# Driver-Based Multivariate Time Series Forecasting: A Comparative Analysis of Meta-Learning vs Ensemble Performances

Nuno M. C. da Costa[1][a], Filipe Novais[1][b], Luis Santos[1][c], Francisco Franco[1], Vaibhav Shah[1][d], Ricardo Rodrigues[1][e], Duarte Fernandes[1][f] and Emanuel Gouveia[2][g]

[1]*DTx – Digital Transformation CoLAB and the ALGORITMI Center, University of Minho, Guimarães, Portugal*

[2]*Universidade Católica Portuguesa, Instituto de Computação e Ciência de Dados, Portugal*

Keywords: Time Series Forecasting, Meta-Learning, Ensemble Methods, Multivariate Forecasting, Driver-Based Forecasting.

Abstract: Driver-based multivariate time series forecasting aims to predict a target time series - the ”driven” - using multiple influencing variables - the ”drivers”. Traditional linear models often struggle to capture the complex, non-linear relationships in such data. While ensemble methods and meta-learning have been applied, they face limitations like high computational complexity and a narrow focus on model selection. This study addresses these challenges by introducing a new ensemble forecasting approach and extending meta-learning to predict not only the forecasting model but also other critical parameters such as input window size and feature selection methods. We conducted experiments using the publicly available ”Air Quality Monitoring in European Cities” dataset, comparing the proposed ensemble and meta-learning methods across various time series lengths and forecasting horizons (1, 12, and 24 hours). Results demonstrate that the Meta-Learner Forecasting approach outperforms the Ensemble Forecasting approach, especially in smaller datasets and shorter forecasting horizons, achieving improved forecasting accuracy. By extending meta-learning to predict multiple forecasting parameters, this research enhances the versatility and efficiency of multivariate time series forecasting, highlighting the importance of tailoring forecasting parameters to specific data characteristics. The Meta-Learner not only improves accuracy but also reduces computational costs by efficiently narrowing the search space for optimal parameters, making it applicable to more complex forecasting environments.

## 1 INTRODUCTION

Multivariate time series forecasting predicts a target (the “driven” series) using multiple influencing variables (“drivers”), which may be exogenous or endogenous and can include the driven series’ own history. Such problems arise in domains like finance, meteorology, and environmental monitoring. Classical linear methods (e.g., ARIMA (Box et al., 2016)) often fail to capture complex nonlinear dependencies, motivating nonlinear models such as artificial neural networks and ensemble techniques (Tealab, 2018; Gastinger et al., 2021).

Ensemble methods combine model predictions to reduce bias and variance but can be computationally expensive and may not adapt well to heterogeneous data (Gastinger et al., 2021). Our ensemble pipeline emphasises per-series parameter search to find the best forecasting configuration rather than blind aggregation. Given these challenges, (Gastinger et al., 2021) suggested a meta-learning approach as a viable solution, showing promising results.

Meta-learning extracts meta-features from time series and recommends configurations aligned with data characteristics, improving generalization across datasets (Lemke and Gabrys, 2010). Automating configuration selection reduces manual tuning (Lemke and Gabrys, 2010; Talagala et al., 2023) and the need to train numerous candidate models, lowering overfitting risk and computational cost, particularly when historical data is short (Tealab, 2018).

This study introduces an ensemble forecasting

[a] https://orcid.org/0000-0002-8425-3501
[b] https://orcid.org/0000-0003-0717-4292
[c] https://orcid.org/0000-0002-4121-1133
[d] https://orcid.org/0000-0001-9431-8826
[e] https://orcid.org/0000-0001-7986-3754
[f] https://orcid.org/0000-0001-9736-5812
[g] https://orcid.org/0000-0002-7785-2047



da Costa, N. M. C., Novais, F., Santos, L., Franco, F., Shah, V., Rodrigues, R., Fernandes, D. and Gouveia, E.
Driver-Based Multivariate Time Series Forecasting: A Comparative Analysis of Meta-Learning vs Ensemble Performances.
DOI: 10.5220/0014634500004058

In *Proceedings of the 14th International Conference on Model-Based Software and Systems Engineering (MODELSWARD 2026)*, pages 568-575
ISBN: 978-989-758-798-6; ISSN: 2184-4348

pipeline and extends meta-learning beyond model selection to predict forecasting parameters (model choice, input window size, and feature-selection) from series meta-features, while prior work focused on model selection alone (Talagala et al., 2023; Talkhi et al., 2024). We evaluate both approaches on a public air-quality dataset across multiple series lengths and horizons to compare accuracy and computational cost. Developed with industry partners as a driver-based decision-support framework for scenario generation and feature-importance analysis, experiments here use public data because confidentiality prevents releasing proprietary datasets. We show that the driver-based Meta-Learner, which maps meta-features to forecasting configurations, matches or outperforms an exhaustive ensemble while substantially reducing computational search cost in small-sample and short-horizon settings.

The remainder of this paper is structured as follows: Section 2 outlines the methodology used in this study, describing the dataset, features (drivers and driven time series), benchmarking setup, and evaluation metrics. Section 3 provides a detailed explanation of the implemented system, including the different execution modes. In Section 4, we present the experimental results, followed by a discussion in Section 5 that compares the performance of Ensemble Forecasting approach with the Meta-Learning approach across multiple horizons and data sizes. Finally, Section 6 summarizes insights from the study, provides recommendations for future research, and is followed by the acknowledgments and references sections.

# 2 METHODOLOGY

This study aims to compare the forecasting performance of two distinct approaches to driver-based time series forecasting an ensemble and a meta-learner. For this purpose, the recently published 'Air Quality Monitoring in European Cities' dataset was used as the empirical basis for analysis (Angelis et al., 2024). The dataset is publicly available on Zenodo (Angelis, 2024).

## 2.1 Dataset Description

The dataset provides a comprehensive resource for monitoring air pollution dynamics across three European cities. Collected between 2020 and 2023, it features an hourly temporal resolution in 'YYYY-MM-DD HH' format, offering detailed insights into air quality conditions over time. The dataset includes key variables that are crucial for environmental health analysis and forecasting, such as major air pollutants (Ozone, Nitrogen Dioxide, and Particulate Matter), along with meteorological indicators like wind speed (U and V components), air temperature (°C), total precipitation, and relative humidity. These features are the 'drivers' time series in the forecasting system. The data is collected from various monitoring stations across the cities, given in the *station_name* column serving as a unique identifier.

## 2.2 Feature Selection

Given the computational demands of the experiments, it was critical to eliminate redundant or low-impact features to improve processing efficiency. This feature selection process also identified O3 (Ozone concentration) as the most suitable target variable for forecasting due to its clear dependence on meteorological factors like wind speed and temperature. The target feature is the 'driven' time series in the forecasting system.

## 2.3 Benchmarking Setup

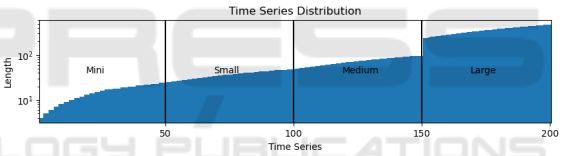


Figure 1: Time series distribution across size categories.

To systematically evaluate model performance, the benchmarking setup was structured around the periodicity and length of the time series, informed by an analysis using both the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF). This analysis identified a clear 24-hour periodic cycle, which aligns with the hourly resolution of the dataset. Based on this periodicity, time series of varying lengths were created to assess model effectiveness at different input window sizes.

The time series were grouped into four temporal scales: **Mini** (0–24h),**Small** (24–48h), **Medium** (48–96h), and **Large** (240–480h).

Each category was planned to have 50 time series, as shown in Figure 1. Given the limited number of monitoring stations in the dataset (fewer than 100), longer time series were segmented into appropriate lengths to meet the target number of 50 per category. This step was necessary to ensure sufficient data points for analysis, nevertheless the care was also taken to avoid bias that could result from over-representation of particular stations.

We evaluated three **horizons (1, 12, 24)**: 1 hour, 12 hours (half periodic cycle) and 24 hours (full periodic cycle).

## 2.4 Evaluation Metrics

To evaluate model performance, a combination of Mean Absolute Error (MAE) and Mean Absolute Scaled Error (MASE) metrics was applied due to their complementary properties. MAE provides an intuitive measure of average forecast error in the same units as the data, facilitating straightforward interpretation. MASE normalises the error relative to a naive benchmark, enabling scale-independent comparisons across different datasets. Together, these metrics offer both absolute and relative performance assessments, ensuring fair evaluation regardless of the data's scale or characteristics.

These metrics were applied to three distinct forecasting approaches: Ensemble Forecasting, Meta-Learner Forecasting and Naive Forecasting. All methods used the same set of time series across the defined horizons to ensure results' comparability. However, for the 'Mini' and 'Small' categories, certain metrics for larger horizons could not be calculated, as the Ensemble Forecasting approach uses the forecast horizon for validation, and at least two samples are required to train a model.

Overall, this benchmarking setup provided a solid framework to evaluate the effectiveness of the proposed ensemble and meta-learning methods for driver-based time series forecasting in air quality monitoring.

# 3 SYSTEM IMPLEMENTATION

Figure 2 shows the system architecture as modular pipelines that share preprocessing, training, and forecasting stages. Preprocessing handles missing data (resampling, interpolation), followed by validation training for 'Ensemble Forecasting' and 'Train Meta-Learner' pipelines modes or final training for 'Meta-Learner Forecasting' pipeline. Forecasting is multistep and driven by the user-specified horizon. The pipelines produce scenario forecasts, driver importance rankings, and exportable model configurations for reproducible deployment, enabling integration with MBSE toolchains for traceability and what-if analyses.

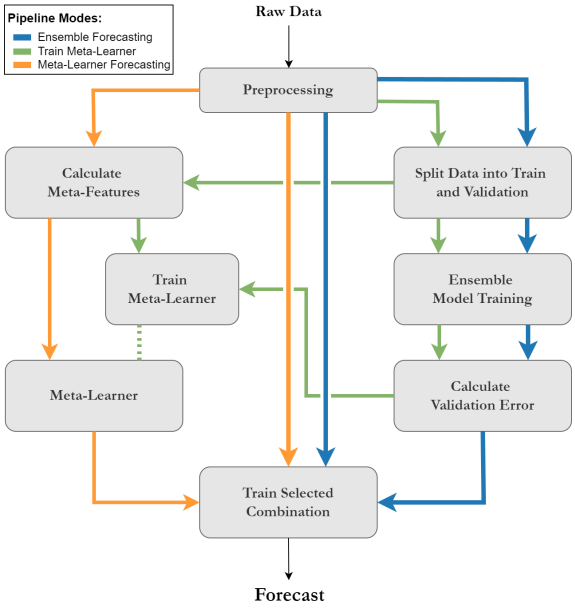


Figure 2: Overview of the pipeline workflow illustrating the sequential steps in the forecasting process.

## 3.1 System Execution Modes

The system provides three compact execution modes: (1) **Ensemble Training & Forecasting** — exhaustive search over forecasting parameter combinations on the "validation set" to determine the optimal combination based on validation error, through training and testing of labelled data with retrain on the full series using the best validation configuration; (2) **Train Meta-learner**, — build a model that maps series meta-features to a probability distribution over forecasting parameter combinations (multi-class classification), also the "validation set" is leveraged to identify the best parameter combination for each time series' meta-features; (3) **Meta-Learner Forecasting** — deploy the trained Meta-Learner to predict parameters for new series and run final forecasts. All modes share preprocessing and produce the same deployable artefacts (models, configurations, and importance scores).

## 3.2 Ensemble Forecasting Pipeline

### 3.2.1 Ensemble of Forecasting Algorithms

The primary goal of the Ensemble Forecasting pipeline is to evaluate all possible combinations of forecasting parameters and select the optimal configuration based on a validation set that matches the forecast horizon specified by the user. This approach assumes that the parameter combination yielding the

best performance on the validation set will also perform optimally when retrained on the full dataset, without reserving additional data for validation. By aligning the validation set length with the forecast horizon, the pipeline ensures that the model is tuned to the user's specific forecasting objectives.

#### 3.2.2 Training Set Preparation

The process begins by scaling and splitting the data with a predefined scaler. The validation set is selected to match the user-defined forecast horizon. After training on the initial training set, the validation error is calculated using a chosen metric, with MAE being the default. The combination of forecasting parameters that achieves the lowest validation MAE is then retrained on the full dataset using the same parameters.

This approach provides a comprehensive evaluation of different parameter combinations, ensuring the final configuration is tailored to the dataset and forecasting task. A full list of the forecasting parameters explored, including feature selection methods, input window sizes, and forecasting models, can be found in the appendix.

### 3.3 Train Meta-Learner Pipeline

The main goal of the Train Meta-Learner pipeline is to build a Meta-Learner model that can predict the optimal combination of forecasting parameters for a given time series. Like the Ensemble Forecasting pipeline, the data is split to create a validation set matching the forecast horizon. However, instead of selecting the parameter combination with the lowest validation error, this pipeline stores the validation error for all parameter combinations. This information is used to assign probabilities to each combination, which will serve as target data for training the Meta-Learner. The meta-features of each time series serve as the inputs to the model, while the target variable is the probability of selecting each parameter combination. Once trained, the Meta-Learner model is stored and used in the subsequent Meta-Learner Forecasting pipeline.

#### 3.3.1 Meta-Features

The meta-features are used as input to train the Meta-Learner and were carefully chosen to capture key characteristics of the time series and its drivers, as shown in Table 1.

These meta-features compactly capture structural properties (periodicity, trend, variability, correlations) for both drivers and the target, giving the Meta-Learner informative inputs to predict suitable forecasting configurations. Meta-features are computed from the training portion of each series and may contain missing values when drivers are absent.

Table 1: Meta-features used for each driver and for the target time series.

| For each driver | For the target time series |
|---|---|
| • Mean<br>• Standard deviation<br>• Skewness<br>• Kurtosis<br>• Correlation with the target time series | • The length of the time series<br>• The dominant period<br>• Trend strength<br>• Seasonality strength<br>• Residual standard deviation<br>• Outlier count |

#### 3.3.2 Forecasting Parameters

The key forecasting parameters explored in this study include: Input Window Size; Feature selection method; Forecasting Model.

The feature selection methods explored in this study included the Variance Inflation Factor (VIF) to remove highly correlated features and Shapley-based (SHAP) feature selection to identify the most relevant features (Galvão and Araújo, 2009; Murray et al., 2012). Additionally, options such as not applying any feature selection, as well as a univariate feature selection approach that considers only the target time series, were also evaluated.

The input window sizes were selected based on the time series periodicity within mini, small, medium and large categories, aiming to determine the optimal historical context for accurate forecasting. The window sizes were chosen to avoid being excessively large, as this could potentially overload the models with information and result in prolonged training times.

The forecasting models evaluated cover a broad range of techniques, from classical methods such as AutoARIMA (Hyndman and Khandakar, 2008) and simpler approaches like linear regression and spline models, to tree- and ensemble-based methods (Random Forest, XGBoost (Chen and Guestrin, 2016), CatBoost (Prokhorenkova et al., 2018)), regularized linear models (Lasso (Tibshirani, 2018)), SVM regression (Drucker et al., 1996), and deep learning architectures such as multilayer perceptrons (Murtagh, 1991) and long short-term memory networks (Hochreiter and Schmidhuber, 1997).

The process of training the Meta-Learner involves evaluating parameter combinations on the validation set and using the result to generate a probability distribution over these combinations as the training target, as described in the previous section. To gener-

ate this training target, for each forecasting parameter, each validation error from the highest validation error was subtracted, effectively inverting the errors so that better-performing models receive higher scores. The combination with the highest error is assigned a zero value, and all the scores are normalised to ensure the probabilities sum to one. This approach emphasizes parameter combinations with lower validation errors by assigning them higher probabilities, guiding the Meta-Learner to prioritize the best-performing forecasting parameters during training.

Table 2 summarizes the explored forecasting parameters and example categories used in our experiments.

Table 2: Explored forecasting parameters and example categories.

| Parameter | Example categories |
|---|---|
| Input window size | 0-14 mini, 24-48 small, 48-96 medium, 240+ large (hours) |
| Feature selection | None, VIF, SHAP, Univariate |
| Forecasting model | AutoARIMA, Lasso, RandomForest, XGBoost, MLP, LSTM |
| Validation metric | MAE, MASE |

### 3.3.3 Meta-Learner Model

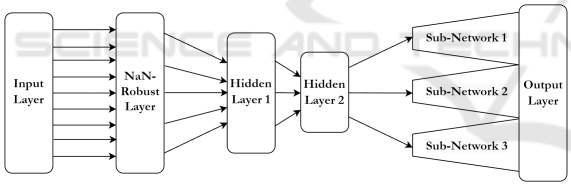


Figure 3: Neural network architecture of the Meta-Learner.

The Meta-Learner is essentially a neural network designed to predict the optimal combination of forecasting parameters based on meta-features extracted from time series data. It addresses key challenges such as handling missing values (NaNs) in the meta-features, performing simultaneous multiclass classification for multiple parameters, and implementing overfitting reduction strategies to enhance generalization, as illustrated in Figure 3.

The overall architecture comprises: **Input Layer**, receives tensor with meta-feature data; **NaN-Robust Layer**, processes the input while handling NaNs; **Hidden Layer 1**, a fully connected layer that reduces dimensionality, followed by batch normalization, ReLU activation, and dropout for regularization; **Hidden Layer 2**, another fully connected layer that further reduces dimensionality, also followed by batch normalization, ReLU activation, and dropout; **Sub-Networks**, after the shared hidden layers, the network branches into three separate sub-networks for each forecasting parameter - feature selection method, input window size, and forecasting model. Each Sub-Network consists of fully connected layers leading to an output layer that provides logits for the respective parameter categories; **Output Layer**, this layer doesn't include neurons, it combines the outputs from all sub-networks to deliver the final result, supporting simultaneous predictions for every forecasting parameter.

The Meta-Learner's architecture includes a custom layer called the NaN-Robust Layer, designed to effectively handle meta-features containing missing values without requiring imputation or discarding incomplete samples. This layer creates a mask to identify NaN values in the input tensor and adjusts the corresponding weights and biases during the forward pass. By ignoring the weights and biases associated with NaN inputs, the network prevents the propagation of invalid values through its computations, allowing the model to robustly process incomplete tensors.

The Meta-Learner simultaneously addresses three multiclass classification tasks within a single model. By sharing the initial layers (Hidden Layers 1 and 2) among all tasks, the network captures common patterns in the meta-features, while specialized sub-networks focus on the specifics of each forecasting parameter. To ensure balanced learning, the Cross-Entropy Loss function was used for each classification task, and total loss was calculated as the average of individual losses. The model is optimized using the Adam optimizer, known for its efficiency in training deep neural networks.

To enhance generalization and reduce overfitting, several strategies are employed. Dropout regularization with a 30% rate is applied after each initial layer to prevent over-reliance on specific neurons, and batch normalization after linear layers stabilizes the learning process by normalizing inputs, thereby accelerating training and improving performance. Weight decay is implemented in the optimizer to penalize large weights, while gradient clipping prevents exploding gradients by limiting the magnitude of updates during backpropagation. Early stopping monitors the validation loss and halts training when improvements plateau, preventing overfitting, and learning rate scheduling adjusts the learning rate based on validation loss, reducing it when the loss plateaus to allow finer adjustments as the model converges.

Overall, the architecture, loss function, optimizer, and combined overfitting reduction techniques assist

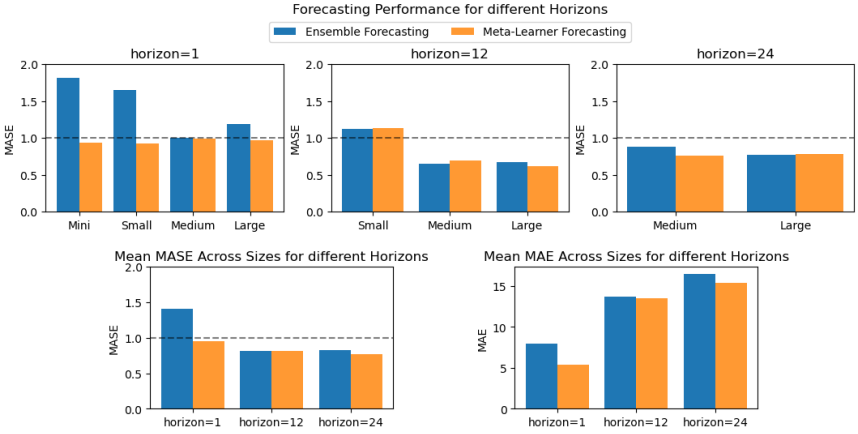


Figure 4: Forecasting performance across different horizons, illustrating the MASE and MAE error for each size per horizon and the final average across all sizes.

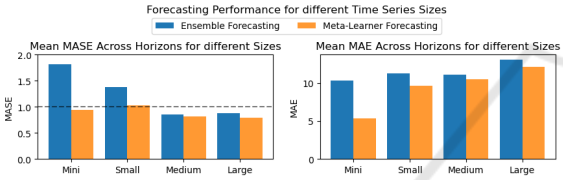


Figure 5: Forecasting performance across varying time series sizes (average MASE and MAE).

the Meta-Learner in fitting the training data appropriately while retaining generalization to unseen data.

## 3.4 Meta-Learner Forecasting Pipeline

The Meta-Learner Forecasting pipeline leverages the trained Meta-Learner model to predict the optimal combination of forecasting parameters for new time series data, streamlining the process by eliminating the need for exhaustive validation across all parameter combinations. The process begins with a preprocessing phase, where the new time series data is prepared for analysis. This includes handling missing data through resampling and interpolation, as well as extracting the same set of meta-features mentioned in section 3.3.1. Once the meta-features are extracted, they are input into the trained Meta-Learner, which predicts the most suitable forecasting parameters: the feature selection method, input window size, and forecasting model.

The final step involves applying the optimal combination of forecasting parameters, as predicted by the Meta-Learner, to the actual time series data, training the models, and producing the forecasts.

# 4 EXPERIMENT RESULTS

The Ensemble and Meta-Learner forecasting methods were evaluated across different time series sizes—Mini, Small, Medium, and Large—and forecasting horizons of 1, 12, and 24 hours. Prediction accuracy was assessed using MASE and MAE.

To characterise the variability of results and improve robustness reporting, key experiments were repeated over multiple random train/validation/test splits and we report mean ± standard deviation for the main metrics (MAE and MASE). These repeated-split checks help to expose sensitivity to data partitioning and provide a concise measure of variability.

As presented in Figure 4, at Horizon 1, the Ensemble Forecasting yielded MASE values exceeding 1 for Mini and Small datasets, indicating performance worse than the Naive Forecasting method. For Medium and Large datasets, the Ensemble's MASE values dropped below 1, demonstrating improved accuracy. The Meta-Learner consistently achieved MASE values below 1 across all dataset sizes.

MAE increased with longer forecasting horizons for both approaches. At Horizon 1, both models exhibited low MAE, with the Meta-Learner Forecasting outperforming the Ensemble Forecasting in Mini and Small datasets. As horizons extended to 12 and 24 hours, MAE rose for both models, and the performance gap narrowed at Horizon 24.

Across all horizons, the Meta-Learner Forecasting maintained lower MASE and MAE values compared to the Ensemble Forecasting, especially in smaller datasets, as demonstrated in Figure 5. The Ensemble's performance improved with larger time series sizes.

In the Ensemble Forecasting approach, a diverse

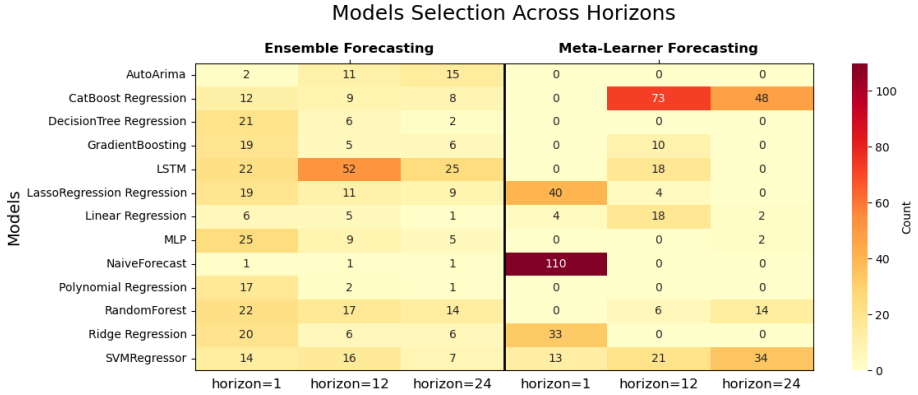


Figure 6: Heatmap displaying the model selection distribution for two approaches: the Ensemble Forecasting and the Meta-Learner Forecasting. The color intensity represents the frequency with which each model is selected, highlighting the similarities.

range of models was selected, as shown in Figure 6. At Horizon 1, Long Short-Term Memory (LSTM), Decision Tree, and Multilayer Perceptron (MLP) were frequently chosen. As the horizon increased, there was a shift toward simpler models like Lasso Regression and Random Forest. In the Meta-Learner Forecasting approach, Naive Forecasting was predominantly selected at Horizon 1, while more complex models such as CatBoost and Lasso Regression were incorporated at longer horizons.

## 5 DISCUSSION

The Meta-Learner approach demonstrated robust performance across all dataset sizes. The Meta-Learner excels in small-sample and short-horizon settings by adapting configurations to data characteristics. While the Ensemble benefits from larger datasets where exhaustive search yields stronger candidates. This outcome suggests that the Meta-Learner's dynamic parameter adjustment enhances generalization and reduces overfitting, particularly in smaller datasets. Model-selection patterns indicate the Ensemble favors complex models for short horizons and simpler models for longer horizons. While the Meta-Learner balances complexity to trade off accuracy and computational cost by selecting simpler models for short-term predictions and incorporating complex models for longer horizons.

These findings align with the objectives of understanding how each forecasting framework handles varying time series sizes and forecasting horizons. Compared to state-of-the-art methods, the Meta-Learner's adaptability offers improved performance in scenarios with limited data, while the Ensemble gains efficacy with larger datasets.

However, despite these promising results, the study has certain limitations. It primarily focuses on model selection within the Meta-Learner framework, potentially overlooking the impact of other forecasting parameters—such as feature selection and input window size—that should be evaluated as well. Other limitations include a single-domain dataset and one Meta-Learner architecture. Future work will broaden domains (e.g., energy, finance), run ablation studies on forecasting parameters, and evaluate alternative meta-learner families to validate and extend these findings.

## 6 CONCLUSION

The study demonstrates that the Meta-Learner approach outperforms the Ensemble method in driver-based time series forecasting, particularly for smaller sample sizes and shorter forecasting horizons. The Meta-Learner's ability to dynamically adapt model selection based on data characteristics ensures optimal forecasting accuracy across diverse conditions. In contrast, the Ensemble approach, although versatile with larger datasets, struggles to generalize effectively with smaller datasets and shorter horizons.

These findings highlight the importance of tailoring forecasting models to specific meta-features of the time series data. The Meta-Learner not only improves accuracy, but also reduces computational costs by efficiently narrowing the search space and identifying optimal parameters. This resource-efficient solution underscores its potential applicability to more complex forecasting environments.

Future work will evaluate additional forecasting parameters (scaling, hyperparameters, optimizers) and alternative meta-learner families (boosting,

tree-based, probabilistic methods), and will test robustness across domains (e.g., energy, finance) to broaden external validity.

## ACKNOWLEDGMENTS

This work was supported under the base funding project of the DTx CoLAB - Collaborative Laboratory, under the Missão Interface of the Recovery and Resilience Plan (PRR), integrated in the notice 01/C05-i02/2022, which aims to deepen the effort to expand and consolidate the network of interface institutions between the academic, scientific and technological system and the Portuguese business fabric. We thank our Portuguese industry partners for their collaboration. To protect intellectual property, some experiments were performed on confidential datasets and are not reported here. The experiments and results presented in this paper were produced using publicly available data to ensure reproducibility.

## REFERENCES

Angelis, G.-F. (2024). Regional datasets for air quality monitoring in european cities. Datasets Repository DOI: 10.5281/zenodo.11220965.

Angelis, G.-F., Emvoliadis, A., Theodorou, T.-I., Zamichos, A., Drosou, A., and Tzovaras, D. (2024). Regional datasets for air quality monitoring in european cities. *IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2016). *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, Hoboken, NJ, 5th edition.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. Association for Computing Machinery.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A., and Vapnik, V. (1996). Support vector regression machines. In Mozer, M., Jordan, M., and Petsche, T., editors, *Advances in Neural Information Processing Systems*, volume 9. MIT Press.

Galvão, R. and Araújo, M. (2009). 3.05 - variable selection. In Brown, S. D., Tauler, R., and Walczak, B., editors, *Comprehensive Chemometrics*, pages 233–283. Elsevier, Oxford.

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

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

Hyndman, R. J. and Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software*, 27(3):1–22.

Lemke, C. and Gabrys, B. (2010). Meta-learning for time series forecasting and forecast combination. *Neurocomputing*.

Murray, L., Nguyen, H., Lee, Y.-F., Remmenga, M. D., and Smith, D. W. (2012). Variance inflation factors in regression models with dummy variables. In *Conference on Applied Statistics in Agriculture*.

Murtagh, F. (1991). Multilayer perceptrons for classification and regression. *Neurocomputing*, 2(5):183–197.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2018). Catboost: unbiased boosting with categorical features. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Talagala, T. S., Hyndman, R. J., and Athanasopoulos, G. (2023). Meta-learning how to forecast time series. *Journal of Forecasting*, 42(6):1476–1501.

Talkhi, N., Akhavan Fatemi, N., Jabbari Nooghabi, M., Soltani, E., and Jabbari Nooghabi, A. (2024). Using meta-learning to recommend an appropriate time-series forecasting model. *BMC Public Health*, 24(1):148.

Tealab, A. (2018). Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal*, 3(2):334–340.

Tibshirani, R. (2018). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.