | 487.96 |
| RF | 3.50 ± 6.59 | 1.83 | 0.17 ± 0.21 | 0.10 | **993.64 ± 1503.74** | **477.93** |

A detailed analysis of this table reveals an advantage for the LGBM-based metalearner. The most significant difference is seen in the MASE metric. The LGBM framework achieves a lower mean error, and more importantly, its standard deviation is less than half that of the RF framework. This indicates that the system is considerably more reliable and consistent when guided by LGBM. LGBM also holds a consistent edge across all NMAE statistics, with a lower mean, lower standard deviation, and a lower median. In contrast, the RF model shows a slight advantage in the RMSE metric, achieving a lower mean, standard deviation, and median.

When considering the overall performance, the improvement in consistency provided by LGBM on the scaled MASE metric outweighs the small deficit in the absolute RMSE metric. A highly reliable and stable system is preferable, and the reduction in MASE volatility makes LGBM the best choice. Based on this base-level analysis, LGBM is the metalearner solution to be adopted in the final framework and serves as the reference metalearner for the analysis conducted in this thesis.

### 5.3.3 Quarterly analysis

This final part of the ablation studies investigates the performance of the proposed framework on a different data domain, which is the time series with a quarterly frequency. The previous sections have focused on results from other frequencies (the monthly frequency) where the framework has shown considerable success. The goal is to test the versatility and robustness of the approach when faced with data that has different characteristics, given that quarterly series are typically much shorter and have less defined patterns than their higher frequency counterparts.

It is important to position this analysis appropriately. While the initial goal was to build a comprehensive, multi-frequency forecasting framework, the results for the quarterly data revealed unique challenges, and the framework's performance did not meet the high standards set in previous evaluations. For this reason, this investigation is included within the ablation studies. It serves to transparently explore the current limitations of the framework and to identify specific areas where further development is needed. This analysis is valuable as it helps define a clear path for future work towards the end goal of a developing a universal and solid forecasting system.

#### 5.3.3.1 Presentation of results

To evaluate the performance on quarterly data, the same set of analytical tools used in the previous research question's evaluation was employed. For detailed explanations of how to interpret each visualization and table, please refer to the descriptions in Section 5.1.1. In the following subsections, the results from each of these analyses will be presented, accompanied by a preliminary discussion of the key insights drawn from each one.

**Statistical Significance Heatmap**

The heatmap in Figure 5.8 provides a pairwise statistical comparison of all models on the NMAE metric, using the Wilcoxon signed-rank test to determine if performance differences are significant. The results in the figure constitute a resumed version of the full one, which is A.4.

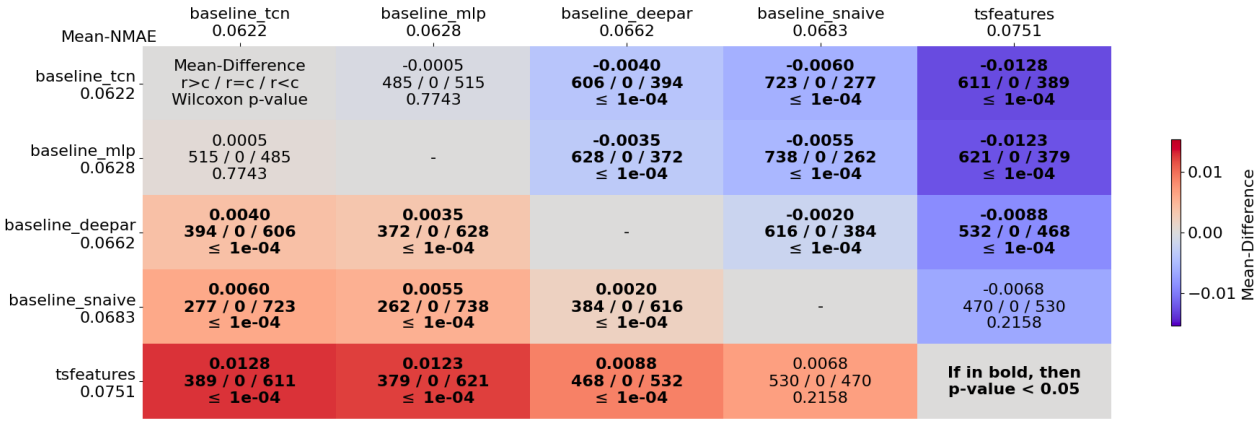



Figure 5.8: Heatmap of pairwise NMAE comparison with Wilcoxon signed-rank test of TSFeatures and some of the baseline models.

The results from the heatmap clearly indicate that the TSFeatures framework struggles on the quarterly dataset. The cells in the TSFeatures row and column are, respectively, predominantly red and blue, meaning that its mean NMAE is higher than almost all other models. The p-values confirm that this underperformance is statistically significant. In contrast, models such as TCN, MLP, and DeepAR emerge as the top performers, with their columns showing mostly red cells, indicating significantly lower NMAE compared to their competitors. This provides the first piece of statistical evidence that the framework's effectiveness is reduced on this data frequency.

**Consistency Score Plot**

The Figure 5.9 visualizes the trade-off between MASE and NMAE consistency. A position in the top-right corner is ideal, representing high reliability.

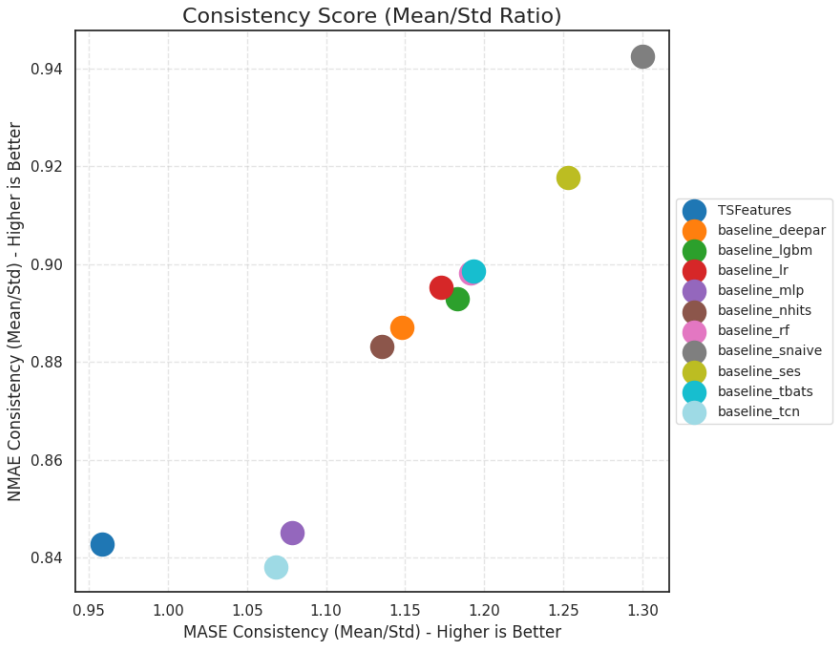



Figure 5.9: Consistency Score plot of all the methods of the quarterly frequency.

The consistency plot reveals a reversal of the trends observed in the previous sections' analyses. The TSFeatures framework is positioned in the bottom-left quadrant, which indicates poor consistency on both MASE and NMAE, suggesting its performance on quarterly data is not only less accurate but also highly erratic. On the other hand, other models, particularly the simpler statistical methods like Seasonal Naive and Simple Exponential Smoothing, are placed the ideal top-right region, demonstrating a much more stable and reliable performance profile on this specific dataset.

**RMSE-based Performance Space**

The relationship between absolute error (mean RMSE) and its volatility (standard deviation) is shown in the performance space plot in Figure 5.10. The optimal position in this plot is the bottom-left corner.

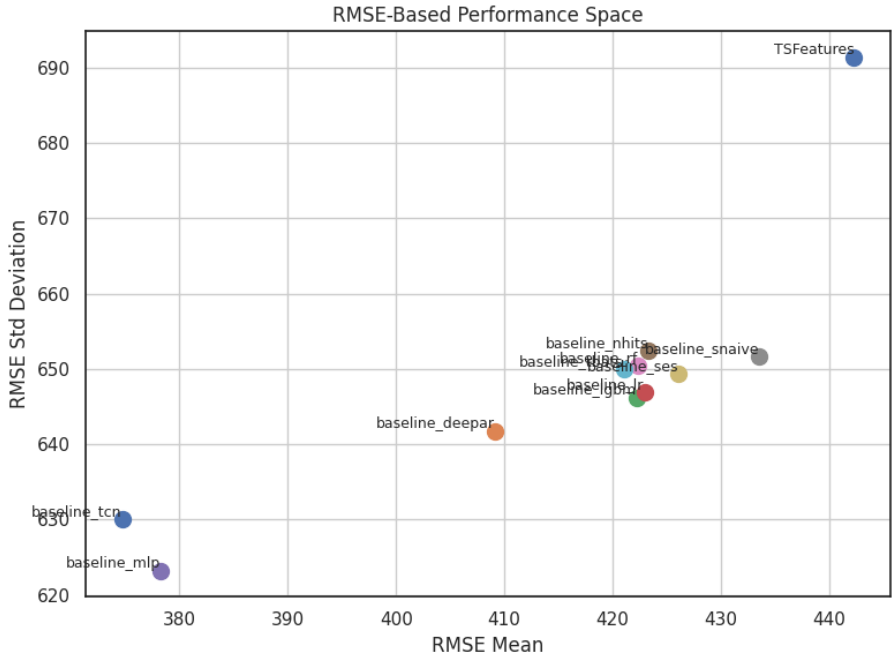



Figure 5.10: RMSE-based Performance Space plot of all the methods of the quarterly frequency.

This plot reinforces the narrative of the framework's poor performance on quarterly data. The TSFeatures model is located in the top-right corner, representing the worst-case scenario of having both a high average error and high performance volatility. This contrasts with models such as TCN and MLP, which are positioned in the ideal bottom-left corner, establishing them as a vastly superior performance tier in terms of both accuracy and consistency for the RMSE metric.

**Per-series Performance Analysis**

The Table 5.4 details the number of times each model achieved the single lowest error, changing the focus from average performance to dominance on individual series.

Table 5.4: Number of per-series wins (lowest error) per model for MASE, NMAE, and RMSE metrics for the quarterly frequency.

| Model | MASE Wins | NMAE Wins | RMSE Wins |
|---|---|---|---|
| TSFeatures | 389 | **289** | **301** |
| Baseline_DeepAR | 0 | 70 | 62 |
| Baseline_LGBM | 0 | 24 | 19 |
| Baseline_LR | 0 | 19 | 24 |
| Baseline_MLP | 0 | 170 | 167 |
| Baseline_NHITS | 0 | 45 | 45 |
| Baseline_RF | 0 | 29 | 28 |
| Baseline_sNaive | 0 | 50 | 42 |
| Baseline_SES | 0 | 9 | 9 |
| Baseline_TBATS | 0 | 45 | 40 |
| Baseline_TCN | **611** | 250 | 263 |

The win count analysis reveals a complex but telling story. The most striking result is the dominance of TCN on the MASE metric, where it achieved 611 wins, surpassing by far the 389 wins from

TSFeatures. This indicates that for a majority of quarterly series, a single deep learning model was the best choice. While the TSFeatures framework still managed to secure the most wins for NMAE and RMSE, its lead is substantially smaller than in previous results. Furthermore, other models like MLP and DeepAR became highly competitive, securing a large number of wins themselves. This suggests that the metalearner's ability to consistently identify the best model was significantly lower for this data frequency.

#### 5.3.3.2 Analysis and Discussion

The results from the quarterly data analysis present a evident contrast to the framework's performance on other data frequencies. The evidence across all plots suggests that the TSFeatures framework struggles to maintain its competitive edge, and in many respects, underperforms compared to several individual baseline models.

The RMSE-Based Performance Space and the Consistency Score plot are particularly revealing. In the RMSE plot, the TSFeatures framework is positioned in the top-right quadrant, indicating both high average error and high performance volatility, which is the least desirable outcome. This is corroborated by the Consistency Score plot, where it is found in the bottom-left, meaning poor consistency on both MASE and NMAE. The statistical significance heatmap confirms these visual findings, showing that the framework's NMAE is significantly worse than that of top performing models for this dataset, such as TCN and MLP.

The Per-Series Wins table provides the most nuanced insight into this performance shift. For the MASE metric, TCN emerges as the most dominant model with 611 wins, surpassing the 389 wins of the TSFeatures framework. This suggests that for a large portion of quarterly series, a single deep learning model was a much more effective choice. The TSFeatures framework's poor position in both the RMSE plot (top-right) and the Consistency plot (bottom-left) explains its reduced ability to win. A model that is, on average, unreliable and inaccurate is much less likely to be the top performer on a regular basis. While the framework still secured the most wins for NMAE and RMSE, this result, when paired with its poor average scores, suggests its performance on quarterly data is highly volatile, performing either very well or very poorly on any given series. This indicates that the metalearner's ability to select the best model was severely held back for this data type.

There are two likely reasons for this pronounced drop in performance. Firstly, the nature of quarterly data itself poses a challenge. With far fewer data points per series, the statistical patterns are less clear, which can make feature extraction less effective. Secondly, and as a consequence, the meta-feature representation generated may not be as discriminative for these shorter series. If the features do not clearly distinguish between series that benefit from different models, the metalearner is essentially working with poor quality information, leading to suboptimal model selections.

This analysis highlights a current limitation of the framework. To achieve the goal of a robust, multi-frequency system, more development time would be needed. Future work should focus on engineering or selecting meta-features designed for short, low-frequency time series, which would likely restore the metalearner's decisive advantage.

# Chapter 6

# Conclusion

This final chapter brings the work presented in this thesis to its conclusion. The purpose is to consolidate the entire research made, summarizing the key findings from the experimental evaluations and reflecting on their implications. The research questions that have guided this study will be revisited, offering a summary of the findings and the definitive answers derived from them. Following this, the main contributions of this work will be addressed, followed by the coverage of the limitations of the current research. Finally, building upon the insights and limitations identified, the chapter will be concluded by proposing several promising directions that could guide future work in this area.

## 6.1 Research Questions

This section serves to answer the research questions that were established at the beginning of this thesis and that guided the whole work. The following answers are the culmination of the experimental results and analyses presented in the previous Chapter, providing a summary of the key findings of this work.

**RQ1:** *What is the advantage of leveraging the metalearner for choosing between individual models and decomposition-based methods for the formation of forecasting combinations?*

The advantage is an increase in the forecasting system's performance. By adopting a decomposition-based strategy, the metalearner enhances the system from a state of high error and high volatility to one of lower error and higher reliability. The key advantages are increased accuracy and robustness, given that the system effectively mitigates the risk of large forecast errors by applying a more sophisticated decomposition approach to series that require it leading to lower average errors across all metrics, and dramatically improved reliability, due to a significant reduction in the standard deviation of errors meaning that the performance is dependable and predictable.

In essence, the results attest that the metalearning approach, when paired with decomposition, produces substantially better results than its non-decomposition counterpart. This finding holds true across both of the chosen feature sets, as the TSFeatures and Catch22 methods consistently outperformed their respective non-decomposed versions. This successfully answers the research question, providing clear evidence that integrating selective decomposition within a metalearning framework is an effective strategy to improve the accuracy and reliability of the final forecasts when compared to a metalearning approach that does not leverage decomposition.

**RQ2:** *What is the advantage of leveraging this approach in comparison with individual models?*

The main advantage of leveraging the proposed metalearning and decomposition-based approach (TS-Features) is that it delivers a forecasting system that is not only more accurate, but also and more importantly, is more reliable and robust than the alternative of relying on any single one of the individual models evaluated in this study. This advantage is not marginal but constitutes a significant leap into a higher tier of performance.

The findings have demonstrated this superiority across three key dimensions. First, the framework's performance is statistically superior to all benchmarked individual models. Second, it possesses a unique and balanced consistency, achieving low error and low volatility at the same time. Finally, the analysis of per-series wins shows that the framework is frequently the optimal choice, winning the most individual forecasting matchups by a large margin.

In essence, the advantage is the move from a static and not much flexible modeling choice to a dynamic and adaptive strategy. This is demonstrated by the fact that the framework's overall performance surpassed that of every individual baseline model it has at its disposal, highlighting its effectiveness at making the best out of a given set of models. By intelligently selecting its approach, the framework mitigates the risk of poor model-to-problem fit, resulting in a system that is more accurate, more consistent, and more often the single best solution for the task at hand.

## 6.2 Main Contributions

This section covers the main contributions of this thesis to the field of time series forecasting. Derived from the design, implementation and rigorous evaluation of the proposed metalearning framework, these contributions offer both practical solutions and valuable insights for researchers. Each key contribution of this work is detailed in the following paragraphs.

Firstly, this thesis contributes with an extensive literature review that portraits the current landscape of automated time series forecasting, with a focus on metalearning and decomposition-based methods. This review process served a purpose beyond summarizing existing knowledge, as it allowed for the identification of gaps in the field. While many systems use metalearning for model selection, and others apply decomposition as a standard preprocessing step, there was a lack of research into frameworks that dynamically and intelligently choose whether to apply decomposition based on the characteristics of the series. This identified void directly motivated the research questions and guided the design of the framework proposed in this work.

Building on this identified gap, a second major contribution is the empirical validation of a selective, decomposition-based strategy within a metalearning context. The research provides strong evidence that allowing a framework to dynamically decide whether to decompose a time series based on its characteristics leads to performance gains. As demonstrated in the first research question's analysis, this approach enhances the reliability and robustness of the forecasts, proving that a hybrid strategy is a powerful and effective alternative to one-size-fits-all methods.

Furthermore, this thesis contributes with a complete forecasting framework that is demonstrably better than its individual models. The analysis for the second research question showed that the framework, through its selection mechanism, achieved an overall performance that surpassed every single one of the

baseline models it had at its disposal. This highlights the significant value added by the metalearning layer itself, proving that the orchestration of existing models results in a system that is more accurate, more reliable, and more frequently the optimal choice than any single model used in isolation.

Finally, this work also puts forward a comprehensive methodology for the evaluation of complex forecasting systems. Throughout the analysis, a move was made beyond simple average error metrics and a combination of multiple perspectives was leveraged, including statistical significance tests, visualizations of the accuracy-reliability trade-off, per-series win analysis, and targeted ablation studies. This approach provides a more trustworthy understanding of model performance and offers a template that can be used for future research in the design and assessment of adaptive forecasting systems.

In summary, the contributions of this thesis are an extensive literature review that identified a key research gap, the empirical validation of selective decomposition, the delivery of a complete framework that outperforms all the baseline models taking the best out of the set of them, and the demonstration of a rigorous evaluation methodology. Together, these findings provide not only a high performing forecasting tool but also valuable insights that lay a foundation for future advancements in creating automated forecasting solutions.

## 6.3 Limitations

To provide a balanced view of this research, it is important to acknowledge the limitations of the current work, given that recognizing these boundaries helps to correctly position the findings and conclusions of this thesis, and also provides a clear foundation for identifying valuable directions for future research.

One limitation of this study is related to the scope of the data used for evaluation. While the framework was tested on a large and diverse set of time series, its performance on other types of data has not been verified. For instance, its effectiveness on very high-frequency financial data or intermittent demand data, which have different characteristics, remains an open question, given that only the monthly and the quarterly frequencies were studied. In an initial phase, working with the weekly frequency was attempted, but the data available was too scarce to draw meaningful verdicts from it. Therefore, the conclusions drawn about the framework's strong performance are currently best applied to the types of data on which it was tested.

As was demonstrated in the ablation studies, another limitation of the current framework is its suboptimal performance on quarterly time series. The analysis revealed that the chosen meta-feature representation was less effective for these shorter, low-frequency series, leading to poor model selection by the metalearner and worse overall forecasting accuracy. This finding indicates that the framework in its current form is not yet a universal solution for all data frequencies and that further development is needed for it to perform well on this type of data.

Finally, the framework's performance is naturally limited by the set of individual models included in its selection pool. The metalearner can only choose the best model from the options it has been given, as it does not have a mechanism to create new model architectures or fine-tune the hyperparameters of the existing ones. Fine-tuning of the hyperparameters is performed in the forecasting phase of the framework, but the parameters are not chosen by the metalearner, only the model to be used. This means the framework's maximum potential performance on any given series is highly limited by the

best performing model within its predefined suite. Experimentation with a different set of models would surely yield different forecasting results.

In summary, the main limitations identified are related to the generalizability of the results to new data types, the specific underperformance on quarterly data, and the constraints of a fixed suite of models. These limitations do not weaken the findings of this thesis, but they do help to define the context in which those findings are strongest. They also provide a clear and valuable roadmap for the future research directions discussed in the following section.

## 6.4 Future Work

Building on the findings and the limitations identified in the previous section, this part of the chapter covers and proposes several promising directions that could guide future work, extending the foundation established by this thesis:

- **Improving Low-Frequency Data Performance:** Enhancing the framework's effectiveness on shorter time series (e.g., quarterly data) by researching and integrating specialized meta-features designed to capture their unique patterns.
- **Expanding Decomposition Techniques:** Incorporating more advanced decomposition methods, such as wavelet analysis or deep learning-based approaches, to provide the metalearner with a more diverse portfolio of strategies.
- **Enriching the Model Portfolio:** Expanding the set of base forecasting models available to the metalearner. Integrating other popular or state-of-the-art algorithms would increase the framework's versatility and its potential to find an even better suited model for any given series.
- **Data Augmentation for Robustness:** Employing generative models to create synthetic time series. This would augment the training data, increasing its diversity and improving the metalearner's robustness, especially for less common time series patterns.
- **Integrated Hyperparameter Optimization:** Advancing the framework by having the metalearner predict the optimal hyperparameters for a given model and series directly. This would replace the separate hyperparameters optimization step and create a more intelligent, unified modeling pipeline.
- **Scaling the Training Dataset:** Evaluating the framework on the full time series dataset, which was subsetted for this thesis due to computational constraints. This would enable a detailed analysis of the performance and computational cost trade-off at different data scales.
- **Generalization to New Domains:** Assessing the framework's versatility by applying it to new and diverse data domains. This would not only test its current robustness but also drive the development of new, domain-specific features.

In conclusion, the directions outlined above represent several of the many potential paths for extending this research. The current work has built and validated a strong foundation for an adaptive forecasting framework. These future steps offer great opportunities to make this system even more powerful, versatile, and automated.

# References


[1] R. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. OTexts, 2018.

[2] X. Kong, Z. Chen, W. Liu, *et al.*, "Deep learning for time series forecasting: A survey", *Int. J. Mach. Learn. Cyber.*, 2025.

[3] M. Santos, A. d. Carvalho, and C. Soares, "Metafore: Algorithm selection for decomposition-based forecasting combinations", *International Journal of Data Science and Analytics*, pp. 1–14, 2024.

[4] G. D. Silvestre, M. R. dos Santos, and A. C. de Carvalho, "Seasonal-trend decomposition based on loess+ machine learning: Hybrid forecasting for monthly univariate time series", in *2021 international joint conference on neural networks (IJCNN)*, IEEE, 2021, pp. 1–7.

[5] S. Baratsas, F. Iseri, and E. N. Pistikopoulos, "A hybrid statistical and machine learning based forecasting framework for the energy sector", *Computers & Chemical Engineering*, vol. 188, p. 108 740, 2024.

[6] M. AlShafeey and C. Csaki, "Adaptive machine learning for forecasting in wind energy: A dynamic, multi-algorithmic approach for short and long-term predictions", *Heliyon*, vol. 10, no. 15, 2024.

[7] A. Bauer, M. Züfle, J. Grohmann, N. Schmitt, N. Herbst, and S. Kounev, "An automated forecasting framework based on method recommendation for seasonal time series", in *Proceedings of the ACM/SPEC International Conference on Performance Engineering*, 2020, pp. 48–55.

[8] E. Vaiciukynas, P. Danenas, V. Kontrimas, and R. Butleris, "Two-step meta-learning for time-series forecasting ensemble", *IEEE Access*, vol. 9, pp. 62 687–62 696, 2021.

[9] V. Freitas, *Parsifal*, `https://parsif.al/`, Accessed: 2024-11-06.

[10] Y. Zhang, J. Zhang, L. Yu, Z. Pan, C. Feng, Y. Sun, and F. Wang, "A short-term wind energy hybrid optimal prediction system with denoising and novel error correction technique", *Energy*, 2022.

[11] Y. Hao, X. Wang, J. Wang, and W. Yang, "A new perspective of wind speed forecasting: Multi-objective and model selection-based ensemble interval-valued wind speed forecasting system", *Energy Conversion and Management*, vol. 299, p. 117 868, 2024.

[12] K. U. Jaseena and B. C. Kovoor, "Decomposition-based hybrid wind speed forecasting model using deep bidirectional lstm networks", *Energy Conversion and Management*, vol. 234, p. 113 944, 2021.

[13] J. Wang, L. Zhang, Z. Liu, and X. Niu, "A novel decomposition-ensemble forecasting system for dynamic dispatching of smart grid with sub-model selection and intelligent optimization", *Expert Systems with Applications*, 2022.

[14] J. Wang, Y. Wang, H. Li, H. Yang, and Z. Li, "Ensemble forecasting system based on decomposition-selection-optimization for point and interval carbon price prediction", *Applied Mathematical Modelling*, 2023.

[15] X. Li, Y. Zhang, L. Chen, J. Li, and X. Chu, "A decomposition-ensemble-integration framework for carbon price forecasting", *Expert Systems with Applications*, vol. 257, p. 124 954, 2024.

[16] E. Ahn and J. Hur, "A short-term forecasting of wind power outputs using the enhanced wavelet transform and arimax techniques", *Renewable Energy*, vol. 212, pp. 394–402, 2023.

[17] Y. Yin, I. Ahmadianfar, F. K. Karim, and H. Elmannai, "Advanced forecasting of covid-19 epidemic: Leveraging ensemble models, advanced optimization, and decomposition techniques", *Computers in Biology and Medicine*, 2024.

[18] Z. Gao, X. Yin, F. Zhao, H. Meng, Y. Hao, and M. Yu, "A two-layer ssa-xgboost-mlr continuous multi-day peak load forecasting method based on hybrid aggregated two-phase decomposition", *Energy Reports*, vol. 8, pp. 12 426–12 441, 2022.

[19] Y. Zhang, Z. Pan, H. Wang, J. Wang, Z. Zhao, and F. Wang, "Achieving wind power and photovoltaic power prediction: An intelligent prediction system based on a deep learning approach", *Energy*, 2023.

[20] Q. Lin, H. Cai, H. Liu, X. Li, and H. Xiao, "A novel ultra-short-term wind power prediction model jointly driven by multiple algorithm optimization and adaptive selection", *Energy*, vol. 288, p. 129 724, 2024.

[21] Q. Zhou, C. Wang, and G. Zhang, "A combined forecasting system based on modified multi-objective optimization and sub-model selection strategy for short-term wind speed", *Applied Soft Computing*, vol. 94, p. 106 463, 2020.

[22] C. Wang, S. Zhang, P. Liao, and T. Fu, "Wind speed forecasting based on hybrid model with model selection and wind energy conversion", *Renewable Energy*, vol. 196, pp. 763–781, 2022.

[23] T. K. Ho, "Random decision forests", in *Proceedings of 3rd international conference on document analysis and recognition*, IEEE, vol. 1, 1995, pp. 278–282.

[24] S. Shahoud, H. Khalloof, M. Winter, C. Duepmeier, and V. Hagenmeyer, "A meta learning approach for automating model selection in big data environments using microservice and container virtualization technologies", in *Proceedings of the 12th International Conference on Management of Digital EcoSystems*, 2020, pp. 84–91.

[25] D. Zhang, S. Chen, L. Liwen, and Q. Xia, "Forecasting agricultural commodity prices using model selection framework with time series features and forecast horizons", *IEEE Access*, 2020.

[26] E. Sayed, M. Maher, O. Sedeek, A. Eldamaty, A. Kamel, and R. El Shawi, "Gizaml: A collaborative meta-learning based framework using llm for automated time-series forecasting.", in *EDBT*, 2024, pp. 830–833.

[27] H. Engbers and M. Freitag, "Automated model selection for multivariate anomaly detection in manufacturing systems", *Journal of Intelligent Manufacturing*, pp. 1–19, 2024.

[28] S. Shahoud, H. Khalloof, C. Duepmeier, and V. Hagenmeyer, "Incorporating unsupervised deep learning into meta learning for energy time series forecasting", in *Proceedings of the Future Technologies Conference (FTC) 2020, Volume 1*, Springer, 2021, pp. 326–345.

[29] M. R. Santos, L. R. Mundim, and A. C. Carvalho, "Evaluation of error metrics for meta-learning label definition in the forecasting task", in *International Conference on Hybrid Artificial Intelligence Systems*, Springer, 2020, pp. 397–409.

[30] X. Wang, K. Smith-Miles, and R. Hyndman, "Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series", *Neurocomputing*, vol. 72, no. 10-12, pp. 2581–2594, 2009.

[31] F. Bahrpeyma, V. M. Ngo, M. Roantree, and A. McCarren, "A meta-learner approach to multistep-ahead time series prediction", *International Journal of Data Science and Analytics*, 2024.

[32] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. C. H. Hoi, "Deeptime: Deep time-index meta-learning for non-stationary time-series forecasting", *CoRR*, vol. abs/2207.06046, 2022. arXiv: 2207.06046.

[33] G. M. J. George E. P. Box, "Time series analysis: Forecasting and control", *Holden-Day*, 1970.

[34] W. Jiang, L. Ling, D. Zhang, R. Lin, and L. Zeng, "A time series forecasting model selection framework using cnn and data augmentation for small sample data", *Neural processing letters*, vol. 55, no. 5, pp. 5783–5810, 2023.

[35] T. L. van Zyl, "Late meta-learning fusion using representation learning for time series forecasting", in *26th International Conference on Information Fusion, FUSION 2023, Charleston, SC, USA, June 27-30, 2023*, IEEE, 2023, pp. 1–8.

[36] J. Gastinger, S. Nicolas, D. Stepić, M. Schmidt, and A. Schülke, "A study on ensemble learning for time series forecasting and the need for meta-learning", in *2021 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2021, pp. 1–8.

[37] A. Saadallah, "Online explainable model selection for time series forecasting", in *2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA)*, IEEE, 2023, pp. 1–10.

[38] R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose, "A state space framework for automatic forecasting using exponential smoothing methods", *International Journal of Forecasting*, vol. 18, no. 3, pp. 439–454, 2002.

[39] R. J. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder, *Forecasting with Exponential Smoothing: The State Space Approach*. Springer, 2008.

[40] A. De Livera, R. Hyndman, and R. Snyder, "Forecasting time series with complex seasonal patterns using exponential smoothing", *Journal of the American Statistical Association*, vol. 106, pp. 1513–1527, 2010.

[41] S. Taghiyeh, D. C. Lengacher, and R. B. Handfield, "Forecasting model selection using intermediate classification: Application to monarchfx corporation", *Expert Systems with Applications*, vol. 151, p. 113 371, 2020.

[42] M. Abdallah, R. Rossi, K. Mahadik, S. Kim, H. Zhao, and S. Bagchi, "Autoforecast: Automatic time-series forecasting model selection", in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 5–14.

[43] R. Fischer and A. Saadallah, "Autoxpcr: Automated multi-objective model selection for time series forecasting", in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 806–815.

[44] Q. Zhou, C. Wang, and G. Zhang, "Hybrid forecasting system based on an optimal model selection strategy for different wind speed forecasting problems", *Applied Energy*, 2019.

[45] Y. Yao, D. Li, H. Jie, L. Chen, T. Li, J. Chen, J. Wang, F. Li, and Y. Gao, "Simplets: An efficient and universal model selection framework for time series forecasting", *Proceedings of the VLDB Endowment*, vol. 16, no. 12, pp. 3741–3753, 2023.

[46] C. H. Valencia, M. M. Vellasco, and K. Figueiredo, "Echo state networks: Novel reservoir selection and hyperparameter optimization model for time series forecasting", *Neurocomputing*, vol. 545, p. 126 317, 2023.

[47] A. Jati, V. Ekambaram, S. Pal, B. Quanz, W. M. Gifford, P. Harsha, S. Siegel, S. Mukherjee, and C. Narayanaswami, "Hierarchical proxy modeling for improved hpo in time series forecasting", in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 891–900.

[48] M. Maya, W. Yu, and X. Li, "Time series forecasting with missing data using neural network and meta-transfer learning", in *IEEE Symposium Series on Computational Intelligence, SSCI 2021, Orlando, FL, USA, December 5-7, 2021*, IEEE, 2021, pp. 1–6.

[49] V. Cerqueira, L. Torgo, and C. Soares, "Model selection for time series forecasting an empirical analysis of multiple estimators", *Neural processing letters*, vol. 55, no. 7, pp. 10 073–10 091, 2023.

[50] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The m4 competition: 100,000 time series and 61 forecasting methods", *International Journal of Forecasting*, vol. 36, no. 1, pp. 54–74, 2020.

[51] M. R. d. S. Gabriel Dalforno Silvestre and A. C. de Carvalho, *Identifying seasonality in time series by applying fast fourier transform*, Available at https://ieeexplore.ieee.org/abstract/document/9074776, 2021.

[52] C. H. Lubba, S. S. Sethi, P. Knaute, S. R. Schultz, B. D. Fulcher, and N. S. Jones, "Catch22: Canonical time-series characteristics: Selected through highly comparative time-series analysis", *Data mining and knowledge discovery*, vol. 33, no. 6, pp. 1821–1852, 2019.

[53] C. et al., *mlforecast: Scalable machine learning based time series forecasting*, Available at https://github.com/Nixtla/mlforecast, 2022.

[54] G. et al., *NeuralForecast: User friendly state-of-the-art neural forecasting models.* Available at https://github.com/Nixtla/neuralforecast, 2022.

[55] A. Ismail-Fawaz, A. Dempster, C. W. Tan, M. Herrmann, L. Miller, D. F. Schmidt, S. Berretti, J. Weber, M. Devanne, G. Forestier, *et al.*, "An approach to multiple comparison benchmark evaluations that is stable under manipulation of the comparate set", *arXiv preprint arXiv:2305.11921*, 2023.

[56] O. Borchert, D. Salinas, V. Flunkert, T. Januschowski, and S. Günnemann, "Multi-objective model selection for time series forecasting", *arXiv preprint arXiv:2202.08485*, 2022.

[57] M. Jakobs and A. Saadallah, "Explainable adaptive tree-based model selection for time-series forecasting", in *2023 IEEE International Conference on Data Mining (ICDM)*, IEEE, 2023, pp. 180–189.

[58] Z. Dong, R. Jiang, H. Gao, H. Liu, J. Deng, Q. Wen, and X. Song, "Heterogeneity-informed meta-parameter learning for spatiotemporal time series forecasting", in *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, 2024, pp. 631–641.

[59] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "Meta-learning framework with applications to zero-shot time-series forecasting", in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, 2021, pp. 9242–9250.

[60] S. Barak, M. Nasiri, and M. Rostamzadeh, "Time series model selection with a meta-learning approach; evidence from a pool of forecasting algorithms", *arXiv preprint arXiv:1908.08489*, 2019.

# Appendix A

## A.1 Summary of articles explored in the Literature Review phase

Table A.1: Research purpose description for the studies published since 2019.

| Reference | Research Purpose | Approach/Domain |
|---|---|---|
| [56] | To address the multi-objective nature of model selection in time series forecasting, considering not only accuracy but also other criteria like training time and latency, and to propose a method, PARETOSELECT, for learning default models in this multi-objective setting. | Time series forecasting, multi-objective optimization, model selection, benchmarking classical and deep learning methods, Pareto front analysis, PARETOSELECT algorithm. |
| [37] | To develop an online, adaptive, and explainable model selection method for time series forecasting that leverages the concept of Regions of Competence (RoCs) to handle the time-evolving nature of time series data and provide model-agnostic explanations for model choices. | Time series forecasting, online model selection, adaptive clustering, Regions of Competence (RoCs), concept drift detection, model-agnostic explanations, empirical evaluation on real-world datasets. |
| [46] | To propose a novel model, ESN-GA-SRG, that optimizes hyperparameters and topology selection for Echo State Networks (ESNs) in multi-step time series forecasting, using a Genetic Algorithm and Separation Ratio Graph (SRG) to improve forecasting accuracy. | Time series forecasting, computational intelligence, Echo State Networks (ESNs), hyperparameter optimization, topology selection, Genetic Algorithm, Separation Ratio Graph (SRG), empirical evaluation on time series benchmarks. |
| [57] | To develop a novel online model selection method for tree-based time series forecasting that leverages the TreeSHAP explainability method to address overfitting, adapt to concept drift, and provide multi-level explanations for model selection and predictions. | Time series forecasting, tree-based models, online model selection, TreeSHAP explainability, concept drift detection, input importance, model output explanation, empirical evaluation on real-world datasets. |

| Reference | Research Purpose | Approach/Domain |
|---|---|---|
| [4] | To propose a hybrid forecasting framework that combines STL decomposition with machine learning and seasonal naive forecasting to improve time series prediction accuracy across various domains. | Hybrid forecasting, time series decomposition (STL), machine learning, seasonal naive forecasting. |
| [3] | To propose a novel time series forecasting approach, METAFORE, combining STL decomposition and metalearning to recommend optimal algorithm combinations for improved predictive performance. | Time series forecasting, metalearning, STL decomposition, algorithm recommendation. |
| [27] | To develop a meta-learning-based model selection approach for multivariate anomaly detection in manufacturing systems to automate the selection of optimal algorithms and hyperparameters. | Multivariate anomaly detection, meta-learning, model selection, manufacturing systems. |
| [21] | To develop a novel combined forecasting system for wind speed time series of a wind farm by combining SSAWD denoising with sub-model selection and multi-objective optimization to improve forecasting accuracy. | Wind speed forecasting, time series forecasting, artificial neural networks, SSAWD denoising, sub-model selection, multi-objective optimization. |
| [24] | To develop a generic microservice-based meta-learning framework for Big Data environments, specifically focusing on time series model selection, to automate the process of selecting suitable machine learning algorithms. | Meta-learning, microservices, Big Data, time series forecasting, model selection. |
| [31] | To develop a meta-learning approach for multi-step ahead time series prediction by leveraging time series metrics as meta-features to predict the performance of various machine learning models. | Time series forecasting, meta-learning, machine learning, model selection, feature engineering. |
| [29] | To investigate the impact of different error metrics on the performance of meta-learning for time series forecasting. | Meta-learning, time series forecasting, error metrics, machine learning. |
| [28] | To develop an Energy Meta Learning System to recommend suitable short-term load forecasting models for buildings, using Descriptive Statistics Time-based Meta Features and an encoded representation of meta-features. | Meta-learning, time series forecasting, model selection, feature engineering, deep learning, autoencoders. |
| [15] | To develop a novel framework, DecEnsInt, for accurate carbon price forecasting by using CEEMDAN decomposition and an ensemble of models tailored to different frequency domains, optimized using SLSQP. | Carbon price forecasting, time series forecasting, ensemble learning, time series decomposition. |
| [5] | To develop the Energy Price Index (EPIC) framework, a comprehensive system for forecasting energy prices of various products using a combination of statistical and machine learning methods, including deep neural networks. | Energy price forecasting, time series forecasting, hybrid forecasting, machine learning, deep learning |
| [11] | To develop a multi-objective and model selection-based ensemble system for interval-valued wind speed forecasting, aiming to improve decision support and risk management. | Interval-valued time series forecasting, wind speed forecasting, ensemble learning, model selection, multi-objective optimization. |

| Reference | Research Purpose | Approach/Domain |
|---|---|---|
| [13] | To develop a hybrid ensemble forecasting scheme for STLF by combining ICEEMDAN decomposition, sub-model selection, MOGOA optimization, and point and interval prediction to improve forecasting accuracy. | Short-term load forecasting, ensemble learning, time series decomposition. |
| [20] | To develop a novel ultra-short-term wind power prediction model using multiple algorithm optimization, adaptive selection, and VMD decomposition to improve prediction accuracy and robustness. | Wind power forecasting, VMD decomposition, whale optimization algorithm, hybrid kernel extreme learning machine, adaptive selection. |
| [16] | To develop a novel ensemble model based on wavelet transform and ARIMAX for short-term wind power output forecasting in South Korea, aiming to improve grid reliability and reduce market service costs. | Short-term wind power forecasting, wavelet transform, ARIMAX, ensemble modeling, comparison analysis, NMAE. |
| [10] | To improve wind energy forecasting accuracy and stability by combining WSTD denoising, MSP-based model selection, BO optimization, and DSEC error correction. | Wind energy forecasting, WSTD, MSP, model selection, optimization. |
| [34] | To propose an improved meta-learning framework for deep learning time series forecasting model selection by using image-based representation and data augmentation techniques to enhance accuracy and efficiency. | Time series forecasting, meta-learning, deep learning, image-based representation, data augmentation, model selection. |
| [18] | To develop a hybrid two-phase decomposition and two-layer prediction model architecture for rolling peak load prediction, combining ICEEMDAN, SE, VMD, XGBoost, MLR, and SSA to improve prediction accuracy. | Peak load prediction, time series forecasting, ICEEMDAN decomposition, ensemble learning, error correction. |
| [19] | To develop an intelligent prediction system for ultra-short-term wind and photovoltaic power prediction by combining improved VMD decomposition and CNN optimized by WHO, aiming to improve prediction accuracy and stability. | Wind and photovoltaic power forecasting, deep learning, decomposition, optimization. |
| [6] | To develop a dynamic hybrid model for wind energy forecasting, combining ANN, SVM, and K-NN with a dynamic switching mechanism to improve both short-term and long-term forecasting accuracy. | Wind energy forecasting, hybrid modeling, time series analysis, evaluation, NMAE. |
| [17] | To develop a novel ensemble framework for COVID-19 forecasting, combining KRidge, dRVFL, and Ridge regression within a linear relationship, optimized by A-DEPSO, and using TVF-EMD decomposition and LGBM feature selection to improve prediction accuracy. | COVID-19 forecasting, ensemble modeling, KRidge, dRVFL, Ridge regression, A-DEPSO, TVF-EMD, LGBM, feature selection, time series analysis. |
| [7] | To develop an automated approach for time series forecasting that extracts time series characteristics, selects the best-suited machine learning method using a recommendation system, and performs accurate forecasts. | Time series forecasting, machine learning, feature extraction, recommendation systems, automated forecasting. |

| Reference | Research Purpose | Approach/Domain |
|---|---|---|
| [42] | To develop a forecasting meta-learning approach called AutoForecast for fast automatic selection of the best forecasting model for a new unseen time-series dataset without the need for extensive training and evaluation. | Time series forecasting, meta-learning, model selection, machine learning. |
| [43] | To develop AutoXPCR, a novel AutoML method for time series forecasting that considers multiple objectives (predictive error, complexity, resource demand) and provides explainable recommendations. | Time series forecasting, AutoML, deep neural networks, meta-learning, explainable AI, multi-objective optimization. |
| [12] | To develop a hybrid wind speed forecasting framework combining data decomposition techniques and BiDLSTM networks with skip connections to improve prediction accuracy and stability. | Wind speed forecasting, data decomposition, Wavelet Transform, deep learning, time series analysis. |
| [14] | To develop a novel ensemble forecasting system for carbon price prediction, combining data decomposition, feature selection, optimal sub-model determination, and improved multi-objective optimization to improve point and interval forecasting accuracy. | Carbon price forecasting, ensemble learning, data decomposition, feature selection, multi-objective optimization, point and interval forecasting. |
| [25] | To develop a novel model selection framework for agricultural commodity price forecasting, combining time series features, forecast horizons, and machine learning models (ANN, SVR, ELM) with RF or SVM as meta-learners, and using MRMR for feature selection. | Agricultural commodity price forecasting, model selection, time series analysis, feature engineering. |
| [41] | To develop an expert system for time series forecasting model selection, combining relative in-sample and out-of-sample performance to automatically select the best performing forecasting model without requiring decision-maker intervention. | Time series forecasting, model selection, expert systems, machine learning, classification, performance evaluation. |
| [44] | To develop a novel wind speed forecasting system that includes data analysis, model selection strategy, forecasting processing combined with a modified multi-objective optimization algorithm, and model evaluation to improve forecasting performance. | Wind speed forecasting, model selection, optimization, time series analysis, hybrid forecasting systems. |
| [45] | Develop a versatile framework (SimpleTS) for time series forecasting that offers high efficiency and accuracy across various data types. | Time series forecasting, model selection, classification, clustering, soft labeling, Alibaba Cloud database monitoring system. |
| [22] | To develop a novel multi-objective optimization algorithm to optimize the parameters of different models, combined with a model selection strategy to improve the accuracy and stability of wind speed forecasting and wind power conversion. | Wind speed forecasting, multi-objective optimization, model selection, wind power conversion, time series analysis. |
| [26] | To develop a meta-learning-based framework (GizaML) for automated algorithm selection and hyperparameter tuning for time-series forecasting. | Time series forecasting, meta-learning, algorithm selection, hyperparameter tuning, data and feature engineering, Bayesian optimization, Large Language Models. |

| Reference | Research Purpose | Approach/Domain |
|---|---|---|
| [58] | To develop a novel Heterogeneity-Informed Meta-Parameter Learning scheme (HimNet) for spatiotemporal time series forecasting, aiming to capture and leverage spatiotemporal heterogeneity. | Spatiotemporal time series forecasting, meta-parameter learning, spatial and temporal embeddings, deep learning, model interpretability. |
| [35] | To develop a novel hybrid and feature-based stacking ensemble model (DeFORMA) for time series forecasting, and to provide a unified taxonomy for model fusion techniques. | Time series forecasting, hybrid models, feature-based stacking, deep learning, model fusion, M4 dataset. |
| [32] | To develop a meta-optimization framework (DeepTime) for learning deep time-index models for time series forecasting, addressing the limitations of naive deep time-index models and achieving efficient and accurate forecasting. | Time series forecasting, deep learning, time-index models, meta-optimization, long sequence time series forecasting. |
| [59] | To investigate whether meta-learning can discover generic ways of processing time series data to improve generalization on new, unseen time series datasets. | Meta-learning, time series forecasting, residual connections, zero-shot learning, neural networks, time series analysis, data generalization. |
| [36] | To introduce a collection of ensemble methods for time series forecasting and demonstrate their effectiveness through extensive experiments and develop a meta-learning approach to select the optimal ensemble method and hyperparameter configuration for a given dataset based on meta-features. | Time series forecasting, ensemble learning, meta-learning, model selection, hyperparameter tuning, M4, M5, M3, FRED datasets. |
| [48] | To develop a combined meta-learning and transfer-learning approach to address the missing data problem in time series forecasting, specifically for air pollution prediction. | Time series forecasting, meta-learning, transfer learning, missing data imputation, air pollution prediction, neural networks. |
| [8] | To develop a meta-learning approach to adaptively select the optimal ensemble of forecasting methods and their pooling strategy for a given time series. | Time series forecasting, ensemble learning, meta-learning, feature engineering, random forest, M4 competition. |
| [60] | To address the limitations of existing meta-learning approaches for time series forecasting, this study focuses on analyzing the impact of various factors such as the subset of top forecasters, error measures, and feature dimensionality on the performance of meta-learners. | Time series forecasting, meta-learning, model selection, feature engineering, dimensionality reduction, NN5 competition dataset. |
| [49] | To investigate the effectiveness of different performance estimation methods for model selection in time series forecasting, specifically analyzing how often they select the best model and the associated performance loss when they fail to do so. | Time series forecasting, model selection, performance estimation, empirical analysis. |
| [47] | To address the limitations of traditional temporal cross-validation for hyperparameter optimization (HPO) in time series forecasting by proposing a novel technique, H-Pro, that leverages data hierarchies to improve test performance. | Time series forecasting, hyperparameter optimization, data hierarchies, test proxies, empirical evaluation on hierarchical forecasting datasets (Tourism, Wiki, Traffic, Tourism-L, M5 competition). |

## A.2 Comparison of statistical difference of all baseline models and TSFeatures using the NMAE metric.

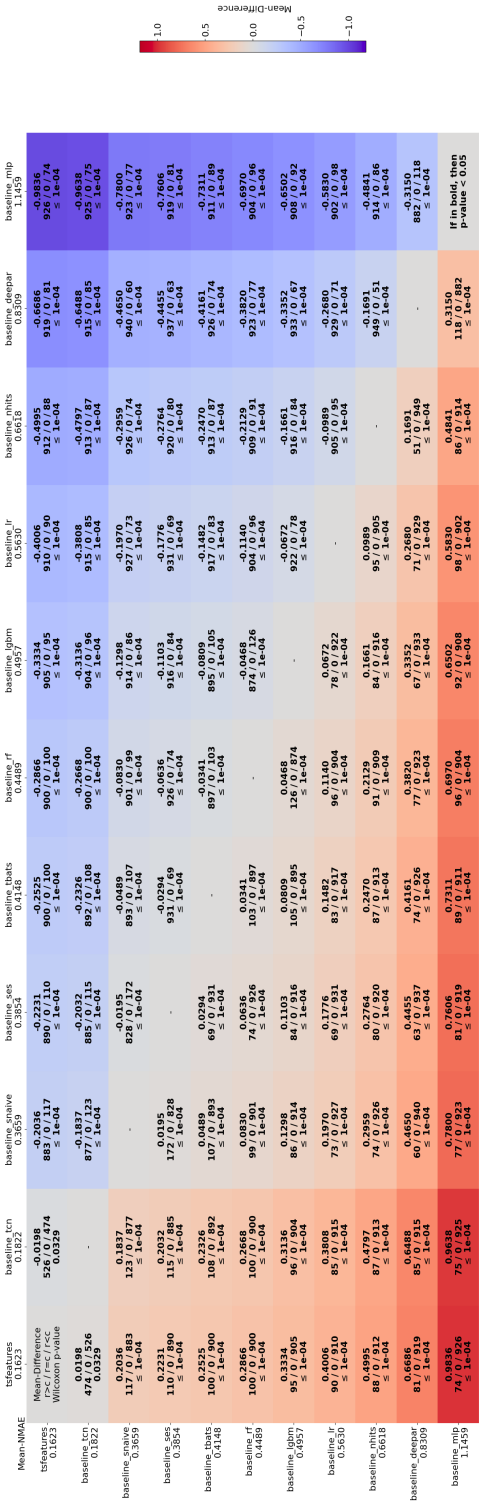



Figure A.1: Heatmap of pairwise NMAE comparison with Wilcoxon signed-rank test of TSFeatures and all the baseline models.

## A.3 Comparison of performance metrics across methods

Table A.2: Performance metrics.

| Model | MASE | MASE md | NMAE | NMAE md | RMSE | RMSE md |
|---|---|---|---|---|---|---|
| TSFeatures | **2.73 ± 2.52** | **1.80** | **0.16 ± 0.20** | **0.09** | 1005.13 ± 1570.46 | **487.96** |
| DeepAR | 77.72 ± 188.11 | 24.73 | 0.83 ± 1.13 | 0.44 | 7157.16 ± 13027.90 | 2803.48 |
| LGBM | 87.66 ± 193.56 | 37.93 | 0.50 ± 0.60 | 0.30 | 5199.68 ± 9221.95 | 2297.22 |
| LR | 84.52 ± 192.28 | 33.09 | 0.56 ± 0.71 | 0.33 | 5650.93 ± 10095.43 | 2435.62 |
| MLP | 73.57 ± 185.61 | 20.36 | 1.15 ± 1.67 | 0.55 | 8582.73 ± 15970.89 | 3175.77 |
| NHITS | 80.94 ± 190.68 | 28.77 | 0.66 ± 0.87 | 0.36 | 6250.32 ± 11285.66 | 2537.15 |
| RF | 91.02 ± 194.92 | 41.58 | 0.45 ± 0.53 | 0.28 | 4860.76 ± 8542.72 | 2163.11 |
| SNaive | 101.88 ± 199.97 | 51.51 | 0.37 ± 0.40 | 0.24 | 4181.41 ± 7180.99 | 1901.37 |
| SES | 97.80 ± 198.17 | 47.95 | 0.39 ± 0.43 | 0.25 | 4371.79 ± 7562.30 | 1968.65 |
| TBATS | 94.98 ± 197.23 | 45.15 | 0.42 ± 0.47 | 0.26 | 4606.80 ± 8015.25 | 2059.13 |
| TCN | 3.44 ± 7.18 | 1.94 | 0.18 ± 0.32 | 0.10 | **1000.36 ± 1436.43** | 517.92 |

## A.4 RMSE values across folds and mean values of the LGBM and RF metalearners

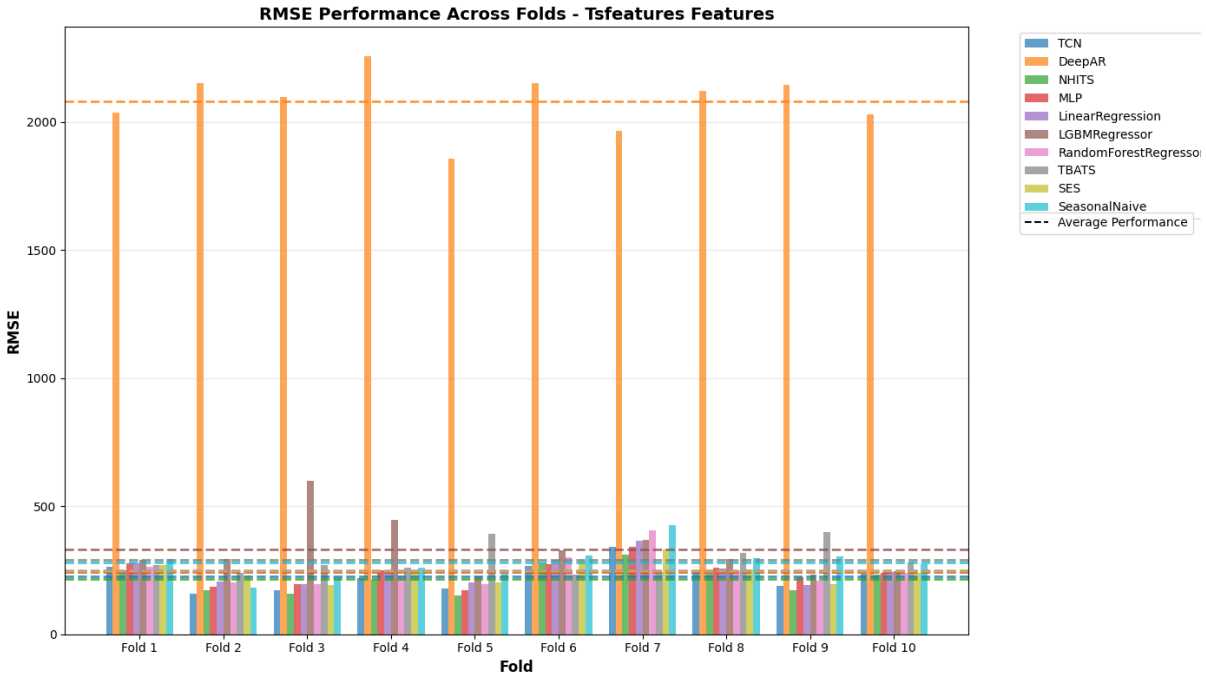



Figure A.2: RMSE mean and across folds values of the LGBM based metalearner.

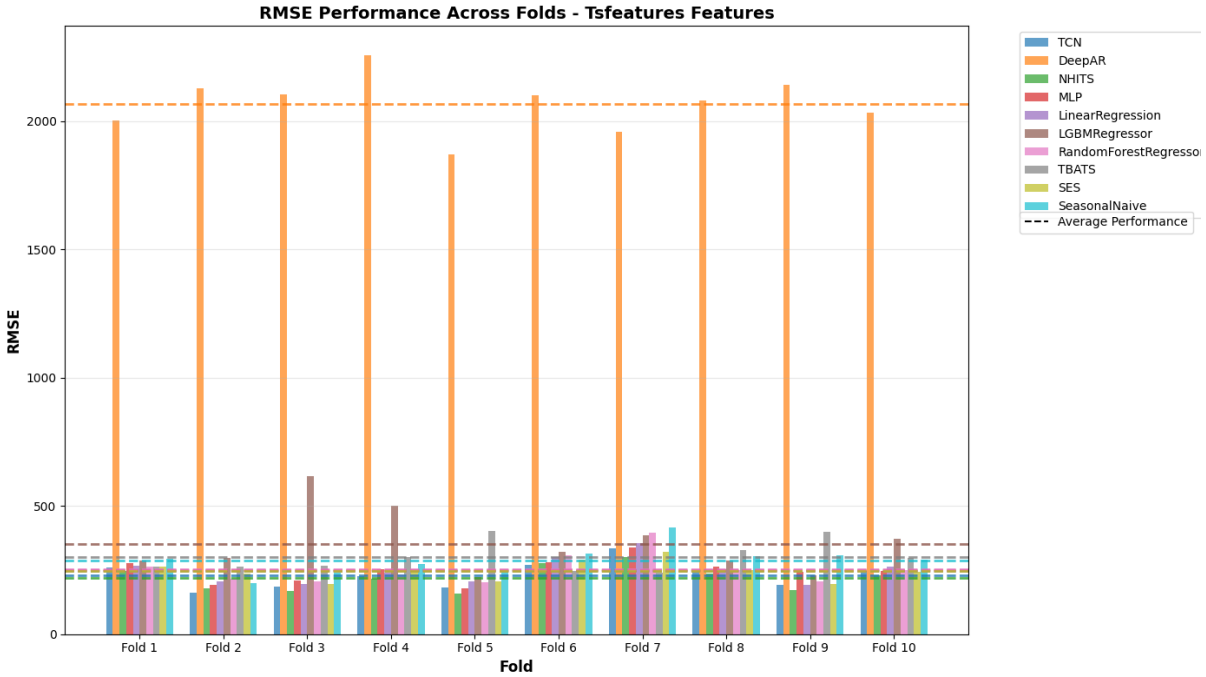



Figure A.3: RMSE mean and across folds values of the Random Forest based metalearner.

## A.5 Comparison of statistical difference of all baseline models and TSFeatures using the NMAE metric for the quarterly frequency.

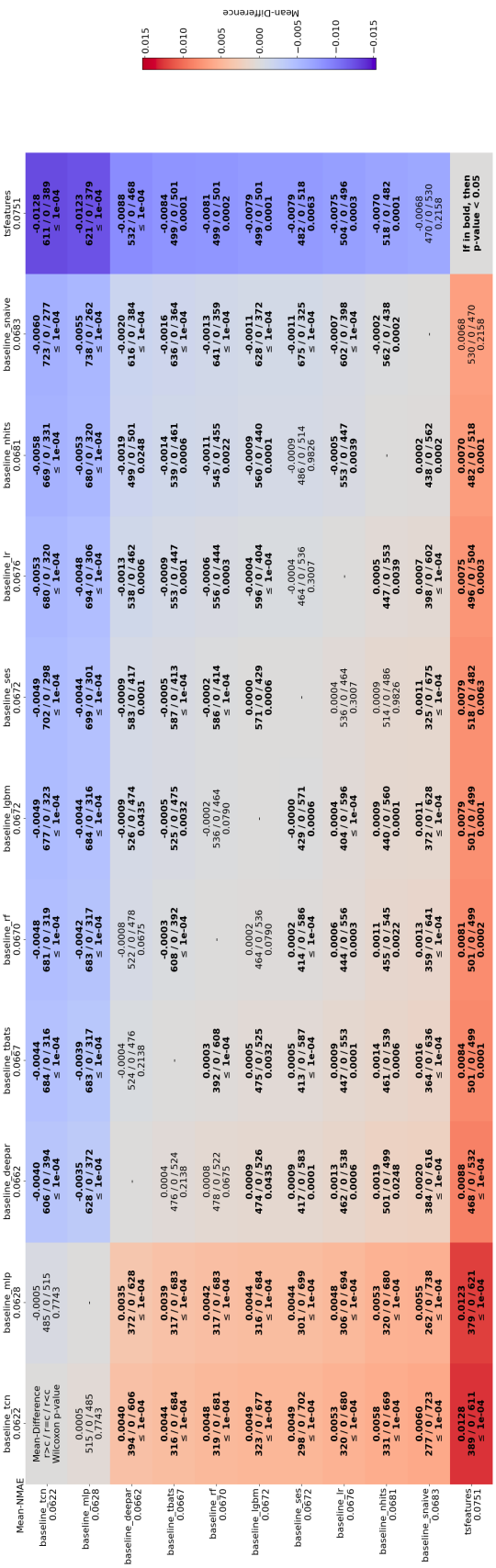



Figure A.4: Heatmap of pairwise NMAE comparison with Wilcoxon signed-rank test of TSFeatures and some of the baseline models.