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

José Luís Araújo
Faculty of Engineering
University of Porto
Porto, Portugal
email@fe.up.pt

Carlos Soares
Faculty of Engineering
University of Porto
Porto, Portugal
csoares@fe.up.pt

Luís Paulo Reis
Faculty of Engineering
University of Porto
Porto, Portugal
lpreis@fe.up.pt

Moisés Santos
Faculty of Engineering
University of Porto
Porto, Portugal
mrsantos@fe.up.pt

***Abstract*—Time series forecasting is essential in fields such as finance, healthcare, and supply chain management, where accurate predictions support critical decisions. Traditional methods often fail to model the complex nonlinear patterns and multiple seasonal variations found in real-world data. Although decomposition techniques like MSTL improve forecasting by separating time series components, the selection of an appropriate model for each component remains a challenging, heuristic-driven task. To address this, we introduce a novel metalearning framework that automates model selection in a decomposition-aware manner. Unlike existing methods that apply fixed models or treat series monolithically, our approach uses a granular, component-based strategy. It first decides whether to decompose a series via residual analysis, then recommends the suitable forecasting model from a diverse set of statistical and Deep Learning algorithms for each component. Experiments on 5000 monthly series from the M4 Competition show that our framework achieves statistically significant improvements over both non-decomposed metalearning baselines (23% lower NMAE, 61% lower error variance) and state-of-the-art models such as TCN and DeepAR.**

***Index Terms*—Forecasting, Metalearning, Decomposition, AutoML.**

I. Introduction

Time series forecasting is critical for data-driven decision-making across domains such as energy and retail, where accurate predictions optimize operations and manage risk [1]. Real-world series, however, often exhibit high volatility, non-stationarity, and multiple seasonal patterns, challenging traditional statistical approaches.

Decomposition methods like Seasonal-Trend decomposition using Loess (STL) [2] address this by separating a series into interpretable components—trend, seasonality, and residuals—enabling more focused modeling. A key difficulty is the *Algorithm Selection Problem* (ASP) [3]: different components may require distinct modeling techniques, and manually selecting suitable per-component models is computationally intensive and expertise-dependent.

Metalearning mitigates the ASP by using prior experience to predict algorithm performance on new tasks [4], [5]. Recent frameworks such as METAFORE [6] integrate decomposition with algorithm selection but often impose simplifications—for example, applying heuristic forecasts to seasonal components or using a single model for combined trend and residuals—thus limiting the potential of component-specific modeling.

This work proposes a refined metalearning framework that selects models granularly per component. Our contributions are:

- An automatic criterion, based on residual properties, to decide whether decomposition is necessary, avoiding undue complexity for simple series.
- A component-wise recommendation system employing MSTL [7] to handle multiple seasonalities, choosing from a portfolio of 10 base-learners spanning classical methods (e.g., TBATS, ETS) and deep learning models (e.g., TCN, DeepAR).
- A comprehensive evaluation on 5000 monthly series from the M4 Competition [8], showing that our framework outperforms individual state-of-the-art models and delivers more consistent and reliable forecasts.

The document is structured as follows: Section II reviews related work. Section III details the proposed framework. Section IV outlines the experimental setup. Section V presents and discusses the results. Section VI concludes.

II. Related Work

The landscape of time series forecasting has evolved from standalone statistical models to complex, hybrid systems integrating decomposition and metalearning. This section reviews the state-of-the-art in these domains.

A. Time Series Decomposition Techniques

Decomposition is pivotal for handling non-stationary data by isolating deterministic patterns from noise. The classical *Seasonal-Trend Decomposition using Loess* (STL) [2] remains a standard due to its robustness against outliers. However, its inability to handle multiple seasonalities led to the development of MSTL (Multiple STL) [7], which iteratively extracts multiple seasonal components, making it suitable for high-frequency data (e.g., hourly electricity demand).

Beyond Loess-based methods, signal processing techniques have gained traction. *Empirical Mode Decomposition* (EMD) and its variants, such as *Ensemble EMD* (EEMD) and *Improved Complete Ensemble EMD with Adaptive Noise*

(ICEEMDAN), are widely used to decompose non-linear signals into Intrinsic Mode Functions (IMFs) [9], [10]. Similarly, *Variational Mode Decomposition* (VMD) [11] offers a non-recursive approach that is robust to noise and sampling variability. While effective, these advanced signal processing methods often suffer from high computational costs and "mode mixing" issues. Consequently, our framework adopts MSTL for its balance of computational efficiency and ability to handle the multiple seasonalities typical of business and economic data.

B. Evolution of Forecasting Algorithms

Forecasting methodologies have shifted from traditional statistical models to Machine Learning (ML) and Deep Learning (DL) architectures. Statistical methods like ARIMA [12] and Exponential Smoothing (ETS) [13] are prized for interpretability and performance on linear data. However, they often fail to capture complex non-linear dependencies.

To address this, ML algorithms such as Support Vector Regression (SVR) [14] and XGBoost [15] have been extensively applied, often yielding superior results in regression tasks. More recently, Deep Learning has defined the state-of-the-art. Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) networks, effectively model temporal dependencies [16]. Furthermore, architectures like Temporal Convolutional Networks (TCN) [17] and probabilistic transformers like DeepAR [18] allow for global modeling across thousands of related series, capturing latent patterns that local models miss. Our framework incorporates a diverse portfolio encompassing both traditional (TBATS, SES) and DL models (TCN, DeepAR, NHITS) to exploit their complementary strengths.

C. Metalearning and Algorithm Selection

The *Algorithm Selection Problem* (ASP) [3] is particularly acute in forecasting due to the diversity of available models. Metalearning addresses this by mapping dataset characteristics (meta-features) to algorithm performance.

Early approaches, such as the work by Wang et al. [5], used clustering and rule induction based on statistical features. This evolved into sophisticated frameworks like FFORMS (Feature-based FORecast Model Selection) [4], which utilizes Random Forests to classify the best model from a pool. Hybrid approaches like the *Energy Meta Learning System* (EMLS) [19] combine multiple metalearners (e.g., Neural Networks and Random Forests) to improve recommendation stability.

Specific to decomposition-based forecasting, the METAFORE framework [6] pioneered the selection of algorithms for trend and residual components. However, METAFORE is limited by its static treatment of seasonality (forcing a seasonal naive approach) and the aggregation of components before model selection. Our work advances this by: (1) implementing a fully component-wise selection strategy, including learnable models for multiple seasonalities; (2) introducing a pre-decomposition decision step based on residual normality; and (3) employing a regression-based metalearner to predict error magnitudes rather than simple classification rankings.

D. Meta-Feature Engineering

The success of metalearning relies on the quality of meta-features. Standard libraries like *TSFeatures* [20] extract interpretable properties (e.g., entropy, stability, crossing points), while *Catch22* [21] focuses on a reduced set of canonical physics-based features. Other works have explored custom feature sets, including domain-specific characteristics for energy consumption [19] or multivariate correlations [22]. Based on our ablation studies, we utilize *TSFeatures* as it provides the most discriminative representation for the diverse M4 dataset.

III. Proposed Framework

The proposed solution is an automated pipeline consisting of two main phases: *Metadata Collection* (training) and *Model Recommendation* (inference).

Figure 1 illustrates the metadata collection phase during training. The pipeline begins with conditional decomposition assessment: each training series undergoes Shapiro-Wilk testing on preliminary decomposition residuals. If residuals are non-normal ($p < 0.05$), indicating that decomposition introduces artifacts, the series bypasses MSTL and proceeds directly to meta-feature extraction. Otherwise, MSTL decomposes the series into trend, seasonal, and residual components. Each component is then processed through TSFeatures extraction (42 features per component), and all 10 base-learners are trained on each component to compute ground-truth MAE values. These meta-features and corresponding errors form the training dataset for the LightGBM metalearner, which learns to map component characteristics to expected model performance.

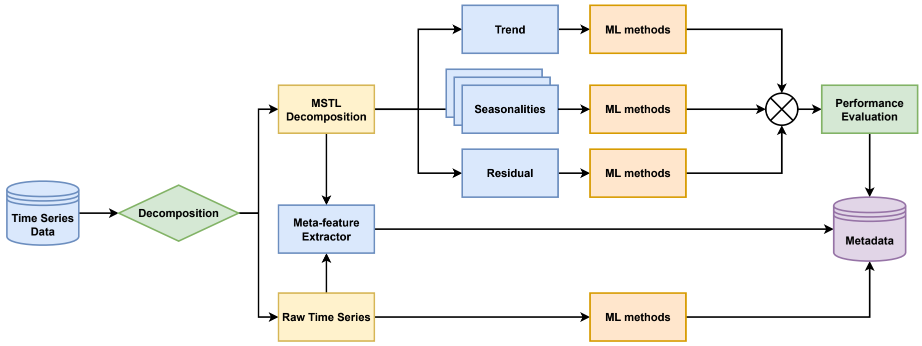
A. Decomposition Strategy: MSTL

We adopt MSTL to handle complex periodic patterns. Mathematically, MSTL decomposes a time series Y_t into a trend T_t, K seasonal components $S_t^{(k)}$, and a remainder R_t:

$$Y_t = T_t + \sum_{k=1}^{K} S_t^{(k)} + R_t \tag{1}$$

Ideally, the decomposition isolates deterministic patterns, leaving R_t as stationary noise.

A novel aspect of our framework is the conditional decomposition. Before processing, we analyze the distribution of the series' residuals. If the residuals of a preliminary decomposition deviate significantly from normality (using Shapiro-Wilk test), suggesting the decomposition is artifact-prone or the series is purely stochastic, the framework skips decomposition and models the raw series directly. This adaptive step prevents the "over-decomposition" of simple signal structures.



.pdf

Fig. 1. Pipeline of the metadata collection step showing decomposition decision, component extraction, and metadata generation.

B. Meta-Feature Extraction

To train the metalearner, we must convert variable-length time series (or components) into fixed-length feature vectors. We investigated two state-of-the-art libraries:

1) **TSFeatures** [20]: Extracts 42 statistical features, including strength of trend/seasonality, spectral entropy, autocorrelation, and crossing points.
2) **Catch22** [21]: A reduced set of 22 features selected for their high discriminative power and computational efficiency across diverse scientific domains.

As discussed in Section V, our ablation studies identified *TSFeatures* as the superior descriptor for this specific task, providing better stability in metalearning predictions.

C. The Base-Learner Portfolio

The framework selects from a diverse pool of 10 forecasting models, ensuring coverage of linear, non-linear, and structural patterns:

- **Statistical:** Seasonal Naive, SES (Simple Exponential Smoothing), TBATS (Trigonometric Box-Cox ARMA Trend Seasonal) [23].
- **Machine Learning:** Random Forest Regressor, LightGBM (LGBM) Regressor, Linear Regression.
- **Deep Learning:** MLP (Multilayer Perceptron), TCN (Temporal Convolutional Network) [17], DeepAR [18], NHITS [24].

During metadata collection (training the metalearner), all 10 base-learners use default hyperparameters to reduce computational cost. During inference (final forecasting on test series), we use grid-search optimized hyperparameters from established M4 benchmarks [8]. This creates a train-test distribution gap—the metalearner learns from default-configuration model performance but recommends tuned models—which we acknowledge as a limitation. Despite this mismatch, results demonstrate the metalearner successfully generalizes across the performance landscape.

This model diversity is critical for metalearning, as the system relies on the performance variance across models to learn meaningful selection boundaries.

D. Metalearner Architecture

We formulate the metalearning task as a multi-output regression problem. The input is the vector of meta-features for a specific component (e.g., Trend), and the output is a vector containing the predicted Mean Absolute Error (MAE) for each of the 10 base-learners.

We employ a LightGBM Regressor as the metalearner. While Random Forest is a common choice in literature [4], our base-level evaluation (Section V-D) demonstrated that LightGBM yields more consistent MASE (Mean Absolute Scaled Error) predictions.

The metalearner training procedure operates as follows: For each of the 4,000 training series, we (1) apply MSTL decomposition to extract trend, seasonal, and residual components (or skip if Shapiro-Wilk $p < 0.05$), (2) extract TSFeatures (42 features) from each component independently, (3) train all 10 base-learners on each component using temporal cross-validation and record their actual MAE on a held-out validation fold, (4) create training instances mapping component meta-features to 10-dimensional MAE vector. We train separate LightGBM regressors for each component type (trend, seasonal, residual) plus one for raw series (when decomposition is bypassed). Each LightGBM uses 100 estimators, max_depth=5, learning_rate=0.1, trained via 10-fold cross-validation.

During inference for a new series Y_new, the model recommendation pipeline (Figure 2) executes the following sequence:

1) Decomposition Decision: Apply Shapiro-Wilk test to residuals. If $p >= 0.05$, proceed with MSTL decomposition into components $C_1, ..., C_m$; otherwise, treat Y_{new} as a single component.

2) Feature Extraction: Extract TSFeatures meta-features (42-dimensional vector) for each component independently.

3) Metalearner Prediction: For each component, the trained LightGBM metalearner receives the feature vector and outputs predicted MAE values for all 10 base-learners.

4) Model Selection: Select the base-learner with minimum predicted MAE for each component. Train the selected model on the component's historical data.

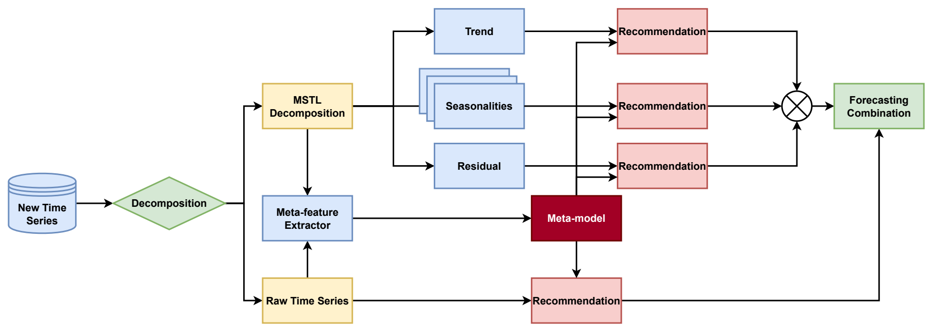


Fig. 2. Pipeline of the model recommendation step showing component-wise metalearner selection and forecast aggregation.

5) Forecast Aggregation: Generate forecasts for each component using its selected model. Sum component forecasts to produce the final prediction $\hat{Y}_{new} = \sum_i^m C_i$.

This component-wise strategy (visualized in Figure 2) ensures that each component is modeled by the algorithm best suited to its specific characteristics—for example, TBATS for highly seasonal components, TCN for complex trend patterns, and Seasonal Naive for stable, repetitive seasonality.

IV. Experimental Procedures

This chapter covers the experimental setup used to assess the performance of the proposed forecasting framework.

A. Dataset

We utilized data from the M4 Forecasting Competition [8]. Due to the high computational cost of training deep learning models within a metalearning loop, we sampled 5000 Monthly series using random sampling to preserve the distribution of series characteristics (length, trend strength, seasonality) from the full M4 monthly dataset (48,000 series). The data was split into training and test sets using an 80/20 ratio.

B. Evaluation Metrics

We assess performance at the *Base-Level* (accuracy of the final forecast). The primary metrics are:

- **MASE (Mean Absolute Scaled Error):** Scale-independent metric, ideal for comparing across diverse series.
- **NMAE (Normalized Mean Absolute Error):** The MAE normalized by the mean of the actual values, providing a percentage-like interpretation.
- **RMSE (Root Mean Squared Error):** Penalizes large errors more heavily.

C. Baselines

We compare our proposed framework (*TSFeatures* with decomposition) against:

1) **No-Decomposition Baselines:** The same metalearning framework but restricted to selecting models for the raw series only (*TSFeatures_und*).
2) **Individual Models:** The 10 base-learners running independently on the full dataset without selection.

Statistical significance is verified using the Wilcoxon signed-rank test.

V. Results and Discussion

This chapter presents and interprets the experimental results from the evaluation procedure conducted in chapter IV.

A. RQ1: The Advantage of Decomposition

Our first research question investigates whether the dynamic decomposition strategy yields better results than a monolithic metalearning approach. Table I presents the comparison.

TABLE I
Performance Metrics: Decomposition vs. No-Decomposition

Method	MASE (Mean $\pm$ Std)	NMAE	RMSE
TSFeatures (Ours)	**2.73 $\pm$ 2.52**	**0.16**	**1005**
TSFeatures_und	3.54 $\pm$ 5.50	0.19	1077
Catch22 (Decomp)	3.50 $\pm$ 6.59	0.17	993
Catch22_und	3.30 $\pm$ 6.26	0.17	1057

The results clearly favor the proposed approach. The *TSFeatures* decomposition framework achieves the lowest mean MASE (2.73) compared to the non-decomposed version (3.54). More importantly, the standard deviation of the error is drastically reduced (from 5.50 to 2.52). This indicates that decomposition acts as a stabilizer, preventing catastrophic failures on complex series where single models might struggle to capture both trend and seasonality simultaneously.

Pairwise statistical testing confirms this superiority. The *TSFeatures* approach is statistically significantly better than *TSFeatures_und* with a p-value of 0.0034.

B. RQ2: Comparison with Individual Models

We next compare the framework against the individual base-learners. Figure 3 summarizes the pairwise NMAE comparison.

The proposed framework achieves the lowest NMAE (0.1623) across the entire dataset, statistically outperforming the second-best model, TCN (0.1822). Traditional deep learning powerhouses like DeepAR (0.8309) and MLP (1.1459) performed poorly on this specific subset of monthly data, likely

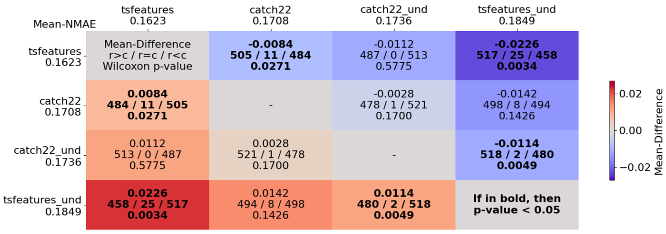


Fig. 3. Statistical significance heatmap for pairwise NMAE comparison using Wilcoxon signed-rank test. Blue (red) cells indicate the row (column) method statistically outperforms with $p < 0.05$. Cell values show mean difference, win/tie/loss counts, and p-values. TSFeatures achieves statistical superiority over all competitors.

TABLE II
NUMBER OF PER-SERIES WINS (LOWEST ERROR)

Model	MASE Wins	NMAE Wins	RMSE Wins
TSFeatures (Ours)	**526**	**508**	**494**
TCN	474	435	462
DeepAR	0	8	5
LGBM	0	1	1
S. Naive	0	6	4

due to the limited length of some M4 series which hampers the training of data-hungry global models.

Table II shows per-series wins across the 1,000 test series. The metalearner achieves lowest MASE on 526 series (52.6%), surpassing TCN (474 series, 47.4%). Notably, strong generalist models such as LGBM secured almost zero wins. This suggests that while LGBM is a robust "average" performer, it is rarely the *best* specific choice for a given series, whereas the metalearner can exploit the specific strengths of niche models (such as TBATS for strong seasonality) when appropriate.

C. Consistency Analysis

To visualize reliability, we define a consistency score as the ratio of Mean Error to Standard Deviation. A high score implies high accuracy with low volatility.

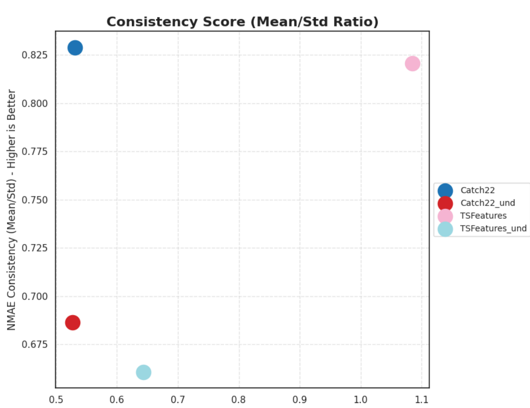


Fig. 4. Consistency Score Plot. The proposed TSFeatures framework (top-right) exhibits superior stability compared to individual baselines (bottom-left), effectively breaking the accuracy-reliability trade-off.

As illustrated in the Figure 4, our framework occupies the ideal top-right quadrant. Base models like MLP or Linear Regression fall into the bottom-left (low accuracy, high volatility). The decomposition strategy effectively buffers against the high variance inherent in individual deep learning stochastic initializations.

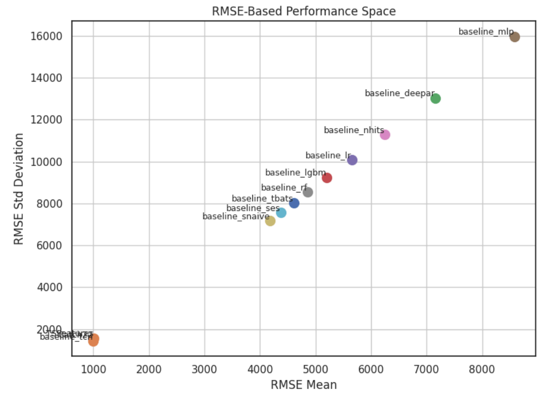


Fig. 5. RMSE-based Performance Space. TSFeatures and TCN occupy the ideal bottom-left corner (low error, low volatility).

Figure 5 provides a complementary analysis using RMSE as the error metric. Plotting RMSE mean (x-axis) against RMSE standard deviation (y-axis), we again observe that TSFeatures and TCN cluster in the bottom-left optimal region (low error, low volatility), while most other baselines scatter toward higher error and variance. Notably, DeepAR, MLP, and Linear Regression exhibit dramatically higher RMSE volatility, suggesting these methods produce catastrophic failures on certain series. In contrast, our framework demonstrates consistent performance across the heterogeneous test set. This robustness is critical for production deployment, where unpredictable failures can undermine user trust even when average performance is acceptable.

D. Ablation Studies

Through 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.

1) Feature Sets: Comparing *TSFeatures* and *Catch22* (Table I), *TSFeatures* yields better MASE and NMAE. While *Catch22* offers a slight edge in RMSE, the 42 features of *TSFeatures* appear to capture the structural nuances of monthly data (e.g., seasonality strength) more effectively than the canonical physics-based features of *Catch22* for this specific forecasting task.

2) Metalearner Choice: We compared Random Forest (RF) and LightGBM as the meta-regressor. LightGBM provided a significantly lower standard deviation in MASE predictions (2.52 vs 6.59 for RF). This stability in the meta-level prediction translates directly to more robust model selection at the base level.

3) Quarterly Data Limitation: We also evaluated the framework on M4 Quarterly data. Here, the performance dropped, with TCN outperforming our framework (611 wins vs 389 for TSFeatures). We hypothesize that the shorter length of quarterly series makes feature extraction less reliable, and the lower frequency obscures the complex patterns that decomposition thrives on. This highlights a boundary condition for our approach: it is most effective when series are long enough to exhibit extractable, stable components.

VI. Conclusion

This paper presented a novel metalearning framework for time series forecasting that integrates adaptive MSTL decomposition with component-wise algorithm selection. By treating the forecasting problem at the component level, our system leverages the specific strengths of diverse algorithms—using statistical methods where appropriate and deep learning where necessary. Experimental results on the M4 Monthly dataset confirm that this granular approach yields statistically significant improvements in accuracy and, crucially, reliability. The framework successfully avoids the high error variance associated with individual model selection. Even though positive results were achieved, our approach has two key limitations. First, the hyperparameter train-test mismatch (metalearner trained on default-configuration models but recommending tuned models) may reduce recommendation accuracy, though results suggest this gap is bridged by the metalearner's ability to learn relative performance rankings. Second, performance degrades on quarterly data, where shorter series lengths reduce meta-feature reliability. Future work will focus on improving performance for short, low-frequency series (e.g., Quarterly) by engineering specialized meta-features and integrating end-to-end hyperparameter optimization within the metalearning loop.

Acknowledgment

This work is a result of Agenda "Center for Responsible AI", nr. C645008882-00000055, investment project nr. 62, financed by the Recovery and Resilience Plan (PRR) and by European Union - NextGeneration EU. Funded by the European Union – NextGenerationEU. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them. AISym4Med (101095387) supported by Horizon Europe Cluster 1: Health, ConnectedHealth (n.o 46858), supported by Competitiveness and Internationalisation Operational Programme (POCI) and Lisbon Regional Operational Programme (LISBOA 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF). This work was financially supported by: UID/00027 of the LIACC - Artificial Intelligence and Computer Science Laboratory - funded by Fundação para a Ciência e a Tecnologia, I.P./ MCTES through the national funds.

References

[1] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 2nd ed. Melbourne, Australia: OTexts, 2018. [Online]. Available: OTexts.com/fpp2
[2] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: A seasonal-trend decomposition procedure based on loess," *Journal of Official Statistics*, vol. 6, no. 1, pp. 3–73, 1990.
[3] J. R. Rice, "The algorithm selection problem," *Advances in Computers*, vol. 15, pp. 65–118, 1976.
[4] T. S. Talagala, R. J. Hyndman, and G. Athanasopoulos, "Meta-learning how to forecast time series," Monash Econometrics and Business Statistics Working Papers, Tech. Rep. 18/18, 2018.
[5] 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.
[6] M. Santos, A. de Carvalho, and C. Soares, "Metafore: algorithm selection for decomposition-based forecasting combinations," *International Journal of Data Science and Analytics*, pp. 1–14, 2024.
[7] K. Bandara, C. Bergmeir, and S. Smyl, "Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach," *Expert Systems with Applications*, vol. 140, p. 112896, 2020.
[8] 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.
[9] J. Wang *et al.*, "A novel decomposition-ensemble forecasting system for dynamic dispatching of smart grid," *Expert Systems with Applications*, vol. 117201, 2022.
[10] Y. Hao *et al.*, "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. 117868, 2024.
[11] Q. Lin *et al.*, "A novel ultra-short-term wind power prediction model jointly driven by multiple algorithm optimization and adaptive selection," *Energy*, vol. 288, p. 129724, 2024.
[12] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. Holden-Day, 1970.
[13] R. J. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder, *Forecasting with Exponential Smoothing: The State Space Approach*. Springer, 2008.
[14] 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.
[15] Y. Zhang *et al.*, "A short-term wind energy hybrid optimal prediction system with denoising and novel error correction technique," *Energy*, 2022.
[16] 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. 113944, 2021.
[17] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.
[18] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.
[19] S. Shahoud *et al.*, "Incorporating unsupervised deep learning into meta learning for energy time series forecasting," in *Proceedings of the Future Technologies Conference (FTC) 2020*. Springer, 2021, pp. 326–345.
[20] R. J. Hyndman, Y. Kang, T. Talagala, F. Wang, and Y. Yang, *tsfeatures: Time Series Feature Extraction*, 2020, r package version 1.0.2. [Online]. Available: https://CRAN.R-project.org/package=tsfeatures
[21] C. H. Lubba *et al.*, "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.
[22] H. Engbers and M. Freitag, "Automated model selection for multivariate anomaly detection in manufacturing systems," *Journal of Intelligent Manufacturing*, pp. 1–19, 2024.
[23] A. M. De Livera, R. J. Hyndman, and R. Snyder, "Forecasting time series with complex seasonal patterns using exponential smoothing," *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1513–1527, 2011.
[24] C. Challu *et al.*, "Nhits: Neural hierarchical interpolation for time series forecasting," *arXiv preprint arXiv:2201.12886*, 2022.